

Metodología de evaluación de Sistemas de Gestión de Procesos de Negocio

Informe de proyecto de grado



Alexandra Castelli
Germán Lagrega
Bettina Neira

Instituto de Computación
Universidad de la República

Tutor: Andrea Delgado
Co-tutor: Daniel Calegari

Julio 2016

Resumen

Los Sistemas de Gestión de Procesos de Negocio (BPMS, del inglés Business Process Management Systems) proveen soporte para el modelado, ejecución y evaluación de los procesos de negocio de una organización. La selección de un BPMS debe tener en cuenta principalmente los requerimientos de la organización, sin descuidar los aspectos técnicos y comerciales.

Enmarcado en la línea de investigación del grupo Componentes.: Objetos.: Arquitecturas.: Lenguajes (COAL) y como parte de la evolución de proyectos anteriores que han evaluado BPMS gratuitos y comerciales, centrándose principalmente en aspectos funcionales, el presente proyecto de grado propone ampliar la metodología de evaluación de BPMS perteneciente al grupo COAL. El principal objetivo es incorporar aspectos no funcionales de los BPMS, además de aplicar la metodología actualizada para la evaluación de una selección de BPMS existentes tanto *open source* como comerciales. Finalmente se busca generar una aplicación web que asista al proceso de evaluación.

Se llevó a cabo un estudio de metodologías de evaluación de BPMS, comparándolas con la metodología desarrollada por el grupo COAL, obteniendo así una lista de carencias y virtudes que permitió incorporar aspectos no funcionales, así como un método para su evaluación, además de un plan de pruebas que permite organizar y sistematizar los pasos a seguir dentro del proceso de evaluación de las herramientas.

Como resultado se obtuvo una evaluación detallada de nueve herramientas, utilizando la lista de características de la metodología actualizada a partir de un proyecto previo. Para realizar las evaluaciones se implementó un caso de estudio y casos de pruebas que implementen las contrucciones habituales en un proceso de negocio. Los resultados cuantitativos obtenidos son la entrada de la aplicación web. La aplicación permite administrar: usuarios, roles, categorías, características, subcaracterísticas, herramientas y sus versiones. Además es posible ingresar resultados cuantitativos, obtener evaluaciones ponderando las subcaracterísticas y comparar los resultados de dos herramientas.

Índice general

1. Introducción	1
1.1. Contexto y motivación	1
1.2. Descripción del proyecto	2
1.3. Objetivos	2
1.4. Alcance	3
1.5. Organización del documento	3
2. Estado del arte	5
2.1. Conceptos y definiciones	5
2.1.1. Proceso de Negocio	5
2.1.2. Business Process Managment	6
2.1.3. Ciclo de vida	6
2.1.4. Business Process Managment System	8
2.2. Metodología de evaluación de BPMS (Delgado et al., 2015, [1])	10
2.3. Otras metodologías de Evaluación	13
2.3.1. Selection criteria for Tools Supporting Business Process Transformations for Electronic Commerce (Tsalgatidou et al., 1998, [2])	14
2.3.2. COTS Evaluation Using Desmet Methodology & Analytic Hierarchy Process (AHP) (Morera, 2002, [3])	16
2.3.3. A Formal Process for Evaluating COTS Software Products (Lawlis et al., 2001, [4])	18
2.3.4. OpenBQR: a framework for the assessment of OSS (Taibi et al., 2007, [5])	20
2.3.5. Evaluation and Selection COTS Software Process: The State of the Art (Tarawneh et al., 2011, [6])	22
2.3.6. Comparación de otras propuestas con la metodología COAL	24
2.4. Resumen	29

3. Características a evaluar	31
3.1. Introducción	31
3.2. Extensión de la metodología: Plan de pruebas	31
3.3. Casos de prueba	33
3.4. Enfoque sobre la selección de características	34
3.5. Estructura de las características	34
3.6. Clasificación general de características	35
3.6.1. Definición de características técnicas no funcionales	35
3.7. Método de evaluación de características no funcionales	37
3.8. Resumen	38
4. Evaluación de características	39
4.1. Introducción	39
4.2. Selección de herramientas	40
4.3. Evaluación de herramientas	41
4.3.1. Red Hat JBoss BPM Suite	41
4.3.2. Bonita BPM	43
4.3.3. Intalio/bpms	45
4.3.4. Activiti	49
4.3.5. Bizagi BPM Suite	52
4.3.6. Camunda	56
4.3.7. Aris Express	60
4.3.8. Orchestra	62
4.3.9. ProcessMaker	64
4.3.10. Comparación cuantitativa de resultados	67
4.4. Resumen	69
5. Caso de estudio	71
5.1. Introducción	71
5.2. Caso de Estudio	72
5.2.1. Solicitud de licencia vacacional	72
5.3. Análisis del Caso de Estudio	75
5.3.1. JBoss BPM Suite	75
5.3.2. Bonita BPM	77
5.3.3. Intalio/bpms	78
5.3.4. Activiti	79
5.3.5. Bizagi Suite	81

Índice general	v
5.3.6. Camunda BPMN	82
5.3.7. Orchestra	84
5.3.8. ProcessMaker	84
5.4. Resumen	86
6. Proyecto web	87
6.1. Requerimientos de la aplicación web	87
6.1.1. Casos de uso	87
6.2. Diseño	89
6.3. Implementación	91
6.4. Resumen	96
7. Conclusiones y trabajo a futuro	97
7.1. Conclusiones	97
7.2. Trabajo futuro	99
Bibliografía	101

Capítulo 1

Introducción

El grupo Componentes.: Objetos.: Arquitecturas.: Lenguajes (COAL), entorno a la Gestión y Tecnologías de Procesos de Negocio ha desarrollado una metodología de evaluación de Sistemas de Gestión de Procesos de Negocio (BPMS, del inglés Business Process Management Systems). Estos sistemas proveen soporte para el modelado, ejecución y evaluación de los procesos de negocio de una organización.

Enmarcado en dicha línea de investigación y como parte de la evolución de proyectos anteriores que han evaluado BPMS gratuitos y comerciales, centrándose principalmente en aspectos funcionales, este proyecto busca ampliar la metodología, trazándose como objetivo principal incorporar aspectos no funcionales de los BPMS.

La selección de un BPMS requiere de una evaluación detallada de sus capacidades. Al extender esta metodología se pretende ayudar a la organización que la implemente a elegir los BPMS que mejor se adecuen a sus requerimientos.

1.1. Contexto y motivación

En 2010, se realizaron dos proyectos de grados centrados en herramientas *open source* [7, 8]. Uno de los proyectos se enfocó en la evaluación de motores XPDL, mientras que el otro proyecto llevó a cabo la evaluación de motores WS-BPEL, incluyendo el estándar BPMN 2.0 en sus versiones iniciales. Estos proyectos lograron definir una lista de características inicial y una metodología de evaluación informal.

En segunda instancia, dos proyectos enmarcados en el COAL se enfocaron en desarrollar una metodología de evaluación de BPMS. En 2011, se realizó la unificación de las características y la metodología, aplicándola a productos uruguayos [9]. Por otro lado, el proyecto realizado en 2013 llevó a cabo una reestructura de la lista de características y de la metodología,

además se realizó una evaluación de herramientas comerciales en general [1].

La constante evolución de los BPMS en los últimos diez años ha llevado a diferentes organizaciones a incorporarlos, esto ha motivado la realización de una metodología que permita evaluar de forma rápida varias herramientas de acuerdo a las necesidades de la organización. En este sentido, el grupo COAL en los últimos seis años ha desarrollado una metodología de evaluación de BPMS que permite a las organizaciones obtener de forma objetiva herramientas que se adapten a sus necesidades, este proyecto de grado busca ampliar dicha metodología. Desarrollar una metodología de evaluación de BPMS no es una tarea sencilla, se debe tener en cuenta aspectos relacionados a los BPMS en sí y a las organizaciones que buscan incorporar estos sistemas.

1.2. Descripción del proyecto

El grupo COAL cuenta con una metodología de evaluación de BPMS estable, sin embargo, la metodología no tiene en cuenta aspectos no funcionales de los BPMS.

Este proyecto busca ampliar la lista de criterios, incorporando criterios no funcionales que se adecuen a los BPMS, a través de una evaluación comparativa con otras metodologías de evaluación de software con el fin de abarcar la mayor cantidad de criterios de decisión posible. De esta manera, distintas organizaciones podrán utilizar la evaluación ponderando los criterios según sus necesidades.

Se aplicará la metodología actualizada para evaluar distintos tipos de BPMS y se desarrollará una aplicación web que asista al proceso de evaluación.

1.3. Objetivos

El objetivo de este proyecto es extender la metodología considerando principalmente aspectos no funcionales (performance, seguridad, usabilidad, etc.), así como otros aspectos de interés involucrados en metodologías de evaluación existentes.

Los objetivos específicos son:

1. Realizar una evaluación comparativa de la metodología existente con otras metodologías de evaluación de software.
2. Extender la metodología de evaluación de BPMS con la inclusión de aspectos no funcionales, generando un benchmark que uniformice las evaluaciones.
3. Aplicar la metodología actualizada para la evaluación de una selección de BPMS existentes tanto open source como comerciales.
4. Generar una aplicación web que asista al proceso de evaluación.

1.4. Alcance

El alcance del proyecto cubrió el estudio de metodologías de evaluación de software, un análisis y estudio comparativo con la metodología existente, la extensión de la metodología con la incorporación de elementos no funcionales. Además de la aplicación de la metodología a nueve herramientas utilizando la nueva lista de características y el desarrollo de una aplicación web de soporte al proceso de evaluación.

1.5. Organización del documento

Este documento está dividido en capítulos, a continuación se describe el contenido de los capítulos que siguen:

En el capítulo 2 se aborda el estado del arte donde se presentan conceptos relevantes para entender el proyecto. Así como también otras metodologías investigadas.

En el capítulo 3 se describe el plan de pruebas propuesto como parte de la actualización de la metodología descrita en [1] y se brinda una breve descripción de las categorías incorporadas.

En el capítulo 4 se describe la metodología completa y una selección de los resultados obtenidos de las evaluaciones de cada herramienta.

El capítulo 5 especifica el caso de estudio realizado. Incluye el análisis del caso para cada herramienta.

El capítulo 6 refiere a la aplicación web, muestra su arquitectura y funcionalidades.

En el capítulo 7 se presentan las conclusiones del proyecto, también brinda los lineamientos para realizar trabajos futuros.

El documento cuenta con una serie de anexos en los cuales se incluye información detallada sobre cada capítulo.

En el Anexo A se definen todas las características que pertenecen a la metodología.

En el Anexo B se describe la metodología completa y los resultados de las evaluaciones de cada herramienta.

El Anexo C especifica el caso de estudio realizado. Incluye el análisis completo del caso para cada herramienta.

El Anexo D refiere a la aplicación web, muestra los casos de uso.

Capítulo 2

Estado del arte

En este capítulo se abordan los temas tomados como base para la implementación del proyecto. Conceptos y definiciones de los procesos de negocio, origen y evolución de éstos, y el estado actual de la metodología a expandir son alguno de ellos.

A su vez se presentan metodologías de evaluación en general, de software y específicamente metodologías de evaluación de BPMS.

Se profundiza en particular respecto a las metodologías de evaluación de BPMS actuales haciendo una comparación entre ellas en los aspectos más relevantes.

En la sección 2.1 se desarrollan los conceptos y definiciones antes mencionados. En la sección 2.2, se describe la metodología de evaluación que existe actualmente. La sección 2.3 presenta la descripción de otras metodologías de evaluación estudiadas, así como una comparación con la metodología actual. Por último, en la sección 2.4 se brinda un resumen del capítulo.

2.1. Conceptos y definiciones

En esta sección se presentan conceptos y definiciones que favorecen al mejor entendimiento del proyecto. En este sentido, se abordan conceptos tales como proceso de negocio, gestión de proceso de negocio, ciclo de vida.

2.1.1. Proceso de Negocio

Un Proceso de Negocio (BP, del inglés Business Process) es un conjunto de actividades que se realizan coordinadas en un ámbito organizacional y técnico, las cuales cumplen con un objetivo de negocio. En conjunto, los procesos de negocio se vuelven claves para una organización, por lo que se deben definir de la mejor manera posible, y para esto la informática es fundamental, dado que informatizando dichos procesos la organización se

asegura de cumplir con las restricciones y/o reglamentos que cada proceso posee [10].

Las organizaciones en general llevan a cabo varios procesos de negocio diariamente. Hay procesos de negocio internos y procesos que interactúan con procesos de otras organizaciones.

A un proceso de negocio se lo puede ver como una serie discreta de actividades o pasos de tareas que pueden incluir personas, aplicativos, eventos de negocio, tareas y organizaciones [11].

2.1.2. Business Process Managment

La Gestión de Procesos de Negocio (BPM, del inglés Business Process Managment) es influenciada por conceptos y tecnologías de diferentes áreas de la administración de empresas y ciencias de la computación. Basado en los primeros trabajos de organización y gestión, tiene sus raíces en la tendencia “diseño orientado a procesos” que tuvo su auge en la década de 1990, cuando se propuso una nueva forma de organización en las empresas sobre la base de los procesos de negocio [10].

BPM se lo puede ver como el entendimiento, la visibilidad, el modelado y control de los procesos de negocio de una organización. Su objetivo es la descomposición de la actividad global en un conjunto de “Procesos” cuyas acciones repetitivas puedan ser automatizadas. Esto permite optimizar tiempos, oportunidades y costos, sin perder la capacidad de adaptación constante y rápida a los cambios. Todo esto conservando la coexistencia de métodos seguros con la necesaria flexibilidad para facilitar la intervención activa y fundamental de las personas en los procesos.

2.1.3. Ciclo de vida

Según Weske [10] el ciclo de vida de un proceso de negocio consiste de cuatro fases relacionadas entre sí. Dichas fases se muestran en la figura 2.1.

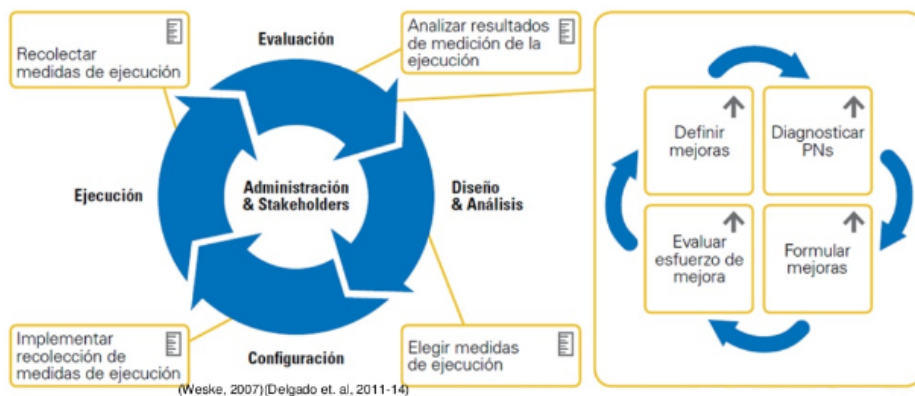


Figura 2.1 Ciclo de vida de los procesos de negocio

En ésta se muestran las fases tomadas de [10] con la extensión del artículo [12] que destaca el aspecto de medidas de ejecución y ciclo de mejora continua. Las fases se organizan en una estructura cíclica, la cual no implica un orden temporal para ser ejecutadas. Muchas actividades de diseño y desarrollo se llevan a cabo en cada una de estas fases. No son inusuales enfoques incrementales y evolutivos relacionados con las actividades concurrentes en múltiples fases.

2.1.3.1. Diseño y Análisis

El objetivo de esta fase es obtener los modelos de proceso. Esto se logra a través de estudios sobre cómo se llevan a cabo los procesos de negocio y el ambiente organizacional y técnico en el cual se realizan. Se debe de hacer hincapié en los detalles para conocer cómo funcionan dichos procesos.

Durante esta fase, son usadas técnicas de modelado de proceso de negocio, así como también validación, simulación y verificación.

Las técnicas de simulación pueden ser utilizadas para apoyar la validación del modelo, por ejemplo, pueden simularse ciertas secuencias de ejecución no deseadas con el fin de ver los déficits en el modelo de proceso. La simulación de procesos de negocio también permite a los interesados ir a través del proceso paso a paso mostrando como sería el comportamiento en la realidad, en distintos escenarios de ejecución [10].

2.1.3.2. Configuración

En esta fase se incluye al modelo de proceso todas las particularidades necesarias para la óptima ejecución y aprovechamiento de funcionalidades del software BPM seleccionado. Esto puede hacerse mediante políticas y procedimientos propios de la organización.

En caso de utilizar un BPMS, este tiene que ser configurado de acuerdo al contexto de la empresa, esto incluye la interacción de los empleados con el sistema, así como la integración con los sistemas existentes. Una vez que el sistema es configurado, la implementación tiene que ser testeada.

Entre las tareas que se realizan en esta etapa se encuentran los controles de formulario y el diseño, así como el contenido del mensaje de correo electrónico y la integración de sistemas.

2.1.3.3. Ejecución

Una vez completada la fase de configuración, las instancias de proceso de negocio pueden ser ejecutadas. Durante esta fase, se lleva a cabo la ejecución real del proceso. Se inician las instancias de proceso de negocio para cumplir los objetivos de la compañía. El inicio de una instancia de proceso, típicamente ocurre luego de un evento definido, por ejemplo, la recepción de una orden enviada por un cliente. El BPMS controla activamente la

ejecución de las instancias de proceso definidas en el modelo. El estado de dichas instancias e información precisa se puede visualizar a través de un componente de monitoreo del BPMS. Esa información es útil por ejemplo, para responder a una petición de un cliente sobre el estado de su orden [10].

2.1.3.4. Evaluación

Esta fase es para la evaluación del proceso construido en las etapas anteriores. Para esto se utiliza la información disponible y se tiene como objetivo mejorar los modelos de procesos de negocios y sus implementaciones. Por ejemplo, se analizan logs de eventos registrados en ejecución, esto permite un análisis “on-line” de la ejecución de procesos. Al hacer eso encontramos oportunidades de mejora de los modelos y sistemas que los implementan. La fase de evaluación está estrechamente relacionada con la simulación de los procesos de negocio, ya que comparten información importante [10].

2.1.4. Business Process Managment System

Los BPMS son software genérico (también podemos verlos como un sistema de información genérico basado en procesos) que apuntan a dar soporte a todo el ciclo de vida de un proceso de negocio [10].

Un BPMS es una aplicación o un conjunto de aplicaciones que está destinada a simplificar los procesos de negocio y flujos de trabajo con el fin de ser más eficiente y adaptarse a entornos cambiantes.

Mediante los BPMS las empresas pueden gestionar los ciclos de vida aplicando las definiciones propias y mantener las mejores prácticas en sus procesos. Como tal, BPMS es una herramienta valiosa para el desarrollo y la mejora de los procesos de negocio.

Un BPMS tiene un conjunto de herramienta que le permiten definir, automatizar, ejecutar, administrar y mejorar los procesos de negocio. Estas herramientas son:

- **Modelador de procesos:** Es una herramienta gráfica que permite a los analistas de proceso definir procesos en términos de flujo de actividades, reglas de negocio y flujo de información.
- **Herramienta de simulación de procesos:** Permite a los analistas de proceso simular condiciones reales e identificar cuellos de botella o requerimientos de recursos. De esta forma pueden predecir y corregir problemas de negocio en la etapa de diseño.
- **Kit de herramientas de desarrollo de aplicaciones:** Aplicaciones de procesos de negocios construidos con BPMS requieren esfuerzos de desarrollo de sistemas para la creación de interfaz de usuario (formularios) y lograr la integración de los motores

BPM (servidores de workflow y reglas de negocio) con las aplicaciones empresariales desplegadas dentro y fuera de los límites de la organización. Kit de herramientas de desarrollo de proceso proporciona a los desarrolladores un conjunto de herramientas, que pueden reducir al mínimo los esfuerzos de desarrollo del sistema.

- Motor de Workflow y reglas de negocio: Ejecuta flujo de actividades, flujo de información y reglas de negocio. Asigna tareas a los ejecutantes basado en el flujo de actividad y reglas definidas e interopera con otros sistemas backend.
- Interfaz de usuario de ejecución de procesos: Proporciona un entorno en tiempo de ejecución a los ejecutantes de proceso. Esta interfaz permite a los usuarios iniciar las instancias de procesos, ver tareas y prioridades asociadas y llevar a cabo las tareas que tienen asignadas.
- Kit para la administración del proceso: Proporciona una aplicación para monitorizar la actividad del proceso y rendimientos, y una herramienta para consultas y análisis que apunta a las necesidades de los supervisores de proceso. Estos pueden monitorizar el rendimiento de varios parámetros de proceso y tomar acciones proactivas, correctivas en caso de demoras y excepciones. Por otro lado manejan datos de análisis de proceso y rendimiento para identificar áreas de mejora del proceso. Sistemas avanzados soportan la exportación de datos del proceso desde entornos de producción a ambientes de simulación.
- Módulo de administración del sistema: Permite a los administradores de sistemas configurar servidores, monitorizar su rendimiento, llevar a cabo la gestión de derechos de acceso de usuario, etc. [13].

Características de BPMS

A grandes rasgos, las características que un BPMS tiene son:

- Asignar actividades a las personas de forma automática y según cualquier criterio, o según cargas de trabajo.
- Recordar a las personas sus actividades
- Optimizar la colaboración entre personas que comparten actividades.
- Automatizar y controlar el flujo de documentos, datos e imágenes.
- Definir y controlar “alertas” según criterios de tiempo, de evento o de condición, provocando así algún mensaje a un supervisor, un “escalado” de actividades a otras personas para que las resuelvan, y/o una resignación automática.
- Modificar los procesos y gestionar excepciones “en vivo”, o “al vuelo”, y desde cualquier lugar. Es decir, permitir modificar cualquier instancia de proceso ya iniciada, sin necesidad de volver a iniciarla y sin necesidad de conocimiento técnico. Además, esto se puede realizar desde cualquier navegador, desde cualquier lugar.

- Proveer una vista “on-line” para supervisores. La lista contiene el estado e histórico de cada instancia de proceso, de cada actividad, y del desempeño de las personas.
- Hacerles llegar a cada persona sus actividades y alertas, a través de la WEB, Email, SMS, o cualquier otro dispositivo móvil.
- Proveer métricas para responsables de áreas, organizadores, gestores de procesos y calidad, tanto para efectos de Mejora Continua como de Indicadores de Calidad y de Gestión.
- Integrarse fácilmente con otros sistemas, aplicaciones y ERPs (Enterprise Resource Planning).

2.2. Metodología de evaluación de BPMS (Delgado et al., 2015, [1])

Desde el 2010 hasta la actualidad varios proyectos enmarcados en el COAL se enfocaron en desarrollar una metodología de evaluación de BPMS.

Como resultado se obtuvo una lista de criterios funcionales y un método de ponderación. Permitiendo así, obtener una evaluación de las herramientas, la cual posibilita la comparación entre dos o más herramientas de forma objetiva teniendo en cuenta las necesidades de cada organización.

En 2013, en el marco del Proyecto de grado “Evaluación de características de Sistemas de Gestión de Procesos de Negocio comerciales” [1] se logró actualizar la lista de criterios. Se incluyeron nuevos requerimientos, de acuerdo a la evolución que han tenido los BPMS, esto llevó a agregar un nuevo nivel en la organización de los criterios. Este trabajo es el que se toma como base para el desarrollo de nuestro trabajo, que es presentado en los capítulos siguientes.

La organización de las características cuenta con un primer nivel denominado: Módulo. Hay dos tipos de módulos, técnico y no técnico, los mismos se definen de la siguiente manera: **Técnico:** Abarca todo lo referido al software en sí, es decir, requerimientos funcionales y no funcionales.

No técnico: Engloba características deseables a nivel estratégico, por ejemplo, si la herramienta es comercial o no.

A continuación se presenta un resumen de dichos módulos, también se incluyen las categorías que pertenecen a cada uno de ellos.

Tabla 2.1 Categorías por módulo

Categoría	Descripción
Módulo: Técnico	
Tecnología, Arquitectura e Interoperabilidad	Se enfoca en la plataforma tecnológica sobre la cual funciona la aplicación. Tiene en cuenta la arquitectura de la aplicación, los mecanismos de interoperabilidad con otros sistemas, por ejemplo, la integración de servicios web.
Diseño y modelado de procesos	Presenta características relacionadas a los componentes que permiten modelar. En este sentido, funcionalidades relacionadas al trabajo colaborativo, a la estética de la interfaz de usuario, al tipo de modelador e información del proceso son algunas de las características que forman parte de esta categoría.
Gestión de formularios	Agrupar características que refieren a la gestión de los formularios. Los formularios son una parte importante del modelado del proceso, son las interfaces gráficas a las que van a acceder los usuarios finales. Es por esto que se espera sean altamente configurables y su diseño pueda ser llevado a cabo por un <i>key user</i> .
Motor de <i>Workflow</i>	Engloba características relacionadas a las funcionalidades que permiten el trabajo colaborativo entre los usuarios finales, gestión de recursos, asignación de tareas en forma eficiente (notificaciones y escalamiento). También se incluyen características relacionadas a funcionalidades que permitan modificar el flujo a través de la asignación de tareas o definición de reglas.
Sistema de gestión documental	Es el módulo encargado de gestionar los documentos en el sistema. La categoría brinda características relacionadas a las funcionalidades para gestionar la configuración, mantener la metadata y realizar búsquedas sobre los documentos.
Portal	Esta categoría reúne características relacionadas al portal del sistema. De gran importancia, el portal es el principal punto de acceso al sistema por parte de los usuarios finales. Las características presentes en esta categoría son: inicio y fin de sesión, presentación de las tareas, personalización del portal, mecanismos de búsqueda y ayuda.

Categoría	Descripción
Administración de la seguridad	Agrupar funcionalidades asociadas a la seguridad del sistema. Es de suma importancia la posibilidad de definir roles y perfiles de acceso para los usuarios, ya que el sistema va a ser utilizado por distintos tipos de usuario con diferentes posiciones jerárquicas dentro de la estructura de la organización definida, por lo que se necesita un acceso diferenciado. En particular, se destacan los mecanismos de autenticación de los usuarios, la gestión de los mismos y la granularidad de los permisos que poseen.
Gestión, monitoreo y auditoría	Las funcionalidades que están asociadas a la gestión de los procesos forman parte de esta categoría. Así como las relacionadas a monitorizar su actividad y la posibilidad de realizar auditorías que permitan evaluar el rendimiento del sistema.
Módulo: No técnico	
Instalación y soporte	Las características asociadas a la instalación y el soporte del sistema se agrupan en esta categoría. Entre ellas se incluyen: ambientes diferenciados para la instalación, productos externos y paquetes de instalación.
Madurez	Esta categoría agrupa ciertos requerimientos que permiten determinar la madurez de una herramienta, como lo son la actividad de la comunidad, colaboración con otros productos, presencia regional, visión de futuro.
Comercial	Esta categoría está enfocada a los aspectos relacionados con la modalidad de licenciamiento. En este sentido, se consideran los costos asociados a la obtención y uso, teniendo en cuenta la variación de los mismos en el tiempo.

La metodología define un mecanismo de evaluación el cual se basa en escalas de importancia y valoración.

Escala de importancia Las características están clasificadas según su criticidad definiendo así el siguiente nivel de importancia:

- *Obligatorio*: debe formar parte de la herramienta.
- *Prioridad media*: es altamente deseable que forme parte de la herramienta.
- *Prioridad baja*: es deseable que forme parte de la herramienta pero con menor prioridad que el resto de los requerimientos.

Escala de valoración Se definen tres niveles de soporte:

- *Soportada totalmente*: la herramienta posee la característica.
- *Soportada parcialmente*: la herramienta posee la característica pero no de forma completa. Puede deberse a que no está cubierta toda la especificación o existe alguna restricción.
- *No soportada*: la herramienta no posee la característica.

Las características que no clasifican en esta escala de valoración se le asigna el valor No Aplica.

Además se definen tres niveles de cumplimiento:

- *Soportada en forma nativa*: la herramienta posee la característica en forma nativa.
- *Soportada vía particularización*: la herramienta posee la característica pero es necesario realizar algún desarrollo.
- *Soportada vía integración*: la herramienta posee la característica luego de incluir algún componente externo.

Evaluación cuantitativa La evaluación cuantitativa se hace con la finalidad de comparar de forma objetiva, con respecto a cierta característica, entre 2 o más herramientas. A partir de los criterios definidos, se puede realizar una evaluación cuantitativa de la herramienta, asignando puntajes de acuerdo a los siguientes criterios:

1. Asignar un puntaje a cada uno de los niveles de importancia o prioridad.
2. Asignar un puntaje a cada nivel de cumplimiento.
3. Asignar un puntaje a cada nivel de soporte.
4. Definir un factor de acuerdo al esfuerzo estimado para desarrollar cada funcionalidad (para características soportadas de forma nativa o vía integración el factor vale uno). Para el resto el factor varía entre 0,10 y 0,90.

$$\text{Puntaje} = \text{Importancia} * \text{Nivel de Soporte} * \text{Nivel de Cumplimiento} * \text{Factor}$$

Siendo “Importancia” el puntaje de acuerdo a la prioridad de la funcionalidad; “Soporte” el puntaje asociado al nivel de soporte de la funcionalidad; “Cumplimiento” el puntaje asociado al nivel de cumplimiento de la funcionalidad y “Factor”, un coeficiente variable, de acuerdo al esfuerzo estimado para desarrollar la funcionalidad.

2.3. Otras metodologías de Evaluación

Para un mejor desarrollo de la extensión de la metodología [1] que se establece en este documento, se investigaron y se hicieron comparaciones con varias metodologías de evaluación de software. Algunas de éstas no fueron desarrolladas para evaluación de un BPMS, sino que están enfocadas a sistemas más generales. Lo que se pretende lograr es complementar a [1] con aportes vistos en estas metodologías existentes. El estudio se basa

en una investigación intensa de los requerimientos no funcionales de un BPMS. Sin embargo, también se estudiaron algunos requerimientos funcionales y su forma de evaluación.

A continuación se presenta un resumen de los artículos investigados junto con su metodología.

2.3.1. Selection criteria for Tools Supporting Business Process Transformations for Electronic Commerce (Tsalgatidou et al., 1998, [2])

El artículo define un criterio de selección para las herramientas BPMTs (Business Process Modeling Tools) y WFMSs (Workflow Management Systems) definiendo un conjunto de características a tener en cuenta al momento de la selección de un BPMT y un WFMS para E-Commerce [2].

2.3.1.1. Metodología

Antes de meternos en la metodología es importante recordar que un *BPMT* (*Business Process Modeling Tools*) es una herramienta de desarrollo de modelos de procesos de negocios y que un *WFMS* (*Workflow Management Systems*) es un motor de procesos de negocio.

Para eso define y desarrolla un conjunto de categorías, basado en la transformación y el ciclo de vida de los procesos de negocio.

Cada categoría hacen referencia a las dos herramientas, pero con una descripción de los requerimientos específicos para cada una.

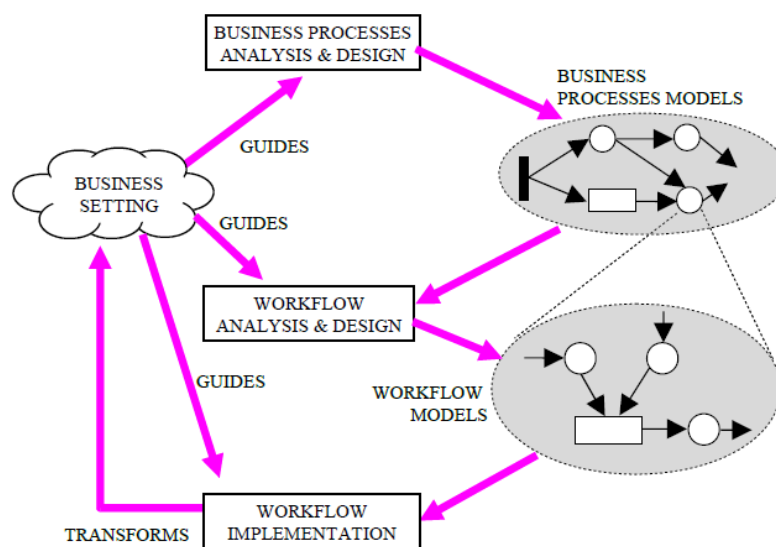


Figura 2.2 Ciclo de vida del proceso utilizado por la metodología [2]

2.3.1.2. Comparación con el método dado

El artículo en cuestión no define una metodología para la evaluación del criterio, sino que sugiere un conjunto de requerimientos a tener en cuenta al momento de la selección.

[1] define un conjunto de características a evaluar y una metodología de evaluación. Dividen las características a evaluar en dos grandes categorías las cuales a su vez tienen subcategorías, como se puede ver en la sección 2.2.

El artículo divide las características en cinco categorías:

1. Interfaz de usuario: separa las características en dos grandes grupos dependiendo de si afectan al usuario o a la herramienta. En el primero propone que la interfaz de usuario sea gráfica e interactiva. Para el segundo se fija en la adaptabilidad y la portabilidad.
2. Modelado: para el caso del modelado propone tener en cuenta algunos aspectos como son la filosofía del modelado, esto es la expresividad del modelo y la profundidad del mismo. La utilización de mecanismo conceptuales de modelado, como los mecanismos de abstracción y estructuración del modelo. También propone tener en cuenta la estructura organizacional, los recursos, las anotaciones. Otras de las características implican la representación del control de flujo, los datos y los materiales, así como una correcta definición del tipo de flujo y un modelado flexible y explícito del tiempo.
3. Análisis y validación: propone que los procesos de negocio y el modelado del flujo deben ser formal, para el análisis estático y la validación. Además contar con procesos de negocio ejecutables y con modelado de flujo, para el análisis dinámico y la validación.
4. Aspectos técnicos: propone tener en cuenta la interoperabilidad vertical y horizontal, lo que implica el poder exportar y compartir los procesos y que los mismos puedan ser compatibles con diferentes motores. Además indica que es importante que el uso de herramientas orientadas a objetos lo que permite mejor representación de la realidad y que las mismas cuenten con un repositorio para el modelado de los procesos.
5. Requerimientos para la automatización de procesos: estos requerimientos implican contar con monitoreo del estado del proceso, balanceo de la asignación de las tareas de forma automática y que permita también asignar tareas en forma manual. Debe proveer permisos para el inicio del proceso, el manejo de los documentos, la visibilidad de los reportes, entre otros. Poder obtener estadísticas, permitir el enrutamiento de la información y contar con procesamientos en paralelo. Otros de los aspectos a tener en cuenta son la posibilidad de incluir documentos y que los mismos sean transferidos entre tareas, así como poder configurar los deadlines de las tareas, que el proceso cuente con herramientas de comunicación y que cuente con soporte para APIs.

También menciona el control de la concurrencia, la recuperación y la posibilidad de realizar transacciones avanzadas.

Se puede decir que el artículo menciona muchas de las categorías que se encuentran en la metodología de [1]. Pero si bien algunas categorías se repiten, las mismas muchas veces están compuestas de diferentes características. Por ejemplo, para la categoría *Modelado* [1] no toma en cuenta el factor *representación del tiempo*, mientras que el artículo lo menciona como factor importante a tener en cuenta.

2.3.2. COTS Evaluation Using Desmet Methodology & Analytic Hierarchy Process (AHP) (Morera, 2002, [3])

El propósito de este artículo es para presentar cómo la metodología DESMET se puede combinar con la de AHP de una manera tal que la selección de COTS sea más fácil y precisa. Este nuevo enfoque de evaluación tiene como objetivo que la selección sea menos manual y más confiable [3].

Proporciona un ranking con puntaje de las diferentes alternativas, esto tiene como consecuencia que las decisiones tomadas sean más empíricas que humanas.

Introduce el tema diciendo que cada vez más los procesos de desarrollo son menos “artísticos” y tienden a verse como un proceso industrial. Utilizan pequeñas piezas de software para armar un sistema más grande, ya que la tendencia no es construir un gran sistema desde cero. Esto tiene como consecuencia una demandante en las metodologías de evaluación de software.

Como consecuencia de esto el European Software Institute(ESI) [14], investigó sobre varios modelos de evaluación y obtuvo como resultado un nuevo enfoque basado en DESMET y AHP.

2.3.2.1. Metodología

En la metodología presentada se toman dos metodologías como base DESMET y AHP. Por su parte DESMET pretende ayudar al evaluador en una organización en particular a planear y ejecutar la evaluación de manera imparcial y confiable, dando unos lineamientos para poder decidir si las herramientas actuales son suficientes para los objetivos y los procesos de calidad dentro de la organización o si se debe cambiar o mejorar, ya sea una herramienta o los métodos. En cambio AHP se basa en la idea de que la complejidad inherente a un problema de toma de decisión con criterios múltiples, se puede resolver mediante la jerarquización de los problemas planteados.

Para el método de DESMET se debe definir el objetivo de la evaluación y seleccionar un método de evaluación. En cuanto a los evaluadores es necesario definir roles, responsabilidades,

tiempo y cantidad de trabajo de cada uno. Por último se debe de analizar y presentar los resultados. Mientras que para AHP primero se deben de identificar los objetivos finales y alternativas, además evaluar las compensaciones entre los objetivos clave y soluciones alternativas, y finalmente llegar a un acuerdo final sobre la solución.

Luego de presentar ambas metodologías base, que en un principio no fueron consideradas para la evaluación de COTS, se presenta una combinación de ellas. Esta combinación toma los pasos de DESMET en primera instancia dejando de lado la presentación de resultados, y luego aplica AHP sobre el resultado obtenido. De esta manera se llega a lista final de productos seleccionados, estos se encuentran ordenados, estando primero el producto que mejor aplica según la organización.



Figura 2.3 Proceso jerárquico de evaluación de COTS. [3]

Como se ve en la figura dos grandes grupos participan en el proceso de evaluación:

- Grupo Técnico: involucrados en la primer parte de la búsqueda. Se obtiene la primera lista de candidatos.
- Grupo de Negocios: involucrados en la parte final de la selección. Propone el candidato final tomando como base la lista de candidatos proporcionada por el grupo técnico.

Para finalizar el análisis de este nuevo enfoque, se explica cada uno de los pasos, detallando que se debe de hacer en cada paso y por quien debe ser ejecutado.

2.3.2.2. Comparación con el método dado

No se encontró comparación directa con la metodología descrita por [1], dado que este artículo trata de la planificación de la evaluación y no describe los pasos específicos como si lo hace la metodología antes mencionada.

Es una metodología más genérica que la de [1] pero tienen en común el uso de un método numérico de puntuación para rankear las diferentes opciones.

2.3.3. A Formal Process for Evaluating COTS Software Products (Lawlis et al., 2001, [4])

Este artículo plantea la misma problemática que el artículo anterior, en cuanto a que cada vez es más grande la dependencia que tienen las agencias o empresas de desarrollo de los software COTS. Esto genera la necesidad de tener habilidad para elegir el producto que más se adecue a sus necesidades, teniendo en cuenta a todas las ofertas que hay en el mercado [4].

La evaluación que promueven asegura la calidad del producto seleccionado, esto se ejemplifica mediante con un caso de uso utilizado en la fuerza aérea de EEUU.

2.3.3.1. Metodología

Este método difiere de los procesos habituales de evaluación principalmente porque se basa en requisitos definidos por el usuario, además tiene en cuenta a todos los candidatos posibles y no una lista acotada como hacen la mayoría de los métodos estudiados. Para implementar este método es necesario que las mismas personas analicen los mismos productos, en lo posible cada producto debe tener la misma configuración, además cada evaluación debe de usar escenarios y datos similares. Y los responsables de evaluar el producto deben de aplicar los mismos requerimientos y criterios a los productos evaluados.

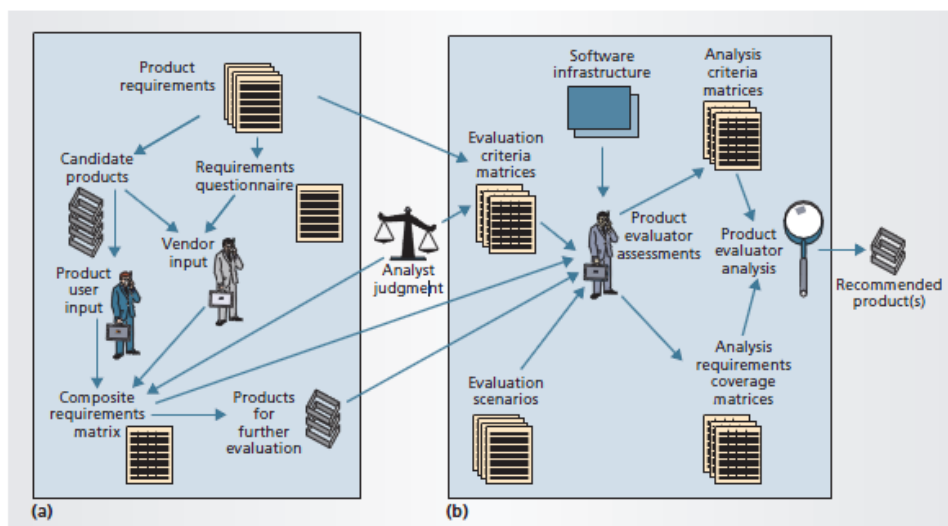


Figura 2.4 Proceso de evaluación basado en los requerimientos [4]

En la figura 2.4 se explica de forma abstracta la implementación del método. Se muestra en primera instancia que todo parte de la lista de requerimientos y de productos candidatos. A partir de estos, participan diferentes roles como son el usuario y el vendedor del producto para terminar construyendo una matriz de requerimientos en el cual se identifica la fuente del

requerimiento y su importancia. En base a esta matriz los evaluadores de productos construyen una nueva matriz en la cual le asignan un puntaje a cada uno de los requerimientos evaluados en cada uno de los productos. Luego de finalizada esta etapa se hace un cálculo explicado luego para llegar al valor final de cada producto. De esta manera se obtiene una lista ordenada de productos según el puntaje que obtuvo de dicho cálculo.

Componer los requerimientos en una matriz:

Se le asigna un peso entre 0 y 10, según el nivel de importancia que tiene.

Requirement	Products				Weight	Products			
	A	B	...	Z		A	B	...	Z
Runs on appropriate platform(s)	vf,ap	vp		vf	8	1/2	1/2		1
Supports appropriate configuration(s)	vf	vf,up		vp	5	1	1/2		1/2
⋮									
Displays use windows with graphics	vf	vf,up		vp	10	1	1/2		1/2
Online help is context sensitive	vf			vp	5	1			1/2
⋮									
Prepares specific analysis 1	vf,uf	vp		vf,ap	8	1	1/2		1/2
Prepares specific analysis 2	vp,up	vf,ap		vp	3	1/2	1/2		1/2
Prepares standard reports	vf	vf		vf	10	1	1		1
Prepares ad hoc reports	vp				5	1/2			
⋮									
				Totals	54	46	27		34

(a)

(b)

Figura 2.5 Matriz de requerimientos

Matriz de composición de requerimientos (a) Las letras u, v y a indican la fuente del requerimiento (vendedor, usuario o analista respectivamente), f y p indican si cumple este requerimiento en forma total (f) o parcial (p), en el caso que no cumpla se deja la celda en blanco. (b) Los analistas agregan una columna de pesos y traducen los indicadores en valores numéricos para el cálculo.

Ratings for product P using scenario x		Evaluators				Combined evaluators
Criterion	Weight	1	2	...	N	
Performance on specific platform 1	8	5	8		3	128
Ease of configuration	5	8	7		5	100
⋮						
Presentation of windows displays	10	7	9		4	200
Value of online help	5	6	9		5	100
⋮						
Adequacy of specific analysis 1	8	5	6		3	112
Adequacy of specific analysis 2	3	8	8		5	63
Adequacy of standard reports	10	9	7		5	210
Adequacy of ad hoc reports	5	5	5		1	55
⋮						
Totals	54	7	7		4	6

Figura 2.6 Matriz de análisis de los requerimientos

En esta matriz los analistas presentan una combinación de las respuestas de los evaluadores. Para obtener el resultado final (R), los analistas añaden clasificaciones de los evaluadores

para cada criterio (r_i), multiplican esta cantidad por el peso correspondiente para el criterio (w_i), y se divide el resultado por el producto de la suma de los pesos (W) y el número de evaluadores (N). Esta fórmula, $R = \frac{w_1r_1 + w_2r_2 + w_3r_3}{WN}$.

2.3.3.2. Comparación con el método dado

Tiene la una gran diferencia con la metodología de [1] dado a que la metodología desarrollada en este artículo se basa en los requerimiento de los usuarios.

Como extensión a la metodología de [1] se sugiere definir un escenario de evaluación aplicando el método formal que se propone en este artículo. Por ejemplo definiendo los roles de los evaluadores, que las mismas personas evalúen una cierta característica para sistemas diferentes. También se puede incluir la matriz que promedia los puntajes de los distintos evaluadores dado un escenario dado.

2.3.4. OpenBQR: a framework for the assessment of OSS (Taibi et al., 2007, [5])

El artículo define un marco de evaluación para herramientas Open Source. El mismo utiliza el método OpenBQR el cual es una extensión e integración de los métodos OpenBRR y QSOS. Tiene como objetivo cubrir las carencias de los métodos de evaluación de Softwares Open Source existentes [5].

Alguno de los problemas que ayuda a solucionar el siguiente método son:

- La mayoría de los métodos existentes se enfocan en aspectos específicos de los OSS.
- Algunos métodos proceden a evaluar los indicadores sin haberles asignado un peso previamente, esto provoca que factores con poco peso o ninguno, sean medidos o evaluados generando un gasto innecesario de tiempo y esfuerzo.
- Ningún método de evaluación de OSS trata adecuadamente con los aspectos internos y externos del producto.
- No evalúan la dependencia del usuario con los OSS, especialmente la disponibilidad de soporte y el costo de incluir módulos propietarios desarrollados por terceras partes.

2.3.4.1. Metodología

La metodología consta de tres fases:

Fase 1: Filtros de evaluación rápida Al contrario que sucede en el método OpenBRR, se define una lista de características a evaluar las cuales son medidas solo luego de que se les asigna un peso a las mismas.

Para la definición de esta lista se toma en cuenta la selección de indicadores basados en el alcance y los objetivos, el análisis de cualidades externas e internas, el soporte del producto en el tiempo y la existencia de funcionalidad.

Fase 2: Obtención de datos y procesamiento De la fase anterior se obtiene como resultado una lista de indicadores

Poda: todos los indicadores con peso cero o peso menor al umbral definido por el usuario, son eliminados.

Medición: se miden las características restantes.

Normalización: los pesos de las cinco áreas de la fase anterior son normalizados a un total de 100. Esto permite una comparación más equitativa entre las diferentes áreas.

Valoración: se obtiene el puntaje final de cada área, y se computa el puntaje final como la suma de los puntajes de cada área. Esto puede ser de utilidad para una primera valoración, permitiendo descartar aquellos productos cuyos puntajes sean demasiado bajos. Para una comparación más profunda se deben tomar en cuenta los valores de los puntajes de cada área junto con una evaluación de los costos.

Fase 3: Traducción de los datos Visualización de los resultados de la evaluación de varios productos para su comparación.

2.3.4.2. Comparación con el método dado

La metodología OpenBQR propone para cada fase algunas de las características a evaluar, para las cualidades internas recomienda utilizar las características definidas en la ISO 9126 o de la complejidad ciclomática de McCabe. Además toma en cuenta los requerimientos del usuario.

La metodología de [1] divide las características en dos grandes módulos, mientras que la metodología OpenBQR provee cinco categorías de características.

OpenBQR propone un método de poda, mientras que [1] no.

La metodología OpenBQR define la importancia mediante la asignación de pesos en el rango 0-9, luego las funcionalidades requeridas son evaluadas en un rango 0-100 y normaliza los pesos por área en un rango 0-100.

En [1] para el cálculo del valor final se tiene en cuenta la importancia, el nivel de soporte, nivel de cumplimiento y el factor. En la metodología OpenBQR, el puntaje final se obtiene de la suma de los puntajes de cada área descartando los productos con menos peso, luego realiza una evaluación más profunda basada en los valores de cada área y en los costos.

2.3.5. Evaluation and Selection COTS Software Process: The State of the Art (Tarawneh et al., 2011, [6])

Este artículo discute los principales problemas de los métodos existentes y presenta los principales procesos y métodos de evaluación para los Commercial Off-The-Shelf (COTS - Sistemas comerciales existentes).

Un software COTS es un software desarrollado para ser reutilizado y comercializado, que se integra a otros proyectos de software para proveer funcionalidades adicionales a un sistema final. Un sistema COTS-based está desarrollado basado en la selección, adaptación e integración de uno o más software COTS, este proceso también se llama COTS-Based System Development (CBSD), este proceso también mejora la fiabilidad, flexibilidad y reutilización del sistema [6].

El artículo también menciona los problemas más comunes a la hora de seleccionar un COTS. La falta de un proceso de selección y evaluación de software COTS que sea sistemático, repetible y bien definido, es el primer problema a solucionar, a eso hay que agregar que no se consideran las incompatibilidades entre los requerimientos del usuario y las características del COTS.

Los requerimientos no funcionales (rendimiento, interfaz, soporte y mantenimiento, costo, etc.) deben ser considerados, no manejarlos implica un riesgo, se pueden producir fallas, demoras o un incremento del costo y el esfuerzo.

Según el artículo una de las tareas más importantes para poder entender, evaluar y seleccionar el software apropiado, es la identificación de los criterios que serán evaluados. Para poder definir los criterios se establece el siguiente marco de calidad:

1. Un buen criterio debería incluir una declaración medible y un método de medición.
2. El criterio puede ser uno bueno si el dato que es recolectado soporta el método de medición.
3. La capacidad de discriminar, es un aspecto importante para que un buen criterio ayude a los evaluadores a distinguir entre los softwares.
4. Un buen criterio no se debería superponer con otros porque puede traer conflictos en el análisis de las características, haciendo confusos los resultados y malgastando esfuerzos.
5. Si el criterio no tiene valor en el contexto del software, entonces no debe ser utilizado.

Para finalizar, se sugiere la utilización de ISO y estándares para incluir atributos de calidad y un conjunto de métodos de medición. Las técnicas más comunes que se han utilizado son: Goal Question Metric (GQM) y un marco para los requerimientos no funcionales.

2.3.5.1. Metodología

El principal objetivo es dar un nuevo marco para el apoyo y la mejora en la evaluación y selección de COTS en la industria. Esta metodología consiste en cuatro estados: Estudio del estado del arte, Estudio del estado de la práctica, Marco de desarrollo, Marco de evaluación.

Estudio del estado del arte En esta etapa se estudian en profundidad estudios anteriores, enfocándose principalmente en los procesos y los criterios de evaluación comunes que han sido utilizados en la evaluación y selección de COTS.

Estudio del estado de la práctica En esta etapa se investiga sobre las prácticas actuales de la evaluación y selección de COTS los cuales están relacionadas con el uso de los procesos, factores y problemas relevantes.

Marco de desarrollo Los resultados de los estudios teóricos y prácticos se utilizarán para desarrollar un nuevo marco. El propósito del marco será integrar un conjunto de procesos y sus actividades, también las relaciones entre ellas serán establecidas. Se determinarán los requerimientos no funcionales, se utilizarán varias técnicas para obtener los requerimientos del usuario y además para identificar las incompatibilidades entre las características del COTS y los requerimientos del cliente.

Marco de evaluación El objetivo de esta fase es evaluar la eficiencia y aceptabilidad del marco en un ambiente real. Un caso de estudio será adoptado como método cualitativo. El proceso de evaluación se iniciará mediante la determinación de los criterios que se utilizarán para evaluar el marco. Los datos se introducen en una herramienta para el análisis, modificación y refinamiento que se llevarán a cabo, si es necesario.

2.3.5.2. Comparación con el método dado

Después de leer sobre los problemas más comunes que se tienen al evaluar y seleccionar COTS, y de los métodos existentes, dónde se define una lista de características, pero no define cómo se deben evaluar, simplemente da un método general para puntuar todas las características de la misma manera. No se toman en cuenta los requerimientos no funcionales, ni los requerimientos de usuario.

Primero, se definen las distintas etapas que debería tener una metodología de evaluación y selección de COTS: Supporting, Preparation, Evaluation and Selection. Segundo, se explican 3 estrategias para evaluar los distintos criterios, las estrategias son distintas y dependiendo del caso combinar las estrategias puede mejorar la evaluación.

Finalmente, explica la metodología del artículo. En la cual plantea cuatro etapas: estados del arte, de la práctica, marco de desarrollo y marco de evaluación. Se basa mucho en las evaluaciones existentes y datos históricos, pero tratando de reproducir la metodología para conseguir los datos y ver las dificultades.

2.3.6. Comparación de otras propuestas con la metodología COAL

En esta sección se muestra una tabla donde se comparan algunas características de las distintas metodologías estudiadas en la sección anterior. La tabla 2.2 muestra las características, una breve descripción y para cada metodología se indica si cumple con esa característica.

Tabla 2.2 Comparación de características con otras metodologías

Características	Delgado et al., 2015, [1]	Tsalgatidou et al., 1998 [2]	Morera, 2002, [3]	Lawlis et al., 2001, [4]	Taibi et al., 2007, [5]	Tarawneh et al., 2011, [6]	Descripción
Provee lista de criterios	✓	✓	✗	✗	✓	✗	Define una lista de criterios que debería ser tomada en cuenta al momento de realizar la evaluación.
Provee caso de estudio	✓	✗	✗	✓	✓	✗	Provee un conjunto de casos de estudio para aplicar la metodología, ya sea dando la descripción del caso de estudio y un modelo asociado, o simplemente la descripción de un problema, del cual se extraen uno o más modelos.
Utiliza métodos de filtrado	✗	✗	✓	✓	✓	✓	Sugiere la utilización de filtros en la etapa de evaluación, esto incluye tanto las metodologías para realizar el filtrado cómo los filtros en sí mismos.

Características	Delgado et al., 2015, [1]	Tsalgatidou et al., 1998 [2]	Moreira, 2002, [3]	Lawlis et al., 2001, [4]	Taibi et al., 2007, [5]	Tarawneh et al., 2011, [6]	Descripción
Utiliza métodos de poda	✗	✗	✗	✗	✓	✓	Sugiere la utilización de métodos de poda en la etapa de evaluación, esto incluye las metodologías para realizar las podas.
Requerimientos no funcionales	✗	✓	✗	✗	✗	✓	Toma en cuenta requerimientos no funcionales en los criterios de evaluación. Los requerimientos pueden estar dados dentro de una lista de criterios a evaluar, o dentro de una lista específica de características no funcionales, o simplemente se sugiere fuertemente tener en cuenta los requerimientos no funcionales.
Roles responsables	✗	✗	✓	✓	✗	✗	Se definen roles que tienen que evaluar cada criterio, cómo se definen, que conocimiento tiene que tener la persona que se asigna a ese rol.
Framework de calidad	✗	✗	✗	✓	✗	✓	Se adecuan al framework de calidad definido.

Características	Delgado et al., 2015, [1]	Tsalgatidou et al., 1998 [2]	Moreira, 2002, [3]	Lawlis et al., 2001, [4]	Taibi et al., 2007, [5]	Tarawneh et al., 2011, [6]	Descripción
Tiene etapas definidas	✓	✓	✓	✓	✓	✓	Divide la metodología en etapas. El método sugiere dividir los criterios a evaluar en las diferentes etapas del ciclo de vida del proceso de negocio (configuración, evaluación, ejecución, mantenimiento.)
Tiene en cuenta el histórico	✗	✗	✗	✗	✗	✓	Tiene en cuenta los datos históricos. La metodología utiliza una base de datos que contiene los resultados de evaluaciones anteriores.
Método de evaluación de criterios	✓	✗	✗	✗	✓	✗	Define como deben ser evaluadas las características. La metodología da un método para evaluar cada uno de los criterios, puede o no dar las métricas para la evaluación.
Comparativa con otras metodologías	✗	✗	✗	✓	✓	✓	Presenta un análisis de diferencias y similitudes con otras metodologías.

Características	Delgado et al., 2015, [1]	Tsalgatidou et al., 1998 [2]	Moreira, 2002, [3]	Lawlis et al., 2001, [4]	Taibi et al., 2007, [5]	Tarawneh et al., 2011, [6]	Descripción
Metodología específica	✓	✓	✗	✗	✓	✗	Se aplica a un software específico. La metodología se aplica a un determinado tipo de software, por ejemplo, BMP, ERP, CRM, etc.
Metodología genérica	✗	✗	✓	✓	✗	✓	Se aplica a cualquier tipo de software. Por ejemplo, open source, e-commerce, COTS, etc.
Análisis del caso de estudio	✓	✗	✗	✗	✓	✗	Realiza un análisis luego de aplicado el caso de estudio.
Provee proceso de evaluación	✓	✗	✓	✓	✗	✓	Define un proceso a seguir para aplicar la metodología, ya sea indicando los pasos a seguir, o guiando la evaluación con diferentes etapas.
Provee método de puntuación específico por criterio	✗	✗	✗	✗	✓	✗	Define formas de evaluación específicas para cada uno de los criterios definidos. La metodología provee métricas de evaluación para cada criterio.

Características	Delgado et al., 2015, [1]	Tsalgatidou et al., 1998 [2]	Moreira, 2002, [3]	Lawlis et al., 2001, [4]	Taibi et al., 2007, [5]	Tarawneh et al., 2011, [6]	Descipción
Toma en los requerimientos del usuario	✖	✖	✖	✔	✔	✔	Incluye los requerimientos del usuario como criterios a evaluar. La metodología utiliza los requerimientos del usuario como criterios de corte o dentro de la lista de criterios no funcionales a evaluar.
Provee un método de puntuación	✔	✖	✖	✔	✔	✖	Define un método de puntuación para la evaluación de las características. La forma de evaluar las características puede ser la misma para todas o se puede especificar el un método de puntuación específico para cada característica.

Referencias:

- ✔ Esta figura en la tabla representa que la metodología cumple con la característica.
- ✖ Esta figura en la tabla representa que la metodología no cumple con la característica.

2.3.6.1. Conclusiones

Dado que las características que se encuentran agrupadas bajo el módulo no técnico son un complemento de las características que se encuentran en el módulo técnico, consideramos que el módulo técnico debería dividirse en funcional y no funcional. Generando así un nuevo nivel de jerarquía.

Al intentar clasificar los criterios no funcionales según las escalas utilizadas en el proyecto anterior para clasificar los criterios funcionales, se notó que la misma no se adecua a las características no funcionales. Por ejemplo si intentamos clasificar la característica *latencia*, se podría llegar a decir que la misma es *soportada totalmente* por el BPMS evaluado. Entendemos por soportado algo que “trae” el BPMS, que se asociaría más con una funcionalidad del mismo y no a la calidad.

Por otro lado, la evaluación de los atributos de calidad es subjetiva, determinar el nivel de *madurez* de un producto u organización dependerá del evaluador (basado por ejemplo, en los requerimientos del usuario) y para qué será evaluada.

A medida que investigamos nuevas características no funcionales, notamos que la evaluación de las mismas se podían realizar en diferentes etapas del ciclo de vida del proceso de negocio e incluso una misma característica podría tener diferentes evaluaciones en diferentes etapas. Por lo tanto, entendemos que es conveniente clasificar las características a evaluar en las diferentes etapas del ciclo de vida en la que las mismas tengan sentido ser evaluadas.

2.4. Resumen

En este capítulo se introdujeron los conceptos necesarios para generar un entendimiento en el marco teórico. En primera instancia se hicieron las definiciones necesarias de la gestión de procesos de negocio, concepto de un BPM, sus características y beneficios. Se introdujeron los BPMS y sus principales funciones, que es en definitiva el objetivo de las evaluaciones. Además se presentó el ciclo de vida seguido por la metodología y los proyectos anteriores que se tomaron como base para la realización de la lista de características. Por último, se mostraron otras metodologías de evaluación investigadas y la comparación con la metodología actual a extender.

Capítulo 3

Características a evaluar

Uno de los objetivos del proyecto de grado es extender la metodología de evaluación de BPMS presentada en la sección 2.2 del capítulo 2. En este sentido, se desea incluir a la misma una lista de criterios no funcionales.

En la sección 3.1 se realiza una introducción al tema tratado en el capítulo. En la sección 3.2 se describe una extensión realizada a la metodología denominada plan de pruebas. En la sección 3.3 se definen los casos de prueba. En la sección 3.4, se describe cual fue el enfoque utilizado para la selección de las nuevas características. La sección 3.5 está enfocada a describir los distintos niveles en los que están divididas las características, así como también la estructura utilizada para definir las. La sección 3.6 presenta la lista de criterios no funcionales siguiendo la estructura definida en la sección anterior. En la sección 3.7 se desarrolla la metodología de evaluación de las características no funcionales. Por último, en la sección 3.8 se brinda un resumen del capítulo.

3.1. Introducción

La metodología cuenta con una lista de criterios enfocados en evaluar las funcionalidades del sistema. Los criterios no funcionales que se incluyeron en la metodología, forman parte de una lista existente. Por lo tanto, se mantuvo el primer nivel de jerarquía que posee la lista, así como también, la estructura de definición de cada una de ellas.

3.2. Extensión de la metodología: Plan de pruebas

Una de las propuestas del proyecto sobre la metodología, fue la inclusión de una nueva tarea en el proceso definido en la sección 3.2, en la figura 1 en [1]. Esta nueva tarea se corresponde a un subproceso el cual define como realizar el plan de pruebas. El subproceso se ubica entre la tarea de selección de características a evaluar “Select characteristic to

evaluar” y el subproceso de definición de casos de pruebas “Define test cases”. El propósito del plan de pruebas que se plantea en esta sección, es permitir definir los lineamientos a seguir para realizar la planeación de la etapa de pruebas. En este sentido, se especifica cómo debe ser la ejecución del plan de pruebas.

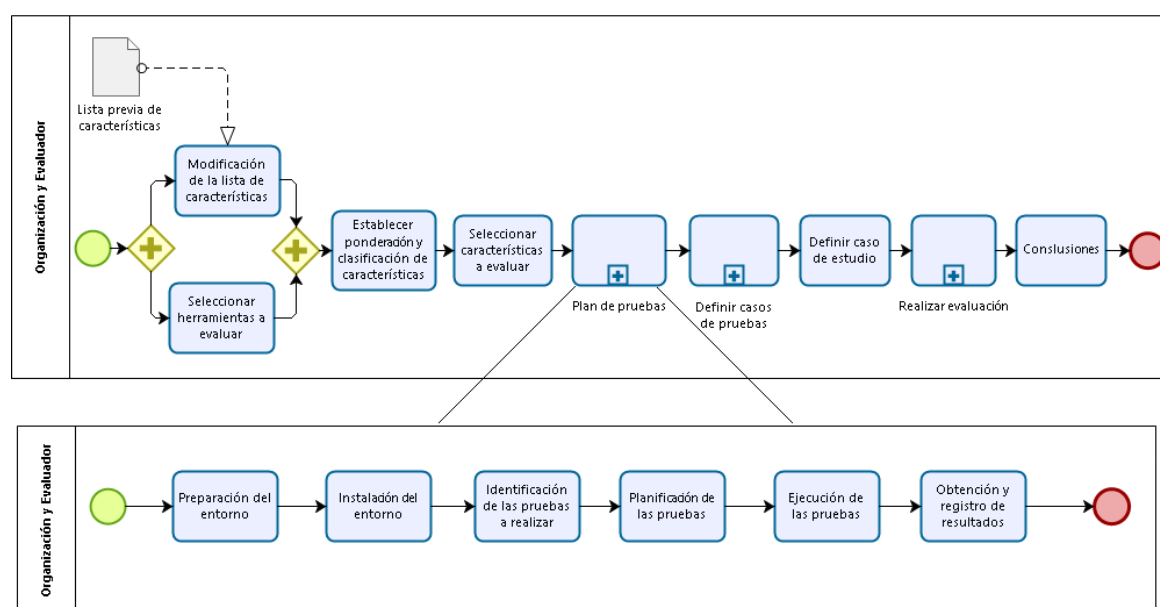


Figura 3.1 Proceso BPMN con subproceso de Plan de Pruebas

A continuación se desglosan los pasos requeridos para la ejecución del plan de pruebas:

Requerimientos para la ejecución del plan de pruebas

- Especificación de requisitos
- Manual de usuario
- Sistema a probar

Tareas a realizar

1. Preparación del entorno.

Se recomienda un entorno de pruebas específico para la ejecución del plan. En esta tarea se debe definir el ambiente en el que se deben de ejecutar los casos de prueba definidos en la sección 3.3

2. Instalación del entorno

En esta tarea se hace la instalación de los componentes necesarios (hardware y software) del ambiente para poder empezar con la evaluación de la herramienta.

3. Identificación de las pruebas a realizar.

En esta tarea se especifica las características que se van a evaluar en los casos

de prueba. Se hace un filtrado de las características funcionales y no funcionales definidas.

4. Planificación de las pruebas.

Se establecerán las posibles dependencias que hubiera entre las pruebas y se establecerá el orden de ejecución de las pruebas basándose en dichas dependencias. Además quedan definidos los roles y actividades por rol de los participantes en la ejecución de las pruebas.

5. Ejecución de las pruebas.

Los participantes con el rol de ejecución de las pruebas, llevan a cabo la ejecución de las mismas según el plan establecido.

6. Obtención y registro de resultados.

Durante las pruebas realizadas en la etapa de ejecución se completará una tabla que contiene las características a evaluar, indicando el nivel de cumplimiento. Posteriormente se procesan estos resultados.

3.3. Casos de prueba

Los casos de prueba fueron definidos con el objetivo de homogenizar la evaluación de las herramientas. Una vez completado el plan de pruebas de la sección 3.2 se procede a ejecutar cada caso de prueba con el fin de evaluar determinadas características.

En primer lugar, los casos de pruebas permiten a los integrantes del proyecto estandarizar las pruebas, sin tener en cuenta la experiencia de cada integrante y las herramientas.

En segundo lugar, los casos de prueba presentan dos aspectos a tener en cuenta, la estructura organizacional y la estructura del propio caso de pruebas. La estructura organizacional refiere a los usuarios y roles que se emplean. Por otro lado, la estructura del caso de pruebas se presenta de la siguiente manera:

- *Introducción:* muestra una explicación a grandes rasgos de lo que realiza el caso de prueba.
- *Modelo proceso:* muestra el diagrama del proceso.
- *Explicativo de la secuencia:* conjunto de pasos explicativos de cada prueba.
- *Observaciones:* se muestran observaciones de la explicación de la secuencia.
- *Características abarcadas:* conjunto de características incluidas en la lista que abarca el caso de prueba.

3.4. Enfoque sobre la selección de características

En esta sección se describirá cuál fue la óptica utilizada para la selección de las nuevas características.

En primer lugar, la búsqueda se orientó a características no funcionales de varios tipos de software. De esta búsqueda se obtuvo una gran cantidad de características no funcionales, pero se extrajeron aquellas que eran aplicables a sistemas BPM.

Posteriormente, se realizó un estudio de las taxonomías [15] y [16]. Como consecuencia, se logró obtener una lista de criterios de calidad, como parte de los criterios no funcionales.

En última instancia, se revisó la lista de características dadas. La lista presentaba características que incluían aspectos no funcionales. Entonces, se planteó la necesidad de redefinir algunas características existentes y una nueva jerarquía, que son desarrolladas en la sección 3.5.

En síntesis, una vez investigadas las características para varios tipos de software, de haber estudiado las diferentes taxonomías y de haber revisado la lista de características existente, se obtuvo una lista de características que se ajusta a los BPMS.

3.5. Estructura de las características

En esta sección se especificará la jerarquía utilizada para dividir a las características. Así como también, la estructura que se utiliza en la definición de cada una de ellas.

Como se explicó en la sección anterior las características en un primer nivel se dividen en módulos técnico y no técnico. Unas de las modificaciones propuestas para la metodología existente fue la división del módulo técnico en dos grandes ramas: Funcionales y No funcionales.

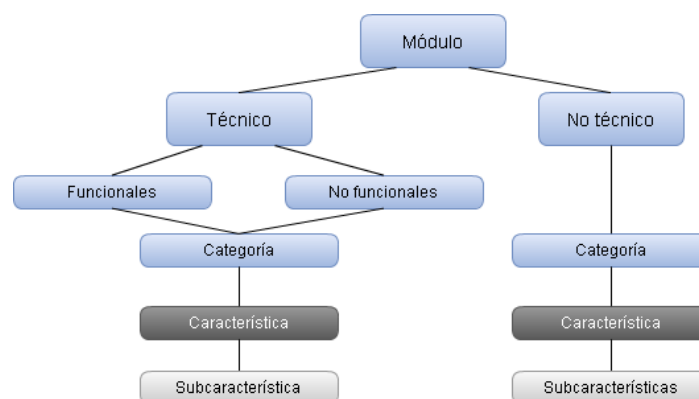


Figura 3.2 Estructura jerárquica de las características

La figura 3.2, ilustra de forma esquemática la jerarquía lograda. Donde se identifica un nuevo nivel en el que se encuentran las categorías, luego las características. Al mismo tiempo, cada característica puede tener subcaracterísticas. Por otro lado, se mantuvo la forma de presentar a cada característica. Cada una se define mediante una descripción. Si la característica cuenta con subcaracterísticas, las mismas son enumeradas. Por último, se incluye la fuente con una o más referencias. A continuación, se describe cada uno de los ítems que componen la estructura de la definición de las características:

Descripción: Se presentan aspectos generales de su funcionamiento, se especifica su significado y se definen conceptos involucrados.

Subcaracterísticas: Se propone una clasificación para las herramientas basada en esta característica o se listan funcionalidades estrechamente relacionadas con ésta, dependiendo del caso.

Referencias: Se incluyen las fuentes.

En resumen, se obtuvo una lista de características re-estructurada, mejorada y extendida con nuevas características no funcionales. La lista se organiza en niveles, lo que permite ordenar la forma de evaluar e identificar los distintos tipos de funcionalidades. Así como también unificar la lista que existía hasta el momento, con los requisitos incorporados.

3.6. Clasificación general de características

Como se indicó anteriormente, uno de los objetivos de este proyecto es ampliar la lista de criterios que posee la metodología. En esta sección se brinda una descripción de las características no funcionales incorporadas.

3.6.1. Definición de características técnicas no funcionales

Como resultado de combinar las taxonomías [15], [16] y las utilizadas en [1], se logró confeccionar una lista de características no funcionales.

Seguidamente, se presenta una versión resumida de los criterios, describiendo las categorías que conforman el módulo técnico no funcional. La versión completa y detallada de la misma se encuentra en el Apéndice A.

3.6.1.1. Fiabilidad

La fiabilidad es la capacidad que tiene el sistema de mantenerse operativo, durante un tiempo establecido y bajo un conjunto de condiciones definidas [15].

La categoría se enfoca en aspectos relacionados a la disponibilidad de los componentes del sistema. En este sentido, se tiene en cuenta la madurez tanto del sistema como de la empresa que lo respalda. También contempla la capacidad para ser reparado que tiene el

producto, teniendo en cuenta la modularidad, reusabilidad, analizabilidad y la capacidad para ser modificado y probado.

3.6.1.2. Compatibilidad

En esta categoría se agrupan aspectos relacionados a la compatibilidad del producto. La compatibilidad es la capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software [16].

Se estudian dos aspectos relacionados a la compatibilidad. En primer lugar, la posibilidad de que el producto coexista con otros productos independientes, en un entorno común o compartiendo recursos sin detrimento. En segundo lugar, la capacidad que tiene el sistema para intercambiar información y hacer uso de la misma.

3.6.1.3. Performance

La performance refiere a la capacidad de respuesta: o bien el tiempo requerido para responder a eventos específicos o el número de eventos procesados en un intervalo de tiempo determinado [15]. En este sentido, la categoría atiende los siguientes aspectos referidos al rendimiento del producto: el tiempo de respuesta, el throughput y la capacidad.

3.6.1.4. Portabilidad

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro [16].

En esta categoría se toman en cuenta aspectos relacionados a la portabilidad del sistema. Por ejemplo, la capacidad que tiene el sistema de soportar diferentes sistemas operativos, navegadores. Por otro lado, contempla las dificultades para instalar y/o desinstalar de forma exitosa el entorno. También tiene en cuenta la posibilidad de exportar el proceso, así como instalar la herramienta en diferentes ambientes.

3.6.1.5. Usabilidad

Esta categoría agrupa requisitos relacionados a las aptitudes que presenta la herramienta para su utilización.

Brevemente, la categoría presenta características enfocadas a la capacidad que tiene el producto para que un usuario aprenda a utilizarla. En particular, toma en cuenta la estética de la interfaz de usuario, la accesibilidad a manuales, mensajes de error y sugerencias.

3.6.1.6. Seguridad

Capacidad de software para cumplir con los niveles de riesgo permitidos, tanto para posibles daños físicos como para posibles riesgos de datos.

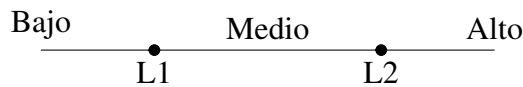
La categoría refiere a requisitos relacionados con la seguridad. Entre ellas, confidencialidad, integridad, no repudio, responsabilidad y autenticidad.

3.7. Método de evaluación de características no funcionales

Para poder determinar el nivel de cumplimiento de cada característica el evaluador debe definir los valores: *Alto*, *Medio* y *Bajo*. Dichos valores se utilizarán para evaluar a todas las características y además se calcularán los umbrales de la siguiente forma: se toma los valores *Alto*, *Medio* y *Bajo* definido por el evaluador, en base a esto se calcula el límite entre *Bajo* y *Medio* como la media entre los valores ingresados para *Bajo* y *Medio*, llamemos a este límite ‘umbral mínimo’, y el límite entre *Alto* y *Medio* como la media entre los valores ingresados para *Medio* y *Alto*, llamemos a este límite ‘umbral máximo’.

Umbrales	
$umbral_{mimo} = \frac{Medio + Bajo}{2}$	(3.1)
$umbral_{mximo} = \frac{Alto + Medio}{2}$	(3.2)

Cada característica puede tener subcaracterísticas, en ese caso, se debe evaluar cada subcaracterística por separado. Para calcular el nivel de cumplimiento de la subcaracterística se deben establecer los valores ($L1$ y $L2$) que permiten definir los rangos de cumplimiento, a partir de las definiciones dadas en cada subcaracterísticas.



Además se debe calcular el valor final de la subcaracterística (v_{sub}), por tanto el nivel de cumplimiento se determina de la siguiente manera:

- $v_{sub} < L1 \rightarrow$ Nivel de cumplimiento Bajo.
- $L1 \leq v_{sub} < L2 \rightarrow$ Nivel de cumplimiento Medio.
- $L2 \leq v_{sub} \rightarrow$ Nivel de cumplimiento Alto.

Entonces, el evaluador definirá un peso entre 0 y 1, para cada subcaracterística, el cual se multiplicará por el nivel de cumplimiento de esa subcaracterística.

Puntaje de la subcaracterística

$$pSub = nivelCumpSubCar * peso \quad (3.3)$$

Siendo $pSub$ el puntaje de la subcaracterística y $nivelCumpSubCar$, el nivel de cumplimiento de la subcaracterística.

Luego se suma el valor obtenido por todas las subcaracterísticas y se divide entre la suma de los pesos de las subcaracterísticas. Éste número obtenido es el ‘valor final’ (vf) del nivel de cumplimiento de la característica.

Valor final

$$vf = \frac{\sum_{k=1}^N pSub_k}{\sum_{k=1}^N peso_k} \quad (3.4)$$

Siendo vf el ‘valor final’ de la característica y N la cantidad de subcaracterísticas.

Por último, se toma el vf de la característica y se clasifica el nivel de cumplimiento de la misma de esta manera:

- $vf < umbral_{mnimo} \rightarrow$ Nivel de cumplimiento Bajo.
- $umbral_{mnimo} \leq vf < umbral_{mximo} \rightarrow$ Nivel de cumplimiento Medio.
- $umbral_{mximo} \leq vf \rightarrow$ Nivel de cumplimiento Alto.

En el caso de las características que no tienen subcaracterísticas, se clasifica directamente.

3.8. Resumen

Se presentaron todos los aspectos involucrados con la actualización de la lista de características deseables de los sistemas BPM. En este sentido, se describieron los enfoques utilizados para la selección de los criterios que actualmente conforman la lista, tomando como punto de partida la lista anterior, explicando cuales fueron las consideraciones que se tuvieron en el momento de la confección. Además, se mostró la estructura definida tanto para las características como para su definición. Considerando que se introdujo un nuevo nivel dentro del módulo técnico, de esta manera, quedaron bien diferenciados los criterios funcionales de los no funcionales, cumpliendo así, el primer objetivo del proyecto.

Luego, se describieron a modo de resumen las categorías que componen el módulo técnico - no funcional. Para finalizar, se desarrolló el método de evaluación para las características no funcionales, en consecuencia, quedó determinado en su conjunto, el método de evaluación de la metodología. Además se describió el plan de pruebas que es parte fundamental del benchmark, cumpliendo así el segundo objetivo del proyecto.

Capítulo 4

Evaluación de características

Aplicar la metodología actualizada para la evaluación de una selección de BPMS es el tercer objetivo de este proyecto. En consecuencia, este capítulo presenta el método de evaluación para las características no funcionales y algunos resultados que se obtuvieron de la aplicación de la metodología.

En la sección 4.1 se introduce brevemente el tema tratado en el capítulo. La sección 4.2 presenta las herramientas objeto de la evaluación. La sección 4.3 muestra un resumen de los resultados obtenidos de la evaluación de las características para cada herramienta. Finalmente, en la sección 4.4 se brinda un resumen del capítulo.

4.1. Introducción

El objetivo principal de este capítulo es mostrar la aplicación de la metodología actualizada. En este sentido, se presenta la lista de características seleccionadas para realizar la evaluación.

En primer lugar, para poder aplicar la metodología se realizó una selección de herramientas tanto comerciales como *open source*. En segundo lugar, se extendió el método de evaluación, incluyendo un método específico para evaluar las características no funcionales.

Finalmente, se presenta un resumen de los resultados obtenidos para cada herramienta. Además, se muestra una tabla comparativa de los resultados de cada herramienta para un conjunto reducido de criterios.

4.2. Selección de herramientas

Actualmente, existen diversos tipos de BPMS, elaborados por distintos proveedores. Inicialmente, la selección de los BPMS incluía 9 herramientas *open source* o comerciales. Además, se seleccionaron BPMS cuyo motor de *Workflow* soporta alguno de los siguientes estándar: XPDL, BPEL o BPMN 2.0.

A continuación se presenta una breve descripción de los BPMS a evaluar:

Tabla 4.1 Herramientas a evaluar

Herramienta	Versión	Descripción	Referencia
JBoss BPM Suite	6.1.0	Es una herramienta open source que utiliza el estándar BPMN 2.0 cuyos componentes se obtuvieron mediante descarga en la página de la herramienta.	[17]
Bonitasoft	Community 7.1	Es una herramienta <i>open source</i> que utiliza el estándar XPDL.	[18]
Intaliolbpms	Enterprise 7.5.0	Utiliza el estándar BPEL, además es una herramienta open source. Se obtuvo una licencia académica a través de sitio web de la herramienta.	[19]
Activiti	6.0.0.Beta2	Es una herramienta open source que utiliza el estándar BPMN 2.0. Los componentes se descargaron del propio sitio de la herramienta.	[20]
Bizagi BPM Suite	Enterprise 10.7	Se utilizó una versión completa de la herramienta, únicamente limitada por la cantidad de usuarios del sistema.	[21]
Camunda	Community 7.3.0	Es una herramienta <i>open source</i> que utiliza el estándar BPMN 2.0.	[22]
Aris Express	Student	Es una herramienta comercial que utiliza el estándar BPMN 2.0. No fue posible ejecutar ningún proceso, por lo tanto, solo se realizó la evaluación de las características teóricas.	[23]

Herramienta	Versión	Descripción	Referencia
Orchestra	Community 4.9.0	Es una herramienta open source que utiliza el estándar BPEL. El software se obtuvo del sitio de la herramienta.	[24]
ProcessMaker	Community 3.0.1.5	El software se obtuvo a través de un contacto de la empresa luego del registro en la página de la herramienta.	[25]

4.3. Evaluación de herramientas

La metodología aplicada a las herramientas seleccionadas dio un conjunto de resultados. En esta sección se muestra una reseña de los resultados, para ello se seleccionaron las siguientes características: Arquitectura del BPMS, Capacidad de aprendizaje, Tiempo de respuesta e Integridad. Además, se realiza un análisis general de los resultados basado en la experiencia y los resultados obtenidos durante la evaluación. La tabla 4.2 que se encuentra al final de la sección muestra los resultados obtenidos de cada característica para cada herramienta, teniendo en cuenta la escala de valoración.

4.3.1. Red Hat JBoss BPM Suite

La versión utilizada para la evaluación es la 6.1.0 cuyos componentes se obtuvieron mediante descarga en la página de la herramienta [17].

4.3.1.1. Arquitectura del BPMS

Red Hat JBoss BPM Suite incluye todas las funcionalidades de JBoss BRMS, con capacidades adicionales de monitoreo de las actividades, inicio de procesos y gestión de tareas desde el Business Central. JBoss BPM Suite también provee un repositorio central para almacenar los procesos y las reglas de negocio.

Componentes claves:

JBoss BPM Central (Business Central): es una aplicación web que permite crear, editar, construir, administrar y realizar el seguimiento de los activos empresariales de JBoss BPM Suite. También permite la ejecución de procesos de negocio y la gestión de las tareas creadas por esos procesos.

Business Activity Monitoring (BAM) Dashboards : ofrece capacidades de generación de informes. Permite utilizar un tablero de instrumentos predefinido e incluso crear su propio panel de control personalizado.

Maven Artifact Repository : los proyectos de JBoss BPM Suite se construyen como proyectos Apache Maven y la ubicación predeterminada del repositorio Maven es

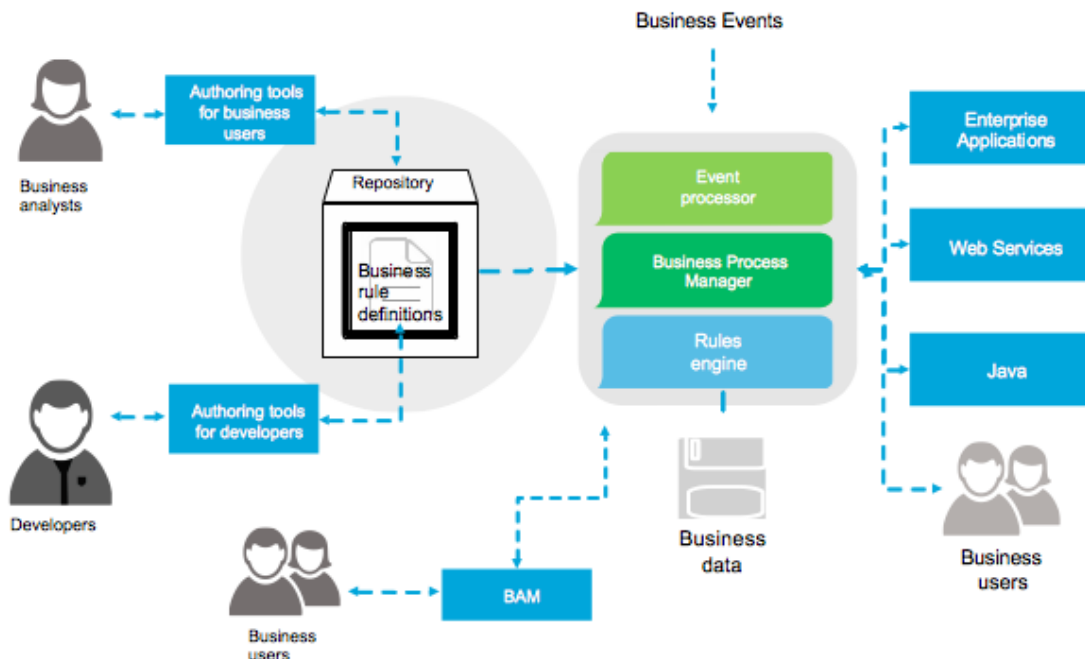


Figura 4.1 Arquitectura JBoss BPM Suite [26]

<working-directory>/repositories/kie . Se puede especificar otro repositorio cambiando la propiedad org.guvnor.m2repo.dir. Cada proyecto genera un archivo JAR llamado Kjar. Se pueden almacenar los componentes del proyecto y los jar dependientes en este repositorio.

Execution Engine : el motor de ejecución de JBoss BPM Suite es responsable de la ejecución de los procesos de negocio y la gestión de las tareas que se derivan de estos procesos. El Business Central proporciona una interfaz de usuario para la ejecución de procesos y la gestión de las tareas.

Business Central Repository : los componentes de un proyecto JBoss BPM Suite como modelos de procesos, reglas y formularios son almacenada en repositorios Git gestionados a través del Business Central. También se puede acceder a estos repositorios de manera externa a Business Central, a través de los protocolos git o SSH.

4.3.1.2. Capacidad de aprendizaje

JBossBPM Suite brinda funcionalidades tanto en el modelador como en el portal que facilitan el aprendizaje de los mismos. El modelador cuenta con tooltips en todos los elementos de la paleta brindando una breve descripción de la funcionalidad del elemento, en caso de que el mismo no sea demasiado intuitivo. El modelador permite la validación del modelo donde en caso de existir errores se marcan las tareas con posibles errores y al

realizar clic en una de ellas se muestran los errores y sugerencias de posibles soluciones. Tanto el modelador como el portal web poseen manuales de uso pero para acceder a los mismos se debe ingresar a la página de jboss bpm suite, los accesos directos no te dirigen a los manuales.

4.3.1.3. Tiempo de respuesta

No fue posible evaluar el tiempo de respuesta para esta herramienta. La causa fue que no se logró iniciar sesión en la aplicación desde la herramienta de performance. En las pruebas realizadas, se lograba realizar la grabación del script, el cual, inicia sesión en la herramienta y luego ejecuta el proceso. Esta grabación se realizó con éxito, pero al momento de la ejecución del test, el mismo no se ejecutaba correctamente e indicaba que el problema se debía a que la sesión expiraba. Una de las medidas tomadas fue obtener el id de sesión que se generaba al momento de la autenticación y luego usarlo en las llamadas del script. Se logró obtener el id de sesión pero de todas maneras las pruebas continuaron fallando, indicando como causa del error que la sesión se encontraba expirada. No se logró identificar la causa de este error, por lo que fue imposible ejecutar las pruebas para esta herramienta.

4.3.1.4. Integridad

La creación de usuarios en JBoss BPM Suite se puede realizar al momento de la instalación. Luego es necesario definir los usuarios mediante la ejecución de un archivo ejecutable. La herramienta provee la funcionalidad del acceso diferenciado según el rol del usuario. El acceso a los documentos es diferenciado según el rol, pero los mismos no se guardan cifrados en la base de datos. La sesión no expira, por lo que permanece activa hasta que el usuario decida cerrarla.

4.3.2. Bonita BPM

La versión utilizada para la evaluación es la *Community 7.1* cuyos componentes se obtuvieron mediante descarga en la página de la herramienta.

4.3.2.1. Arquitectura del BPMS

El motor de Bonita Suite está compuesto por API's y servicios. A su vez los servicios se pueden dividir en servicios genéricos y servicios específicos de BPM.

Este motor es completamente configurable desde un archivo XML. En este archivo se encuentran todos los servicios usados por defecto, así Bonita permite cambiar o reemplazar la implementación de estos servicios.

La instalación del motor se puede hacer en cualquier JVM, contenedor JEE o puede ser usado como una librería JAVA.

Dentro del módulo BPM Services se debe destacar los conectores que son lo que permiten al motor la interacción con una base de datos, con diferentes servidores de correo, acceso a un

LDAP para la autenticación de usuarios y otros sistemas que se pueden ver en [27]. Además de los conectores este módulo se encarga del manejo de los procesos, usuarios y tareas.

En el módulo de Servicios Genéricos encontramos las funcionalidades de autenticación, manejo de eventos y persistencia entre otros.

Por último para el armado del motor de bpm de bonita tenemos el módulo de api, mediante ésta el motor interactúa con el portal y aplicaciones desarrolladas por usuarios.

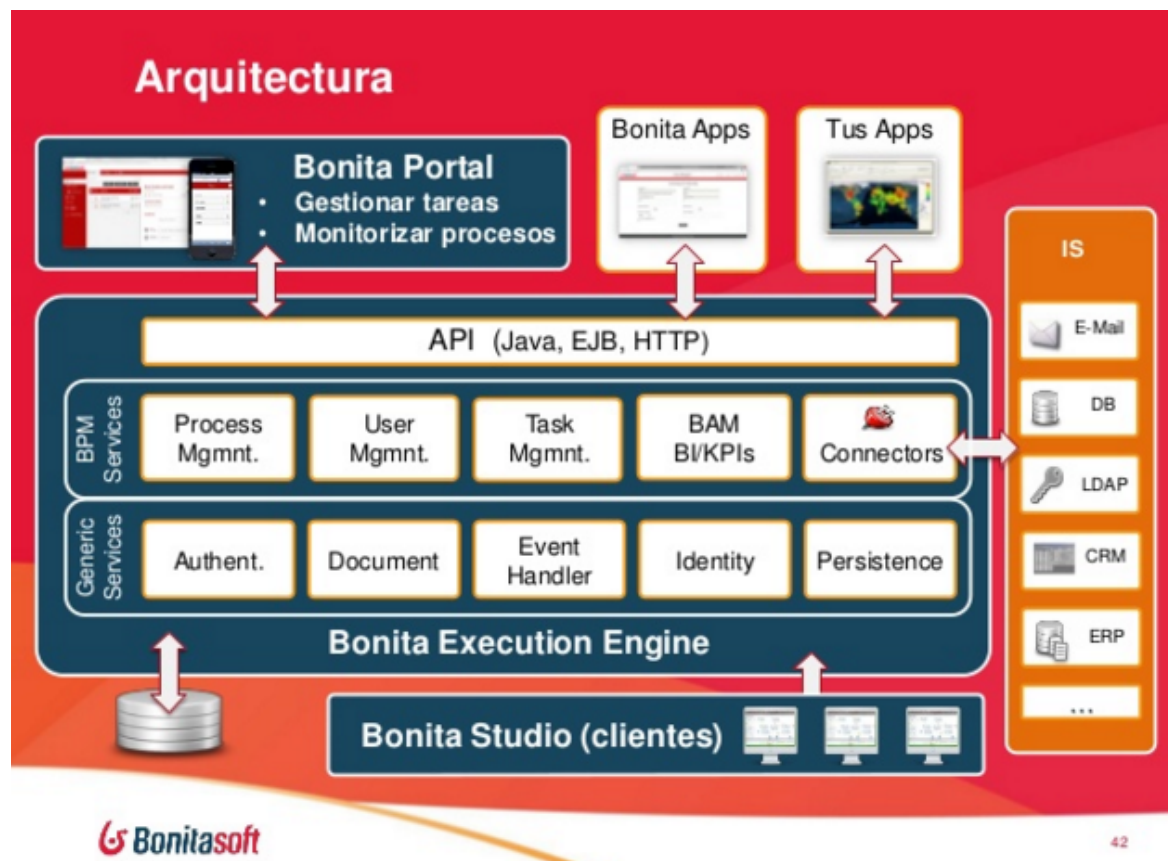


Figura 4.2 Arquitectura Bonita [18]

4.3.2.2. Capacidad de aprendizaje

Bonita BPM cuenta con tutoriales tanto en documentos con formato PDF como vídeos. Si bien estos documentos están solo en inglés, los vídeos se pueden ver en 7 idiomas (inglés, alemán, francés, japonés, español, portugués e italiano). Con esto se logra un aprendizaje eficaz, logrando en poco tiempo poder modelar y ejecutar un proceso.

4.3.2.3. Tiempo de respuesta

Se realizaron varias ejecuciones de un conjunto de test para determinar el tiempo de respuesta. La figura 4.3 muestra el crecimiento del tiempo de respuesta al aumentar la cantidad de usuarios. El gráfico azul representa los valores obtenidos luego de la ejecución y

la función naranja indica la tendencia. La tendencia corresponde a una función lineal, esto implica que a medida que aumenta la cantidad de usuarios, el tiempo de respuesta aumente en forma directamente proporcional.

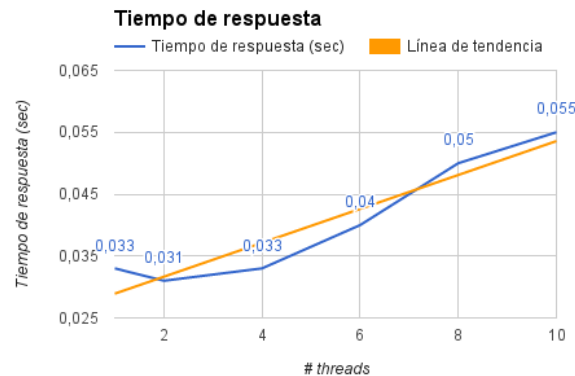


Figura 4.3 Tiempo de respuesta

Tomando un promedio de los valores que se obtuvieron, el tiempo de respuesta para la herramienta es: 0,044 segundos

4.3.2.4. Integridad

El usuario con permisos de administrador puede crear roles y asignar los usuarios a uno o más roles. Según el rol que tenga el usuario podrá ver ciertas funcionalidades o no. Los documentos no se guardan cifrados. Bonita no cuenta con límite de sesión activa.

4.3.3. Intalio|bpms

Se obtuvo una licencia académica a través del sitio web de la herramienta. Junto con la licencia el contacto adjunto los enlaces para las descargas tanto del modelador como del servidor. Se utilizó la edición *Enterprise 7.5.0*.

4.3.3.1. Arquitectura del BPMS

Intalio es una software Open Source cuyo fin es el desarrollo de procesos de negocios basados en los estándares: BPMN, BPEL y BPEL4People. Intalio, permite diseñar, desplegar y optimizar procesos de negocio, sin necesidad de escribir ni una línea de código.

Intalio BPM provee la tecnología para crear una capa que provee servicios (Web Services), que modelen los procesos de negocio de la organización y todas sus reglas de negocio. El BPM debe descansar sobre una arquitectura orientada a servicios (SOA).

La figura 4.4 muestra los componentes de la arquitectura y su interacción.

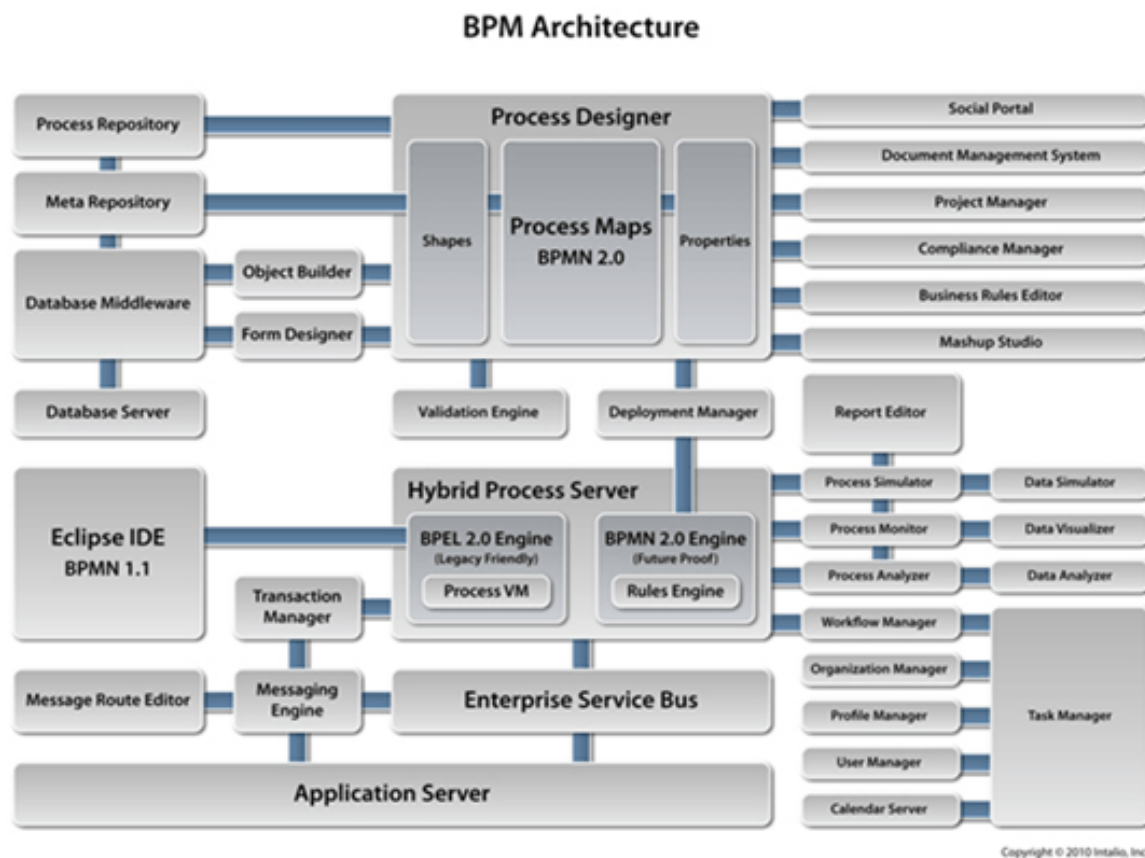


Figura 4.4 Arquitectura Intalio\Designer [28]

[28] describe la arquitectura de Intalio y muestra sus dos componentes principales:

Designer: dedicado al modelado del proceso, implementación y deployment.

Servidor: dedicado a la ejecución de procesos.

Los componentes a su vez están integrados por diferentes módulos. Algunos que componen la edición community y otros, sólo disponibles para la edición enterprise. A continuación se describen los componentes más importantes de *Intalio\Designer* e *Intalio\Server*.

Intalio\Designer

BPMN Modeler es la herramienta que permite generar el modelo del proceso.

Deployment Manager garantiza la condición de cero código, es decir, con este módulo, un usuario es capaz de crear un proceso sin necesidad de utilizar código.

Process Implementation Tools permite manipular los datos de configuración y tiempo de ejecución.

Database Connector permite a Intalio tener un sistema de gestión de bases de datos más general.

Form Editor provee la herramienta AJAX que tiene un entorno para ver, crear e integrar componentes a un formulario.

Para la edición enterprise, se agregan los siguientes módulos:

BAM (Business Activity Monitoring) refiere a las prácticas necesarias para:

- Proveer información relevante y de forma oportuna.
- Realizar toma de decisiones relacionadas al resultado del análisis.
- Implementar las acciones necesarias para el mejoramiento del desempeño del proceso.

BRE (Business Rules Engine) es el módulo que contiene el motor de reglas de negocios.

El módulo provee una herramienta llamada *business freandly decision table editor*, que está incluida en el editor y se puede incluir al modelo utilizando drang-and-drop.

LDAP es el sistema de gestión de usuarios.

Intalio|Server

Intalio permite utilizar como servidores Tomcat, JBoss o Weblogic.

A continuación se describen los módulos que componen Intalio|Server:

BPEL Engine es el motor de procesos de negocios, Intalio utiliza Apache ODE[29]

Tempo BPEL4People gestiona diversos patrones de flujo de trabajo. Una de sus principales características es que expone sus APIs mediante servicios web.

Los siguientes módulos son dos aplicaciones web:

Process Console esta aplicación permite monitorear los procesos habilitados.

Task List interfaz dedicada al usuario final, donde puede gestionar sus tareas.

Para la edición enterprise, se agregan los siguientes módulos:

BAM (Business Activity Monitoring) refiere a las prácticas necesarias para:

- Proveer información relevante y de forma oportuna.
- Realizar toma de decisiones relacionadas al resultado del análisis.
- Implementar las acciones necesarias para el mejoramiento del desempeño del proceso.

BRE (Business Rules Engine) es el módulo que contiene el motor de reglas de negocios.

Permite ver las reglas de negocios independientemente del proceso.

LDAP es el sistema de gestión de usuarios.

ESB (Enterprise Service Bus) permite la integración completa para una arquitectura SOA.

Portal

DMS (Document management system) el sistema de gestión de documentos que se recomienda es Alfresco.

4.3.3.2. Capacidad de aprendizaje

Tanto Intaliol designer como el portal brindan herramientas para un rápido aprendizaje de las mismas. Cada elemento de la paleta que se puede agregar al modelo, posee un tooltip que indica su funcionalidad. Para algunos elementos que no son tan intuitivos, además de la funcionalidad indica cómo se deben utilizar. Si el modelo tiene un error, con una cruz roja indica donde está y si aparece una lámpara, quiere decir que tiene una sugerencia de posible solución. A pesar de no tener manuales de uso, Intaliol designer cuenta con una serie de tutoriales explicativos.

Del lado del portal, solamente los iconos que no tienen una descripción poseen tooltips. Intalio cuenta con una página de soporte donde se encuentran los manuales de uso [30].

Si bien brinda mensajes de errores descriptivos, por ejemplo, cuando un campo del formulario que es numérico se completa con letras, al completar la tarea indica el error; pero no brinda mensajes cuando hay un error al finalizar una tarea debido a un problema de conexión a la base, a un servicio web o un campo que no se completó porque no era obligatorio, pero el valor del campo era utilizado en el proceso.

4.3.3.3. Tiempo de respuesta

Se realizaron varias ejecuciones de un conjunto de test para determinar el tiempo de respuesta. El gráfico de la figura 4.5 muestra el crecimiento del tiempo de respuesta, conforme aumenta la cantidad de usuarios. El gráfico azul representa los valores obtenidos luego de la ejecución, mientras que la función naranja indica la tendencia. La tendencia corresponde a una función lineal, esto implica que a medida que aumenta la cantidad de usuarios, el tiempo de respuesta aumente en forma directamente proporcional.

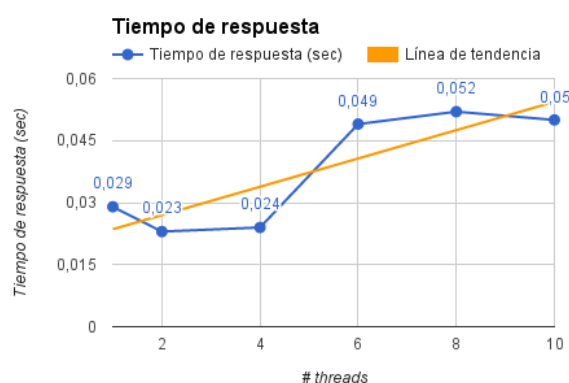


Figura 4.5 Tiempo de respuesta

Tomando un promedio de los valores que se obtuvieron, el tiempo de respuesta para la herramienta es: 0,0395 segundos

4.3.3.4. Integridad

Un usuario administrador es capaz de crear nuevos roles y asignar permisos a dichos roles, esto permite visualizar diferentes funciones según el rol.

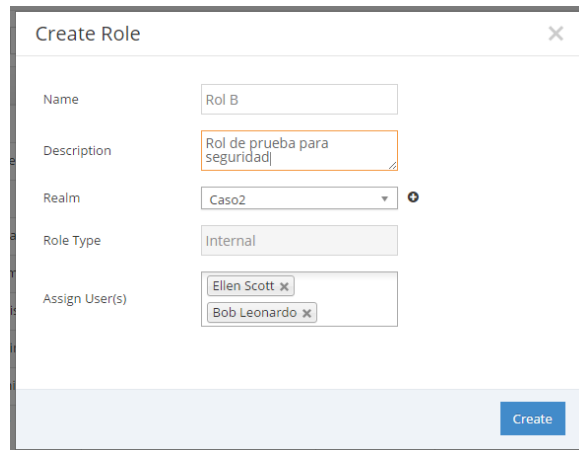


Figura 4.6 Creación de roles

El acceso a los documentos depende del DMS (Document Management System) que se utilice, aunque se puede programar el acceso, según el rol. Los documentos no se guardan cifrados. Intalio cuenta con un tiempo límite de sesión activa.

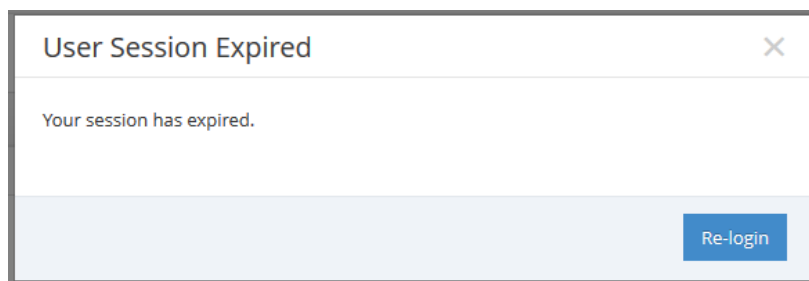


Figura 4.7 Sesión finalizada

4.3.4. Activiti

Para la evaluación de Activiti se utilizó la versión 6.0.0.Beta2 la cual fue descargada de la página de la herramienta[20]. Como es una versión Beta, la documentación no se encontraba actualizada y algunas funcionalidades no estaban disponibles.

4.3.4.1. Arquitectura del BPMS

Activiti es una plataforma BPM orientada a responsables de negocio, desarrollados y administradores de sistemas. Su núcleo es un motor de procesos de BPMN 2 desarrollado en Java. Es una herramienta open-source y es distribuida bajo la licencia de Apache. Activiti

puede ser ejecutado en cualquier aplicación Java, en un servidor, en un cluster o en la nube. Se integra con Spring, es ligera y basada en conceptos simples.

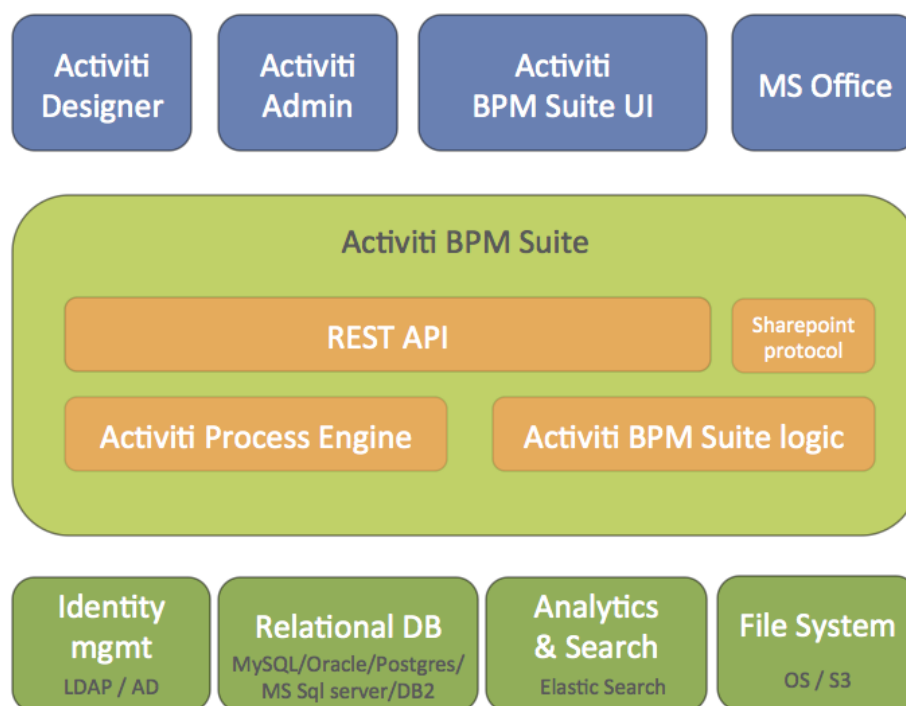


Figura 4.8 Componentes de Activiti [31]

Activiti Engine : es el componente principal del proyecto Activiti. Es un motor de procesos Java que corre procesos BPMN 2 de forma nativa. Mejora la comunicación entre usuarios no técnicos y los usuarios técnicos.

Event listeners: esta funcionalidad permite ejecutar acciones que pueden ser definidas mediante código Java o mediante un *script*. Esto significa que un desarrollador puede decorar el proceso con detalles técnicos que no son mostrados en el diagrama.

Custom activities: en el caso que sea necesario utilizar un tipo de tarea que no esté definido en Activiti, el desarrollador puede crear una actividad *custom* en Java, que implemente el comportamiento que necesita.

Activiti Explorer (Activiti-App en la versión 6): es un ejemplo de aplicación web que provee acceso al Activiti Engine, para todos los usuarios el sistema. Incluye un gestor de tareas, un inspector de instancias de proceso y un gestor de despliegues.

Principales usos del Gestor de Tareas: Ver una lista de tareas personales Ver una lista de tareas para la cual el usuario es candidato. Crear nuevas tareas (no relacionadas a los

procesos) Completar una tarea enviando datos en un formulario de la tarea. Reasignar una tarea a un usuario diferente. Ver que otras personas están involucrados en procesos, en relación a esa tarea.

Principales usos del inspector de instancias de procesos: Ver los detalles de una instancia de proceso. Los dueños del proceso o el administrador pueden ver los detalles de una instancia de proceso que está ejecutando.

Manejo de despliegues: Ver los despliegues Ver las definiciones de aplicaciones

Activiti Modeler Es una aplicación web basada en Angular JS. La principal diferencia con las versiones previas (Signavio Core Components) es que el nuevo modelador es mantenido y desarrollado por Activiti. Activiti Modeler soporta todos los elementos BPM y las extensiones soportadas por Activiti Engine.

Activiti Designer Es un plugin de Eclipse que permite el modelado de procesos BPMN 2.0. Además permite hacer el deploy del proceso en el mismo IDE.

4.3.4.2. Capacidad de aprendizaje

Activiti brinda funcionalidades que facilitan la utilización del modelador. El modelador cuenta con tooltips en todos los elementos de la paleta brindando una breve descripción de la funcionalidad del elemento, en caso de que el mismo no sea demasiado intuitivo. En tanto en el portal web existen algunos elementos que poseen tooltips y otros que no. El modelador no cuenta con un botón específico para la validación del modelo pero no deja incluir elementos de la paleta en lugares que no sean los correctos, coloreando de rojo la zona en la que no se puede colocar el elemento. Ni el portal ni el modelador cuentan con acceso directo a los manuales de usuario, sin embargo existen manuales de usuarios los cuales se encuentran accesibles en la carpeta Doc del código de Activiti o desde la página web de la herramienta pero referencia a una versión anterior del manual.

4.3.4.3. Tiempo de respuesta

Para determinar el tiempo de respuesta se realizaron varias ejecuciones de un conjunto de test. El gráfico de la figura 4.9 muestra el crecimiento del tiempo de respuesta, según aumenta la cantidad de usuarios. En la misma se observan dos gráficos, el azul corresponde al tiempo de respuesta, en tanto el naranja corresponde a la tendencia del gráfico. En este caso se observa una tendencia polinómica de segundo grado, esto quiere decir que el tiempo de respuesta crece en proporción cuadrática de acuerdo a la cantidad de usuarios incorporados.

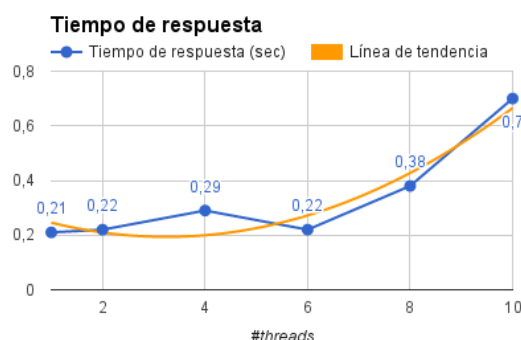


Figura 4.9 Tiempo de respuesta

Tomando un promedio de los valores que se obtuvieron, el tiempo de respuesta para la herramienta es: 0,455 segundos

4.3.4.4. Integridad

Solo el usuario administrador es capaz de crear grupos, nuevos usuarios y asignar los usuarios a grupos. Por lo tanto hay funciones diferenciadas según el rol. El administrador puede acceder al modelado de procesos y a la definición de nuevas aplicaciones mientras que el resto de los usuarios no puede hacerlo. Los permisos, sin embargo, no pueden ser administrados siendo posible ser administrador (superusuario) que posee todas las funcionalidades, o no serlo. Los grupos que no son administradores, solo pueden acceder a las tareas y procesos. El acceso a los documentos esta diferenciado según el rol del usuario, pero los mismos no se guardan cifrados en la base. La sesión no expira, por lo que permanece activa hasta que el usuario decida cerrar la sesión.

4.3.5. Bizagi BPM Suite

Se utilizó una versión completa de la herramienta, únicamente limitada por la cantidad de usuarios del sistema.

4.3.5.1. Arquitectura del BPMS

La arquitectura interna de Bizagi se apoya en un diseño moderno con uso de tecnologías igualmente modernas que permiten ofrecer una solución completa de nivel empresarial.

Bizagi no genera código intermedio y al ser orientado al modelo, se puede fácilmente: configurar una arquitectura de alta disponibilidad, escalar horizontalmente dicha solución, o utilizar mecanismos adicionales a nivel de la base de datos para optimizar el acceso a los datos para las operaciones diarias y para los reportes.

El siguiente diagrama ilustra los componentes principales internos de Bizagi en ejecución y su arquitectura interna:

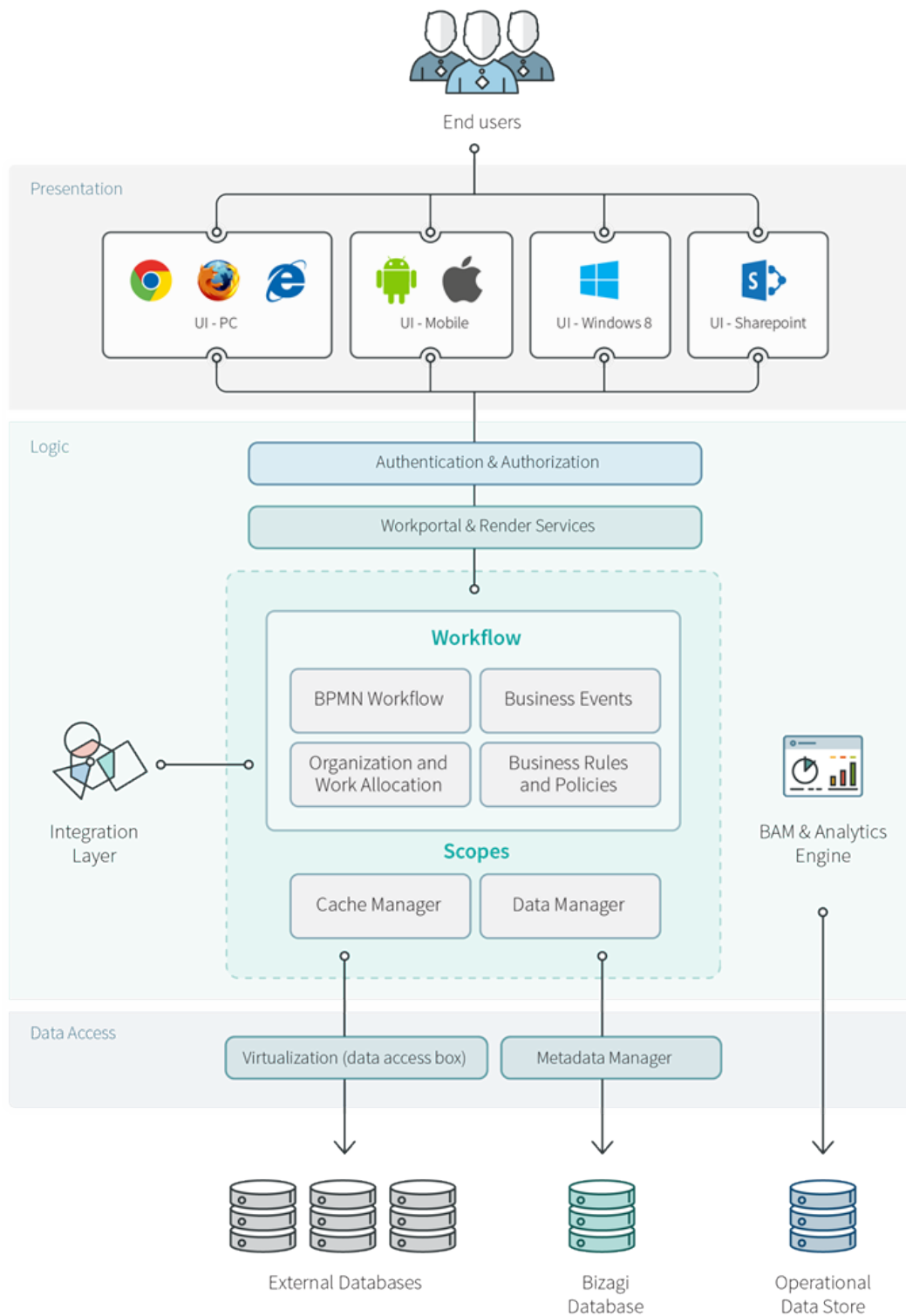


Figura 4.10 Arquitectura Bizagi Suite BPM [32]

Funcionalidad y componentes De acuerdo a la ilustración 4.10, las funcionalidades y características de Bizagi se soportan a través de los componentes principales que se listan a continuación.

Portal de trabajo El Portal de Trabajo es el punto de acceso de los usuarios finales donde podrán interactuar con el sistema (por ejemplo, podrán empezar nuevos procesos, realizar tareas, acceder a reportes, etc.). El acceso al Portal de Trabajo requiere únicamente de un navegador, y éste puede ser configurado para Intranet o para Internet, tener soporte multi-idomas y realizar personalizaciones.

Motor de flujo de trabajo BPMN El Motor BPMN es conocido en Bizagi como el “WorkflowEngine”, y controla y ejecuta el flujo de proceso. Este motor interpreta directamente el modelo BPMN como es creado previamente desde *Bizagi Modeler*. Esto incluye todas las funcionalidades sofisticadas de la notación BPM tales como los elementos de colaboración entre procesos como señales y mensajes, transacciones de larga duración, diseño de subprocessos, entre otros.

BAM y Análisis Las opciones de análisis y BAM proveen una poderosa opción de reportes para indicadores de gestión (para monitoreo, control y mejoramiento de los procesos). Ambas opciones tienen la posibilidad de un análisis gráfico, BAM (análisis en tiempo real) y de análisis histórico (identificación de tendencias, cuellos de botella, etc.). Para las aplicaciones donde hay un gran volumen de datos y consultas de análisis se presenta la opción de utilizar un ODS (Operational Data Store) para el uso de este módulo, y de esta manera mejorar el rendimiento

Administrador de datos Bizagi presenta un potente motor de datos que se conoce como el ‘EntityManager’. Este motor administra la información de manera segura, consistente y sincronizada entre sus diferentes componentes, permitiendo el acceso a los datos desde el proceso.

Seguridad y Organización El módulo de seguridad maneja las opciones de autenticación y autorización para los diferentes roles, habilidades, posiciones, áreas y ubicaciones definidas en la estructura de la organización. Las opciones de Autenticación incluyen Autenticación Windows o Integración LDAP, entre otras.

Reglas de negocio Bizagi ofrece un motor para la ejecución de reglas de negocio o políticas que dan la lógica al proceso de negocio.

Este motor se ejecuta en *Bizagi Engine* para permitir las diferentes transiciones en el flujo del proceso, controlar y sincronizar la ejecución de las actividades, realizar decisiones de forma automática y ejecutar reglas sofisticadas (por ejemplo tablas de decisión, árboles

de decisión) y realizar cálculos, entre otras. Las reglas de negocio se definen en Bizagi de manera asistida, con la posibilidad de un editor gráfico que permita su inclusión sin la necesidad de programación.

Administración de Carga de Trabajo Bizagi permite definir el balanceo de carga de trabajo en la asignación de actividades de acuerdo a los algoritmos principales como: colas, round robin y balance por carga. Para la administración y optimización de la carga de trabajo, es importante tener en cuenta algunos criterios como calendario laboral y calendario de vacaciones de los usuarios, y el uso de delegación de trabajo y reasignaciones.

Manejador de Caché El manejador de Caché es una implementación importante especialmente en proyectos de Misión crítica de clientes corporativos grandes, estos procesos involucran generalmente un gran número de usuarios finales (de forma simultánea). El manejador de Caché provee una mejora en el rendimiento del Portal de Trabajo mediante la optimización al acceso de datos y la persistencia en la base de datos mediante scopes [32].

4.3.5.2. Capacidad de aprendizaje

En el sitio de descarga se ofrece una documentación amplia pero a la vez se puede encontrar algún tema específico rápidamente gracias a su presentación. Tiene una búsqueda del tipo "like" que se encarga de buscar en toda la documentación los temas que hacen referencia a lo que se introduce en ese campo de texto. Además se ofrecen tutoriales en videos en diversos idiomas, depende del idioma que se elija la cantidad de temas que se muestran. La mayoría de los temas tienen su correspondiente tutorial en inglés. En los demás idiomas solo se encuentra tutoriales de temas más básicos.

Bizagi modeler permite a un usuario sin conocimientos técnicos poder modelar los procesos de negocio, con tooltips que ayudan y guían al usuario.

Bizagi Studio introduce un poco más de complejidad en algunos temas como web services, para el cual hay que tener cierto conocimiento para poder realizar la tarea.

Los mensajes de error tanto en el modelador como en el studio son descriptivos.

4.3.5.3. Tiempo de respuesta

No fue posible evaluar el tiempo de respuesta para Bizagi. No se logró iniciar sesión en la aplicación desde la herramienta que se utilizó para medir la performance. Como en las demás herramientas se trató de realizar la grabación del script, o sea, iniciar sesión y ejecutar el proceso. Esta grabación se realizó con éxito, pero al momento de la ejecución del test, el mismo no se ejecutaba correctamente e indicaba que el problema se debía a que no estaba iniciada la sesión. Para poder iniciar sesión igualmente se trató de obtener el id de sesión que se generaba al momento de la autenticación y luego usarlo en las llamadas del script. Esto se

logró pero de todas maneras las pruebas continuaron fallando, indicando el mismo error. Por estas razones no se pudo llevar a cabo las pruebas de performance de esta herramienta.

4.3.5.4. Integridad

Un usuario administrador es capaz de crear nuevos roles y áreas, a partir de esto puede asignar permisos. Se puede asignar una tarea a un grupo o un área en particular y también elegir el tipo de asignación de las tareas para un grupo. Por ejemplo puede distribuir la carga de actividades entre los miembros del grupo.

Los documentos se guardan en un directorio dentro del espacio de trabajo. Estos no se guardan cifrados y además no es posible asignar permisos sobre ellos.

El portal cuenta con un tiempo límite de sesión activa, el cual es configurable. Pasado este tiempo el usuario es redirigido a la pantalla de login.

4.3.6. Camunda

Es una herramienta *open source* que utiliza el estándar BPMN 2.0.

4.3.6.1. Arquitectura del BPMS

Camunda es una plataforma Open Source para la automatización de procesos de negocios. Es ejecutable en BPMN 2.0, es liviana y escala fácilmente.

Camunda está basado en Java. Sus principales componentes están escritos en Java y provee a los programadores Java herramientas para diseñar, implementar y ejecutar procesos de negocio y workflows en JVM. Por otro lado, permite a los programadores que no utilizan Java acceder al motor de procesos a través del REST API, el cual permite que las aplicaciones se conecten de forma remota al motor de procesos.

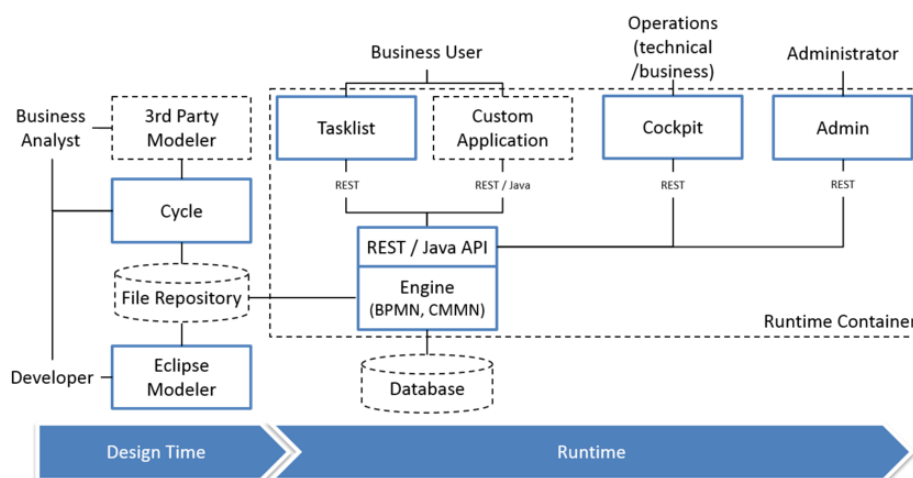


Figura 4.11 Arquitectura Camunda BPM [33]

Camunda BPM puede ser utilizado tanto como un servicio de procesos de negocio independiente, así como también embebido dentro de una aplicación Java ya existente. La figura 4.11 muestra los componentes más importantes al momento de modelar, así como también en tiempo de ejecución.

Motor de proceso e infraestructura

Motor de proceso es una librería de java responsable de la ejecución de workflows y procesos BPMN 2.0. Tiene un núcleo POJO¹ ligero y utiliza una base relacional para la persistencia. La figura B.48 muestra la arquitectura del motor de procesos.

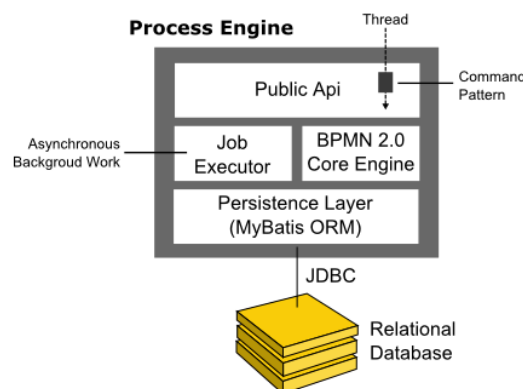


Figura 4.12 Arquitectura del motor de procesos

- Public API es un servicio que permite a las aplicaciones Java comunicarse con el motor de proceso. Las diferentes responsabilidades que tiene el motor de procesos (administrar tareas, la interacción del proceso en tiempo de ejecución, el repositorio del proceso, etc.) están separadas en servicios individuales. Los hilos de entrada al motor de procesos están enrutados a través de un interceptor de Camunda el cual es utilizado para configurar el contexto del hilo como una transacción.
- BPMN 2.0 Core Engine es el núcleo del motor del proceso. Se caracteriza por ser un motor de ejecución ligero para estructuras gráficas (PVM - Process Virtual Machine), un analizador sintáctico de BPMN 2.0 es utilizado para transformar archivos BPMN 2.0 XML en Java Objects, además provee la implementación de un conjunto de comportamientos de BPMN, por ejemplo, constructores como los Gateway o los Service Task.
- Job Executor es responsable de procesar de forma asincrónica el trabajo de fondo, por ejemplo, los Timers.

¹Plain Old Java Object, enfatiza el hecho que las clases java utilizadas son simples y no dependen de un framework.

- The persistence layer es la capa encargada de guardar los datos de las instancias de los procesos en una base de datos relacional.

Spring Framework Integration es un núcleo que da soporte para inserción de dependencias, manejo de transacciones, aplicaciones web, acceso a datos, etc. La integración se encuentra dentro del módulo camunda-engine-spring y puede ser agregado a cualquier proyecto apache basado en maven.

CDI/Java EE Integration El módulo camunda-engine-cdi permite la integración de modelos de programación con CDI (Context and Dependency Injection).

Runtime Container Integration permite la integración con infraestructuras de servidor de aplicaciones.

Aplicaciones web

REST API permite utilizar el motor de procesos desde una aplicación remota o una aplicación JavaScript.

Camunda Tasklist es una aplicación web para el manejo del flujo de trabajo y tareas de usuario que permite a los participantes de los procesos ver sus tareas dentro del flujo de trabajo y navegar por los formularios de las tareas para producir datos de salida.

Camunda Cockpit es una aplicación web para el monitoreo y operación de procesos que permite buscar instancias de proceso, ver sus estados y reparar instancias.

Camunda Admin es una aplicación web para el manejo de usuarios que permite administrar usuarios, grupos y autorizaciones.

Camunda Cycle es una aplicación web para la sincronización de modelos de proceso BPMN 2.0 entre diferentes herramientas de modelado y modeladores.

Herramientas adicionales

Camunda Modeler es un plugin para Eclipse para modelado de procesos.

bpmn.io es un modelador web el cual es utilizado por las aplicaciones web Cockpit y Tasklist para mostrar los modelos de proceso en el navegador.

4.3.6.2. Capacidad de aprendizaje

Camunda modeler es una herramienta que permite un aprendizaje rápido, en particular para los usuarios que alguna vez utilizaron Eclipse. Cada elemento que integra el modelo, cuando se pasa el mouse sobre ellos, se muestra una paleta de opciones, cada elemento de esa paleta tiene un tooltip que no sólo indica que es, sino también su funcionalidad. Camunda modeler cuenta con un manual de usuario [34].

Las aplicaciones web (Admin, Tasklist, Cockpit) poseen tooltip en aquellos iconos que no presentan una descripción. Los mensajes de error son descriptivos, aparecen en rojo, en la parte inferior derecha de la pantalla y, permanecen allí hasta que el usuario los cierra.

También muestra mensajes cuando una acción se realizó correctamente, en ese caso los mensajes son azules. Dentro de la guía del usuario, cada portal cuenta con un manual. Ver [34].

Para finalizar, tanto las aplicaciones web como el plugin de Eclipse brindan sugerencias para solucionar un error. En el caso de Eclipse, el mensaje aparece en la consola de errores con un símbolo de advertencia amarillo. En tanto las aplicaciones web, lo hacen a través de mensajes emergentes.

4.3.6.3. Tiempo de respuesta

El gráfico de la figura 4.13 muestra el tiempo de respuesta en función de la cantidad de usuarios. En la gráfica se observan dos funciones, la de color azul representa los valores obtenidos luego de la ejecución, mientras que la función naranja indica la tendencia. La tendencia corresponde a una función lineal, esto indica que el tiempo de respuesta crece en igual proporción a la cantidad de usuarios que estén utilizando el sistema al mismo tiempo.

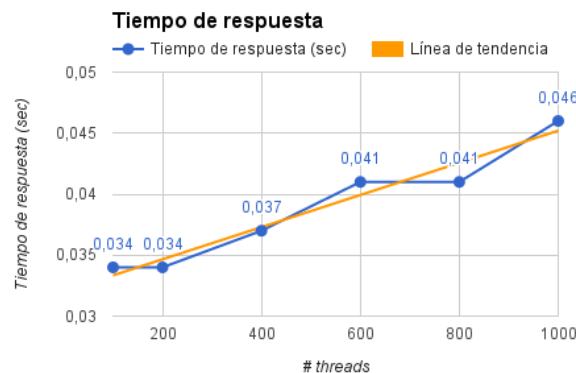


Figura 4.13 Tiempo de respuesta

Tomando un promedio de los valores que se obtuvieron, el tiempo de respuesta para la herramienta es: 0,04 segundos

4.3.6.4. Integridad

Un usuario administrador es capaz de crear nuevos grupos y asignar permisos a dichos grupos, esto permite visualizar diferentes funciones según el grupo al que pertenece un usuario. Además, es posible asignar diferentes permisos a un usuario particular. Camunda cuenta con una aplicación web para administrar a los usuarios, en dicha aplicación un usuario administrador puede dar diferentes permisos y autorizaciones tanto a usuarios como a grupos.

El acceso a los documentos no depende de los permisos que tenga el usuario. Por ejemplo, si se conoce el nombre de la variable donde se cargó el documento, el identificador de la tarea o el identificador de la instancia del proceso, es posible obtener el documento a través

del REST API de Camunda. Los documentos no se guardan cifrados, se guarda un array de bytes en la base de datos. Camunda cuenta con un tiempo límite de sesión activa, pasado ese tiempo, si se intenta realizar una acción, el usuario es redirigido hacia el login.

4.3.7. Aris Express

Para la evaluación de Aris Express se utilizó la versión 2.4, la cual se obtuvo de la página de la herramienta [23].

4.3.7.1. Arquitectura del BPMS

Aris Express es una herramienta de modelado que se basa en los métodos estándares de Aris. Provee una interfaz intuitiva y asistencia para modelado de procesos.

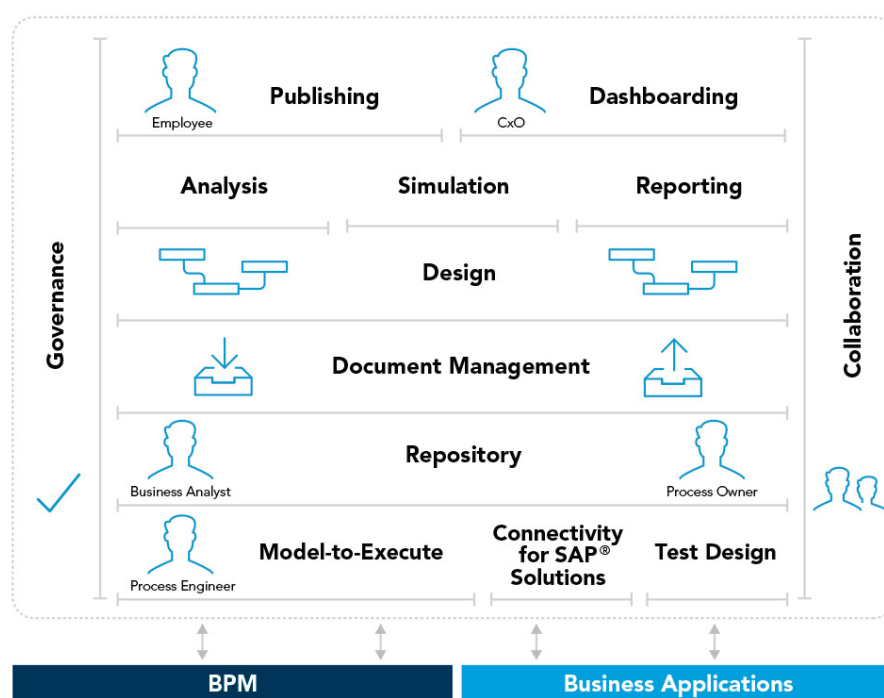


Figura 4.14 Arquitectura - Aris [35]

4.3.7.2. Capacidad de aprendizaje

El modelador de Aris cuenta con una página de inicio en la cual se brinda acceso a una amplia cantidad de acciones y en la que se pueden ver ejemplos de algunos de los modelos así como también se muestra un historial de los últimos modelos guardados.

El modelador posee tooltips que describen la acción del icono.

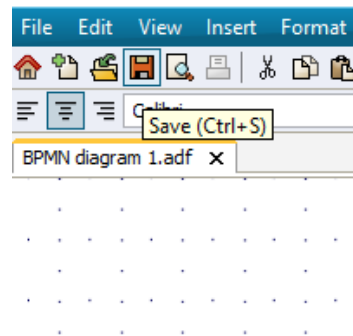


Figura 4.15 Tooltips - Aris

Los mensajes de error que muestran en el modelador son descriptivos y poseen un botón de ayuda en el cual se brindan sugerencia de posibles soluciones. En la barra de tareas existe un menú de ayuda denominado *Help* mediante el cual se puede acceder a los manuales de uso, tutoriales, *help topics* entre otros.

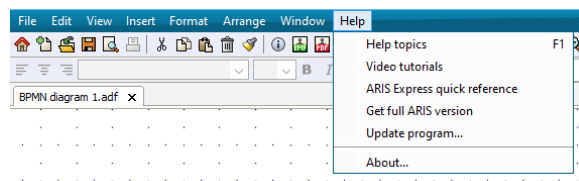


Figura 4.16 Menú de Ayuda - Aris

4.3.8. Orchestra

Es una herramienta open source que utiliza el estándar BPEL. El software se obtuvo del sitio de la herramienta.

4.3.8.1. Arquitectura del BPMS

La "Process Virtual Machine" define un motor genérico de procesos que permite el soporte para múltiples procesos en diferentes idiomas como por ejemplo BPEL y XPD. L.

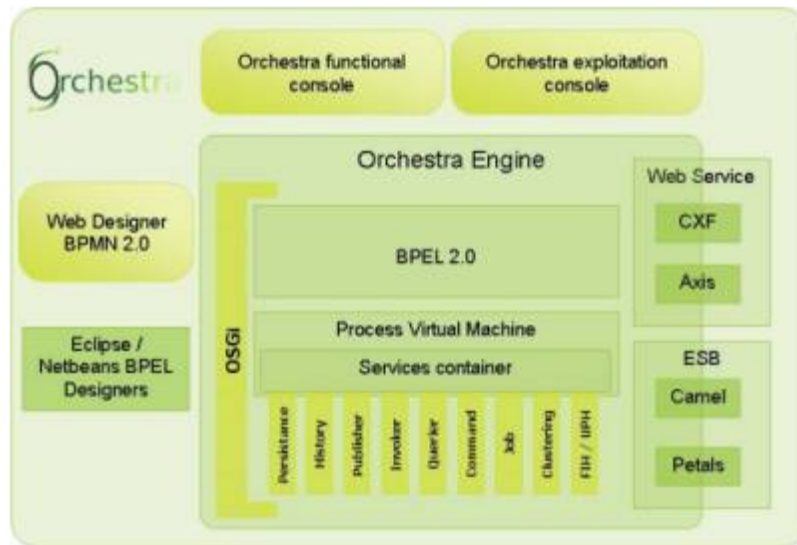


Figura 4.17 Arquitectura - Orchestra [24]

La nueva arquitectura tiene como objetivo crear un acoplamiento flexible entre el motor y la capa de web services y las capas de persistencia, a través del uso de las API en ambos lados (para hacer fácilmente los cambios en las capas de web services o de persistencia)

Además, como las aplicaciones cliente envían sus datos ya formateados en XML, se tiene la necesidad de transformar a XML los datos todo el tiempo, ya que es totalmente portátil y fácil de manipular con el software de código abierto.

Para la capa de web services, necesitamos dos interfaces:

- La API de servicios Web
- El API Engine

La interfaz de los web services tendrá métodos necesarios por el motor, para enviar los servicios de mensajes o llamadas. Y la API del motor tendrá métodos necesarios para el web service para dar mensajes recibidos al proceso, o para indicar que un servicio ha contestado.

Funcionamiento del motor Un proceso BPEL se ejecuta en 2 pasos:

- deploy: en la que los archivos WSDL y BPEL se analizan para crear objetos DOM. Estos DOM se analizan de nuevo para crear la ejecución de Java (con actividades, etc) y

para realizar comprobaciones de validez estáticas como se requiere en la especificación BPEL.

- ejecución: el motor pasa a través de todo el árbol de Java construida durante el despliegue. actuaciones de ejecución son un tema crítico durante la ejecución.

Requerimientos de Hardware Según la documentación se recomienda tener 1GHz de procesador con un mínimo de 512 MB de RAM. También se indica que aumentando estos valores se verá reflejado en la performance del sistema.

4.3.8.2. Capacidad de aprendizaje

El aprendizaje de la herramienta tiene bastantes dificultades ya que no cuenta con tutoriales de vídeo y la documentación que hay es muy escasa, prácticamente sin imágenes. Además hay muy poca participación de los usuarios en los foros.

Los símbolos que se utilizan tanto en el designer como en el portal no son todos mnemotécnicos.

Como positivo hay que destacar que el modelador tiene menú de ayuda y en el sitio tiene un espacio de preguntas frecuentes para el usuario.

4.3.8.3. Tiempo de respuesta

Se realizaron varias ejecuciones de un conjunto de test para determinar el tiempo de respuesta. La figura 4.18 muestra el crecimiento del tiempo de respuesta al aumentar la cantidad de usuarios. El gráfico azul representa los valores obtenidos luego de la ejecución y la función naranja indica la tendencia. La tendencia corresponde a una función lineal, esto implica que a medida que aumenta la cantidad de usuarios, el tiempo de respuesta aumente en forma directamente proporcional.

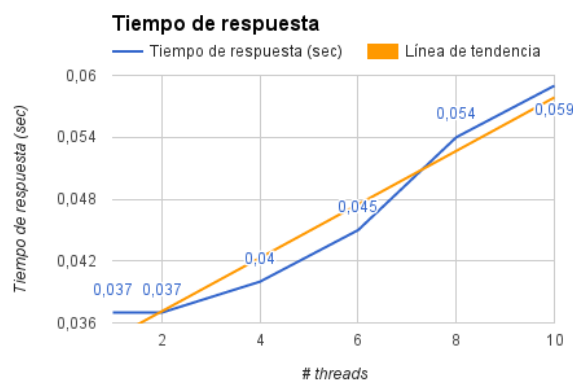


Figura 4.18 Tiempo de respuesta

Tomando un promedio de los valores que se obtuvieron, el tiempo de respuesta para la herramienta es: 0,048 segundos

4.3.8.4. Integridad

Orchestra no implementa los roles o grupos de usuarios, solamente cuenta con la funcionalidad de crear nuevos usuarios mediante configuración en el servidor.

No tiene manejo de documentos en los procesos.

Ni el portal ni el designer cuentan con un tiempo límite de sesión activa.

4.3.9. ProcessMaker

El software se obtuvo a través de un contacto de la empresa luego del registro en la página de la herramienta.

4.3.9.1. Arquitectura del BPMS

ProcessMaker es una solución Open Source que permite el desarrollo de workflow para la automatización y mejora de procesos de negocio. El software está basado completamente en una aplicación web, esto facilita la coordinación del flujo de trabajo entre usuarios, departamentos y organizaciones. ProcessMaker puede interconectarse con sistemas que incluyen la gestión de documentos, ERP, CRM y otras aplicaciones, ya que funciona como una aplicación de SOA de gran alcance.

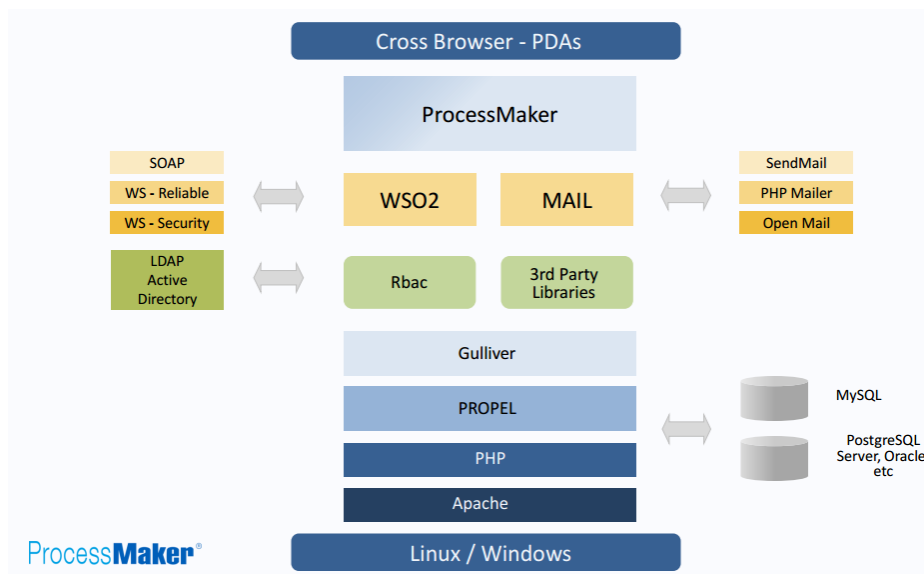


Figura 4.19 Arquitectura BPM ProcessMaker [36]

[36] describe la arquitectura del sistema ProcessMaker, la figura 4.19 muestra los principales componentes de la arquitectura, los cuales se describen a continuación:

WSO2: se utiliza para encriptar la información SSL de 128 bits, esto asegura la confidencialidad e integridad de los datos que forman parte del flujo, o al momento de establecer comunicación con otros sistemas a través de servicios web.

MAIL: este componente permite establecer la comunicación de correos electrónicos a través de cuentas estándares como SMTP.

Rbac: permite el manejo de base de datos de administración de cuentas de usuarios. Además tiene la capacidad de integrarse al directorio activo de Windows.

3er Party Libraries: en este componente se incorporaran las librerías plug-in para la integración con otros sistemas.

GULLIVER: es un framework de desarrollo que está compuesto de una serie de clases que permiten desarrollar nuevas funcionalidades.

Cross browser: permite que el acceso a ProcessMaker sea desde cualquier navegador web (Internet Explorer, Netscape, Mozilla y otros.) Esto garantiza que se pueda trabajar en cualquier tipo de plataforma.

PROPEL: es la técnica de programación ORM (Object Relational Mapping) que convierte datos entre diferentes tipos de sistemas, en base de datos relacional orientado a objetos. Además la técnica permite tener preparado el sistema para manejar base de datos virtual orientado a objetos y, poder ser utilizado por cualquier lenguaje de programación permitiendo el intercambio de información con cualquier base de datos (Oracle, SQL Server, Postgress, MySQL, etc.)

4.3.9.2. Capacidad de aprendizaje

ProcessMaker es un portal web que trae integrado el modelador. Esto permite compartir los manuales de uso, la herramienta cuenta con una wiki [37] pero el primer paso sugerido para aprender a utilizar la herramienta una vez instalada es seguir una demostración para ver los diferentes aspectos que presenta la herramienta [38].

Tanto el modelador como el resto del portal poseen tooltips. En el portal los tooltips que se encuentran describen la acción del ícono, la mayoría de los íconos tienen un texto descriptivo con la utilidad del mismo. Mientras que en el modelador, todos los íconos del modelo poseen tooltips, ya que carecen de texto.

Los mensajes de error que muestran el portal y el modelador son descriptivos. Si bien no brindan sugerencias para solucionar, el mensaje de error es lo suficientemente descriptivo para encontrar una solución.

4.3.9.3. Tiempo de respuesta

Para determinar el tiempo de respuesta se realizaron varias ejecuciones. El gráfico 4.20 muestra el crecimiento del tiempo de respuesta, conforme aumenta la cantidad de usuarios. La función azul representa los valores obtenidos luego de la ejecución, mientras que la función naranja indica la tendencia. La tendencia corresponde a un polinomio de segundo

grado, esto quiere decir que el tiempo de respuesta crece en proporción cuadrática de acuerdo a la cantidad de usuarios incorporados.

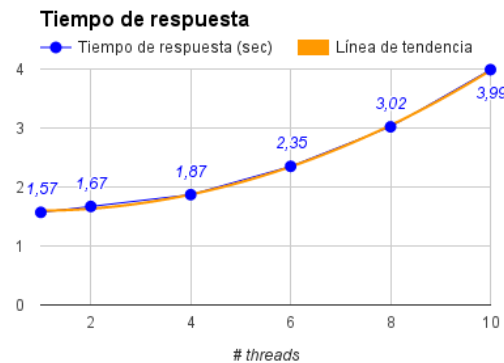


Figura 4.20 Tiempo de respuesta

Tomando un promedio de los valores que se obtuvieron, el tiempo de respuesta para la herramienta es: 2,78 segundos

4.3.9.4. Integridad

Un usuario que tenga permiso de usuarios es capaz de crear nuevos roles y asignar permisos a dichos roles, esto permite visualizar diferentes funciones según el rol del usuario.

Los documentos se guardan en un directorio dentro del espacio de trabajo en el cual se ha iniciado sesión dentro del portal. Si bien los documentos no se guardan cifrados, es posible asignar permisos sobre el documento, de manera tal, que aunque un usuario sin permiso puede acceder a él y descargarlo, no será capaz de ver su contenido. También es posible asignar permisos para crear, ver y eliminar carpetas, así como también para agregar archivos a las mismas.

El portal cuenta con un tiempo límite de sesión activa.

4.3.10. Comparación cuantitativa de resultados

La tabla 4.2 muestra los resultados de las características evaluadas en cada herramienta. La metodología define dos umbrales para comparar resultados cuantitativos, en este caso los umbrales se definieron de la siguiente manera:

- umbral mínimo: 1
- umbral máximo: 2

En consecuencia, se generan tres intervalos que corresponden a tres niveles de cumplimiento: alto, medio y bajo.



A continuación se muestra la tabla comparativa para las herramientas, JBPM Suite, Bonita BPM, Intalio bpm, Activiti, Bizagi, Camunda, Orchestra y ProcessMaker:

Tabla 4.2 Comparación de resultados cuantitativos

Características								
Tecnología, Arquitectura e Interoperabilidad								
Compatibilidad con estándares	●	●	●	●	●	●	●	●
Base de datos	●	●	●	●	●	●	●	●
Sistema operativo del modelador	●	●	●	●	●	●	●	●
Navegador Web	●	●	●	●	●	●	●	●
Extensibilidad	●	●	●	●	●	●	●	●
Integración con Servicios Web	●	●	●	●	●	●	●	●
Diseño y modelado de procesos								
Interfaz gráfica para key users	●	●	●	●	●	●	●	●
Tipo de modelador	●	●	●	●	●	●	●	●
Notación gráfica de procesos compatible con BPMN 2.0	●	●	●	●	●	●	●	●
Gestión de formularios								
Definición de formularios	●	●	●	●	●	●	●	●
Formato de formularios	●	●	●	●	●	●	●	●
Elementos de formularios	●	●	●	●	●	●	●	●
Formularios dinámicos	●	●	●	●	●	●	●	●
Reglas de llenado de campos	●	●	●	●	●	●	●	●

Características								
Validación de datos del formulario	●	●	●	●	●	●	●	●
Motor de Workflow								
Iniciación de procesos en forma manual y automática	●	●	●	●	●	●	●	●
Definición de reglas de negocio	●	●	●	●	●	●	●	●
Asignación de tareas	●	●	●	●	●	●	●	●
Reasignación de tareas	●	●	●	●	●	●	●	●
Adjuntar documentos a tareas	●	●	●	●	●	●	●	●
Versionado de procesos	●	●	●	●	●	●	●	●
Tareas de borrador	●	●	●	●	●	●	●	●
Gestión, monitoreo y auditoría								
Gestión de instancias por usuarios	●	●	●	●	●	●	●	●
Mantenimiento histórico de procesos	●	●	●	●	●	●	●	●
Monitoreo de procesos	●	●	●	●	●	●	●	●
Portal								
Presentación de las tareas	●	●	●	●	●	●	●	●
Usabilidad del modelador								
Capacidad de aprendizaje	●	●	●	●	●	●	●	●
Capacidad para ser usado	●	●	●	●	●	●	●	●
Protección contra errores de usuarios	●	●	●	●	●	●	●	●
Estética de la interfaz de usuario	●	●	●	●	●	●	●	●
Accesibilidad	●	●	●	●	●	●	●	●
Usabilidad del portal								
Capacidad de aprendizaje	●	●	●	●	●	●	●	●
Capacidad para ser usado	●	●	●	●	●	●	●	●
Protección contra errores de usuarios	●	●	●	●	●	●	●	●
Estética de la interfaz de usuario	●	●	●	●	●	●	●	●
Accesibilidad	●	●	●	●	●	●	●	●

Características								
Seguridad								
Confidencialidad	●	●	●	●	●	●	●	●
Integridad	●	●	●	●	●	●	●	●
No repudio	●	●	●	●	●	●	●	●
Responsabilidad	●	●	●	●	●	●	●	●
Autenticidad	●	●	●	●	●	●	●	●
Performance								
Tiempo de respuesta	●	●	●	●	●	●	●	●
Throughput	●	●	●	●	●	●	●	●
Capacidad	●	●	●	●	●	●	●	●

Referencias:

Nivel de cumplimiento

● Alto || ● Medio || ● Bajo || ● No evaluado

La observación de los resultados de cada herramienta permiten ver cual se destaca más en una categoría. En este sentido, se observa que la característica que satisface mejor las características perteneciente a la categoría “Gestión de formularios” es ProcessMaker seguido por Intalio. En dicha categoría la herramienta menos satisfactoria es Orchestra.

Dentro del módulo técnico - no funcional, no se observa una herramienta que se destaque del resto. En términos de usabilidad del modelado la herramienta menos satisfactoria es ProcessMaker. Mientras que la usuabilidad del portal la herramienta menos satisfactoria es Intalio.

4.4. Resumen

En este capítulo se presentaron las herramientas objeto de la evaluación.

Además, se brindó una breve introducción de los resultados obtenidos para cada herramienta, junto con un análisis de los mismos. Luego de presentar los resultados, se realizó una comparación cuantitativa de los mismos para las herramientas.

Capítulo 5

Caso de estudio

En este capítulo se presenta la especificación del caso de estudio y un breve análisis. La sección 5.1 introduce la relación entre los objetivos del proyecto con el caso de estudio. En la sección 5.2 se presentan los modelos de los procesos implementados, con una breve descripción de cada uno de ellos y la especificación paso a paso de las actividades que conforman el proceso. En la sección 5.3 se realiza un breve análisis sobre los resultados del caso de estudio en cada herramienta. Finalmente en la sección 5.4 se brinda un resumen del capítulo.

5.1. Introducción

Dentro de los objetivos del proyecto se encuentra la realización de un caso de estudio en el que se pueden aplicar un subconjunto de características. A su vez se pide la documentación que describa el caso de estudio y su solución.

En el presente capítulo se especifica el caso de estudio realizado. Es importante marcar la diferencia entre los casos de prueba definidos y el caso de estudio. Los casos de prueba tienen el fin de homogeneizar la forma en que se evalúan las características (cada caso de prueba incluye un conjunto de características a evaluar), mientras que el caso de estudio se centra en realizar un caso común en las herramientas.

Para finalizar, se debe tener en cuenta que el proceso implementado está basado en un caso existente. Sin embargo, modela la realidad de forma diferente, puesto que el objetivo principal del caso de estudio es probar las funcionalidades mencionadas de un proceso de negocios.

5.2. Caso de Estudio

El caso de estudio presentado en esta sección fue realizado con la intención de abarcar la mayor cantidad de construcciones que provee el estándar BPMN 2.0. Entre ellas, subprocesos, distintos tipos de tareas, eventos de tiempo, consumo de servicios, etc. Además de los elementos que habitualmente forman parte de un proceso de negocio.

5.2.1. Solicitud de licencia vacacional

El proceso se inicia cuando un trabajador desea hacer una solicitud de vacaciones. Para esto el trabajador debe llenar una solicitud, ingresando cantidad de días y fecha de inicio de la licencia. Al completar la solicitud, la misma es enviada a la sección registro de la Facultad de Ingeniería para ingresar la solicitud del trabajador. Luego que la sección Registro de la Facultad de Ingeniería ingresa la solicitud, ésta pasa a la sección de RRHH dónde un administrativo verifica los días disponibles con que cuenta el trabajador. Una vez que se verifica la disponibilidad, la solicitud es enviada al Jefe de la unidad para autorizar la solicitud, y en base a esto se aprobará o desaprobará la solicitud. En caso que el trabajador tenga días acumulados o el jefe de la unidad lo haya autorizado, un administrativo del departamento de Recursos Humanos valida y analiza la solicitud. En caso que el Administrativo de RRHH no analice la solicitud antes de los 15 días, la misma pasa a ser analizada por el Jefe de área. En todos los casos se debe notificar al solicitante, en caso de aprobación se indica la cantidad de días y la fecha de inicio de las vacaciones, en caso de rechazo se informa los motivos del mismo.

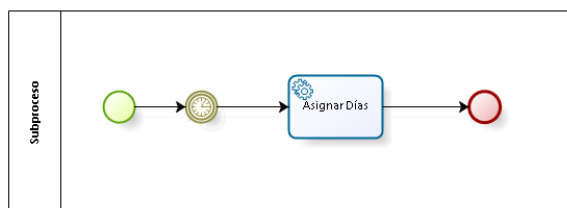


Figura 5.1 Subproceso - Asignar Días

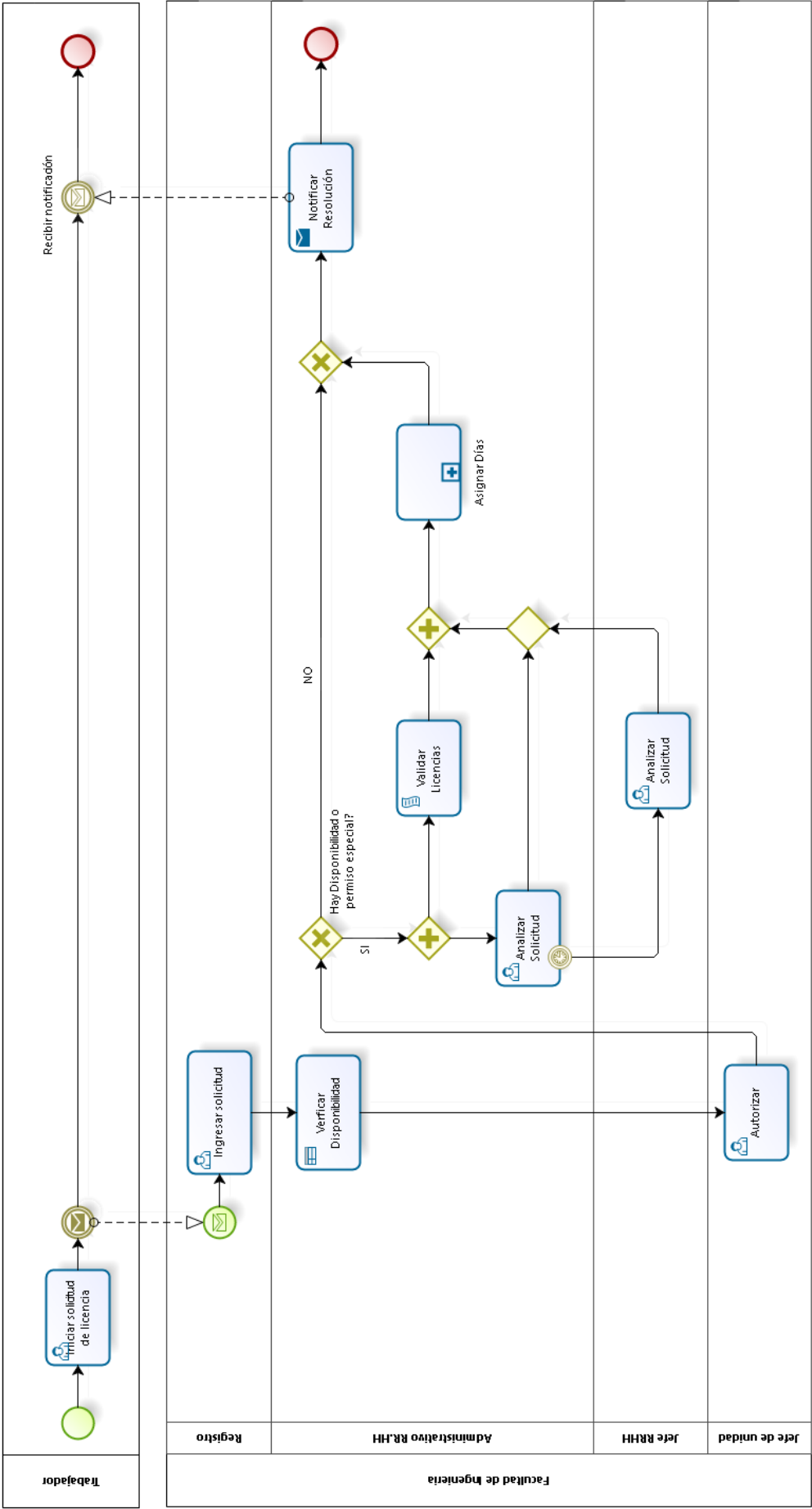


Figura 5.2 Caso de estudio

5.2.1.1. Secuencia del proceso

1. Inicio: El caso de uso comienza cuando en la sección Registro se recibe una solicitud de licencia por parte de un Trabajador.
2. Ingresar solicitud: En esta actividad, un usuario de la sección Registro ingresa los siguientes campos en un formulario: cantidad de días, fecha de inicio.
3. Verificar disponibilidad: Un usuario con rol Administrativo RRHH realiza esta tarea, mediante una regla de negocio, verifica si la cantidad de días ingresados en la tarea *Ingresar solicitud* es menor o igual a los días disponibles que tiene el empleado que solicitó la licencia. Para obtener los días disponibles se realiza una consulta a la base de datos.
4. Autorizar: Un usuario con rol Jefe RRHH marca en un checkbox si autoriza o no la solicitud. En caso de rechazo debe escribir los motivos.
 - En caso de no haber disponibilidad o de no ser autorizada, se notifica el rechazo.
 - En caso de haber disponibilidad o ser autorizada, se realizan las tareas *Validar licencia* y *Analizar solicitud* en paralelo.
5. Validar licencias: Se ejecuta un *Script* y verifica que la cantidad de días solicitados sea menor o igual a la cantidad de días disponibles. Retorna el resultado en una variable.
6. Analizar solicitud: Esta tarea la realiza un usuario con rol Administrativo RRHH. Debe elegir en un radiobutton si aprueba o no la solicitud, en caso de rechazo, debe escribir en un campo de texto el motivo. Esta tarea se debe realizar en menos de 15 días (que se traducen en 15 segundos en el proceso), pasado este tiempo, se realiza un escalamiento reasignando la tarea a un usuario con rol Jefe RRHH.
7. Analizar solicitud: Esta tarea la realiza un usuario con rol Jefe RRHH. Debe elegir en un radiobutton si aprueba o no la solicitud, en caso de rechazo, debe escribir en un campo de texto el motivo.
8. Asignar días: Es un subproceso en el cual se espera una semana antes de asignar los días de licencia correspondientes. Una vez pasada la semana, se consulta mediante un servicio web, el mail del empleado.
9. Notificar resolución: Esta tarea es realizada por un usuario con rol Administrativo RRHH, donde le envía al empleado la resolución de su solicitud. En caso de aprobación se le comunica la cantidad de días y la fecha de inicio de la licencia. Si la solicitud es rechazada se le envía al empleado el motivo del rechazo.

5.3. Análisis del Caso de Estudio

Para realizar el análisis del caso de estudio en cada herramienta, es necesario documentar particularidades de las mismas a medida que se implementa el caso. Es importante definir los aspectos relevantes sobre los cuales realizar el análisis, así como también destacar los cambios realizados en el proceso para poder implementarlo en una herramienta determinada.

El estándar BPMN 2.0 define una extensa cantidad de elementos, de los cuales analizaremos los siguientes:

1. Eventos: mensajes de recepción y envío, temporales ininterrumpibles y de espera.
2. Actividades: tareas de servicios, script, regla de negocio, envío de mensaje y subprocessos.
3. Compuertas: *exclusive* y *parallel*.
4. Comunicación entre *pools*

A continuación se presenta el análisis para cada herramienta, incluyendo imágenes ilustrativas de cambios realizados al modelo para poder implementar el caso.

5.3.1. JBoss BPM Suite

El software de base utilizado para la implementación del caso de estudio fue:

- Base de dato PostgreSQL, versión 9.2. En la misma se encuentra la base de datos de JBoss BPM Suite.
- Se utilizó el navegador Chrome, versión 49.
- El proceso fue ejecutado en el Sistema Operativo Windows 10.

JBoss BPM Suite permite modelar eventos de inicio, de fin, de *throw*, de *catch* y eventos *boundary*. Los eventos necesarios en el caso de estudio se pudieron modelar y ejecutar de forma correcta.

El caso de estudio requiere de la implementación de una gran variedad de tipo de tareas. La herramienta cuenta con tareas de: usuario, envío, recepción, manual, servicio, reglas de negocio, script y vacía. Además provee implementaciones específicas para algunas tareas de servicios, estas son: envío de email, web services, servicios REST y logs. Dado que las tareas que provee la herramienta se corresponden con las solicitadas en el caso de estudio, fue posible implementar todas las tareas del caso de estudio de manera correcta.

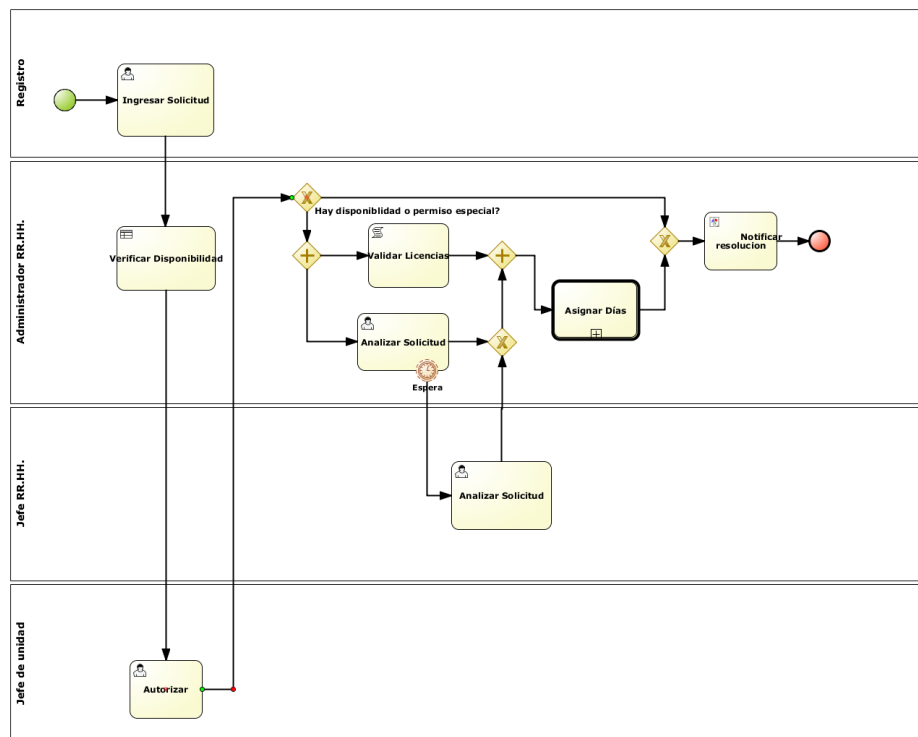


Figura 5.3 Caso de Estudio JBoss BPM Suite

JBoss BPM Suite cuenta con compuertas exclusivas basadas en datos (XOR), basadas en eventos, paralelas e inclusivas. El caso de estudio requiere dos tipos de compuertas, compuertas paralelas y compuertas exclusivas. Para ingresar la condición que establece que camino seguirá el proceso se debe crear una expresión con resultado *booleano*. Esta expresión puede crearse de dos maneras. Una es seleccionando de un combobox una variable del proceso y de otro combobox una condición (es nulo, es falso, es verdadero). La otra forma es escribiendo un script en el cual se pueden utilizar más de una variable del proceso. Fue posible implementar las compuertas requeridas en el caso de estudio.

La herramienta permite la implementación de *Line* y *Sub-process*. Es posible crear subprocesos embebidos o reutilizables. Para la implementación del caso de estudio se utilizó un subproceso reutilizable. Para ello fue necesario crear un nuevo proceso y luego utilizar el identificador del proceso para asociarlo en la actividad del subproceso del caso de estudio.

No fue posible implementar la comunicación entre *pools* dado que la herramienta no cuenta con la definición de *pools*.

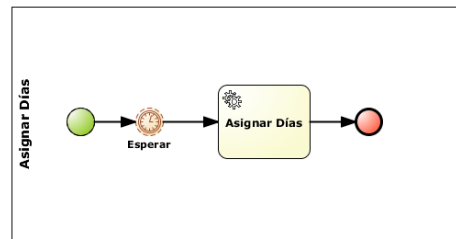


Figura 5.4 Subproceso del caso de Estudio JBoss BPM Suite

5.3.2. Bonita BPM

El software utilizado para la ejecución del caso de prueba fue:

- Navegador Mozilla versión 44.0.2.
- Sistema operativo Windows 10

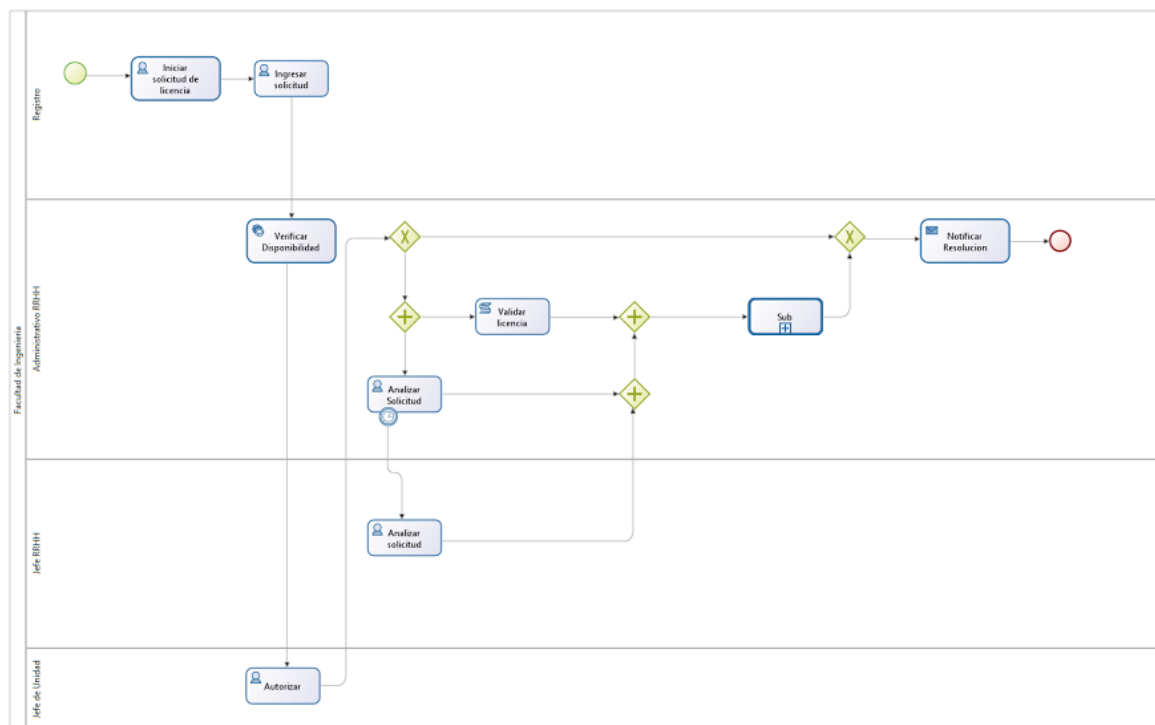


Figura 5.5 Caso de estudio - Bonita

Bonita permite modelar eventos de inicio, intermedios y de fin con envío y recepción de mensajes en ellos. Gracias a estos se logró implementar todos los eventos requeridos en el caso de estudio.

Bonita permita implementar 6 tipo de tareas: de servicio, humana, de script, recibir mensaje, enviar mensaje y abstracta. Para la tarea de regla de negocio se hizo una expresión Groovy.

De esta manera se pudo implementar todas las tareas del caso de estudio.

Para la construcción del caso de estudio son necesarias 2 tipos de compuertas, paralelas y exclusivas. Bonita cuenta con estas 2 y además provee la compuerta inclusiva. En el caso de la compuerta exclusiva requiere de una condición para poder ejecutarla. Dado esto fue posible implementar sin problemas las compuertas del caso de estudio.

La comunicación entre pools se diseñó mediante tareas de mensajes y evento de recepción. En la ejecución se presentaron errores entonces se decidió ejecutar un solo pool, modificando el diseño original.

5.3.3. IntalioIbpms

Para implementar el caso de estudio en esta herramienta se utilizaron las siguientes consideraciones:

- Una base de datos externas. Se utilizó PostgreSQL, versión 9.4. La base de datos tenía la tabla *Empleado*, la misma contenía información sobre los empleados (nombre, email y cantidad de días disponibles).
- Se utilizó el navegador Chrome, versión 49.

Los eventos aceptados por el modelador son eventos de inicio y fin, eventos de *throw* y *catch*. Sin embargo, no acepta tipos de eventos *boundary*, es decir, eventos que se adjuntan a las tareas. Para implementar este tipo de eventos se recomienda incluir un subproceso, porque estos aceptan eventos *boundary*. Para cumplir con el funcionamiento esperado del caso de estudio, se cambió la tarea “Analizar solicitud” por un subproceso que contiene un *time* y dentro un proceso. Este proceso tiene un evento de inicio vacío, una tarea de usuario y un evento de fin. En definitiva, fue posible implementar el comportamiento de los eventos requeridos por el caso de estudio. A pesar de los cambios mencionados anteriormente, se logró mantener el comportamiento esperando.

El caso de estudio dado en la sección 5.2 propone evaluar diferentes tipos de actividades. Respecto a la actividad de tipo *Business Rule Task*, Intalio utiliza tablas de decisión, se creó una nueva, la cual se asoció a la tarea. Este tipo de tareas solo está disponible en la versión *Enterprise*. La *Script Task* solo requiere definir el script dentro del *Mapper* (el *Mapper* es la herramienta que permite asociar variables a los parámetros de entrada y salida del formularios asociado a una tarea).

El modelado de compuertas en Intalio es muy sencillo. Los *Gateway* de apertura, permiten indicar cuál es el flujo por defecto. De la misma manera, en cada flujo de la secuencia se debe definir cuál es la condición que se debe cumplir para que el proceso prosiga por dicho flujo. Las condiciones se definen en el *Mapper*.

Intalio provee dentro de la paleta de objetos que se pueden incluir en el modelo, *Pool*, *Line* y *Sub-process*. Es posible incluir más de un *pool* en cada modelo, cada *pool* tiene la opción de ser ejecutable. En caso de serlo, luego del *deploy*, el proceso que contiene el evento de inicio vacío quede disponible para ser ejecutado. Esto permitió definir tres *pool* dentro del modelo, marcando como ejecutable el proceso que corresponde al trabajador y a la organización. El tercer *pool* no es ejecutable ya que contiene los formularios, la tabla de decisión (contiene la regla de negocio) y el llamado al servicio web que están asociados a alguna tarea.

Los eventos entre los procesos, son sencillos de implementar. La figura 5.6 muestra el modelo con la comunicación entre los *pool*. Una vez establecida la comunicación, fue necesario asociar las variables de entrada y salida dentro del *Mapper*.

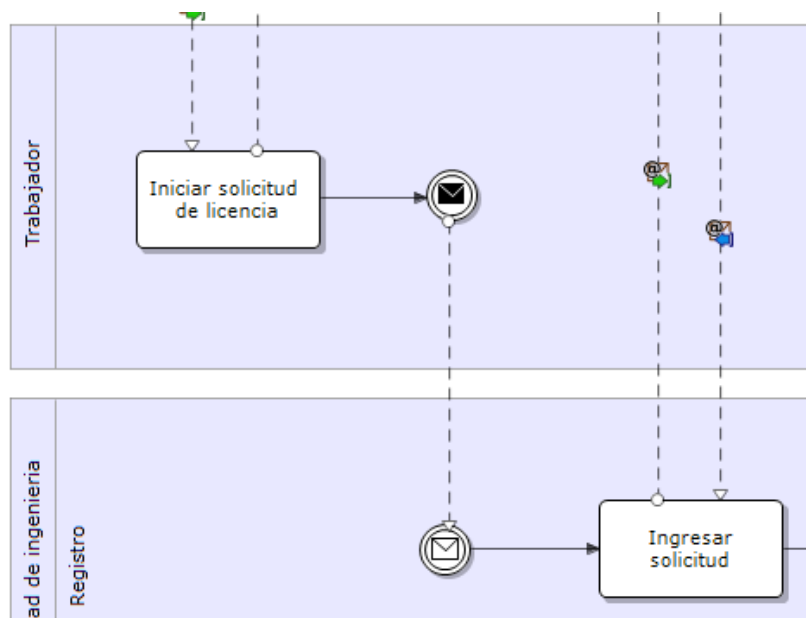


Figura 5.6 Comunicación entre pools

Finalmente, se implementó un subprocesso. Para ello, el modelador permite insertar un subprocesso dentro del modelo, dentro del cual se inserta el proceso. El modelador permite minimizar el proceso, facilitando la legibilidad del mismo.

5.3.4. Activiti

El software de base utilizado para la implementación del caso de estudio fue:

- Base de dato MySQL, versión 5.7.10. En la misma se encuentra la base de datos de Activiti y la base de datos utilizada en el proceso. Esta última contiene la tabla *Empleado* con nombre, email y cantidad de días disponibles.
- Se utilizó el navegador Chrome, versión 49.

- El proceso fue ejecutado en el Sistema Operativo Windows 10.
- Para la implementación de las clases java que fueron necesarias para cubrir algunos aspectos no soportados nativamente por la herramienta, se utilizó Eclipse Mars2.

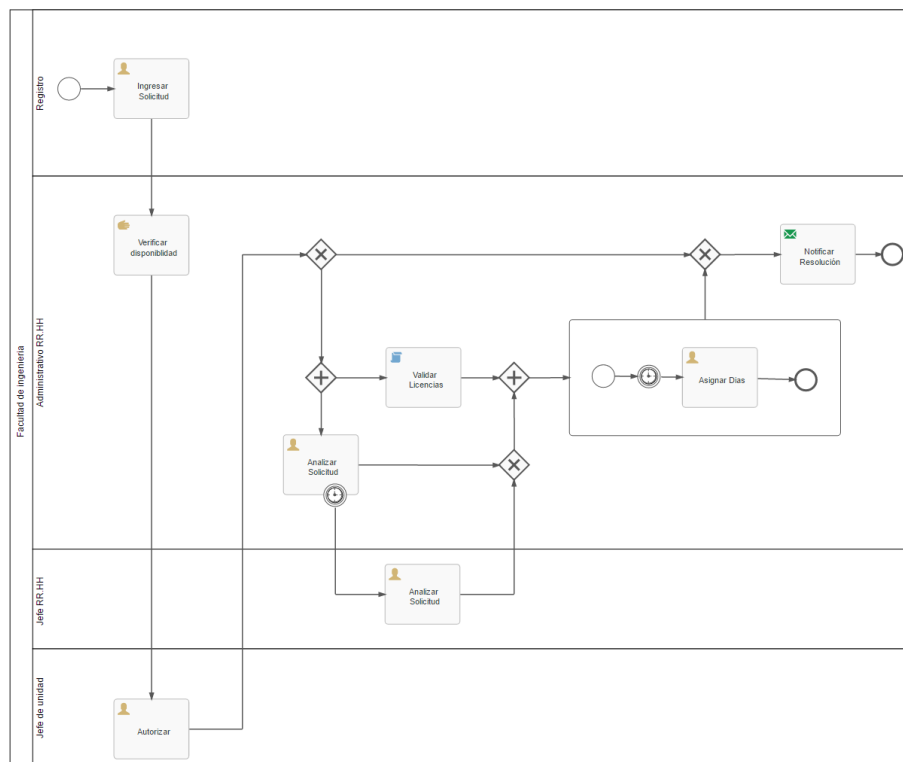


Figura 5.7 Caso de Estudio - Activiti

En el caso de los eventos, Activiti permite modelar eventos de inicio, de fin, de *throw*, de *catch* y eventos *boundary*. Esto implica que fue posible modelar todos los eventos requeridos en el caso de estudio de manera satisfactoria.

Activiti permite modelar una amplia gama de actividades pero no todas pueden ser ejecutadas. De las actividades presentes en el caso de estudio, es posible ejecutar: tareas de usuario, de *email* y de *script*. Las tareas de servicio y de regla de negocio no pudieron ser utilizadas de forma nativa, sino que fue necesario la implementación de *TaskListener* que realizaban las funcionalidades que deberían ser realizadas por las tareas antes mencionadas. En resumen, Activiti no permite todas las actividades requeridas en el caso de estudio, pero si se realizan particularizaciones es posible ejecutar el caso de estudio en su totalidad.

El caso de estudio requería de dos tipos de compuertas, compuertas paralelas y compuertas exclusivas. Para el caso de las compuertas exclusivas es necesario definir la condición que define cuales de los dos flujos será ejecutado. Para definir las condiciones se pueden utilizar

variables de procesos y se usan conectores *booleanos* para indicar que se deben cumplir más de una condición. Fue posible utilizar los dos tipos de compuertas necesarias para ejecutar el caso de estudio.

La herramienta permite la implementación de *Pool*, *Line* y *Sub-process*. Se implementó un subproceso. La única manera de implementarlo en Activiti es insertando un subproceso dentro del modelo, y dentro de este insertar el proceso. Para el caso de la comunicación entre *pools* es posible modelar la misma pero no es posible ejecutar el proceso. Según la sección 13.2 de [39] la herramienta sólo permite la definición de un solo *pool* por proceso de otra manera aparece un error al momento del *deploy*.

5.3.5. Bizagi Suite

Para implementar el caso de estudio en esta herramienta se utilizó:

- Base de datos SQLServer, versión 2012. La base de datos tenía la tabla *Empleado*, la misma contenía información sobre los empleados (nombre, email y cantidad de días disponibles).
- Se utilizó el navegador Chrome, versión 49.

Bizagi permite definir eventos de inicio, fin e intermedio, además cuenta con varios tipos de eventos tales como: *Timer*, *Message*, *Signal*, *Compensate*, *Escalation*, *Conditional*, *Paralell Multiple*, *Multiple*. Dado esto se pudo definir de manera satisfactoria lo que requiere el caso de estudio.

Bizagi no provee la capacidad de captar errores de conexión, por ejemplo en la actividad que consume un servicio web, si al ejecutar esta tarea el usuario no tiene acceso a Internet se debería captar un error, pero éste es ignorado. De todas formas, vía particularización se puede tratar el error.

En este caso particular se requiere implementar dos tipos de compuertas, según una condición y paralelas. Las compuertas pueden implementarse sin problemas.

Bizagi permite la conexión entre *pools* en el modelado y en ejecución. La ejecución de esta interacción tiene una complejidad importante por lo que se decidió modificar el proceso para la ejecución de un solo *pool*.

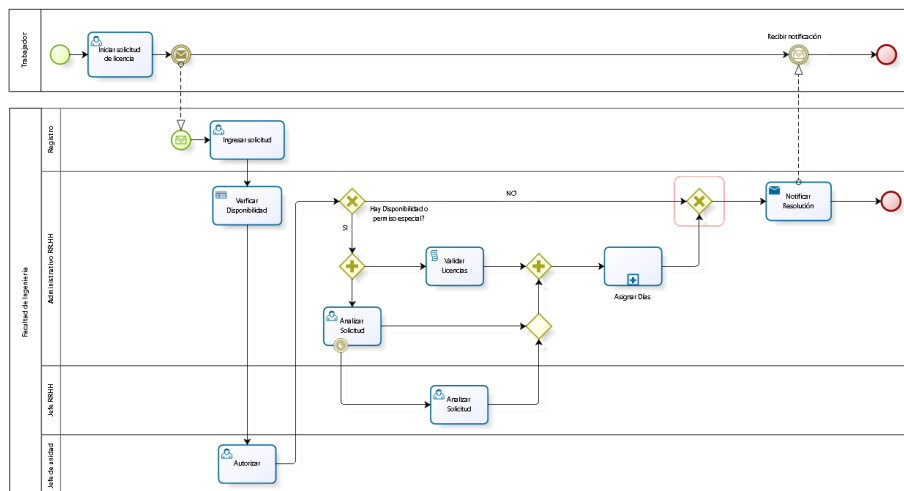


Figura 5.8 Caso de Estudio Modificado Bizagi

5.3.6. Camunda BPMN

Para implementar el caso de estudio en esta herramienta se utilizaron las siguientes consideraciones:

- Una base de datos externas. Se utilizó PostgreSQL, versión 9.4. La base de datos tenía la tabla *Empleado*, la misma contenía información sobre los empleados (nombre, email y cantidad de días disponibles).
- Se utilizó el navegador Chrome, versión 49.

Los eventos aceptados por el modelador son eventos de inicio y fin, eventos de *throw* y *catch*, también soporta tipos de eventos *boundary*, es decir, eventos que se adjuntan a las tareas. En definitiva, fue posible implementar todos los eventos requeridos por el caso de estudios, descrito en la sección 5.2.

El caso de estudio dado en la sección 5.2 propone evaluar diferentes tipos de actividades. Tanto el modelador como el motor de Camunda soportan los siguientes tipos de actividades: manual, de usuario, *script*, *business rule*, *service*, *choreography*, envío y recepción. Con respecto a las actividades de tipo *Service Task*, *Business Rule Task*, fue necesario desarrollar clases java. Ésta última además necesitó la integración con el motor de reglas de negocio Drools. Mientras que la *Script Task* solo requería el desarrollo de un script dentro de las propiedades, ver [34].

En síntesis, Camunda permite utilizar los tipos de actividades requeridos por el caso de estudio antes mencionado.

5.3.6.1. Compuertas

El modelado de compuertas en Camunda es muy sencillo. Los *Parallel Gateway* de apertura, permite indicar cuál es el flujo por defecto. De la misma manera, en cada flujo de la secuencia se debe definir cuál es la condición que se debe cumplir para que el proceso prosiga por dicho flujo.

La herramienta provee dentro de la paleta de objetos que se pueden incluir en el model, *Pool*, *Line* y *Sub-process*, entre otros. Cada *pool* que se incluye en el modelo tiene la opción de ser ejecutable. Si es ejecutable, luego del *deploy*, el proceso que está dentro del *pool* queda disponible para ser ejecutado. Esto permitió definir dos *pool* dentro del modelo, marcando solamente como ejecutable el proceso que corresponde al trabajador.

Con respecto a los eventos entre los procesos, fue necesario implementar clases java para establecer la correlación entre los procesos, ver [34].

Finalmente, se implementó un subproceso. Para ello, el modelador permite insertar un subproceso dentro del modelo, dentro del cual se inserta el proceso. El modelador permite minimizar el proceso, facilitando la legibilidad del modelo, ver figura 5.9

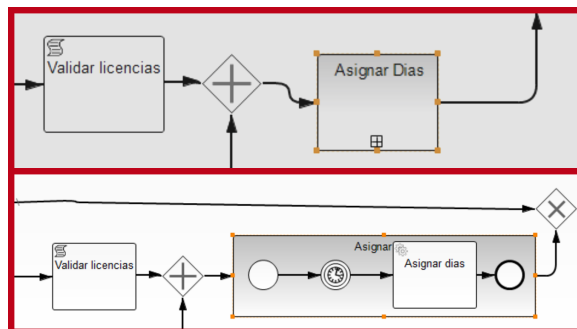


Figura 5.9 Subprocesos

5.3.7. Orchestra

El software utilizado para la ejecución del caso de prueba fue:

- Navegador Chrome versión 49.
- Sistema operativo Windows 10.
- Base de datos PostgreSQL versión 9.2

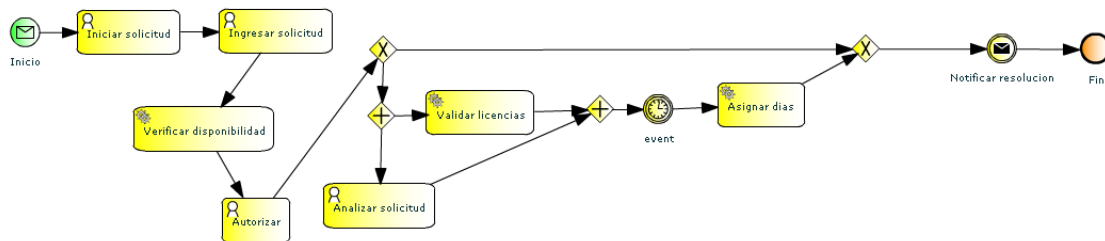


Figura 5.10 Caso de Estudio Modificado Orchestra

Orchestra permite eventos de inicio y de fin, de envío y recepción de mensajes y un evento de *timer*.

Orchestra permite implementar 2 tipos de tareas: de script y humana. Por lo tanto no se pudieron ejecutar las tareas de regla de negocio ni la de servicio, ni la parte de escalización. Tampoco permite ejecutar subprocessos.

Para la construcción del caso de estudio son necesarias 2 tipos de compuertas, paralelas y exclusivas. Orchestra cuenta con estas 2 por lo que en este sentido no hubo problemas para su construcción.

Orchestra no permite ni el diseño ni la ejecución de más de un *pool*. Por lo tanto no fue posible ejecutar 2 pools en el caso de estudio.

5.3.8. ProcessMaker

Para implementar el caso de estudio en esta herramienta se utilizaron las siguientes consideraciones:

- Una base de datos externas. Se utilizó PostgreSQL, versión 9.4. La base de datos tenía la tabla *Empleado*, la misma contenía información sobre los empleados (nombre, email y cantidad de días disponibles).
- Se utilizó el navegador Chrome, versión 49.

El modelador permite incluir en el proceso eventos intermedios de envío de mensaje, pero no de recepción. También permite agregar un evento de inicio de recepción de mensaje y un

evento de fin de envío de mensaje. Este tipo de eventos requieren de un *Message Type*, el cual se define dentro del modelo y queda disponible para todo el proceso, ver [37]. Sin embargo, el motor provisto para la versión *Community* no reconoce estos eventos. Por lo tanto, el caso de estudio para esta herramienta fue modificado, eliminando los eventos de mensajes, además se modificó el inicio del proceso, dejando solamente el *pool* correspondiente a la organización, cambiando el evento de inicio con recepción de mensaje por un inicio vacío.

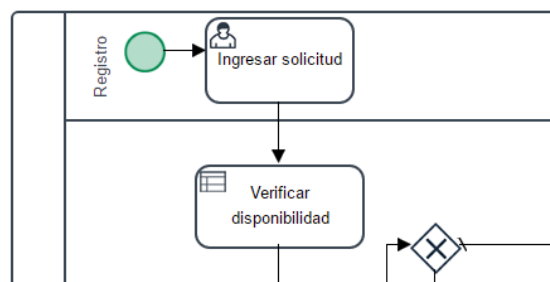


Figura 5.11 Evento de inicio - Caso de estudio

Se presentó la misma situación al ejecutar el subprocesso *Asignar Días*. El subprocesso presentaba un evento *Timer Intermediate*, el cual provee el modelador, pero el motor no reconoce. Se procedió a modificar el subprocesso, eliminando el evento de *timer* y agregando en la tarea un *trigger* para indicar la duración de la tarea.

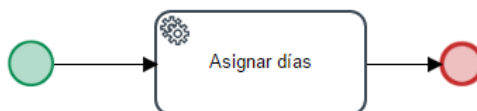


Figura 5.12 Subproceso - Caso de estudio

La versión *Community* de ProcessMaker provee un motor que no es capaz de distinguir los distintos tipos de actividades, tampoco provee un motor de reglas de negocio, por lo tanto no fue posible implementar la actividad *Verificar disponibilidad*. A pesar de esto, es posible simular el comportamiento esperado del proceso con los elementos provistos por el modelador.

Las actividades *Verificar disponibilidad* (regla de negocio), *Validar licencias* (script) y *Asignar días* (servicio) se implementaron a través de un *trigger*, el mismo permite realizar

conexiones a servicios web, a bases de datos externas o realizar cálculos con variables del proceso.

El caso de estudio requería implementar dos tipos de compuertas, según una condición y paralelas. Las compuertas pueden implementarse sin problemas, teniendo en cuenta que luego de una compuerta paralela convergente debe haber una actividad.

Es posible modelar la comunicación entre *pools*. La herramienta provee un ejemplo, donde explica los pasos a seguir para lograr la comunicación, ver [37]. Si bien fue posible modelar la comunicación, no fue posible ejecutar el proceso ya que la versión utilizada no soporta la ejecución de la comunicación.

5.4. Resumen

Se brindó la definición del caso de estudio. En este sentido, se mostraron los modelos correspondientes al caso junto con la secuencia del proceso, donde se describió cada tarea.

Luego, se presentaron los análisis del caso de estudio para cada herramienta. En este sentido, para cada herramienta, previo al análisis, se brindaron consideraciones a tomar en cuenta, ya que no fue posible implementar todo el caso de estudio en todas las herramientas.

Capítulo 6

Proyecto web

Uno de los objetivos del proyecto, consiste de la implementación de una aplicación web. La misma debe permitir, entre otras, aplicar la metodología para la evaluación de un BPMS, tomando como base resultados de evaluaciones de herramientas previamente existentes en el sistema. También sugerir posible herramientas que cumplan con las expectativas del usuario. En este capítulo se presentan los requerimientos, el diseño y algunos aspectos de la implementación de la aplicación web.

En la sección 6.1 se presentan los requerimientos de la aplicación web junto a una lista con la descripción de los caso de uso del sistema. En la sección 6.2 se muestra la arquitectura de la aplicación y el diseño de la base de datos. En la sección 6.3 se detallan las funcionalidades de la aplicación. Por último en la sección 6.4 se brinda un resumen del capítulo.

6.1. Requerimientos de la aplicación web

En esta sección se describen los requerimientos de la aplicación web. La aplicación debe permitir crear: usuarios, roles, categorías, características, subcaracterísticas, herramientas, versiones de herramientas. Al mismo tiempo, debe permitir ingresar resultados de evaluaciones, obtener evaluaciones ponderando las subcaracterísticas y comparar los resultados de dos herramientas.

6.1.1. Casos de uso

Los casos de uso son una forma de definir las funcionalidades de un sistema. No todos los usuarios pueden realizar todas las acciones, en este sentido se definen tres roles:

Administrador: es el usuario que administra el sistema. Un usuario con rol “Administrador” puede manejar usuarios, roles, categorías, características, subcaracterísticas, herramientas, versiones de herramientas. También puede ingresar resultados, realizar evaluaciones y comparaciones de herramientas.

Evaluador: es el usuario encargado de realizar las evaluaciones. Un usuario con este rol puede ingresar resultados. Además es capaz de manejar herramientas y versiones.

Cliente: este tipo de usuarios solamente pueden realizar comparaciones entre dos herramientas o ponderar los criterios de evaluación para obtener las herramientas que se ajusten más a dichas ponderaciones.

A lo largo de esta sección se utilizará el acrónimo CRUD de Crear, Leer, Actualizar y Borrar (del original en inglés: Create, Read, Update and Delete).

CRUD de Categorías Las categorías se definen mediante un nombre de categoría y un módulo (técnico funcional, técnico no funcional y no técnico). La aplicación permite ver la lista de las categorías ingresadas y para cada una de estas permite modificar los datos, ver la información ingresada en la definición de la misma y borrarlas. Estas funcionalidades se encuentran disponible para los usuarios con rol administrador.

CRUD de Características Las características se definen mediante un nombre, una descripción y además se debe seleccionar la categoría a la que pertenece. Por lo tanto, debe existir al menos una categoría para crear de manera satisfactoria una característica. La aplicación cuenta con la funcionalidad del listado de las características. Es posible ver los datos ingresados de cada característica, modificar una característica y hasta borrarlas. Estas funcionalidades estas disponibles para los usuarios con rol administrador.

CRUD de Subcaracterísticas Según este proyecto una subcaracterística se representa mediante una pregunta. Es por esta razón que para crear una subcaracterística se debe ingresar el título, la pregunta que la define y la característica a la que pertenece. Es necesario para poder crear una subcaracterística, haber creado previamente una característica. La aplicación cuenta con una vista que permite visualizar la lista de subcaracterísticas ingresadas y para cada subcaracterística se puede ver los datos de la misma, modificar sus datos y borrarla. Estas funcionalidades se encuentran disponibles para los usuarios con rol administrador.

CRUD de Herramientas Una herramienta es creada en la aplicación mediante el ingreso de un nombre que la identifique. La aplicación permite ver la lista de herramientas ya definidas, ver el nombre seleccionado para la misma, editar dicho nombre y borrar la herramienta. Los usuarios con rol administrador y evaluador son los que disponen de estas funcionalidades.

CRUD de Versiones de herramientas Una herramienta puede contar con diferentes versiones. Es por eso que la aplicación permite crear versiones y asociarlas a una herramienta existente. Para crear una nueva versión es necesario seleccionar una herramienta y luego definir un nombre para la versión. La aplicación permite ver, editar y borrar una versión. Además en los detalles de una herramienta es posible ver la lista de versiones definidas para la misma. Los usuarios con rol administrador y evaluador son los que disponen de estas funcionalidades.

CRUD de Usuarios La aplicación permite el registro de usuarios. Para registrarse es necesario ingresar nombre, apellido, email y contraseña. Se puede ver el detalle, editar y listar los usuarios. Los usuarios con rol administrador pueden editar y ver los usuarios creados. Sin embargo el registro de usuario es accesible a todos. Los usuarios registrados tienen por defecto el rol cliente.

CRUD de Roles La aplicación permite crear, ver los detalles, editar y listar roles. Por defecto la aplicación cuenta con tres roles: evaluador, cliente y administrador. Para que un usuario pueda realizar alguna función en el sistema debe tener asociado al menos uno de estos tres roles. Solo el usuario con rol administrador posee las funcionalidades antes mencionadas y es el único que puede asociar un rol a un usuario.

Ingresar Resultados La definición de un resultado implica el ingreso de respuestas a las preguntas definidas dentro de las subcaracterísticas. Es por eso que es necesario que exista al menos una subcaracterística en el sistema. Si el tipo de característica es funcional entonces se debe seleccionar nivel de soporte, nivel de cumplimiento e ingresar un valor entre 0 y 1 que corresponde al factor. Si la característica es no funcional se debe seleccionar solo el nivel de cumplimiento. Solo los usuarios con roles administrador y evaluador se encuentran habilitados para ingresar resultados.

Crear Evaluación La creación de una evaluación implica ingresar, el nombre, los valores para los umbrales, elegir las versiones de herramientas y seleccionar para cada subcaracterística el nivel de importancia que puede ser obligatorio, medio y bajo. Además tiene la opción de ignorar la subcaracterística, en ese caso, la misma no es tomada en cuenta para la evaluación. Solo los usuarios con rol administrador y evaluador pueden utilizar dicha funcionalidad.

Comparación de Herramientas La comparación de herramienta permite seleccionar dos herramientas del sistema comparar lado a lado los valores de los resultados ingresados para cada subcaracterística del sistema. Cualquier usuario que tenga asignado al menos un rol (administrador, evaluador, cliente) puede utilizar esta funcionalidad.

6.2. Diseño

La siguiente figura muestra el modelo entidad relación o MER utilizado en la aplicación web. En el mismo se pueden ver las entidades (User, Rol, UserRol, Category, Characteristic, Subcharacteristic, SubCharImportance, Tool, Version, Result, ObjectiveEvaluation, Evaluation, Threshold, Importance) y la relación entre ellas.

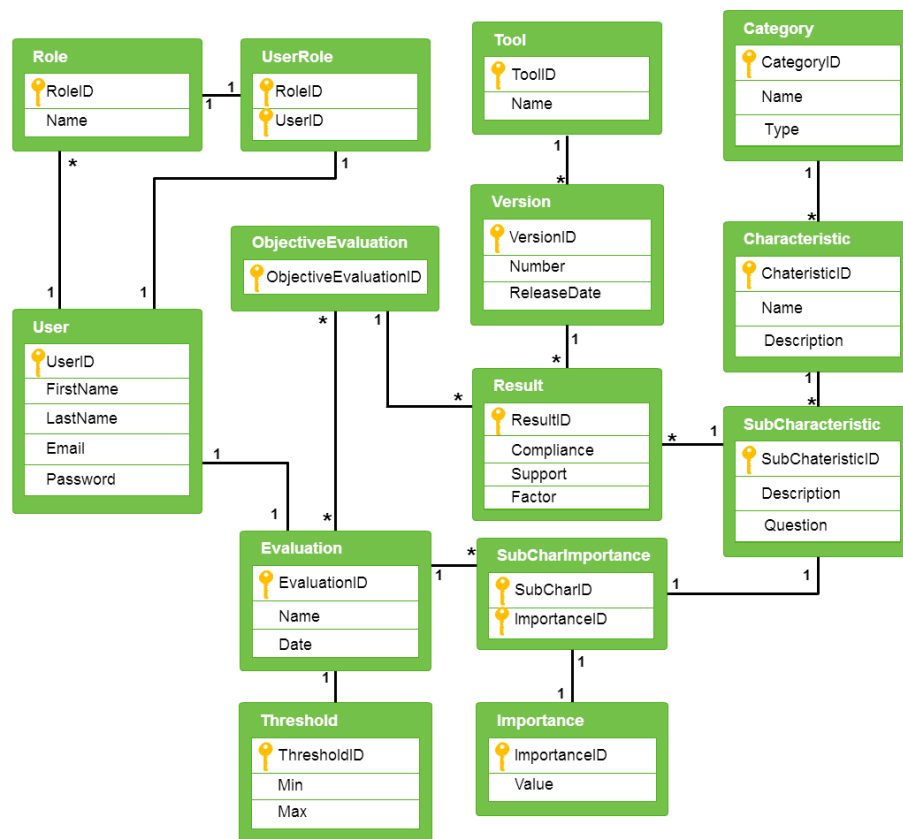


Figura 6.1 Modelo de la base de datos de EvalBPMS

EvalBPMS consta de dos componentes, el servidor de aplicaciones (IIS) y la base de datos (MySQL). La aplicación web que se encuentra alojada en el servidor de aplicaciones y esta se comunica con la base de datos, tanto para almacenar datos como para consultarlos. En la siguiente figura se muestra la comunicación de los usuarios con la aplicación web y de ésta con la base de datos.

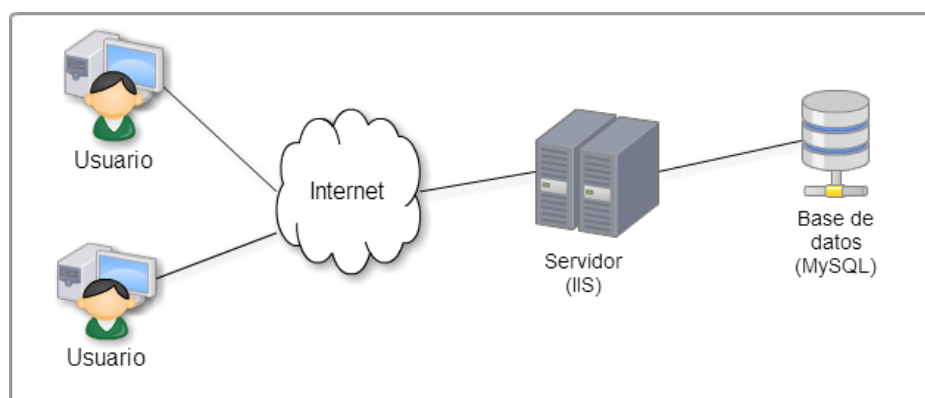


Figura 6.2 Arquitectura de EvalBPMS

6.3. Implementación

Para la implementación de la aplicación web se utilizó el framework .NET 4.5.2 con MVC 4 y EntityFramework para la creación y actualización de la base de datos (MySQL). El lenguaje de desarrollo utilizado es C#.

Pantalla de inicio La pantalla de inicio es donde el usuario encuentra toda la información necesaria sobre el sistema. A la izquierda se puede ver la barra de navegación donde se muestran todas las funcionalidades que puede realizar el usuario administrador.



Figura 6.3 Pantalla de inicio EvalBPMS - Administrador

A continuación se detallan algunas de las funcionalidades más relevantes de EvalBPMS.

Resultados Realizando click en el link del menú *Resultados*, se accede a la pantalla donde se listan todos los resultados del sistema. Un resultado es un conjunto de valores asociados a una subcaracterística, que surgen de aplicar la metodología definida a un BPMS. No siempre los resultados implican un valor para todas las subcaracterísticas definidas, sino que pueden ingresarse resultados parciales.

Ingresar resultado

Herramienta: 6.2.0

Técnico - Funcional

Tecnología, Arquitectura e Interoperabilidad

Arquitectura del BPMS

¿Cuál es el nivel de cumplimiento?

No aplica

No soporta

1

Compatibilidad con estándares

Navegador Web

Base de datos

Extensibilidad

Técnico - No funcional

Usabilidad

Protección contra errores de usuarios

¿Tiene mensajes de advertencia?

No aplica

¿Posee validación de datos y formularios?

No aplica

¿Cuenta con opciones para deshacer y rehacer una acción?

No aplica

Capacidad para ser usado

Estética de la interfaz de usuario

Accesibilidad

Figura 6.4 Ingreso de Resultados en EvalBPMS

- **Ingresar Resultados:** Realizando click en el link *Ingresar resultado* se accede a la pantalla de ingreso de resultados. En esta pantalla se listan todas las categorías, características y subcaracterísticas y dependiendo del módulo al que pertenezca la subcaracterística deberá:
 - **Módulo TécnicoFuncional:** seleccionar el nivel de cumplimiento, el nivel de satisfacción y en caso de que el nivel de cumplimiento sea vía particularización se debe ingresar un factor del 0.1 al 0.9 que se corresponde con el nivel de dificultad asociado a la particularización.
 - **Módulo TécnicoNo Funcional:** seleccionar el nivel de cumplimiento.

Una vez que se hayan ingresado todos los datos realizar click en el botón *Guardar* y se mostrarán los resultados antes ingresados asociados a la herramienta

- **Editar Resultados:** Si se desea editar un resultado, se puede realizar click en el ícono con del lápiz. El usuario es redirigido a la pantalla de edición, en ésta es posible modificar los valores seleccionados para las subcaracterísticas. Una vez que se modifiquen los datos deseados se debe realizar click en el botón *Guardar* y las modificaciones quedarán guardadas en el sistema.
- **Borrar Resultados:** Para borrar un resultado se debe realizar click en el ícono de la papelera. Una vez que se realiza clic el usuario es redirigido a la pantalla de confirmación de eliminación en la que se consulta al usuario si está seguro que desea borrar ese resultado, mostrando la herramienta a la cual corresponde ese resultado. Si

se realiza clic en el botón *Borrar*, el resultado es eliminado de manera permanente del sistema.

- Ver Resultados: Si se desea ver un resultado, se puede realizar click en el ícono del ojo. En la vista *ver resultado* se puede encontrar una tabla con las características y subcaracterísticas y para cada una de ellas los valores ingresados.

Resultados de JBoss BPM Suite v.6.1.0

	Resultado
Módulo Técnico - Funcional	
Diseño y modelado de procesos	
Notación gráfica de procesos	
¿Cuál es el nivel de cumplimiento?	● ✓ 1
Tipo de modelador	
¿Cuál es el nivel de cumplimiento?	● ✓ 1
Interfaz gráfica para key users	
¿Cuál es el nivel de cumplimiento?	● ✓ 1
Gestión de Formularios	
Validación de datos	
¿Cuál es el nivel de cumplimiento?	● ✓ 1
Formularios dinámicos	
¿Cuál es el nivel de cumplimiento?	● ✓ 1

Figura 6.5 Ver resultados en EvalBPMS

Evaluaciones Realizando click en el link del menú *Evaluaciones*, es posible acceder a la pantalla donde se encuentra el historial de las evaluaciones. Dado que las evaluaciones toman en cuenta el interés del usuario, la lista de evaluaciones es particular de cada uno de estos. Por lo que cada usuario solo verá sus evaluaciones. El usuario administrador es el único usuario capaz de ver todas las evaluaciones del sistema.

Tus evaluaciones

[Crear](#) 

Nombre	Fecha	
Para D. Calegari	12/06/2016 19:57:39	  
Para A. Delgado	06/05/2016 20:11:00	  

© 2016 - eval(BPMS)

Figura 6.6 Lista de Evaluaciones de EvalBPMS

- **Crear Evaluación:** Realizando click en el link *Crear* se accede a la pantalla de creación de evaluación. En esta pantalla se listan todas las características junto a sus subcaracterísticas. El usuario debe seleccionar los valores para los umbrales mínimos y máximos y luego debe seleccionar la importancia de cada una de las subcaracterísticas que desea evaluar. Solo las subcaracterísticas para las que se hayan seleccionado el nivel de importancia son la que serán tenidas en cuenta para la evaluación. Una vez que se hayan ingresado todos los datos realizar click en el botón *Guardar* y se mostrarán los resultados de la evaluación.
- **Editar Evaluación:** Si se desea editar el nombre de una evaluación, se puede realizar click en el ícono con del ojo. El usuario es redirigido a la pantalla de edición, en ésta es posible modificar el nombre. Una vez que se modifica el nombre se debe realizar click en el botón *Guardar* y las modificaciones quedarán guardadas en el sistema.
- **Borrar Evaluación:** Para borrar una evaluación solo se debe realizar click en el ícono de la papelera. Una vez que se realiza click en la misma el usuario es redirigido a la pantalla de confirmación de eliminación. En la misma el sistema consulta al usuario si está seguro de que desear realizar esa acción. En caso de que se realice click en el botón *Borrar*, la evaluación será eliminada del sistema de manera permanente.
- **Ver Evaluación:** Realizando click en el ícono del ojo se pueden ver los detalles de una evaluación. En la misma se listan todas las características junto a sus subcaracterísticas y el usuario podrá ver los resultados de la evaluación. También es posible filtrar los resultados por categoría y ponderación.

	Activiti v.6.Beta2 (1,00)	JBoss BPM Suite v.6.1.0 (2,50)
Módulo Técnico - Funcional		
Portal	0,00	2,50
Presentación de las tareas	⊞ ● (0,00)	● (2,50)
Módulo Técnico - No funcional		
Usabilidad	1,00	0,00
Capacidad de aprendizaje	⊞ ● (1,00)	● (0,00)

Figura 6.7 Ver evaluación de EvalBPMS

Herramientas Realizando click en el link del menú *Herramientas*, es posible acceder a la pantalla donde se listan todas las herramientas existentes en el sistema. Una herramienta consta de un nombre y una versión, por lo que puede existir una herramienta asociada a múltiples versiones.

Lista de herramientas

[Crear](#) +

Nombre	Versiones	
JBoss BPM Suite	6.1.0	✎ 🗑 👁 +Agregar versión
Activiti	6.Beta2	✎ 🗑 👁 +Agregar versión

© 2016 - eval(BPMS)

Figura 6.8 Lista de Herramientas de EvalBPMS

- **Crear Herramienta:** Para crear una herramienta se debe realizar click en el link *Crear*. Una vez en la pantalla de creación, el usuario debe ingresar el nombre de la herramienta que se desea crear y realizar click en el botón *Guardar*.
- **Crear Versión de Herramienta:** Para crear una versión de una herramienta realizar click en el link *Agregar versión*. Una vez en la pantalla de creación, el usuario debe ingresar el número de la versión y la fecha de la misma. Para que la versión sea creada debe realizarse click en el botón *Guardar*.

- **Editar Herramienta:** Realizando click en el ícono del lápiz es posible editar una herramienta. Los datos que se pueden modificar de la herramienta es su nombre. Una vez hecha la modificación se debe realizar click en el botón *Guardar* para que se almacenen los cambios.
- **Editar Versión de Herramienta:** existen dos maneras de editar una versión. Desde la lista de versiones a la cual se puede acceder realizando click en el link *Herramientas* del menú y luego en el link *Versiones* donde se accede a la lista de versiones por herramientas. Aquí debemos realizar click en el ícono del lápiz que se encuentra al lado de la versión que se quiere editar. La otra manera es desde la pantalla de edición de herramientas, realizando click en el lápiz que se encuentra al lado de la versión de la herramienta. El usuario es redirigido a la pantalla de edición de versiones donde puede modificar el número y la fecha de la versión.
- **Borrar Herramienta:** Para borrar una herramienta se debe realizar click en el ícono de la papelera. El usuario es redirigido a la pantalla de confirmación y si realiza click en el botón *Borrar* entonces la herramienta es eliminada de manera permanente del sistema.
- **Ver Herramienta:** Realizando click en el ícono del ojo se pueden ver los detalles de una herramienta. En la misma se puede ver el nombre y las versiones de la misma.
- **Ver Versiones de Herramientas:** Estando en la pantalla de listado de versiones se puede realizar click en el ícono del ojo y se mostrarán los datos de la versión.

6.4. Resumen

En este capítulo se presentaron los requerimientos de la aplicación web, se presentó una descripción de los casos de uso del sistema. Además se mostró tanto la arquitectura como el diseño de la base de datos. Finalmente se detallaron las funcionalidades de la aplicación.

Capítulo 7

Conclusiones y trabajo a futuro

Como se señalara en el Capítulo 1, este proyecto surge por la necesidad de extender la metodología de evaluación de BPMS, que se desarrolla dentro del grupo COAL. Este capítulo presenta las conclusiones del proyecto, producto del trabajo realizado y expuesto en capítulos anteriores. Así mismo, introduce un conjunto de aspectos que deberían considerarse en trabajos a realizarse en el futuro.

7.1. Conclusiones

Para poder cumplir con los objetivos del proyecto el trabajo se dividió en tres etapas: investigación de metodologías de evaluación de software, evaluación de BPMS y elaboración de la aplicación web. A lo largo de estas diferentes etapas se encontraron aspectos destacados e inconvenientes.

En primer lugar, se realizó un estudio de metodologías de evaluación de software. Luego, se compararon dichas metodologías con la metodología del grupo COAL. Como resultado del análisis se obtuvo una lista de carencias y virtudes que se utilizó para mejorar la metodología existente. Además se realizó un relevamiento de requisitos no funcionales de todo tipo de software, seleccionando aquellos que se adecuen a herramientas de BPM. De este relevamiento surgió una lista de características no funcionales que fue incorporada a la lista de criterios de la metodología. También se agregó una forma de evaluar cada característica de forma objetiva, describiendo el procedimiento a seguir para responder las preguntas asociadas a las subcaracterísticas. Si bien en la metodología no se agregó un criterio de corte, el mismo fue utilizado en la aplicación web.

La metodología cuenta con un proceso de evaluación, luego del estudio de otras metodologías se decidió incorporar un subproceso que define los pasos a seguir para la ejecución del plan de pruebas.

En segunda instancia, tomando la lista de herramientas dadas y un subconjunto de características técnicas fue posible aplicar la metodología de evaluación actualizada. En esta etapa se presentaron varios inconvenientes, en principio uno de los BPMS seleccionados era Adonis, el mismo no provee un motor de procesos, por lo que no es posible evaluar gran parte de las características seleccionadas. Se optó por evaluar Aris, de la misma se obtuvo una versión *Student*, pero no se contaba con los requerimientos de hardware necesarios para ejecutar los procesos. Luego de destinar el tiempo de evaluación de una herramienta en Adonis y Aris, se acordó con los tutores evaluar la mayor cantidad de características de la versión *Express* de Aris. Se presentó otro inconveniente al momento de evaluar la performance de los BPMS ya que las herramientas *open source* que se habían investigado, al momento de utilizarlas, presentaban problemas de sesión o variables para alguno de los BPMS. En este sentido, se buscó una herramienta que permitiera evaluar la performance de la mayor cantidad de BPMS posible. A pesar de los inconvenientes se obtuvo un conjunto de datos, los cuales fueron utilizados como base para la implementación de la herramienta web de soporte a la metodología.

La selección de un BPMS no es una tarea sencilla, luego de evaluar nueve herramientas distintas es difícil recomendar una, sin embargo todos estamos de acuerdo que la versión evaluada de Orchestra no cumple con requerimientos mínimos. No es posible decir cuál es la mejor herramienta sin conocer las necesidades de la organización y los conocimientos de los usuarios finales. La mayoría de las herramientas delegan la seguridad de los documentos a terceros, debiendo incorporar componentes externos para soportar el requerimiento, la misma situación se repite con la utilización de firmas digitales y el envío de datos a través de HTTPS.

En última instancia, se desarrolló la aplicación web. La aplicación es una forma de visualizar los resultados de las evaluaciones de una forma amigable. Al mismo tiempo permite realizar diferentes evaluaciones, ponderando cada subcaracterística y seleccionando los umbrales. También es sencillo comparar resultados de dos herramientas diferentes o de dos versiones de la misma herramienta.

En resumen, se logró cumplir con los objetivos del proyecto. Se deja una metodología actualizada y de mejor calidad al introducir aspectos no funcionales que además se evalúan de forma objetiva. También se deja una aplicación web que permitirá reducir los tiempos de los cálculos de los resultados.

7.2. Trabajo futuro

En esta sección se describen algunas sugerencias de posibles trabajos a realizar. Algunas sugerencias están relacionadas a los resultados obtenidos, otras se relacionan a aspectos más generales.

A continuación se enumeran acciones generales y otras como conclusión del proyecto:

- Completar la metodología de evaluación agregando nueva características para herramientas en la *cloud*. Al evaluar este tipo de software no solo hay que tener en cuenta las características del propio software, sino también hay que tener en cuenta por ejemplo, la velocidad de conexión que tiene la organización.
- Incorporar características relacionadas a las aplicaciones mobile de los BPMS.
- Actualización y mantenimiento de la lista de características a evaluar. Los sistemas evolucionan, en este sentido, se generan nuevas funcionalidades que hasta ahora no se han tenido en cuenta al momento de actualizar la lista de características.
- Actualizar el método de evaluación de las características, buscando mayor objetividad y mejores resultados para los clientes que usen la aplicación web para elegir su BPMS. Al calcular el valor de las características técnicas funcionales, hay un factor que es determinado por el evaluador, por lo tanto, deja un matiz de subjetividad a la hora de evaluar este tipo características.
- Realizar la evaluación de toda la lista de características para las herramientas evaluadas y también incorporar nuevas herramientas.
- Incorporar nuevos estándares a la metodología, por ejemplo estándares de usabilidad web o de seguridad de aplicaciones.
- Incluir criterios de corte y poda a la metodología de evaluación.
- Agregar roles y capacidades que deberían tener los evaluadores de las herramientas para que el resultado obtenido sea el más objetivo.
- Incluir en la aplicación web organizaciones a las cuales pertenecen los usuarios.
- Relacionado con el punto anterior, poder incorporar en la aplicación web distintos resultados de la misma versión de una herramienta para diferentes organizaciones.

Bibliografía

- [1] Andrea Delgado, Daniel Calegari, Pablo Milanese, Renatta Falcon, and Esteban García. A systematic approach for evaluating bpm systems: case studies on open source and proprietary tools. In *Open Source Systems: Adoption and Impact*, pages 81–90. Springer, 2015.
- [2] Aphrodite Tsalgatidou and TYPAN Panepistimiopolis. Selection criteria for tools supporting business process transformation for electronic commerce. *Proceedings of EURO-MED NET*, 98:244–253, 1998.
- [3] David Morera. Cots evaluation using desmet methodology & analytic hierarchy process (ahp). In *Product Focused Software Process Improvement*, pages 485–493. Springer, 2002.
- [4] Patricia K Lawlis, Kathryn E Mark, Deborah A Thomas, and Terry Courtheyn. A formal process for evaluating cots software products. *Computer*, 34(5):58–63, 2001.
- [5] Davide Taibi, Luigi Lavazza, and Sandro Morasca. Openbqr: a framework for the assessment of oss. In *Open Source Development, Adoption and Innovation*, pages 173–186. Springer, 2007.
- [6] Feras Tarawneh, Fauziah Baharom, Jamaiah Hj Yahaya, and Faudziah Ahmad. Evaluation and selection cots software process: The state of the art. *International Journal of New Computer Architectures and their Applications (IJNCAA)*, 1(2):344–357, 2011.
- [7] Carolina Smith Adriana Acosta, Natalia Beloso. *Evaluación de Tecnologías de Procesos de Negocio centrado en motores XPDL*. PhD thesis, Proyecto de grado, Facultad de Ingeniería, Udelar, 2011.
- [8] Federico Parins Daniel Bonini. *Evaluación de Tecnologías de Procesos de Negocio centrado en motores XPDL*. PhD thesis, Proyecto de grado, Facultad de Ingeniería, Udelar, 2011.
- [9] Agustín Hernández Enrique Galindo. *Evaluación de características de Sistemas de Gestión de Procesos de Negocio para automatización de procesos en el Estado*. PhD thesis, Proyecto de grado, Facultad de Ingeniería, Udelar, 2012.
- [10] Mathias Weske. *Business process management: concepts, languages, architectures*. Springer Science & Business Media, 2012.

- [11] S.L. ITS Barcelona Consulting. ITS BARCELONA CONSULTING. <http://www.its-bcn.com/servicios/gesti%C3%B3n-de-procesos/>, 2016.
- [12] Andrea Delgado, Barbara Weber, Francisco Ruiz, Ignacio Garcia-Rodríguez de Guzmán, and Mario Piattini. An integrated approach based on execution measures for the continuous improvement of business processes realized by services. *Information and Software Technology*, 56(2):134–162, 2014.
- [13] Vinayak Khadye. BPM Series: Components of BPM Systems. <http://it.toolbox.com/blogs/bpm-blog/bpm-series-components-of-bpm-systems-4373>., 2016.
- [14] European Software Institute. European Software Institute website. <http://www.esi.es/>, 2016.
- [15] Mario Barbacci, Mark H Klein, Thomas A Longstaff, and Charles B Weinstock. Quality attributes. Technical report, DTIC Document, 1995.
- [16] ISO 25010. <http://iso25000.com/index.php/normas-iso-25000/iso-25010>, 2010. [Online; accessed 19-July-2008].
- [17] Inc. Red Hat. Red Hat JBoss BPM Suite 6.1 website. <https://www.redhat.com/es/resources/red-hat-jboss-bpm-suite-6>, 2016.
- [18] Inc. Bonitasoft. Bonitasoft website. <http://es.bonitasoft.com/>, 2016.
- [19] Inc. Intalio. Intaliolbpms website. <http://www.intalio.com/>, 2015.
- [20] activiti and contributors. Activiti website. <http://activiti.org/>, 2016.
- [21] bizagi web. Bizagi website. <http://www.bizagi.com/es/>, 2016.
- [22] Camunda services GmbH. Camunda website. <https://camunda.com/>, 2013.
- [23] SOFTWARE AG. BPMN process modeling & free modeling tool. <http://www.ariscommunity.com/aris-express/bpmn-2-free-process-modeling-tool>, Mayo 2015.
- [24] Orchestra website. <http://orchestra.ow2.org/xwiki/bin/view/Main/WebHome>, 2011.
- [25] ProcessMaker Inc. ProcessMaker website. <https://www.processmaker.com/>, 2016.
- [26] Inc. Red Hat. Red Hat JBoss BPM Suite 6.1 Arquitectura. <http://www.jboss.org/products/bpmsuite/get-started/#!project=bpmsuite>, 2016.
- [27] Bonitasoft connectors. <http://documentation.bonitasoft.com/search/site/connectors>, 2016.
- [28] Webinar BPMS SOA Advanced Techniques by Intalio. <http://support.intalio.com/deskpro/faq.php>, 2012.
- [29] Apache ODE. <http://ode.apache.org/extensions/xpath-extensions.html>, 2008.
- [30] FAQ. <https://youtu.be/oQB7OR0-TPk>, 2015.

- [31] activiti and contributors. Activiti Arquitectura. http://docs.alfresco.com/activiti/docs/admin-guide/1.3.2/#_high_level_architecture, 2016.
- [32] bizagi web. Arquitectura de Bizagi. <http://help.bizagi.com/bpmsuite/es/index.html?arquitectura.htm>, 2016.
- [33] camunda and contributors. Camunda Architecture. <https://docs.camunda.org/manual/7.3/guides/user-guide/>, 2013.
- [34] camunda and contributors. Camunda User Guide. <https://docs.camunda.org/manual/7.3>, May 2015.
- [35] SOFTWARE AG. Arquitectura de Aris. https://www.softwareag.com/corporate/rc/rc_perma.asp?id=tcm:16-111830, Mayo 2015.
- [36] 3C Soluciones Integrales. Technical Overview AGPLv3 Spanish. http://3csoluciones.com/documentos/Processmaker/Brochure_ProcessMaker.pdf, Diciembre 2015.
- [37] ProcessMaker Inc. ProcessMaker Community Edition v. 3.0. <http://wiki.processmaker.com/>, Mayo 2015.
- [38] ProcessMaker Inc. ProcessMaker Demonstration & Overview (Spanish). <http://www.processmaker.com/training/webinar-on-demand-3>, Mayo 2015.
- [39] activiti and contributors. Activiti User Guide. http://activiti.org/userguide/index.html?_ga=1.222962523.1505348422.1452718063#_importing_existing_models, 2016.
- [40] Rossella Aiello. Workflow performance evaluation. *University of Salerno, Italy*, page 158, 2004.
- [41] Jakob Nielsen. Usabilidad. diseño de páginas web. *DE INFORMACIÓN*., 2000.
- [42] Joram Barrez. The Activiti Performance Showdown 2015. <http://www.jorambarrez.be/blog/tag/performance/>, July 2015.
- [43] Red Hat JBoss BPM Suite 6.1 Administration And Configuration Guide. https://access.redhat.com/documentation/en-US/Red_Hat_JBoss_BPM_Suite/6.1/pdf/Administration_And_Configuration_Guide/Red_Hat_JBoss_BPM_Suite-6.1-Administration_And_Configuration_Guide-en-US.pdf, 2015.
- [44] Red Hat JBoss BPM Suite 6 Supported Configurations. <https://access.redhat.com/articles/704703>, 2015.
- [45] Red Hat JBoss BPM Suite 6 User Guide. https://access.redhat.com/documentation/en-US/Red_Hat_JBoss_BPM_Suite/6.0/pdf/User_Guide/Red_Hat_JBoss_BPM_Suite-6.0-User_Guide-en-US.pdf, 2015.
- [46] Red Hat JBoss BPM Suite 6 User Guide. https://access.redhat.com/documentation/en-US/Red_Hat_JBoss_BPM_Suite/6.0/pdf/User_Guide/Red_Hat_JBoss_BPM_Suite-6.0-User_Guide-en-US.pdf, 2015.
- [47] Walter Ovidio Sánchez et al. La usabilidad en ingeniería de software: definición y características. 2011.

- [48] Hugues M ISMAEL GHALIMI. Eclipse bpmn modeler-introducing intalio designer. http://www.eclipsecon.org/summiteurope2006/presentations/ESE2006-EclipseModelingSymposium1_BPMS%26Eclipse.pdf, 2006.
- [49] Intalio Web Service. http://support.intalio.com/bpms-version-7-5/#Supported_Web_Services, 2015.
- [50] Keith Ballinger, David Ehnebuske, Christopher Ferris, Martin Gudgin, Canyang Kevin Liu, Mark Nottingham, and Prasad Yendluri. Basic Profile Version 1.1. <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>, 2006.
- [51] Data Mapper. http://support.intalio.com/bpms-news/document/data_mapper/, 2015.
- [52] Evertteam. Modeling Timer Start Event. <http://support.evertteam.com/?p=2522>, 2015.
- [53] Decision Table as a Web Service Runtime Details. http://support.intalio.com/bpms-news/document/decision_table_as_a_web_service_runtime_details-2/, 2015.
- [54] Workflow Administration. <http://support.intalio.com/bpms-news/document/workflow/>, 2015.
- [55] Escalation. <http://support.intalio.com/bpms-news/document/escalation-2/>, 2015.
- [56] Task Attachment Service. <http://support.intalio.com/bpms-news/document/task-attachment-service/>, 2015.
- [57] Ad hoc Reporting. <http://support.intalio.com/bpms-news/document/ad-hoc-reporting/>, 2015.
- [58] Securing Server. http://support.intalio.com/bpms-news/document/intalio_server_using_https-2/, 2015.
- [59] Modeler Eclipse Kepler with Camunda plugin. <https://camunda.org/release/camunda-eclipse-plugin/prepackaged/kepler/camunda-modeler-kepler-latest.zip>, May 29, 2015.
- [60] camunda and contributors. Camunda Cycle and Rountrip. <https://docs.camunda.org/get-started/cycle/roundtrip-other-tools/>, May 2015.
- [61] Niall Deehan. Start and Complete a process with the REST API. <https://blog.camunda.org/post/2015/08/start-and-complete-process-with-rest-api/>, Aug 5, 2015.
- [62] Falko Menge. Business Rule Task - Demo Drools. <https://github.com/camunda/camunda-consulting/tree/master/one-time-examples/order-confirmation-rules>, Jul 2015.
- [63] ProcessMaker Inc. ProcessMaker BPM Software Mashups e Integraciones BPM con software de terceros. <http://www.processmaker.com/es/bpm-business-system-integration-videos>, Mayo 2015.
- [64] ProcessMaker Inc. ProcessMaker - SigPlus V 1.0. http://enterpriseedition.wiki.processmaker.com/index.php/ProcessMaker_-_SigPlus_V_1.0, Mayo 2015.

Anexo A

Características a evaluar

En este anexo se presenta la lista de criterios a partir del enfoque descrito en la sección 3.4 del capítulo 3.

En la sección A.1 se realiza una introducción al tema tratado en el capítulo, también se detalla la estructura de la definición de cada característica. La sección A.2 presenta la lista de criterios funcionales existentes. En la sección A.3 se brinda la definición de los criterios no funcionales siguiendo la estructura definida en la introducción. La sección A.4 define los criterios no técnicos de acuerdo a la estructura presentada en la introducción. Por último, en la sección A.5 se brinda un resumen del capítulo.

A.1. Introducción

Los criterios no funcionales son parte de una lista de características que se fueron definidos en proyectos anteriores. Al tratarse de una extensión, se mantuvo la estructura tanto de la agrupación de las características así como también la forma de la presentación de cada una de ellas, las cuales tiene la siguiente estructura:

Descripción: Se presentan aspectos generales de su funcionamiento, se especifica su significado y se definen conceptos involucrados.

Subcaracterísticas: Se propone una clasificación para las herramientas basada en esta característica o se listan funcionalidades estrechamente relacionadas con ésta, dependiendo del caso.

Referencias: Se incluyen las fuentes.

La estructura además cuenta con dos módulos, uno técnico y uno no técnico, definidos como sigue:

Técnico: Abarca todo lo referido al software en sí. Este módulo está dividido en dos grupos: requerimientos funcionales y no funcionales.

No Técnico: Engloba características deseables a nivel estratégico, por ejemplo, si la herramienta es comercial o no.

A.2. Definición de criterios funcionales existentes

En esta sección se presentan las categorías y características funcionales. Es decir, aquellas características que describen las funcionalidades deseadas en un BPMS. Abarcan todo lo referido al software en sí, como arquitectura, componentes, comportamiento, etc. Estos criterios están definidos en [1].

A.2.1. Tecnología, Arquitectura e Interoperabilidad

Esta categoría se enfoca en la plataforma tecnológica sobre la cual funciona la aplicación. Tiene en cuenta la arquitectura de la aplicación, los mecanismos de interoperabilidad con otros sistemas, por ejemplo, la integración de servicios web.

A.2.1.1. Compatibilidad con estándares para modelado de procesos de negocios

Descripción Permite importar/exportar procesos en formatos estándar como: XPD, BPEL y BPMN.

Subcaracterísticas

No aplica

Referencias Característica A.3.1.1 en [1]

A.2.1.2. Base de Datos

Descripción Permite elegir la plataforma DBMS a utilizar, algunas de las posibilidades son: IBM DB2, Microsoft SQL Server, MySQL, Oracle y PostgreSQL.

Subcaracterísticas

- Brinda la posibilidad de que los datos sean persistidos en un único repositorio.

Referencias Característica A.3.1.2 en [1]

A.2.1.3. Arquitectura de la aplicación

Descripción La arquitectura de la aplicación permite la distribución del procesamiento en al menos 3 capas (de datos, de negocio y de presentación). La capa de presentación se basa en un navegador web (excepto para el modelado, la simulación y demás herramientas de diseño y optimización).

Subcaracterísticas

No aplica

Referencias Característica A.3.1.3 en [1]

A.2.1.4. Soporta procesos configurables almacenados en la nube de acuerdo al paradigma SaaS

Descripción Es posible definir procesos con una estructura común que pueden ser configurados y almacenados en la nube que son ofrecidos en formato SaaS. Esto permite que distintas

organizaciones puedan compartir un modelo de proceso, adaptándolo de acuerdo a sus particularidades.

Subcaracterísticas

No aplica

Referencias Característica A.3.1.4 en [1]

A.2.1.5. Arquitectura del BPMS

Descripción Descripción de cómo se estructura el BPMS. Teniendo en cuenta los componentes tecnológicos que incluye. Por ejemplo, ciertas herramientas trabajan con componentes obligatorios para su funcionamiento, lo cual a su vez ofrece otra serie de servicios que potencian el sistema.

Subcaracterísticas

No aplica

Referencias Característica A.3.1.5 en [1]

A.2.1.6. Sistema operativo del modelador

Descripción Utilización del modelador de procesos en sistemas operativos Unix o Windows.

Subcaracterísticas

No aplica

Referencias Característica A.3.1.6 en [1]

A.2.1.7. Navegador web

Descripción Posee compatibilidad con los navegadores Microsoft Internet Explorer, Mozilla Firefox y Google Chrome.

Subcaracterísticas

No aplica

Referencias Característica A.3.1.7 en [1]

A.2.1.8. Extensibilidad

Descripción Incluye o permite integrar bibliotecas de objetos, Interfaces de programación de aplicaciones, APIs (Application Programming Interface, API) u otras herramientas para potenciales desarrollos, además de la documentación de las mismas.

Subcaracterísticas

No aplica

Referencias Característica A.3.1.8 en [1]

A.2.1.9. Integración con portales

Descripción Permite operar con portales preexistentes, tales como IBM WebSphere Portal, Microsoft SharePoint, mySAP y Oracle Portal.

Subcaracterísticas

- Integración con portales permitiendo la búsqueda conjunta de contenidos internos y externos al BPMS.

Referencias Característica A.3.1.9 en [1]

A.2.1.10. Integración con servicios web

Descripción Provee mecanismos para ofrecer servicios web que permitan la integración con otros sistemas heredados así como consumir servicios de los mismos.

Subcaracterísticas

- Soporta estándares de servicios web tales como SOAP, UDDI, WSDL, etc.
- Soporta la integración de información desde distintos servicios web.

Referencias Característica A.3.1.10 en [1]

A.2.1.11. Debugger

Descripción Permite al usuario realizar un seguimiento de la ejecución paso a paso, de manera tal que se pueda examinar el código a medida que se ejecuta y detenerse cuando se cumplen determinadas condiciones especificadas por el usuario.

Subcaracterísticas

No aplica

Referencias Característica A.3.1.11 en [1]

A.2.1.12. Integración con redes sociales

Descripción Permite consumir las Interfaces de programación de aplicaciones o APIs (Application Programming Interface) provistas por las redes sociales, específicamente Facebook, Google+ y Twitter.

Subcaracterísticas

No aplica

Referencias Característica A.3.1.12 en [1]

A.2.1.13. Soporte de tecnologías en capa de presentación

Descripción Permite la utilización de tecnologías en la capa de presentación como: AJAX, WPF, Silverlight.

Subcaracterísticas

No aplica

Referencias Característica A.3.1.13 en [1]

A.2.1.14. Definir cantidad de usuarios

Descripción Permite la definición de un rango aproximado de usuarios que utilizan la herramienta, de manera tal que esta pueda variar los servicios que ofrece dependiendo del rango.

Subcaracterísticas

No aplica

Referencias Característica A.3.1.14 en [1]

A.2.1.15. Plataformas de servidores

Descripción Soporta plataforma de servidores tales como: IBM iSeries (AS/400), Linux Mainframe, Microsoft server, Unix, Tomcat.

Subcaracterísticas

No aplica

Referencias Característica A.3.1.15 en [1]

A.2.2. Diseño y modelado de procesos

El modelado de procesos de negocio es una etapa clave en el ciclo de vida de la aplicación. En general se espera que el diseño de alto nivel pueda ser llevado a cabo por parte de los *key users* (que no necesariamente tienen formación técnica). En esta Sección se incluyen características relativas a los componentes que permiten modelar los procesos de negocio y entre otros requerimientos se destacan la usabilidad, funcionalidades que permiten el trabajo colaborativo así como la utilización de tipos de procesos ya definidos con el objetivo de reducir el tiempo de diseño.

A.2.2.1. Diseño gráfico para *key users*

Descripción Permite el diseño de los modelos de procesos sin necesidad de programación.

El diseño incluye el desarrollo, visualización, modificación de los diagramas de flujo y de los atributos de cada elemento del modelo.

Subcaracterísticas

No aplica

Referencias Característica A.3.2.1 en [1]

A.2.2.2. Interfaz de diseño intuitiva y centrada en la usabilidad

Descripción Se refiere a la interfaz del modelador de procesos.

Subcaracterísticas

- *Drag-and-Drop* de los componentes en el *workflow*.
- Optimización del gráfico (alineación automática y corrección de curvas).
- Paleta con los componentes que pueden ser incluidos en el diagrama. Además puede incluir actividades o eventos estándar (por ejemplo, enviar un correo o consumir un servicio web).
- Copiar y pegar uno o varios componentes de un proceso en otro.
- Ampliar zonas del mapa de un proceso.
- Ayuda contextual.

Referencias Característica A.3.2.2 en [1]

A.2.2.3. Tipo de modelador

Descripción Modelador de procesos web o de escritorio.

Subcaracterísticas

No aplica

Referencias Característica A.3.2.3 en [1]

A.2.2.4. Definición de tipos de procesos

Descripción Permite la definición de tipos de procesos. Por ejemplo, definir un tipo de proceso llamado “inscripción a materia” e “inscripción a materia con horario” un proceso de dicho tipo. Podrían haber 10 procesos distintos del tipo “inscripción a materia” para los que se quiere generar un reporte con toda la información agregada.

Subcaracterísticas

- Obtener reportes con la información agregada para un tipo de proceso.
- Definir una estructura básica para los procesos de este tipo, que acelere el modelado de los mismos.

Referencias Característica A.3.2.4 en [1]

A.2.2.5. Notación gráfica de procesos compatible con BPMN 2.0

Descripción Notación gráfica de procesos compatible con BPMN 2.0

Subcaracterísticas

No aplica

Referencias Característica A.3.2.5 en [1]

A.2.2.6. Listado de versiones de modelos de procesos

Descripción Permite al usuario visualizar el listado de versiones de un proceso. Desde el listado se puede acceder directamente a la especificación del proceso.

Subcaracterísticas

No aplica

Referencias Característica A.3.2.6 en [1]

A.2.2.7. Procedimientos de validación automática de modelos de procesos

Descripción Procedimientos de validación de la consistencia interna de los procesos.

Subcaracterísticas

- Detección de *loops* infinitos.
- Detección de estados sin condiciones de salida.
- Detección de condiciones de salida no excluyente para un mismo estado.
- Detección de referencias circulares.

Referencias Característica A.3.2.7 en [1]

A.2.2.8. Descubrimiento automático de procesos de negocio

Descripción Mediante el uso de process mining, permite que a partir de datos históricos (puede ser el histórico del BPMS o log de cualquier otro sistema legado) la aplicación genere un diagrama del proceso y lo represente gráficamente.

Subcaracterísticas

No aplica

Referencias Característica A.3.2.8 en [1]

A.2.2.9. Trabajo colaborativo sobre procesos

Descripción Permite que los usuarios encargados de diseñar los procesos puedan realizar determinadas tareas. Además un administrador puede proteger un proceso para impedir que le realicen cambios.

Subcaracterísticas

- Ver un listado de los procesos disponibles y sus versiones.
- Posibilidad de descargar los procesos disponibles.
- Puedan hacer check-in y check-out sobre los mismos.
- Agregar comentarios para una versión en cada check-in.
- Elegir si un check-in genera una versión mayor o menor.

Referencias Característica A.3.2.9 en [1]

A.2.2.10. Información sobre procesos

Descripción La metadata de cada proceso incluye, al menos, información sobre el autor del proceso, el tipo de proceso, roles de usuarios autorizados a visualizar el diagrama de flujo y a modificarlo.

Subcaracterísticas

No aplica

Referencias Característica A.3.2.10 en [1]

A.2.2.11. Notificaciones a autores de procesos

Descripción Notifica vía email a los dueños del proceso cada vez que una nueva versión de un proceso se pone en producción (recordando, por ejemplo, que deben actualizar la información de ayuda).

Subcaracterísticas

No aplica

Referencias Característica A.3.2.11 en [1]

A.2.2.12. Plantillas de procesos

Descripción La herramienta cuenta con procesos predefinidos, que pueden utilizarse como base y/o ejemplo para la creación de nuevos procesos.

Subcaracterísticas

No aplica

Referencias Característica A.3.2.12 en [1]

A.2.3. Gestión de formularios

La gestión de formularios refiere a la capacidad de diseñar las interfaces gráficas a las que van a acceder los usuarios finales a través del portal de la aplicación o a través de aplicaciones nativas para dispositivos móviles (en caso que la herramienta lo soporte). Es importante que los formularios sean altamente configurables y puedan ser diseñados por usuarios sin conocimiento técnico.

A.2.3.1. Definición de formularios

Descripción Herramienta de definición y modificación de formularios con recursos gráficos.

Como mínimo, drag-and-drop y reglas de llenado (disparo asociado a tareas y usuarios), sin que se requiera participación de un programador.

Subcaracterísticas

- Permite formularios compartidos.
- Permite formularios multipágina.

Referencias Característica A.3.3.1 en [1]

A.2.3.2. Formato de formularios

Descripción Incluye plantillas para cada tipo de formulario y permite configurar distintos elementos, en particular: fuente, estilo (**bold**, *italic* y underline), color y alineación para cada campo. Los campos tienen tamaño dinámico de acuerdo al tamaño del dato a completar.

Subcaracterísticas

No aplica

Referencias Característica A.3.3.12 en [1]

A.2.3.3. Soporte para dispositivos móviles

Descripción Permite definir formularios específicos para usuarios que accedan al portal a través de un dispositivo móvil.

Subcaracterísticas

- Permite incluir formularios nativos para algún dispositivo en particular.
- A partir de formularios se generan formularios nativos para algún dispositivo.
- Permite integrar funcionalidades nativas del dispositivo, por ejemplo, la cámara, el GPS.

Referencias Característica A.3.3.3 en [1]

A.2.3.4. Elementos del formulario

Descripción Posibilidad de incluir listas desplegables cuyo conjunto de valores se guarde en una tabla. Permite definir botones que disparan acciones predefinidas, a su vez los mismos pueden estar habilitados/deshabilitados y ocultos.

Subcaracterísticas

No aplica

Referencias Característica A.3.3.4 en [1]

A.2.3.5. Validación de campos del formulario

Descripción Debe permitir como mínimo verificar la validez del dato de acuerdo a su tipo, definir campos requeridos y notificar al usuario de los errores que se producen.

Subcaracterísticas

No aplica

Referencias Característica A.3.3.5 en [1]

A.2.3.6. Formularios dinámicos

Descripción Permite que las opciones de una lista, la protección de una acción o incluso ocultar/mostrar un campo dependan del valor de algún campo del formulario. Por ejemplo, si hay campos “Departamento” y “Localidad”, al seleccionar el departamento “Cerro Largo”, la lista desplegable “Localidad” contendrá únicamente localidades de Cerro Largo.

Subcaracterísticas

No aplica

Referencias Característica A.3.3.6 en [1]

A.2.3.7. Reglas de llenado de campos

Descripción Permite definir reglas para la actualización automática de campos a partir de la carga de un campo clave.

Subcaracterísticas

- Mediante acceso a la base de datos (por ejemplo al ingresar la cédula de identidad de un ciudadano se cargan los demás campos que correspondan al misma).
- Acceso a aplicaciones externas.
- Cálculos algorítmicos de acuerdo a alguna fórmula.
- Campos completados en otros formularios.
- Información de la ejecución del proceso. Por ejemplo, la fecha en la que se inició el proceso o cuando fue aprobado por el gerente.

Referencias Característica A.3.3.7 en [1]

A.2.3.8. Restricciones de acuerdo al rol

Descripción Permite definir reglas para visualizar o modificar campos de acuerdo al rol del usuario.

Subcaracterísticas

No Aplica

Referencias Característica A.3.3.8 en [1]

A.2.4. Motor de *Workflow*

El motor de workflow es sin lugar a dudas el núcleo del sistema. Permite ejecutar los procesos definidos en la etapa de diseño y modelado, y a su vez produce la información necesaria para generar reportes y estadísticas sobre su funcionamiento a través del módulo de monitoreo. En esta Sección se incluyen las funcionalidades necesarias para el trabajo colaborativo entre los usuarios finales, así como aquellas que posibilitan gestionar los recursos y la asignación de tareas en forma eficiente, entre las que se encuentran mecanismos de notificaciones y escalamiento. En general interesa poder definir reglas para la resolución automática de situaciones habituales pero contar con la posibilidad de poder modificar el flujo o la asignación de tareas en forma manual (y en general a través de un usuario administrador) para resolver situaciones excepcionales.

A.2.4.1. Soporte de patrones de workflow

Descripción Permite implementar patrones de flujo.

Subcaracterísticas

- Sequence
- Parallel Split
- Synchronization
- Exclusive Choice
- Simple Merge
- Multi-Choice
- Structured Synchronizing Merge
- Multi-Merge
- Generalised AND-Join
- Deferred Choice
- Cancel Task
- Arbitrary Cycles
- Structured Loop
- Implicit Termination
- Thread Split (opcional)

Referencias Característica A.3.4.1 en [1]

A.2.4.2. Iniciación de procesos en forma manual y automática

Descripción Iniciación de procesos en forma manual, en forma automática de acuerdo a agenda y también automáticamente por un servicio web.

Subcaracterísticas

- Permite la creación de instancias de proceso basado en datos externos o eventos.

- Asignación de identificador único a las instancias de los procesos iniciados de forma automática.
- Chequeo de privilegios en caso de ser manual.

Referencias Característica A.3.4.2 en [1]

A.2.4.3. Opciones de escalamiento

Descripción Permite definir de qué manera van a escalar los eventos.

Subcaracterísticas

- Notificar usuarios con un rol determinado.
- Notificar a usuarios específicos.
- Mecanismo de escalamiento personalizado.

Referencias Característica A.3.4.3 en [1]

A.2.4.4. Mecanismos de notificación

Descripción Permite configurar la forma en la que se realizan las notificaciones, por ejemplo, por email, a la bandeja de entrada del usuario en la aplicación, cuadros de diálogo, etc., conforme transcurre el proceso. Debe ser programable mediante el uso de una herramienta ad-hoc (objetos de notificación incluidos en la herramienta de diseño). Otros mecanismos de notificación posibles son: Email, consultas SQL, llamadas a servicios web, llamadas a determinadas APIs, hacer *push* a un mensaje en una cola de mensajes, notificaciones al teléfono móvil.

Subcaracterísticas

- Notificaciones integradas con aplicaciones de escritorio, por ejemplo, Windows Mail.
- Notificaciones integradas en el escritorio.
- Notificaciones integradas con el Manejador de Base de Datos.
- Envío de notificaciones en forma concurrente.
- Se definen notificaciones para un determinado rango de tiempo.
- Posibilidad de que el usuario habilite o deshabilite las notificaciones.
- Las notificaciones incluyen hipervínculos a contenidos.
- Los mensajes asociados a una notificación pueden usar parámetros del proceso definido por el usuario.

Referencias Característica A.3.4.4 en [1]

A.2.4.5. Definición de reglas de negocio

Descripción Brinda la posibilidad de definir reglas de negocio para modificar el estado del flujo, en forma manual y automática (espera, suspensión, cancelación, eliminación, reactivación y disparo de acciones).

Subcaracterísticas

- Reglas en scripts embebidos.
- Reglas de negocio basadas en las variables del proceso.
- Reglas de negocio basadas en la información de la ejecución del proceso.
- Reglas de negocio basadas en plazos.
- Se pueden combinar todas las subcaracterísticas anteriores

Referencias Característica A.3.4.5 en [1]

A.2.4.6. Definición de controles de negocio

Descripción Es posible definir controles de negocio como pueden ser el control de los recursos o control temporal de las tareas.

Subcaracterísticas

No Aplica

Referencias Característica A.3.4.6 en [1]

A.2.4.7. Modificar el flujo de trabajo

Descripción Permite modificar el flujo de proceso.

Subcaracterísticas

- El flujo de trabajo puede ser modificado manualmente.
- El flujo de trabajo puede ser modificado automáticamente basado en el tiempo.
- El flujo de trabajo puede ser modificado automáticamente basado en datos externos y/o eventos.

Referencias Característica A.3.4.7 en [1]

A.2.4.8. Asignación de tareas

Descripción Es posible asignar una tarea a un usuario, grupo de usuarios o a un rol determinado. En particular, debe ser posible asignar una tarea a un grupo de usuarios y que, cuando la tarea sea tomada por uno de los usuarios, se desasigne automáticamente al resto de los usuarios del grupo.

Subcaracterísticas

- La asignación de tareas se realiza en forma manual y/o automática según criterios definidos.
- Si la asignación es automática, se puede realizar la asignación en función de variables y estados.
- Existen distintos algoritmos predefinidos para la asignación de tareas (balanceo de carga, round-robin, basados en el rol que tienen, etc.).

Referencias Característica A.3.4.8 en [1]

A.2.4.9. Reasignación de tareas

Descripción Permite la reasignación manual y/o automática de actividades y tareas, a un supervisor o a otro usuario, de acuerdo a reglas de negocio definidas. Pueden ser tareas puntuales o todas las de un usuario.

Subcaracterísticas

No aplica

Referencias Característica A.3.4.9 en [1]

A.2.4.10. Gestión de colas de actividades y tareas

Descripción Permite el ABM (alta, baja y modificación) en colas de actividades y tareas, tanto por usuario como por rol, basado en el balanceo simple de carga o balanceo basado en la eficiencia individual de los usuarios.

Subcaracterísticas

- Permite definir notificaciones y triggers específicos para cada cola.
- Permite definir prioridad para cada tarea y que las mismas sean resueltas de acuerdo a la misma en vez de First In First Out.
- Permite que para cada cola se pueda definir una política de aging (por ejemplo, cada tarea que pasa un día sin ser resuelta aumenta su prioridad).

Referencias Característica A.3.4.10 en [1]

A.2.4.11. Modelado de datos

Descripción Es posible definir tipo de datos y objetos a manejar por el flujo. Deberá admitir, al menos, los tipos de dato text, numeric, date, large text y document.

Subcaracterísticas

No aplica

Referencias Característica A.3.4.11 en [1]

A.2.4.12. Adjuntar documentos a tareas

Descripción Permite adjuntar a una tarea uno o más documentos, con reglas de negocio para el tamaño de los archivos y el tipo de archivo.

Subcaracterísticas

- El adjunto sea sólo para la instancia de proceso a la que se agrega.
- El adjunto es compartido para todas las instancias de proceso.

Referencias Característica A.3.4.12 en [1]

A.2.4.13. Ligado de variables

Descripción El ligado de variables del proceso se puede definir únicamente en forma estática o dinámica. Se pueden ligar variables de entrada del usuario, de bases de dato externas, de servicios web o llamadas a algún procedimiento externo.

Subcaracterísticas

No Aplica

Referencias Característica A.3.4.13 en [1]

A.2.4.14. Vinculación entre tarea y documento

Descripción Posibilidad de vincular una tarea a un documento plantilla, por ejemplo, para envío por email o correo de cartas predefinidas, certificaciones, comprobantes y facturas.

Subcaracterísticas

No Aplica

Referencias Característica A.3.4.14 en [1]

A.2.4.15. Registro del organigrama y escalamiento de responsabilidades

Descripción Provee funcionalidad de registro del organigrama en una jerarquía multinivel. Además, los vínculos entre los distintos niveles deben estar disponibles al momento de configurar el escalamiento de responsabilidades sobre una tarea.

Subcaracterísticas

No Aplica

Referencias Característica A.3.4.15 en [1]

A.2.4.16. Versionado de proceso

Descripción Brinda la posibilidad de que existan varias versiones de un mismo proceso. Los cambios en una versión no afectan los procesos ya instanciados, en particular, distintas versiones de un proceso pueden estar ejecutando al mismo tiempo. Además, debe permitir la migración de procesos a la versión actual en caso de que los cambios sean posteriores al paso en ejecución.

Subcaracterísticas

- Posibilidad de reactivar una versión anterior como la versión actual.

Referencias Característica A.3.4.16 en [1]

A.2.4.17. Integración con motores de *workflow* de terceros

Descripción Brinda la posibilidad de integrarse con motores de *workflow* externos.

Subcaracterísticas

No Aplica

Referencias Característica A.3.4.17 en [1]

A.2.4.18. Calendario del sistema completo y configurable

Descripción El calendario del sistema distingue los días de trabajo, vacaciones y los horarios de inicio y fin de la oficina.

Subcaracterísticas

- Herramienta para la gestión de licencias y suplencias: herencia de tareas (no es posible heredar todo: puede haber permisos con los que cuente el titular pero no el suplente).

Referencias Característica A.3.4.18 en [1]

A.2.4.19. Tratamiento de excepciones

Descripción Posibilidad de intervenir o tratar excepciones (debidas a errores de modelado) en forma manual, durante la ejecución, con el fin de introducir ajustes o actualizar datos por parte de usuarios clave.

Subcaracterísticas

No Aplica

Referencias Característica A.3.4.19 en [1]

A.2.4.20. Deshacer acción sobre una tarea

Descripción Posibilidad de que un usuario supervisor pueda deshacer una tarea realizada por un usuario, en caso de que este haya cometido algún error. Se debe permitir retomar el trámite desde la tarea anterior a aquella en la que se produjo el error.

Subcaracterísticas

- Permitir que lo haga el usuario que cometió el error, pero supervisado por el supervisor.
- Exigir que se necesite autorización de más de un supervisor.

Referencias Característica A.3.4.20 en [1]

A.2.4.21. Tareas de borrador

Descripción Posibilidad de que una tarea pueda ser guardada temporalmente en carácter de borrador para retomarla luego sin perder la información ingresada.

Subcaracterísticas

No Aplica

Referencias Característica A.3.4.21 en [1]

A.2.4.22. Escalabilidad del motor

Descripción Deben estar especificados ciertos aspectos.

Subcaracterísticas

- Cantidad de usuarios soportados.
- Cantidad de usuarios soportados de forma concurrente.
- Cantidad de procesos soportados en paralelo.

Referencias Característica A.3.4.22 en [1]

A.2.4.23. Monitoreo del contenido

Descripción Para cada objeto (documento, imagen, email, etc.) contenido en el repositorio es posible acceder a sus propiedades, métodos y eventos. Permite disparar acciones

ante eventos asociados a un objeto, como puede ser la modificación de un archivo o la presencia de un nuevo documento en una carpeta determinada del repositorio. Los métodos incluyen copiar un archivo a una carpeta, crear un enlace a un archivo, borrarlo, etc. y pueden ser invocados como actividades predefinidas.

Subcaracterísticas

- Es posible configurar con un *wizard* sin necesidad de código.

Referencias Característica A.3.4.23 en [1]

A.2.4.24. Case Management dinámico

Descripción Permite cierta flexibilidad en los procesos y modificaciones a su estructura en tiempo de ejecución (no de diseño) siempre y cuando el usuario tenga los permisos adecuados.

Subcaracterísticas

- Permite incluir nuevos procesos en algún punto del flujo; en ese caso el histórico de los mismos va a estar incluido en el proceso del que derivan.
- Permite adjuntar de forma dinámica nuevos objetos al proceso (documentos, atributos, recursos).
- Permite saltar actividades.
- Permite volver a realizar una actividad.
- Permite modificar el orden en que se realizan las actividades.

Referencias Característica A.3.4.24 en [1]

A.2.5. Administración de la seguridad

Esta categoría agrupa funcionalidades asociadas a la seguridad del sistema. Es de suma importancia la posibilidad de definir roles y perfiles de acceso para los usuarios, ya que el sistema va a ser utilizado por distintos tipos de usuario con diferentes posiciones jerárquicas dentro de la estructura de la organización definida, por lo que se necesita un acceso diferenciado. En particular, se destacan los mecanismos de autenticación de los usuarios, la gestión de los mismos y la granularidad de los permisos que poseen.

A.2.5.1. Definición de roles

Descripción Permite la definición de roles, estos roles pueden ser globales al sistema o específicos de cada proceso. A su vez, cada rol define el nivel de acceso que tienen los usuarios que poseen dicho rol. Además se deben cumplir las siguientes características:

- Cada usuario puede poseer uno o más roles.
- La asignación puede ser manual por medio de un administrador general.

Subcaracterísticas

- Permite la definición de roles dentro del repositorio.
- Permite la eliminación temporal de un rol, es decir, su asignación.

- El acceso se restringe con procesos de autenticación.
- Permite definir jerarquías de roles con granularidad ilimitada. Ejemplo: se definen los roles técnico, técnico forestal y técnico ganadero los cuales heredan las propiedades de técnico. Si a una tarea se le asigna el rol técnico, tanto técnico forestal como técnico ganadero pueden realizarla.

A.2.5.2. Definición de permisos sobre objetos

Descripción Permite la definición por parte de los usuarios administradores o clave de los permisos de cada usuario sobre los objetos, documentos, modelos, formularios y pantallas.

Subcaracterísticas

- Se valorará si el formato utilizado para la distribución de permisos se asemeja al formato ACL (Access Control List).

Se valorará si el formato utilizado para la distribución de permisos se asemeja al formato ACL (Access Control List).

A.2.5.3. Mecanismos de permisos sobre usuarios

Descripción Se proveen perfiles de acceso basados en permisos cuya granularidad debe llegar hasta el nivel de proceso, documentos adjuntos o de campos de formulario. Además, se deberán tener las siguientes características:

- El control lo realizará un módulo de la herramienta.
- Bloqueo de consultas y actualizaciones según perfil.
- Control automatizado y parametrizable de generación y cambio de contraseñas.

Subcaracterísticas

- Permite realizar un listado de todos los empleados según su perfil.
- Posibilidad de definir permisos basados en rangos de valores de determinadas variables (por ejemplo, montos y horarios).

A.2.5.4. Seguridad en dispositivos móviles

Descripción La seguridad es un aspecto crítico para las aplicaciones nativas en dispositivos móviles. Se requiere que como mínimo, la información que se almacena en el dispositivo esté cifrada y todo intercambio de información con el servidor sea sobre HTTPS.

A.2.5.5. Importar/exportar estructura organizacional

Descripción La herramienta cuenta con la capacidad de importar la estructura organizacional desde otros sistemas y/o exportarla. Subcaracterísticas Estructura organizacional refiere a la definición de usuarios y roles que participan en los procesos.

- Definición de los formatos en que se realiza la importación/exportación.
- Compatibilidad con otras herramientas.

A.2.6. Gestión, monitoreo y auditoría

Esta categoría agrupa funcionalidades asociadas a la gestión de los procesos, su monitorización y la posibilidad de realizar auditorías que permitan evaluar el rendimiento del sistema. Algunas de las características importantes son la obtención de reportes avanzados que en conjunto con la herramienta de optimización permitan detectar cuellos de botella e identificar puntos de mejora valiosos, ya que surgen del análisis de datos reales.

A.2.6.1. Gestión de instancias por usuarios

Descripción Permite el manejo de instancias por parte de los usuarios.

Subcaracterísticas

- Crear instancias de proceso.
- Pausar o reanudar instancias de proceso.
- Reinstalar instancias de proceso.
- Borrar instancias de proceso.
- Abortar instancias de proceso.
- Sobrescribir manualmente datos de las instancias de proceso. actividades.

Referencias Característica A.3.6.1 en [1]

A.2.6.2. Mantenimiento histórico del proceso

Descripción Se tiene un mantenimiento completo del histórico de cada proceso durante la vida activa del mismo, permitiendo visualizar de forma simple y cronológica todas las actividades, tareas realizadas, comentarios y documentos adjuntos (accediendo mediante vínculos).

Subcaracterísticas

- Visor de workflow: Posibilidad de que el usuario final pueda seguir en cualquier momento la historia de un proceso gráficamente, con información en texto de las tareas ejecutadas (quién ejecutó qué y cuándo), así como la situación actual y los asuntos pendientes de resolución.

Referencias Característica A.3.6.2 en [1]

A.2.6.3. Realización de reportes

Descripción Se proveen reportes y consultas sobre: volumen, tiempos empleados para la ejecución de cada tarea (tiempos medios, desviaciones, tiempos máximos y mínimos), estado de procesos, rendimiento de usuarios, oficinas y unidades ejecutoras, con la posibilidad de presentar datos en forma agregada por actividad o proceso, por nivel jerárquico, por unidad ejecutora y distribución horaria, diaria, mensual y anual de la demanda de cada proceso. Y tendrán las siguientes características:

- Disponibilidad de filtros que permitan configurar el contenido y alcance del reporte o consulta: elección del período, proceso y tarea, gráficos de evolución, navegación en BD (vínculos que permitan obtener mayor detalle).
- Posibilidad de definir plantillas para la consulta de históricos.
- Posibilidad de enviar los reportes a pantalla con opción de impresión o guardarlos como archivo.
- Exportación a Excel de listados (resultados de consultas y reportes).

Subcaracterísticas

- Generación de reportes de carga de trabajo de los usuarios.
- Generación de reportes de estado según: tareas completadas, pendientes, vencidas.
- Generación de estadísticas sobre:
 - El tiempo de ciclo de la instancia.
 - Las rutas que se utilizaron en los flujos de trabajo.
 - Los pasos que se utilizaron en los flujos de trabajo.
 - Los usuarios que realizaron los flujos de trabajo.
 - Los roles que se utilizaron en los flujos de trabajo.
 - El número de revisitas sobre la misma tarea.
- Generación de tendencias:
 - Volumen de actividades.
 - Tasa de cierre de tareas.
 - Volumen de retraso.
 - Duración promedio de atrasos.
 - Duración promedio de tareas.
- Permite el acceso a las tablas del BPMS, que contienen datos de información de proceso, que son los utilizados para la generación de los reportes. Permitiendo de esta manera realizar consultas específicas.

Referencias Característica A.3.6.3 en [1]

A.2.6.4. Monitoreo de procesos

Descripción Se proveen herramientas para el monitoreo de los procesos y los usuarios que participan, además se generan indicadores y estadísticas comparativas (entre usuarios y de usuarios respecto a los indicadores), incluyendo monitoreo en tiempo real, para la identificación de atrasos y cuellos de botella, a través de la interfaz gráfica de los supervisores.

Subcaracterísticas

- Permite la personalización de criterios de monitoreo incluyendo indicadores clave.

- Cuenta con un sistema de notificación de alarmas sobre posibles problemas o desvíos en la ejecución de procesos.
- Comparación de los datos surgidos de la ejecución con parámetros de calidad de servicio, generando alarmas o reportes, según se establezca.
- Almacenamiento de información del proceso, incluyendo: proceso. actividades.
 - Cambio de estado de cada actividad con la fecha y hora correspondiente.
 - Usuario que ha trabajado cada una de las actividades
 - Información relevante del proceso (información procedente de los formularios)

Referencias Característica A.3.6.4 en [1]

A.2.6.5. Definición y generación de Indicadores Claves de Rendimiento

Descripción Los Indicadores Claves de Rendimiento (*Key-Performance Indicators*, KPI) pueden ser definidos para medir las características de instancias de proceso, procesos, recursos o de las operaciones generales de la empresa. Ejemplos de interés pueden ser la cantidad de ejecuciones exitosas de los procesos de negocio en comparación con la cantidad total de ejecuciones de los procesos, la cantidad total de trabajo humano necesario o la cantidad de recursos necesarios para un proceso. Además provee recursos para la generación de estadísticas, informes, gráficas y tablas de indicadores, con recurso *drill-down*, sin necesidad de programación. También, tiene un panel de control que permite al usuario o administrador generar vistas, seleccionando los indicadores a mostrar, los cuales estarán basados en los parámetros de los procesos y en datos obtenidos de bases externas.

Subcaracterísticas

Posibilidad de visualizar y editar la tabla de indicadores con la siguiente información, de modo de permitir el monitoreo y la obtención de los indicadores:

- Descripción del objeto asociado al indicador.
- Nombre del indicador.
- Fórmula de cálculo.
- Valor objetivo.
- Frecuencia de cálculo.
- Propietario y unidad de medida

Referencias Característica A.3.6.5 en [1]

A.2.6.6. Mecanismos de simulación y optimización

Descripción Provee un módulo de simulación, la herramienta ofrece opciones de simulación de procesos que permite mejorarlos, efectuar análisis de uso de los recursos, cuellos de botella, demoras, reprocesamiento, sobrecargas, utilizando para ello datos históricos.

Además se dispondrá de una herramienta que permita inyectar datos de prueba al modelo, no necesariamente históricos. Algunas de las posibles optimizaciones son:

- Reconfiguración automática de flujo de trabajo basado en datos históricos.
- Análisis predictivo de los resultados de flujo de trabajo.
- Ajuste automático de la asignación de los participantes para eliminar cuellos de botella.
- Análisis de flujo de trabajo de la ruta crítica del proceso.

Subcaracterísticas

No Aplica

Referencias Característica A.3.6.6 en [1]

A.2.6.7. Mecanismos de auditorías

Descripción Provee reglas para el registro de información histórica que permiten realizar auditorías sobre las acciones ejecutadas por un usuario (como mínimo inicio y fin de sesiones, actualización de campos, *check-in* y *check-out* de documentos), registrando nombre, fecha, hora y tipo de acción, dirección IP, dirección MAC. A su vez, el histórico no puede ser modificado por ningún usuario del BPMS.

Subcaracterísticas

- Permite la personalización de información a incluir en los registros de información histórica de auditoría.

Referencias Característica A.3.6.7 en [1]

A.2.6.8. Copias de seguridad

Descripción Provee herramientas para la creación, programación y mantenimiento de copias de seguridad.

Subcaracterísticas

No Aplica

Referencias Característica A.3.6.8 en [1]

A.2.6.9. Repositorio de errores

Descripción La herramienta permite configurar la ruta de una carpeta en la cual se depositarán los logs generados durante la ejecución del sistema. Estos logs son de gran ayuda al momento de detectar errores y corregirlos.

Subcaracterísticas

No Aplica

Referencias Característica A.3.6.9 en [1]

A.2.7. Sistema de gestión documental

Es el módulo encargado de gestionar los documentos en el sistema. Brinda funcionalidades para gestionar la configuración, mantener la metadata y realizar búsquedas sobre los mismos.

A.2.7.1. Integridad y seguridad

Descripción Se da garantía de integridad y de seguridad de acceso a los documentos almacenados. Además del bloqueo de acceso directo a documentos, es decir, encriptación. Los documentos almacenados en el Sistema de gestión documental sólo pueden ser accedidos mediante el BPMS, con usuario autenticado y autorizado y nunca directamente por el sistema de archivos o BD.

Subcaracterísticas

No Aplica

Referencias Característica A.3.7.1 en [1]

A.2.7.2. Reglas de almacenamiento

Descripción Se proveen reglas de almacenamiento parametrizables para cada actividad o tarea: almacenamiento automático en carpetas vinculadas a cada proceso con jerarquía de carpetas; almacenamiento controlado de diferentes versiones de un mismo documento.

Además, se puede consultar la ubicación exacta de un documento en cualquier momento.

Subcaracterísticas

No Aplica

Referencias Característica A.3.7.2 en [1]

A.2.7.3. Definición de metadata

Descripción Se provee la definición de metadata para los documentos de modo de permitir búsquedas y ordenación, además, la asignación de atributos a la metadata puede realizarse a un documento o un lote.

Subcaracterísticas

No Aplica

Referencias Característica A.3.7.3 en [1]

A.2.7.4. Mecanismo de indexación y búsqueda

Descripción Se provee indexación y búsqueda en todo el texto para los formatos (siempre que sea posible): .pdf, .doc, .xls, .ppt, .odt.

La búsqueda debe realizarse también en la metadata de los documentos.

Subcaracterísticas

No Aplica

Referencias Característica A.3.7.4 en [1]

A.2.7.5. Gestión de configuración

Descripción Se proveen mecanismos de SCM (Supply Chain Management) sobre los documentos, tales como:

- Control de actualización de documentos con operaciones de check-in y checkout.
- Control automático de las versiones, dejando registro para auditoría con el nombre de usuario, fecha y hora e identificador del proceso al que está asociado.
- Posibilidad de visualizar todas las versiones de los distintos documentos asociados a un trámite.
- Posibilidad de anexar un mismo documento a más de un proceso.

Subcaracterísticas

No Aplica

Referencias Característica A.3.7.5 en [1]

A.2.8. Portal

Es el principal punto de acceso al sistema para los usuarios finales. Tras autenticarse permite a los mismos acceder a su bandeja de entrada desde la cual pueden tomar tareas y completarlas. Se destacan requerimientos como la usabilidad y la posibilidad de personalizar la interfaz gráfica de acuerdo a sus necesidades para mejorar la experiencia de usuario.

A.2.8.1. Inicio y fin de sesión

Descripción La herramienta permite que cada usuario tenga un identificador único y una contraseña para iniciar sesión en el portal. Además permite que el mismo finalice la sesión.

Subcaracterísticas

- Utiliza el protocolo de acceso ligero a directorios (LDAP) para la autenticación del usuario. Por ejemplo, utiliza *Windows Active Directory*.
- Cierre de sesión automático y parametrizable tras un período de inactividad.

Referencias Característica A.3.8.1 en [1]

A.2.8.2. Presentación de las tareas

Descripción Se presenta al usuario (al ingresar al BPMS) su bandeja de entrada con los procesos o tareas en los que debe participar. Además, se permite:

- Ordenar los trámites por fecha y hora (de llegada, límite de resolución), tipo, prioridad, estado y/o demás campos del proceso.
- Ubicar, por defecto, las tareas de alta prioridad al principio y asignarles una visualización especial.
- Filtrar los procesos visualizados en la bandeja de entrada a través de distintos parámetros.
- Brindar distintas vistas (separadas) para tareas pendientes, completadas, atrasadas y redireccionadas.

Subcaracterísticas

- Permite agregar comentarios o mensajes en las instancias de procesos o tareas.

Referencias Característica A.3.8.2 en [1]

A.2.8.3. Personalización del portal

Descripción Permite la personalización de la bandeja de entrada de cada usuario (por el usuario o supervisor):

- El sistema recuerda la configuración de visualización de cada usuario.
- Permite ordenar las filas por una o varias columnas.
- Permite agregar, quitar y reordenar columnas según el filtrado aplicado.

Subcaracterísticas

No Aplica

Referencias Característica A.3.8.3 en [1]

A.2.8.4. Módulo de ayuda

Descripción Se provee ayuda hipertextual con conceptos generales sobre la aplicación y procesos implementados. Además se permite la obtención de información sobre los pasos del proceso bajo responsabilidad de cada usuario incluyendo visualización del diagrama de flujo, ayuda contextual sobre el diagrama, descripción de actividades y tareas.

Subcaracterísticas

- Posibilidad de que los usuarios puedan actualizar el contenido de la ayuda.

Referencias Característica A.3.8.4 en [1]

A.2.8.5. Mecanismos de búsqueda

Descripción Permite la búsqueda basada en palabras clave, por tipo y nombre de proceso. Además, provee filtros o consultas para acotar el resultado de la búsqueda, tales como: nombre de la tarea, nombre de la instancia de proceso, estado, fecha de inicio, fecha de fin, nombre del participante, datos del proceso.

Subcaracterísticas

- Las consultas pueden ser guardadas para su reutilización.

Referencias Característica A.3.8.5 en [1]

A.2.8.6. Experiencia de usuario

Descripción Permite la personalización de la interfaz de usuario, es decir, intenta ser lo más amigable posible con el usuario, para ello:

- Interfaz de usuario completamente en español local.
- Provee herramientas para que los usuarios
 - Puedan ajustar menús *drop-down*, botones, pantallas.
 - Puedan introducir altas, bajas y modificaciones de registros en las tablas que alimentan las listas desplegables.
 - Puedan definir nuevos tipos, códigos y estados.

- Presentación de mensajes (nombres y tamaños de campos, teclas, etc.) estandarizada.
- Descripciones emergentes en campos y botones.
- Navegación entre páginas con posibilidad de retorno sin pérdida de información (alertando de que se pueden perder datos si el usuario abandona la página actual).

Subcaracterísticas

No Aplica

Referencias Característica A.3.8.6 en [1]

A.2.8.7. Herramientas y técnicas de social media

Descripción El portal incluye funcionalidades para la colaboración en tiempo real entre los usuarios a partir de una vista que integra eventos relevantes para el usuario y permite tomar acción desde una interfaz unificada.

Subcaracterísticas

- *Stream* con eventos relevantes para el usuario en tiempo real.
- Permite suscribirse a distintos tipos de *feeds* del BPMS o externos.
- Permite hacer búsquedas o aplicar filtros sobre los eventos del *stream*.
- Permite completar tareas desde la misma interfaz del feed o desde el dispositivo móvil.
- Permite colaborar, hacer comentarios o preguntas sobre los eventos del *stream*.
- Permite realizar actualizaciones de estado o preguntas (que conforman un *feed* al que pueden suscribirse otros usuarios).
- Permite elegir los destinatarios de las actualizaciones de estado o preguntas.

Referencias Característica A.3.8.7 en [1]

A.3. Definición de criterios no funcionales

En esta sección se definen los criterios no funcionales. Estos criterios especifican el funcionamiento deseable de un BPMS. Para definir estos criterios se utilizó como taxonomía base [15], integrando parte de las taxonomías utilizadas en [1] y [16].

A.3.1. Soporte

En esta categoría se detallan las características que están relacionadas con la documentación y los distintos mecanismos de soporte con los que cuenta el software.[1]

Para la evaluación de esta categoría se tienen en cuenta las siguientes subcaracterística:

A.3.1.1. Proveer soporte para múltiples lenguaje

Descripción Brinda soporte en el lenguaje del usuario y además en otros lenguajes.

Subcaracterísticas

- Soporta el lenguaje del usuario.
- Soporta otros lenguajes.

Referencias [1]**A.3.2. Fiabilidad**

Capacidad del software de mantenerse operativo, durante un tiempo establecido y bajo un conjunto de condiciones definidas.[15] Esta categoría se divide a su vez en las siguientes características:

A.3.2.1. Robusto en ejecución

Descripción Comportamiento estable en ejecución de instancias.

Subcaracterísticas

No aplica

Referencias Característica A.4.2.1 en [1]

A.3.2.2. Disponibilidad del motor

Descripción La disponibilidad de un sistema es una medida de su disposición para el uso.

La disponibilidad es un elemento a considerar cuando se está evaluando la fiabilidad de un sistema.

Subcaracterísticas

- Tiempo medio de fallo (MTTF): Es el tiempo medio de espera hasta el primer fallo de una pieza del sistema.
- Tiempo medio de recuperación (MTTR): El tiempo necesario para reparar una falla de un módulo de software o hardware.
- Cluster: soporta clusterización.

Referencias [15]

A.3.2.3. Disponibilidad del modelador

Descripción La disponibilidad de un sistema es una medida de su disposición para el uso.

La disponibilidad es un elemento a considerar cuando se está evaluando la fiabilidad de un sistema.

Subcaracterísticas

- Tiempo medio de fallo (MTTF): Es el tiempo medio de espera hasta el primer fallo de una pieza del sistema.
- Tiempo medio de recuperación (MTTR): El tiempo necesario para reparar una falla de un módulo de software o hardware.

Referencias [15]

A.3.2.4. Disponibilidad del portal

Descripción La disponibilidad de un sistema es una medida de su disposición para el uso.

La disponibilidad es un elemento a considerar cuando se está evaluando la fiabilidad de un sistema.

Subcaracterísticas

- Tiempo medio de fallo (MTTF): Es el tiempo medio de espera hasta el primer fallo de una pieza del sistema.
- Tiempo medio de recuperación (MTTR): El tiempo necesario para reparar una falla de un módulo de software o hardware.

Referencias [15]**A.3.2.5. Mantenibilidad**

Descripción Es la capacidad para ser reparado y evolucionar. Es una medida menos precisa que la fiabilidad y la disponibilidad.

Subcaracterísticas

- MTTR: Tiempo medio de reparación.
- Modularidad: Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.
- Reusabilidad: Capacidad de un activo que permite que sea utilizado en más de un sistema o en la construcción de otros activos.
- Analizabilidad: Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, identificar las partes a modificar.
- Capacidad para ser modificado: Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- Capacidad de ser probado: Capacidad para validar los cambios que se realizaron en el software.

Referencias [15]**A.3.3. Compatibilidad**

Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software.[16] Esta categoría se divide a su vez en las siguientes características:

A.3.3.1. Coexistencia

Descripción Capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos sin detrimento.

Subcaracterísticas

- El sistema se puede ejecutar en una base compartida.
- El sistema puede ser ejecutado en un servidor compartido.

Referencias [16]

A.3.3.2. Interoperabilidad

Descripción Capacidad de dos o más sistemas o componentes para intercambiar información y hacer uso de la misma.

Subcaracterísticas

- Soporte integración con motores de workflow de terceros.
- Soporta integración con portales (LDAP, IBM WebSphere Portal, Microsoft SharePoint, etc.)

Referencias [1]

A.3.4. Performance

La performance refiere a la capacidad de respuesta: o bien el tiempo requerido para responder a eventos específicos o el número de eventos procesados en un intervalo de tiempo determinado.[15]

A.3.4.1. Tiempo de respuesta

Descripción Es el intervalo de tiempo en el cual se garantiza que la respuesta a un evento será ejecutada.

Subcaracterísticas

- Tiempo de respuesta: es el tiempo desde que se inicia un proceso hasta que comienza la primera tarea.

Referencias [15] [40]

A.3.4.2. Throughput

Descripción Refiere al número de eventos de respuesta que han sido completados en un intervalo de observación dado. Se debe definir más de un intervalo de observación.

Subcaracterísticas

No aplica

Referencias [15]

A.3.4.3. Capacidad

Descripción Es la medida de la cantidad de trabajo que un sistema puede rendir. La capacidad usualmente es definida en términos del throughput.

Subcaracterísticas

No aplica

Referencias [15]

A.3.5. Portabilidad

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro.[16] Esta categoría se divide a su vez en las siguientes características:

A.3.5.1. Adaptabilidad

Descripción Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.

Subcaracterísticas

- Soporta diferentes sistemas operativos (Windows, Linux, OS)
- Cantidad de RAM requerida.
- Cantidad de espacio en disco requerida.
- Soporta diferentes navegadores (IE, Chrome, Firefox)

Referencias [1] [16]

A.3.5.2. Capacidad para ser instalado

Descripción Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.

Subcaracterísticas

- Paquetes de instalación diferenciados: por ejemplo, si tiene el instalador para más de un sistema operativo, si tiene varios paquetes de instalación, uno básico, community, enterprise, etc.
- Dificultad de instalación del ambiente: si tiene un instalador, este puede ser ejecutado desde consola o wizard.
- Tiempo de instalación: El tiempo de instalación del ambiente debe estar especificado. Se puede caracterizar de acuerdo a la granularidad, por ejemplo, en horas, días, semanas o meses.

Referencias [1]

A.3.5.3. Capacidad para ser reemplazado

Descripción Capacidad del producto para ser utilizado en lugar de otro producto de software determinado con el mismo propósito y en el mismo entorno.

Subcaracterísticas

- Capacidad de exportar los procesos en un formato estándar.
- Capacidad de reutilización de los datos de la organización en la base de datos.

Referencias [16]

A.3.5.4. Ambientes diferenciados

Descripción Provee herramientas para poder instalar y probar ambientes de producción, prueba, capacitación y desarrollo diferenciados.

Subcaracterísticas

No Aplica.

Referencias [1]

A.3.5.5. Productos externos

Descripción Tiene una lista de todos los productos externos necesarios para la instalación del producto, por ejemplo, .NET framework 4.0, GeneXus, etc.

Subcaracterísticas

No Aplica.

Referencias [1]

A.3.6. Usabilidad

Esta categoría se divide a su vez en las siguientes características:

A.3.6.1. Capacidad de aprendizaje

Descripción Capacidad del producto que permite al usuario aprender su aplicación.

Subcaracterísticas

- Tooltips - Pregunta: ¿posee tooltips?
- Errores descriptivos - Pregunta: ¿tiene mensajes de error descriptivos?
- Mensajes de sugerencias - Pregunta: ¿brinda sugerencia de posible solución ante un error?
- Manuales de uso - Pregunta: ¿posee manuales de uso?

Referencias [16]

A.3.6.2. Capacidad para ser usado

Descripción Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.

Subcaracterísticas

- Menú de ayuda - Pregunta: ¿brinda menú de ayuda?
- Símbolos mnemotécnicos - Pregunta: ¿tiene símbolos mnemotécnicos?
- Preguntas frecuentes - Pregunta: ¿brinda un espacio de preguntas frecuentes?
- Manuales de uso - Pregunta: ¿posee acceso directo a manuales de uso?

Referencias [16]

A.3.6.3. Protección contra errores de usuario

Descripción Capacidad del sistema para proteger a los usuarios de cometer errores.

Subcaracterísticas

- Mensajes de advertencia - Pregunta: ¿tiene mensajes de advertencia?
- Validación de datos y formularios - Pregunta: ¿posee validación de datos y formularios?
- Deshacer y rehacer - Pregunta: ¿cuenta con opciones para deshacer y rehacer una acción?

Referencias [1] [16]

A.3.6.4. Estética de la interfaz de usuario

Descripción Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.

Subcaracterísticas

- Información innecesaria - Pregunta: ¿incluye información innecesaria?
- Accesibilidad de la información - Pregunta: La información relevante ¿se encuentra accesible y fácil de encontrar?

Referencias [41]

A.3.6.5. Accesibilidad

Descripción Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

Subcaracterísticas

- Comando de voz - Pregunta: ¿incluye comando de voz?
- Acciones se representan con imágenes - Pregunta: ¿las acciones se encuentran representadas con imágenes adecuadas?
- Colores - Pregunta: ¿los colores son distinguibles aunque el usuario sea daltónico?
- Sonidos - Pregunta: ¿los sonidos se corresponden con la acción que se está ejecutando?

Referencias [16]

A.3.7. Seguridad

Capacidad de software para cumplir con los niveles de riesgo permitidos, tanto para posibles daños físicos como para posibles riesgos de datos.

A.3.7.1. Confidencialidad

Descripción Capacidad de proteger a los datos y procesos de la divulgación no autorizada.

Subcaracterísticas

- Políticas de seguridad que aplica. - Pregunta: ¿utiliza técnicas criptográficas?
- La información que se almacena en el dispositivo está cifrada. - Pregunta: Mobile ¿almacena los datos cifrados en el dispositivo?
- Todo intercambio de información con el servidor es sobre HTTPS. - Pregunta: ¿los intercambios de datos con el servidor son sobre HTTPS?

Referencias [1] [15]

A.3.7.2. Integridad

Descripción Capacidad del sistema o componente para prevenir modificaciones no autorizados a datos.

Subcaracterísticas

- Definición de roles. - Pregunta: ¿permite definir roles?

- Definición de permisos sobre objetos. - Pregunta: ¿cuenta con acceso diferenciado a los documentos según el rol?
- Restricciones de acuerdo al rol (registro del organigrama y escalamiento de responsabilidades). - Pregunta: ¿permite visualizar diferentes funciones según el rol?
- Mecanismos de permisos sobre usuarios. - Pregunta: ¿permite administrar permisos?
- Integridad y seguridad de Documentos. - Pregunta: ¿los documentos se guardan cifrados?
- Sesión limitada. - Pregunta: ¿cuenta con un tiempo límite de sesión activa?

Referencias [1] [15]

A.3.7.3. No repudio

Descripción Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.

Subcaracterísticas

- Cuenta con historial de ejecución de procesos. - Pregunta: ¿cuenta con un historial de ejecución de procesos?
- Identificación de acciones de los usuarios. - Pregunta: ¿se puede identificar que usuario y en qué momento se ejecutó un proceso?

Referencias [16]

A.3.7.4. Responsabilidad

Descripción Capacidad de rastrear de forma inequívoca las acciones de una entidad.

Subcaracterísticas

- Las tareas y los procesos tienen identificador único. - Pregunta: ¿las tareas y los procesos tienen identificador único?

Referencias [16]

A.3.7.5. Autenticidad

Descripción Capacidad de demostrar la identidad de un sujeto o un recurso.

Subcaracterísticas

- Sign-in/Sign-out. - Pregunta: ¿cuenta con alguna forma de autenticación de usuarios?
- Soporte para firmas digitales. - Pregunta: ¿soporta firmas digitales?

Referencias [16]

A.4. Definición de características no técnicas

Esta sección posee categorías y características deseables a nivel estratégico, en donde se tienen en cuenta aspectos comerciales.

A.4.1. Licenciamiento

En esta categoría se detallan las características que están relacionadas con la modalidad de licenciamiento y los costos asociados para la obtención y uso, teniendo en cuenta la variación de los mismos en el tiempo.

A.4.1.1. Costo de licencias

Descripción Costo de las licencias al momento de obtención y en forma posterior para su utilización, teniendo en cuenta la variación de los mismos en el tiempo.

Subcaracterísticas

- Para obtención de la herramienta.
- Para la continuidad de su uso.
- Para el soporte brindado.

Referencias Característica A.4.3.1 en [1]

A.4.2. Instalación y soporte

Esta categoría agrupa características relacionadas tanto a la instalación como al soporte. Dentro de esta categoría se incluyen características relacionadas a la documentación, mecanismos de soporte, tiempo de instalación, ambiente diferenciados para la instalación, los productos externos y paquetes de instalación.

A.4.2.1. Dificultad de la instalación del ambiente

Descripción Se especifica que tan difícil es la instalación del ambiente.

Subcaracterísticas

No aplica

Referencias Característica A.4.1.1 en [1]

A.4.2.2. Tiempo de instalación

Descripción El tiempo de instalación del ambiente debe estar especificado. Se puede caracterizar de acuerdo a la granularidad, por ejemplo, en horas, días, semanas o meses.

Subcaracterísticas

- Tiempo de instalación de los distintos módulos que no estén integrados.
- La instalación y desinstalación del producto insume un tiempo y experiencia considerable.

Referencias Característica A.4.1.2 en [1]

A.4.2.3. Ambientes diferenciados

Descripción Provee herramientas para poder instalar y probar ambientes de producción, prueba, capacitación y desarrollo diferenciados.

Subcaracterísticas

No aplica

Referencias Característica A.4.1.4 en [1]

A.4.2.4. Productos externos

Descripción Tiene una lista de todos los productos externos necesarios para la instalación del producto, por ejemplo, .NET framework 4.0, GeneXus, etc.

Subcaracterísticas

No aplica

Referencias Característica A.4.1.5 en [1]

A.4.2.5. Paquetes de instalación

Descripción Se indican los distintos paquetes que ofrece la herramienta.

Subcaracterísticas

- Plataformas en las que corre.
- Versión (Express, Gratuita, etc.).

Referencias Característica A.4.1.8 en [1]

A.4.2.6. Documentación disponible

Descripción Cuenta con manuales de instalación, técnicos y de usuario.

Subcaracterísticas

- El producto posee manuales de instalación
- El producto posee manuales técnicos.
- El producto tiene manuales de usuario.

Referencias [1]

A.4.2.7. Soporte para usuario en línea

Descripción Dispone de un *chat* en línea para consultas específicas.

Subcaracterísticas

No Aplica

Referencias [1]

A.4.2.8. Soporte Telefónico

Descripción Dispone de un *call center* para consultas específicas.

Subcaracterísticas

No Aplica

Referencias [1]

A.4.3. Madurez

Esta categoría agrupa ciertos requerimientos que permiten determinar la madurez de una herramienta, como lo son la actividad de la comunidad, colaboración con otros productos, presencia regional, etc.

A.4.3.1. Tiempo en mercado

Descripción Cantidad de años que la herramienta está disponible en el mercado.

Subcaracterísticas

No aplica

Referencias Característica A.4.2.2 en [1]

A.4.3.2. Actividad de la comunidad

Descripción Interacción registrada entre usuarios en la comunidad, a través de foros, blogs, redes sociales, etc.

Subcaracterísticas

No aplica

Referencias Característica A.4.2.3 en [1]

A.4.3.3. Presencia regional

Descripción Se valora la presencia de la empresa en la región, así como el soporte técnico y comercial regional.

Subcaracterísticas

No aplica

Referencias Característica A.4.2.4 en [1]

A.4.3.4. Visión de futuro y tendencia

Descripción La herramienta tiene un plan de trabajo que asegura su tendencia a la madurez y a apegarse a estándares.

Subcaracterísticas

No aplica

Referencias Característica A.4.2.5 en [1]

A.4.3.5. Costo de pasaje a producción comercial

Descripción El pasaje a producción implica sólo cambios en archivos de configuración, como por ejemplo, en el caso de consumir servicios web, que sólo sea necesario modificar la URL.

Subcaracterísticas

No aplica

Referencias Característica A.4.2.6 en [1]

A.4.4. Lista de características

La tabla A.1 contiene todas las características que forman parte de la metodología. Las características están agrupadas por categoría, a su vez las categorías están divididas en tres grandes grupos: Módulo técnico - No funcional; Módulo técnico - Funcional; Módulo no técnico.

Tabla A.1 Listas de características.

Categoría	Característica
Módulo técnico - Funcional	
Tecnología, arquitectura e interoperabilidad	Compatibilidad con estándares para el modelado de procesos de negocio
	Base de datos
	Arquitectura de la aplicación
	Soporta procesos configurables almacenados en la nube de acuerdo al paradigma SaaS
	Arquitectura del BPMS
	Sistema operativo del modelador
	Navegador Web
	Extensibilidad
	Integración con portales
	Integración con servicios web
	Debugger
	Integración con redes sociales
	Soporte de tecnologías en capa de presentación
	Definir cantidad de usuarios
	Plataformas de servidores
Diseño y modelado de procesos	Diseño gráfico para <i>key users</i>
	Interfaz de diseño intuitiva y centrada en la usabilidad
	Tipo de modelador
	Notación gráfica de procesos compatible con BPMN 2.0
	Listado de versiones de modelos de procesos
	Procedimientos de validación automática de modelos de procesos
	Descubrimiento automático de procesos de negocio
	Trabajo colaborativo sobre procesos
	Información sobre procesos
	Notificaciones a autores de procesos
	Plantillas de procesos
Gestión de formularios	Definición de formularios
	Formato de formularios
	Soporte para dispositivos móviles
	Elementos del formulario

Categoría	Característica
	Validación de campos del formulario
	Formularios dinámicos
	Reglas de llenado de campos
	Restricciones de acuerdo al rol
Motor de <i>Workflow</i>	Soporte de patrones de <i>workflow</i>
	Iniciación de procesos en forma manual y automática
	Opciones de escalamiento
	Mecanismos de notificación
	Definición de reglas de negocio
	Definición de controles de negocio
	Modificar el flujo de trabajo
	Asignación de tareas
	Reasignación de tareas
	Gestión de colas de actividades y tareas
	Modelado de datos
	Adjuntar documentos a tareas
	Ligado de variables
	Vinculación entre tarea y documento
	Registro del organigrama y escalamiento de responsabilidades
	Versionado de proceso
	Integración con motores de <i>workflow</i> de terceros
	Calendario del sistema completo y configurable
	Tratamiento de excepciones
	Deshacer acción sobre una tarea
	Tareas de borrador
	Escalabilidad del motor
	Monitoreo del contenido
	Case Management dinámico
Gestión, monitoreo y auditoría	Gestión de instancias por usuarios
	Mantenimiento histórico del proceso
	Realización de reportes
	Monitoreo de procesos
	Definición y generación de Indicadores Claves de Rendimiento
	Mecanismos de simulación y optimización
	Mecanismos de auditorías

Categoría	Característica
	Copias de seguridad
	Repositorio de errores
Sistema de gestión documental	Integridad y seguridad
	Reglas de almacenamiento
	Definición de metadata
	Mecanismos de indexación y búsqueda
	Gestión de configuración
Portal	Inicio y fin de sesión
	Presentación de las tareas
	Personalización del portal
	Módulo de ayuda
	Mecanismos de búsqueda
	Experiencia de usuarios
	Herramientas y técnicas de social media
Módulo técnico - No funcional	
Soporte	Proveer soporte para múltiples lenguajes
Fiabilidad	Robusto en ejecución
	Disponibilidad del motor
	Disponibilidad del modelador
	Disponibilidad del portal
	Mantenibilidad
Compatibilidad	Coexistencia
	Interoperabilidad
Performance	Tiempo de respuesta
	Throughput
	Capacidad
Portabilidad	Adaptabilidad
	Capacidad para ser instalado
	Capacidad para ser reemplazado
	Ambientes diferenciados
	Productos externos
Usabilidad	Capacidad de aprendizaje
	Capacidad para ser usado
	Protección contra errores de usuario

Categoría	Característica
	Estética de la interfaz de usuario
	Accesibilidad
Seguridad	Confidencialidad
	Integridad
	No repudio
	Responsabilidad
	Autenticidad
Módulo no técnico	
Licenciamiento	Costo de licencias
Instalación y soporte	Dificultad de la instalación del ambiente
	Tiempo de instalación
	Ambientes diferenciados
	Productos externos
	Paquetes de instalación
	Documentación disponible
	Soporte para usuario en línea
	Soporte telefónico
Madurez	Tiempo en mercado
	Actividad de la comunidad
	Presencia regional
	Visión de futuro y tendencia
	Costo de pasaje a producción comercial

A.5. Resumen

Se presentaron todos los criterios utilizados en la metodología. Incluyendo los criterios funcionales existentes, las categorías y características no funcionales que se integraron al módulo técnico. Para la definición se utilizó una estructura que fue definida en la introducción del anexo.

Por último, se brindó la tabla que contiene todas las características que actualmente pertenecen a la metodología.

Anexo B

Evaluación de las herramientas

Este anexo presenta las evaluaciones completas de las herramientas. Además, se brinda la definición de los casos de prueba utilizados en el plan de pruebas definido en la sección 3.2 del capítulo 3.

En la sección B.1 se introduce el tema del anexo. En la sección B.2 se especifica el método de evaluación para cada criterio no funcional. La sección B.3 desarrolla los casos de prueba utilizados en la evaluación de las herramientas. En la sección B.4 se presentan las evaluaciones de todas las herramientas para el conjunto de características seleccionadas.

B.1. Introducción

En esta sección se presenta en forma general cómo se realiza la evaluación de las características en las herramientas.

Al inicio de la sección B.4 se muestra el conjunto de características que son evaluadas en cada herramienta. Se seleccionó un subconjunto de las características descritas en la sección 3.6 del capítulo 3, abarcando tanto características funcionales como no funcionales. Se mantienen los dos criterios de evaluación utilizados en [1]:

- Evaluación práctica: se definen casos de pruebas donde cada uno abarca un conjunto de características específico.
- Evaluación teórica: la evaluación se basa en la documentación de las herramientas y/o en estudios previos.

B.2. Método de evaluación de características no funcionales

En esta sección se mostrará la forma de evaluar cada criterio no funcional definido en la sección A.3 del apéndice A. Las características están divididas por categorías, para ver la definición de las categorías y características ver la sección antes mencionada.

B.2.1. Soporte

B.2.1.1. Proveer soporte para múltiples lenguaje

Subcaracterísticas

Sub1 Soporta el lenguaje del usuario.

Sub2 Soporta otros lenguajes.

Parámetros

No aplica.

Forma de evaluación

Sub1_Q.1: ¿Brinda soporte en lenguaje del usuario?

Sub2_Q.1: ¿Brinda soporte en otro lenguaje?

El nivel de cumplimiento de cada subcaracterística se obtiene de la siguiente manera:

- $R_n = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $R_n = 1 \rightarrow$ Nivel de cumplimiento Alto.

Siendo R_n la respuestas a las preguntas Sub n _Q.1 con $n \in [1, 2]$

Una vez obtenido el nivel de cumplimiento para cada subcaracterística se clasifica la característica siguiendo el método en la sección 3.7.

B.2.2. Fiabilidad

B.2.2.1. Robusto en ejecución

Subcaracterísticas

No aplica.

Parámetros

No aplica.

Forma de evaluación

Q.1 ¿Tiene comportamiento estable en ejecución de instancias?

El $\forall f$ de la característica se obtiene de la siguiente manera:

- $R_1 = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $R_1 = 1 \rightarrow$ Nivel de cumplimiento Alto.

Siendo R1 la respuestas a la pregunta Q.1

Una vez obtenido el νf de la característica se clasifica la característica siguiendo el método en la sección 3.7.

B.2.2.2. Disponibilidad del motor

Subcaracterísticas

- Tiempo medio de fallo (MTTF): Es el tiempo medio de espera hasta el primer fallo de una pieza del sistema.
- Tiempo medio de recuperación (MTTR): El tiempo necesario para reparar una falla de un modulo de software o hardware.
- Cluster: soporta clusterización.

Parámetros

1. $L1$ = cantidad mínima de tiempo hasta que el producto falle por primera vez
 $L2$ = cantidad deseable de tiempo hasta que el producto falle por primera vez
2. $L1$ = cantidad máxima de tiempo para reparar una falla
 $L2$ = cantidad deseable de tiempo para reparar una falla

Forma de evaluación Q.1: ¿Soporta clusterización?

Se utilizará una herramienta (ej. JMeter) para la evaluación de la disponibilidad. Se ejecutan procesos concurrentes, en base a esto se mide la disponibilidad del motor.

El nivel de cumplimiento de las 2 primeras subcaracterísticas se calcula :

- $R1 < L1_n \rightarrow$ Nivel de cumplimiento Bajo.
- $L1_n \leq R1 < L2_n \rightarrow$ Nivel de cumplimiento Medio.
- $L2_n \leq R1 \rightarrow$ Nivel de cumplimiento Alto.
- $R2 > L1_n \rightarrow$ Nivel de cumplimiento Bajo.
- $L1_n > R2 > L2_n \rightarrow$ Nivel de cumplimiento Medio.
- $L2_n > R2 \rightarrow$ Nivel de cumplimiento Alto.

Siendo R1 el MTTF calculado y R2 el MTTR calculado

Si la pregunta 1 tiene una respuesta afirmativa el valor de la Sub3 será alto, en caso contrario, será bajo

Una vez que se obtiene el nivel de cumplimiento de cada subcaracterística se utiliza el método definido en la sección 3.7 para clasificar a la característica.

B.2.2.3. Disponibilidad del modelador

Subcaracterísticas

- Tiempo medio de fallo (MTTF): Es el tiempo medio de espera hasta el primer fallo de una pieza del sistema.
- Tiempo medio de recuperación (MTTR): El tiempo necesario para reparar una falla de un modulo de software o hardware.

Parámetros

1. $L1$ = cantidad mínima de tiempo hasta que el producto falle por primera vez
 $L2$ = cantidad deseable de tiempo hasta que el producto falle por primera vez
2. $L1$ = cantidad máxima de tiempo para reparar una falla
 $L2$ = cantidad deseable de tiempo para reparar una falla

Forma de evaluación Se utilizará una herramienta (ej. Selenium) para la evaluación de la disponibilidad. Se ejecutan diferentes tareas de modelado concurrentes, en base a esto se mide la disponibilidad del motor.

El nivel de cumplimiento de las subcaracterísticas se calcula :

- $R1 < L1_n \rightarrow$ Nivel de cumplimiento Bajo.
- $L1_n \leq R1 < L2_n \rightarrow$ Nivel de cumplimiento Medio.
- $L2_n \leq R1 \rightarrow$ Nivel de cumplimiento Alto.
- $R2 > L1_n \rightarrow$ Nivel de cumplimiento Bajo.
- $L1_n > R2 > L2_n \rightarrow$ Nivel de cumplimiento Medio.
- $L2_n > R2 \rightarrow$ Nivel de cumplimiento Alto.

Siendo $R1$ el MTTF calculado y $R2$ el MTTR calculado

Una vez que se obtiene el nivel de cumplimiento de cada subcaracterística se utiliza el método definido en la sección 3.7 para clasificar a la característica.

B.2.2.4. Disponibilidad del portal**Subcaracterísticas**

- Tiempo medio de fallo (MTTF): Es el tiempo medio de espera hasta el primer fallo de una pieza del sistema.
- Tiempo medio de recuperación (MTTR): El tiempo necesario para reparar una falla de un modulo de software o hardware.

Parámetros

1. $L1$ = cantidad mínima de tiempo hasta que el producto falle por primera vez
 $L2$ = cantidad deseable de tiempo hasta que el producto falle por primera vez
2. $L1$ = cantidad máxima de tiempo para reparar una falla
 $L2$ = cantidad deseable de tiempo para reparar una falla

Se utilizará una herramienta (ej. Selenium) para la evaluación de la disponibilidad. Se ejecutan diferentes tareas de modelado concurrentes, en base a esto se mide la disponibilidad del motor.

El nivel de cumplimiento de las subcaracterísticas se calcula :

Forma de evaluación • $R1 < L1_n \rightarrow$ Nivel de cumplimiento Bajo.

- $L1_n \leq R1 < L2_n \rightarrow$ Nivel de cumplimiento Medio.
- $L2_n \leq R1 \rightarrow$ Nivel de cumplimiento Alto.

- $R2 > L1_n \rightarrow$ Nivel de cumplimiento Bajo.
- $L1_n > R2 > L2_n \rightarrow$ Nivel de cumplimiento Medio.
- $L2_n > R2 \rightarrow$ Nivel de cumplimiento Alto.

Siendo $R1$ el MTTF calculado y $R2$ el MTTR calculado

Una vez que se obtiene el nivel de cumplimiento de cada subcaracterística se utiliza el método definido en la sección 3.7 para clasificar a la característica.

B.2.2.5. Mantenibilidad

Subcaracterísticas

Sub1 MTTR: Tiempo medio de reparación.

Sub2 Modularidad: Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.

Sub3 Reusabilidad: Capacidad de un activo que permite que sea utilizado en más de un sistema o en la construcción de otros activos.

Sub4 Analizabilidad: Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, identificar las partes a modificar.

Sub5 Capacidad para ser modificado: Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.

Sub6 Capacidad de ser probado: Capacidad para validar los cambios que se realizaron en el software.

Parámetros

1. $L1$ = cantidad máxima de tiempo de reparación
 $L2$ = cantidad deseable de tiempo hasta que el producto falle por primera vez
2. $L1$ = cantidad mínima de pruebas que debe proveer el producto.
 $L2 = L1$

Forma de evaluación

Preguntas:

Sub1 Q1 ¿Cuál es el tiempo medio de reparación?

Sub2 Q1 ¿Cuál es el nivel de cohesión?

Sub3 Q1 ¿Es el sistema reutilizable?

Sub4 Q1 ¿Cuál es el nivel de acoplamiento?

Sub5 Q1 ¿Dispone de módulos diferenciados?

Sub6 Q1 ¿Provee demo del producto?

Q2 ¿Provee procesos de pruebas para probar conectores?

- Q3 ¿Provee procesos de pruebas para probar acceso a base de datos?
- Q4 ¿Provee procesos de pruebas para probar notificaciones por mails?
- Q5 ¿Provee procesos de pruebas para probar la ejecución de un proceso?
- Q6 ¿Provee procesos de pruebas para probar roles en un proceso? ¿Cuántos?

B.2.3. Compatibilidad

Esta categoría se divide a su vez en las siguientes características:

B.2.3.1. Coexistencia

Subcaracterísticas

- Sub1 El sistema se puede ejecutar en una base compartida.
- Sub2 El sistema puede ser ejecutado en un servidor compartido.

Parámetros

No aplica.

Forma de evaluación

Sub1 Q.1: ¿El sistema es capaz de ser ejecutado en una base de datos compartida con otras aplicaciones?

Sub2 Q.1: ¿El sistema es capaz de ser ejecutado en un servidor compartido con otras aplicaciones?

El nivel de cumplimiento de cada subcaracterística se obtiene de la siguiente manera:

- $R_n = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $R_n = 1 \rightarrow$ Nivel de cumplimiento Alto.

Siendo R_n la respuestas a las preguntas Sub n _Q.1 con $n \in [1, 2]$

Una vez obtenido el nivel de cumplimiento para cada subcaracterística se clasifica la característica siguiendo el método en la sección 3.7.

B.2.3.2. Interoperabilidad

Subcaracterísticas

- Sub1 Soporta integración con portales (LDAP, IBM WebSphere Portal, Microsoft SharePoint, etc.)
- Sub2 Soporte integración con motores de workflow de terceros.

Parámetros

- Sub1 L1 = mínima cantidad de portales soportados
- L2 = cantidad deseable de portales soportados.

Forma de evaluación

- Sub1_Q.1: ¿El portal se integra con LDAP?
- Q.2: ¿El portal se integra con IBM WebSphere Portal?
- Q.3: ¿El portal se integra con Microsoft SharePoint?
- Q.4: ¿El portal se integra con mySAP?

Q.5: ¿El portal se integra con Oracle Portal?

Sub2_Q.1: ¿Soporta integración con motores de workflow de terceros?

Para calcular el nivel de cumplimiento de la subcaracterística 1, primero se obtiene v_{sub} de la siguiente manera:

$$v_{sub} = \sum_{i=1}^5 Sub1\_Ri$$

El nivel de cumplimiento de la subcaracterística 2 se obtiene de la siguiente manera:

- $v_{sub} > L1 \rightarrow$ Nivel de cumplimiento Bajo.
- $L1 > v_{sub} > L2 \rightarrow$ Nivel de cumplimiento Medio.
- $L2 > v_{sub} \rightarrow$ Nivel de cumplimiento Alto.

El nivel de cumplimiento de la subcaracterística 2 se obtiene de la siguiente manera:

- $Sub2\_R1 = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $Sub2\_R1 = 1 \rightarrow$ Nivel de cumplimiento Alto.

Siendo $Sub2\_R1$ la respuestas a las preguntas $Sub2\_Q.1$.

Una vez obtenido el nivel de cumplimiento para cada subcaracterística se clasifica la característica siguiendo el método en la sección 3.7.

B.2.4. Performance

B.2.4.1. Tiempo de respuesta

Subcaracterísticas

Sub1 Tiempo de respuesta: es el tiempo desde que se inicia un proceso hasta que comienza la primera tarea. [40]

Parámetros

Sub1 $L1$ = tiempo de respuesta deseable, en segundos.

$L2$ = tiempo de respuesta máximo aceptable, en segundos.

Forma de evaluación

Sub1_Q.1: ¿Cuál es el tiempo de respuesta?

$$v_{sub} = tiempoRespuesta$$

El nivel de cumplimiento para la característica se obtiene de la siguiente manera:

- $v_{sub} < L1 \rightarrow$ Nivel de cumplimiento Alto.
- $L1 \leq v_{sub} < L2 \rightarrow$ Nivel de cumplimiento Medio.
- $L2 \leq v_{sub} \rightarrow$ Nivel de cumplimiento Bajo.

Modo de prueba Para la evaluación de tiempo de ejecución de tareas se definió el caso de prueba 6 definido en la sección B.3.0.1. Además, tomando como guía el benchmark definido en [42], se ejecutarán de forma automática, al menos 100 veces cada uno de los siguientes casos: 1 usuario, 2, usuarios, 4 usuarios, 6 usuarios, 8 usuarios y 10

usuarios de forma secuencial. Las ejecuciones se realizaran con una base local, sin guardar la historia del proceso.

A partir de los datos recabados se toma el tiempo promedio y este valor se considerará el *tiempoRespuesta* buscado.

B.2.4.2. Throughput

Subcaracterísticas

No aplica

Parámetros

L1 = Es el porcentaje mínimo de procesos que se deben ejecutar en el período *PeriodoEjecucion*.

L2 = Es el porcentaje deseable de procesos que se deben ejecutar en el período *PeriodoEjecucion*.

Forma de evaluación Se define un período de ejecución (*PeriodoEjecucion*) y la cantidad de servicios que se ejecutarán (*CantTotalServicios*) en ese período. Se determina el valor del Throughput de la siguiente manera:

$$v_{sub} = \frac{CantServiciosIniciados * 100}{CantTotalServicios} \quad (B.1)$$

Siendo *CantServiciosIniciados*, la cantidad de servicios que se iniciaron en *PeriodoEjecucion*.

El nivel de cumplimiento del Throughput se calcula de la siguiente manera:

- $v_{sub} < L1 \rightarrow$ Nivel de cumplimiento Bajo.
- $L1 \leq v_{sub} < L2 \rightarrow$ Nivel de cumplimiento Medio.
- $L2 \leq v_{sub} \rightarrow$ Nivel de cumplimiento Alto.

Modo de prueba Con el caso de prueba definido en la sección B.3.0.1 se ejecutará automáticamente durante 30 minutos. Tomando como guía el benchmark definido en [42], se definen los siguientes casos: 1 usuario, 2, usuarios, 4 usuarios, 6 usuarios, 8 usuarios y 10 usuarios de forma secuencial. La cantidad total de servicios es 1000. Las ejecuciones se realizaran con una base local, sin guardar la historia del proceso.

B.2.4.3. Capacidad

Subcaracterísticas

No aplica

Parámetros

L1 = Es el porcentaje mínimo de procesos completos en el *TiempoEjecucion*.

L2 = Es el porcentaje deseable de procesos completos en el *TiempoEjecucion*.

Forma de evaluación Se define un período de ejecución (*PeriodoEjecucion*) y la cantidad de servicios que se ejecutarán en ese período. El valor final de la Capacidad se determina

de la siguiente manera:

$$vf = \frac{CantServiciosCompletados * 100}{CantTotalServicios} \quad (B.2)$$

Siendo *CantServiciosCompletados*, la cantidad de servicios que se completaron en un tiempo de ejecución menor o igual a *TiempoEjecucion* (es el tiempo deseable para la finalización de un proceso, este tiempo es definido por el evaluador) y, *CantTotalServicios*, es la cantidad de servicios que se completaron en el período de ejecución. El nivel de cumplimiento de la Capacidad se calcula de la siguiente manera:

- $v_{sub} < L1 \rightarrow$ Nivel de cumplimiento Bajo.
- $L1 \leq v_{sub} < L2 \rightarrow$ Nivel de cumplimiento Medio.
- $L2 \leq v_{sub} \rightarrow$ Nivel de cumplimiento Alto.

Modo de prueba Para obtener el valor final de esta característica se definió el caso de prueba 6, ver sección B.3.0.1. A partir del benchmark definido en [42], se definieron los siguientes casos: 1 usuario, 2, usuarios, 4 usuarios, 6 usuarios, 8 usuarios y 10 usuarios de forma secuencial. Estos casos se ejecutan durante 30 minutos y se esperan una cantidad total de 1000 servicios, con una base local, sin guardar el historial del proceso.

B.2.5. Portabilidad

B.2.5.1. Adaptabilidad

Subcaracterísticas

- Sub1 Soporta diferentes sistemas operativos (Windows, Linux, OS)
- Sub2 Cantidad de RAM requerida.
- Sub3 Cantidad de espacio en disco requerida.
- Sub4 Soporta diferentes navegadores (IE, Chrome, Firefox)

Parámetros

- Sub1 L1 = Mínima cantidad de sistemas operativos soportados.
L2 = Cantidad deseable de sistemas operativos soportados.
- Sub2 L1 = Cantidad de RAM deseable.
L2 = Máxima cantidad de RAM requerida.
- Sub3 L1 = Cantidad de espacio en disco deseable.
L2 = Máxima cantidad de espacio en disco requerido.
- Sub4 L1 = Mínima cantidad de navegadores soportados.
L2 = Cantidad deseable de navegadores soportados.

Forma de evaluación

- Sub1_Q.1: ¿Soporta el sistema operativo Windows?

Q.2: ¿Soporta el sistema operativo Linux?

Q.3: ¿Soporta el sistema operativo OS?

Sub2_Q.1: ¿Cuál es el tamaño mínimo de RAM requerido?

Sub3_Q.1: ¿Cuál es la mínima cantidad de espacio en disco requerida?

Sub4_Q.1: ¿Soporta el navegador IE?

Q.2: ¿Soporta el navegador Chrome?

Q.3: ¿Soporta el navegador Mozilla?

El v_{sub} de la subcaracterística 1 se calcula:

$$v_{sub} = \sum_{i=1}^3 Sub1\_Ri$$

Donde Sub1_Ri vale uno si la respuesta es positiva y cero si la respuesta es negativa.

El nivel de cumplimiento de la subcaracterística 1, se obtiene de la siguiente manera:

- $v_{sub} < L1 \rightarrow$ Nivel de cumplimiento Bajo.
- $L1 \leq v_{sub} < L2 \rightarrow$ Nivel de cumplimiento Medio.
- $L2 \leq v_{sub} \rightarrow$ Nivel de cumplimiento Alto.

El nivel de cumplimiento de la subcaracterística 2 se obtiene de la siguiente manera:

- $Sub2\_R1 < L1 \rightarrow$ Nivel de cumplimiento Alto.
- $L1 \leq Sub2\_R1 < L2 \rightarrow$ Nivel de cumplimiento Medio.
- $L2 \leq Sub2\_R1 \rightarrow$ Nivel de cumplimiento Bajo.

El nivel de cumplimiento de la subcaracterística 3 se obtiene de la siguiente manera:

- $Sub3\_R1 < L1 \rightarrow$ Nivel de cumplimiento Alto.
- $L1 \leq Sub3\_R1 < L2 \rightarrow$ Nivel de cumplimiento Medio.
- $L2 \leq Sub3\_R1 \rightarrow$ Nivel de cumplimiento Bajo.

El v_{sub} de la subcaracterística 4 se calcula:

$$v_{sub} = \sum_{i=1}^3 Sub4\_Ri$$

Donde Sub4_Ri vale uno si la respuesta es positiva y cero si la respuesta es negativa.

El nivel de cumplimiento de la subcaracterística 4, se obtiene de la siguiente manera:

- $v_{sub} < L1 \rightarrow$ Nivel de cumplimiento Bajo.
- $L1 \leq v_{sub} < L2 \rightarrow$ Nivel de cumplimiento Medio.
- $L2 \leq v_{sub} \rightarrow$ Nivel de cumplimiento Alto.

B.2.5.2. Capacidad para ser instalado

Subcaracterísticas

- Sub1 Paquetes de instalación diferenciados: por ejemplo, si tiene el instalador para más de un sistema operativo, si tiene varios paquetes de instalación, uno básico, community, enterprise, etc.
- Sub2 Dificultad de instalación del ambiente: si tiene un instalador, este puede ser ejecutado desde consola o wizard.
- Sub3 Tiempo de instalación: El tiempo de instalación del ambiente debe estar especificado. Se puede caracterizar de acuerdo a la granularidad, por ejemplo, en horas, días, semanas o meses.

Parámetros

- Sub1 L1 = la cantidad mínima de paquetes de instalación diferenciados que define el evaluador.
L2 = cantidad deseable de paquetes de instalación diferenciados.
- Sub2 L1 = mínima cantidad de elementos necesarios para la instalación
L2 = máxima cantidad de elementos necesarios para la instalación
- Sub3 L1 = tiempo deseable de instalación.
L2 = máximo tiempo de instalación.

Forma de evaluación

- Sub1_Q.1: ¿Cuántos paquetes de instalación diferenciados provee?
- Sub2_Q.1: ¿Se necesitan plugins para la instalación?
Q.2: ¿Se necesita la adquisición de servidores para la instalación?
Q.3: ¿Se necesita la adquisición de una base de datos para la instalación?
Q.4: ¿Es necesario un instalador o tiene comandos por consola?
- Sub3_Q.1: ¿Cuál es el tiempo de instalación?

El nivel de cumplimiento de la subcaracterística 1, se obtiene de la siguiente manera:

- $Sub1\_R1 < L1 \rightarrow$ Nivel de cumplimiento Bajo.
- $L1 \leq Sub1\_R1 < L2 \rightarrow$ Nivel de cumplimiento Medio.
- $L2 \leq Sub1\_R1 \rightarrow$ Nivel de cumplimiento Alto.

El v_{sub} de la subcaracterística 2 se calcula:

$$v_{sub} = \sum_{i=1}^4 Sub2\_Ri$$

Donde Sub2_Ri vale uno si la respuesta es positiva y cero si la respuesta es negativa.

El nivel de cumplimiento de la subcaracterística 1, se obtiene de la siguiente manera:

- $v_{sub} < L1 \rightarrow$ Nivel de cumplimiento Alto.
- $L1 \leq v_{sub} < L2 \rightarrow$ Nivel de cumplimiento Medio.
- $L2 \leq v_{sub} \rightarrow$ Nivel de cumplimiento Bajo.

El nivel de cumplimiento de la subcaracterística 3, se obtiene de la siguiente manera:

- $Sub3\_R1 < L1 \rightarrow$ Nivel de cumplimiento Alto.
- $L1 \leq Sub3\_R1 < L2 \rightarrow$ Nivel de cumplimiento Medio.
- $L2 \leq Sub3\_R1 \rightarrow$ Nivel de cumplimiento Bajo.

B.2.5.3. Capacidad para ser reemplazado

Subcaracterísticas

Sub1 Capacidad de exportar los procesos en un formato estándar.

Sub2 Capacidad de reutilización de los datos de la organización en la base de datos.

Parámetros

No aplica.

Forma de evaluación

Sub1_Q.1: ¿Permite exportar los procesos en un formato estándar?

Sub2_Q.1: ¿Es posible reutilizar los datos de la organización en la base de datos?

El nivel de cumplimiento de la subcaracterística 1 se obtiene de la siguiente manera:

- $Sub1\_R1 = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $Sub1\_R1 = 1 \rightarrow$ Nivel de cumplimiento Alto.

El nivel de cumplimiento de la subcaracterística 2 se obtiene de la siguiente manera:

- $Sub2\_R1 = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $Sub2\_R1 = 1 \rightarrow$ Nivel de cumplimiento Alto.

B.2.5.4. Ambientes diferenciados

Subcaracterísticas

No Aplica.

Parámetros

L1 = mínima cantidad de ambientes.

L2 = cantidad deseable de ambientes.

Forma de evaluación

Q.1: ¿Provee un ambiente de producción?

Q.2: ¿Provee un ambiente de pruebas?

Q.3: ¿Provee un ambiente de capacitación?

Q.4: ¿Provee un ambiente de desarrollo?

Para obtener el nivel de cumplimiento de la característica se calcula la cantidad de ambiente que provee:

$$cantAmbientes = \sum_{i=1}^4 Ri$$

Donde Ri vale uno si la respuesta es positiva y cero si la respuesta es negativa.

El nivel de cumplimiento de la característica, se obtiene de la siguiente manera:

- $cantAmbientes < L1 \rightarrow$ Nivel de cumplimiento Bajo.
- $L1 \leq cantAmbientes < L2 \rightarrow$ Nivel de cumplimiento Medio.
- $L2 \leq cantAmbientes \rightarrow$ Nivel de cumplimiento Alto.

B.2.5.5. Productos externos

Subcaracterísticas

No aplica.

Parámetros

No aplica.

Forma de evaluación

Q.1: ¿Cuenta con una lista de productos externos?

El nivel de cumplimiento de la característica se calcula:

- $R1 = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $R1 = 1 \rightarrow$ Nivel de cumplimiento Alto.

B.2.6. Usabilidad

Esta categoría se divide a su vez en las siguientes características:

B.2.6.1. Capacidad de aprendizaje

Subcaracterísticas

Sub1 Tooltips

Sub2 Errores descriptivos

Sub3 Sugerencias

Sub4 Manuales de uso

Forma de evaluación

Q.1: ¿Posee tooltips?

Q.2: ¿Tiene mensajes de error descriptivos?

Q.3: ¿Brinda sugerencia de posible solución ante un error?

Q.4: ¿Posee manuales de uso?

El nivel de cumplimiento de las subcaracterísticas se calcula:

- $R1 = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $R1 = 1 \rightarrow$ Nivel de cumplimiento Alto.

Una vez obtenido el nivel de cumplimiento para cada subcaracterística se clasifica la característica siguiendo el método en la sección 3.7.

B.2.6.2. Capacidad para ser usado

Subcaracterísticas

Sub1 Menú de ayuda

Sub2 Símbolos mnemotécnicos

Sub3 Preguntas frecuentes

Sub4 Manuales de uso

Forma de evaluación

Q.1: ¿Brinda un menú de ayuda?

Q.2: ¿Tiene símbolos mnemotécnicos?

Q.3: ¿Brinda un espacio de preguntas frecuentes?

Q.4: ¿Posee acceso directo a manuales de uso?

El nivel de cumplimiento de las subcaracterísticas se calcula:

- $R1 = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $R1 = 1 \rightarrow$ Nivel de cumplimiento Alto.

Una vez obtenido el nivel de cumplimiento para cada subcaracterística se clasifica la característica siguiendo el método en la sección 3.7.

B.2.6.3. Protección contra errores de usuario

Subcaracterísticas

Sub1 Mensajes de advertencia

Sub2 Validación de datos y formularios

Sub3 Deshacer y rehacer

Forma de evaluación

Q.1: ¿Tiene mensajes de advertencia?

Q.2: ¿Posee validación de datos y formularios?

Q.3: ¿Cuenta con opciones para deshacer y rehacer una acción?

El nivel de cumplimiento de las subcaracterísticas se calcula:

- $R1 = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $R1 = 1 \rightarrow$ Nivel de cumplimiento Alto.

Una vez obtenido el nivel de cumplimiento para cada subcaracterística se clasifica la característica siguiendo el método en la sección 3.7.

B.2.6.4. Estética de la interfaz de usuario

Subcaracterísticas

Sub1 Información innecesaria

Sub2 Accesibilidad de la información

Forma de evaluación

Q.1: ¿Incluye información innecesaria?

Q.2: La información relevante ¿Se encuentra accesible y fácil de encontrar?

El nivel de cumplimiento de la Sub1 se calcula:

- $R1 = 0 \rightarrow$ Nivel de cumplimiento Alto.
- $R1 = 1 \rightarrow$ Nivel de cumplimiento Bajo.

El nivel de cumplimiento de la Sub2 se calcula:

- $R2 = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $R2 = 1 \rightarrow$ Nivel de cumplimiento Alto.

Una vez obtenido el nivel de cumplimiento para cada subcaracterística se clasifica la característica siguiendo el método en la sección 3.7.

B.2.6.5. Accesibilidad

Subcaracterísticas

Sub1 Comando de voz

Sub2 Acciones se representan con imágenes

Sub3 Colores

Sub4 Sonidos

Forma de evaluación

Q.1: ¿Incluye comando de voz?

Q.2: ¿Las acciones se encuentran representadas con imágenes adecuadas?

Q.3: ¿Los colores son distinguibles aunque el usuario sea daltónico?

Q.4: ¿Los sonidos se corresponden con la acción que se está ejecutando?

El nivel de cumplimiento de las subcaracterísticas se calcula:

- $R2 = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $R2 = 1 \rightarrow$ Nivel de cumplimiento Alto.

Una vez obtenido el nivel de cumplimiento para cada subcaracterística se clasifica la característica siguiendo el método en la sección 3.7.

B.2.7. Seguridad

B.2.7.1. Confidencialidad

Subcaracterísticas

Sub1 Políticas de seguridad que aplica.

Sub2 La información que se almacena en el dispositivo está cifrada.

Sub3 Todo intercambio de información con el servidor es sobre HTTPS.

Parámetros

No aplica

Forma de evaluación

Sub1_Q.1: ¿Utiliza técnicas criptográficas?

Sub2_Q.1: En caso de tener un cliente mobile ¿Almacena los datos cifrados en el dispositivo?

Sub3_Q.1: ¿Los intercambios de datos con el servidor son sobre HTTPS?

El nivel de cumplimiento de cada subcaracterística se obtiene de la siguiente manera:

- $Rn = 0 \rightarrow$ Nivel de cumplimiento Bajo.

- $R_n = 1 \rightarrow$ Nivel de cumplimiento Alto.

Siendo R_n la respuestas a las preguntas Subn_Q.1 con $n \in [1, 3]$

Una vez obtenido el nivel de cumplimiento para cada subcaracterística se clasifica la característica siguiendo el método en la sección 3.7 del capítulo.

Modo de prueba

Técnicas criptográficas : Se verifica en la base que las contraseñas estén cifradas.

Datos cifrados en mobile : Se verifica que los archivos que almacena en el dispositivo estén cifrados.

Información sobre HTTPS : Se ejecuta un proceso con una tarea, en dicha tarea se hace un pedido de un archivo y se verifica mediante alguna herramienta, que el intercambio haya sido sobre HTTPS.

B.2.7.2. Integridad

Subcaracterísticas

Sub1 Definición de roles.

Sub2 Definición de permisos sobre objetos.

Sub3 Restricciones de acuerdo al rol (registro del organigrama y escalamiento de responsabilidades).

Sub4 Mecanismos de permisos sobre usuarios.

Sub5 Integridad y seguridad de Documentos.

Sub6 Sesión limitada.

Parámetros

No aplica

Forma de evaluación

Sub1_Q.1: ¿Permite definir roles?

Sub2_Q.1: ¿Cuenta con acceso diferenciado a los documentos según el rol?

Sub3_Q.1: ¿Permite visualizar diferentes funciones según el rol?

Sub4_Q.1: ¿Permite administrar permisos?

Sub5_Q.1: ¿Los documentos se guardan cifrados?

Sub6_Q.1: ¿Cuenta con un tiempo límite de sesión activa?

El nivel de cumplimiento de cada subcaracterística se obtiene de la siguiente manera:

- $R_n = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $R_n = 1 \rightarrow$ Nivel de cumplimiento Alto.

Siendo R_n la respuestas a las preguntas Subn_Q.1 con $n \in [1, 6]$

Una vez obtenido el nivel de cumplimiento para cada subcaracterística se clasifica la característica siguiendo el método en la sección 3.7.

Modo de prueba

Definición de roles: Verificar de forma teórica que tenga esta subcaracterística. En el caso de poder definir roles, se definen al menos dos roles (Admin, Usuario).

Definición de permisos sobre objetos: Con rol Admin, se ejecuta un proceso que tiene una tarea para cargar un archivo con permiso de Admin. En la segunda tarea se carga un archivo con permiso Usuario. La tercer tarea está asignada a usuarios con rol Usuario, donde aparecen todos los archivos con permiso de Usuario.

Restricciones de acuerdo al rol: Dados los roles definidos anteriormente se verifica que ciertas acciones sean visibles sólo para el usuario con rol Admin.

Mecanismos de permisos sobre usuarios: Verificar que un usuario con rol Admin puede asignarle permisos a los roles definidos por debajo de su jerarquía.

Integridad y seguridad de documentos: Se ejecuta un proceso que tiene una tarea de cargar y guardar un archivo (pdf). Luego se verifica en el destino que el archivo esté cifrado.

Sesión limitada: Verificar de forma teórica que tenga esta subcaracterística. En caso de tenerla esperar la cantidad de tiempo necesaria y ejecutar una acción. Esta acción debería ser ejecutada y debería aparecer la pantalla de login.

B.2.7.3. No repudio

Subcaracterísticas

Sub1 Cuenta con historial de ejecución de procesos.

Sub2 Identificación de acciones de los usuarios.

Parámetros

No aplica.

Forma de evaluación

Sub1_Q.1: ¿Cuenta con un historial de ejecución de procesos?

Sub2_Q.1: ¿Se puede identificar que usuario y en que momento se ejecutó un proceso?

El nivel de cumplimiento de cada subcaracterística se obtiene de la siguiente manera:

- $R_n = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $R_n = 1 \rightarrow$ Nivel de cumplimiento Alto.

Siendo R_n la respuestas a las preguntas Subn_Q.1 con $n \in [1, 2]$

Una vez obtenido el nivel de cumplimiento para cada subcaracterística se clasifica la característica siguiendo el método en la sección 3.7.

Modo de prueba

Historial e identificación: Se ejecuta un proceso con 2 usuarios diferentes. Luego se verifica en el portal y en el log (depende de la herramienta) si quedó registrada la ejecución del proceso, quién y en qué momento fueron ejecutados.

B.2.7.4. Responsabilidad

Subcaracterísticas

Sub1 Las tareas y los procesos tienen identificador único.

Forma de evaluación

Sub1_Q.1: ¿Las tareas y los procesos tienen identificador único?

El nivel de cumplimiento se obtiene de la siguiente manera:

- $R1 = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $R1 = 1 \rightarrow$ Nivel de cumplimiento Alto.

Siendo $R1$ la respuesta a la pregunta Sub1_Q.1

Una vez obtenido el nivel de cumplimiento para la subcaracterística se clasifica la característica siguiendo el método en la sección 3.7.

Modo de prueba

Identificador Único: Se verifica en la base de datos que en las tablas donde se guardan los procesos y las tareas, el campo identificador tenga la propiedad *unique*.

B.2.7.5. Autenticidad

Subcaracterísticas

Sub1 Sign-in/Sign-out.

Sub2 Soporte para firmas digitales.

Forma de evaluación

Sub1_Q.1: ¿Cuenta con alguna forma de autenticación de usuarios?

Sub2_Q.1: ¿Soporta firmas digitales?

El nivel de cumplimiento de cada subcaracterística se obtiene de la siguiente manera:

- $Rn = 0 \rightarrow$ Nivel de cumplimiento Bajo.
- $Rn = 1 \rightarrow$ Nivel de cumplimiento Alto.

Siendo Rn la respuestas a las preguntas Subn_Q.1 con $n \in [1, 2]$

Una vez obtenido el nivel de cumplimiento para cada subcaracterística se clasifica la característica siguiendo el método en la sección 3.7.

Modo de prueba

Sign-in/Sign-out: Para evaluar esta subcaracterística se inicia la aplicación con las credenciales definidas en la base. Luego de iniciada la sesión, ejecutar un proceso de prueba previamente cargado. Para finalizar esta prueba se cierra sesión y se trata de ejecutar una acción mediante la URL. La aplicación deberá rechazar esta acción.

Soporte para firmas digitales: Se verifica teóricamente esta funcionalidad. En caso de tenerla se ejecuta un proceso que tiene una tarea de cargar y enviar un archivo (pdf). Luego se verifica en el destino que el archivo esté cifrado.

B.3. Casos de prueba

En esta sección se definen los casos de prueba utilizados para llevar a cabo la ejecución del plan de pruebas definido en la sección 3.2.

Los casos presentan dos aspectos a tener en cuenta, la estructura organizacional y la estructura del propio caso de pruebas. La estructura organizacional refiere a los usuarios y roles que se emplean. A continuación se muestra la estructura del caso de pruebas:

- *Introducción*: muestra una explicación a grandes rasgos de lo que realiza el caso de prueba.
- *Modelo proceso*: muestra el diagrama del proceso.
- *Explicativo de la secuencia*: conjunto de pasos explicativos de cada prueba.
- *Observaciones*: se muestran observaciones de la explicación de la secuencia.
- *Características abarcadas*: conjunto de características incluidas en la lista que abarca el caso de prueba.

B.3.0.1. Caso de prueba 1

Introducción

Este caso de prueba tiene como principal objetivo evaluar las características de la categoría performance. Con base en el benchmark [42]. El caso presenta un proceso simple, con una tarea de usuario, sin formularios o llamados externos. Con esto se busca obtener de manera sencilla el tiempo de respuesta del motor al crear una tarea.

Diagrama del caso



Figura B.1 Modelo del caso de prueba 1

Explicación de la secuencia

1. Inicio: El caso de uso comienza cuando un usuario inicia una instancia del proceso.
2. Tarea A: El usuario que inicio la instancia finaliza la tarea.

Características abarcadas

1. Tiempo de respuesta

2. Throughput

3. Capacidad

B.3.0.2. Caso de prueba 2

Introducción

En este caso se definen diferentes roles con el fin de probar la seguridad. Cuenta con actividades en las cuales se cargan y guardan documentos.

Diagrama del caso

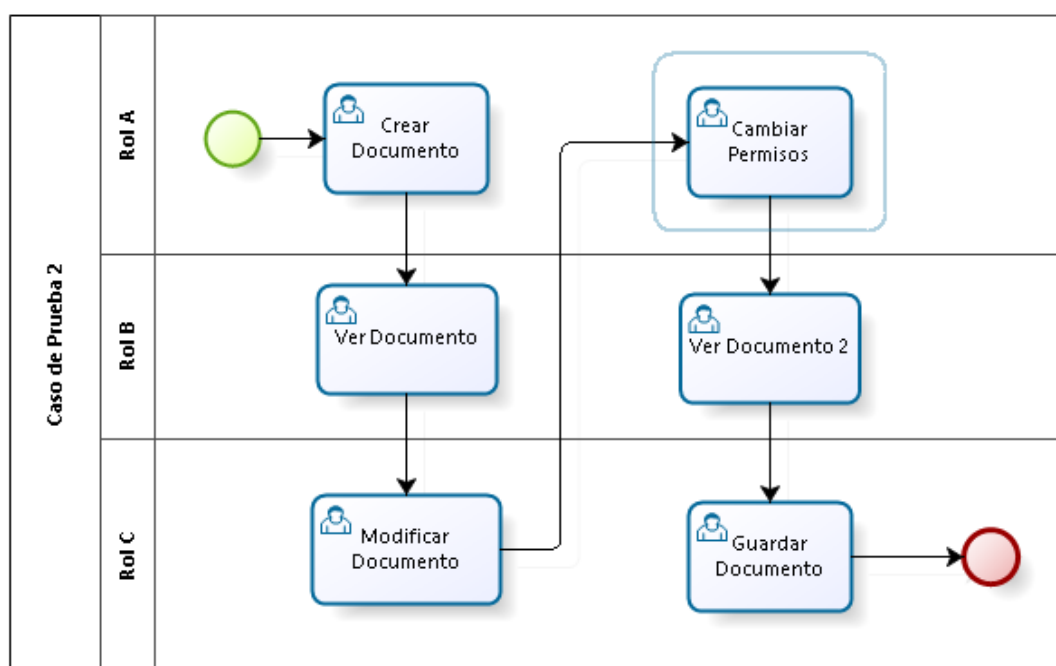


Figura B.2 Modelo del caso de prueba 2

Explicación de la secuencia

1. Inicio: El usuario inicia una instancia del proceso
2. Crear Documento: Actividad realizada por usuarios con rol A en donde se crea un documento con permisos de lectura para el rol A y rol C
3. Ver Documento: Actividad ejecutada por usuario con rol B que intenta ver el documento creado por el rol A
4. Modificar Documento: Abre el documento creado por el rol A y lo modifica
5. Cambiar permisos: Actividad realizada por usuarios con rol A en donde se modifica los permisos de lectura y escritura para que los usuarios con el rol B puedan ver y modificar el documento.
6. Ver documento 2: Un usuario con el rol B abre el documento modificado por el rol A

7. Guardar documento: Actividad en la cual se guarda el documento.

Características No funcionales abarcadas

1. Confidencialidad
2. Integridad
3. No repudio
4. Responsabilidad
5. Autenticidad (Sing in/Sing out)

B.3.0.3. Caso de prueba 3

Introducción

Este caso contiene diferentes tipos de actividades (ingreso de datos, comunicación con la base de datos y envío de notificaciones). El objetivo de este caso de pruebas es poder probar tanto la conexión con una base de datos, como el envío de notificaciones.

Diagrama del caso

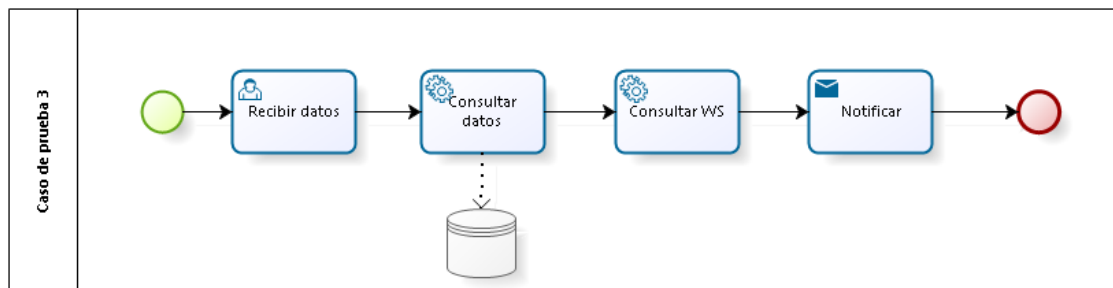


Figura B.3 Modelo del caso de prueba 3

Explicación de la secuencia

1. Inicio: El caso de uso comienza cuando un usuario inicia una instancia del proceso.
2. Recibir datos: En esta actividad, el usuario que inicia la instancia del proceso ingresa un dato en un formulario, por ejemplo, un nombre.
3. Consultar datos: Esta actividad es automática y realiza una consulta a una base de datos, con el dato ingresado en la actividad “Recibir datos”. Al finalizar la actividad se obtiene un resultado de la consulta.
4. Consultar WS: Es una actividad de servicio que consume un servicio web, obteniendo algún resultado.
5. Enviar datos: El valor que se obtiene en la consulta a la base de la actividad “Consultar datos” y, el que se obtiene en la consulta al servicio web en la

tarea “Consultar WS” se envía en una notificación. Esta actividad también es automática.

Observaciones

Se recomienda crear una tabla con no más de 3 columnas, por ejemplo, identificador, nombre y rol. El formulario de la actividad “Recibir datos” puede tener un campo de texto, donde se ingrese el nombre, luego, en la actividad “Consultar datos”, se obtiene el rol que corresponda al nombre ingresado y ese dato se enviará en la notificación.

Características abarcadas

1. Gestión de instancias por usuario
2. Mantenimiento del histórico de procesos
3. Monitoreo de procesos

B.3.0.4. Caso de prueba 4

Introducción

El objetivo de este caso de prueba es probar la gestión de formularios, por lo que se utilizan dos *user task* con el fin de poder ingresar datos en la primer tarea y luego poder mostrar los datos ingresados en la segunda tarea.

Diagrama del caso

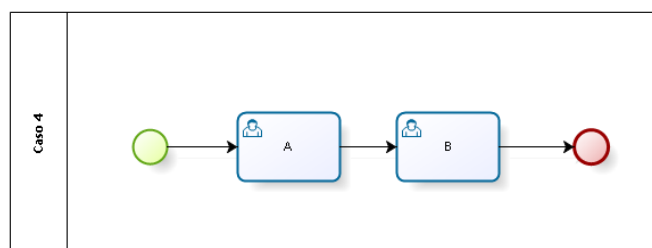


Figura B.4 Modelo del caso de prueba 4

Explicación de la secuencia

1. Inicio: El caso de uso comienza cuando un usuario inicia una instancia del proceso.
2. A: En esta actividad, el usuario que inicia la instancia del proceso ingresa datos en los campos: A (campo de texto, no obligatorio), B (campo numérico, obligatorio) y selecciona un valor de la lista desplegable C (se habilita la lista desplegable D en el formulario) al seleccionar un valor de la lista desplegable C, se filtran los valores mostrados en la lista desplegable D.
3. B: En esta actividad se muestran los valores seleccionados en los campos A, B, C y D de la tarea A y finaliza el proceso.

Características abarcadas

1. Definición de formularios
2. Formato de formularios
3. Elementos de formulario
4. Formulario dinámico
5. Reglas de llenado
6. Validación de datos

B.3.0.5. Caso de prueba 5

Introducción

El objetivo de este caso de prueba es evaluar las características relacionadas con el motor de workflow. Para ello se necesitan al menos dos roles (Rol A y Rol B) para poder asignar y reasignar tareas, además se cuenta con diferentes tipos de tareas y compuertas de bifurcación para evaluar los diferentes tipos de reglas de negocio.

Diagrama del caso

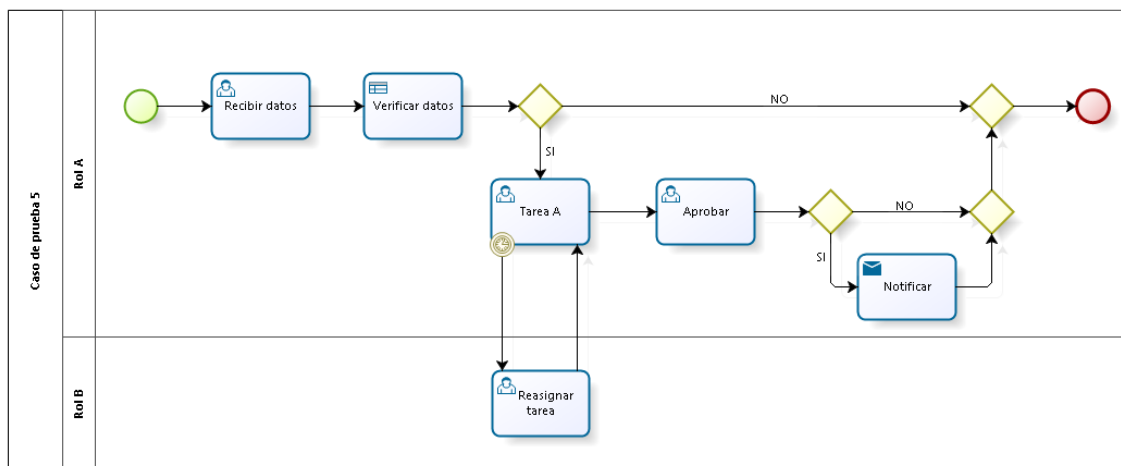


Figura B.5 Modelo del caso de prueba 5

Explicación de la secuencia

1. Inicio: El caso de uso comienza cuando un usuario inicia una instancia del proceso.
2. Recibir datos: En esta actividad, un usuario con “Rol A” ingresa en un formulario, un archivo, un número.
3. Verificar datos: Se ejecuta un script que realiza un calculo con el número ingresado en la actividad *Recibir datos* y devuelve el resultado en una variable. Si el resultado es *true*, se activa la transición SI, en caso contrario se activa la transición NO.

4. Tarea A: Un usuario con “Rol A” tiene un plazo determinado para realizar esta tarea. Pasado ese tiempo, se activa la transición hacia la tarea *Reasignar tarea*.
5. Reasignar tarea: Un usuario con “Rol B”, le asigna la tarea *Tarea A* a un usuario con rol “Rol A”.
6. Aprobar: En esta actividad cuenta con un formulario que contiene un campo de tipo checkbox. Si el usuario con rol “Rol A” selecciona el checkbox, se activa la transición SI, en caso contrario se activa la transición NO.
7. Notificar: Es una actividad automática, donde se envía una notificación.

Características abarcadas

1. Definición de reglas de negocio
2. Asignación de tareas
3. Reasignación de tareas
4. Adjuntar documentos a tareas

B.4. Evaluación de herramientas

En esta sección se muestran los resultados de las evaluaciones. Para realizar la evaluación de cada herramienta se utilizaron los casos de pruebas. Además, una vez finalizada la ejecución del plan de pruebas, se obtuvieron resultados cualitativos, los cuales se transformaron en resultados cuantitativos a través del métodos de evaluación.

Luego de actualizar la lista de características incluyendo criterios no funcionales. Se registraron un total de 109 características, divididas en 15 categorías. Por lo tanto, se acordó con los tutores evaluar un conjunto de estas características incluyendo criterios funcionales y no funcionales. La tabla B.1 muestra las 49 características que fueron seleccionadas para la evaluación.

Tabla B.1 Características a evaluar.

Modulo	Característica	Teórico
Técnico - Funcional		
	Tecnología, arquitectura e interoperabilidad	
	Arquitectura del BPMS	Si
	Compatibilidad con estándares	Si
	Base de datos	Si
	Sistema operativo del modelador	Si
	Navegador Web	Si
	Extensibilidad	Si
	Integración con servicios web	No

Modulo	Característica	Teórico
	Diseño y modelado de procesos	
	Interfaz gráfica para key users	No
	Tipo de modelador	Si
	Notación gráfica de procesos compatible con BPMN 2.0	Si
	Gestión de formularios	
	Definición de formularios	No
	Formato de formularios	No
	Elementos de formularios	No
	Formularios dinámicos	No
	Reglas de llenado de campos	No
	Validación de campos del formularios	No
	Motor de workflow	
	Iniciación de procesos en forma manual y automática	No
	Definición de reglas de negocio	No
	Asignación de tareas	No
	Reasignación de tareas	No
	Adjuntar documentos a tareas	No
	Versionado de procesos	No
	Tareas de borrador	No
	Gestión, monitoreo y auditorias	
	Gestión de instancias por usuarios	No
	Mantenimiento del histórico del proceso	No
	Monitoreo de procesos	No
	Portal	
	Presentación de las tareas	No
Técnico - No funcional		
	Usabilidad	
	Capacidad de aprendizaje	No
	Protección contra errores de usuarios	No
	Estética de la interfaz de usuario	No
	Accesibilidad	No
	Seguridad	
	Confidencialidad	Si
	Integridad	No
	No repudio	No

Modulo	Característica	Teórico
	Responsabilidad	No
	Autenticidad	Si
	Performance	
	Tiempo de respuesta	No
	Throughput	No
	Capacidad	No

B.5. Evaluación de Red Hat JBoss BPM Suite

B.5.1. Evaluación de las características teóricas

B.5.1.1. Arquitectura del BPMS

Red Hat JBoss BPM Suite incluye todas las funcionalidades de JBoss BRMS, con capacidades adicionales de monitoreo de las actividades, inicio de procesos y gestión de tareas desde el Business Central. JBoss BPM Suite también provee un repositorio central para almacenar los procesos y las reglas de negocio.

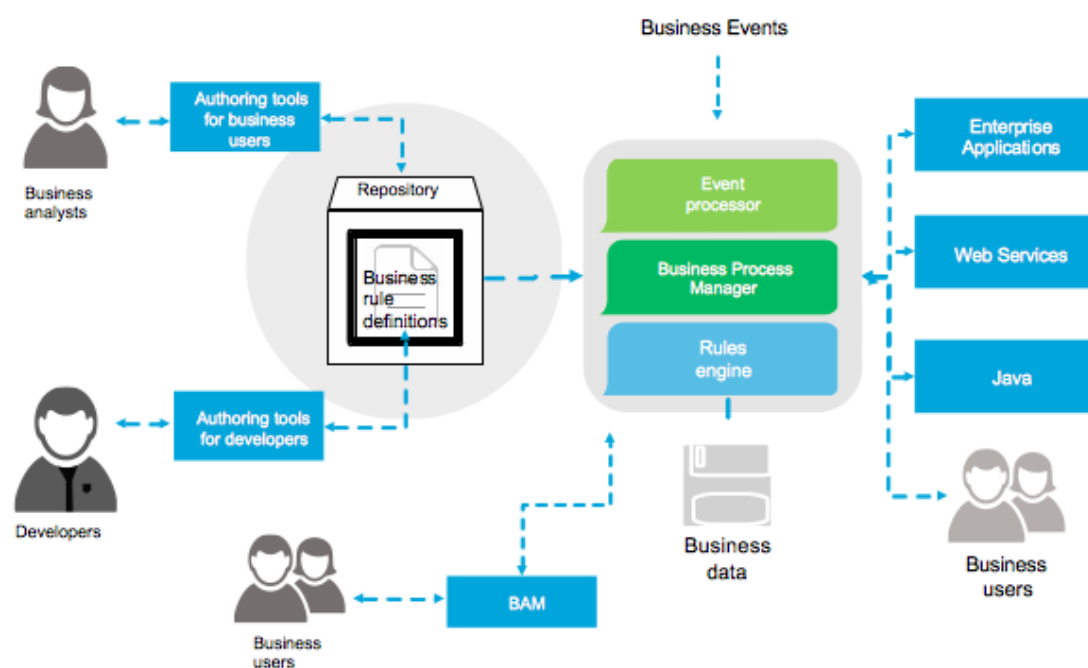


Figura B.6 Arquitectura JBoss BPM Suite [26]

Componentes claves:

JBoss BPM Central (Business Central): es una aplicación web que permite crear, editar, construir, administrar y realizar el seguimiento de los activos empresariales de JBoss BPM Suite. También permite la ejecución de procesos de negocio y la gestión de las tareas creadas por esos procesos.

Business Activity Monitoring (BAM) Dashboards: ofrece capacidades de generación de informes. Permite utilizar un tablero de instrumentos predefinido e incluso crear su propio panel de control personalizado.

Maven Artifact Repository: los proyectos de JBoss BPM Suite se construyen como proyectos Apache Maven y la ubicación predeterminada del repositorio Maven es `<working-directory>/repository`. Se puede especificar otro repositorio cambiando la propiedad `org.guvnor.m2repo.dir`. Cada proyecto genera un archivo JAR llamado `Kjar`. Se pueden almacenar los componentes del proyecto y los jar dependientes en este repositorio.

Execution Engine: el motor de ejecución de JBoss BPM Suite es responsable de la ejecución de los procesos de negocio y la gestión de las tareas que se derivan de estos procesos. El Business Central proporciona una interfaz de usuario para la ejecución de procesos y la gestión de las tareas.

Business Central Repository: los componentes de un proyecto JBoss BPM Suite como modelos de procesos, reglas y formularios son almacenada en repositorios Git gestionados a través del Business Central. También se puede acceder a estos repositorios de manera externa a Business Central, a través de los protocolos git o SSH.

B.5.1.2. Compatibilidad con estándares

Según [43] JBoss BPM Suite permite importar procesos en formatos BPMN2.0, JSON y migrar procesos de JPDL a BPMN2.0; sin embargo solo permite exportar los procesos en formato BPMN2.0.

B.5.1.3. Base de datos

Soporta conexión con las bases de datos:

- H2 Database, es la base que trae predeterminada y no es soportada en producción.
- MS SQL Server (2012)
- MySQL (5.5)
- Oracle (11g R2, 12c)
- PostgreSQL (8.4, 9.2)
- DB2 (9.7, 10.5)

B.5.1.4. Sistema operativo del modelador

El soporte para JBoss BPM Suite es brindado solo para el uso con el sistema operativo Red Hat Enterprise Linux 7 (RHEL7). Sin embargo se puede utilizar en cualquier sistema operativo que soporte alguna de las siguientes JVMs:

- Oracle JDK (1.6, 1.7, 1.8)
- IBM JDK (1.6, 1.7)
- Open JDK (1.6, 1.7, 1.8)
- Zing / Zulu (1.7)

Para la evaluación se utilizó el sistema operativo Windows 10 con Oracle JDK 1.8.

B.5.1.5. Navegador web

Según [44], esta probado que JBoss BPM Suite funciona en los siguientes navegadores:

- Firefox 24
- Firefox 31
- Internet Explorer 10
- Internet Explorer 11

Para esta evaluación se utilizó el navegador:

- Google Chrome 46

B.5.1.6. Tipo de modelador

JBoss BPM Suite cuenta con un modelador web, también provee la opción de utilizar un modelador de escritorio mediante la utilización de JBoss Developer Studio y plugins específicos.

B.5.1.7. Notación gráfica de procesos compatible con BPMN 2.0

JBoss BPM Suite es compatible con el estándar BPMN 2.0.

B.5.2. Evaluación de las características prácticas

B.5.2.1. Integración con servicios web

La herramienta permite la integración con servicios Web externos. El modelador cuenta con un tipo de tarea “WS Task” la cual cuenta con todos los parámetros necesarios para poder comunicarse con un servicio Web. Los estándares soportados para servicios web son SOAP y REST [45].

En la evaluación se utilizó una conexión tipo SOAP.

B.5.2.2. Interfaz gráfica para key users

La herramienta cuenta con un modelador web el cual permite el modelado de procesos a partir de una interfaz gráfica amigable. El modelador permite el desarrollo, visualización, modificación de los diagramas de flujo y de los atributos de cada elemento del modelo. En la evaluación se pudo ver que el modelador cuenta con algunos problemas referidos a la resolución de la pantalla, que en algunas ocasiones dificultan la utilización del mismo,

pero no la impiden. Si se requiere de algún tipo de datos que no es implementado por el modelador o para la configuración de tareas de Servicios o Script es necesario de un usuario con formación técnica.

B.5.2.3. Definición de formularios

JBoss BPM Suite cuenta con el Form Modeler que permite la definición de formularios de forma dinámica. A su vez si en la tarea de usuario se definen las asignaciones de variables y sus tipos y cuales son variables de entradas y de salidas, entonces también permite la creación automática del formulario sin necesidad de intervención del usuario. Form Modeler no cuenta con la funcionalidad de drag and drop, pero si se realiza clic en alguno de los componentes de la paleta el mismo se incluye al formulario. El modelador permite definir reglas de llenado de formularios.

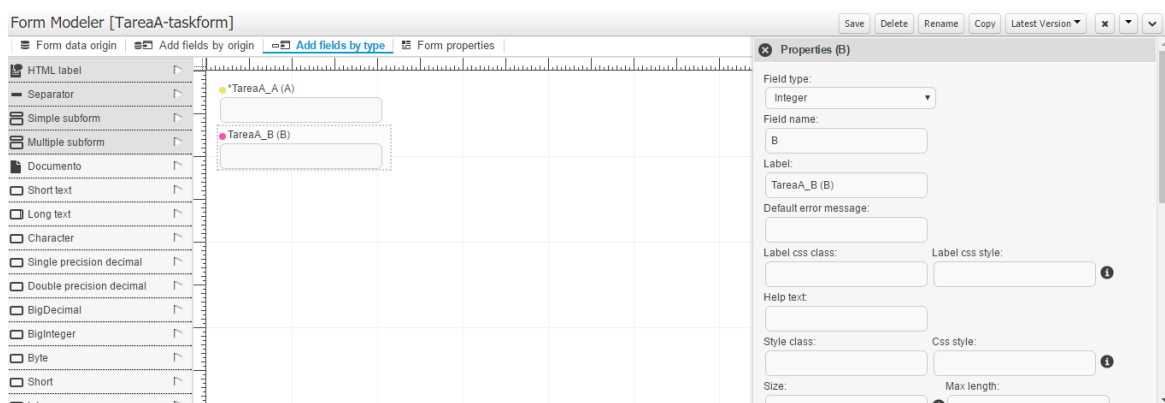


Figura B.7 Definición de Formularios JBoss BPM Suite

B.5.2.4. Formato de formularios

Permite configurar distintos elementos de un formulario a través de la definición de estilos CSS tanto para las etiquetas como para los campos. Posee la capacidad de adjuntar plantillas de formularios customizados [46]. También permite definir la alineación de los campos y brinda diferentes opciones de alineaciones predefinidas.

B.5.2.5. Elementos del formulario

Permite incluir listas desplegables donde los datos de las listas son tomados de Data Providers que son previamente definidos. No permite definir botones que realicen acciones.

B.5.2.6. Formularios dinámicos

La herramienta permite ingresar formulas en los campos de los formularios mediante la cual se calcule el valor del campo y en las cuales se pueden usar los valores de variables del proceso, valores por defecto, y permite definir acciones asociadas al cambio en el estado del campo (mediante javascript o jquery). También se pueden asociar a la tarea variables de entradas que son mapeadas a valores de otras variables o a un valor fijo, y definir variables de

salida las cuales también pueden ser mapeadas a un valor fijo o a una variable de otra tarea o del proceso.

B.5.2.7. Reglas de llenado de campos

Además de lo definido en “Formularios dinámicos” los formularios también permiten definir patrones que deben seguir los datos ingresados en el campo (por ejemplo expresiones regulares), además permite ingresar una fórmula que defina el rango de valores que puede tomar el campo.

B.5.2.8. Validación de campos de formularios

Los campos son validados según su tipo de dato y según sean o no requeridos. Se muestra un mensaje de error indicando cuando un campo requerido no fue completado y solo marca en rojo las etiquetas de los campos cuando no cumplen con el formato requerido.



Figura B.8 Validación de campos de Formularios JBoss BPM Suite

B.5.2.9. Iniciación de procesos en forma manual y automática

Los procesos pueden ser iniciados mediante servicios REST. Se pueden definir tareas para usuarios y trabajos pero no se pueden asociar a un proceso.

B.5.2.10. Definición de reglas de negocio

JBoss BPM Suite está integrado con JBoss BRMS que es un sistema de gestión de reglas de negocio. Existe una tarea de regla de negocio disponible en el proceso a la cual se le asocia un archivo .drl que implementa la regla. La herramienta permite crear un nuevo archivo drl y editarlo, crear tablas de decisión mediante templates o crear las reglas directamente mediante templates. En las reglas se pueden utilizar variables del proceso y se pueden basar en plazos o combinaciones de las anteriores.

B.5.2.11. Asignación de tareas

Permite asignar tareas a usuarios y a grupos de usuarios. Solo permite la asignación manual de las tareas.

B.5.2.12. Reasignación de tareas

Se pueden definir reasignación de tareas definiendo una fecha en la que la tarea expira y el estado de la tarea (no empezada, no completada) o de forma manual el usuario puede reasignar la tarea ingresando el nombre del usuario al cual se le será asignada.

B.5.2.13. Adjuntar documentos a tareas

Se pueden adjuntar documentos a las tareas definiendo un campo de tipo documento en la misma. El documento adjunto es solo para esa instancia de la tarea.

B.5.2.14. Versionado de procesos

Existe la posibilidad de que existan varias versiones activas de un proceso. No permite volver a desplegar la misma versión de un proceso si el mismo tiene instancias activas. En este caso se deben eliminar todas las instancias activas y borrar el despliegue anterior o generar una nueva versión del proceso. Si hay instancias activas y se sube una nueva versión, las anteriores siguen funcionando según estaban definidas en su versión.

B.5.2.15. Tareas de borrador

El portal permite, luego de asignarse una tarea, guardarla para retomarla luego. Se guarda la información ya ingresada.

B.5.2.16. Gestión de instancias por usuarios

El portal permite crear, pausar o reanudar, reinstalar, borrar y abortar instancias de un proceso, pero no permite modificar los datos ingresado en las tareas ya completadas.

B.5.2.17. Mantenimiento de histórico del proceso

Existe un histórico de procesos en el cual se encuentra información de los procesos activos, completos y abortados. Se muestra el id del proceso, el nombre, la descripción, quien inicio el proceso, la fecha en la que se inicio el mismo y el estado. Además permite ver información más específica del proceso realizando clic en el mismo. Luego también se puede ver el histórico de tareas donde se muestra el id, el nombre, la descripción, la prioridad, el estado, la fecha de iniciada y la fecha de finalizada, además si se realiza clic en la misma se puede ver información más específica.

B.5.2.18. Monitoreo de procesos

Para el monitoreo y gestión de instancias y procesos la herramienta cuenta con un módulo de monitoreo. También provee un dashboard, en el cual se pueden agregar diferentes widget que son configurables y muestra datos de un proceso, las instancias de un proceso o de las tareas respecto a un usuario, a un rol o al estado de la misma. Permite obtener diferentes reportes según la aplicación de diferentes filtros tales como: fecha de finalización de proceso

o tareas, estado de las tareas o los procesos. Los resultados se muestran en tablas y en gráficos y los datos de los reportes se pueden exportar a CSV o XLS.

B.5.2.19. Presentación de las tareas

El portal cuenta con una página Home en la cual se detallan las funcionalidades de la herramienta a modo de resumen. Para visualizar las tareas que tiene un usuario, el mismo tiene que dirigirse a la pestaña “Task” y luego “Task List” donde podrá ver el historial de las tareas y sus tareas pendientes. En la pantalla de tareas hay diferentes vistas según las tareas Activas, Personales, Grupos, Todas y en caso de ser usuario administrador tiene las tareas de Admin. En cada una de estas vistas se puede ordenar por prioridad, nombre, identificador, fecha.

B.5.3. Evaluación de las características no funcionales

B.5.3.1. Capacidad de aprendizaje

JBossBPM Suite brinda funcionalidades tanto en el modelador como en el portal que facilitan el aprendizaje de los mismos. El modelador cuenta con tooltips en todos los elementos de la paleta brindando una breve descripción de la funcionalidad del elemento, en caso de que el mismo no sea demasiado intuitivo. El modelador permite la validación del modelo donde en caso de existir errores se marcan las tareas con posibles errores y al realizar clic en una de ellas se muestran los errores y sugerencias de posibles soluciones. Tanto el modelador como el portal web poseen manuales de uso pero para acceder a los mismos se debe ingresar a la página de jboss bpm suite, los accesos directos no te dirigen a los manuales.

Tabla B.2 Evaluación - Capacidad de aprendizaje

	Modelador	Portal
¿Posee tooltips?	SI	SI
¿Tiene mensajes de error descriptivos?	SI	SI
¿Brinda sugerencia de posible solución ante un error?	SI	NO
¿Posee manuales de uso?	SI	SI

B.5.3.2. Capacidad para ser usado

JBoss BPM Suite brinda un menú de ayuda tanto en el modelador como en el portal y los símbolos son mnemotécnicos pero en algunos casos las acciones están ubicadas en lugares poco intuitivos lo que puede ocasionar confusión para el usuario.

Sin embargo no se cuenta con un espacio de preguntas frecuentes en ninguno de los dos y los accesos directos a los manuales de uso no funcionan correctamente por lo que no son accesibles de forma fácil.

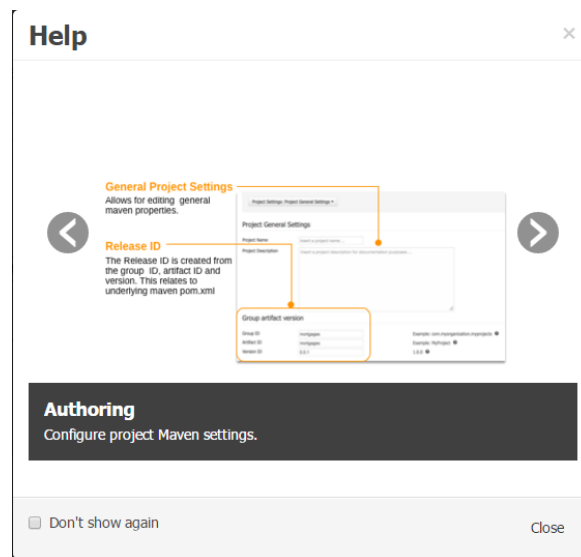


Figura B.9 Ayuda de JBoss BPM Suite

Tabla B.3 Evaluación - Capacidad para ser usado

	Modelador	Portal
¿Brinda menú de ayuda?	SI	SI
¿Tiene símbolos mnemotécnicos?	SI	SI
¿Brinda un espacio de preguntas frecuentes?	NO	NO
¿Posee acceso directo a manuales de uso?	NO	NO

B.5.3.3. Protección contra errores de usuarios

Tanto en el modelador como en el portal se muestra al usuario mensajes de advertencia antes de ejecutar una acción. En ambos se muestran mensajes para avisar al usuario cuando una acción se ejecuto de forma correcta o incorrecta.

Del mismo modo se muestran mensajes de advertencia cuando se ingresan datos incorrectos en los formularios tanto desde el modelador como desde el portal.

El modelador cuenta con opciones para deshacer acciones mientras que el portal no ofrece dicha funcionalidad.

Tabla B.4 Evaluación - Protección contra errores de usuarios

	Modelador	Portal
¿Tiene mensajes de advertencia?	SI	SI
¿Posee validación de datos y formularios?	SI	SI
¿Cuenta con opciones para deshacer y rehacer una acción?	SI	NO

Figura B.10 Mensaje de validación de JBoss BPM Suite

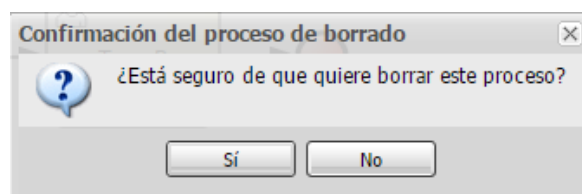


Figura B.11 Mensaje de advertencia de JBoss BPM Suite

B.5.3.4. Estética de la interfaz de usuario

La interfaz de usuario del modelador y del portal posee los elementos relevantes y no incluye información innecesaria. La información y las acciones importantes son de fácil acceso. Algunas acciones secundarias se encuentran en lugares poco intuitivos.

Tabla B.5 Evaluación - Estética de la interfaz de usuario

	Modelador	Portal
¿Incluye información innecesario?	NO	NO
La información relevante ¿Se encuentra accesible y fácil de encontrar?	SI	SI

B.5.3.5. Accesibilidad

El portal web y el modelador no incluyen comandos de voz para representar acciones. Las acciones son representadas mediante imágenes adecuadas y los colores son distinguibles. Se utiliza el color rojo para los mensajes de error y el verde para los mensajes de éxito. Las prioridades también son representadas mediante colores siendo el rojo prioridad alta, el verde prioridad baja y el amarillo prioridad media.

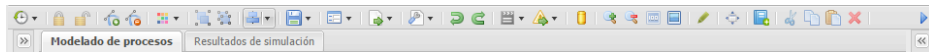


Figura B.12 Opciones para deshacer o rehacer acciones de JBoss BPM Suite

Tabla B.6 Evaluación - Accesibilidad

	Modelador	Portal
¿Incluye comando de voz?	NO	NO
¿Las acciones se encuentran representadas con imágenes adecuadas?	SI	SI
¿Los colores son distinguibles aunque el usuario sea daltónico?	SI	SI
¿Los sonidos se corresponden con la acción que se está ejecutando?	NO	NO

B.5.3.6. Confidencialidad

JBoss BPM Suite posee una técnica criptográfica para almacenar las contraseñas . No cuenta aún con una aplicación mobile. El intercambio de datos con el servidor no es sobre HTTPS.

Tabla B.7 Evaluación - Confidencialidad

	Portal
¿Utiliza técnicas criptográficas?	SI
Mobile ¿Almacena los datos cifrados en el dispositivo?	NO
¿Los intercambios de datos con el servidor son sobre HTTPS?	NO

B.5.3.7. Integridad

La creación de usuarios en JBoss BPM Suite se puede realizar al momento de la instalación. Luego es necesario definir los usuarios mediante la ejecución de un archivo ejecutable. La herramienta provee la funcionalidad del acceso diferenciado según el rol del usuario. El acceso a los documentos es diferenciado según el rol, pero los mismos no se guardan cifrados en la base de datos. La sesión no expira, por lo que permanece activa hasta que el usuario decida cerrarla.

Tabla B.8 Evaluación - Integridad

	Portal
¿Permite definir roles?	SI
¿Cuenta con acceso diferenciado a los documentos según el rol?	SI
¿Permite visualizar diferentes funciones según el rol?	SI
¿Permite administrar permisos?	SI
¿Los documentos se guardan cifrados?	NO
¿Cuenta con un tiempo limite de sesión activa?	NO

B.5.3.8. No repudio

JBoss BPM Suite posee un historial de ejecución del proceso además permite identificar que usuario y en que momento realizó determinada acción.

Tabla B.9 Evaluación - No repudio

	Portal
¿Cuenta con un historial de ejecución de procesos?	SI
¿Se puede identificar que usuario y en que momento se ejecutó un proceso?	SI

B.5.3.9. Responsabilidad

Las tareas y procesos de la herramienta cuentan con un identificador numérico que es único.

Tabla B.10 Evaluación - Responsabilidad

	Portal
¿Las tareas y los procesos tienen identificador único?	SI

B.5.3.10. Autenticidad

JBoss BPM Suite cuenta con autenticación de los usuarios para el ingreso al portal y al modelador. La herramienta no cuenta con soporte para firmas digitales.

Tabla B.11 Evaluación - Autenticidad

	Portal
¿Cuenta con alguna forma de autenticación de usuarios?	SI
¿Soporta firmas digitales?	NO

B.5.3.11. Tiempo de respuesta

No fue posible evaluar el tiempo de respuesta para esta herramienta. La causa fue que no se logró iniciar sesión en la aplicación desde la herramienta de performance. En las pruebas realizadas, se lograba realizar la grabación del script, el cual, inicia sesión en la herramienta y luego ejecuta el proceso. Esta grabación se realizó con éxito, pero al momento de la ejecución del test, el mismo no se ejecutaba correctamente e indicaba que el problema se debía a que la sesión expiraba. Una de las medidas tomadas fue obtener el id de sesión que se generaba al momento de la autenticación y luego usarlo en las llamadas del script. Se logró obtener el id de sesión pero de todas maneras las pruebas continuaron fallando, indicando como causa del error que la sesión se encontraba expirada. No se logró identificar la causa de este error, por lo que fue imposible ejecutar las pruebas para esta herramienta.

B.5.4. Tabla de nivel de cumplimiento

La tabla B.12 muestra los resultados obtenidos para las características funcionales. Las características están divididas por categoría. De cada característica se muestra si la evaluación es teórica (Teó), el nivel de soporte (NS) y el nivel de cumplimiento (NC).

Tabla B.12 Características técnicas - Nivel de cumplimiento.

Modulo	Característica	Teó	NS	NC
Técnico				
	Tecnología, Arquitectura e Interoperabilidad			
	Arquitectura de BPMS	Si	⊗	⊗
	Compatibilidad con estándares	Si	●	✓
	Base de datos	Si	●	✓
	Sistema operativo del modelador	Si	●	✓
	Navegador Web	Si	●	✓
	Extensibilidad	Si	●	✓
	Integración con Servicios Web	No	●	✓
	Diseño y modelado de procesos			
	Interfaz gráfica para key users	No	●	✓
	Tipo de modelador	Si	●	✓
	Notación gráfica de procesos compatible con BPMN 2.0	Si	●	✓
	Gestión de formularios			
	Definición de formularios	No	●	✓
	Formato de formularios	No	●	✓
	Elementos de formularios	No	●	✓

Modulo	Característica	Teó	NS	NC
	Formularios dinámicos	No	●	✓
	Reglas de llenado de campos	No	●	✓
	Validación de datos del formulario	No	●	✓
	Motor de Workflow			
	Iniciación de procesos en forma manual y automática	No	●	✓
	Definición de reglas de negocio	No	●	✓
	Asignación de tareas	No	●	✓
	Reasignación de tareas	No	●	✓
	Adjuntar documentos a tareas	No	●	✓
	Versionado de procesos	No	●	✓
	Tareas de borrador	No	●	✓
	Gestión, monitoreo y auditoría			
	Gestión de instancias por usuarios	No	●	✓
	Mantenimiento histórico de procesos	No	●	✓
	Monitoreo de procesos	No	●	✓
	Portal			
	Presentación de las tareas	No	●	✓

Referencias:

- ⊘ No aplica
- ✓ Cumple de forma nativa
- ⚙ Cumple vía integración
- ✂ Cumple vía particularización
- Soporta totalmente
- Soporta parcialmente
- No soportada

B.5.4.1. Resultado numérico de las características técnicas

Las tablas B.14 y B.15 presentan los resultados numéricos correspondientes a las características funcionales y no funcionales evaluadas. En la sección 3.7 se describe el método de evaluación, donde se presentan los cálculos a realizar para poder clasificar a las características según los umbrales definidos. Primero, hay que definir los valores de importancia, para poder definir los umbrales. Además, es necesario dar valores a los niveles de cumplimiento y de soporte.

Tabla B.13 Definición de constantes

Importancia		Nivel de cumplimiento		Nivel de soporte	
Alto	3	Nativo	2	Total	3
Medio	2	Vía integración	1	Parcial	2
Bajo	1	Vía particularización	1	No soportado	1

Las columnas de la tabla B.14 contienen: las características, importancia (I), nivel de soporte (NS), nivel de cumplimiento (NC), el factor (F) y el puntaje total (Total). El puntaje total se calcula de la siguiente manera:

Puntaje total

$$Puntaje = I * NS * NC * Factor \quad (B.3)$$

Donde I es la importancia, NS el nivel de soporte, NC el nivel de cumplimiento y Factor un valor entre 0,1 y 0,9.

Tabla B.14 Características técnicas - Calculo del puntaje.

Característica	I	NS	NC	F	Total
Tecnología, Arquitectura e Interoperabilidad					
Compatibilidad con estándares	1	1	2	1	2
Base de datos	1	2	2	1	4
Sistema operativo del modelador	1	2	2	1	4
Navegador Web	1	2	2	1	4
Extensibilidad	1	2	2	1	4
Integración con Servicios Web	1	1	1	0,3	0,3
Diseño y modelado de procesos					
Interfaz gráfica para key users	1	2	2	1	4
Tipo de modelador	1	2	2	1	4
Notación gráfica de procesos compatible con BPMN 2.0	1	2	2	1	4
Gestión de formularios					
Definición de formularios	1	2	2	1	4
Formato de formularios	1	1	2	1	2
Elementos de formularios	1	1	2	1	2
Formularios dinámicos	1	2	2	1	4
Reglas de llenado de campos	1	1	2	1	2

Característica	I	NS	NC	F	Total
Validación de datos del formulario	1	2	2	1	4
Motor de Workflow					
Iniciación de procesos en forma manual y automática	1	1	2	1	2
Definición de reglas de negocio	1	2	1	0,1	1
Asignación de tareas	1	2	2	1	4
Reasignación de tareas	1	2	2	1	4
Adjuntar documentos a tareas	1	2	2	1	4
Versionado de procesos	1	2	2	1	4
Tareas de borrador	1	0	0	1	0
Gestión, monitoreo y auditoría					
Gestión de instancias por usuarios	1	1	2	1	2
Mantenimiento histórico de procesos	1	2	2	1	4
Monitoreo de procesos	1	0	0	1	0
Portal					
Presentación de las tareas	1	1	2	1	2

Por otro lado tenemos los cálculos de las características no funcionales. Las columnas de la tabla B.15 contienen: el peso de cada subcaracterística (P_i), el nivel de cumplimiento de la subcaracterísticas (s_i), el valor final de la subcaracterística definido en la ecuación B.18 y el nivel de cumplimiento (NC). El nivel de cumplimiento se obtiene de la siguiente manera:

- $vf < 1 \rightarrow$ Nivel de cumplimiento Bajo.
- $1 \leq vf < 2 \rightarrow$ Nivel de cumplimiento Medio.
- $3 \leq vf \rightarrow$ Nivel de cumplimiento Alto.

Valor final

$$vf = \frac{\sum_{i=1}^N (s_i * P_i)}{\sum_{i=1}^N P_i} \quad (B.4)$$

Siendo vf el 'valor final' de la característica y N la cantidad de subcaracterísticas.

Se considera que todas las subcaracterísticas son importantes y en este sentido se asigna el valor 1 al peso de cada subcaracterísticas.

B.6. Evaluación de Bonita BPM

La versión utilizada para la evaluación es la *Community 7.1* cuyos componentes se obtuvieron mediante descarga en la página de la herramienta.

B.6.1. Evaluación de las características teóricas

B.6.1.1. Arquitectura del BPMS

El motor de Bonita Suite esta compuesto por API's y servicios. A su vez los servicios se pueden dividir en servicios genéricos y servicios específicos de BPM.

Este motor es completamente configurable desde un archivo XML. En este archivo se encuentran todos los servicios usados por defecto, así Bonita permite cambiar o reemplazar la implementación de estos servicios.

La instalación del motor se puede hacer en cualquier JVM, contenedor JEE o puede ser usado como una librería JAVA.

Dentro del módulo BPM Services se debe destacar los conectores que son lo que permiten al motor la interacción con una base de datos, con diferentes servidores de correo, acceso a un LDAP para la autenticación de usuarios y otros sistemas que se pueden ver en [27]. Además de los conectores este modulo se encarga del manejo de los procesos, usuarios y tareas.

En el módulo de Servicios Genéricos encontramos las funcionalidades de autenticación, manejo de eventos y persistencia entre otros.

Por último para el armado del motor de bpm de bonita tenemos el módulo de api, mediante ésta el motor interactúa con el portal y aplicaciones desarrolladas por usuarios.

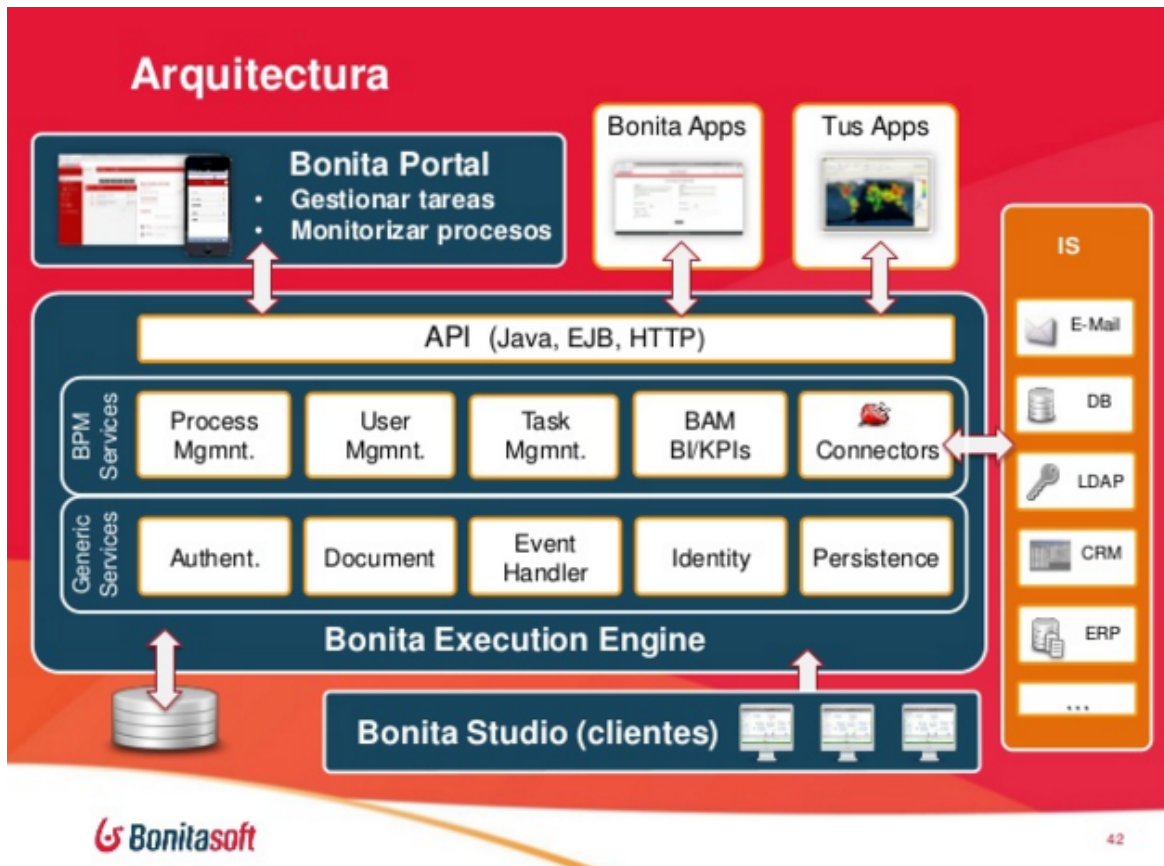


Figura B.13 Arquitectura Bonita [18]

B.6.1.2. Compatibilidad con estándares

Se puede importar/exportar procesos de las siguiente herramientas:

- Bonita BPM
- BPMN 2.0
- Microsoft Visio 2010
- ARIS BPM 7.x
- XPD L 1.0
- jBPM 3.2

B.6.1.3. Base de datos

Soporta conexión con:

- H2 Database, base predeterminada.
- Microsoft SQL Server (2012)
- MySQL (5.5)
- Oracle Database (11gR2)
- PostgreSQL (9.3)

Bases de datos recomendadas:

- PostgreSQL 9.3
- Oracle database 11gR2

B.6.1.4. Sistema operativo del modelador

No requiere ningún sistema operativo en particular, la única condición es que el sistema elegido sea compatible con alguna de las siguiente JVMs:

- Oracle Java SE JRE 7
- OpenJDK 7

Sistemas operativos recomendados:

- Ubuntu 14.04 LTS 64 bits
- Red Hat Enterprise Linux 6
- Windows Server 2012 R2

B.6.1.5. Navegador web

Soporta los navegadores:

- Internet Explorer 9 o superior
- Mozilla Firefox
- Google Chrome

B.6.1.6. Extensibilidad

Bonita BPM ofrece un rico conjunto de características por defecto. Estas características están diseñadas para satisfacer las necesidades de la mayoría de sus proyectos. Sin embargo, si un proyecto tiene una necesidad que no estaba prevista y no puede ser satisfecha por las características predeterminadas, Bonita BPM ha sido diseñado para ser extensible. A continuación se muestran las extensiones disponibles:

- Connectors
- Actor filters
- Engine API
- REST API
- Custom pages
- Custom widgets
- Import and export exchange files
- Authentication service
- Event handlers
- BonitaStudioBuilder
- Portal look & feel
- Portal language pack
- Custom data types

B.6.1.7. Tipo de modelador

Bonita cuenta con una aplicación de escritorio para modelar los procesos.

B.6.1.8. Notación gráfica de procesos compatible con BPMN 2.0

Según la documentación Bonita es compatible con BPMN 2.0

B.6.2. Evaluación de las características prácticas**B.6.2.1. Integración con servicios web**

Mediante conectores se pueden hacer invocaciones a web services. Dichos conectores se deben de adjuntar a las tareas. Según la documentación soporta llamados de tipo SOAP y REST.

B.6.2.2. Interfaz gráfica para key users

No es necesario programar para diseñar los procesos. La mayoría de la acciones tienen un wizard asociado para su configuración sin la necesidad de programar. Se puede agregar lógica de programación en caso de ser necesario por complejidad del proceso.

B.6.2.3. Definición de formularios

Bonita cuenta con la funcionalidad de definir formularios para las tareas que sean necesarios. Cuenta con una interfaz amigable que permite el diseño fácil y rápido por parte del usuario. Presenta dificultades para acceder desde el modelador. Por otra parte, tiene reglas de validación para el llenado de campos. Las ultimas versiones cuentan con un UI designer que es una herramienta web para la creación de formularios y páginas. Para el diseño de estos formularios se necesitan conocimientos de programación.

B.6.2.4. Formato de formularios

En los formularios de Bonita se puede configurar diferentes formatos (fuente, color, alineación, etc). No cuenta con plantillas de formularios.

B.6.2.5. Elementos del formulario

En los formularios que permite definir Bonita se pueden incluir listas desplegables con datos obtenidos de una base de datos. Bonita en esta versión presenta el UI designer para formularios y páginas.

B.6.2.6. Formularios dinámicos

Bonita permite los formularios dinámicos, que un valor dependa de otro elegido. Además se pueden ocultar los campos dependiendo de otro.

B.6.2.7. Reglas de llenado de campos

Hay cuatro formas diferentes de inicialización de los controles:

- Constante: Se inicia con el valor por defecto definido
- Script: Script groovy para inicializar
- Entrada de contrato: Se inicia con el valor de una variable del proceso
- Parámetros

B.6.2.8. Validación de campos de formularios

Los validadores que posee Bonita son los encargados de llevar a cargo esta tarea. Estos se activan una vez que se hace submit de un formulario. Para su implementación es necesario definir un tipo de Validador, estos se describen en la documentación de Bonita. El tipo de Validador, estos se describen en la documentación de Bonita. Además Bonita hace la validación automática de los campos marcados como requeridos.

B.6.2.9. Iniciación de procesos en forma manual y automática

Bonita permite iniciar procesos tanto de forma manual desde el portal como desde su API. Para esto provee el método `startProcess` que tiene como parámetro el id del proceso que está deployado. Al ejecutarse este método se genera una nueva instancia del proceso.

B.6.2.10. Definición de reglas de negocio

Bonita permita a través de expresiones Groovy definir reglas de negocio. Estas expresiones se pueden incluir en los Validadores nombrados anteriormente. Además se pueden agregar en conectores, en scripts y en compuertas de decisión de flujo.

B.6.2.11. Asignación de tareas

Al momento de asignar una tarea, Bonita permite que esta asignación sea a un rol, a un grupo o a un usuario. La asignación puede ser manual o automática. Para la asignación automática se pueden tener en cuenta las diferentes variantes en ejecución del proceso, por ejemplo los valores de las variables.

B.6.2.12. Reasignación de tareas

Teniendo el rol correspondiente un usuario puede retirarle la asignación de una tarea a un usuario y asignársela a otro usuario.

B.6.2.13. Adjuntar documentos a tareas

Bonita permite el manejo de documentos. Estos se deben de definir por cada pool del proceso. Se puede adjuntar archivos locales y externos. Además se puede definir un tamaño máximo para estos.

B.6.2.14. Tareas de borrador

Bonita permite guardar el estado de una tarea con todos los datos ingresados. Esto permite que al volver a retomar la tarea pendiente, el usuario no tiene que volver a ingresar todos los datos nuevamente, sino que parte desde lo que guardó la ultima vez.

B.6.2.15. Gestión de instancias por usuarios

Todos los usuarios son capaces de iniciar instancias de los procesos que tienen disponibles en el portal. Para pausar o reanudar, borrar, reinstalar o abortar instancias del proceso.

B.6.2.16. Mantenimiento del histórico del proceso

Los usuarios con rol administrador tienen acceso al monitoreo de instancias y procesos, esto les permite ver el histórico de los procesos y sus instancias.

De los procesos se puede ver, la versión y quién realizó el deploy, el estado (activo/inactivo) y, un resumen de las instancias de esos procesos (cuántas están en progreso, completadas, terminadas y cuántas fallaron). De las instancias se puede ver, el identificador, el proceso al que corresponden, el estado, fecha de comienzo, datos de las actividades de la instancia y un histórico de los eventos. Para visualizar el usuario que realizó una actividad, es necesario ingresar a los datos de las actividades.

Se puede ver los documentos que se adjuntaron a las instancias.

B.6.2.17. Monitoreo de procesos

Teniendo el rol administrador, se pueden ver los procesos que están deployados, las instancias que se generaron de cada uno de ellos y en qué tarea del proceso se encuentra cada instancia.

No cuenta con generación de reportes en algún formato específico.

B.6.2.18. Presentación de las tareas

Al ingresar al portal de Bonita, se visualizan las tareas pendientes que tiene el usuario, preseleccionada la primera tarea. Las tareas se ordenan por fecha límite para su realización, estando primera la que queda menos tiempo para realizarla, o la en el caso de que este pasada dicha fecha, queda primera la que tiene mayor tiempo desde que caducó.

B.6.3. Evaluación de las características no funcionales

B.6.3.1. Capacidad de aprendizaje

Bonita BPM cuenta con tutoriales tanto en pdf como videos. Si bien los pdf están solo en inglés, los videos se pueden ver en 7 idiomas (inglés, alemán, francés, japonés, español, portugués e italiano). Con esto se logra un aprendizaje eficaz, logrando en poco tiempo poder modelar y ejecutar un proceso.

Tabla B.16 Evaluación - Capacidad de aprendizaje

	Modelador	Portal
¿Posee tooltips?	SI	SI
¿Tiene mensajes de error descriptivos?	SI	SI
¿Brinda sugerencia de posible solución ante un error?	SI	NO
¿Posee manuales de uso?	SI	SI

B.6.3.2. Capacidad para ser usado

Bonita BPM cuenta con un menú con símbolos mnemotécnicos y Además un título en cada opción. En la paleta, si se tiene el mouse sobre algún objeto, se muestra un tooltip con la descripción o funcionalidad asociada a ese objeto.

En el portal al igual que en el designer se tiene cada opción con un símbolo mnemotécnico asociado y un título.

Tabla B.17 Evaluación - Capacidad para ser usado

	Modelador	Portal
¿Brinda menú de ayuda?	SI	NO
¿Tiene símbolos mnemotécnicos?	SI	SI
¿Brinda un espacio de preguntas frecuentes?	NO	NO
¿Posee acceso directo a manuales de uso?	SI	NO

B.6.3.3. Protección contra errores de usuarios

Cuando ocurre algún problema con el modelo, se muestra un triángulo amarillo donde está el problema, con un mensaje de advertencia asociado como tooltip. Se puede utilizar Ctrl + Z y Ctrl + Y, para deshacer y rehacer una acción.

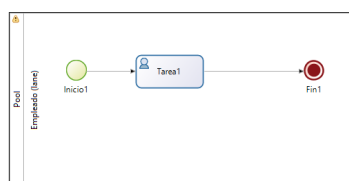


Figura B.14 Símbolo de advertencia

Tabla B.18 Evaluación - Protección contra errores de usuarios

	Modelador	Portal
¿Tiene mensajes de advertencia?	SI	SI
¿Posee validación de datos y formularios?	SI	SI
¿Cuenta con opciones para deshacer y rehacer una acción?	SI	NO

B.6.3.4. Estética de la interfaz de usuario

La información que se incluye tanto en el portal como en la aplicación de diseño no es innecesaria. Todo lo relevante se encuentra de manera accesible.

Tabla B.19 Evaluación - Estética de la interfaz de usuario

	Modelador	Portal
¿Incluye información innecesario?	NO	NO
La información relevante ¿Se encuentra accesible y fácil de encontrar?	SI	SI

B.6.3.5. Accesibilidad

Las acciones tanto en el modelador como en el portal se representan con imágenes adecuadas. Los colores y los símbolos también son adecuados en ambas aplicaciones.

Ni el modelador, ni el portal cuentan con comando de voz o utilizan sonido alguno.

Tabla B.20 Evaluación - Accesibilidad

	Modelador	Portal
¿Incluye comando de voz?	NO	NO
¿Las acciones se encuentran representadas con imágenes adecuadas?	SI	SI
¿Los colores son distinguibles aunque el usuario sea daltónico?	SI	SI
¿Los sonidos se corresponden con la acción que se está ejecutando?	NO	NO

B.6.3.6. Confidencialidad

Las contraseñas se guardan cifradas. Algunos intercambios son http y otros https.

Tabla B.21 Evaluación - Confidencialidad

	Portal
¿Utiliza técnicas criptográficas?	SI
Mobile ¿Almacena los datos cifrados en el dispositivo?	NO
¿Los intercambios de datos con el servidor son sobre HTTPS?	NO

B.6.3.7. Integridad

El usuario con permiso de administrador puede crear roles y asignar los usuarios a uno o mas roles. Según el rol que tenga el usuario podrá ver ciertas funcionalidades o no. Los documentos no se guardan cifrados. Bonita no cuenta con limite de sesión activa.

Tabla B.22 Evaluación - Integridad

	Portal
¿Permite definir roles?	SI
¿Cuenta con acceso diferenciado a los documentos según el rol?	SI
¿Permite visualizar diferentes funciones según el rol?	SI
¿Permite administrar permisos?	SI
¿Los documentos se guardan cifrados?	NO
¿Cuenta con un tiempo limite de sesión activa?	NO

B.6.3.8. No repudio

Bonita cuenta con un historial de ejecución de procesos en el cual se puede ver que usuario y en que momento se ejecutó un tarea.

Tabla B.23 Evaluación - No repudio

	Portal
¿Cuenta con un historial de ejecución de procesos?	SI
¿Se puede identificar que usuario y en que momento se ejecutó un proceso?	SI

B.6.3.9. Responsabilidad

Las tareas y los procesos cuentan con un identificador numérico que es único.

Tabla B.24 Evaluación - Responsabilidad

	Portal
¿Las tareas y los procesos tienen identificador único?	SI

B.6.3.10. Autenticidad

Bonita provee su propia forma de autenticación, pero también permite integral el sistema con bases externas como LDAP.

Tabla B.25 Evaluación - Autenticidad

	Portal
¿Cuenta con alguna forma de autenticación de usuarios?	SI
¿Soporta firmas digitales?	NO

B.6.3.11. Tiempo de respuesta

Para determinar el tiempo de respuesta se realizaron varias ejecuciones sobre un conjunto de test. El gráfico de la figura B.15 muestra el crecimiento del tiempo de respuesta, en función de la cantidad de usuarios. El gráfico azul corresponde a los valores obtenidos luego de la ejecución, mientras que la función naranja indica la tendencia. La tendencia corresponde a un polinomio de segundo grado, esto quiere decir que el tiempo de respuesta crece en proporción cuadrática de acuerdo a la cantidad de usuarios incorporados.

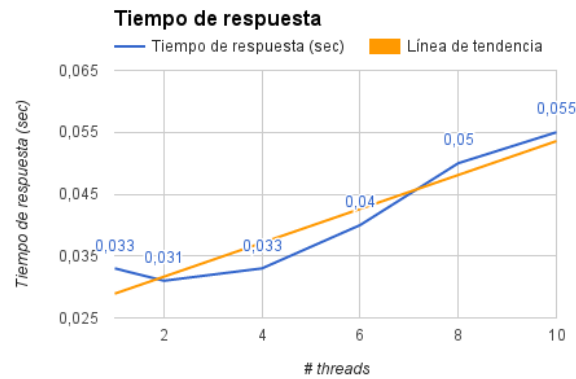


Figura B.15 Tiempo de respuesta

Tomando un promedio de los valores que se obtuvieron, el tiempo de respuesta para la herramienta es: 0,044 segundos

B.6.3.12. Throughput

El throughput refiere al número de eventos de respuesta en un intervalo. Los test utilizados para determinar el tiempo de respuesta de la herramienta, contribuyeron en la obtención del valor del throughput para la herramienta. En este sentido, se ejecutaron durante 30 minutos cada uno de los test. Luego, se utilizó la ecuación B.1 para calcular el throughput. La gráfica B.93 muestra el porcentaje de procesos iniciados en 30 minutos, de acuerdo a la cantidad de usuarios.

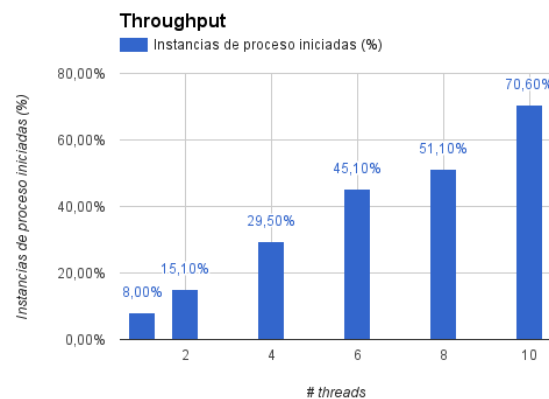


Figura B.16 Throughput (%) frente a Usuarios

Tomando la mediana de los valores que se obtuvieron, el throughput para la herramienta es: 37,30%

B.6.3.13. Capacidad

La capacidad de un sistema es la cantidad de trabajo que este puede rendir y está relacionado con el throughput. Los datos para obtener la capacidad se obtuvieron a partir de las ejecuciones de los test utilizados para calcular el throughput. En este caso se observó la cantidad de procesos que finalizaron en 30 minutos, mientras que para el throughput se observó la cantidad de procesos iniciados. La gráfica B.17 muestra el porcentaje de procesos finalizados en 30 minutos, de acuerdo a la cantidad de usuarios.

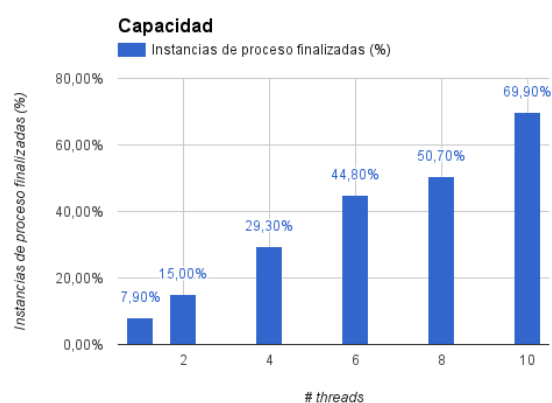


Figura B.17 Capacidad (%) frente a Usuarios

Tomando la mediana de los valores que se obtuvieron, la capacidad para la herramienta es: 37,05 %

B.6.4. Tabla de nivel de cumplimiento

La tabla B.26 muestra los resultados obtenidos para las características funcionales. Las características están divididas por categoría. De cada característica se muestra si la evaluación es teórica (Teó), el nivel de soporte (NS) y el nivel de cumplimiento (NC).

Tabla B.26 Características técnicas - Nivel de cumplimiento.

Modulo	Característica	Teó	NS	NC
	Tecnología, Arquitectura e Interoperabilidad			
	Arquitectura de BPMS	Si	⊗	⊗
	Compatibilidad con estándares	Si	●	✓
	Base de datos	Si	●	✓
	Sistema operativo del modelador	Si	●	✓
	Navegador Web	Si	●	✓
	Extensibilidad	Si	●	✓

Modulo	Característica	Teó	NS	NC
	Integración con Servicios Web	No	●	✓
	Diseño y modelado de procesos			
	Interfaz gráfica para key users	No	●	✓
	Tipo de modelador	Si	●	✓
	Notación gráfica de procesos compatible con BPMN 2.0	Si	●	✓
	Gestión de formularios			
	Definición de formularios	No	●	✓
	Formato de formularios	No	●	✓
	Elementos de formularios	No	●	✓
	Formularios dinámicos	No	●	✓
	Reglas de llenado de campos	No	●	✗
	Validación de datos del formulario	No	●	✓
	Motor de Workflow			
	Iniciación de procesos en forma manual y automática	No		
	Definición de reglas de negocio	No	●	✓
	Asignación de tareas	No	●	✓
	Reasignación de tareas	No	●	✓
	Adjuntar documentos a tareas	No	●	✓
	Versionado de procesos	No	●	✓
	Tareas de borrador	No		
	Gestión, monitoreo y auditoría			
	Gestión de instancias por usuarios	No	●	✓
	Mantenimiento histórico de procesos	No	●	✓
	Monitoreo de procesos	No	●	⊘
	Portal			
	Presentación de las tareas	No	●	✓

Referencias:

-  No aplica
-  Cumple de forma nativa
-  Cumple vía integración
-  Cumple vía particularización
-  Soporta totalmente
-  Soporta parcialmente
-  No soportada

B.6.4.1. Resultado numérico de las características técnicas

Las tablas B.28 y B.30 presentan los resultados numéricos correspondientes a las características funcionales y no funcionales evaluadas. En la sección 3.7 se describe el método de evaluación, donde se presentan los cálculos a realizar para poder clasificar a las características según los umbrales definidos.

Primero, hay que definir los valores de importancia, para poder definir los umbrales. Además, es necesario dar valores a los niveles de cumplimiento y de soporte.

Tabla B.27 Definición de constantes

Importancia		Nivel de cumplimiento		Nivel de soporte	
Alto	3	Nativo	2	Total	3
Medio	2	Vía integración	1	Parcial	2
Bajo	1	Vía particularización	1	No soportado	1

Las columnas de la tabla B.28 contienen: las características, importancia (I), nivel de soporte (NS), nivel de cumplimiento (NC), el factor (F) y el puntaje total (Total). El puntaje total se calcula de la siguiente manera:

Puntaje total

$$Puntaje = I * NS * NC * Factor \quad (B.5)$$

Donde I es la importancia, NS el nivel de soporte, NC el nivel de cumplimiento y Factor un valor entre 0,1 y 0,9.

Tabla B.28 Características técnicas - Cálculo del puntaje.

Característica	I	NS	NC	F	Total
Tecnología, Arquitectura e Interoperabilidad					
	1	0	0	1	0
Compatibilidad con estándares	1	1	2	1	2
Base de datos	1	2	2	1	4
Sistema operativo del modelador	1	2	2	1	4
Navegador Web	1	2	2	1	4
Extensibilidad	1	2	2	1	4
Integración con Servicios Web	1	1	2	1	2
Diseño y modelado de procesos					
Interfaz gráfica para key users	1	1	2	1	2
Tipo de modelador	1	1	2	1	2
Notación gráfica de procesos compatible con BPMN 2.0	1	1	2	1	2
Gestión de formularios					
Definición de formularios	1	2	2	1	4
Formato de formularios	1	1	2	1	2
Elementos de formularios	1	1	2	1	2
Formularios dinámicos	1	2	2	1	4
Reglas de llenado de campos	1	1	1	0,4	0,4
Validación de datos del formulario	1	2	2	1	4
Motor de Workflow					
Iniciación de procesos en forma manual y automática	1	0	0	1	0
Definición de reglas de negocio	1	2	2	1	4
Asignación de tareas	1	2	2	1	4
Reasignación de tareas	1	1	2	1	2
Adjuntar documentos a tareas	1	1	2	1	2
Versionado de procesos	1	1	2	1	2
Tareas de borrador	1	0	0	1	0
Gestión, monitoreo y auditoría					
Gestión de instancias por usuarios	1	2	2	1	4
Mantenimiento histórico de procesos	1	2	2	1	4
Monitoreo de procesos	1	0	0	1	0
Portal					
Presentación de las tareas	1	1	2	1	2

Por otro lado tenemos los cálculos de las características no funcionales. Las columnas de la tabla B.30 contienen: el peso de cada subcaracterística (P_i), el nivel de cumplimiento de la subcaracterística (s_i), el valor final de la subcaracterística definido en la ecuación B.18 y el nivel de cumplimiento (NC). El nivel de cumplimiento se obtiene de la siguiente manera:

- $vf < 1 \rightarrow$ Nivel de cumplimiento Bajo.
- $1 \leq vf < 3 \rightarrow$ Nivel de cumplimiento Medio.
- $3 \leq vf \rightarrow$ Nivel de cumplimiento Alto.

Valor final

$$vf = \frac{\sum_{i=1}^N (s_i * P_i)}{\sum_{i=1}^N P_i} \quad (B.6)$$

Siendo vf el ‘valor final’ de la característica y N la cantidad de subcaracterísticas.

En el caso de las características incluidas en la categoría Performance, hay que definir cuales son los valores de $L1$ y $L2$ para cada una de ellas, de esta manera, se podrá determinar el valor final de cada característica. Para determinar estos valores se consideraron estudios previos de performance así como de usabilidad [47] [42]. A continuación se muestra una tabla con los valores determinados:

Tabla B.29 Parámetros

Característica	L1	L2
Tiempo de respuesta	3 seg	10 seg
Throughput	30%	80 %
Capacidad	20%	70 %

Se considera que todas las subcaracterísticas son importantes y en este sentido se asigna el valor 1 al peso de cada subcaracterística.

B.7. Evaluación de Intalio/bpms

Se obtuvo una licencia académica a través del sitio web de la herramienta. Junto con la licencia el contacto adjunto los enlaces para las descargas tanto del modelador como del servidor. Se utilizó la edición *Enterprise 7.5.0*.

B.7.1. Evaluación de las características teóricas

B.7.1.1. Arquitectura del BPMS

Intalio es una software Open Source cuyo fin es el desarrollo de procesos de negocios basados en los estándares: BPMN, BPEL y BPEL4People. Intalio, permite diseñar, desplegar y optimizar procesos de negocio, sin necesidad de escribir ni una línea de código.

Intalio BPM provee la tecnología para crear una capa que provee servicios (Web Services), que modelen los procesos de negocio de la organización y todas sus reglas de negocio. El BPM debe descansar sobre una arquitectura orientada a servicios (SOA).

La figura B.18 muestra los componentes de la arquitectura y su interacción.

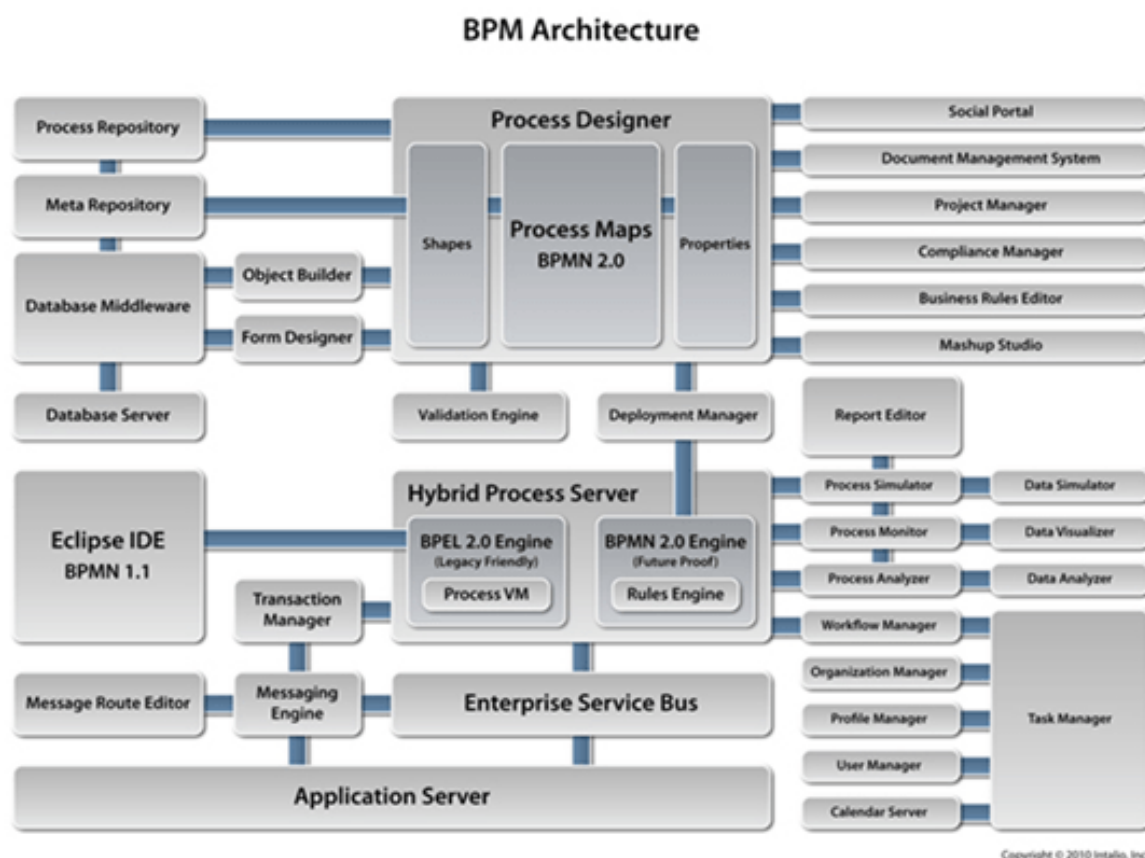


Figura B.18 Arquitectura IntalioDesigner [28]

[28] describe la arquitectura de Intalio y muestra sus dos componentes principales:

Designer: dedicado al modelado del proceso, implementación y deployment.

Servidor: dedicado a la ejecución de procesos.

Los componentes a su vez están integrados por diferentes módulos. Algunos que componen la edición community y otros, sólo disponibles para la edición enterprise. A continuación se describen los componentes más importantes de *Intalio|Designer* e *Intalio|Server*.

Intalio|Designer

BPMN Modeler es la herramienta que permite generar el modelo del proceso.

Deployment Manager garantiza la condición de cero código, es decir, con este módulo, un usuario es capaz de crear un proceso sin necesidad de utilizar código.

Process Implementation Tools permite manipular los datos de configuración y tiempo de ejecución.

Database Connector permite a Intalio tener un sistema de gestión de bases de datos más general.

Form Editor provee la herramienta AJAX que tiene un entorno para ver, crear e integrar componentes a un formulario.

Para la edición enterprise, se agregan los siguientes módulos:

BAM (Business Activity Monitoring) refiere a las practicas necesarios para:

- Proveer información relevante y de forma oportuna.
- Realizar toma de decisiones relacionadas al resultado del análisis.
- Implementar las acciones necesarias para el mejoramiento del desempeño del proceso.

BRE (Business Rules Engine) es el módulo que contiene el motor de reglas de negocios.

El módulo provee una herramienta llamada *business freandly decision table editor*, que está incluida en el editor y se puede incluir al modelo utilizando drang-and-drop.

LDAP es el sistema de gestión de usuarios.

Intalio|Server

Intalio permite utilizar como servidores Tomcat, JBoss o Weblogic.

A continuación se describen los módulos que componen Intalio|Server:

BPEL Engine es el motor de procesos de negocios, Intalio utiliza Apache ODE[29]

Tempo BPEL4People gestiona diversos patrones de flujo de trabajo. Una de sus principales características es que expone sus APIs mediante servicios web.

Los siguientes módulos son dos aplicaciones web:

Process Console esta aplicación permite monitorear los procesos habilitados.

Task List interfaz dedicada al usuario final, donde puede gestionar sus tareas.

Para la edición enterprise, se agregan los siguientes módulos:

BAM (Business Activity Monitoring) refiere a las practicas necesarios para:

- Proveer información relevante y de forma oportuna.
- Realizar toma de decisiones relacionadas al resultado del análisis.
- Implementar las acciones necesarias para el mejoramiento del desempeño del proceso.

BRE (Business Rules Engine) es el módulo que contiene el motor de reglas de negocios.

Permite ver las reglas de negocios independientemente del proceso.

LDAP es el sistema de gestión de usuarios.

ESB (Enterprise Service Bus) permite la integración completa para una arquitectura SOA.

Portal

DMS (Document management system) el sistema de gestión de documentos que se recomienda es Alfresco.

B.7.1.2. Compatibilidad con estándares

Permite importar/exportar procesos en formatos BPEL. Además desde el portal se puede exportar el modelo en formato pdf, png y svg.

B.7.1.3. Base de datos

Soporta conexión con las bases de datos:

- H2 Database, es la base que trae predeterminada y no es soportada en producción.
- MS SQL Server (2008, 2012, 2014)
- MySQL (5.5, 5.6)
- Oracle (10g, 11g, 12c)
- PostgreSQL (9.1, 9.2, 9.3)

Para poder utilizar MySQL, Oracle o MS SQL Server, es necesario descargar el driver. Las configuraciones para cada una de las bases se encuentran dentro del directorio del servidor en \databases.

B.7.1.4. Sistema operativo del modelador

Intalio solo cuenta con un modelador de escritorio, el mismo es soportado en los siguientes sistemas operativos:

- Windows 7 32/64-bit
- Windows 8 and 8.1 64-bit
- OS X 10
- Linux Ubuntu 32/64-bit
- Kubuntu 12 32/64-bit

B.7.1.5. Navegador web

Intalio depende de JavaScript para la experiencia del usuario, según la documentación los navegadores que proveen el mejor motor de JavaScript para la mejor experiencia del usuarios son:

- Internet Explorer 8 en Windows XP
- Internet Explorer 9 en Windows XP/7
- Internet Explorer 10 en Windows 7/8/8.1
- Internet Explorer 11 en Windows 7/8/8.1
- Firefox – versiones recientes, en todos los sistemas operativos.
- Chrome – versiones recientes, en todos los sistemas operativos.

Se ha demostrado un correcto funcionamiento en los navegadores:

- Mozilla Firefox - versión 41.0.1 en Windows 10
- Google Chrome - versión 45.0.2454.101 en Windows 10

B.7.1.6. Extensibilidad

Una de las principales características que tiene IntalioIbpms es la posibilidad de extender la herramienta, ya sea agregando APIs, por ejemplo, Apache ActiveMQ para mensajería; así como también integrando herramientas para la administración de documentos (Alfresco), sistemas de autenticación (LDAP).

B.7.1.7. Tipo de modelador

Intalio solo cuenta con un modelador de escritorio, si bien se puede obtener mediante el sitio web, el modelador es una versión de Eclipse con los cambios apropiados para modelar.

B.7.1.8. Notación gráfica de procesos compatible con BPMN 2.0

[48] afirma que soporta el estándar BPMN 2.0 para el modelado.

B.7.2. Evaluación de las características prácticas

B.7.2.1. Integración con servicios web

[49] recomienda la integración con Core y SOAP para los tipos de servicios web WS-Addressing, también permite la integración con WS-BasicProfile versión 1.1 [50] y con los WS-Security versiones 1.0 y 1.1 (este tipo de servicios web, proporcionan un medio para aplicar seguridad a los servicios web).

Adicionalmente, Intalio permite la integración con servicios XML, XPath, XQuery, BPEL. La conexión a un servicio web se realiza a través de conectores, los cuales se comunican con las tareas, permitiendo la asociación de variables del proceso al servicio web, tanto de entrada como de salida.

Para realizar las pruebas se utilizó un servicio web de tipo SOAP.

B.7.2.2. Interfaz gráfica para key users

Con IntalioDesigner Modeling el usuario es capaz de realizar modelos de procesos sin necesidad de programar. Es posible desarrollar, visualizar y modificar tanto los diagramas de flujo como los atributos de cada elemento del modelo.

B.7.2.3. Definición de formularios

Los formularios se definen mediante AXAJForm, esto permite crear distintos tipos de formularios. Los elementos se agregan mediante *drag-and-drop*, una vez en el formulario se pueden establecer reglas de llenado sobre esos campos, sin necesidad de programación, así mismo, los formularios cuentan con la posibilidad de introducir javascript para reglas de llenado más complejas.

B.7.2.4. Formato de formularios

La edición de formularios permite modificar la apariencia de cada campo, color, alineación, fuente, tamaños. Permite agregar tooltips, imágenes de error, etc. Sin embargo, los campos no tienen tamaño dinámico, es decir, no se adaptan al tamaño del dato a completar.

Intalio permite crear distintos tipos de formularios, incluso reutilizar formularios de otros procesos. También permite copiar un formulario existente y extenderlo. Los formularios son independientes de las actividades, esto permite asociar el mismo formulario a más de una actividad.

B.7.2.5. Elementos del formulario

Dentro de un formulario es posible incluir listas desplegables cuyo conjunto de valores están en una base de datos. Es necesario definir un conector a la base con la consulta correspondiente, luego se crea un XML Mapping para asociar la consulta con el elemento del formulario. Es posible definir botones con acciones predefinidas, por ejemplo, completar un formulario con valores predeterminados.

B.7.2.6. Formularios dinámicos

Es posible ocultar/mostrar campos de un formulario dependiendo del valor de otro campo o del rol del usuario. Además permite actualizar los valores de un campo, de acuerdo a la selección de otro campo del formulario. Por ejemplo, si hay dos campos de tipo lista, departamento y nombre, una vez que se selecciona un departamento, la lista nombre se actualiza con el nombre del departamento seleccionado.

B.7.2.7. Reglas de llenado de campos

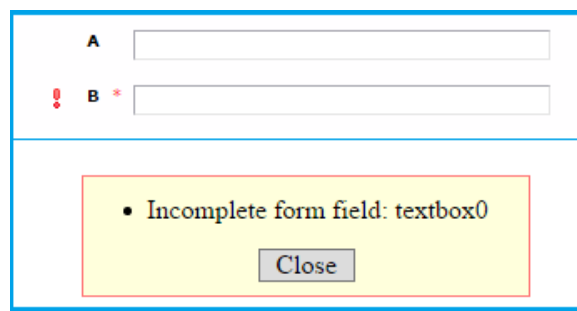
La definición de reglas de llenado de campos en Intalioldesigner presenta varios aspectos. No solo soporta totalmente la característica *Formularios dinámicos* (ver sección B.8.2.6), además, es capaz de conectarse a un servicio web, para obtener un valor del formulario a partir de otro.

También cuenta con una herramienta llamada *Data Mapper*[51] que permite asociar campos completados de otros formularios, el resultado de un cálculo algorítmico, información de ejecución del proceso[29] o una variable del proceso, al campo de un formulario.

B.7.2.8. Validación de campos de formularios

AXAJFrom permite validar cada campo del formularios, según el tipo (email, money, number, etc.) o mediante una expresión regular. Esta validación la realiza una vez que se envía el formulario, mostrando un mensaje de error cuando el campo no cumple con la validación.

También permite indicar si un campo es requerido, en ese caso, si el formulario es enviado sin el dato requerido, se muestra un mensaje de error (como se ve en la figura B.19), y vuelve al formulario, donde se ven los datos que ya se habían completado. Los campos requeridos se marcan con un asterisco (*), además, mientras el campo no es completado, tiene un signo de interrogación rojo.



The image shows a web form with two input fields, labeled A and B. Field B is marked as required with an asterisk (*) and a red question mark icon. Below the form, a yellow error message box is displayed, stating "Incomplete form field: textbox0" with a "Close" button.

Figura B.19 Validación de datos en formularios

B.7.2.9. Iniciación de procesos en forma manual y automática

Las instancias de proceso se pueden iniciar tanto de forma manual como automática. Para iniciar un proceso de forma manual se debe identificar el proceso que se quiere iniciar en la sección *Process* del menú lateral del portal. Los usuarios encuentran en esta sección solo los procesos que están activos y ellos tiene el rol o están asignados a la primer tarea del mismo.

Respecto al inicio automático de instancias de proceso. Intalio permite utilizar *Timer Start Event*. Esto permite, definir un momento específico en el cual se debe iniciar una instancia de proceso, además de crear procesos recurrentes, es decir que se crea una instancia cada cierto período de tiempo, ver [52]

B.7.2.10. Definición de reglas de negocio

Para poder definir reglas de negocio en la herramienta se debe crear una tabla de decisión, donde cada fila es una regla de negocio, las columnas son las condiciones (se unen con el AND lógico), todas las reglas (filas) en una tabla de condición se unen con un OR lógico. Si las condiciones para una regla son verdaderas, entonces todas las acciones para la regla son ejecutadas. Las reglas pueden tener condiciones solapadas.

Las variables del proceso y la información del proceso se pueden asociar a las condiciones de la regla de negocio, mientras que los plazos se pueden definir a través de opciones de las reglas donde se puede definir el período de validez de la regla de negocio.

Si bien no se pueden importar archivos DRL (Drools) al Intaliol designer, se puede crear una tabla de decisión como un servicio web, ver [53].

Las reglas se pueden actualizar de forma externa, sin necesidad de hacer un deploy sobre el proyecto, para ello es necesario que la regla de negocio se encuentre en un repositorio (Drools Guvnor), esto además permite reutilizar las reglas de negocio en diferentes proyectos.

B.7.2.11. Asignación de tareas

Permite la asignación de tareas de forma manual o automática. En particular, es posible asignar una tarea a un grupo de usuarios y que, cuando la tarea es tomada por uno de los usuarios, la misma automáticamente deja de estar disponible para el resto de los usuarios del grupo. A su vez, el usuario puede liberar la tarea, en ese momento vuelve a estar disponible para los usuarios y roles asignados antes de ser reclamada [54].

La asignación automática se hace a través de una regla de negocio, por lo tanto se puede realizar a través de variables y estados.

B.7.2.12. Reasignación de tareas

La herramienta permite la reasignación manual y automática de tareas. La reasignación manual se realiza a través del portal, mientras que la reasignación automática se realiza a través de una tarea de *Escalation*[55], la cual permite reasignar a un supervisor si expira el tiempo de finalización de la tarea. También permite reasignar a través de una regla de negocio.

B.7.2.13. Adjuntar documentos a tareas

Permite adjuntar a una tarea uno o más documentos, indicando nombre, tamaño y tipo de documento. También permite agregar documentos a la instancia de un proceso o agregar un control para subir archivos en el formulario. Intalio permite utilizar diferentes DMS (sistema de manejo de archivos), como Alfresco y SharePoint, pero no al mismo tiempo, se debe elegir sólo un sistema. Ver [56]

B.7.2.14. Versionado de procesos

Brinda la posibilidad de que existan varias versiones de un mismo proceso. Sin embargo, solo una de las versiones puede estar activa. Si hay instancias activas de un proceso y se genera una nueva versión, dichas instancias seguirán ejecutando de acuerdo a la versión activa. Por lo tanto los cambios se verán reflejados en las tareas pendientes, esto permite la migración de los procesos a versiones actuales.

Intalio permite reactivar versiones anteriores como la versión actual, solamente cuando no hay versiones activas, es decir, la versión que está activa hay que retirarla, antes de activar una versión anterior.

B.7.2.15. Tareas de borrador

Al ingresar a una tarea, hay tres opciones disponibles, guardar, completar o reclamar (en caso que la tarea este asignada a un grupo). Si se completan los datos del formulario y se selecciona la opción guardar, cuando se reingresa a la tarea, la misma mantiene los datos ingresados.

B.7.2.16. Gestión de instancias por usuarios

Todos los usuarios son capaces de iniciar instancias de los procesos que tienen disponibles en el portal. Para pausar o reanudar, borrar, reinstalar o abortar instancias del proceso, el usuario debe tener permisos para administrar el monitoreo de los procesos e instancias, dicho permiso es otorgado por los usuarios que tienen rol administrador en el sistema.

Intalio no permite sobrescribir manualmente datos de las instancias del proceso.

B.7.2.17. Mantenimiento de histórico del proceso

Los usuarios con rol administrador tienen acceso al monitoreo de instancias y procesos, esto les permite ver el histórico de los procesos y sus instancias.

De los procesos se puede ver, la versión y quién realizó el deploy, el estado (activo/inactivo) y, un resumen de las instancias de esos procesos (cuántas están en progreso, completadas, terminadas y cuántas fallaron). De las instancias se puede ver, el identificador, el proceso al que corresponden, el estado, fecha de comienzo y de la última actividad, el diagrama, datos de las actividades de la instancia y un histórico de los eventos. Para visualizar el usuario que realizó una actividad, es necesario ingresar a los datos de las actividades, el cual muestra en formato xml los datos de la misma.

No es posible acceder a los documentos que se adjuntaron a las instancias, tampoco que un usuario final pueda seguir la historia de un proceso gráficamente, ya que necesita que un administrador le brinde acceso, tanto a la página de administración de procesos e instancias como al dashboard del sistema.

B.7.2.18. Monitoreo de procesos

Para el monitoreo y gestión de instancias y procesos la herramienta cuenta con un módulo de monitoreo, el cual es visible sólo para usuarios con rol administrador. También provee un dashboard, en el cual se pueden agregar diferentes widget que son configurables y muestra datos de un proceso, las instancias de un proceso o de las tareas respecto a un usuario, a un rol o al estado de la misma.

Intalio brinda la posibilidad de generar reportes personalizados[57]. Cuando se diseña el proceso se puede generar un reporte (archivo .rptdesign). Una vez que el proceso está activo en el portal, dentro de la sección *Reports* se agrega el archivo y se define un nuevo widget, el cual permite ver los datos del reporte dentro del dashboard.

B.7.2.19. Presentación de las tareas

Al ingresar al portal de Intalio, no se visualizan las tareas que tiene el usuario. La sesión se inicia en la última pantalla que tenía abierta el usuario al momento de finalizar la sesión previa. Si es la primera vez que el usuario inicia sesión, verá una pantalla en blanco, y a la izquierda el menú con las opciones que tiene disponibles.

De todas maneras, una vez que se accede al listado de tareas, las mismas se pueden filtrar, ordenar, priorizar.

Si el usuario ingreso al portal a través de Mozilla Firefox, en el menú superior izquierdo, tiene dos opciones que no aparecen en Chrome. My Tasks y Personal Process Feed, donde se puede suscribir ya sea a través de los marcadores dinámicos, el email de yahoo o una aplicación.

B.7.3. Evaluación de las características no funcionales

B.7.3.1. Capacidad de aprendizaje

Tanto Intaliolddesigner como el portal brindan herramientas para un rápido aprendizaje de las mismas. Cada elemento de la paleta que se puede agregar al modelo, posee un tooltip que indica su funcionalidad. Algunos elementos que no son tan intuitivos, además de la funcionalidad indica como se deben utilizar. Si el modelo tiene un error, con una cruz roja indica donde está y si aparece una lámpara, quiere decir que tiene una sugerencia de posible solución. A pesar de no tener manuales de uso, Intaliolddesigner cuenta con una serie de tutoriales explicativos.

Del lado del portal, solamente los iconos que no tienen una descripción poseen tooltips. Intalio cuenta con una página de soporte donde se encuentran los manuales de uso [30].

Si bien brinda mensajes de errores descriptivos, por ejemplo, cuando un campo del formulario que es numérico se completa con letras, al completar la tarea indica el error; pero no brinda mensajes cuando hay un error al finalizar una tarea debido a un problema de conexión a la base, a un servicio web o un campo que no se completo porque no era obligatorio, pero el valor del campo era utilizado en el proceso.

Tabla B.31 Evaluación - Capacidad de aprendizaje

	Modelador	Portal
¿Posee tooltips?	SI	SI
¿Tiene mensajes de error descriptivos?	SI	NO
¿Brinda sugerencia de posible solución ante un error?	SI	NO
¿Posee manuales de uso?	SI	SI

B.7.3.2. Capacidad para ser usado

Intaliol designer brinda un conjunto de tips que son accesibles desde la barra de herramientas. Utiliza símbolos mnemotécnicos para representar acciones conocidas (copiar, pegar, cortar, borrar, etc.)

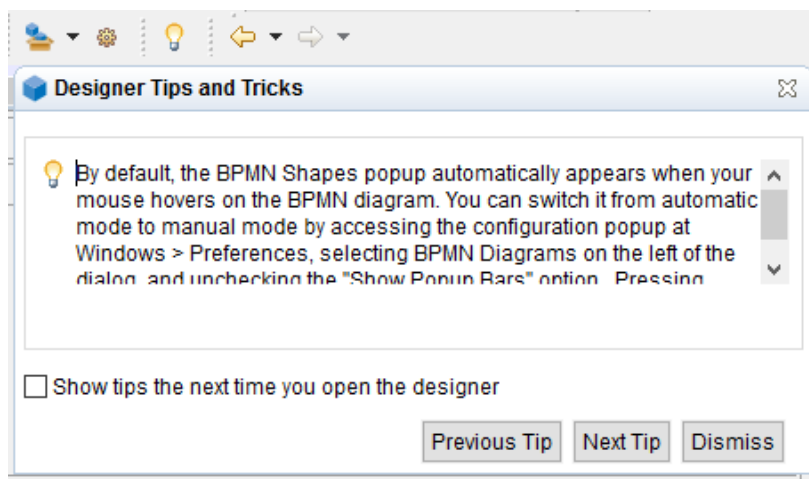


Figura B.20 Intaliol designer - Tips

Al igual que el designer, el portal utiliza símbolos mnemotécnicos. Además, permite acceder a los manuales de uso a través de un link dentro del menú de información del usuario. Intalio brinda un espacio de preguntas frecuentes, dentro de la página de soporte[30], donde también hay un espacio disponible para reportar problemas del BPMs.

Tabla B.32 Evaluación - Capacidad para ser usado

	Modelador	Portal
¿Brinda menú de ayuda?	SI	NO
¿Tiene símbolos mnemotécnicos?	SI	SI
¿Brinda un espacio de preguntas frecuentes?	SI	SI
¿Posee acceso directo a manuales de uso?	NO	SI

B.7.3.3. Protección contra errores de usuarios

Cuando hay un problema con el modelo, se visualiza un triángulo amarillo donde está el problema, con un mensaje de advertencia. Si bien se permite agregar validación sobre los datos de un formulario, esta validación no se verifica al momento de compilar el modelo. Se puede utilizar Ctl + Z y Ctl + Y, para deshacer y rehacer una acción.



Figura B.21 Símbolo de advertencia

Es posible validar los datos de un formulario desde el portal. También brinda mensajes de advertencia, por ejemplo, cuando no se completo un dato requerido de un formulario o cuando se quiere ingresar a una tarea, que ya fue reclamada por otro usuario. El portal no cuenta con opciones para deshacer y rehacer acciones.

Tabla B.33 Evaluación - Protección contra errores de usuarios

	Modelador	Portal
¿Tiene mensajes de advertencia?	SI	SI
¿Posee validación de datos y formularios?	SI	SI
¿Cuenta con opciones para deshacer y rehacer una acción?	SI	NO

B.7.3.4. Estética de la interfaz de usuario

La información que se incluye tanto en el portal como en el designer no es innecesaria. Sin embargo, la información relevante del portal si bien es accesible, no es fácil de encontrar. Por ejemplo, quién ejecuto una tarea y cuando.

Tabla B.34 Evaluación - Estética de la interfaz de usuario

	Modelador	Portal
¿Incluye información innecesario?	NO	NO
La información relevante ¿Se encuentra accesible y fácil de encontrar?	SI	NO

B.7.3.5. Accesibilidad

Las acciones tanto en el modelador como en el portal se representan con imágenes adecuadas. Los colores son distinguibles en el modelador, pero dentro del portal, información

importante como la prioridad de una tarea, utiliza el mismo símbolo para todas las prioridades y la prioridad crítica (roja) y normal (verde) podrían no ser distinguibles para todas las personas.

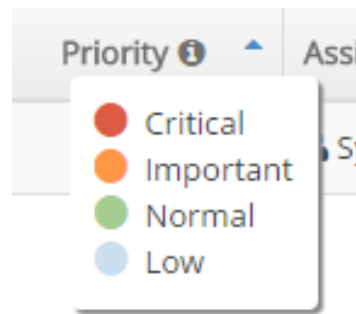


Figura B.22 Prioridad

Ni el modelador, ni el portal cuentan con comando de voz o utilizan sonido alguno.

Tabla B.35 Evaluación - Accesibilidad

	Modelador	Portal
¿Incluye comando de voz?	NO	NO
¿Las acciones se encuentran representadas con imágenes adecuadas?	SI	SI
¿Los colores son distinguibles aunque el usuario sea daltónico?	SI	NO
¿Los sonidos se corresponden con la acción que se está ejecutando?	NO	NO

B.7.3.6. Confidencialidad

Las contraseñas se guardan cifradas en la base de datos. Intalio permite configurar el servidor para que utilice el protocolo https, para más información ver [58].

Tabla B.36 Evaluación - Confidencialidad

	Portal
¿Utiliza técnicas criptográficas?	SI
Mobile ¿Almacena los datos cifrados en el dispositivo?	NO
¿Los intercambios de datos con el servidor son sobre HTTPS?	SI

B.7.3.7. Integridad

Un usuario administrador es capaz de crear nuevos roles y asignar permisos a dichos roles, esto permite visualizar diferentes funciones según el rol.

Figura B.23 Creación de roles

El acceso a los documentos depende del DMS que se utilice, aunque se puede programar el acceso, según el rol. Los documentos no se guardan cifrados. Intalio cuenta con un tiempo limite de sesión activa.

Figura B.24 Sesión finalizada

Tabla B.37 Evaluación - Integridad

	Portal
¿Permite definir roles?	SI
¿Cuenta con acceso diferenciado a los documentos según el rol?	SI
¿Permite visualizar diferentes funciones según el rol?	SI
¿Permite administrar permisos?	SI
¿Los documentos se guardan cifrados?	NO
¿Cuenta con un tiempo limite de sesión activa?	SI

B.7.3.8. No repudio

Intalio cuenta con un historial de ejecución, cómo ya se explico en la sección B.7.2.17.

Tabla B.38 Evaluación - No repudio

	Portal
¿Cuenta con un historial de ejecución de procesos?	SI
¿Se puede identificar que usuario y en que momento se ejecutó un proceso?	SI

B.7.3.9. Responsabilidad

Las tareas y los procesos cuentan con un identificador numérico que es único.

Tabla B.39 Evaluación - Responsabilidad

	Portal
¿Las tareas y los procesos tienen identificador único?	SI

B.7.3.10. Autenticidad

Intalio provee su propia forma de autenticación, pero también permite integrar el sistema con bases externas como LDAP.

No hay ninguna evidencia que demuestre que Intalio soporta firmas digitales.

Tabla B.40 Evaluación - Autenticidad

	Portal
¿Cuenta con alguna forma de autenticación de usuarios?	SI
¿Soporta firmas digitales?	SI

B.7.3.11. Tiempo de respuesta

Se realizaron varias ejecuciones de un conjunto de test para determinar el tiempo de respuesta. El gráfico de la figura B.25 muestra el crecimiento del tiempo de respuesta, conforme aumenta la cantidad de usuarios. El gráfico azul representa los valores obtenidos luego de la ejecución, mientras que la función naranja indica la tendencia. La tendencia corresponde a una función lineal, esto implica que a medida que aumenta la cantidad de usuarios, el tiempo de respuesta aumente en forma directamente proporcional.

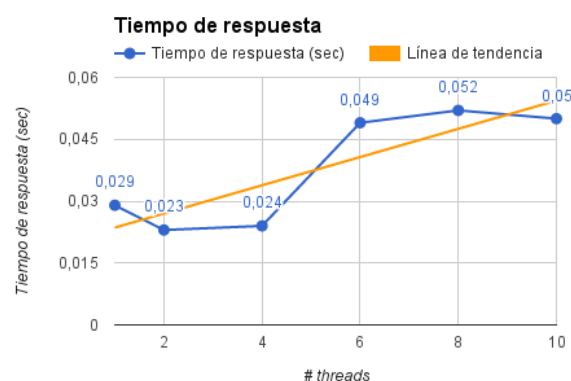


Figura B.25 Tiempo de respuesta

Tomando un promedio de los valores que se obtuvieron, el tiempo de respuesta para la herramienta es: 0,0395 segundos

B.7.3.12. Throughput

El throughput refiere al número de eventos de respuesta en un intervalo. Los test utilizados para determinar el tiempo de respuesta de la herramienta, contribuyeron en la obtención del valor del throughput para la herramienta. En este sentido, se ejecutaron durante 30 minutos cada uno de los test. Luego, se utilizó la ecuación B.1 para calcular el throughput. La gráfica B.26 muestra el porcentaje de procesos iniciados en 30 minutos, de acuerdo a la cantidad de usuarios.

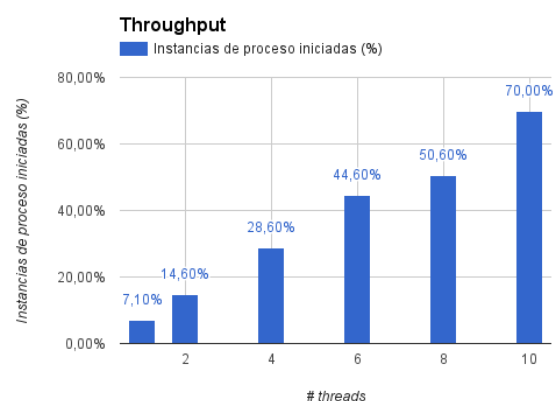


Figura B.26 Throughput (%) frente a Usuarios

Tomando la mediana de los valores que se obtuvieron, el throughput para la herramienta es: 36,60%

B.7.3.13. Capacidad

La capacidad de un sistema es la cantidad de trabajo que este puede rendir. La capacidad y el throughput son conceptos relacionados, en este sentido, para obtener los datos de la capacidad se ejecutaron los test utilizados para calcular el throughput. En este caso se observó la cantidad de procesos que finalizaron en 30 minutos, mientras que para el throughput se observó la cantidad de procesos iniciados. La gráfica de la figura B.27 muestra el porcentaje de procesos finalizados en 30 minutos, de acuerdo a la cantidad de usuarios.

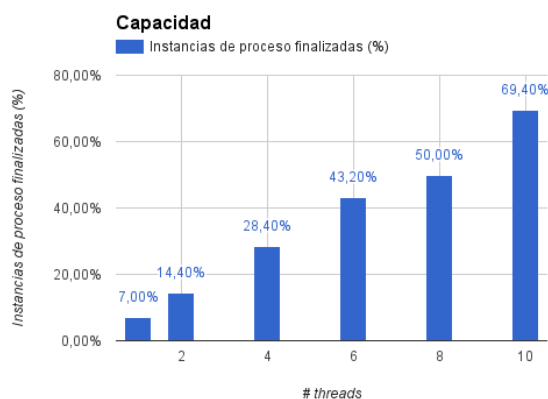


Figura B.27 Capacidad (%) frente a Usuarios

Tomando la mediana de los valores que se obtuvieron, la capacidad para la herramienta es: 35,80 %

B.7.4. Tabla de nivel de cumplimiento

La tabla B.41 muestra los resultados obtenidos para las características funcionales. Las características están divididas por categoría. De cada característica se muestra si la evaluación es teórica (Teó), el nivel de soporte (NS) y el nivel de cumplimiento (NC).

Tabla B.41 Características técnicas - Nivel de cumplimiento.

Modulo	Característica	Teó	NS	NC
Técnico				
	Tecnología, Arquitectura e Interoperabilidad			
	Arquitectura de BPMS	Si	⊗	⊗
	Compatibilidad con estándares	Si	●	✓
	Base de datos	Si	●	✓
	Sistema operativo del modelador	Si	●	✓
	Navegador Web	Si	●	✓

Modulo	Característica	Teó	NS	NC
	Extensibilidad	Si	●	⚙️
	Integración con Servicios Web	No	●	✓
	Diseño y modelado de procesos			
	Interfaz gráfica para key users	No	●	✓
	Tipo de modelador	Si	●	✓
	Notación gráfica de procesos compatible con BPMN 2.0	Si	●	✓
	Gestión de formularios			
	Definición de formularios	No	●	✓
	Formato de formularios	No	●	✓
	Elementos de formularios	No	●	✓
	Formularios dinámicos	No	●	✓
	Reglas de llenado de campos	No	●	✓
	Validación de datos del formulario	No	●	✓
	Motor de Workflow			
	Iniciación de procesos en forma manual y automática	No	●	✓
	Definición de reglas de negocio	No	●	✓
	Asignación de tareas	No	●	✓
	Reasignación de tareas	No	●	✓
	Adjuntar documentos a tareas	No	●	⚙️
	Versionado de procesos	No	●	✓
	Tareas de borrador	No	●	✓
	Gestión, monitoreo y auditoría			
	Gestión de instancias por usuarios	No	●	✓
	Mantenimiento histórico de procesos	No	●	✓
	Monitoreo de procesos	No	●	✓
	Portal			
	Presentación de las tareas	No	●	✓

B.7.4.1. Resultado numérico de las características técnicas

Las tablas B.43 y B.45 presentan los resultados numéricos correspondientes a las características funcionales y no funcionales evaluadas. En la sección 3.7 se describe el método de evaluación, donde se presentan los cálculos a realizar para poder clasificar a las

características según los umbrales definidos.

Primero, hay que definir los valores de importancia, para poder definir los umbrales. Además, es necesario dar valores a los niveles de cumplimiento y de soporte.

Tabla B.42 Definición de constantes

Importancia		Nivel de cumplimiento		Nivel de soporte	
Alto	3	Nativo	2	Total	3
Medio	2	Vía integración	1	Parcial	2
Bajo	1	Vía particularización	1	No soportado	1

Las columnas de la tabla B.43 contienen: las características, importancia (I), nivel de soporte (NS), nivel de cumplimiento (NC), el factor (F) y el puntaje total (Total). El puntaje total se calcula de la siguiente manera:

Puntaje total

$$Puntaje = I * NS * NC * Factor \quad (B.7)$$

Donde I es la importancia, NS el nivel de soporte, NC el nivel de cumplimiento y Factor un valor entre 0,1 y 0,9.

Tabla B.43 Características técnicas - Cálculo del puntaje.

Característica	I	NS	NC	F	Total
Tecnología, Arquitectura e Interoperabilidad					
Compatibilidad con estándares	1	1	2	1	2
Base de datos	1	2	2	1	4
Sistema operativo del modelador	1	2	2	1	4
Navegador Web	1	2	2	1	4
Extensibilidad	1	2	1	1	2
Integración con Servicios Web	1	2	2	1	4
Diseño y modelado de procesos					
Interfaz gráfica para key users	1	2	2	1	4
Tipo de modelador	1	2	2	1	4
Notación gráfica de procesos compatible con BPMN 2.0	1	2	2	1	4
Gestión de formularios					
Definición de formularios	1	1	2	1	2

Característica	I	NS	NC	F	Total
Formato de formularios	1	2	2	1	4
Elementos de formularios	1	2	2	1	4
Formularios dinámicos	1	2	2	1	4
Reglas de llenado de campos	1	2	2	1	4
Validación de datos del formulario	1	2	2	1	4
Motor de Workflow					
Iniciación de procesos en forma manual y automática	1	2	2	1	4
Definición de reglas de negocio	1	2	2	1	4
Asignación de tareas	1	2	2	1	4
Reasignación de tareas	1	2	2	1	4
Adjuntar documentos a tareas	1	2	1	1	2
Versionado de procesos	1	1	2	1	2
Tareas de borrador	1	2	2	1	4
Gestión, monitoreo y auditoría					
Gestión de instancias por usuarios	1	1	2	1	2
Mantenimiento histórico de procesos	1	1	2	1	2
Monitoreo de procesos	1	1	2	1	2
Portal					
Presentación de las tareas	1	1	2	1	2

Por otro lado tenemos los cálculos de las características no funcionales. Las columnas de la tabla B.45 contienen: el peso de cada subcaracterística (P_i), el nivel de cumplimiento de la subcaracterística (s_i), el valor final de la subcaracterística definido en la ecuación B.18 y el nivel de cumplimiento (NC). El nivel de cumplimiento se obtiene de la siguiente manera:

- $vf < 1 \rightarrow$ Nivel de cumplimiento Bajo.
- $1 \leq vf < 3 \rightarrow$ Nivel de cumplimiento Medio.
- $3 \leq vf \rightarrow$ Nivel de cumplimiento Alto.

Valor final

$$vf = \frac{\sum_{i=1}^N (s_i * P_i)}{\sum_{i=1}^N P_i} \quad (B.8)$$

Siendo vf el 'valor final' de la característica y N la cantidad de subcaracterísticas.

En el caso de las características incluidas en la categoría Performance, hay que definir cuales son los valores de $L1$ y $L2$ para cada una de ellas, de esta manera, se podrá determinar el valor final de cada característica. Para determinar estos valores se consideraron estudios previos de performance así como de usabilidad [47] [42]. A continuación se muestra una tabla con los valores determinados:

Tabla B.44 Parámetros

Característica	L1	L2
Tiempo de respuesta	3 seg	10 seg
Throughput	30 %	80 %
Capacidad	20 %	70 %

Se considera que todas las subcaracterísticas son importantes y en este sentido se asigna el valor 1 al peso de cada subcaracterística.

B.8. Evaluación de Activiti

Para la evaluación de Activiti se utilizó la versión 6.0.0.Beta2 la cual fue descargada de la página de la herramienta[20]. Como es una versión Beta, la documentación no se encontraba actualizada y algunas funcionalidades no estaban disponibles.

B.8.1. Evaluación de las características teóricas

B.8.1.1. Arquitectura del BPMS

Activiti es una plataforma BPM orientada a responsables de negocio, desarrollados y administradores de sistemas. Su núcleo es un motor de procesos de BPMN 2 desarrollado en Java. Es una herramienta open-source y es distribuida bajo la licencia de Apache. Activiti puede ser ejecutado en cualquier aplicación Java, en un servidor, en un cluster o en la nube. Se integra con Spring, es ligera y basada en conceptos simples.

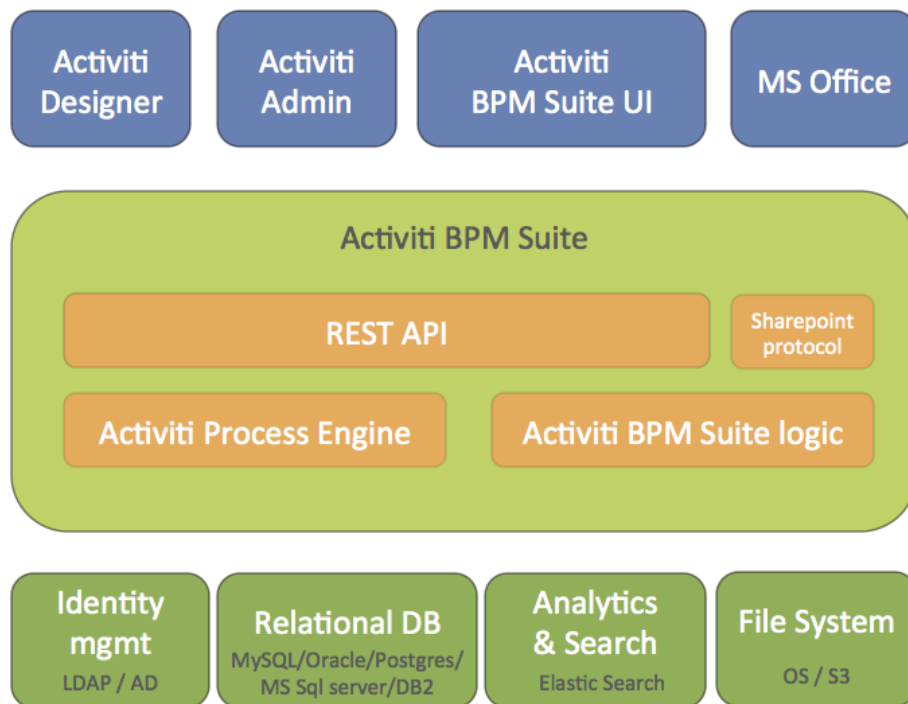


Figura B.28 Componentes de Activiti [31]

Activiti Engine : es el componente principal del proyecto Activiti. Es un motor de procesos Java que corre procesos BPMN 2 de forma nativa. Mejora la comunicación entre usuarios no técnicos y los usuarios técnicos.

Event listeners: esta funcionalidad permite ejecutar acciones que pueden ser definidas mediante código Java o mediante un *script*. Esto significa que un desarrollador puede

decorar el proceso con detalles técnicos que no son mostrados en el diagrama.

Custom activities: en el caso que sea necesario utilizar un tipo de tarea que no este definido en Activiti, el desarrollador puede crear una actividad *custom* en Java, que implemente el comportamiento que necesita.

Activiti Explorer (Activiti-App en la versión 6): es un ejemplo de aplicación web que provee acceso al Activiti Engine, para todos los usuarios el sistema. Incluye un gestor de tareas, un inspector de instancias de proceso y un gestor de despliegues.

Principales usos del Gestor de Tareas: Ver una lista de tareas personales Ver una lista de tareas para la cual el usuario es candidato. Crear nuevas tareas (no relacionadas al procesos) Completar una tarea enviando datos en un formulario de la tarea. Reasignar una tarea a un usuario diferente. Ver que otras personas están involucrados en procesos, en relación a esa tarea.

Principales usos del inspector de instancias de procesos: Ver los detalles de una instancia de proceso. Los dueños del proceso o el administrador pueden ver los detalles de una instancia de proceso que esta ejecutando.

Manejo de despliegues: Ver los despliegues Ver las definiciones de aplicaciones

Activiti Modeler Es una aplicación web basada en Angular JS. La principal diferencia con las versiones previas (Signavio Core Components) es que el nuevo modelador es mantenido y desarrollado por Activiti. Activiti Modeler soporta todos los elementos BPM y las extensiones soportadas por Activiti Engine.

Activiti Designer Es un plugin de Eclipse que permite el modelado de procesos BPMN 2.0. Además permite hacer el deploy del proceso en el mismo IDE.

B.8.1.2. Compatibilidad con estándares

Según las secciones *14.2 Importing existing models* de [39] Activiti permite importar procesos solo en el formato BPMN, ya sea con extensión .bpmn o .bpmn20.xml y según la sección *14.4 Export model to BPMN XML* de [39] permite exportarlos en el formato BPMN2.0.

B.8.1.3. Base de datos

En la guía de usuarios de Activiti en la sección *3.5 - Supported Databases* de [39] se puede ver que las bases de datos soportadas son:

- H2 Database, es la base que trae predeterminada y no es soportada en producción.
- MS SQL Server
- MySQL
- Oracle

- PostgreSQL
- DB2

B.8.1.4. Sistema operativo del modelador

Activiti se puede utilizar en cualquier sistema operativo que tenga Java y Apache Tomcat instalados.

Para la evaluación se utilizó el sistema operativo Windows 10 con JDK 1.8.

B.8.1.5. Navegador web

No se encontró material que especifique los navegadores soportados por Activiti, en el manual de usuario hay funcionalidades en las cuales se menciona que pueden funcionar o no dependiendo de si es soportado por el navegador que se este utilizando. Para la evaluación se utilizo el navegador Google Chrome 48.0.2564.103 m

B.8.1.6. Extensibilidad

Activiti cuenta con una API REST que referencia un conjunto de funcionalidades que facilita al usuario la programación para manipular a los procesos, obtener información de los mismos, etc. La herramienta además es integrable con otros servicios externos y con funcionalidades de otro tipo de herramientas: correo electrónico, DBMS, Camel, Mule, etc. Por más información ver las secciones 15, 16 y 17 de (author?) [39].

B.8.1.7. Tipo de modelador

Activiti cuenta con un modelador web, también provee la opción de utilizar un modelador de escritorio mediante la utilización de Eclipse y pluggins ver sección 14.1 de [39].

B.8.1.8. Notación gráfica de procesos compatible con BPMN 2.0

Activiti es compatible con el estándar BPMN 2.0 ver sección 7 y 8 de [39].

B.8.2. Evaluación de las características prácticas

B.8.2.1. Integración con servicios web

La integración con web Services no es nativa. Activiti define una tipo de tarea WS Task pero solo para modelado. Para poder consumir un web services desde una tarea de Activiti es necesario definir una clase java que implemente "TaskListener". Esta clase debe encontrarse en un archivo .jar cuyo nombre sea el mismo que el nombre del proceso. Luego se debe incluir el archivo como librería de activiti-app. En una tarea de servicio (Service Task) se debe definir el task listener y seleccionar el momento (al ejecutarse, al completarse o al tomarse la tarea) en el que mismo será ejecutado.

B.8.2.2. Interfaz gráfica para key users

La herramienta cuenta con el modelador web lo que permite al usuario realizar modelos de procesos sin necesidad de programar. El modelador permite desarrollar, visualizar y modificar los diagramas de flujo así como los atributos del modelo.

B.8.2.3. Definición de formularios

La herramienta brinda la posibilidad de definir formularios personalizados para cada tarea. Permite la reutilización de los formularios ya que los mismos son almacenados y al momento de crear un nuevo formulario para una tarea, la herramienta te lista los formularios existentes brindando la posibilidad de reutilizarlos. Para la definición de los formularios Activiti provee una paleta con los elementos que se pueden agregar al formulario mediante *drag and drop*. Permite definir reglas de llenado para los elementos del formulario.

B.8.2.4. Formato de formularios

En la creación de un nuevo formulario se puede cambiar la ubicación de un campo, pero solo la alineación vertical. Los formularios no cuentan con la funcionalidad de edición de fuente, estilo y color. El tamaño de los campos del formulario es dinámico de acuerdo al dato a completar.

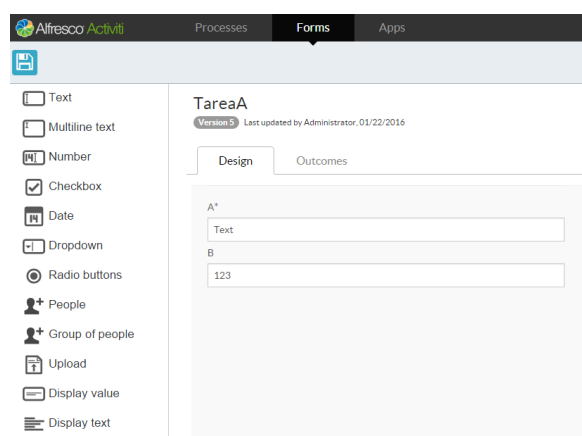


Figura B.29 Creación de formulario en Activiti

B.8.2.5. Elementos del formulario

Los formularios permiten definir listas desplegables en las cuales es posible seleccionar los valores que tomarán los elementos de la lista e indicar cuál de ellos será el elemento seleccionado por defecto. Los valores de la lista son persistidos en la tabla *forms* junto con los otros datos del formulario. En los formularios se pueden definir múltiples botones de *submit* los cuales no tienen acciones predefinidas, sino pueden ser utilizados en condiciones, para acciones posteriores al *submit*. Los formularios permiten definir si un campo es requerido o no. No permiten definir campos ocultos.

B.8.2.6. Formularios dinámicos

Los formularios no permiten actualizar valores de una lista dependiendo de los valores ingresados en otras listas o campos del formulario. Si permite mostrar el valor de una variable

Figura B.30 Crear *label* y listas en Activiti

que ya haya sido definida en un formulario previo. También permite incluir texto dentro del cual utilice valores de alguna de las variables definidas en un formulario previo.

B.8.2.7. Reglas de llenado de campos

B.8.2.8. Validación de campos de formularios

Los formularios cuentan con validación de los datos. No permite el ingreso de valores que no sean del tipo especificado por el campo del formulario. También permite definir cuales son campos requeridos y cuales no lo son. Activiti notifica al usuario cuando un campo es requerido y el mismo no fue completado, no así para las validaciones de tipos.

B.8.2.9. Iniciación de procesos en forma manual y automática

Los procesos en Activiti pueden ser iniciados de forma manual, o de forma automática por un servicio web. No cuenta con agendas para la iniciación de los procesos.

B.8.2.10. Definición de reglas de negocio

A nivel de modelado es posible incluir en el modelo una tarea de tipo Regla de Negocio. Esta tarea solo sirve para modelar dado que no se provee una implementación de dicha funcionalidad. Se intentó realizar la misma mediante el desarrollo de código java y si bien se logró implementar la funcionalidad la misma no pudo ser ejecutado debido a que no se logró importar la definición de la regla de negocio a la aplicación.

B.8.2.11. Asignación de tareas

En la etapa de modelado las tareas pueden asignarse a un usuario, a un grupo de usuarios, a una lista de usuarios candidatos o una lista de grupos candidatos. También se puede indicar que el usuario que realizará la tarea será el mismo que inicie el proceso.

Figura B.31 Asignación de tarea en Activiti

No se encontró en la documentación una forma de realizar la asignación de las tareas en forma automática, tampoco se puede realizar la asignación de tareas en base a algoritmos predefinidos.

B.8.2.12. Reasignación de tareas

La reasignación de las tareas se puede realizar solo en forma manual. El usuario que tiene la tarea asignada es el único que puede reasignar la tarea a otro usuario. Aunque el usuario que tiene la tarea asignada haya tomado la misma, igual la tarea puede ser reasignada a otro usuario. No se encontró la manera de realizar la reasignación mediante reglas de negocio o de reasignar una lista de tareas de un usuario a otros.

B.8.2.13. Adjuntar documentos a tareas

Activiti permite adjuntar cualquier tipo de documentos a las tareas. Solo el usuario que realiza la tarea puede ver el documento, a solo que se asignen involucrados a la tarea. De esta forma los el usuario que completa la tarea y los usuarios involucrados en la misma, son los habilitados a y borrar el documento incluido en la tarea pero no pueden editar el mismo.

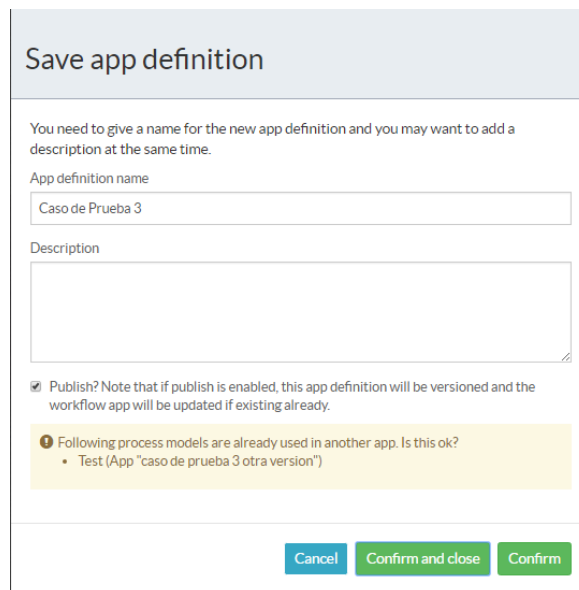
El documento solo es visible en la instancia del proceso en la que el mismo fue adjunto. No se transfiere a la próxima instancia. El usuario que este habilitado a ver la información del proceso, puede ver el documento si ingresa a ver los detalles de la tarea donde se adjunto el documento.

No se encontró en la documentación la manera de definir reglas de negocio para controlar el tamaño del documento y el tipo de documento.

B.8.2.14. Versionado de procesos

Activiti permite el versionado de procesos. Cuando se edita un proceso y se realiza clic en guardar la herramienta pregunta si se quiere crear una nueva versión del proceso. Se pueden ejecutar a la vez varias versiones de un proceso siempre y cuando el mismo pertenezca a otra aplicación. Al momento de crear una nueva aplicación con una nueva versión de un proceso que ya esta ejecutando, la herramienta muestra un mensaje de advertencia sobre la versión en ejecución del mismo proceso. Si se confirma la acción se puede ejecutar la nueva versión del proceso junto con la versión anterior.

Si no se genera una nueva aplicación y se edita la aplicación existente generando una nueva publicación de la misma, quedan las tareas pendientes de la versión anterior y las nuevas instancias de ese procesos son las correspondientes a la nueva versión del mismo. Las tareas pendientes de la versión antigua pueden completarse sin inconvenientes.



Save app definition

You need to give a name for the new app definition and you may want to add a description at the same time.

App definition name

Caso de Prueba 3

Description

☒ Publish? Note that if publish is enabled, this app definition will be versioned and the workflow app will be updated if existing already.

Warning: Following process models are already used in another app. Is this ok?

- Test (App "caso de prueba 3 otra version")

Cancel Confirm and close Confirm

Figura B.32 Generación de una nueva versión de un proceso en Activiti

B.8.2.15. Tareas de borrador

La herramienta no permite guardar los datos de las tareas. Una vez que se toma una tarea solo se puede completar.

tipo de asignación de usuario (si la tarea fue asignada a él, si está involucrado en la tarea o si es un candidato para completar la misma). El filtro cuenta con un campo de búsqueda en el cual se puede buscar por el nombre de la tarea.

En otra vista se pueden ver la lista de procesos en los que el usuario puede participar. En la lista de procesos se muestra el nombre, la fecha en la que fue iniciado el proceso y el usuario que inició esa instancia. La lista se puede ordenar por los procesos más nuevos y por el más antiguo. Hay tres listas, una contiene todos los procesos, otra contiene solo los procesos que ya finalizaron y los procesos que aún están ejecutando.

En ambas vistas si se realiza clic en cualquiera de los ítems de las mismas, se pueden ver información más detallada.

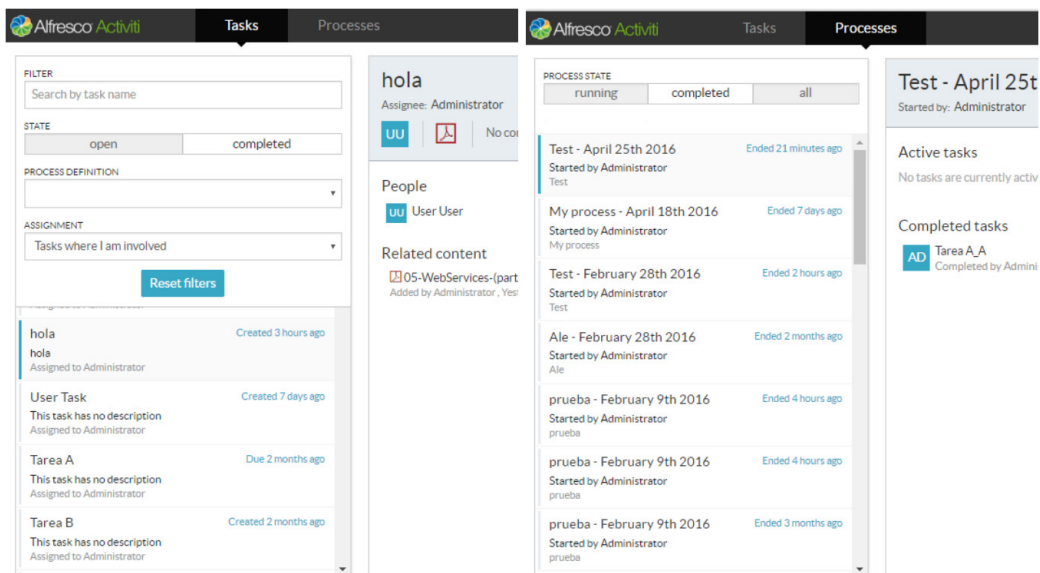


Figura B.34 Vista de la lista de tareas y de la lista de procesos de un usuario en Activiti

B.8.3. Evaluación de las características no funcionales

B.8.3.1. Capacidad de aprendizaje

Activiti brinda funcionalidades que facilitan la utilización del modelador. El modelador cuenta con tooltips en todos los elementos de la paleta brindando una breve descripción de la funcionalidad del elemento, en caso de que el mismo no sea demasiado intuitivo. En tanto en el portal web existen algunos elementos que poseen tooltips y otros que no. El modelador no cuenta con un botón específico para la validación del modelo pero no deja incluir elementos de la paleta en lugares que no sean los correctos, coloreando de rojo la zona en la que no se puede colocar el elemento. Ni el portal ni el modelador cuentan con acceso directo a los

manuales de usuario, sin embargo existen manuales de usuarios los cuales se encuentran accesibles en la carpeta Doc del código de Activiti o desde la pagina web de la herramienta pero referencia a una versión anterior del manual.

Tabla B.46 Evaluación - Capacidad de aprendizaje

	Modelador	Portal
¿Posee tooltips?	SI	SI
¿Tiene mensajes de error descriptivos?	SI	SI
¿Brinda sugerencia de posible solución ante un error?	SI	NO
¿Posee manuales de uso?	SI	SI

B.8.3.2. Capacidad para ser usado

Activiti posee menú de ayuda solo a nivel de modelador y no en el portal. Posee símbolos mnemotécnicos en el portal y en el modelador lo cual hacen que la herramienta sea de fácil uso.

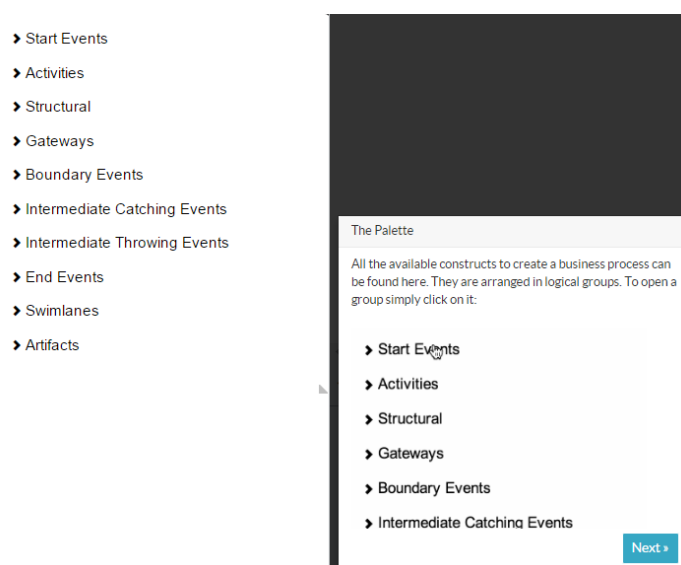


Figura B.35 Ayuda de Activiti

La herramienta no cuenta con un espacio de preguntas frecuentes en ninguno de los dos como tampoco posee accesos directos a los manuales de uso.

Tabla B.47 Evaluación - Capacidad para ser usado

	Modelador	Portal
¿Brinda menú de ayuda?	SI	NO
¿Tiene símbolos mnemotécnicos?	SI	SI
¿Brinda un espacio de preguntas frecuentes?	NO	NO
¿Posee acceso directo a manuales de uso?	NO	NO

B.8.3.3. Protección contra errores de usuarios

Tanto en el modelador como en el portal se muestra al usuario mensajes de advertencia antes de ejecutar una acción. No te permite realizar acciones incorrectas como por ejemplo no te permite escribir texto en un campo que sea numérico, pero en esos caso no te indicando cual es la acción incorrecta. En ambos se muestran mensajes para avisar al usuario cuando una acción se ejecuto de forma correcta y en caso de que sea incorrecta en algunos caso se marca en rojo el campo que tiene errores y mediante tooltips se muestra la acción esperada.

Figura B.36 Mensaje de validación de Activiti

El modelador cuenta con opciones para deshacer acciones mientras que el portal no ofrece dicha funcionalidad.

Tabla B.48 Evaluación - Protección contra errores de usuarios

	Modelador	Portal
¿Tiene mensajes de advertencia?	SI	SI
¿Posee validación de datos y formularios?	SI	SI
¿Cuenta con opciones para deshacer y rehacer una acción?	SI	NO

B.8.3.4. Estética de la interfaz de usuario

La interfaz de usuario de Activiti es amigable tanto en el portal como en el modelador. Posee los elementos relevantes y no incluye información innecesaria. La información y las acciones importantes son de fácil acceso.

Tabla B.49 Evaluación - Estética de la interfaz de usuario

	Modelador	Portal
¿Incluye información innecesario?	NO	NO
La información relevante ¿Se encuentra accesible y fácil de encontrar?	SI	SI

B.8.3.5. Accesibilidad

El portal web y el modelador no incluyen comandos de voz para representar acciones. Las acciones son representadas mediante imágenes adecuadas pero no en todos los casos los colores son distinguibles. Se utiliza el color rojo para los mensajes de error y el verde para los mensajes de éxito.

Tabla B.50 Evaluación - Accesibilidad

	Modelador	Portal
¿Incluye comando de voz?	NO	NO
¿Las acciones se encuentran representadas con imágenes adecuadas?	SI	SI
¿Los colores son distinguibles aunque el usuario sea daltónico?	NO	NO
¿Los sonidos se corresponden con la acción que se está ejecutando?	SI	SI

B.8.3.6. Confidencialidad

Activiti cuenta con técnicas criptográficas para almacenar las contraseñas en la base de datos. No cuenta aún con una aplicación mobile. El intercambio de datos con el servidor no es sobre HTTPS.

Tabla B.51 Evaluación - Confidencialidad

	Portal
¿Utiliza técnicas criptográficas?	SI
Mobile ¿Almacena los datos cifrados en el dispositivo?	NO
¿Los intercambios de datos con el servidor son sobre HTTPS?	NO

B.8.3.7. Integridad

Solo el usuario administrador es capaz de crear grupos, nuevos usuarios y asignar los usuarios a grupos. Por lo tanto hay funciones diferenciadas según el rol. El administrador puede acceder al modelado de procesos y a la definición de nuevas aplicaciones mientras que el resto de los usuarios no puede hacerlo. Los permisos, sin embargo, no pueden ser administrados siendo posible ser administrador (superusuario) que posee todas las funcionalidades, o no serlo. Los grupos que no son administradores, solo pueden acceder a

las tareas y procesos. El acceso a los documentos esta diferenciado según el rol del usuario, pero los mismos no se guardan cifrados en la base. La sesión no expira, por lo que permanece activa hasta que el usuario decida cerrar la sesión.

Tabla B.52 Evaluación - Integridad

	Portal
¿Permite definir roles?	SI
¿Cuenta con acceso diferenciado a los documentos según el rol?	SI
¿Permite visualizar diferentes funciones según el rol?	SI
¿Permite administrar permisos?	NO
¿Los documentos se guardan cifrados?	NO
¿Cuenta con un tiempo limite de sesión activa?	NO

B.8.3.8. No repudio

Activiti cuenta con un historial de ejecución, ver sección B.8.2.17.

Tabla B.53 Evaluación - No repudio

	Portal
¿Cuenta con un historial de ejecución de procesos?	SI
¿Se puede identificar que usuario y en que momento se ejecutó un proceso?	SI

B.8.3.9. Responsabilidad

En Activiti las tareas y los procesos cuentan con un identificador numérico que es único.

Tabla B.54 Evaluación - Responsabilidad

	Portal
¿Las tareas y los procesos tienen identificador único?	SI

B.8.3.10. Autenticidad

Activiti cuenta con autenticación de los usuarios para el ingreso al portal y al modelador, pero también permite integrar el sistema con bases externas como LDAP. Por más información ver *Sección 17* de [39].

Tabla B.55 Evaluación - Autenticidad

	Portal
¿Cuenta con alguna forma de autenticación de usuarios?	SI
¿Soporta firmas digitales?	NO

B.8.3.11. Tiempo de respuesta

Para determinar el tiempo de respuesta se realizaron varias ejecuciones de un conjunto de test. El gráfico de la figura B.37 muestra el crecimiento del tiempo de respuesta, según aumenta la cantidad de usuarios. En la misma se observan dos gráficos, el azul corresponde al tiempo de respuesta, en tanto el naranja corresponde a la tendencia del gráfico. En este caso se observa una tendencia polinómica de segundo grado, esto quiere decir que el tiempo de respuesta crece en proporción cuadrática de acuerdo a la cantidad de usuarios incorporados.

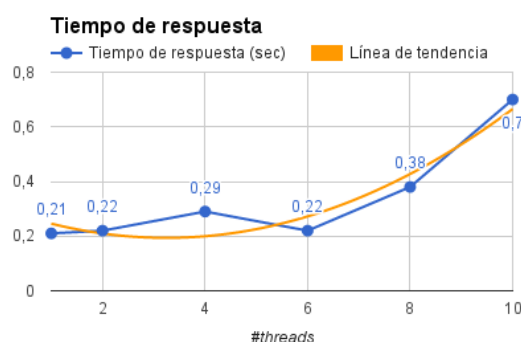


Figura B.37 Tiempo de respuesta

Tomando un promedio de los valores que se obtuvieron, el tiempo de respuesta para la herramienta es: 0,455 segundos

B.8.3.12. Throughput

El throughput refiere al número de eventos de respuesta en un intervalo. Los test utilizados para determinar el tiempo de respuesta de la herramienta, también fueron utilizados para la obtención del valor del throughput. Es así que, se ejecutaron durante 30 minutos cada uno de los test. Luego, se utilizó la ecuación B.1 para calcular el throughput. La gráfica B.38 muestra el porcentaje de procesos iniciados en 30 minutos, de acuerdo a la cantidad de usuarios.

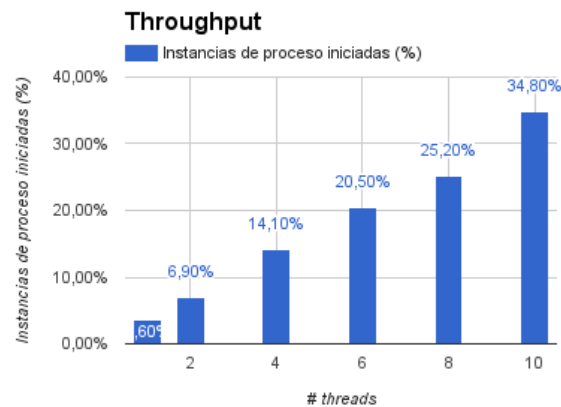


Figura B.38 Throughput (%) frente a Usuarios

Tomando la mediana de los valores que se obtuvieron, el throughput para la herramienta es: 17,30%

B.8.3.13. Capacidad

La capacidad de un sistema es la cantidad de trabajo que este puede rendir y está relacionado con el throughput. Los datos para obtener la capacidad se obtuvieron a partir de las ejecuciones de los test utilizados para calcular el throughput. En este caso se observó la cantidad de procesos que finalizaron en dicho intervalo de tiempo. La gráfica de la figura B.39 muestra el porcentaje de procesos finalizados en 30 minutos, de acuerdo a la cantidad de usuarios.

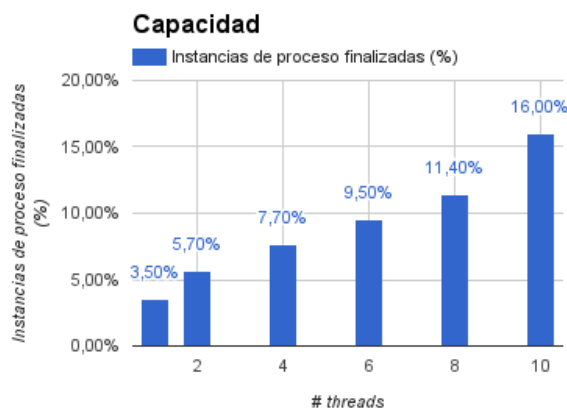


Figura B.39 Capacidad (%) frente a Usuarios

Tomando la mediana de los valores que se obtuvieron, la capacidad para la herramienta es: 8,60%

B.8.4. Tabla de nivel de cumplimiento

La tabla B.56 muestra los resultados obtenidos para las características funcionales. Las características están divididas por categoría. De cada característica se muestra si la evaluación es teórica (Teó), el nivel de soporte (NS) y el nivel de cumplimiento (NC).

Tabla B.56 Características técnicas - Nivel de cumplimiento.

Modulo	Característica	Teó	NS	NC
Técnico				
	Tecnología, Arquitectura e Interoperabilidad			
	Arquitectura de BPMS	Si		
	Compatibilidad con estándares	Si		
	Base de datos	Si		
	Sistema operativo del modelador	Si		
	Navegador Web	Si		
	Extensibilidad	Si		
	Integración con Servicios Web	No		
	Diseño y modelado de procesos			
	Interfaz gráfica para key users	No		
	Tipo de modelador	Si		
	Notación gráfica de procesos compatible con BPMN 2.0	Si		
	Gestión de formularios			
	Definición de formularios	No		
	Formato de formularios	No		
	Elementos de formularios	No		
	Formularios dinámicos	No		
	Reglas de llenado de campos	No		
	Validación de datos del formulario	No		
	Motor de Workflow			
	Iniciación de procesos en forma manual y automática	No		
	Definición de reglas de negocio	No		
	Asignación de tareas	No		
	Reasignación de tareas	No		
	Adjuntar documentos a tareas	No		

Modulo	Característica	Teó	NS	NC
	Versionado de procesos	No		
	Tareas de borrador	No		
	Gestión, monitoreo y auditoría			
	Gestión de instancias por usuarios	No		
	Mantenimiento histórico de procesos	No		
	Monitoreo de procesos	No		
	Portal			
	Presentación de las tareas	No		

Referencias:

-  No aplica
-  Cumple de forma nativa
-  Cumple vía integración
-  Cumple vía particularización
-  Soporta totalmente
-  Soporta parcialmente
-  No soportada

B.8.4.1. Resultado numérico de las características técnicas

Las tablas B.58 y B.60 presentan los resultados numéricos correspondientes a las características funcionales y no funcionales evaluadas. En la sección 3.7 se describe el método de evaluación, donde se presentan los cálculos a realizar para poder clasificar a las características según los umbrales definidos.

Primero, hay que definir los valores de importancia, para poder definir los umbrales. Además, es necesario dar valores a los niveles de cumplimiento y de soporte.

Tabla B.57 Definición de constantes

Importancia		Nivel de cumplimiento		Nivel de soporte	
Alto	3	Nativo	2	Total	3
Medio	2	Vía integración	1	Parcial	2
Bajo	1	Vía particularización	1	No soportado	1

Las columnas de la tabla B.58 contienen: las características, importancia (I), nivel de soporte (NS), nivel de cumplimiento (NC), el factor (F) y el puntaje total (Total). El puntaje total se calcula de la siguiente manera:

Puntaje total

$$Puntaje = I * NS * NC * Factor \quad (B.9)$$

Donde I es la importancia, NS el nivel de soporte, NC el nivel de cumplimiento y Factor un valor entre 0,1 y 0,9.

Tabla B.58 Características técnicas - Calculo del puntaje.

Característica	I	NS	NC	F	Total
Tecnología, Arquitectura e Interoperabilidad					
Compatibilidad con estándares	1	1	2	1	2
Base de datos	1	2	2	1	4
Sistema operativo del modelador	1	2	2	1	4
Navegador Web	1	2	2	1	4
Extensibilidad	1	2	2	1	4
Integración con Servicios Web	1	1	1	0,7	0,7
Diseño y modelado de procesos					
Interfaz gráfica para key users	1	2	2	1	4
Tipo de modelador	1	2	2	1	4
Notación gráfica de procesos compatible con BPMN 2.0	1	2	2	1	4
Gestión de formularios					
Definición de formularios	1	2	2	1	4
Formato de formularios	1	1	2	1	2
Elementos de formularios	1	1	2	1	2
Formularios dinámicos	1	2	2	1	4
Reglas de llenado de campos	1	1	2	1	2
Validación de datos del formulario	1	2	2	1	4
Motor de Workflow					
Iniciación de procesos en forma manual y automática	1	1	2	1	2
Definición de reglas de negocio	1	2	1	0,1	0,2
Asignación de tareas	1	2	2	1	4
Reasignación de tareas	1	2	2	1	4

Característica	I	NS	NC	F	Total
Adjuntar documentos a tareas	1	2	2	1	4
Versionado de procesos	1	2	2	1	4
Tareas de borrador	1	0	0	1	0
Gestión, monitoreo y auditoría					
Gestión de instancias por usuarios	1	1	2	1	2
Mantenimiento histórico de procesos	1	2	2	1	4
Monitoreo de procesos	1	0	0	1	0
Portal					
Presentación de las tareas	1	1	2	1	2

Por otro lado tenemos los cálculos de las características no funcionales. Las columnas de la tabla B.60 contienen: el peso de cada subcaracterística (P_i), el nivel de cumplimiento de la subcaracterística (s_i), el valor final de la subcaracterística definido en la ecuación B.18 y el nivel de cumplimiento (NC). El nivel de cumplimiento se obtiene de la siguiente manera:

- $vf < 1 \rightarrow$ Nivel de cumplimiento Bajo.
- $1 \leq vf < 3 \rightarrow$ Nivel de cumplimiento Medio.
- $3 \leq vf \rightarrow$ Nivel de cumplimiento Alto.

Valor final

$$vf = \frac{\sum_{i=1}^N (s_i * P_i)}{\sum_{i=1}^N P_i} \quad (B.10)$$

Siendo vf el 'valor final' de la característica y N la cantidad de subcaracterísticas.

En el caso de las características incluidas en la categoría Performance, hay que definir cuales son los valores de $L1$ y $L2$ para cada una de ellas, de esta manera, se podrá determinar el valor final de cada característica. Para determinar estos valores se consideraron estudios previos de performance así como de usabilidad [47] [42]. A continuación se muestra una tabla con los valores determinados:

Tabla B.59 Parámetros

Característica	L1	L2
Tiempo de respuesta	3 seg	10 seg
Throughput	30 %	80 %
Capacidad	20 %	70 %

Se considera que todas las subcaracterísticas son importantes y en este sentido se asigna el valor 1 al peso de cada subcaracterística.

B.9. Evaluación de Bizagi BPM Suite

Se utilizó una versión completa de la herramienta, únicamente limitada por la cantidad de usuarios del sistema.

B.9.1. Evaluación de las características teóricas

B.9.1.1. Arquitectura del BPMS

La arquitectura interna de Bizagi se apoya en un diseño moderno con uso de tecnologías igualmente modernas que permiten ofrecer una solución completa de nivel empresarial.

Bizagi no genera código intermedio y al ser orientado al modelo, se puede puede fácilmente: configurar una arquitectura de alta disponibilidad, escalar horizontalmente dicha solución, o utilizar mecanismos adicionales a nivel de la base de datos para optimizar el acceso a los datos para las operaciones diarias y para los reportes.

El siguiente diagrama ilustra los componentes principales internos de Bizagi en ejecución y su arquitectura interna:

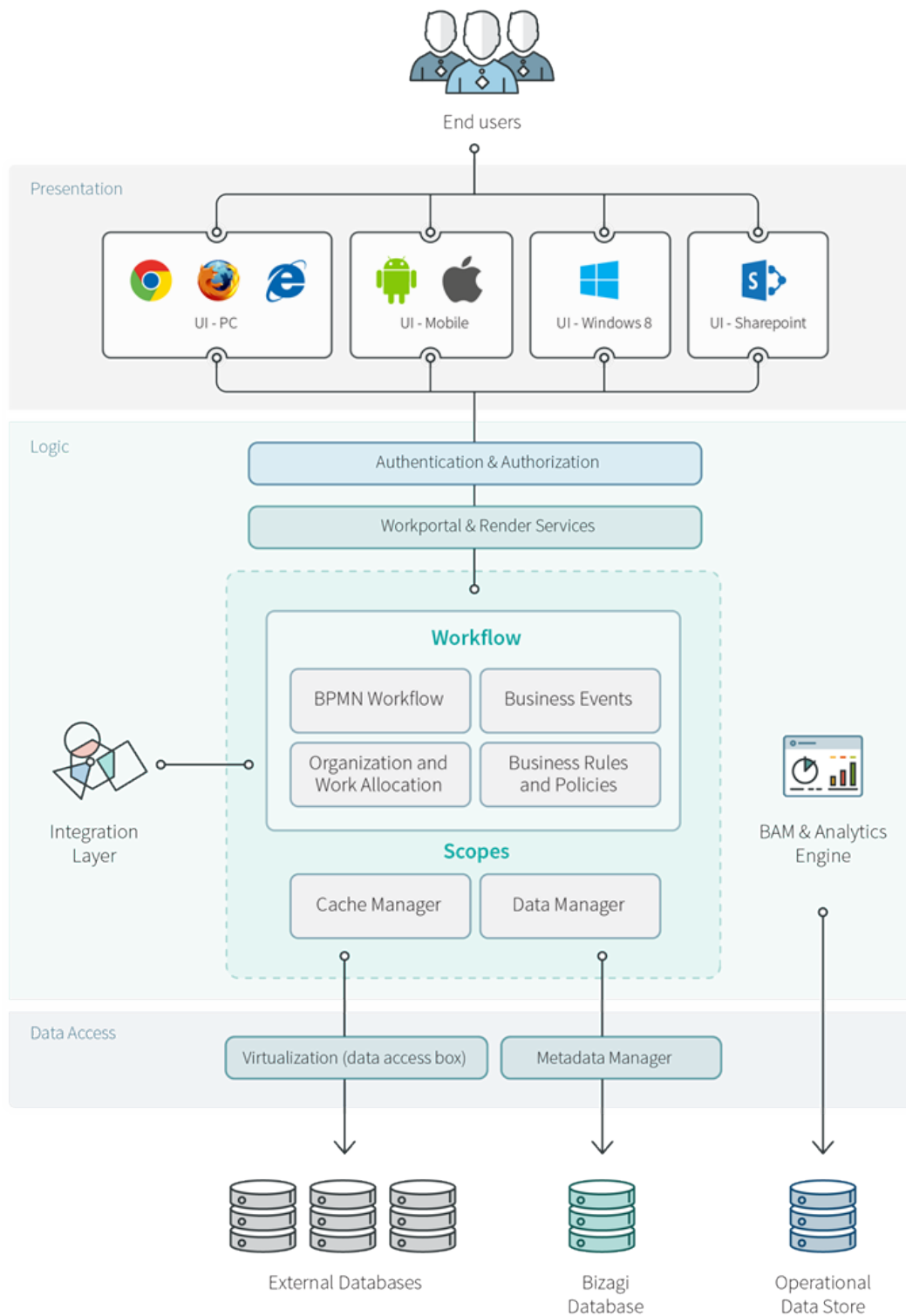


Figura B.40 Arquitectura Bizagi Suite BPM [32]

Funcionalidad y componentes De acuerdo a la ilustración B.40, las funcionalidades y características de Bizagi se soportan a través de los componentes principales que se listan a continuación.

Portal de trabajo El Portal de Trabajo es el punto de acceso de los usuarios finales donde podrán interactuar con el sistema (por ejemplo, podrán empezar nuevos procesos, realizar tareas, acceder a reportes, etc.). El acceso al Portal de Trabajo requiere únicamente de un navegador, y éste puede ser configurado para Intranet o para Internet, tener soporte multi-idomas y realizar personalizaciones.

Motor de flujo de trabajo BPMN El Motor BPMN es conocido en Bizagi como el “WorkflowEngine”, y controla y ejecuta el flujo de proceso. Este motor interpreta directamente el modelo BPMN como es creado previamente desde *Bizagi Modeler*. Esto incluye todas las funcionalidades sofisticadas de la notación BPM tales como los elementos de colaboración entre procesos como señales y mensajes, transacciones de larga duración, diseño de subprocessos, entre otros.

BAM y Análisis Las opciones de análisis y BAM proveen una poderosa opción de reportes para indicadores de gestión (para monitoreo, control y mejoramiento de los procesos). Ambas opciones tienen la posibilidad de un análisis gráfico, BAM (análisis en tiempo real) y de análisis histórico (identificación de tendencias, cuellos de botella, etc.). Para las aplicaciones donde hay un gran volumen de datos y consultas de análisis se presenta la opción de utilizar un ODS (Operational Data Store) para el uso de este módulo, y de esta manera mejorar el rendimiento

Administrador de datos Bizagi presenta un potente motor de datos que se conoce como el ‘EntityManager’. Este motor administra la información de manera segura, consistente y sincronizada entre sus diferentes componentes, permitiendo el acceso a los datos desde el proceso.

Seguridad y Organización El módulo de seguridad maneja las opciones de autenticación y autorización para los diferentes roles, habilidades, posiciones, áreas y ubicaciones definidas en la estructura de la organización. Las opciones de Autenticación incluyen Autenticación Windows o Integración LDAP, entre otras.

Reglas de negocio Bizagi ofrece un motor para la ejecución de reglas de negocio o políticas que dan la lógica al proceso de negocio.

Este motor se ejecuta en *Bizagi Engine* para permitir las diferentes transiciones en el flujo del proceso, controlar y sincronizar la ejecución de las actividades, realizar decisiones de forma automática y ejecutar reglas sofisticadas (por ejemplo tablas de decisión, árboles

de decisión) y realizar cálculos, entre otras. Las reglas de negocio se definen en Bizagi de manera asistida, con la posibilidad de un editor gráfico que permita su inclusión sin la necesidad de programación.

Administración de Carga de Trabajo Bizagi permite definir el balanceo de carga de trabajo en la asignación de actividades de acuerdo a los algoritmos principales como: colas, round robin y balance por carga. Para la administración y optimización de la carga de trabajo, es importante tener en cuenta algunos criterios como calendario laboral y calendario de vacaciones de los usuarios, y el uso de delegación de trabajo y reasignaciones.

Manejador de Caché El manejador de Caché es una implementación importante especialmente en proyectos de Misión crítica de clientes corporativos grandes, estos procesos involucran generalmente un gran número de usuarios finales (de forma simultanea). El manejador de Caché provee una mejora en el rendimiento del Portal de Trabajo mediante la optimización al acceso de datos y la persistencia en la base de datos mediante scopes [32].

B.9.1.2. Compatibilidad con estándares

Es posible importar/exportar procesos en formato XPDL, Visio file. Desde el modelador se puede exportar el modelo en formato bpm, jpg, png y svg.

Además Bizagi permite generar documentación en diferentes formatos tales como doc, pdf, wiki, web y sharepoint.

B.9.1.3. Base de datos

Los motores de base de datos soportados por Bizagi son: Microsoft SQL Server y Oracle.

Para la evaluación se probó con Microsoft SQL Server 2012.

B.9.1.4. Sistema operativo del modelador

Bizagi cuenta con una aplicación de escritorio para modelar un proceso.

Los sistemas operativos en los que puede ejecutar el modelador dependen de la edición de este:

- Edición Enterprise .NET: Windows XP, Windows Vista, Windows 7 y Windows 8.
- Edición Enterprise JEE: Unix - Solaris, Linux y Windows.
- Edición Xpress: Windows XP, Windows Vista, Windows 7 y Windows 8.

B.9.1.5. Navegador web

Bizagi se puede utilizar en los navegadores:

- Google Chrome, últimas versiones.
- Mozilla Firefox, últimas versiones.
- Internet Explorer, últimas versiones.
- Microsoft Edge.

Ademas permite el ingreso al portal desde Celulares y Tablets a través de su aplicación móvil.

B.9.1.6. Extensibilidad

Bizagi permite agregar código propio e integrar APIs.

B.9.1.7. Tipo de Modelador

El modelador de procesos de la herramienta es de escritorio.

B.9.1.8. Notación gráfica de procesos compatible con BPMN 2.0

Bizagi soporta el estándar BPMN 2.0 para el modelado

B.9.2. Evaluación de las características prácticas**B.9.2.1. Integración con servicios web**

Bizagi permite la integración con servicios web externos. Se deberá ingresar la URL en donde está publicado el servicio para que Bizagi identifique los parámetros requeridos y el resultado de la operación.

Se deben de ingresar los siguientes datos:

1. Primero se debe de indicar los datos que se van a enviar como parámetros, esto se hace a partir de la metadata del proceso.
2. En segunda instancia se debe vincular el resultado a alguna variable del proceso para poder tener el dato en la instancia que se esta ejecutando.

B.9.2.2. Interfaz gráfica para key users

Los procesos básicos pueden ser diseñados por usuarios sin conocimientos técnicos. Al agregar tarea mas complejas como interacción con servicios externos o formularios dinámicos se requiere de un usuario con una capacitación en estos temas.

B.9.2.3. Definición de formularios

Bizagi permite la definición de formularios web. Se utiliza drag-and-drop y permite adjuntarles reglas de llenado, validación y control de campos. Además se puede asociar las variables a los controles.

Entre otras funcionalidades se destaca la facilidad de copiar otros formularios de otras tareas.

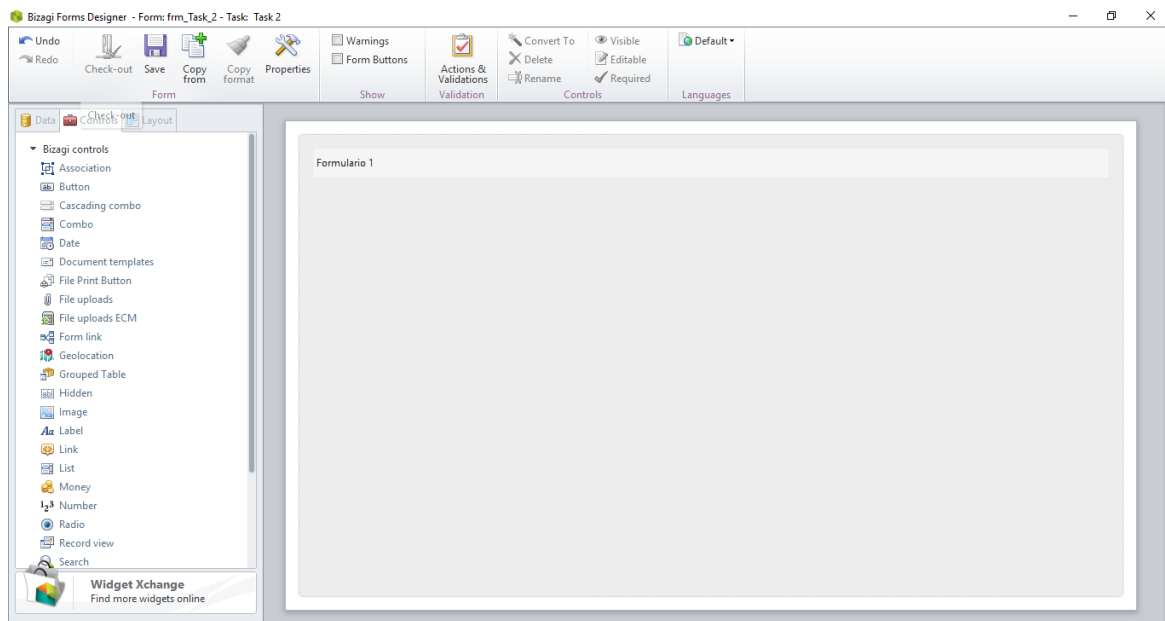


Figura B.41 Diseño de Formularios en Bizagi

B.9.2.4. Formato de formularios

Los diferentes formatos que Bizagi presenta son:

- Orientación
- Alineación
- Formatos de letra: itálica, negrita, tamaño.
- Colores

B.9.2.5. Elementos de formularios

Bizagi cuenta con formularios con listas desplegables, estos datos se pueden tomar de diferentes fuentes: variables de proceso, base de datos externa, ingresados por el usuario en el momento de creación, etc.

De cada elemento de los formularios se puede definir su visibilidad y si es requerido o no, además se pueden definir botones de envío de formulario con cierta lógica asociada.

B.9.2.6. Formularios dinámicos

Bizagi no permite definir valores que dependan de otro dato de ese mismo formulario. Se puede condicionar un valor de acuerdo a una variable de proceso.

B.9.2.7. Reglas de llenado de campos

Bizagi permite definir reglas de llenado para la actualización automática de campos a partir de la carga de alguna variable:

- Mediante el acceso a la base de datos.
- Acceso a aplicaciones externas mediante el llamado a un servicio web.

- Cálculos algorítmicos definidos por el usuario. Además se pueden ingresar expresiones de código que permiten manipular los valores.
- Campos completados en otros formularios y variables del proceso.

B.9.2.8. Validación de campos de formularios

Bizagi cuenta con validación para los datos ingresados para cada control del formulario. Además permite definir expresiones de código para realizar controles definidos por el usuario. Por defecto tiene validaciones de filtros, controles de extensión para los archivos, etc.

Por otro lado, permite validar cada campo del formulario según el tipo (long, string, date, boolean, etc)

B.9.2.9. Iniciación de procesos en forma manual y automática

Bizagi permite iniciar sus procesos tanto de forma manual como automática. Cuenta con métodos en la API REST para la iniciación de los procesos.

B.9.2.10. Definición de reglas de negocio

Bizagi posee una etapa en el ciclo de vida tal como se muestra en la imagen a continuación, que permite la definición de reglas de negocio. Esta etapa se llama Business Rules, y se utiliza para asegurar el cumplimiento de las restricciones que cada proceso de negocio posee. Permitiendo definir scripts embebidos en cualquier actividad que permite mediante código definir lo necesario, como modificar variables, plazos, obtener información, etc.



Figura B.42 Definir regla de negocio en Bizagi

Esto se define desde la interfaz gráfica de una forma muy amigable que permite que un usuario sin conocimientos técnicos avanzados pueda definir correctamente las reglas de negocio.

B.9.2.11. Asignación de tareas

Bizagi permite la asignación de tareas de forma manual o automática. Es posible asignar una tarea a un grupo de usuarios, esto funciona de manera que cuando la tarea es tomada por un usuario de ese grupo al que fue asignada la tarea, se desasigna automáticamente del resto de los usuarios del grupo. En el ciclo de vida de Bizagi posee una etapa en el que se pueden definir para cada tarea de usuario quienes son los participantes. Permitiendo utilizar distintos métodos de asignación, como por ejemplo “Por carga” o “Primero disponible”.

B.9.2.12. Reasignación de tareas

Bizagi permite la reasignación de tareas siempre que el usuario tenga permisos para ejecutar esta acción.

B.9.2.13. Adjuntar documentos a tareas

Existen varias formas de manejar documentos durante la ejecución de los procesos

- Se definen en la metadata variables de tipo “.archivo”, las cuales se manejan de la misma manera que los tipos primitivos. Se puede establecer que estos sean solo de descarga.
- Se permite la visualización de archivos mediante lo llamado letter, esto son plantillas predefinidas.
- Por ultimo se propone manejar documentos de forma global

B.9.2.14. Versionado de procesos

Bizagi cuenta con versionado de procesos, este permite seleccionar cuál se encuentra activo mediante un check. Según la documentación existe la posibilidad de realizar deploy de versiones y ejecuciones al mismo tiempo, esto no se puede verificar sin tener el ambiente de producción.

B.9.2.15. Tareas de borrador

Según lo investigado bizagi no cuenta con la opción de grabar temporalmente un tarea con los datos ingresados.

B.9.2.16. Gestión de instancias por usuarios

La herramienta permite a los usuarios crear, guardar, borrar y abortar. Sin embargo, no es posible pausar o reanudar, reinstalar ni sobrescribir datos de la instancia.

B.9.2.17. Mantenimiento del histórico del proceso

Bizagi cuenta con un mantenimiento histórico del proceso durante la vida activa del mismo. Permite visualizar de forma simple, gráfica y cronológica las actividades de un proceso.

B.9.2.18. Monitoreo de procesos

En la documentación se encuentra una sección dedicada al monitoreo de procesos, denominada “Análisis del Proceso”. En el Portal provee una pestaña de “Reportes”. Mediante

estas dos formas se puede realizar monitoreo a partir de datos históricos y de tiempo real por ejemplo:

- BAM
- Análisis de procesos
- Sensores

B.9.2.19. Presentación de las tareas

Respecto presentación de las tareas Bizagi permite seleccionar y listar tareas favoritas, ver tareas retrasadas o expiradas, etc. Además permite ordenar en forma ascendente o descendente las diferentes columnas que se muestra y editar que columnas a mostrar ni el orden de las mismas.

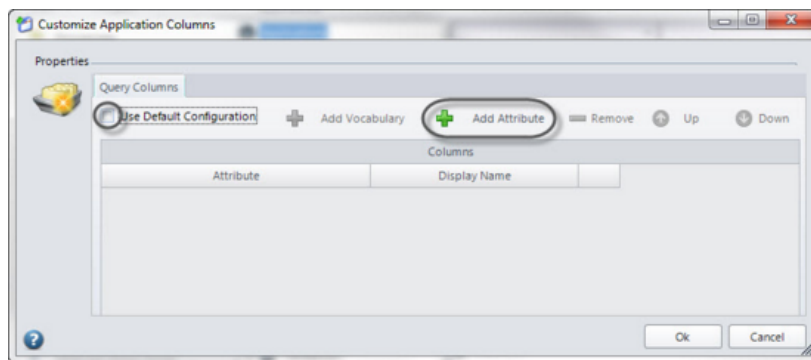


Figura B.43 Edición de Columnas en Bizagi

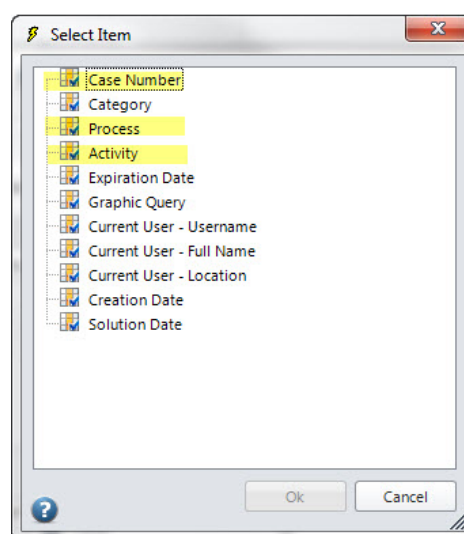


Figura B.44 Selección de Columnas en Bizagi

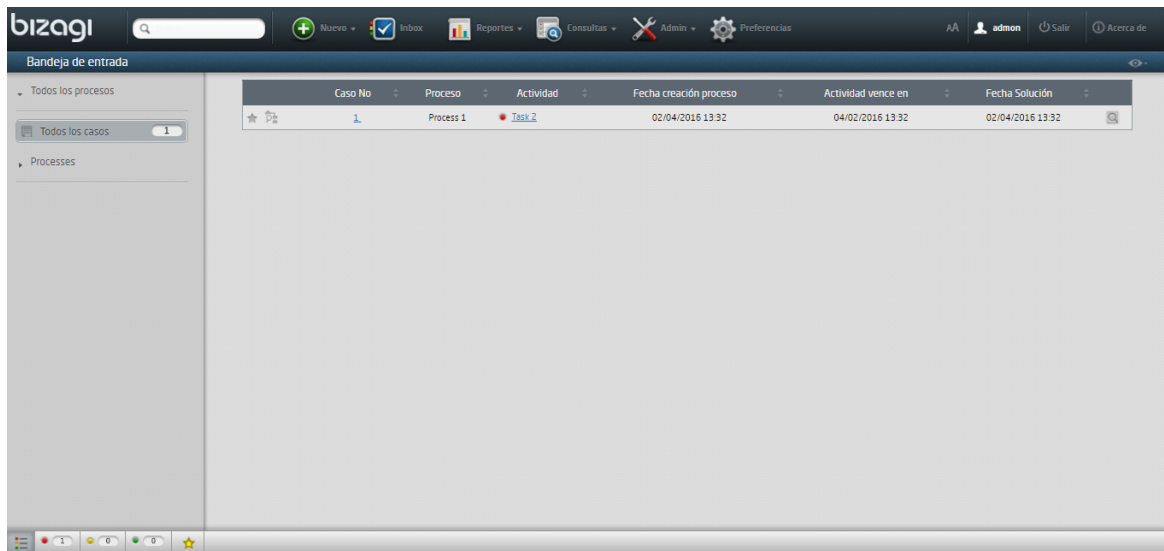


Figura B.45 Presentación de Tareas en Bizagi

B.9.2.20. Capacidad de aprendizaje

En el sitio de descarga se ofrece una documentación amplia pero a la vez se puede encontrar algún tema específico rápidamente gracias a su presentación. Tiene una búsqueda del tipo "like" que se encarga de buscar en toda la documentación los temas que hacen referencia a lo que se introduce en ese campo de texto. Además se ofrecen tutoriales en videos en diversos idiomas, depende del idioma que se elija la cantidad de temas que se muestran. La mayoría de los temas tienen su correspondiente tutorial en inglés. En los demás idiomas solo se encuentran tutoriales de temas más básicos.

Bizagi modeler permite a un usuario sin conocimientos técnicos poder modelar los procesos de negocio, con tooltips que ayudan y guían al usuario.

Bizagi Studio introduce un poco más de complejidad en algunos temas como web services, para el cual hay que tener cierto conocimiento para poder realizar la tarea.

Los mensajes de error tanto en el modelador como en el studio son descriptivos.

Tabla B.61 Evaluación Bizagi - Capacidad de aprendizaje

	Modelador	Portal
¿Posee tooltips?	SI	SI
¿Tiene mensajes de error descriptivos?	SI	SI
¿Brinda sugerencia de posible solución ante un error?	NO	NO
¿Posee manuales de uso?	SI	SI

B.9.2.21. Capacidad para ser usado

Tanto el modelador de Bizagi como el Studio tienen menú de ayuda, con links Tutoriales, Foros, etc. Sin embargo no cuenta con un link al manual de uso.

En la documentación tiene un espacio de preguntas frecuentes para el usuario. Para facilitar la interacción con el usuario Bizagi tanto en el modeler como en el studio y el portal posee símbolos mnemotécnicos.

Tabla B.62 Evaluación Bizagi - Capacidad para ser usado

	Modelador	Portal
¿Brinda menú de ayuda?	SI	SI
¿Tiene símbolos mnemotécnicos?	SI	SI
¿Brinda un espacio de preguntas frecuentes?	SI	SI
¿Posee acceso directo a manuales de uso?	NO	NO

B.9.2.22. Protección contra errores de usuarios

Para las tareas con formularios asociados es posible agregar validación sobre los elementos del formulario, por ejemplo, se puede validar el tipo de campo. Para deshacer o rehacer una acción tanto en el studio como el modeler se puede utilizar $Ctrl + Z$ y $Ctrl + Y$.

Desde el portal es posible validar los datos de un formulario. Cuenta con mensajes de advertencia y error. Dentro del portal no es posible deshacer o rehacer una acción.

Tabla B.63 Evaluación - Protección contra errores de usuarios

	Modelador	Portal
¿Tiene mensajes de advertencia?	SI	SI
¿Posee validación de datos y formularios?	SI	SI
¿Cuenta con opciones para deshacer y rehacer una acción?	SI	NO

B.9.2.23. Estética de la interfaz de usuario

La información relevante dentro del portal, modeler y studio es accesible y fácil de encontrar. La información que muestra en las 3 aplicaciones no es innecesaria.

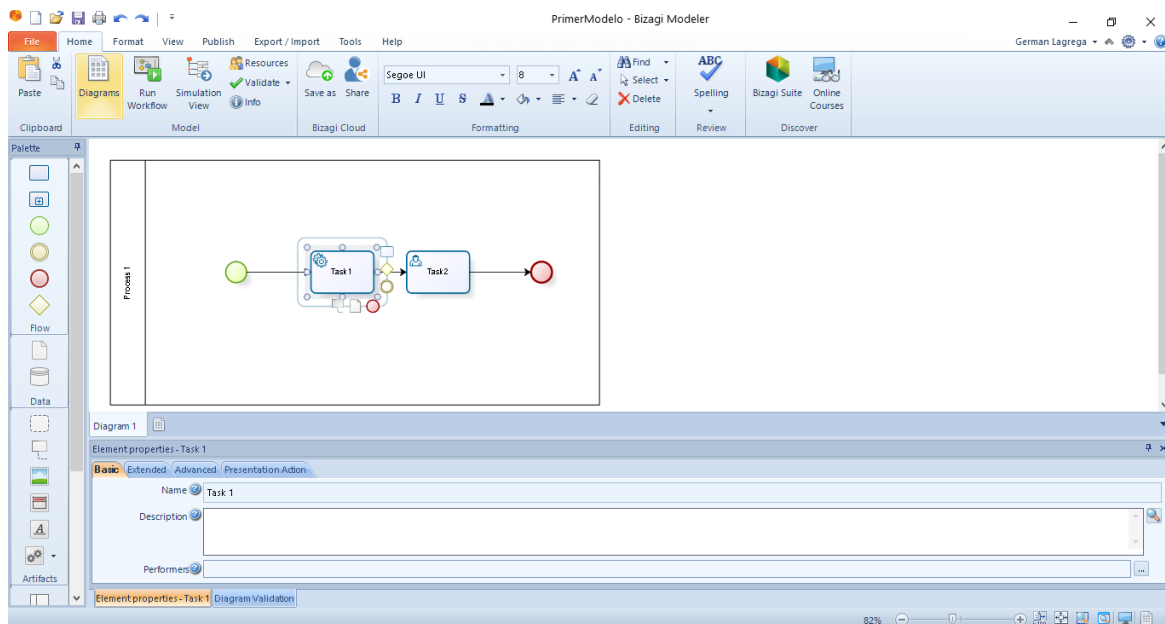


Figura B.46 Modeler de Bizagi

Tabla B.64 Evaluación Bizagi - Estética de la interfaz de usuario

	Modelador	Portal
¿Incluye información innecesario?	NO	NO
La información relevante ¿Se encuentra accesible y fácil de encontrar?	SI	SI

B.9.2.24. Accesibilidad

Las acciones tanto en el modelador, studio y en la aplicación web se representan con imágenes que se manejan en los estándares. Los colores son distinguibles en ambas aplicaciones de escritorio como dentro de la aplicación web. Sin embargo se utiliza el rojo y el verde en muchas imágenes, colores que pueden confundirse para usuarios daltónicos.

Para ninguna de las aplicaciones se cuenta con comando de voz y no tienen sonidos.

Tabla B.65 Evaluación - Accesibilidad

	Modelador	Portal
¿Incluye comando de voz?	NO	NO
¿Las acciones se encuentran representadas con imágenes adecuadas?	SI	SI
¿Los colores son distinguibles aunque el usuario sea daltónico?	NO	NO
¿Los sonidos se corresponden con la acción que se está ejecutando?	NO	NO

B.9.2.25. Confidencialidad

Las contraseñas se guardan cifradas en la base de datos. Bizagi cuenta con una aplicación mobile que no guarda información sensible del usuario en el dispositivo. Los intercambios de datos que realiza con un servidor no son sobre HTTPS.

Tabla B.66 Evaluación Bizagi - Confidencialidad

	Portal
¿Utiliza técnicas criptográficas?	SI
Mobile ¿Almacena los datos cifrados en el dispositivo?	NO
¿Los intercambios de datos con el servidor son sobre HTTPS?	NO

B.9.2.26. Integridad

Un usuario administrador es capaz de crear nuevos roles y áreas, a partir de esto puede asignar permisos.

Los documentos se guardan en un directorio dentro del espacio de trabajo. Estos no se guardan cifrados y además no es posible asignar permisos sobre ellos.

El portal cuenta con un tiempo limite de sesión activa, el cual es configurable.

Tabla B.67 Evaluación - Integridad

	Portal
¿Permite definir roles?	SI
¿Cuenta con acceso diferenciado a los documentos según el rol?	SI
¿Permite visualizar diferentes funciones según el rol?	SI
¿Permite administrar permisos?	SI
¿Los documentos se guardan cifrados?	NO
¿Cuenta con un tiempo limite de sesión activa?	SI

B.9.2.27. No repudio

Bizagi cuenta con un historial de ejecución. Se muestra un registro de cada vez que se realizó la actividad. Permite visualizar de forma simple, gráfica y cronológica las actividades de un proceso.

Tabla B.68 Evaluación - No repudio

	Portal
¿Cuenta con un historial de ejecución de procesos?	SI
¿Se puede identificar que usuario y en que momento se ejecutó un proceso?	SI

B.9.2.28. Responsabilidad

Las tareas y los procesos cuentan con un identificador numérico que es único.

Tabla B.69 Evaluación Bizagi - Responsabilidad

	Portal
¿Las tareas y los procesos tienen identificador único?	SI

B.9.2.29. Autenticidad

Bizagi provee su propia forma de autenticación, pero también permite integrar a un LDAP o algún otro tipo de autenticación externa.

Tabla B.70 Evaluación Bizagi- Autenticidad

	Portal
¿Cuenta con alguna forma de autenticación de usuarios?	SI
¿Soporta firmas digitales?	SI

B.9.3. Tabla de nivel de cumplimiento

La tabla B.71 muestra los resultados obtenidos para las características funcionales. Las características están divididas por categoría. De cada característica se muestra si la evaluación es teórica (Teó), el nivel de soporte (NS) y el nivel de cumplimiento (NC).

Tabla B.71 Características técnicas - Nivel de cumplimiento.

Modulo	Característica	Teó	NS	NC
	Tecnología, Arquitectura e Interoperabilidad			
	Arquitectura de BPMS	Si		
	Compatibilidad con estándares	Si		
	Base de datos	Si		
	Sistema operativo del modelador	Si		
	Navegador Web	Si		
	Extensibilidad	Si		

Modulo	Característica	Teó	NS	NC
	Integración con Servicios Web	No	●	✓
	Diseño y modelado de procesos			
	Interfaz gráfica para key users	No	●	✓
	Tipo de modelador	Si	●	✓
	Notación gráfica de procesos compatible con BPMN 2.0	Si	●	✓
	Gestión de formularios			
	Definición de formularios	No	●	✓
	Formato de formularios	No	●	✓
	Elementos de formularios	No	●	✓
	Formularios dinámicos	No	●	✓
	Reglas de llenado de campos	No	●	✗
	Validación de datos del formulario	No	●	✓
	Motor de Workflow			
	Iniciación de procesos en forma manual y automática	No	●	"⊘"
	Definición de reglas de negocio	No	●	✓
	Asignación de tareas	No	●	✓
	Reasignación de tareas	No	●	✓
	Adjuntar documentos a tareas	No	●	✓
	Versionado de procesos	No	●	✓
	Tareas de borrador	No	●	●
	Gestión, monitoreo y auditoría			
	Gestión de instancias por usuarios	No	●	✓
	Mantenimiento histórico de procesos	No	●	✓
	Monitoreo de procesos	No	●	✓
	Portal			
	Presentación de las tareas	No	●	✓

Referencias:

-  No aplica
-  Cumple de forma nativa
-  Cumple vía integración
-  Cumple vía particularización
-  Soporta totalmente
-  Soporta parcialmente
-  No soportada

B.9.3.1. Resultado numérico de las características técnicas

Las tablas B.73 y B.74 presentan los resultados numéricos correspondientes a las características funcionales y no funcionales evaluadas. En la sección 3.7 se describe el método de evaluación, donde se presentan los cálculos a realizar para poder clasificar a las características según los umbrales definidos.

Primero, hay que definir los valores de importancia, para poder definir los umbrales. Además, es necesario dar valores a los niveles de cumplimiento y de soporte.

Tabla B.72 Definición de constantes

Importancia		Nivel de cumplimiento		Nivel de soporte	
Alto	3	Nativo	2	Total	3
Medio	2	Vía integración	1	Parcial	2
Bajo	1	Vía particularización	1	No soportado	1

Las columnas de la tabla B.73 contienen: las características, importancia (I), nivel de soporte (NS), nivel de cumplimiento (NC), el factor (F) y el puntaje total (Total). El puntaje total se calcula de la siguiente manera:

Puntaje total

$$Puntaje = I * NS * NC * Factor \quad (B.11)$$

Donde I es la importancia, NS el nivel de soporte, NC el nivel de cumplimiento y Factor un valor entre 0,1 y 0,9.

Tabla B.73 Características técnicas - Cálculo del puntaje.

Característica	I	NS	NC	F	Total
Tecnología, Arquitectura e Interoperabilidad					
	1	0	0	1	0
Compatibilidad con estándares	1	1	2	1	2
Base de datos	1	1	2	1	2
Sistema operativo del modelador	1	2	2	1	4
Navegador Web	1	2	2	1	4
Extensibilidad	1	2	2	1	4
Integración con Servicios Web	1	2	2	1	4
Diseño y modelado de procesos					
Interfaz gráfica para key users	1	2	2	1	4
Tipo de modelador	1	1	2	1	2
Notación gráfica de procesos compatible con BPMN 2.0	1	2	2	1	4
Gestión de formularios					
Definición de formularios	1	2	2	1	4
Formato de formularios	1	1	2	1	2
Elementos de formularios	1	1	2	1	2
Formularios dinámicos	1	2	2	1	4
Reglas de llenado de campos	1	2	1	0,6	1,2
Validación de datos del formulario	1	2	2	1	4
Motor de Workflow					
Iniciación de procesos en forma manual y automática	1				1
Definición de reglas de negocio	1	2	2	1	4
Asignación de tareas	1	2	2	1	4
Reasignación de tareas	1	2	2	1	4
Adjuntar documentos a tareas	1	1	2	1	2
Versionado de procesos	1	2	2	1	4
Tareas de borrador	1	0	0	1	0
Gestión, monitoreo y auditoría					
Gestión de instancias por usuarios	1	1	2	1	2
Mantenimiento histórico de procesos	1	1	2	1	2
Monitoreo de procesos	1	1	2	1	2
Portal					
Presentación de las tareas	1	1	2	1	2

Por otro lado tenemos los cálculos de las características no funcionales. Las columnas de la tabla B.74 contienen: el peso de cada subcaracterística (P_i), el nivel de cumplimiento de la subcaracterística (s_i), el valor final de la subcaracterística definido en la ecuación B.18 y el nivel de cumplimiento (NC). El nivel de cumplimiento se obtiene de la siguiente manera:

- $vf < 1 \rightarrow$ Nivel de cumplimiento Bajo.
- $1 \leq vf < 2 \rightarrow$ Nivel de cumplimiento Medio.
- $3 \leq vf \rightarrow$ Nivel de cumplimiento Alto.

Valor final

$$vf = \frac{\sum_{i=1}^N (s_i * P_i)}{\sum_{i=1}^N P_i} \quad (B.12)$$

Siendo vf el ‘valor final’ de la característica y N la cantidad de subcaracterísticas.

Se considera que todas las subcaracterísticas son importantes y en este sentido se asigna el valor 1 al peso de cada subcaracterística.

B.10. Evaluación de Camunda BPM

Es una herramienta *open source* que utiliza el estándar BPMN 2.0.

B.10.1. Evaluación de las características teóricas

B.10.1.1. Arquitectura del BPMS

Camunda es una plataforma Open Source para la automatización de procesos de negocios. Es ejecutable en BPMN 2.0, es liviana y escala fácilmente.

Camunda está basado en Java. Sus principales componentes están escritos en Java y provee a los programadores Java herramientas para diseñar, implementar y ejecutar procesos de negocio y workflows en JVM. Por otro lado, permite a los programadores que no utilizan Java acceder al motor de procesos a través del REST API, el cual permite que las aplicaciones se conecten de forma remota al motor de procesos.

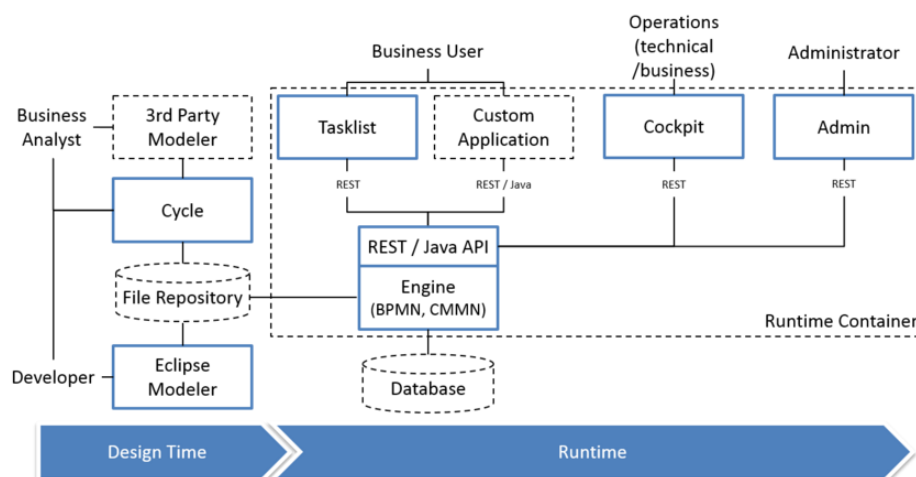


Figura B.47 Arquitectura Camunda BPM [33]

Camunda BPM puede ser utilizado tanto como un servicio de procesos de negocio independiente, así como también embebido dentro de una aplicación Java ya existente. La figura 4.11 muestra los componentes más importantes al momento de modelar, así como también en tiempo de ejecución.

Motor de proceso e infraestructura

Motor de proceso es una librería de java responsable de la ejecución de workflows y procesos BPMN 2.0. Tiene un núcleo POJO¹ ligero y utiliza una base relacional para la persistencia. La figura B.48 muestra la arquitectura del motor de procesos.

¹Plain Old Java Object, enfatiza el hecho que las clases java utilizadas son simples y no dependen de un framework.

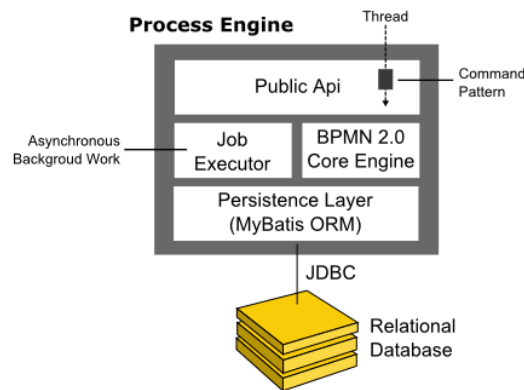


Figura B.48 Arquitectura del motor de procesos

- **Public API** es un servicio que permite a las aplicaciones Java comunicarse con el motor de proceso. Las diferentes responsabilidades que tiene el motor de procesos (administrar tareas, la interacción del proceso en tiempo de ejecución, el repositorio del proceso, etc.) están separadas en servicios individuales. Los hilos de entrada al motor de procesos están enrutados a través de un interceptor de Camunda el cual es utilizado para configurar el contexto del hilo como una transacción.
- **BPMN 2.0 Core Engine** es el núcleo del motor del proceso. Se caracteriza por ser un motor de ejecución ligero para estructuras gráficas (PVM - Process Virtual Machine), un analizador sintáctico de BPMN 2.0 es utilizado para transformar archivos BPMN 2.0 XML en Java Objects, además provee la implementación de un conjunto de comportamientos de BPMN, por ejemplo, constructores como los Gateway o los Service Task.
- **Job Executor** es responsable de procesar de forma asincrónica el trabajo de fondo, por ejemplo, los Timers.
- **The persistence layer** es la capa encargada de guardar los datos de las instancias de los procesos en una base de datos relacional.

Spring Framework Integration es un núcleo que da soporte para inserción de dependencias, manejo de transacciones, aplicaciones web, acceso a datos, etc. La integración se encuentra dentro del módulo camunda-engine-spring y puede ser agregado a cualquier proyecto apache basado en maven.

CDI/Java EE Integration El módulo camunda-engine-cdi permite la integración de modelos de programación con CDI (Context and Dependency Injection).

Runtime Container Integration permite la integración con infraestructuras de servidor de aplicaciones.

Aplicaciones web

REST API permite utilizar el motor de procesos desde una aplicación remota o una aplicación JavaScript.

Camunda Tasklist es una aplicación web para el manejo del flujo de trabajo y tareas de usuario que permite a los participantes de los procesos ver sus tareas dentro del flujo de trabajo y navegar por los formularios de las tareas para producir datos de salida.

Camunda Cockpit es una aplicación web para el monitoreo y operación de procesos que permite buscar instancias de proceso, ver sus estados y reparar instancias.

Camunda Admin es una aplicación web para el manejo de usuarios que permite administrar usuarios, grupos y autorizaciones.

Camunda Cycle es una aplicación web para la sincronización de modelos de proceso BPMN 2.0 entre diferentes herramientas de modelado y modeladores.

Herramientas adicionales

Camunda Modeler es un plugin para Eclipse para modelado de procesos.

bpmn.io es un modelador web el cual es utilizado por las aplicaciones web Cockpit y Tasklist para mostrar los modelos de proceso en el navegador.

B.10.1.2. Compatibilidad con estándares

Es posible importar/exportar procesos en formato BPMN. Desde el modelador se puede exportar el modelo en formato bpm, gif, jpg, png y rle.

B.10.1.3. Base de datos

Soporta conexión con las siguientes bases de datos:

- MySQL, versiones: 5.1, 5.5 y 5.6.
- MariaDB, versión: 10.0.
- Oracle 10g, versiones: 11g y 12c.
- IBM DB2, versiones: 9.7, 10.1 y 10.5 (excluyendo IBM z/OS para todas las versiones).
- PostgreSQL, versiones: 9.1, 9.3 y 9.4.
- Microsoft SQL Server, versiones: 2008 R2, 2012 y 2014.
- H2, versión: 1.3.

Camunda provee scripts para crear, borrar y actualizar la bases de datos H2, PostgreSQL, MySQL, Microsoft SQL Server y Oracle.

B.10.1.4. Sistema operativo del modelador

Camunda cuenta con tres formas de modelar un proceso, un modelador online bpmn.io, un modelador de escritorio propio y provee un plugin para Eclipse (hasta la versión 7.3).

El modelador se puede utilizar en los sistemas operativos Window, Linux y Mac Os, en las versiones que soportan alguna de las siguientes aplicaciones:

- Java, versiones: 6 y 7.
- Java 8 (si es soportada por el servidor).
- Java Runtime.
- Sun/Oracle Hot Spot, versiones: 6, 7 y 8.
- IBM® J9 virtual machine (JVM), versiones: 6, 7 y 8.
- OpenJDK, versiones: 6 y 7.
- Oracle JRockit 6 - R28.2.7

Para la evaluación se utilizó el plugin para Eclipse Kepler, que provee el sitio de Camunda. Ver [59]

B.10.1.5. Navegador web

Los siguiente navegadores son soportados por la herramienta:

- Google Chrome, últimas versiones.
- Mozilla Firefox, últimas versiones.
- Internet Explorer, versiones: 9, 10 y 11.

Se ha demostrado un correcto funcionamiento en los navegadores:

- Mozilla Firefox, versión: 42.0 en Windows 10.
- Google Chrome, versión: 46.0 en Windows 10.

B.10.1.6. Extensibilidad

A través de diferentes plugin, Camunda es capaz de extender sus funcionalidades. Por ejemplo, se puede integrar con XPD, con Spring Framework y con Drools, agregando las dependencias de cada uno en el archivo pom.xml que se encuentra en el proyecto.

B.10.1.7. Tipo de modelador

Hay dos tipos de modeladores para modelar un proceso en Camunda, aplicaciones web y de escritorio. Camunda cuenta con una aplicación de escritorio propia en la cual solamente se puede modelar, otra opción para el escritorio es un plugin para las versiones de Eclipse Indigo, Juno y Kepler, que solo está disponible hasta la versión 7.3 de Camunda. Por otra parte, se recomienda la aplicación web bpmn.io que se utiliza dentro de las aplicaciones web Tasklist y Cocpit.

Hay una cuarta opción para modelar a través de Camunda Cycle y Roundtrip, estas dos herramientas que en conjunto permiten modelar por ejemplo, en Aris, Bizagi, Yaoqiang, Signavio Process Editor, etc. Por más información ver [60].

Para la evaluación se utilizó el plugin en Eclipse Kepler [59].

B.10.1.8. Notación gráfica de procesos compatible con BPMN 2.0

Los elementos marcados en naranja en la figura B.49 son parte de los elementos del estándar BPMN 2.0 que actualmente soporta por Camunda BPM. En [34] se puede ver la lista completa.

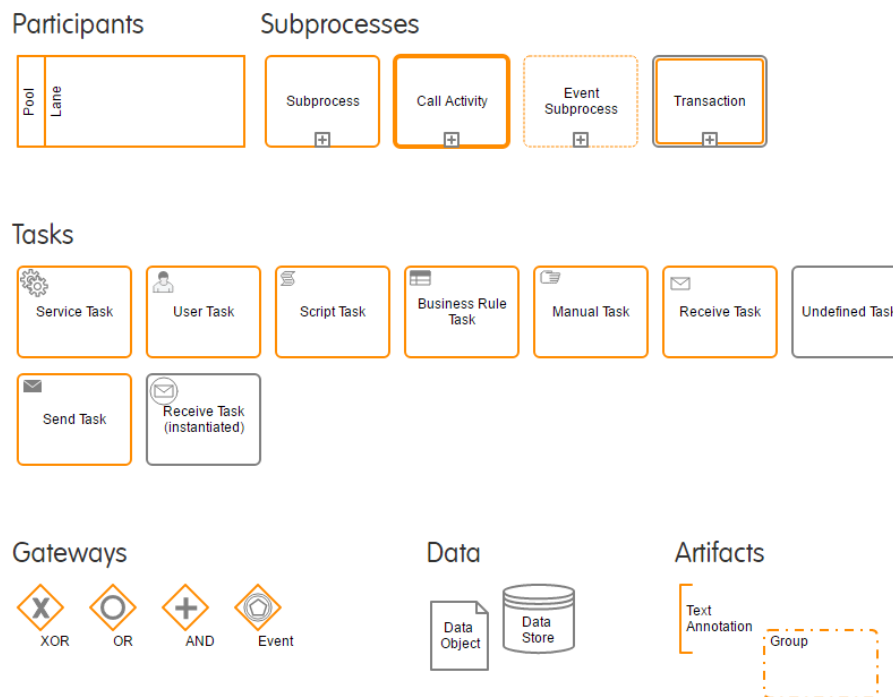


Figura B.49 Elementos del estándar BPMN 2.0

B.10.2. Evaluación de las características prácticas

B.10.2.1. Integración con servicios web

A través de una tarea de tipo *Service Task*, se puede asociar una clase java que implementa *JavaDelegate*, en la cual se puede llamar a un servicio web de tipo SOAP o WSDL con JAX-WS.

[34] muestra la comunicación entre dos procesos mediante servicios web.

Para la evaluación se utilizó una conexión de tipo SOAP.

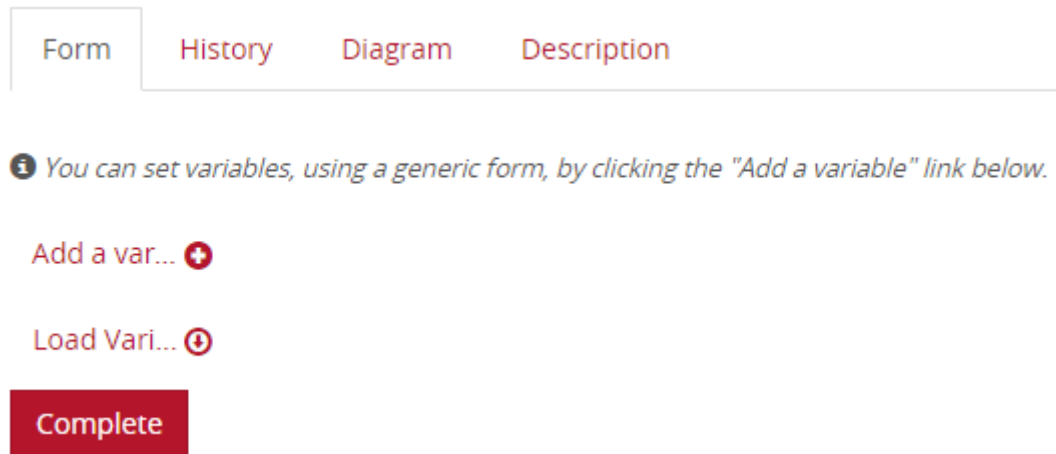
B.10.2.2. Interfaz gráfica para key users

El modelador de Camunda permite que el diseño de alto nivel pueda ser llevado a cabo por un key user. A pesar de esto, si el modelo requiere, por ejemplo, una tarea de tipo *Services Task* o formularios más complejos, es necesario que el mismo lo finalice un usuario con formación técnica.

B.10.2.3. Definición de formularios

Existen cuatro formas de asociar un formulario a una tarea.

- Embebidos: se genera un archivo HTML que contiene el formulario y se indica en la tarea la ruta a dicho archivo. El formulario se muestra en el Tasklist dentro de la tarea. Por más información ver [34].
- Generados: se editan dentro de las propiedades de la tarea, solo se pueden agregar tipos básicos: long, string, date, enum y boolean.
- Externos: el usuario es redirigido a otra aplicación que contiene el formulario.
- Genéricos: si no existe un formulario asociado a la tarea, se muestra un formulario genérico, el cual permite agregar variables del proceso.



Form History Diagram Description

ⓘ You can set variables, using a generic form, by clicking the "Add a variable" link below.

Add a var... +

Load Vari... +

Complete

Figura B.50 Formularios genéricos

En ninguno de los casos se puede utilizar drag-and-drop para agregar elementos al formulario y en los casos de los formularios embebidos y los externos, es necesario la participación de un programador.

B.10.2.4. Formato de formularios

Dentro de un formulario embebido es posible modificar la apariencia del mismo (fuente, estilo, color, alineación, etc). Además los tamaños de los campos se ajustan con los datos. Por otro lado, los formularios de tipo generados o genéricos no permiten modificar su apariencia.

B.10.2.5. Elementos de formularios

Los formularios embebidos permiten incluir listas desplegables cuyos valores se cargan de forma dinámica. Para obtener los datos, por ejemplo de una base de datos, es necesario asociar una clase java a la tarea, dicha clase debe contener no solo la conexión a la base, sino también la carga de los datos en la variable del proceso que está asociada a la lista desplegable.

Con respecto a la definición de botones con acciones predefinidas. Los formularios embebidos permiten la utilización de funciones JavaScript, las mismas deben estar definidas dentro de algún evento, por ejemplo, al cargar el formulario. Es posible, dentro de las funciones, cambiar, obtener o asignar tanto los campos del formulario como las variables del proceso. [34] muestra los diferentes tipos de eventos que provee Camunda BPM cuando está integrado a AngularJS.

B.10.2.6. Formularios dinámicos

Dentro de un formulario embebido se pueden utilizar directivas de AngularJS. Estas directivas permiten ocultar/mostrar campos de un formulario dependiendo del valor de otros campos del formulario. No es posible actualizar los valores de una lista de acuerdo al valor seleccionado en otra lista, aunque permite actualizar un campo de tipo input, por ejemplo, con el valor seleccionado en una lista.

B.10.2.7. Reglas de llenado de campos

Camunda BPM permite definir reglas de llenado para la actualización automática de campos a partir de la carga de algún campo clave, por ejemplo:

- Mediante el acceso a la base de datos, al ingresar la cédula de una persona se puede obtener el email.
- Acceso a aplicaciones externas, cargar un campo con el resultado de una consulta a un servicio web.
- A través de un cálculo que definió el usuario.
- Mediante campos completados en otros formularios, así como variables del proceso.

Los ejemplos antes mencionados necesitan la participación de un programador, ya que se realizan a través de clases java que se asocian a las distintas tareas.

B.10.2.8. Validación de campos de formularios

Es posible indicar en un formulario cuando un campo es requerido. Cuando un usuario ingresa a una tarea que tiene un formulario con un campo requerido, no tendrá la posibilidad de completar el formulario, hasta que el campo requerido se haya completado, ver figura B.51.

Form History Diagram Description

A

aaa

B

C Admin ▼

D Server ▼

Save Complete

Figura B.51 Dato requerido en un formulario

Por otro lado, permite validar cada campo del formulario según el tipo (long, string, date, boolean, etc), ver figura B.52. Además permite agregar restricciones en la cantidad de caracteres y en el caso de un archivo, en el tamaño máximo del mismo. Cuando un campo tiene restricciones y no se cumplen con las mismas, el campo queda en rojo, hasta que se corrige el error. En el caso de los formularios embebidos es posible agregar tooltips y mensajes de error personalizados.

Form History Diagram Description

Numero

a

Only a long value is allowed

Email

Content

Save Complete

Figura B.52 Validación de datos por tipo

B.10.2.9. Iniciación de procesos en forma manual y automática

Las instancias de proceso se pueden iniciar tanto de forma manual como automática. En cuanto al inicio manual, se realiza a través de un link (Start process, ver figura B.53) que se encuentra en la barra superior de Camunda Tasklist. Cuando se selecciona Start process, se abre un dialogo donde muestra los nombres de los procesos que el usuario tiene disponible. Un usuario tiene uno proceso disponible, siempre que tenga autorización para crear ese proceso.

En relación al inicio automático de instancias de proceso. Camunda BPM permite utilizar *Timer Start Event*. Esto posibilita, no solo definir un momento específico en el cual se debe iniciar una instancia de proceso, sino también, para procesos que necesitan crear una instancia del mismo cada cierto período de tiempo.

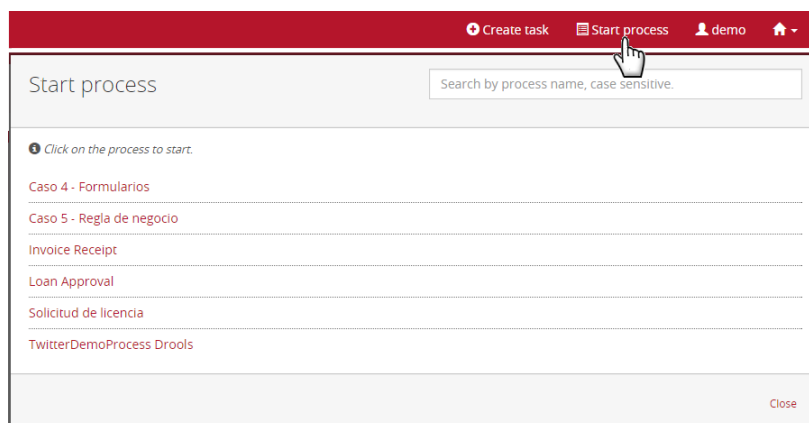


Figura B.53 Iniciar un proceso de forma manual

Para finalizar, la herramienta cuenta con un REST API [34] que permite iniciar procesos de forma externa. Por ejemplo, si se tiene un proceso cuyo nombre clave es *caso4* y el servidor de Camunda BPM se encuentra corriendo en un servidor local, para iniciar el procesos de forma externa se deberá invocar la siguiente URL:

“<http://localhost:8080/engine-rest/process-definition/key/caso4/submit-form>”. Si el inicio del proceso tiene definidas variables, o se quiere crear alguna variable al iniciar el proceso, basta con enviar un objeto JSON con una etiqueta *variables* y dentro de la misma, los nombres de las variables y sus valores. Por más información ver [61]

B.10.2.10. Definición de reglas de negocio

El cumplimiento del estándar BPMN 2.0 (ver sección B.10.1.8) permite utilizar *Business Rule Task*, ver [34].

Este tipo de tareas requieren asociar una clase Java a la misma, ésta clase debe extender

la clase `JavaDelegate` o `ActivityBehavior` [34]. Dentro de la clase Java se puede escribir la lógica de la regla de negocio. Permitiendo definir reglas de negocio basadas en variables del proceso, en la información de la ejecución del proceso o basadas en plazo.

Otro aspecto importante que presenta Camunda BPM respecto a las reglas de negocio es la posibilidad de integrar un motor de reglas de negocios, en particular, JBoss Drools. Ver demo [62]

B.10.2.11. Asignación de tareas

La asignación de tareas se puede realizar de forma manual o automática. Es posible asignar una tarea a un grupo de usuarios. Cuando un usuario del grupo reclama la tarea, la misma deja de estar disponible para el resto de los usuarios del grupo. Pero, este comportamiento no es automático, por lo tanto, si un usuario A reclama primero la tarea, luego un usuario con autorización le quita la tarea al usuario A y se la asigna a un usuario B, si el usuario A no recarga el Tasklist, es capaz de terminar la tarea aunque ya no le pertenezca. Cuando un usuario libera una tarea que en principio estaba asignada a un grupo, la misma queda disponible para dicho grupo.

La asignación manual se puede realizar tanto en el modelo como en el Tasklist. La asignación en el modelo puede ser a un usuario particular, un conjunto de usuarios (no necesariamente pertenecen al mismo grupo) o un grupo de usuarios, mientras que la asignación en el Tasklist solo puede ser a uno o más grupos de usuarios, ver figura B.54. En el caso de la asignación automática se puede realizar en el modelo a través de variables del proceso.

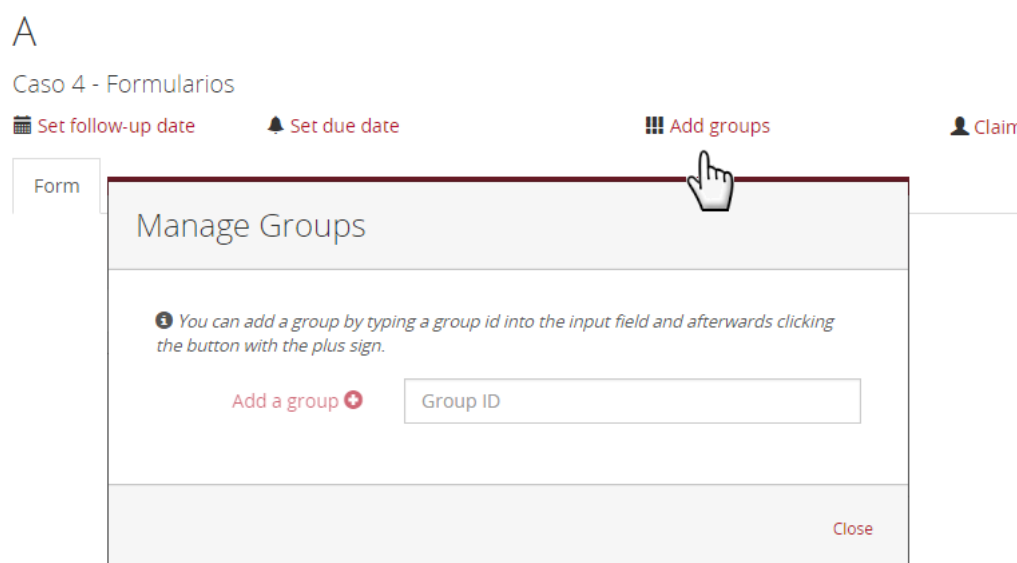


Figura B.54 Asignar tarea a un grupo de usuarios

B.10.2.12. Reasignación de tareas

Camunda provee una forma de escalar las tareas de un usuarios o todas las del sistema. A través de una fecha de vencimiento que se asigna a la tarea. Dicha fecha se puede asignar vía programación, vía REST API o a través de la tarea dentro del Tasklist.

Existen dos formas de asociar un evento al cumplimiento de la fecha límite. Asociando un timer ininterrumpible a la tarea, seguida de un *Service Task*, donde se ejecuta una clase Java que extiende a *JavaDelegate* o, registrando un *TaskListener* a la tarea, donde se crea un *jobhandle*, el cual será llamado cuando venza el plazo de la tarea.

La reasignación de tareas se puede hacer manualmente. Basta con hacer click sobre el nombre del usuario asignado dentro de la tarea y escribir el identificador del usuario al que se quiere asignar la tarea, ver figura B.55



Figura B.55 Reasignar tarea a un usuario

B.10.2.13. Adjuntar documentos a tareas

Para adjuntar un documento a una tarea es necesario que la misma tenga un formulario embebido. En el formulario se puede agregar un elemento input de tipo file, donde se puede especificar el tamaño máximo del archivo. Una vez cargado el archivo, se puede generar un link en un formulario embebido, el cual permite descargar el archivo.

B.10.2.14. Versionado de procesos

Cada proceso mantiene sus versiones. Sin embargo, solo la última versión puede estar activa. Si la última versión del proceso se suspende, al momento de iniciar un proceso, no aparecerá en la lista el proceso suspendido. Por otro lado, si hay instancias activas de una versión anterior, dichas instancias se seguirán ejecutando en la versión en las cuales fueron instanciadas. En definitiva, no es posible migrar las instancias de los procesos a la última versión.

B.10.2.15. Tareas de borrador

Camunda BPM permite guardar los datos ingresados en una tarea. Los datos que se guardan permanecen en la tarea, incluso si el usuario que ingreso los datos no completa la tarea y la tarea es reasignada a otro usuario.

B.10.2.16. Gestión de instancias por usuarios

Un usuario es capaz de gestionar las instancias de un proceso. Si tiene permiso para crear instancias de procesos. Además necesita tener permisos sobre el proceso. Por ejemplo, en el

sistema hay un proceso llamado *Alta de cliente*, si el usuario tiene permiso de creación de instancias, pero no tiene permiso sobre el proceso *Alta de cliente*, no será capaz de iniciar una instancia del mismo, es decir, necesita tener permisos relacionados con las instancias en general y sobre los procesos del sistema para poder iniciar instancias.

Para poder pausar, borrar, abortar o sobrescribir datos de las instancias, no solo tiene que tener permisos sobre instancias de procesos y sobre los procesos en sí, también necesita tener permiso de acceso al Cockpit, que es la aplicación web que permite el manejo de los procesos.

B.10.2.17. Mantenimiento del histórico del proceso

Los usuarios que tienen permiso para acceder al Cockpit, pueden ver los procesos activos (solamente de los que tiene permiso), esto le permite ver las instancias, versiones, el usuario que tiene asignada la tarea actual, los datos de la instancia, el modelo del proceso y cuantas instancias hay en cada tarea, etc. Permite visualizar gráficamente la cronología de las tareas realizadas dentro de una instancia. La historia de una instancia se puede ver en la tarea que esta activa, esto quiere decir que al momento de terminar una instancia, no es posible ver la historia de la misma desde el Cockpit.

[34] muestra como implementar las clases para guardar el histórico de una instancia de un proceso y acceder al él, incluso cuando la instancia está terminada.

No es posible acceder a los documentos que se adjuntaron a las instancias, aunque si es posible reemplazarlos.

B.10.2.18. Monitoreo de procesos

Camunda BPM provee una herramienta para el monitoreo de las instancias (Cockpit) y otra para el monitoreo de los usuarios (Admin). En la versión Community, la aplicación web Cockpit está limitada, solamente se pueden monitorear las instancias activas de un proceso. Sin embargo, se pueden definir indicadores y visualizar datos históricos de un proceso utilizando el REST API y el mantenimiento histórico del proceso, ver sección B.10.2.17.

B.10.2.19. Presentación de las tareas

Camunda brinda una aplicación web para el manejo de tareas llamada Tasklist. Es posible visualizar las tareas del usuario y de los grupos a los cuales pertenece. Para ello necesita tener permisos para visualizar el Tasklist. Si el usuario además tiene permisos de administrador, puede visualizar las tareas que cada usuario tiene asignadas.

Es posible ordenar y filtrar la lista de tareas. En la figura B.56 se muestra a la izquierda, los atributos por los cuales se puede ordenar la lista de tareas. En tanto a la derecha, muestra

las variables por las cuales es posible filtrar la lista. Si la variable es de tipo String, brinda la opción de utilizar los comparadores *like* e *=*

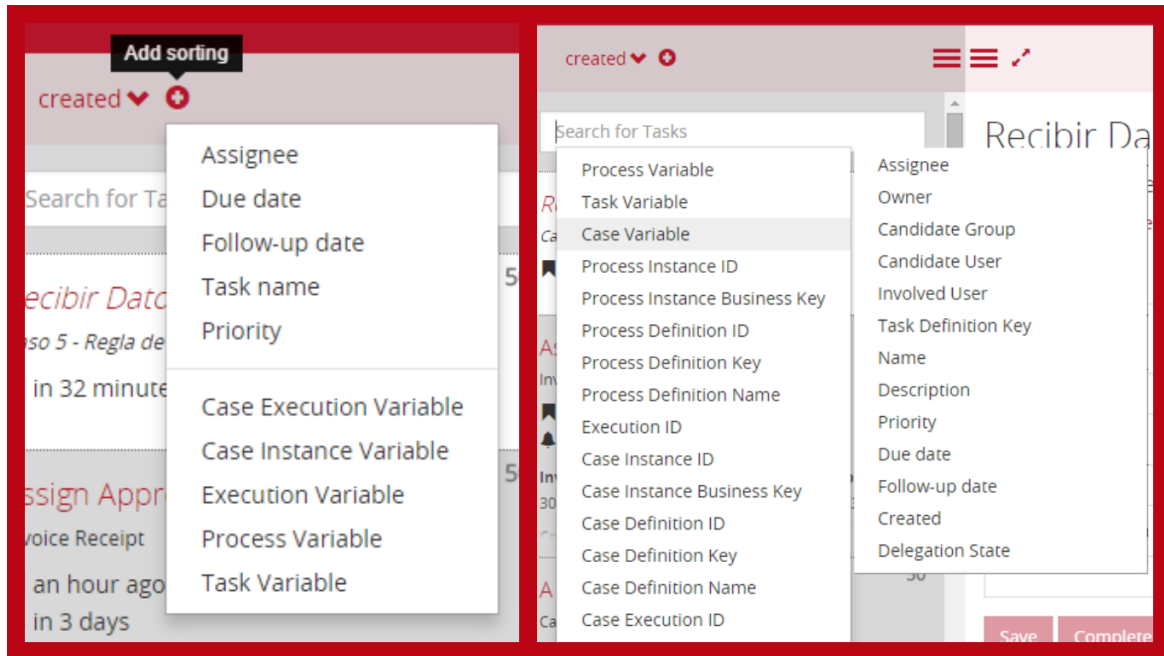


Figura B.56 Ordenar lista de tareas

A pesar de la gran cantidad de opciones para filtrar y ordenar una lista de tareas. No es posible ver las tareas completadas. Tampoco provee vistas separadas para tareas pendientes, atrasadas o redireccionadas.

Si se quiere modificar la prioridad de una tarea en Camunda BPM, solo es posible a través del REST API, por defecto todas las tareas se crean con prioridad 50.

B.10.2.20. Capacidad de aprendizaje

Camunda modeler es una herramienta que permite un aprendizaje rápido, en particular para los usuarios que alguna vez utilizaron Eclipse. Cada elemento que integra el modelo, cuando se pasa el mouse sobre ellos, se muestra una paleta de opciones, cada elemento de esa paleta tiene un tooltip que no solo indica que es, sino también su funcionalidad. Camunda modeler cuenta con un manual de usuario [34]

Las aplicaciones web (Admin, Tasklist, Cockpit) poseen tooltip en aquellos iconos que no presentan una descripción. Los mensajes de error son descriptivos, aparecen en rojo, en la parte inferior derecha de la pantalla y, permanecen allí hasta que el usuario los cierra. También muestra mensajes cuando una acción se realizó correctamente, en ese caso los mensajes son azules. Dentro de la guía del usuario, cada portal cuenta con un manual. Ver

[34].

Para finalizar, tanto las aplicaciones web como el plugin de Eclipse brindan sugerencias para solucionar un error. En el caso de Eclipse, el mensaje aparece en la consola de errores con un símbolo de advertencia amarillo. En tanto las aplicaciones web, lo hacen a través de mensajes emergentes.

Tabla B.75 Evaluación - Capacidad de aprendizaje

	Modelador	Portal
¿Posee tooltips?	SI	SI
¿Tiene mensajes de error descriptivos?	SI	SI
¿Brinda sugerencia de posible solución ante un error?	SI	SI
¿Posee manuales de uso?	SI	SI

B.10.2.21. Capacidad para ser usado

Camunda Modeler no brinda un menú de ayuda, tampoco cuenta con un espacio de preguntas frecuentes ni posee acceso directo a el manual de uso. Las aplicaciones web de Camunda tampoco cuentan con un menú de ayuda, no se encontró un espacio de preguntas frecuentes y no poseen accesos directos a los manuales de uso.

Tanto Camunda Modeler como las aplicaciones web utilizan símbolos mnemotécnicos para representar acciones tales como, crear, eliminar, aceptar, etc.

Tabla B.76 Evaluación - Capacidad para ser usado

	Modelador	Portal
¿Brinda menú de ayuda?	NO	NO
¿Tiene símbolos mnemotécnicos?	SI	SI
¿Brinda un espacio de preguntas frecuentes?	NO	NO
¿Posee acceso directo a manuales de uso?	NO	NO

B.10.2.22. Protección contra errores de usuarios

Cuando una tarea tiene un formulario asociado, dentro del formulario es posible agregar algún tipo de validación sobre los elementos del formulario, por ejemplo, se puede validar el tipo del elemento, por más información ver B.10.2.8. Para deshacer o rehacer una acción, se puede utilizar *Ctrl + Z* y *Ctrl + Y* respectivamente.

Desde el portal es posible validar los datos de un formulario, ver B.10.2.8. Los mensajes de advertencia que brindan las aplicaciones web son rojos y grandes, e indican claramente las

consecuencias si se realiza la acción. La figura muestra el mensaje de advertencia al intentar borrar un usuario. Dentro de los portales no es posible deshacer o rehacer acciones.

Tabla B.77 Evaluación - Protección contra errores de usuarios

	Modelador	Portal
¿Tiene mensajes de advertencia?	SI	SI
¿Posee validación de datos y formularios?	SI	SI
¿Cuenta con opciones para deshacer y rehacer una acción?	SI	NO

B.10.2.23. Estética de la interfaz de usuario

La información relevante dentro de los portales y de Camunda Modeler es accesible y fácil de encontrar. Por ejemplo, la historia de una instancia activa se encuentra en la tarea que está pendiente, dentro de la pestaña *History*. La información que muestran tanto los portales como el modelador, no es innecesaria.

Tabla B.78 Evaluación - Estética de la interfaz de usuario

	Modelador	Portal
¿Incluye información innecesario?	NO	NO
La información relevante ¿Se encuentra accesible y fácil de encontrar?	SI	SI

B.10.2.24. Accesibilidad

Las acciones tanto en el modelador como en las aplicaciones web se representan con imágenes adecuadas. Los colores son distinguibles tanto en el modelador como dentro de las aplicaciones web. Dentro de las aplicaciones web, predominan los colores rojos y blanco, los mensajes de error o advertencia son rojos, mientras que los mensajes de éxito son azules en el Cocktip y verdes en Admin.

Ni el modelador, ni las aplicaciones web cuentan con comando de voz o utilizan sonido alguno.

Tabla B.79 Evaluación - Accesibilidad

	Modelador	Portal
¿Incluye comando de voz?	NO	NO
¿Las acciones se encuentran representadas con imágenes adecuadas?	SI	SI
¿Los colores son distinguibles aunque el usuario sea daltónico?	SI	SI
¿Los sonidos se corresponden con la acción que se está ejecutando?	NO	NO

B.10.2.25. Confidencialidad

Las contraseñas se guardan cifradas en la base de datos. Camunda no cuenta aún con una aplicación mobile. Los intercambios de datos que realiza con un servidor no son sobre HTTPS, pero se pueden configurar los llamados externos dentro de las clases java para que los intercambios de datos se realicen sobre HTTPS.

Tabla B.80 Evaluación - Confidencialidad

	Portal
¿Utiliza técnicas criptográficas?	SI
Mobile ¿Almacena los datos cifrados en el dispositivo?	NO
¿Los intercambios de datos con el servidor son sobre HTTPS?	NO

B.10.2.26. Integridad

Un usuario administrador es capaz de crear nuevos grupos y asignar permisos a dichos grupos, esto permite visualizar diferentes funciones según el grupo al que pertenece un usuario. Además, es posible asignar diferentes permisos a un usuario particular. Camunda cuenta con una aplicación web para administrar a los usuarios, en dicha aplicación un usuario administrador puede dar diferentes permisos y autorizaciones tanto a usuarios como a grupos.

El acceso a los documentos no depende de los permisos que tenga el usuario. Por ejemplo, si se conoce el nombre de la variable donde se cargo el documento, el identificador de la tarea o el identificador de la instancia del proceso, es posible obtener el documento a través del REST API de Camunda. Los documentos no se guardan cifrados, se guarda un array de bytes en la base de datos. Camunda cuenta con un tiempo limite de sesión activa, pasado ese tiempo, si se intenta realizar una acción, el usuario es redirigido hacia el login.

Tabla B.81 Evaluación - Integridad

	Portal
¿Permite definir roles?	SI
¿Cuenta con acceso diferenciado a los documentos según el rol?	NO
¿Permite visualizar diferentes funciones según el rol?	SI
¿Permite administrar permisos?	SI
¿Los documentos se guardan cifrados?	NO
¿Cuenta con un tiempo limite de sesión activa?	SI

B.10.2.27. No repudio

Camunda cuenta con un historial de ejecución, ver sección B.10.2.17.

Tabla B.82 Evaluación - No repudio

	Portal
¿Cuenta con un historial de ejecución de procesos?	SI
¿Se puede identificar que usuario y en que momento se ejecutó un proceso?	SI

B.10.2.28. Responsabilidad

Las tareas y los procesos cuentan con un identificador numérico que es único. Con este identificador es posible obtener datos tanto de las tareas como de los procesos y sus instancias a través del REST API de Camunda.

Tabla B.83 Evaluación - Responsabilidad

	Portal
¿Las tareas y los procesos tienen identificador único?	SI

B.10.2.29. Autenticidad

Camunda provee su propia forma de autenticación, pero también permite integrar el sistema con bases externas como LDAP. Por más información ver [34].

Tabla B.84 Evaluación - Autenticidad

	Portal
¿Cuenta con alguna forma de autenticación de usuarios?	SI
¿Soporta firmas digitales?	NO

B.10.2.30. Tiempo de respuesta

El gráfico de la figura B.57 muestra el tiempo de respuesta en función de la cantidad de usuarios. En la gráfica se observan dos funciones, la de color azul representa los valores obtenidos luego de la ejecución, mientras que la función naranja indica la tendencia. La tendencia corresponde a una función lineal, esto indica que el tiempo de respuesta crece en igual proporción a la cantidad de usuarios que estén utilizando el sistema al mismo tiempo.

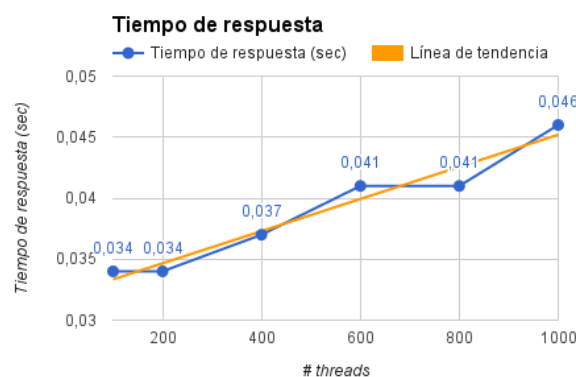


Figura B.57 Tiempo de respuesta

Tomando un promedio de los valores que se obtuvieron, el tiempo de respuesta para la herramienta es: 0,04 segundos

B.10.2.31. Throughput

La gráfica B.58 muestra el porcentaje de procesos iniciados en 30 minutos, de acuerdo a la cantidad de usuarios. Para obtener estos valores se utilizó el mismo conjunto de test que fueron utilizados para obtener el tiempo de respuesta. Cada test se ejecuto durante 30 minutos y se obtuvo la cantidad de procesos iniciados en el intervalo de tiempo antes mencionado, ya que el throughput refiere al número de eventos de respuesta en un intervalo. Finalmente, se utilizó la ecuación B.1 para calcular los valores de throughput que se muestran en la gráfica.

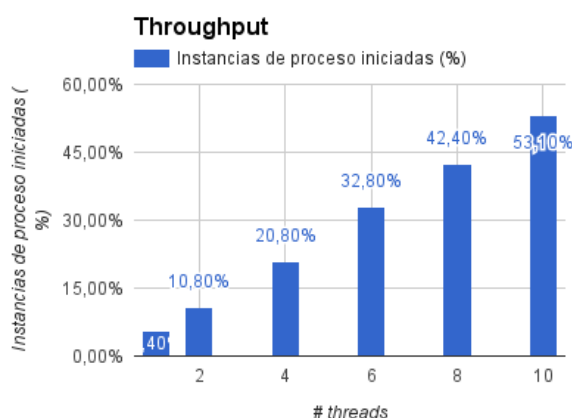


Figura B.58 Throughput (%) frente a Usuarios

Tomando la mediana de los valores que se obtuvieron, el throughput para la herramienta es: 26,80%

B.10.2.32. Capacidad

La gráfica de la figura B.59 muestra el porcentaje de procesos finalizados respecto a la cantidad de usuarios. Los datos se obtuvieron a partir de la ejecución de los test utilizados para calcular el throughput, ya que estos conceptos están relacionados. Para obtener el throughput se observó la cantidad de procesos iniciados, mientras que para la capacidad, se observaron los procesos finalizados.

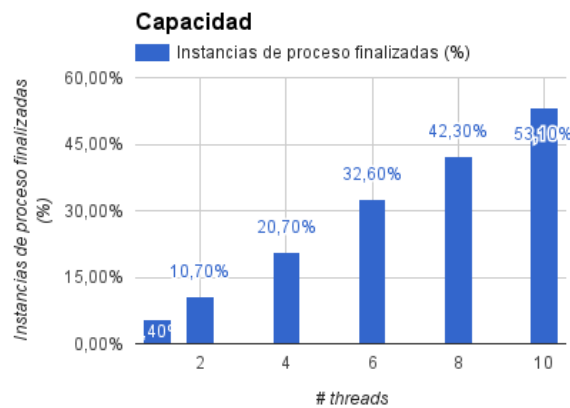


Figura B.59 Capacidad (%) frente a Usuarios

Tomando la mediana de los valores que se obtuvieron, la capacidad para la herramienta es: 26,65 %

B.10.3. Tabla de nivel de cumplimiento

La tabla B.85 muestra los resultados obtenidos para las características funcionales. Las características están divididas por categoría. De cada característica se muestra si la evaluación es teórica (Teó), el nivel de soporte (NS) y el nivel de cumplimiento (NC).

Tabla B.85 Características técnicas - Nivel de cumplimiento.

Modulo	Característica	Teó	NS	NC
	Tecnología, Arquitectura e Interoperabilidad			
	Arquitectura de BPMS	Si	⊗	⊗
	Compatibilidad con estándares	Si	●	✓
	Base de datos	Si	●	✓
	Sistema operativo del modelador	Si	●	✓
	Navegador Web	Si	●	✓
	Extensibilidad	Si	●	⚙
	Integración con Servicios Web	No	●	✓

Modulo	Característica	Teó	NS	NC
	Diseño y modelado de procesos			
	Interfaz gráfica para key users	No		
	Tipo de modelador	Si		
	Notación gráfica de procesos compatible con BPMN 2.0	Si		
	Gestión de formularios			
	Definición de formularios	No		
	Formato de formularios	No		
	Elementos de formularios	No		
	Formularios dinámicos	No		
	Reglas de llenado de campos	No		
	Validación de datos del formulario	No		
	Motor de Workflow			
	Iniciación de procesos en forma manual y automática	No		
	Definición de reglas de negocio	No		
	Asignación de tareas	No		
	Reasignación de tareas	No		
	Adjuntar documentos a tareas	No		
	Versionado de procesos	No		
	Tareas de borrador	No		
	Gestión, monitoreo y auditoría			
	Gestión de instancias por usuarios	No		
	Mantenimiento histórico de procesos	No		
	Monitoreo de procesos	No		
	Portal			
	Presentación de las tareas	No		

Referencias:

-  No aplica
-  Cumple de forma nativa
-  Cumple vía integración
-  Cumple vía particularización
-  Soporta totalmente
-  Soporta parcialmente
-  No soportada

B.10.3.1. Resultado numérico de las características técnicas

Las tablas B.88 y B.90 presentan los resultados numéricos correspondientes a las características funcionales y no funcionales evaluadas. En la sección 3.7 se describe el método de evaluación, donde se presentan los cálculos a realizar para poder clasificar a las características según los umbrales definidos.

Primero, hay que definir los valores de importancia, para poder definir los umbrales. Además, es necesario dar valores a los niveles de cumplimiento y de soporte.

Tabla B.86 Definición de constantes

Importancia		Nivel de cumplimiento		Nivel de soporte	
Alto	3	Nativo	2	Total	3
Medio	2	Vía integración	1	Parcial	2
Bajo	1	Vía particularización	1	No soportado	1

Tabla B.87 Definición de constantes

Las columnas de la tabla B.88 contienen: las características, importancia (I), nivel de soporte (NS), nivel de cumplimiento (NC), el factor (F) y el puntaje total (Total). El puntaje total se calcula de la siguiente manera:

Puntaje total

$$Puntaje = I * NS * NC * Factor \quad (B.13)$$

Donde I es la importancia, NS el nivel de soporte, NC el nivel de cumplimiento y Factor un valor entre 0,1 y 0,9.

Tabla B.88 Características técnicas - Cálculo del puntaje.

Característica	I	NS	NC	F	Total
Tecnología, Arquitectura e Interoperabilidad					
Compatibilidad con estándares	1	1	2	1	2
Base de datos	1	2	2	1	4
Sistema operativo del modelador	1	2	2	1	4
Navegador Web	1	2	2	1	4
Extensibilidad	1	2	1	1	2
Integración con Servicios Web	1	2	2	1	4
Diseño y modelado de procesos					
Interfaz gráfica para key users	1	1	2	1	2
Tipo de modelador	1	2	2	1	4
Notación gráfica de procesos compatible con BPMN 2.0	1	2	2	1	4
Gestión de formularios					
Definición de formularios	1	1	1	0,9	0,90
Formato de formularios	1	2	2	1	4
Elementos de formularios	1	2	1	1	2
Formularios dinámicos	1	2	1	0,9	1,8
Reglas de llenado de campos	1	2	1	0,7	1,4
Validación de datos del formulario	1	2	2	1	4
Motor de Workflow					
Iniciación de procesos en forma manual y automática	1	2	2	1	4
Definición de reglas de negocio	1	2	2	1	4
Asignación de tareas	1	2	2	1	4
Reasignación de tareas	1	2	2	1	4
Adjuntar documentos a tareas	1	2	1	0,5	1
Versionado de procesos	1	1	2	1	2
Tareas de borrador	1	2	2	1	4
Gestión, monitoreo y auditoría					
Gestión de instancias por usuarios	1	2	2	1	4
Mantenimiento histórico de procesos	1	1	2	1	2
Monitoreo de procesos	1	1	1	0,2	0,2
Portal					
Presentación de las tareas	1	1	2	1	2

Por otro lado tenemos los cálculos de las características no funcionales. Las columnas de la tabla B.90 contienen: el peso de cada subcaracterística (P_i), el nivel de cumplimiento de la subcaracterística (s_i), el valor final de la subcaracterística definido en la ecuación B.18 y el nivel de cumplimiento (NC). El nivel de cumplimiento se obtiene de la siguiente manera:

- $vf < 1 \rightarrow$ Nivel de cumplimiento Bajo.
- $1 \leq vf < 3 \rightarrow$ Nivel de cumplimiento Medio.
- $3 \leq vf \rightarrow$ Nivel de cumplimiento Alto.

Valor final

$$vf = \frac{\sum_{i=1}^N (s_i * P_i)}{\sum_{i=1}^N P_i} \quad (B.14)$$

Siendo vf el 'valor final' de la característica y N la cantidad de subcaracterísticas.

En el caso de las características incluidas en la categoría Performance, hay que definir cuales son los valores de $L1$ y $L2$ para cada una de ellas, de esta manera, se podrá determinar el valor final de cada característica. Para determinar estos valores se consideraron estudios previos de performance así como de usabilidad [47] [42]. A continuación se muestra una tabla con los valores determinados:

Tabla B.89 Parámetros

Característica	L1	L2
Tiempo de respuesta	3 seg	10 seg
Throughput	30 %	80 %
Capacidad	20 %	70 %

Se considera que todas las subcaracterísticas son importantes y en este sentido se asigna el valor 1 al peso de cada subcaracterística.

B.11. Evaluación de Aris Express

Para la evaluación de Aris Express se utilizó la versión 2.4, la cual se obtuvo de la página de la herramienta [23].

B.11.1. Evaluación de las características teóricas

B.11.1.1. Arquitectura del BPMS

Aris Express es una herramienta de modelado que se basa en los métodos estándares de Aris. Provee una interfaz intuitiva y asistencia para modelado de procesos.

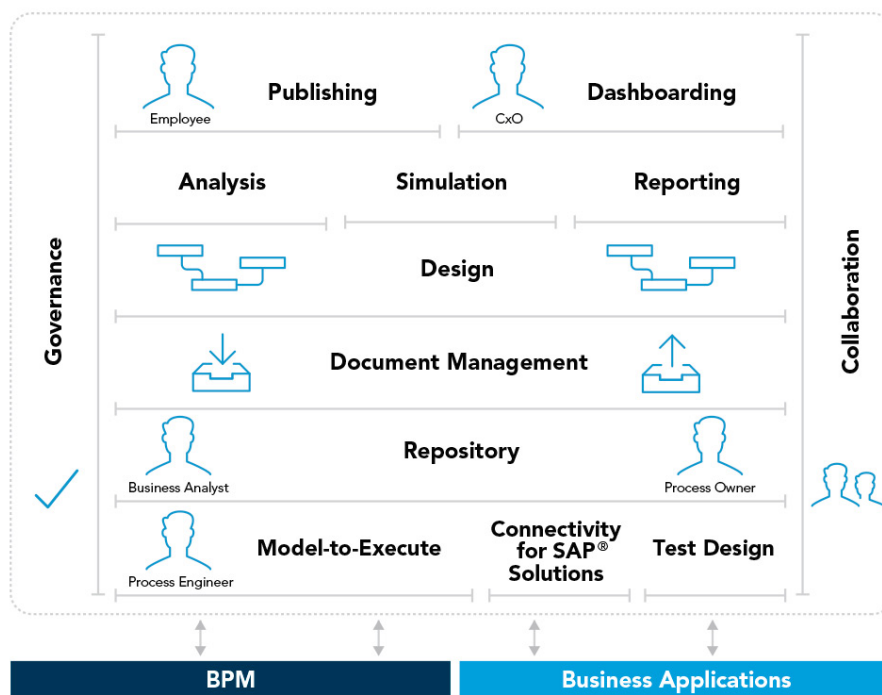


Figura B.60 Arquitectura - Aris [35]

B.11.1.2. Compatibilidad con estándares

Aris le permite importar modelos en formato BPMN y XML y exportarlos BPMN o XML.

B.11.1.3. Sistema operativo del modelador

Aris solo cuenta con un modelador de escritorio, el mismo es soportado en los siguientes sistemas operativos:

- Windows XP
- Windows Vista
- Windows 7

Debería correr en cualquier sistema operativo que cuente con una correcta instalación de Java Runtime Environment (JRE). Por ejemplo, debería correr en Linux pero el usuario podría experimentar algunos problemas en el *look&feel*.

B.11.1.4. Tipo de modelador

Aris solo cuenta con un modelador de escritorio.

B.11.1.5. Notación gráfica de procesos compatible con BPMN 2.0

[23] afirma que soporta el estándar BPMN 2.0 para el modelado.

B.11.1.6. Capacidad de aprendizaje

El modelador de Aris cuenta con una página de inicio en la cual se brinda acceso a una amplia cantidad de acciones y en la que se pueden ver ejemplos de algunos de los modelos así como también se muestra un historial de los últimos modelos guardados.

El modelador posee tooltips que describen la acción del icono.

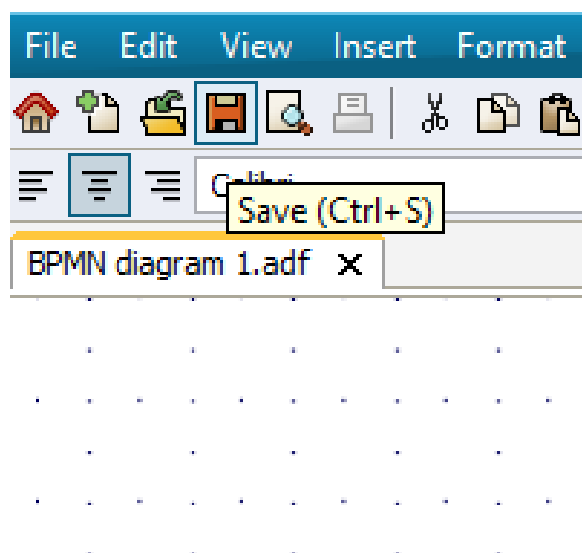


Figura B.61 Tooltips - Aris

Los mensajes de error que muestran en el modelador son descriptivos y poseen un botón de ayuda en el cual se brindan sugerencia de posibles soluciones.

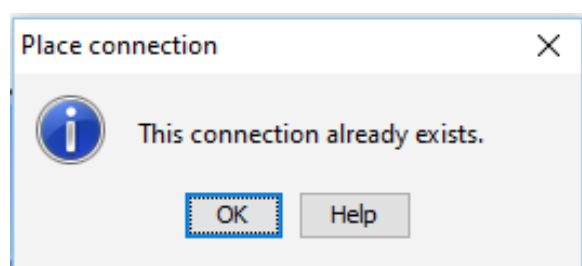


Figura B.62 Ejemplo de mensaje de error - Aris

En la barra de tareas existe un menú de ayuda denominado *Help* mediante el cual se puede acceder a los manuales de uso, tutoriales, *help topics* entre otros.

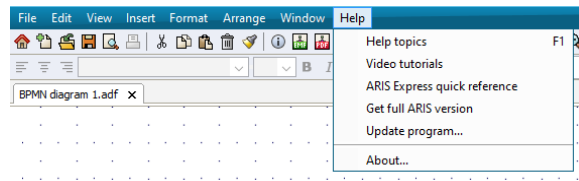


Figura B.63 Menú de Ayuda - Aris

Tabla B.91 Evaluación - Capacidad de aprendizaje

	Modelador	Portal
¿Posee tooltips?	SI	NA
¿Tiene mensajes de error descriptivos?	SI	NA
¿Brinda sugerencia de posible solución ante un error?	SI	NA
¿Posee manuales de uso?	SI	NA

B.11.1.7. Capacidad para ser usado

Al iniciar la aplicación se brinda un menú de ayuda donde se permite seleccionar varias opciones de modelos y además se proveen ejemplos para algunos de ellos. Al ingresar a la modelado existe una barra de herramientas donde una de las opciones es de Ayuda. En ésta se acceso a los manuales, acceso al espacio de preguntas frecuentes, información acerca de la herramienta, entre otras.

Los símbolos en el modelador son mnemotécnicos y se corresponden con las acciones que representan.

Tabla B.92 Evaluación - Capacidad para ser usado

	Modelador	Portal
¿Brinda menú de ayuda?	SI	NA
¿Tiene símbolos mnemotécnicos?	SI	NA
¿Brinda un espacio de preguntas frecuentes?	SI	NA
¿Posee acceso directo a manuales de uso?	SI	NA

B.11.1.8. Protección contra errores de usuarios

La herramienta muestra mensajes de advertencia cuando se realizan acciones como el borrado de objetos o cuando se cierra alguna pestaña sin guardar. Sin embargo no cuenta con

la funcionalidad de validación del modelo. En la barra de tareas se encuentran los botones de rehacer y deshacer que permiten volver a mostrar objetos borrados o volver a un estado anterior en el modelado.

Tabla B.93 Evaluación - Protección contra errores de usuarios

	Modelador	Portal
¿Tiene mensajes de advertencia?	SI	NA
¿Posee validación de datos y formularios?	NO	NA
¿Cuenta con opciones para deshacer y rehacer una acción?	SI	NA

B.11.1.9. Estética de la interfaz de usuario

La herramienta cuenta con diferentes vistas dependiendo del tipo de modelo que se quiera realizar, por lo que la información mostrada y las acciones que se pueden realizar no son todas adecuadas al modelado que se está realizando, por lo que el modelador no cuenta con información innecesaria. Por otro lado la información y las acciones se encuentra accesible y es fácil de encontrar.



Figura B.64 Barra de herramientas - Aris

Tabla B.94 Evaluación - Estética de la interfaz de usuario

	Modelador	Portal
¿Incluye información innecesario?	NO	NA
La información relevante ¿Se encuentra accesible y fácil de encontrar?	SI	NA

B.11.1.10. Accesibilidad

Las acciones en la herramienta no están acompañadas de comandos de voz. Las imágenes de cada una de las acciones son acordes a las mismas y los colores son adecuados. En la paleta de modelado se puede distinguir fácilmente por un usuario daltónico cada uno de los elementos ya que tienen colores que los identifican.

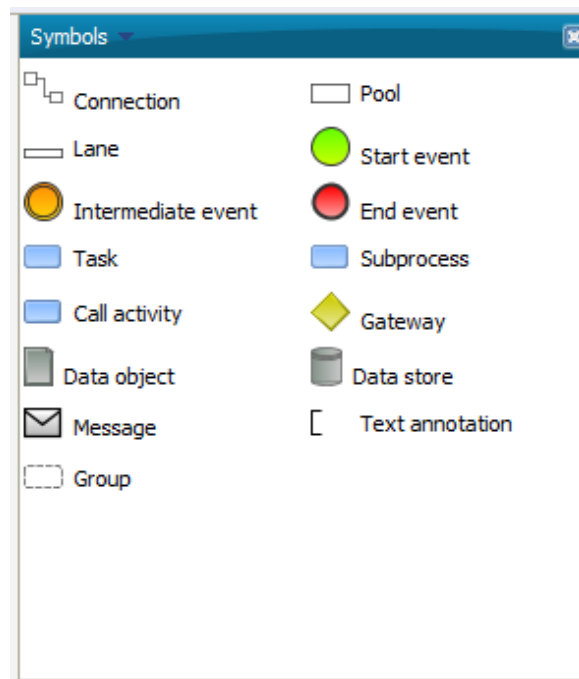


Figura B.65 Símbolos para el modelado - Aris

Tabla B.95 Evaluación - Accesibilidad

	Modelador	Portal
¿Incluye comando de voz?	NO	NA
¿Las acciones se encuentran representadas con imágenes adecuadas?	SI	NA
¿Los colores son distinguibles aunque el usuario sea daltónico?	SI	NA
¿Los sonidos se corresponden con la acción que se está ejecutando?	NO	NA

B.12. Evaluación de Orchestra

Es una herramienta open source que utiliza el estándar BPEL. El software se obtuvo del sitio de la herramienta.

B.12.1. Evaluación de las características teóricas

B.12.1.1. Arquitectura del BPMS

La "Process Virtual Machine" define un motor genérico de procesos que permite el soporte para múltiples procesos en diferentes idiomas como por ejemplo BPEL y XPD. L.

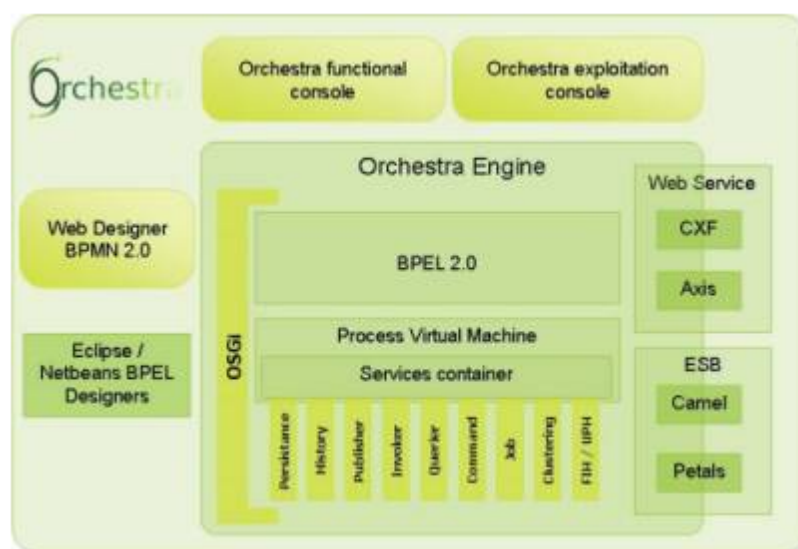


Figura B.66 Arquitectura - Orchestra [24]

La nueva arquitectura tiene como objetivo crear un acoplamiento flexible entre el motor y la capa de web services y las capas de persistencia, a través del uso de las API en ambos lados (para hacer fácilmente los cambios en las capas de web services o de persistencia)

Además, como las aplicaciones cliente envían sus datos ya formateados en XML, se tiene la necesidad de transformar a XML los datos todo el tiempo, ya que es totalmente portátil y fácil de manipular con el software de código abierto.

Para la capa de web services, necesitamos dos interfaces:

- La API de servicios Web
- El API Engine

La interfaz de los web services tendrá métodos necesarios por el motor, para enviar los servicios de mensajes o llamadas. Y la API del motor tendrá métodos necesarios para el web service para dar mensajes recibidos al proceso, o para indicar que un servicio ha contestado.

Funcionamiento del motor Un proceso BPEL se ejecuta en 2 pasos:

- **deploy:** en la que los archivos WSDL y BPEL se analizan para crear objetos DOM. Estos DOM se analizan de nuevo para crear la ejecución de Java (con actividades, etc) y para realizar comprobaciones de validez estáticas como se requiere en la especificación BPEL.
- **ejecución:** el motor pasa a través de todo el árbol de Java construida durante el despliegue. actuaciones de ejecución son un tema crítico durante la ejecución.

Requerimientos de Hardware Según la documentación se recomienda tener 1GHz de procesador con un mínimo de 512 MB de RAM. También se indica que aumentando estos valores se verá reflejado en la performance del sistema.

B.12.1.2. Compatibilidad con estándares

Es posible importar/exportar procesos en formato XPD, BPMN y BPEL.

B.12.1.3. Base de datos

Los motores de base de datos soportados por Orchestra son:

- Oracle
- MySQL
- PostgreSQL

B.12.1.4. Sistema operativo del modelador

El modelador es una aplicación web que es ejecutado en un tomcat. Funciona correctamente tanto en ambiente Linux como Windows.

B.12.1.5. Navegador web

Tanto el modelador como el portal fueron probados en los navegadores listados a continuación y su funcionamiento es correcto.

- Internet explorer 10
- Google Chrome
- Mozilla Firefox

B.12.1.6. Extensibilidad

Orchestra permite agregar código propio a través de rutas Camel e integrar APIs.

B.12.1.7. Tipo de modelador

El modelador de procesos de Orchestra es una aplicación web que ejecuta en un servidor de aplicaciones Tomcat.

B.12.1.8. Notación gráfica de procesos compatible con BPMN 2.0

El motor es compatible con BPMN 2.0 pero no es posible diseñar desde aplicación de diseño web todas las compuertas para cumplir con el estándar. Según la documentación es un trabajo que está en proceso.

B.12.2. Evaluación de las características prácticas

B.12.2.1. Integración con servicios web

La integración con web services se hace a través de Camel. Para esto hay que definir una ruta en Camel que se consuma un web service y desde Orchestra se inicia esta ruta.

Esta configuración se agrega en el archivo BAR.

B.12.2.2. Interfaz gráfica para key users

Los procesos básicos pueden ser diseñados por usuarios sin conocimientos técnicos. Al agregar tarea mas complejas como interacción con servicios externos, se requiere de un usuario con una capacitación en estos temas ya se necesita de código de programación.

B.12.2.3. Listado de versiones de modelos de procesos

Orchestra soporta el versionado de procesos. En el listado de procesos se puede ver el mismo proceso con distintas versiones y los estados de las mismas.

B.12.2.4. Definición de formularios

Según la documentación que presenta Orchestra no permite la definición de formularios en tareas. De acuerdo a la investigación realizada se pueden ingresar variables de proceso solamente en esta versión

B.12.2.5. Formato de formularios

Como se dijo en el punto anterior no es posible definir formularios.

B.12.2.6. Elementos de formularios

Como se dijo en los puntos anteriores no es posible definir formularios.

B.12.2.7. Formularios dinámicos

Como se dijo en los puntos anteriores no es posible definir formularios.

B.12.2.8. Reglas de llenado de campos

La herramienta con cuenta con regla de llenado de campos.

B.12.2.9. Validación de campos de formularios

La herramienta con cuenta validación de campos en formularios.

B.12.2.10. Iniciación de procesos en forma manual y automática

Orchestra provee una API de Servicios para hacer deploy/undeploy de procesos. Además tiene otra API para gestión de instancias con el que se puede suspender, reanudar y terminar la instancia. Sin embargo no se puede iniciar el proceso desde la API.

B.12.2.11. Definición de reglas de negocio

Orchestra no tiene soporte para reglas de negocio.

B.12.2.12. Asignación de tareas

En orchestra las tareas manuales no se asignan, quedan disponibles hasta que un usuario con permisos correspondientes sobre el proceso, la tome y la realice.

B.12.2.13. Reasignación de tareas

Orchestra no permite la reasignación directa de Tareas. Para que una tarea cambie de usuario, el usuario que tiene la tarea debe de dejar la reserva de la misma y cualquier otro usuario con permisos sobre el proceso, puede tomarla.

B.12.2.14. Adjuntar documentos a tareas

Orchestra no tiene soporte adjuntar documentos a las tareas.

B.12.2.15. Tareas de borrador

Orchestra no permite guardar tareas con su contenido.

B.12.2.16. Gestión de instancias por usuarios

La herramienta permite a los usuarios crear, pausar, reanudar y abortar. Sin embargo, no es posible sobrescribir datos de la instancia.



Figura B.67 Gestión de Instancias de Usuario - Orchestra

B.12.2.17. Mantenimiento del histórico del proceso

Los usuarios pueden ver los procesos activos, esto le permite ver las instancias y versiones, el modelo del proceso y cuantas instancias hay de cada proceso. Permite visualizar gráficamente las tareas realizadas dentro de una instancia.

B.12.2.18. Monitoreo de procesos

Para el monitoreo de procesos Orchestra cuenta con un dashboard en el cual se muestran gráficas con los procesos activos y pausados discriminados por versión.

Ademas tiene una pestaña de procesos que se lista los procesos, estado, fecha de deploy, etc.

A continuación se muestran imágenes con las 2 pestañas.

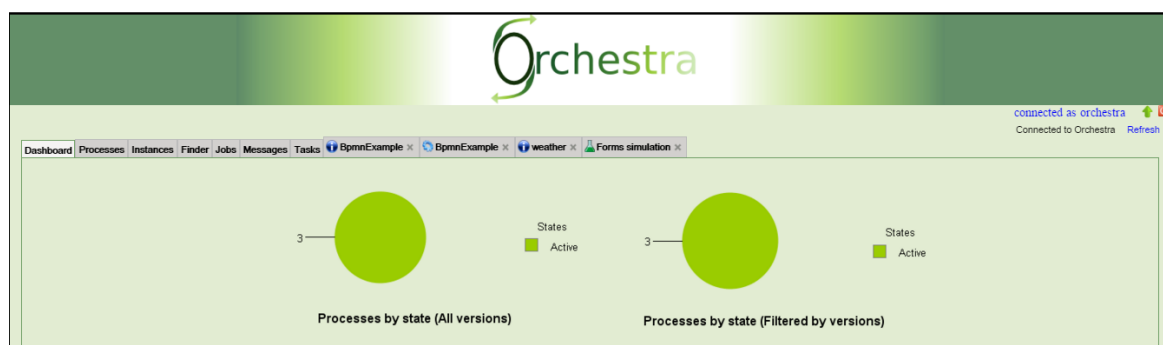


Figura B.68 Dashboard Orchestra



Figura B.69 Procesos Orchestra

B.12.2.19. Presentación de las tareas

Respecto presentación de las tareas Orchestra presenta un listado en el cual se puede ver Fecha de creación, identificador, tipo y estado. No cuenta con la funcionalidad de ordenar en forma ascendente o descendente las diferente columnas que se muestra ni permite editar las columnas a mostrar ni el orden de las mismas.

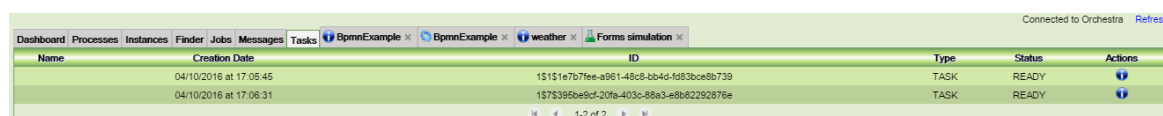


Figura B.70 Presentación de Tareas en Orchestra

B.12.2.20. Capacidad de aprendizaje

El aprendizaje de la herramienta tiene bastantes dificultades ya que no cuenta con tutoriales de vídeo y las documentación que hay es muy escasa, prácticamente sin imágenes. Además hay muy poca participación de los usuarios en los foros.

Los símbolos que se utilizan tanto en el designer como en el portal no son todos mnemotécnicos.

Como positivo hay que destacar que el modelador tiene menú de ayuda y en el sitio tiene un espacio de preguntas frecuentes para el usuario.

Tabla B.96 Evaluación Orchestra - Capacidad de aprendizaje

	Modelador	Portal
¿Brinda menú de ayuda?	SI	NO
¿Tiene símbolos mnemotécnicos?	NO	NO
¿Brinda un espacio de preguntas frecuentes?	NO	SI
¿Posee acceso directo a manuales de uso?	SI	NO

B.12.2.21. Protección contra errores de usuarios

No posee validaciones de datos. Para deshacer o rehacer una acción tanto en el designer se puede utilizar botones específicos para deshacer y rehacer una acción. No tiene los clásicos shortcuts de $Ctl + Z$ y $Ctl + Y$.

Tabla B.97 Evaluación Orchestra - Protección contra errores de usuarios

	Modelador	Portal
¿Tiene mensajes de advertencia?	NO	NO
¿Posee validación de datos y formularios?	NO	NO
¿Cuenta con opciones para deshacer y rehacer una acción?	SI	NO

B.12.2.22. Estética de la interfaz de usuario

La información relevante dentro del portal y designer es accesible y fácil de encontrar. No se muestra información innecesaria.

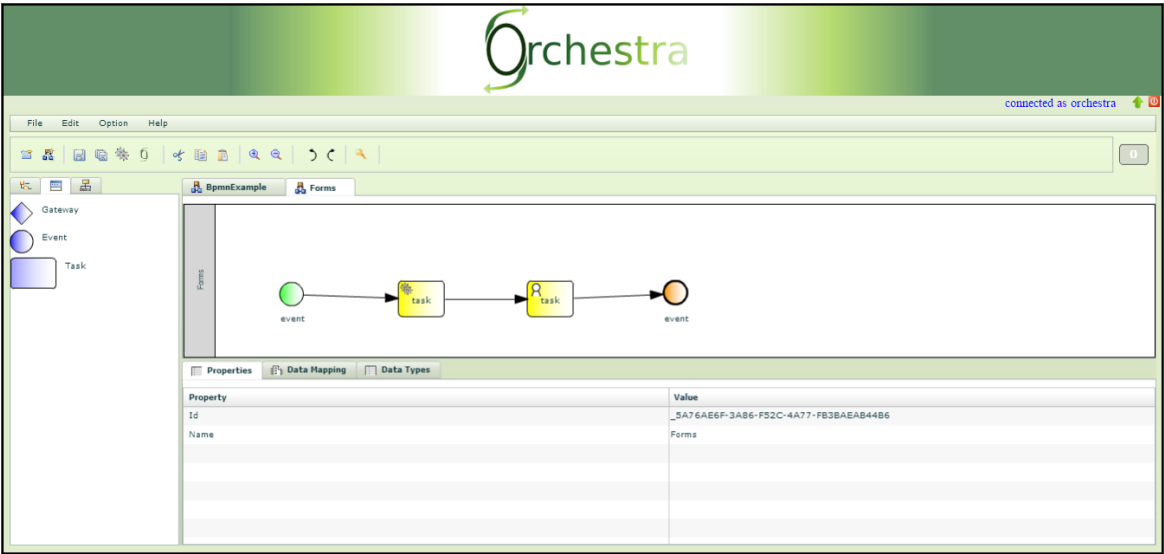


Figura B.71 Designer de Orchestra



Figura B.72 Portal de Orchestra

Tabla B.98 Evaluación Orchestra - Estética de la interfaz de usuario

	Modelador	Portal
¿Incluye información innecesario?	NO	NO
La información relevante ¿Se encuentra accesible y fácil de encontrar?	SI	SI

B.12.2.23. Accesibilidad

Las acciones tanto en el designer como en el portal se representan con imágenes adecuadas. También los colores son distinguibles en ambas aplicaciones.

Ni el modelador, ni el portal cuentan con comando de voz o utilizan sonido alguno.

Tabla B.99 Evaluación - Accesibilidad

	Modelador	Portal
¿Incluye comando de voz?	NO	NO
¿Las acciones se encuentran representadas con imágenes adecuadas?	SI	SI
¿Los colores son distinguibles aunque el usuario sea daltónico?	SI	SI
¿Los sonidos se corresponden con la acción que se está ejecutando?	NO	NO

B.12.2.24. Confidencialidad

Las contraseñas se guardan cifradas en la base de datos. Orchestra no cuenta con una aplicación mobile . Los intercambios de datos que realiza con el servidor no son sobre HTTPS.

Tabla B.100 Evaluación Orchestra - Confidencialidad

	Portal
¿Utiliza técnicas criptográficas?	SI
Mobile ¿Almacena los datos cifrados en el dispositivo?	NO
¿Los intercambios de datos con el servidor son sobre HTTPS?	NO

B.12.2.25. Integridad

Orchestra no implementa los roles o grupos de usuarios, solamente cuenta con la funcionalidad de crear nuevos usuarios mediante configuración en el servidor.

No tiene manejo de documentos en los procesos.

El portal ni el designer cuentan con un tiempo limite de sesión activa.

Tabla B.101 Evaluación - Integridad

	Portal
¿Permite definir roles?	NO
¿Cuenta con acceso diferenciado a los documentos según el rol?	NO
¿Permite visualizar diferentes funciones según el rol?	NO
¿Permite administrar permisos?	SI
¿Los documentos se guardan cifrados?	NO
¿Cuenta con un tiempo limite de sesión activa?	NO

B.12.2.26. No repudio

Orchestra permite visualizar un historial de ejecución. Permite visualizar de forma simple, gráfica y cronológica las actividades de un proceso. Sin embargo no permite ver que usuario ejecutó la tarea.

Tabla B.102 Evaluación Orchestra - No repudio

	Portal
¿Cuenta con un historial de ejecución de procesos?	SI
¿Se puede identificar que usuario y en que momento se ejecutó un proceso?	NO

B.12.2.27. Responsabilidad

Las tareas y los procesos cuentan con un identificador numérico que es único. Se puede ver en la imagen de Presentación de Tareas de Orchestra.

Tabla B.103 Evaluación Orchestra - Responsabilidad

	Portal
¿Las tareas y los procesos tienen identificador único?	SI

B.12.2.28. Autenticidad

La autenticación que provee Orchestra valida con los usuarios declarados en el servidor de aplicaciones.

Tabla B.104 Evaluación Orchestra

	Portal
¿Cuenta con alguna forma de autenticación de usuarios?	SI
¿Soporta firmas digitales?	NO

B.12.2.29. Tiempo de respuesta

El gráfico B.73 muestra el tiempo de respuesta respecto a la cantidad de *threads*. Para obtener los valores se realizaron varias ejecuciones para distintas cantidad de *threads*. En la figura B.73, la función azul representa los valores obtenidos luego de las ejecuciones, en tanto la función naranja indica la tendencia. La tendencia corresponde a una función lineal, esto quiere decir que el tiempo de respuesta tiene un crecimiento que es directamente proporcional a la cantidad de hilos que se ejecuten.

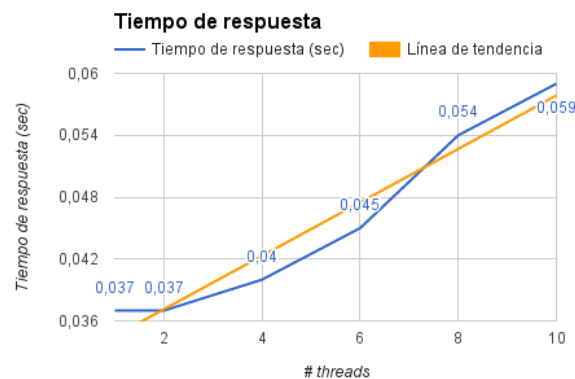


Figura B.73 Tiempo de respuesta

Tomando un promedio de los valores que se obtuvieron, el tiempo de respuesta para la herramienta es: 0,048 segundos

B.12.2.30. Throughput

Para obtener el throughput se utilizaron los test requeridos para determinar el tiempo de respuesta. Se ejecutaron durante 30 minutos cada uno de ellos, obteniendo la cantidad de procesos iniciados en dicho intervalo de tiempo. Una vez que se consiguieron los datos, se utilizó la ecuación B.1 para calcular el throughput. La gráfica B.74 muestra el porcentaje de procesos iniciados en 30 minutos, de acuerdo a la cantidad de usuarios.

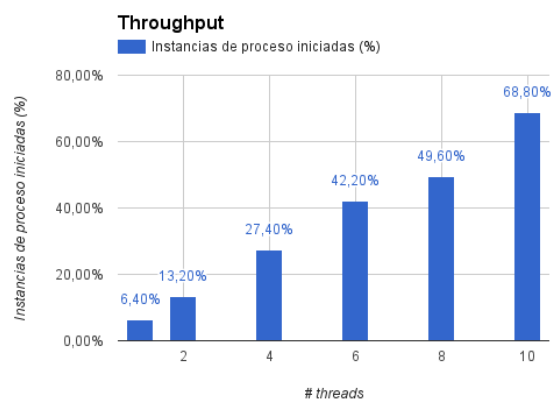


Figura B.74 Throughput (%) frente a Usuarios

Tomando la mediana de los valores que se obtuvieron, el throughput para la herramienta es: 34,80 %

B.12.2.31. Capacidad

La capacidad de un sistema es la cantidad de trabajo que este puede rendir y está relacionado con el throughput. Los datos para obtener la capacidad se obtuvieron a partir de las ejecuciones de los test utilizados para calcular el throughput, observando la cantidad de procesos finalizados. La gráfica B.75 muestra el porcentaje de procesos finalizados en 30 minutos, de acuerdo a la cantidad de usuarios.

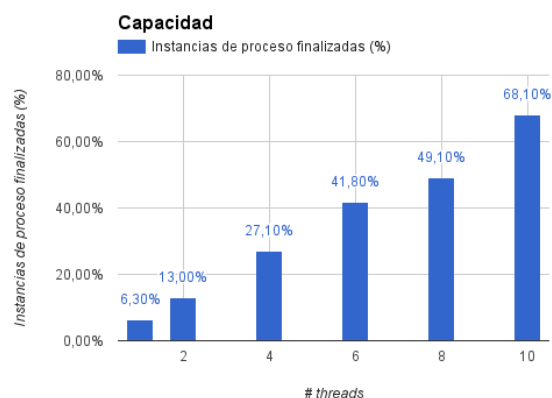


Figura B.75 Capacidad (%) frente a Usuarios

Tomando la mediana de los valores que se obtuvieron, la capacidad para la herramienta es: 34,45 %

B.12.3. Tabla de nivel de cumplimiento

La tabla B.105 muestra los resultados obtenidos para las características funcionales. Las características están divididas por categoría. De cada característica se muestra si la evaluación es teórica (Teó), el nivel de soporte (NS) y el nivel de cumplimiento (NC).

Tabla B.105 Características técnicas - Nivel de cumplimiento.

Modulo	Característica	Teó	NS	NC
	Tecnología, Arquitectura e Interoperabilidad			
	Arquitectura de BPMS	Si		
	Compatibilidad con estándares	Si		
	Base de datos	Si		
	Sistema operativo del modelador	Si		
	Navegador Web	Si		
	Extensibilidad	Si		
	Integración con Servicios Web	No		
	Diseño y modelado de procesos			
	Interfaz gráfica para key users	No		
	Tipo de modelador	Si		
	Notación gráfica de procesos compatible con BPMN 2.0	Si		
	Gestión de formularios			
	Definición de formularios	No		
	Formato de formularios	No		
	Elementos de formularios	No		
	Formularios dinámicos	No		
	Reglas de llenado de campos	No		
	Validación de datos del formulario	No		
	Motor de Workflow			
	Iniciación de procesos en forma manual y automática	No		
	Definición de reglas de negocio	No		
	Asignación de tareas	No		
	Reasignación de tareas	No		
	Adjuntar documentos a tareas	No		
	Versionado de procesos	No		

Modulo	Característica	Teó	NS	NC
	Tareas de borrador	No	●	⊘
	Gestión, monitoreo y auditoría			
	Gestión de instancias por usuarios	No	●	✓
	Mantenimiento histórico de procesos	No	●	✓
	Monitoreo de procesos	No	●	✓
	Portal			
	Presentación de las tareas	No	●	✓

Referencias:

- ⊘ No aplica
- ✓ Cumple de forma nativa
- ⚙ Cumple vía integración
- ✂ Cumple vía particularización
- Soporta totalmente
- Soporta parcialmente
- No soportada

B.12.3.1. Resultado numérico de las características técnicas

Las tablas B.107 y B.109 presentan los resultados numéricos correspondientes a las características funcionales y no funcionales evaluadas. En la sección 3.7 se describe el método de evaluación, donde se presentan los cálculos a realizar para poder clasificar a las características según los umbrales definidos.

Primero, hay que definir los valores de importancia, para poder definir los umbrales. Además, es necesario dar valores a los niveles de cumplimiento y de soporte.

Tabla B.106 Definición de constantes

Importancia		Nivel de cumplimiento		Nivel de soporte	
Alto	3	Nativo	2	Total	3
Medio	2	Vía integración	1	Parcial	2
Bajo	1	Vía particularización	1	No soportado	1

Las columnas de la tabla B.107 contienen: las características, importancia (I), nivel de soporte (NS), nivel de cumplimiento (NC), el factor (F) y el puntaje total (Total). El puntaje total se calcula de la siguiente manera:

Puntaje total

$$Puntaje = I * NS * NC * Factor \quad (B.15)$$

Donde I es la importancia, NS el nivel de soporte, NC el nivel de cumplimiento y Factor un valor entre 0,1 y 0,9.

Tabla B.107 Características técnicas - Calculo del puntaje.

Característica	I	NS	NC	F	Total
Tecnología, Arquitectura e Interoperabilidad					
Compatibilidad con estándares	1	1	2	1	2
Base de datos	1	1	2	1	2
Sistema operativo del modelador	1	2	2	1	4
Navegador Web	1	2	2	1	4
Extensibilidad	1	1	1	1	1
Integración con Servicios Web	1	2	1	1	2
Diseño y modelado de procesos					
Interfaz gráfica para key users	1	1	2	1	2
Tipo de modelador	1	1	2	1	2
Notación gráfica de procesos compatible con BPMN 2.0	1	1	2	1	2
Gestión de formularios					
Definición de formularios	1	0	0	1	0
Formato de formularios	1	0	0	1	0
Elementos de formularios	1	0	0	1	0
Formularios dinámicos	1	0	0	1	0
Reglas de llenado de campos	1	0	0	1	0
Validación de datos del formulario	1	0	0	1	0
Motor de Workflow					
Iniciación de procesos en forma manual y automática	1	1	2	1	2
Definición de reglas de negocio	1	0	0	1	0
Asignación de tareas	1	1	2	1	2
Reasignación de tareas	1	1	2	1	2
Adjuntar documentos a tareas	1	0	0	1	0
Versionado de procesos	1	2	2	1	4
Tareas de borrador	1	0	0	1	0

Característica	I	NS	NC	F	Total
Gestión, monitoreo y auditoría					
Gestión de instancias por usuarios	1	1	2	1	2
Mantenimiento histórico de procesos	1	1	2	1	2
Monitoreo de procesos	1	1	2	1	2
Portal					
Presentación de las tareas	1	1	2	1	2

Por otro lado tenemos los cálculos de las características no funcionales. Las columnas de la tabla B.109 contienen: el peso de cada subcaracterística (P_i), el nivel de cumplimiento de la subcaracterística (s_i), el valor final de la subcaracterística definido en la ecuación B.18 y el nivel de cumplimiento (NC). El nivel de cumplimiento se obtiene de la siguiente manera:

- $vf < 1 \rightarrow$ Nivel de cumplimiento Bajo.
- $1 \leq vf < 3 \rightarrow$ Nivel de cumplimiento Medio.
- $3 \leq vf \rightarrow$ Nivel de cumplimiento Alto.

Valor final

$$vf = \frac{\sum_{i=1}^N (s_i * P_i)}{\sum_{i=1}^N P_i} \quad (B.16)$$

Siendo vf el ‘valor final’ de la característica y N la cantidad de subcaracterísticas.

En el caso de las características incluidas en la categoría Performance, hay que definir cuales son los valores de $L1$ y $L2$ para cada una de ellas, de esta manera, se podrá determinar el valor final de cada característica. Para determinar estos valores se consideraron estudios previos de performance así como de usabilidad [47] [42]. A continuación se muestra una tabla con los valores determinados:

Tabla B.108 Parámetros

Característica	L1	L2
Tiempo de respuesta	3 seg	10 seg
Throughput	30 %	80 %
Capacidad	20 %	70 %

Se considera que todas las subcaracterísticas son importantes y en este sentido se asigna el valor 1 al peso de cada subcaracterística.

Tabla B.109 Características técnicas - Cálculo del puntaje.

[illegible]

B.13. Evaluación de ProcessMaker

El software se obtuvo a través de un contacto de la empresa luego del registro en la página de la herramienta.

B.13.1. Evaluación de las características teóricas

B.13.1.1. Arquitectura del BPMS

ProcessMaker es una solución Open Source que permite el desarrollo de workflow para la automatización y mejora de procesos de negocio. El software está basado completamente en una aplicación web, esto facilita la coordinación del flujo de trabajo entre usuarios, departamentos y organizaciones. ProcessMaker puede interconectarse con sistemas que incluyen la gestión de documentos, ERP, CRM y otras aplicaciones, ya que funciona como una aplicación de SOA de gran alcance.

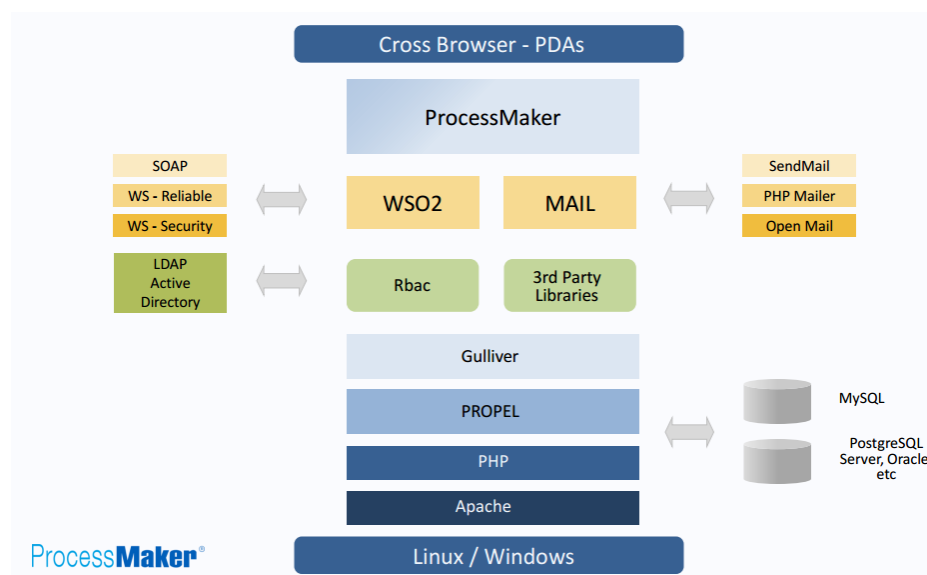


Figura B.76 Arquitectura BPM ProcessMaker [36]

[36] describe la arquitectura del sistema ProcessMaker, la figura B.76 muestra los principales componentes de la arquitectura, los cuales se describen a continuación:

WSO2: se utiliza para encriptar la información SSL de 128 bits, esto asegura la confidencialidad e integridad de los datos que forman parte del flujo, o al momento de establecer comunicación con otros sistemas a través de servicios web.

MAIL: este componente permite establecer la comunicación de correos electrónicos a través de cuentas estándares como SMTP.

Rbac: permite el manejo de base de datos de administración de cuentas de usuarios. Además tiene la capacidad de integrarse al directorio activo de Windows.

3er Party Libraries: en este componente se incorporaran las librerías plug-in para la integración con otros sistemas.

GULLIVER: es un framework de desarrollo que está compuesto de una serie de clases que permiten desarrollar nuevas funcionalidades.

Cross browser: permite que el acceso a ProcessMaker sea desde cualquier navegadores web (Internet Explorer, Netscape, Mozilla y otros.) Esto garantiza que se pueda trabajar en cualquier tipo de plataforma.

PROPEL: es la técnica de programación ORM (Object Relational Mapping) que convierte datos entre diferentes tipos de sistemas, en base de datos relacional orientado a objetos. Además la técnica permite tener preparado el sistema para manejar base de datos virtual orientado a objetos y, poder ser utilizado por cualquier lenguaje de programación permitiendo el intercambio de información con cualquier base de datos (Oracle, SQL Server, Postgress, MySQL, etc.)

B.13.1.2. Compatibilidad con estándares

ProcessMaker permite importar/exportar procesos en formato .pmx, además es posible importar procesos en formato .pm y .bpmn. Mientras que los diagramas se importan/exportar en formato BPMN. [37] muestra como importar y exportar los proyectos.

B.13.1.3. Base de datos

Soporta conexión con la base de datos MySQL, versión 2.5 o superior. [37] muestra como acceder a la base de datos y otros detalles relacionados al acceso y tratamiento de la información.

Por otro lado, a través de la librería PROPEL es posible acceder a bases de datos externas como Microsoft SQL Server, PostgresSQL y Oracle [37]. Se logró conectar a una base de datos PostgresSQL versión 9.4.

B.13.1.4. Sistema operativo del modelador

Dentro de la aplicación web se encuentra el modelador de ProcessMaker. Por lo tanto, el sistema operativo del modelador es un sistema capaz de soportar Apache y PHP, por ejemplo, Linux y Windows.

B.13.1.5. Navegador web

[37] muestra los distintos navegadores web en los cuales ProcessMaker funciona correctamente. La versión Community 3.0 se puede utilizar en los siguientes navegadores:

- Internet Explorer, versiones 10 y 11.
- Mozilla Firefox, versiones 37 y 38.
- Google Chrome, versiones 42 y 43.

Sin embargo, no es compatible con navegadores mobile, pero a partir de la versión 3.0, ProcessMaker provee una aplicación mobile para Android e iOS.

Por otro lado, se ha demostrado un correcto funcionamiento en los navegadores:

- Mozilla Firefox, versión: 43 en Windows 10.
- Google Chrome, versión: 47 en Windows 10.

B.13.1.6. Extensibilidad

Es posible integrar a ProcessMaker con diferentes tipos de software incluyendo CRM, ERP, HRM, software para firmas digitales entre otros.

[63] muestra como integrar diferentes sistemas tales como Alfresco, Outlook, Sharepoint, VMWare Zimbra, también permite la integración con LDAP.

B.13.1.7. Tipo de modelador

El modelador que provee ProcessMaker está integrado a la aplicación web del sistema, por lo tanto el tipo de modelador es web. Se probó un correcto funcionamiento en Google Chrome, versión 47.

B.13.1.8. Notación gráfica de procesos compatible con BPMN 2.0

[37] detalla los elementos de BPMN 2.0 que soporta y los que no soporta ProcessMaker. Dentro de los elementos soportados están las tareas, subprocesos, los eventos de inicio (vacío, recepción de mensaje), intermedios (envío y recepción de mensaje) y de fin (vacío, envío de mensaje). Los gateways disponibles son: exclusive, parallel e inclusive. Tanto el inclusive gateway como el exclusive gateway permiten definir el flujo por defecto. También soporta pool y lanes, artefactos y flujos de secuencia y mensajes.

Si bien es cierto que soporta tareas, la diferencia entre tareas de tipo Script, Receive Message, Send Message, User Task, Service, Manual y Business Rules Task se utiliza solo para propósitos de diseño, ya que el motor no soporta la ejecución de este tipo de tareas.

Para finalizar, los elementos que pertenecen a BPMN 2.0 y no fueron mencionados anteriormente, no son soportados por el motor de ProcessMaker, por ejemplos, los boundaries que son elementos que se adjuntan a las tareas.

B.13.2. Evaluación de las características prácticas

B.13.2.1. Integración con servicios web

WSDL Web Services provee una interfaz de programación para conectar e integrar a ProcessMaker con sistemas externos utilizando protocolos SOA. También permite acceder de forma remota a las funcionalidades de ProcessMaker utilizando tanto SOAP como WSDL. Para establecer una conexión con un servicio web, se debe crear un trigger, el cual se asocia a la tarea. [37] brinda más información acerca de la integración con servicios web que posee

ProcessMaker.

Para realizar las pruebas se utilizó un servicio de tipo WSDL.

B.13.2.2. Interfaz gráfica para key users

ProcessMaker provee un modelador que permite que el diseño del proceso lo puede llevar a cabo un key user. A pesar de esto, cuando el proceso presenta conexiones a bases de datos o a servicios web, es necesaria la intervención de un usuario con formación técnica.

Para configurar el acceso a una base de datos externas no es necesario tener conocimiento técnico. A pesar de esto, para acceder a los datos, se necesita realizar una consulta en el lenguaje SQL, para lo cual, es necesario tener conocimiento técnico.

Los servicios web se acceden a través de *triggers*. El código de acceso al servicio web debe ser escrito en lenguaje PHP, para lo cual se requiere conocimiento técnico.

B.13.2.3. Definición de formularios

La definición de formularios se realiza a través de la creación de DynaForms, los cuales presentan una interfaz intuitiva para usuarios no técnicos [37]. Los elementos se agregan al formulario mediante *drag-and-drop*, una vez agregados se pueden establecer reglas de llenado sobre los campos, sin necesidad de programación. Por otro lado, los DynaForms permiten agregar javascript para reglas de llenado más complejas.

B.13.2.4. Formato de formularios

Dentro de la edición de los formularios es posible modificar la apariencia de los campos a través de javascript. Cada elemento del formulario tiene un campo identificador que es único, con dicho campo se puede, vía javascript, modificar la apariencia del elemento en el formulario. Además, es posible definir placeholder, mensaje de error (si no cumple con la validación) e información sobre el campo.

Datos

A

B *

C

D

Ingrese un correo

Administrator

Ingrese un número entero

Figura B.77 Apariencia de los elementos en los formularios

Además, los campos tienen tamaño dinámico, es decir, se adaptan al tamaño del dato a completar. Los formularios se pueden importar y exportar. Esto permite reutilizarlos

y extenderlos. También son independientes de las tareas, permitiendo asociar el mismo formulario a más de una actividad, sin necesidad de crearlo cada vez.

B.13.2.5. Elementos de formularios

La lista desplegable es uno de los elementos que se puede incluir en un formulario. Los valores de la lista se pueden cargar mediante una consulta a la base de datos del servidor o a una base de datos externa. Para realizar una consulta a una base de datos externa, es necesario crear una conexión a dicha base, ver figura B.78.

Fields marked with asterisk (*) are required.

Figura B.78 Creación de conexión a base de datos externa

De la misma manera, los DynaForms permiten incluir botones dentro de los formularios. Para definir acciones sobre el botón es necesario agregar código javascript en las propiedades del formulario, en particular, las acciones de los botones pueden estar habilitadas/deshabilitadas, además se pueden mostrar/ocultar los botones dependiendo, por ejemplo, de las variables del proceso.

B.13.2.6. Formularios dinámicos

Como indica su nombre, los DynaForms son formularios dinámicos que posibilitan mostrar/ocultar campos dependiendo del valor de otro campo dentro del formularios, como se puede observar en la figura B.79. Principalmente, las listas desplegables pueden estar relacionadas, por ejemplo, si el formulario cuenta con una lista de países que se carga desde una base, una lista de ciudades también se puede cargar de la base, pero la consulta depende del valor del país. Por lo tanto, la lista de ciudades permanecerá vacía hasta que un país es seleccionado.

The image displays two screenshots of a web form titled "Datos". The top screenshot shows a form with three visible fields: "A" (text input with placeholder "Ingrese un correo"), "B" (text input with a red asterisk and an information icon), and "C" (dropdown menu with "Administrator" selected). The bottom screenshot shows the same form, but field "D" is now visible as a blue rectangular box, while field "C" now displays "Desarrolladorwww" with a dropdown arrow. This illustrates how fields can be shown or hidden based on certain conditions.

Figura B.79 Mostrar/ocultar campos en el formulario

B.13.2.7. Reglas de llenado de campos

Las reglas de llenado de campos en los formularios de ProcessMaker se pueden definir tanto dentro como fuera del mismo. Para tener acceso a aplicaciones externas, por ejemplo a un servicio web, es necesario definir un trigger con la consulta, a su vez, el resultado se puede cargar en una variable del proceso. De igual modo, los cálculos algorítmicos se pueden hacer dentro de un trigger o mediante javascript. La información de la ejecución del proceso está disponible en la base de datos, y se puede consultar con el identificador del proceso o la tarea, este dato se encuentran dentro de las variables del proceso que están predefinidas.

Para finalizar, se pueden definir reglas para la actualización automática de campos mediante el acceso a la base de datos, ver sección B.13.2.6. Cabe destacar que a partir de la versión 3.0.1.6, es posible realizar consultas a la base de datos a partir del llenado de campos de textos, de la misma manera que ya podía realizarse con las listas desplegables.

B.13.2.8. Validación de campos de formularios

Los formularios de ProcessMaker presentan diferentes vía para validar un campo. Cada campo debe tener una variable asociada y esta tiene un tipo, por lo tanto, cuando se completa el campo este tipo se debe respetar, de lo contrario no es posible dar como completado el formulario.

De hecho, los DynaForms permiten agregar una expresión regular para validar el campo. Si el campo no cumple con la expresión regular, este queda en rojo, pero no hay un mensaje de

error descriptivo, para estos casos es conveniente utilizar la opción de completar el campo *validation error message*, de esa manera el usuario entenderá porque no es válido el dato, ver figura B.80

En segundo lugar, es posible indicar cuando un campo es requerido. Para ello se selecciona la opción *required* y al lado del campo aparecerá un asterisco rojo (*). Si el campo no es completado y se quiere finalizar la tarea, aparece un cartel rojo indicando que el campo es requerido.

The figure displays two screenshots of a web form titled "Datos".

Top Screenshot (Validation Errors):

- Field A:** A text input containing "esto no es un correo". Below it is a red error message: "La dirección de correo no es correcta".
- Field B:** A text input that is empty. It is marked as required with a red asterisk (*) and a red error message: "This field is required".
- Field C:** A dropdown menu showing "Administrator".
- Field D:** A blue rectangular button.

Bottom Screenshot (Valid Data):

- Field A:** A text input containing "fing@fing.com".
- Field B:** A text input containing "123".
- Field C:** A dropdown menu showing "Administrator".
- Field D:** A blue rectangular button.

Figura B.80 Validación de datos en formularios

B.13.2.9. Iniciación de procesos en forma manual y automática

Las instancias de procesos se pueden iniciar tanto de forma manual como de forma externa, pero no de forma automática. Para iniciar un proceso de forma manual, el usuario debe estar asignado como responsable de la primer tarea del proceso y además debe tener permisos para crear procesos o pertenecer a un grupo que tenga dicho permiso. Si el usuario cuenta con dicho permiso, dentro de la pestaña *Home* verá la opción *New Case*, donde aparecen procesos que el usuario puede iniciar.

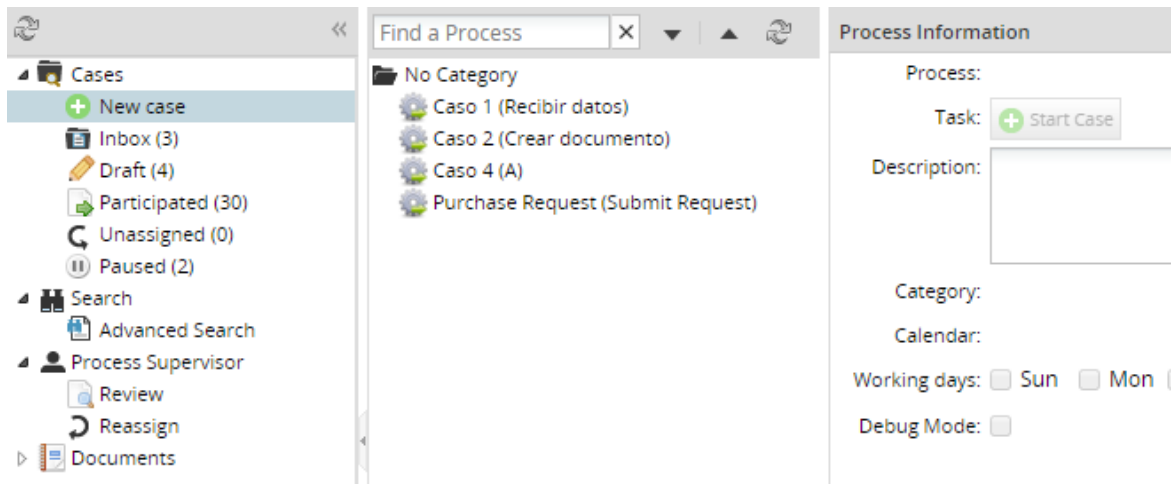


Figura B.81 Iniciar instancia de un proceso de forma manual

ProcessMaker cuenta con una manera muy sencilla de iniciar procesos de forma externa. Basta con seleccionar la opción *Web Entry* del menú del evento de inicio vacío y luego completar los datos. Una vez generado el link, se puede utilizar sin necesidad de ser usuario de sistema.

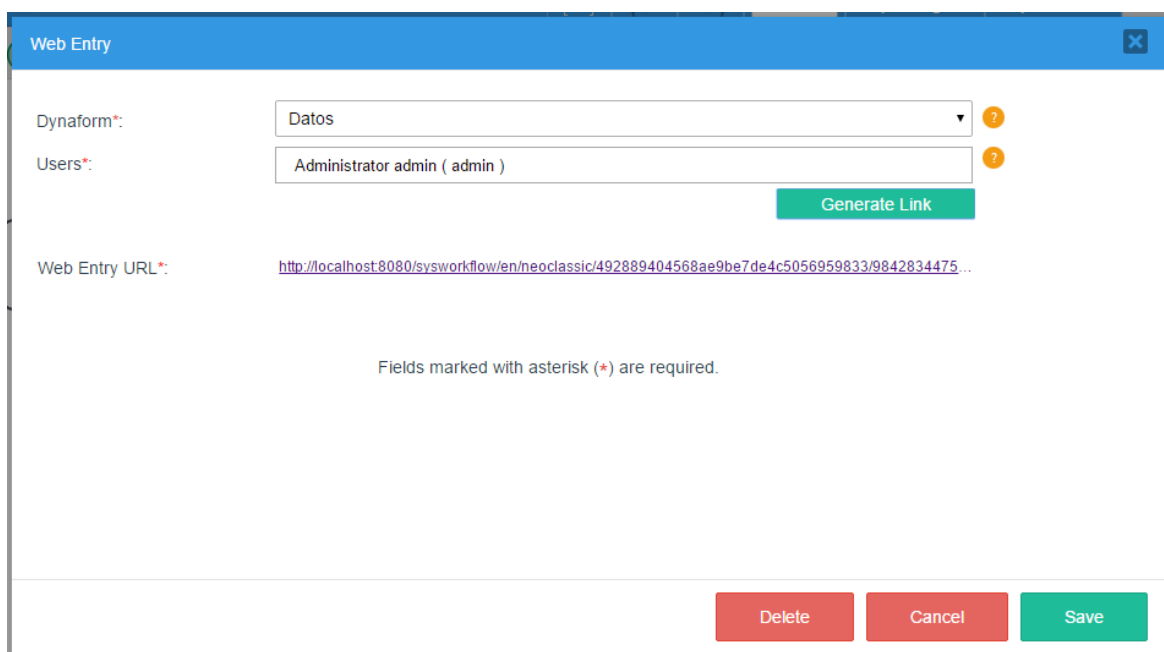


Figura B.82 Generar link

La versión Community de ProcessMaker no tiene la opción de iniciar las instancias de los procesos de acuerdo a una agenda, si bien se puede ingresar a las opciones y configurar el

inicio, el motor no soporta esta funcionalidad y por lo tanto las instancias nunca inician y tampoco da la opción de iniciar la instancia de forma manual.

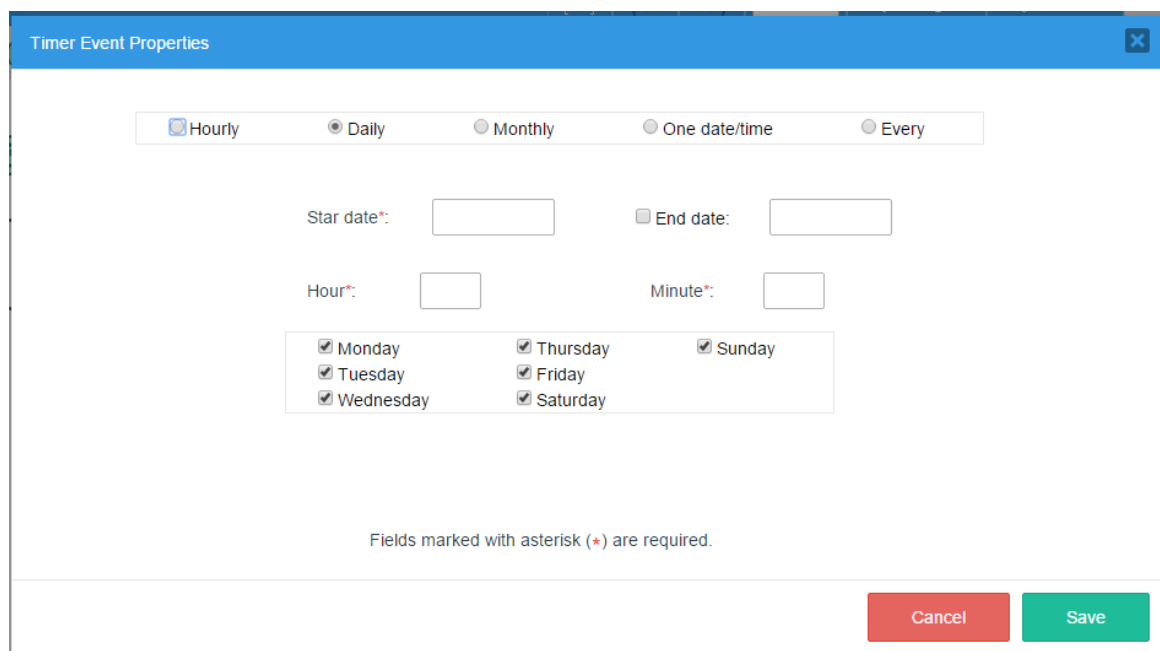


Figura B.83 Configurar agenda para inicio de proceso

B.13.2.10. Definición de reglas de negocio

Si bien cumple con parte del estándar BPMN 2.0, ver sección B.13.1.8, la tarea de tipo *Business Rule Task* es solo para el modelo, ya que el motor no soporta la funcionalidad. La versión Community de ProcessMaker no trae un motor de reglas de negocios y tampoco permite la integración con un motor externo.

B.13.2.11. Asignación de tareas

La herramienta presenta diferentes formas de asignar una tarea, cíclica, manual, basada en un valor, “reportar a”, “self service”, “self service” basada en un valor, ver [37].

Cada tipo de asignación tiene sus características que se explicarán a continuación:

Asignación cíclica: es la asignación por defecto, cuando la tarea es asignada, utilizando el patrón round-robin para seleccionar a un usuario de la lista de usuarios asignados, ver figura B.84

Asignación manual: el usuario que finaliza la tarea previa, debe seleccionar de la lista de usuarios asignados, quien realizará la siguiente tarea.

Asignación basada en un valor: permite asignar un usuario (o grupo de usuarios) a una tarea mediante una variable del proceso, la cual se carga con el identificador del usuario

(o del grupo). En este tipo de asignación no es necesario definir una lista de usuarios asignados.

Reportar a: este tipo de asignación toma en cuenta la estructura de la organización, que está representada por los departamentos definidos en la sección *Admin* de la herramienta. Es una opción recomendable cuando un supervisor necesita revisar el trabajo de las personas de su departamento.

Self-service: permite a cualquier usuario de la lista de usuarios asignados tomar la tarea. Este tipo de asignación puede ser utilizado para reducir la congestión en el workflow.

Self-service basado en un valor: este tipo de asignación es una combinación de self-service y asignación basada en un valor. Se define una variable, en la cual se asigna de forma dinámica usuarios (uno o más), o grupo de usuarios, los cuales serán capaces de reclamar la tarea.

Figura B.84 Tipos de asignación de usuarios a un tarea

Cuando una tarea tiene un usuario asignado, dicha tarea es visible para los usuarios que participaron de la instancia del proceso. Sin embargo, solo el usuario que tiene asignada la tarea puede completar el formulario, el resto de los usuarios que participaron en la instancia del proceso podrán ver un resumen de la instancia y datos de la tarea actual, por ejemplo, el usuario que la tiene asignada.

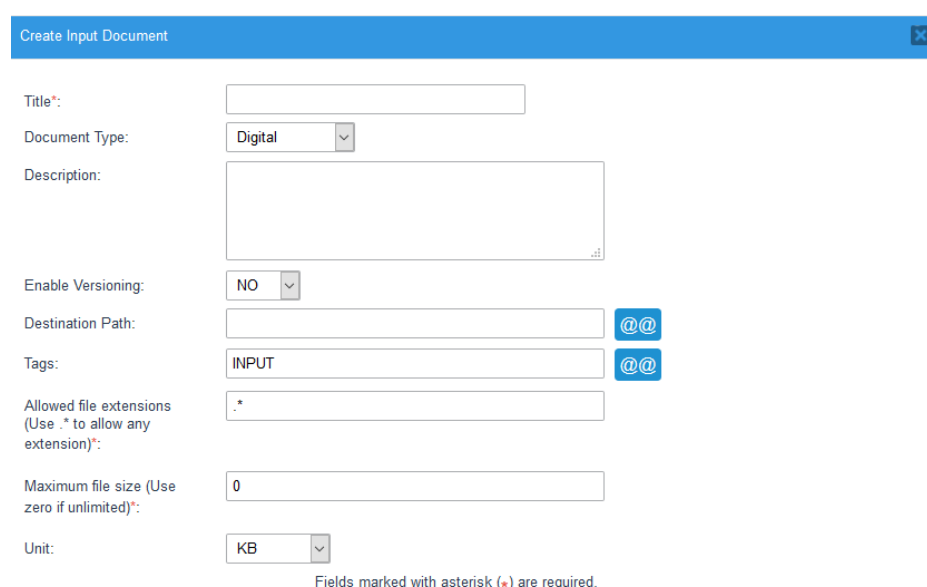
B.13.2.12. Reasignación de tareas

Es posible reasignar tareas en ProcessMaker. Para ello, el usuario debe tener permiso para reasignar casos [37]. Por otro lado, la reasignación automática se debe programar a través de eventos, ver [37]

B.13.2.13. Adjuntar documentos a tareas

La herramienta permite adjuntar y generar archivos en una tarea. En la etapa de modelado, se presenta la posibilidad de crear archivos de tipo *input* y *output*. Dentro de los archivos de tipo *input*, se debe especificar el nombre, los tipos de archivos permitidos y el tamaño máximo, ver figura B.85 Después de crear un documento, hay que decidir si será posible subirlo en una o más instancias. Cada usuario puede cargar más de un documento en los campo de tipo *input*.

Si los documentos necesitan ser cambiados a lo largo de una instancia, los *input document* brindan la posibilidad de habilitar el versionado de los mismos. Todos los documentos son accesibles desde el menú Documents, en la pestaña *Home*.



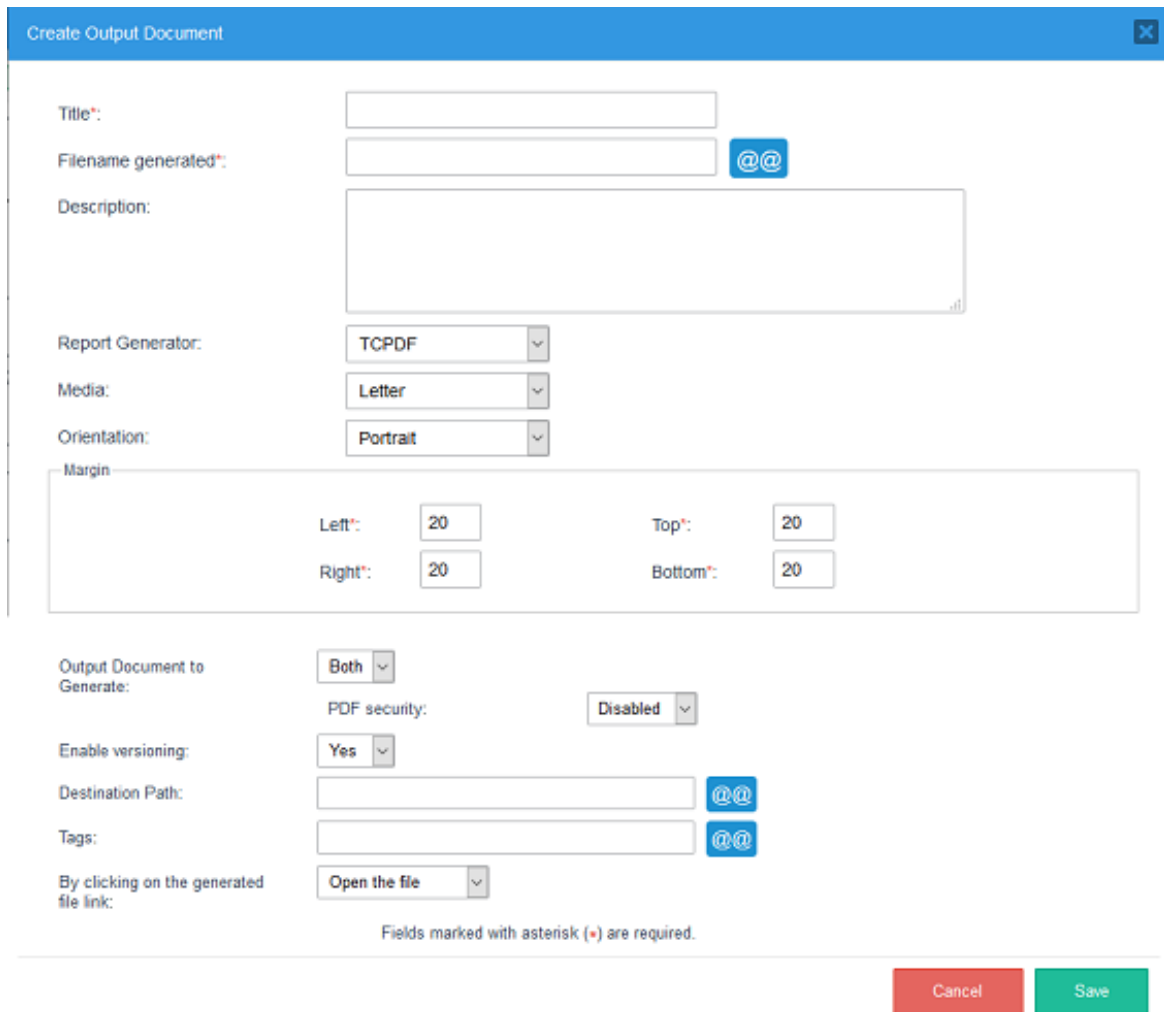
The screenshot shows a 'Create Input Document' dialog box with the following fields and options:

- Title*:** A text input field.
- Document Type:** A dropdown menu with 'Digital' selected.
- Description:** A large text area.
- Enable Versioning:** A dropdown menu with 'NO' selected.
- Destination Path:** A text input field with a blue '@@' icon to its right.
- Tags:** A text input field with 'INPUT' entered and a blue '@@' icon to its right.
- Allowed file extensions (Use .* to allow any extension)*:** A text input field with '.*' entered.
- Maximum file size (Use zero if unlimited)*:** A text input field with '0' entered.
- Unit:** A dropdown menu with 'KB' selected.

At the bottom, a note states: 'Fields marked with asterisk (*) are required.'

Figura B.85 Creación de input document

De la misma manera, permite definir archivos de tipo *output*. Estos tipos de archivos son generados durante la ejecución de una instancia, los cuales se pueden acceder mediante un link. Son útiles para exportar datos de la instancia. Al momento de crear un *output document* se debe ingresar el título, el nombre del archivo generado y las dimensiones del documento, ver figura B.86



Create Output Document

Title*:

Filename generated*: @@

Description:

Report Generator:

Media:

Orientation:

Margin

Left*: Top*:

Right*: Bottom*:

Output Document to Generate:

PDF security:

Enable versioning:

Destination Path: @@

Tags: @@

By clicking on the generated file link:

Fields marked with asterisk (*) are required.

Cancel Save

Figura B.86 Creación de output document

Una vez creado el *output document*, se puede editar su contenido, por ejemplo, creando una plantilla HTML con variables del proceso, las cuales se insertan automáticamente al momento de generar el documento. La figura B.87 muestra el dialogo de edición de un *output document*.

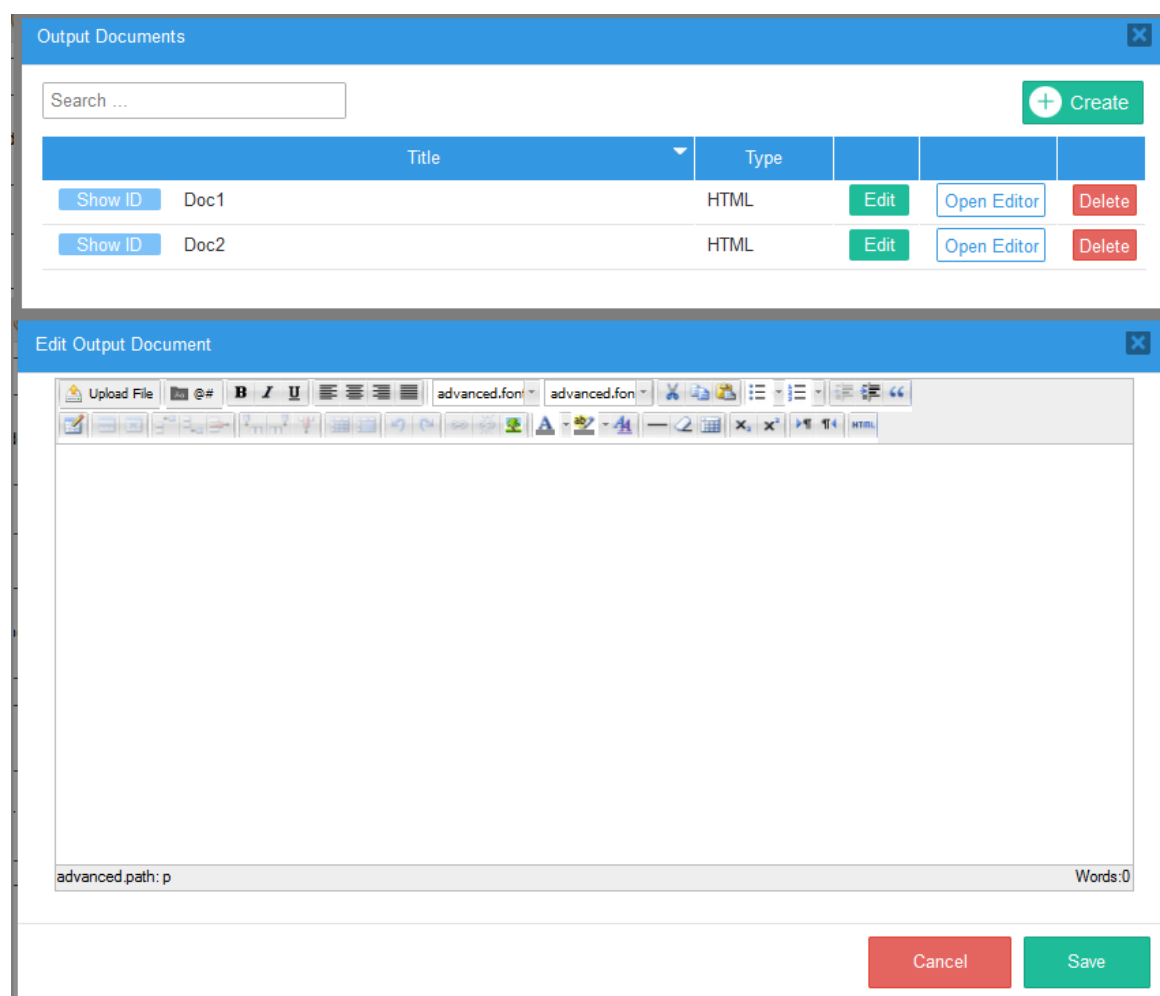


Figura B.87 Edición de output document

B.13.2.14. Versionado de procesos

No es posible tener más de una versión del proceso activa. De hecho, solo la última versión del proceso está activa. En caso de haber instancias en ejecución, al generar una nueva versión del proceso, las instancias en ejecución migran a la versión actual de forma automática. La principal ventaja de este comportamiento es que permite corregir errores detectados cuando se ejecuta una instancia. A pesar de esto, si el proceso se modifica en tareas previas a la tarea que está pendiente en una instancia del proceso, la instancia puede tener comportamientos impredecibles.

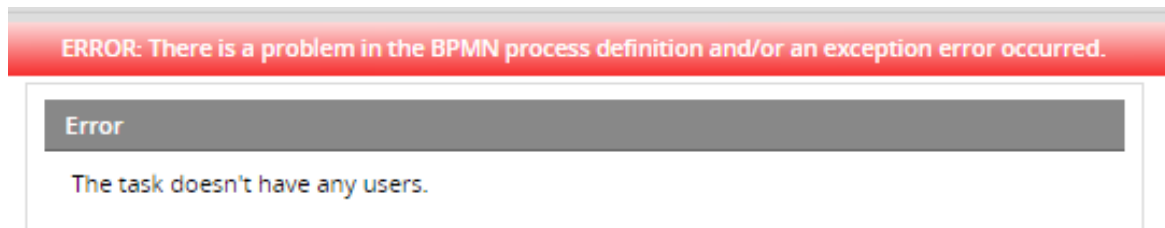


Figura B.88 Error en formulario

B.13.2.15. Tareas de borrador

Cada tarea tiene definida una serie de pasos que se ingresan al momento de diseñar el proceso. El usuario debe ingresar al menos un paso (por ejemplo, un formulario), aún así, cuando se quiere completar una tarea, el sistema genera un paso adicional, donde efectivamente se finaliza la tarea y se indica quien es el usuario que tiene asignada la siguiente tarea a realizar. Si es la última tarea a realizar, entonces se indica que es el fin del proceso.

Lo antes mencionado, permite guardar temporalmente los datos de una tarea. Para ello, el usuario debe completar el formulario, pero no terminar la tarea, así, los datos quedaran guardados incluso si la tarea es reasignada.

B.13.2.16. Gestión de instancias por usuarios

Los usuarios que pueden iniciar instancias de un procesos son aquellos, que además de tener permisos para crear instancias, deben estar en la lista de usuarios asignados a la primer tarea del proceso.

Solamente el usuario que tiene asignada la tarea actual del proceso, es capaz de pausar o reanudar la instancia. Las instancias que se pueden borrar son las que se encuentran en *Draft*, ya que aún no se ha completado la primer tarea, una vez que pasan al *Inbox* no se podrán borrar. Pero si es posible cancelarla, si el usuario tiene permisos para cancelar casos.

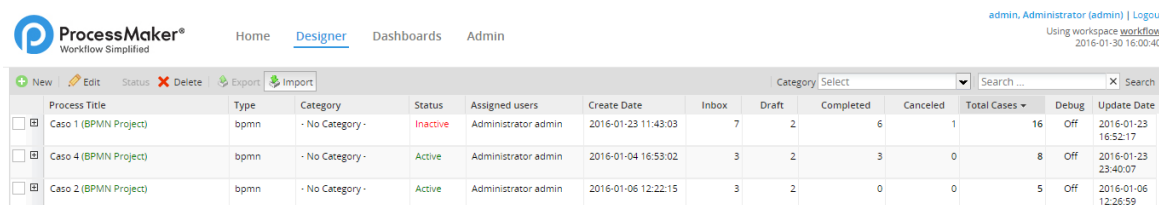
Los usuarios con permisos para ver el *Designer*, pueden activar/desactivar procesos. En ese caso las instancias que estaban activas se podrán seguir ejecutando hasta finalizar el flujo. No se podrán iniciar instancias nuevas.

Por último, no es posible sobrescribir manualmente datos de las instancias de los procesos.

B.13.2.17. Mantenimiento del histórico del proceso

Para poder ver el mantenimiento histórico del proceso, el usuario necesita permiso para acceder al *Designer*. Esto le permite ver el estado (activa/inactiva), la fecha de creación, el usuario asignado, cantidad de instancias con tareas para hacer, iniciadas pero sin completar

la primer tarea, completadas y canceladas.



The screenshot shows the ProcessMaker Designer interface. At the top, there's a navigation bar with 'Home', 'Designer', 'Dashboards', and 'Admin'. Below it, a table lists process instances. The table has columns for Process Title, Type, Category, Status, Assigned users, Create Date, Inbox, Draft, Completed, Canceled, Total Cases, Debug, and Update Date.

Process Title	Type	Category	Status	Assigned users	Create Date	Inbox	Draft	Completed	Canceled	Total Cases	Debug	Update Date
Caso 1 (BPMN Project)	bpmn	- No Category -	Inactive	Administrator admin	2016-01-23 11:43:03	7	2	6	1	16	Off	2016-01-23 16:52:17
Caso 4 (BPMN Project)	bpmn	- No Category -	Active	Administrator admin	2016-01-04 16:53:02	3	2	3	0	8	Off	2016-01-23 23:40:07
Caso 2 (BPMN Project)	bpmn	- No Category -	Active	Administrator admin	2016-01-06 12:22:15	3	2	0	0	5	Off	2016-01-06 12:26:59

Figura B.89 Histórico del proceso

Todo usuario que participe de la instancia de un proceso podrá ver su histórico. Esto le permite ver el mapa del proceso, donde puede ver tareas pendientes, completadas, en progreso. Haciendo click sobre la tarea tiene más información (por ejemplo, el usuario que realizó la tarea). También puede ver información sobre tiempo de duración de la tarea, los documentos (con opción de descarga), el histórico de mensajes, etc.

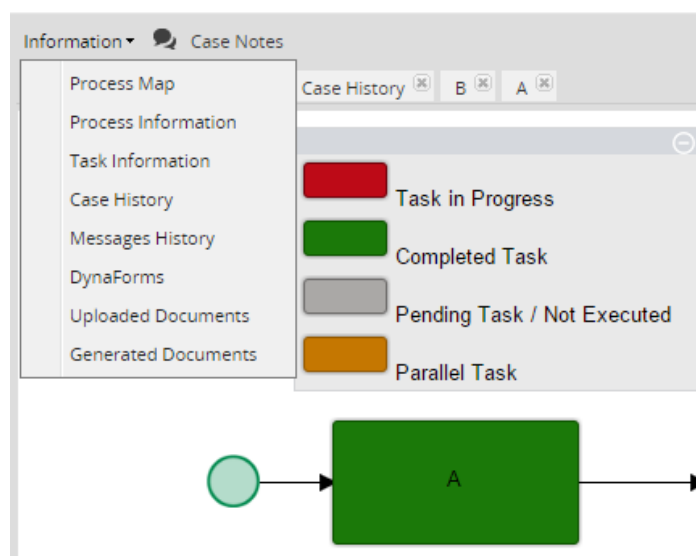


Figura B.90 Histórico de la instancia

Si un usuario tiene permiso para ver todos los casos, entonces puede acceder a la información de las instancias de los procesos aunque no haya participado en ellas. En el menú *Home* aparece una opción de búsqueda avanzada que permite filtrar las instancias por categoría, proceso, usuario, estado (no funciona) y un período de fechas de asignación de tareas.

Open

CategoryAll Categories

ProcessAll Processes

StatusAll Status

UserAll Users

Delegated Date From

To

Search ...

Search

#	Summary	Case Notes	Case	Process	Task	Current User	Last Modify	Delegation Date	Due Date	Status
23			#23	Caso 1	Verificar datos	Doe, John (jdoe)	2016-01-30 14:38:34	2016-01-30 15:38:03	2016-02-01 17:00:00	To do
19			#19	Caso 1	Verificar datos	admin, Administrator (admin)	2016-01-30 14:37:18	2016-01-23 16:30:25	2016-02-01 17:00:00	To do
24			#24	Caso 4	B	Castelli, Ale (acastelli)	2016-01-25 18:47:46	2016-01-23 20:41:52	2016-01-26 17:00:00	To do
25			#25	Caso 4	B	Castelli, Ale (acastelli)	2016-01-25 18:45:44	2016-01-23 20:51:03	2016-01-26 17:00:00	To do
29			#29	Caso 1	Recibir datos	Castelli, Ale (acastelli)	2016-01-25 18:41:07	2016-01-25 19:40:44	2016-01-26 17:00:00	Draft
30			#30	Caso 4	B	Castelli, Ale (acastelli)	2016-01-25 18:43:57	2016-01-25 19:43:57	2016-01-26 17:00:00	To do

Figura B.91 Histórico de las instancias de los procesos

B.13.2.18. Monitoreo de procesos

Los usuarios que cuenten con permiso para ver el *Dashboard*, podrán ver el porcentaje de instancias finalizadas ese día. Si bien ProcessMaker permite agregar plugins, esta opción solo está disponible para la versión Enterprise y no para la Community[37]. Sin embargo, es posible definir tablas de reportes.

Las tablas de reporte se definen dentro del modelo de proceso, se indica los datos que se quieren registrar del proceso, y además la base de datos donde se guardarán. Los usuarios que tienen permiso para ver las tablas de reporte, no solo pueden crear tablas dentro de los procesos, sino también acceder a los datos de las mismas. Más información en [37].

B.13.2.19. Presentación de las tareas

Al ingresar al portal de ProcessMaker, si el usuario solamente tiene permiso para ver el *Home*, entonces verá el *Inbox* con sus tareas activas. En caso que el usuario tenga permiso para ver más cosas, al iniciar el portal verá la pestaña que está a la derecha de la pantalla.

Una vez en el listado de tareas es posible filtrar las instancias por proceso, categoría y por parte del nombre de la tarea. Es posible ordenar de forma ascendente o descendente cada columna, así como también seleccionar que columnas quedan visibles y cuales ocultas.

Si bien la herramienta no brinda vistas separadas para tareas pendientes, completadas, atrasadas y redireccionadas. Presenta cuatro vistas separadas, las mismas son: instancias iniciadas, tareas pendientes y atrasadas, tareas pausadas y todas las tareas del usuario.

Es posible agregar comentarios o mensajes en las instancias del proceso o las tareas[37]. Para ello hay que crear un permiso dentro del proceso, donde se indica que usuarios pueden crear notas, sobre que tarea se puede realizar el comentario, el estado en que tiene que estar la tarea, etc. [37]

B.13.2.20. Capacidad de aprendizaje

ProcessMaker es un portal web que trae integrado el modelador. Esto permite compartir los manuales de uso, la herramienta cuenta con una wiki [37]. pero el primer paso sugerido para aprender a utilizar la herramienta una vez instalada es seguir una demostración para ver los diferentes aspectos que presenta la herramienta, [38].

Tanto el modelador como el resto del portal poseen tooltips. En el portal los tooltips que se encuentran describen la acción del icono, la mayoría de los iconos tienen un texto descriptivo con la utilidad del mismo. Mientras que en el modelador, todos los iconos del modelo poseen tooltips, ya que carecen de texto.

Los mensajes de error que muestran el portal y el modelador son descriptivos. Si bien no brindan sugerencias para solucionar, el mensaje de error es lo suficientemente descriptivo para encontrar una solución.

Tabla B.110 Evaluación - Capacidad de aprendizaje

	Modelador	Portal
¿Posee tooltips?	SI	SI
¿Tiene mensajes de error descriptivos?	SI	SI
¿Brinda sugerencia de posible solución ante un error?	NO	NO
¿Posee manuales de uso?	SI	SI

B.13.2.21. Capacidad para ser usado

Al crear un nuevo modelo lo primero que ve el usuario es un diálogo de ayuda con varias sugerencias. Una vez que se cierra, es posible consultarlo mediante el icono de ayuda dentro de la barra de herramientas [37].

Tanto el modelador como el portal utilizan símbolos mnemotécnicos para representar acciones conocidas (borrar, agregar, minimizar, cerrar, etc.)

[37] es el espacio de preguntas frecuentes tanto para el portal como para el modelador. Para finalizar, si bien el portal tiene acceso directo al manual de uso, solo es posible acceder antes de iniciar sesión en el sistema.

Tabla B.111 Evaluación - Capacidad para ser usado

	Modelador	Portal
¿Brinda menú de ayuda?	SI	NO
¿Tiene símbolos mnemotécnicos?	SI	SI
¿Brinda un espacio de preguntas frecuentes?	SI	SI
¿Posee acceso directo a manuales de uso?	NO	NO

B.13.2.22. Protección contra errores de usuarios

Solamente el portal tiene mensajes de advertencia. Por ejemplo al borrar algo o cuando no se tiene permiso sobre algún objeto. La validación de los datos y los formularios se realiza tanto en el portal como en el modelador. Sólo el modelador tiene opciones de deshacer y rehacer una acción.

Tabla B.112 Evaluación - Protección contra errores de usuarios

	Modelador	Portal
¿Tiene mensajes de advertencia?	NO	SI
¿Posee validación de datos y formularios?	SI	SI
¿Cuenta con opciones para deshacer y rehacer una acción?	SI	NO

B.13.2.23. Estética de la interfaz de usuario

La información que se incluye en el portal no es innecesaria. Además, la información relevante se encuentra fácilmente. Sin embargo, el modelador cuenta con acciones que solo se pueden utilizar en la versión Enterprise, por ejemplo, el *Start timer event* que no es posible utilizar en la versión Community.

Tabla B.113 Evaluación - Estética de la interfaz de usuario

	Modelador	Portal
¿Incluye información innecesario?	SI	NO
La información relevante ¿Se encuentra accesible y fácil de encontrar?	SI	SI

B.13.2.24. Accesibilidad

La herramienta no cuenta con comando de voz, ni ningún tipo de sonido. Las imágenes que representan a las acciones son adecuadas, al igual que los colores. Sin embargo, la fecha de vencimiento de una tarea que es verde cuando no se cumplió y roja cuando está pasada de fecha, podrían no ser distinguibles para todas las personas.

Tabla B.114 Evaluación - Accesibilidad

	Modelador	Portal
¿Incluye comando de voz?	NO	NO
¿Las acciones se encuentran representadas con imágenes adecuadas?	SI	SI
¿Los colores son distinguibles aunque el usuario sea daltónico?	SI	NO
¿Los sonidos se corresponden con la acción que se está ejecutando?	NO	NO

B.13.2.25. Confidencialidad

Las contraseñas se guardan cifradas en la base de datos. Si bien los datos que se intercambian con el servidor no son sobre HTTPS, es posible configurar el servidor Apache para que utilice el protocolo.

Tabla B.115 Evaluación - Confidencialidad

	Portal
¿Utiliza técnicas criptográficas?	SI
Mobile ¿Almacena los datos cifrados en el dispositivo?	NO
¿Los intercambios de datos con el servidor son sobre HTTPS?	NO

B.13.2.26. Integridad

Un usuario que tenga permiso de usuarios es capaz de crear nuevos roles y asignar permisos a dichos roles, esto permite visualizar diferentes funciones según el rol del usuario.

Los documentos se guardan en un directorio dentro del espacio de trabajo en el cual se ha iniciado sesión dentro del portal. Si bien los documentos no se guardan cifrados, es posible asignar permisos sobre el documento, de manera tal, que aunque un usuario sin permiso puede acceder a él y descargarlo, no será capaz de ver su contenido. También es posible asignar permisos para crear, ver y eliminar carpetas, así como también para agregar archivos a las mismas.

El portal cuenta con un tiempo limite de sesión activa.

Tabla B.116 Evaluación - Integridad

	Portal
¿Permite definir roles?	SI
¿Cuenta con acceso diferenciado a los documentos según el rol?	SI
¿Permite visualizar diferentes funciones según el rol?	SI
¿Permite administrar permisos?	SI
¿Los documentos se guardan cifrados?	NO
¿Cuenta con un tiempo limite de sesión activa?	SI

B.13.2.27. No repudio

ProcessMaker cuenta con un historial de ejecución, ver sección B.13.2.17

Tabla B.117 Evaluación - No repudio

	Portal
¿Cuenta con un historial de ejecución de procesos?	SI
¿Se puede identificar que usuario y en que momento se ejecutó un proceso?	SI

B.13.2.28. Responsabilidad

Las tareas y procesos tienen un identificador que es único. Ambos identificadores están definidos como variables del proceso y se pueden utilizar para realizar acciones sobre los mismos, por ejemplo, en un trigger. No es posible modificar estos identificadores, pero si acceder a ellos.

Tabla B.118 Evaluación - Responsabilidad

	Portal
¿Las tareas y los procesos tienen identificador único?	SI

B.13.2.29. Autenticidad

ProcessMaker provee su propia forma de autenticación. La versión Enterprise permite agregar un plugin para integrar el sistema con bases externas como LDAP. Además la versión Enterprise trae integrado un plugin para que el sistema soporte firmas digitales, ver [64]

Tabla B.119 Evaluación - Autenticidad

	Portal
¿Cuenta con alguna forma de autenticación de usuarios?	SI
¿Soporta firmas digitales?	NO

B.13.2.30. Tiempo de respuesta

Para determinar el tiempo de respuesta se realizaron varias ejecuciones. El gráfico B.92 muestra el crecimiento del tiempo de respuesta, conforme aumenta la cantidad de usuarios. La función azul representa los valores obtenidos luego de la ejecución, mientras que la función naranja indica la tendencia. La tendencia corresponde a un polinomio de segundo grado, esto quiere decir que el tiempo de respuesta crece en proporción cuadrática de acuerdo a la cantidad de usuarios incorporados.

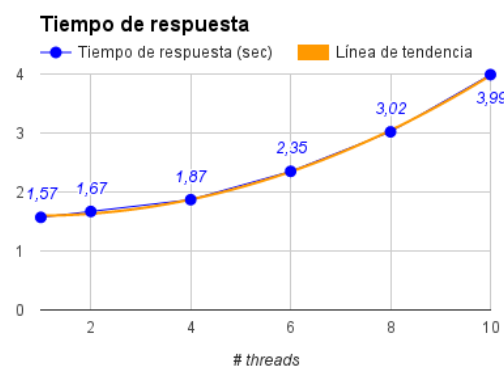


Figura B.92 Tiempo de respuesta

Tomando un promedio de los valores que se obtuvieron, el tiempo de respuesta para la herramienta es: 2,78 segundos

B.13.2.31. Throughput

El throughput refiere al número de eventos de respuesta en un intervalo. Los test utilizados para determinar el tiempo de respuesta de la herramienta, contribuyeron en la obtención del valor del throughput para la herramienta. En este sentido, se ejecutaron durante 30 minutos cada uno de los test. Luego, se utilizó la ecuación B.1 para calcular el throughput. La gráfica B.93 muestra el porcentaje de procesos iniciados en 30 minutos, de acuerdo a la cantidad de usuarios.

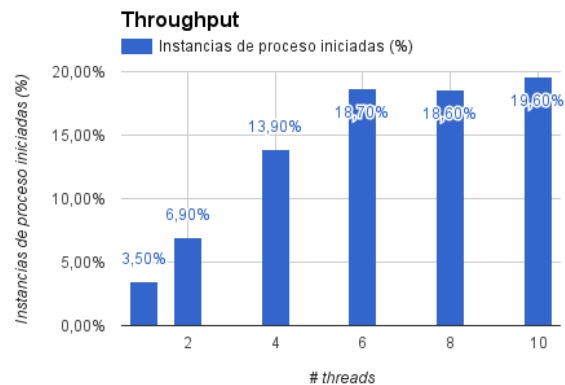


Figura B.93 Throughput (%) frente a Usuarios

Tomando la mediana de los valores que se obtuvieron, el throughput para la herramienta es: 16,25 %

B.13.2.32. Capacidad

La capacidad de un sistema es la cantidad de trabajo que este puede rendir y está relacionado con el throughput. Los datos para obtener la capacidad se obtuvieron a partir de las ejecuciones de los test utilizados para calcular el throughput. En este caso se observó la cantidad de procesos que finalizaron en 30 minutos, mientras que para el throughput se observó la cantidad de procesos iniciados. La gráfica B.94 muestra el porcentaje de procesos finalizados en 30 minutos, de acuerdo a la cantidad de usuarios.

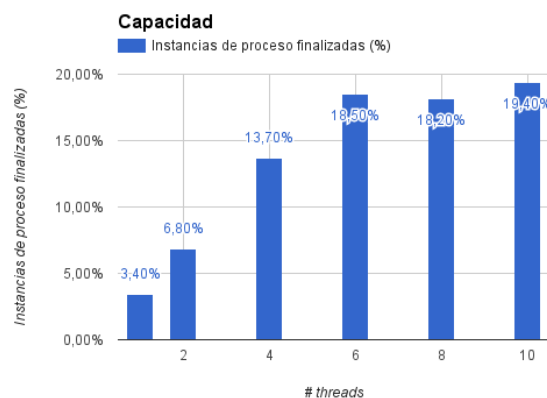


Figura B.94 Capacidad (%) frente a Usuarios

Tomando la mediana de los valores que se obtuvieron, la capacidad para la herramienta es: 15,95 %

B.13.3. Tabla de nivel de cumplimiento

La tabla B.120 muestra los resultados obtenidos para las características funcionales. Las características están divididas por categoría. De cada característica se muestra si la evaluación es teórica (Teó), el nivel de soporte (NS) y el nivel de cumplimiento (NC).

Tabla B.120 Características técnicas - Nivel de cumplimiento.

Modulo	Característica	Teó	NS	NC
Técnico				
	Tecnología, Arquitectura e Interoperabilidad			
	Arquitectura de BPMS	Si		
	Compatibilidad con estándares	Si		
	Base de datos	Si		
	Sistema operativo del modelador	Si		
	Navegador Web	Si		
	Extensibilidad	Si		
	Integración con Servicios Web	No		
	Diseño y modelado de procesos			
	Interfaz gráfica para key users	No		
	Tipo de modelador	Si		
	Notación gráfica de procesos compatible con BPMN 2.0	Si		
	Gestión de formularios			
	Definición de formularios	No		
	Formato de formularios	No		
	Elementos de formularios	No		
	Formularios dinámicos	No		
	Reglas de llenado de campos	No		
	Validación de datos del formulario	No		
	Motor de Workflow			
	Iniciación de procesos en forma manual y automática	No		
	Definición de reglas de negocio	No		
	Asignación de tareas	No		
	Reasignación de tareas	No		
	Adjuntar documentos a tareas	No		

Modulo	Característica	Teó	NS	NC
	Versionado de procesos	No	●	⊘
	Tareas de borrador	No	●	✓
	Gestión, monitoreo y auditoría			
	Gestión de instancias por usuarios	No	●	✓
	Mantenimiento histórico de procesos	No	●	✓
	Monitoreo de procesos	No	●	⊘
	Portal			
	Presentación de las tareas	No	●	✓

Referencias:

- ⊘ No aplica
- ✓ Cumple de forma nativa
- ⚙ Cumple vía integración
- ✂ Cumple vía particularización
- Soporta totalmente
- Soporta parcialmente
- No soportada

B.13.3.1. Resultado numérico de las características técnicas

Las tablas B.122 y B.124 presentan los resultados numéricos correspondientes a las características funcionales y no funcionales evaluadas. En la sección 3.7 se describe el método de evaluación, donde se presentan los cálculos a realizar para poder clasificar a las características según los umbrales definidos.

Primero, hay que definir los valores de importancia, para poder definir los umbrales. Además, es necesario dar valores a los niveles de cumplimiento y de soporte.

Tabla B.121 Definición de constantes

Importancia		Nivel de cumplimiento		Nivel de soporte	
Alto	3	Nativo	2	Total	3
Medio	2	Vía integración	1	Parcial	2
Bajo	1	Vía particularización	1	No soportado	1

Las columnas de la tabla B.122 contienen: las características, importancia (I), nivel de soporte (NS), nivel de cumplimiento (NC), el factor (F) y el puntaje total (Total). El puntaje total se calcula de la siguiente manera:

Puntaje total

$$Puntaje = I * NS * NC * Factor \quad (B.17)$$

Donde I es la importancia, NS el nivel de soporte, NC el nivel de cumplimiento y Factor un valor entre 0,1 y 0,9.

Tabla B.122 Características técnicas - Calculo del puntaje.

Característica	I	NS	NC	F	Total
Tecnología, Arquitectura e Interoperabilidad					
Compatibilidad con estándares	1	1	2	1	2
Base de datos	1	1	2	1	2
Sistema operativo del modelador	1	2	2	1	4
Navegador Web	1	2	2	1	4
Extensibilidad	1	2	1	1	2
Integración con Servicios Web	1	2	2	1	4
Diseño y modelado de procesos					
Interfaz gráfica para key users	1	2	2	1	4
Tipo de modelador	1	2	2	1	4
Notación gráfica de procesos compatible con BPMN 2.0	1	1	2	1	2
Gestión de formularios					
Definición de formularios	1	2	2	1	4
Formato de formularios	1	2	2	1	4
Elementos de formularios	1	2	2	1	4
Formularios dinámicos	1	2	2	1	4
Reglas de llenado de campos	1	2	2	1	4
Validación de datos del formulario	1	2	2	1	4
Motor de Workflow					
Iniciación de procesos en forma manual y automática	1	1	2	1	2
Definición de reglas de negocio	1	0	0	1	0
Asignación de tareas	1	2	2	1	4
Reasignación de tareas	1	2	2	1	4
Adjuntar documentos a tareas	1	2	2	1	4
Versionado de procesos	1	0	0	1	0
Tareas de borrador	1	2	2	1	4

Característica	I	NS	NC	F	Total
Gestión, monitoreo y auditoría					
Gestión de instancias por usuarios	1	1	2	1	2
Mantenimiento histórico de procesos	1	1	2	1	2
Monitoreo de procesos	1	0	0	1	0
Portal					
Presentación de las tareas	1	1	2	1	2

Por otro lado tenemos los cálculos de las características no funcionales. Las columnas de la tabla B.124 contienen: el peso de cada subcaracterística (P_i), el nivel de cumplimiento de la subcaracterística (s_i), el valor final de la subcaracterística definido en la ecuación B.18 y el nivel de cumplimiento (NC). El nivel de cumplimiento se obtiene de la siguiente manera:

- $vf < 1 \rightarrow$ Nivel de cumplimiento Bajo.
- $1 \leq vf < 3 \rightarrow$ Nivel de cumplimiento Medio.
- $3 \leq vf \rightarrow$ Nivel de cumplimiento Alto.

Valor final

$$vf = \frac{\sum_{i=1}^N (s_i * P_i)}{\sum_{i=1}^N P_i} \quad (B.18)$$

Siendo vf el 'valor final' de la característica y N la cantidad de subcaracterísticas.

En el caso de las características incluidas en la categoría Performance, hay que definir cuales son los valores de $L1$ y $L2$ para cada una de ellas, de esta manera, se podrá determinar el valor final de cada característica. Para determinar estos valores se consideraron estudios previos de performance así como de usabilidad [47] [42]. A continuación se muestra una tabla con los valores determinados:

Tabla B.123 Parámetros

Característica	L1	L2
Tiempo de respuesta	3 seg	10 seg
Throughput	30 %	80 %
Capacidad	20 %	70 %

Se considera que todas las subcaracterísticas son importantes y en este sentido se asigna el valor 1 al peso de cada subcaracterística.

Tabla B.124 Características técnicas - Cálculo del puntaje.

[illegible]

Anexo C

Caso de estudio

En este anexo se brinda una versión completa de los casos de estudio presentados en el Capítulo 5. La sección C.1 presenta el tema del Anexo. En la sección C.2 se especifica el caso de estudio. En la sección C.3 se realiza un análisis sobre los resultados del caso de estudio en cada herramienta. Finalmente en la sección C.4 se brinda un resumen del capítulo.

C.1. Introducción

En el presente anexo se especifica el caso de estudio realizado. Es importante marcar la diferencia entre los casos de prueba definidos y el caso de estudio. Los casos de prueba tienen el fin de homogeneizar la forma en que se evalúan las características (cada caso de prueba incluye un conjunto de características a evaluar), mientras que el caso de estudio se centra en realizar un caso común en las herramientas.

El caso de estudio presentado fue realizado con la intención de abarcar la mayor cantidad de construcciones que provee el estándar BPMN 2.0. Entre ellas, subprocesos, distintos tipos de tareas, eventos de tiempo, consumo de servicios, reglas de negocio, eventos de recepción y envío de mensajes. Además de los elementos que habitualmente forman parte de un proceso de negocio.

Para finalizar, se debe tener en cuenta que el proceso implementado está basado en un caso existente. Sin embargo, modela la realidad de forma diferente, puesto que el objetivo principal del caso de estudio es probar las funcionalidades mencionadas de un proceso de negocio.

C.2. Caso de Estudio

C.2.1. Solicitud de licencia vacacional

C.2.1.1. Modelos del procesos

Solicitud de licencia vacacional El proceso se inicia cuando un trabajador desea hacer una solicitud de vacaciones. Para esto el trabajador debe llenar una solicitud, ingresando cantidad de días y fecha de inicio de la licencia. Al completar la solicitud, la misma es enviada a la sección registro de la Facultad de Ingeniería para ingresar la solicitud del trabajador. Luego que la sección Registro de la Facultad de Ingeniería ingresa la solicitud, ésta pasa a la sección de RRHH dónde un administrativo verifica los días disponibles con que cuenta el trabajador. Una vez que se verifica la disponibilidad, la solicitud es enviada al Jefe de la unidad para autorizar la solicitud, y en base a esto se aprobará o desaprobará la solicitud. En caso que el trabajador tenga días acumulados o el jefe de la unidad lo haya autorizado, un administrativo del departamento de Recursos Humanos valida y analiza la solicitud. En caso que el Administrativo de RRHH no analice la solicitud antes de los 15 días, la misma pasa a ser analizada por el Jefe de área. En todos los casos se debe notificar al solicitante, en caso de aprobación se indica la cantidad de días y la fecha de inicio de las vacaciones, en caso de rechazo se informa los motivos del mismo.

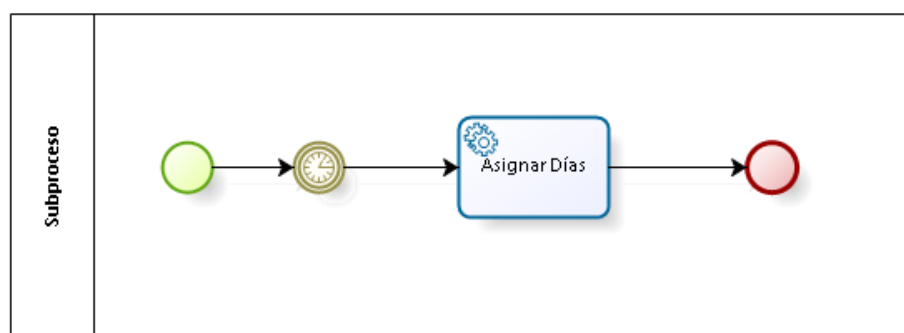


Figura C.1 Subproceso - Asignar Días

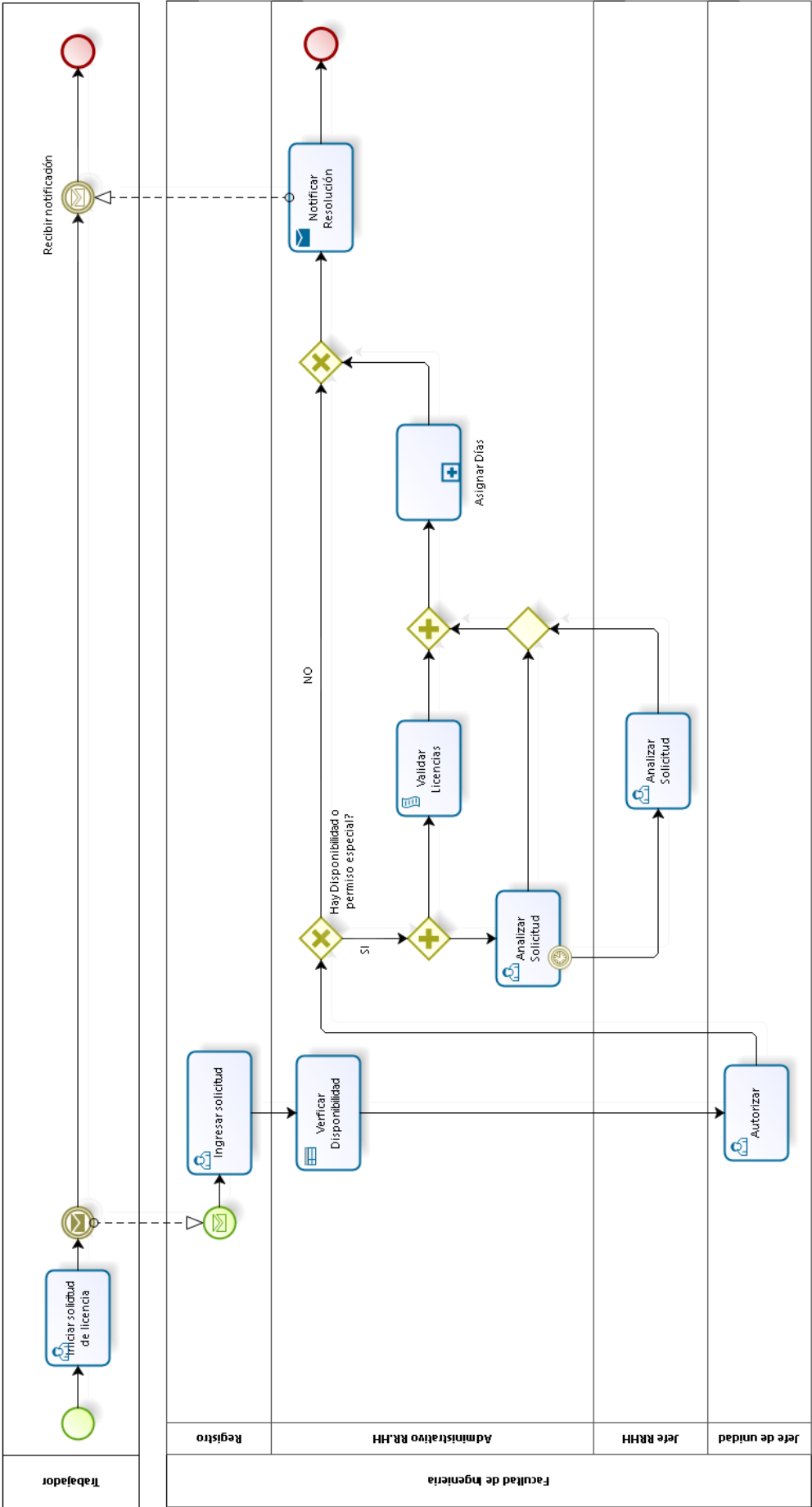


Figura C.2 Caso de estudio

C.2.1.2. Secuencia del proceso

1. Inicio: El caso de uso comienza cuando en la sección Registro se recibe una solicitud de licencia por parte de un Trabajador.
2. Ingresar solicitud: En esta actividad, un usuario de la sección Registro ingresa los siguientes campos en un formulario: cantidad de días, fecha de inicio.
3. Verificar disponibilidad: Un usuario con rol Administrativo RRHH realiza esta tarea, mediante una regla de negocio, verifica si la cantidad de días ingresados en la tarea *Ingresar solicitud* es menor o igual a los días disponibles que tiene el empleado que solicitó la licencia. Para obtener los días disponibles se realiza una consulta a la base de datos.
4. Autorizar: Un usuario con rol Jefe RRHH marca en un checkbox si autoriza o no la solicitud. En caso de rechazo debe escribir los motivos.
 - En caso de no haber disponibilidad o de no ser autorizada, se notifica el rechazo.
 - En caso de haber disponibilidad o ser autorizada, se realizan las tareas *Validar licencia* y *Analizar solicitud* en paralelo.
5. Validar licencias: Se ejecuta un *Script* y verifica que la cantidad de días solicitados sea menor o igual a la cantidad de días disponibles. Retorna el resultado en una variable.
6. Analizar solicitud: Esta tarea la realiza un usuario con rol Administrativo RRHH. Debe elegir en un radiobutton si aprueba o no la solicitud, en caso de rechazo, debe escribir en un campo de texto el motivo. Esta tarea se debe realizar en menos de 15 días (que se traducen en 15 segundos en el proceso), pasado este tiempo, se realiza un escalamiento reasignando la tarea a un usuario con rol Jefe RRHH.
7. Analizar solicitud: Esta tarea la realiza un usuario con rol Jefe RRHH. Debe elegir en un radiobutton si aprueba o no la solicitud, en caso de rechazo, debe escribir en un campo de texto el motivo.
8. Asignar días: Es un subproceso en el cual se espera una semana antes de asignar los días de licencia correspondientes. Una vez pasada la semana, se consulta mediante un servicio web, el email del empleado.
9. Notificar resolución: Esta tarea es realizada por un usuario con rol Administrativo RRHH, donde le envía al empleado la resolución de su solicitud. En caso de aprobación se le comunica la cantidad de días y la fecha de inicio de la licencia. Si la solicitud es rechazada se le envía al empleado el motivo del rechazo.

C.3. Análisis del Caso de Estudio

Para realizar el análisis del caso de estudio en cada herramienta, es necesario documentar particularidades de las mismas a medida que se implementa el caso. Es importante definir

C.3.1.2. Actividades

El caso de estudio requiere de la implementación de una gran variedad de tipo de tareas. La herramienta cuenta con tareas de: usuario, envío, recepción, manual, servicio, reglas de negocio, script y vacía. Además provee implementaciones específicas para algunas tareas de servicios, estas son: envío de email, web services, servicios REST y logs. Dado que las tareas que provee la herramienta se corresponden con las solicitadas en el caso de estudio, fue posible implementar todas las tareas del caso de estudio de manera correcta.

C.3.1.3. Compuertas

JBoss BPM Suite cuenta con compuertas exclusivas basadas en datos (XOR), basadas en eventos, paralelas e inclusivas. El caso de estudio requiere dos tipos de compuertas, compuertas paralelas y compuertas exclusivas. Para ingresar la condición que establece que camino seguirá el proceso se debe crear una expresión con resultado *booleano*. Esta expresión puede crearse de dos maneras. Una es seleccionando de un combobox una variable del proceso y de otro combobox una condición (es nulo, es falso, es verdadero). La otra forma es escribiendo un script en el cual se pueden utilizar más de una variable del proceso. Fue posible implementar las compuertas requeridas en el caso de estudio.

C.3.1.4. Comunicación entre *pools*

La herramienta permite la implementación de *Line* y *Sub-process*. Es posible crear subprocesos embebidos o reutilizables. Para la implementación del caso de estudio se utilizó un subproceso reutilizable. Para ello fue necesario crear un nuevo proceso y luego utilizar el identificador del proceso para asociarlo en la actividad del subproceso del caso de estudio.

No fue posible implementar la comunicación entre *pools* dado que la herramienta no cuenta con la definición de *pools*.

C.3.2. Bonita BPM

El software utilizado para la ejecución del caso de prueba fue:

- Navegador Mozilla versión 44.0.2.
- Sistema operativo Windows 10

C.3.2.1. Eventos

Bonita permite modelar eventos de inicio, intermedios y de fin con envío y recepción de mensajes en ellos. Gracias a estos se logró implementar todos los eventos requeridos en el caso de estudio.

C.3.2.2. Actividades

Bonita permita implementar 6 tipo de tareas: de servicio, humana, de script, recibir mensaje, enviar mensaje y abstracta. Para la tarea de regla de negocio se hizo una expresión Groovy. De esta manera se pudo implementar todas las tareas del caso de estudio.

C.3.2.3. Compuertas

Para la construcción del caso de estudio son necesarias 2 tipos de compuertas, paralelas y exclusivas. Bonita cuenta con estas 2 y además provee la compuerta inclusiva. En el caso de la compuerta exclusiva requiere de una condición para poder ejecutarla. Dado esto fue posible implementar sin problemas las compuertas del caso de estudio.

C.3.2.4. Comunicación entre *pools*

La comunicación entre pools se diseñó mediante tareas de mensajes y evento de recepción. En la ejecución se presentaron errores entonces se decidió ejecutar un solo pool, modificando el diseño original.

C.3.3. IntalioIbpms

Para implementar el caso de estudio en esta herramienta se utilizaron las siguientes consideraciones:

- Una base de datos externa. Se utilizó PostgreSQL, versión 9.4. La base de datos tenía la tabla *Empleado*, la misma contenía información sobre los empleados (nombre, email y cantidad de días disponibles).
- Se utilizó el navegador Chrome, versión 49.

C.3.3.1. Eventos

Los eventos aceptados por el modelador son eventos de inicio y fin, eventos de *throw* y *catch*. Sin embargo, no acepta tipos de eventos *boundary*, es decir, eventos que se adjuntan a las tareas. Para implementar este tipo de eventos se recomienda incluir un subproceso, porque estos aceptan eventos *boundary*. Para cumplir con el funcionamiento esperado del caso de estudio, se cambió la tarea “Analizar solicitud” por un subproceso que contiene un *time* y dentro un proceso. Este proceso tiene un evento de inicio vacío, una tarea de usuario y un evento de fin. La figura C.4 muestra el cambio realizado en el modelo para cumplir con el funcionamiento esperado del caso de estudio.

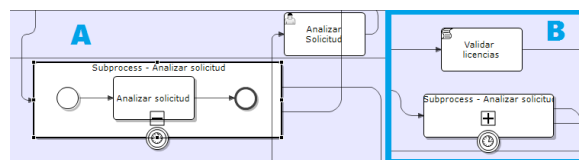


Figura C.4 Eventos boundary en subprocesos

En definitiva, fue posible implementar el comportamiento de los eventos requeridos por el caso de estudio. A pesar de los cambios mencionados anteriormente, se logró mantener el comportamiento esperando.

C.3.3.2. Actividades

El caso de estudio dado en la sección C.2 propone evaluar diferentes tipos de actividades. Tanto el modelador como el motor de Intalio soportan los tipos de actividades que muestra la figura C.5. Respecto a la actividad de tipo *Business Rule Task*, Intalio utiliza tablas de decisión, se creó una nueva, la cual se asoció a la tarea. Este tipo de tareas solo está disponible en la versión *Enterprise*. La *Script Task* solo requiere definir el script dentro del *Mapper* (el *Mapper* es la herramienta que permite asociar variables a los parámetros de entrada y salida del formularios asociado a una tarea).

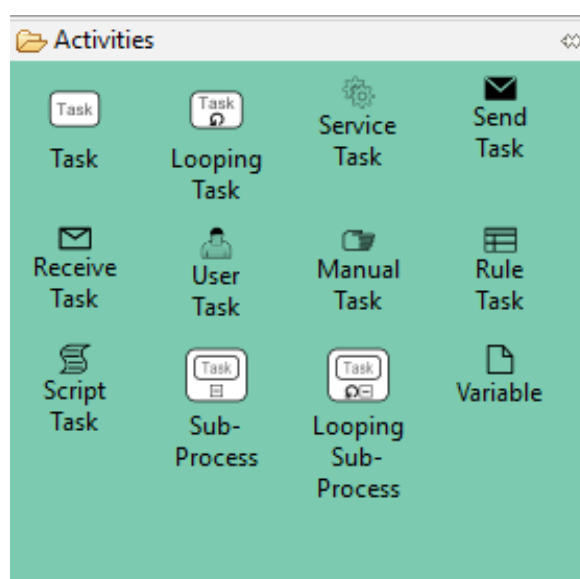


Figura C.5 Tipos de tareas

C.3.3.3. Compuertas

El modelado de compuertas en Intalio es muy sencillo. Los *Gateway* de apertura, permiten indicar cual es el flujo por defecto. De la misma manera, en cada flujo de la secuencia se debe definir cual es la condición que se debe cumplir para que el proceso prosiga por dicho flujo. Las condiciones se definen en el *Mapper*.

C.3.3.4. Comunicación entre *pools*

Intalio provee dentro de la paleta de objetos que se pueden incluir en el modelo, *Pool*, *Line* y *Sub-process*. Es posible incluir más de un *pool* en cada modelo, cada *pool* tiene la opción de ser ejecutable. En caso de serlo, luego del *deploy*, el proceso que contiene el evento de inicio vacío queda disponible para ser ejecutado. Esto permitió definir tres *pool* dentro del modelo, marcando como ejecutable el proceso que corresponde al trabajador y a la organización. El tercer *pool* no es ejecutable ya que contiene los formularios, la tabla de decisión (contiene la regla de negocio) y el llamado al servicio web que están asociados a

alguna tarea.

Los eventos entre los procesos, son sencillos de implementar. La figura C.6 muestra el modelo con la comunicación entre los *pool*. Una vez establecida la comunicación, fue necesario asociar las variables de entrada y salida dentro del *Mapper*.

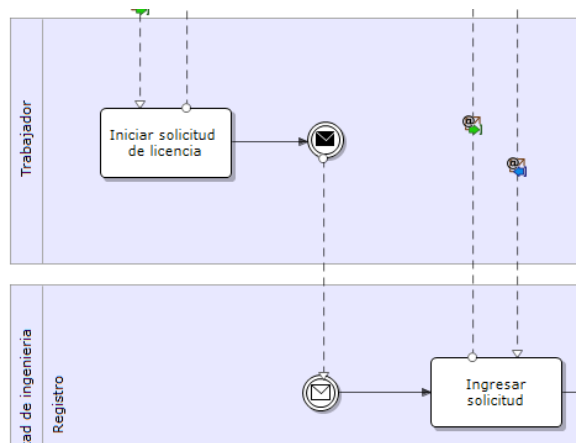


Figura C.6 Comunicación entre *pools*

Finalmente, se implementó un subproceso. Para ello, el modelador permite insertar un subproceso dentro del modelo, dentro del cual se inserta el proceso. El modelador permite minimizar el proceso, facilitando la legibilidad del mismo, ver figura C.7

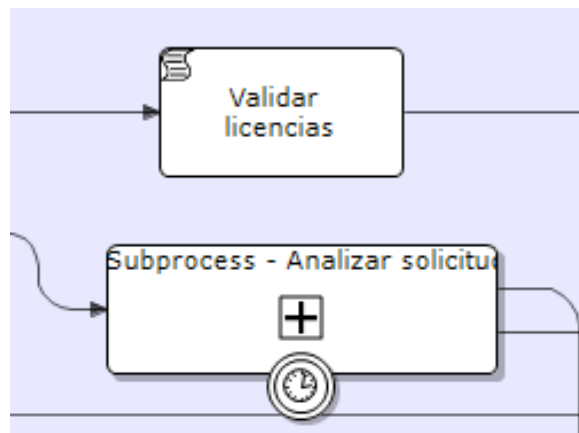


Figura C.7 Tipos de tareas

C.3.4. Activiti

El software de base utilizado para la implementación del caso de estudio fue:

- Base de dato MySQL, versión 5.7.10. En la misma se encuentra la base de datos de Activiti y la base de datos utilizada en el proceso. Esta última contiene la tabla *Empleado* con nombre, email y cantidad de días disponibles.
- Se utilizó el navegador Chrome, versión 49.
- El proceso fue ejecutado en el Sistema Operativo Windows 10.
- Para la implementación de las clases java que fueron necesarias para cubrir algunos aspectos no soportados nativamente por la herramienta, se utilizó Eclipse Mars2.

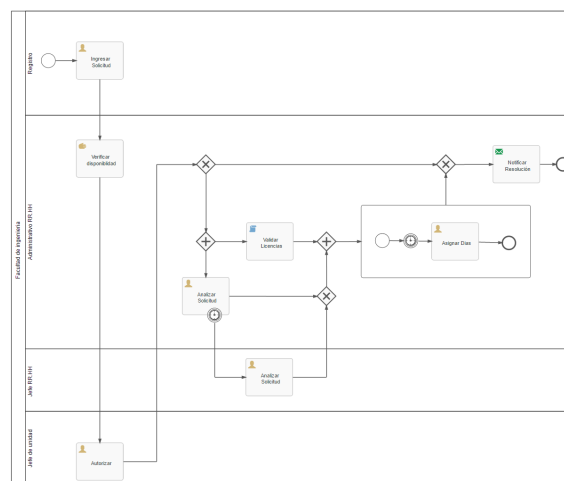


Figura C.8 Modelo del caso de estudio en Activiti

C.3.4.1. Eventos

En el caso de los eventos, Activiti permite modelar eventos de inicio, de fin, de *throw*, de *catch* y eventos *boundary*. Esto implica que fue posible modelar todos los eventos requeridos en el caso de estudio de manera satisfactoria. La figura C.9 muestra en detalle todos los tipos de eventos de inicio y fin, mientras que la figura C.10 muestra los otros tipos de eventos que soporta la herramienta.

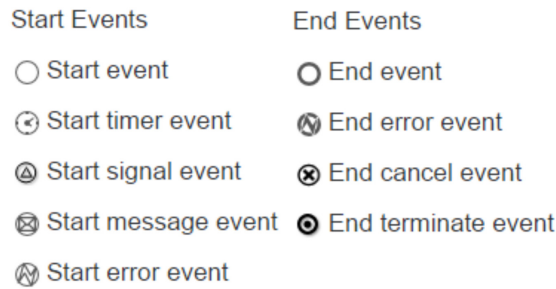


Figura C.9 Eventos de inicio y fin de Activiti

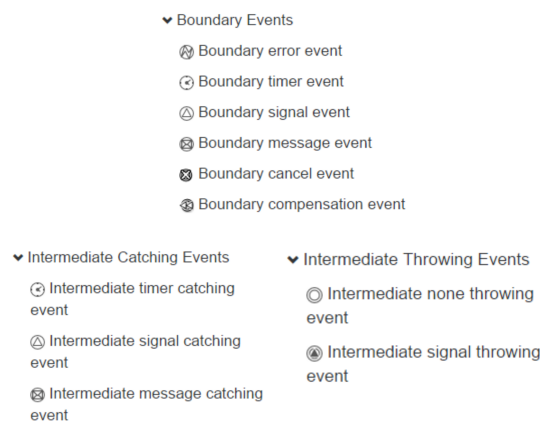


Figura C.10 Eventos Throw, Catch, Boundary de Activiti

C.3.4.2. Actividades

Activiti permite modelar una amplia gama de actividades pero no todas pueden ser ejecutadas. De las actividades presentes en el caso de estudio, es posible ejecutar: tareas de usuario, de *email* y de *script*. Las tareas de servicio y de regla de negocio no pudieron ser utilizadas de forma nativa, sino que fue necesario la implementación de *TaskListener* que realizaban las funcionalidades que deberían ser realizadas por las tareas antes mencionadas. En resumen, Activiti no permite todas las actividades requeridas en el caso de estudio, pero si se realizan particularizaciones es posible ejecutar el caso de estudio en su totalidad.

C.3.4.3. Compuertas

El caso de estudio requería de dos tipos de compuertas, compuertas paralelas y compuertas exclusivas. Para el caso de las compuertas exclusivas es necesario definir la condición que define cuales de los dos flujos será ejecutado. Para definir las condiciones se pueden utilizar variables de procesos y se usan conectores *booleanos* para indicar que se deben cumplir más de una condición. Fue posible utilizar los dos tipos de compuertas necesarias para ejecutar el caso de estudio.

C.3.4.4. Comunicación entre *pools*

La herramienta permite la implementación de *Pool*, *Line* y *Sub-process*. Se implemento un subproceso. La única manera de implementarlo en Activiti es insertando un subproceso dentro del modelo, y dentro de este insertar el proceso. Para el caso de la comunicación entre *pools* es posible modelar la misma pero no es posible ejecutar el proceso. Según la sección 13.2 de [39] la herramienta solo permite la definición de un solo *pool* por proceso de otra manera aparece un error al momento del *deploy*.

C.3.5. Bizagi Suite

Para implementar el caso de estudio en esta herramienta se utilizó:

- Base de datos SQLServer, versión 2012. La base de datos tenía la tabla *Empleado*, la misma contenía información sobre los empleados (nombre, email y cantidad de días disponibles).
- Se utilizó el navegador Chrome, versión 49.

C.3.5.1. Eventos

Bizagi permite definir eventos de inicio, fin e intermedio, además cuenta con varios tipos de eventos tales como: *Timer*, *Message*, *Signal*, *Compensate*, *Escalation*, *Conditional*, *Paralell Multiple*, *Multiple*.

Dado esto se pudo definir de manera satisfactoria lo que requiere el caso de estudio.

C.3.5.2. Actividades

Bizagi no provee la capacidad de captar errores de conexión, por ejemplo en la actividad que consume un servicio web, si al ejecutar esta tarea el usuario no tiene acceso a Internet se debería captar un error, pero éste es ignorado. De todas formas, vía particularización se puede tratar el error.

C.3.5.3. Compuertas

En este caso particular se requiere implementar dos tipo de compuertas, según una condición y paralelas. Las compuertas pueden implementarse sin problemas.

C.3.5.4. Comunicación entre *pools*

Bizagi permite la conexión entre *pools* en el modelado y en ejecución. La ejecución de esta interacción tiene una complejidad importante por lo que se decidió modificar el proceso para la ejecución de un solo *pool*.

C.3.6. Camunda BPMN

Para implementar el caso de estudio en esta herramienta se utilizaron las siguiente consideraciones:

- Una base de datos externas. Se utilizó PostgreSQL, versión 9.4. La base de datos tenía la tabla *Empleado*, la misma contenía información sobre los empleados (nombre, email y cantidad de días disponibles).

- Se utilizó el navegador Chrome, versión 49.

C.3.6.1. Eventos

Los eventos aceptados por el modelador son eventos de inicio y fin, eventos de *throw* y *catch*, también soporta tipos de eventos *boundary*, es decir, eventos que se adjuntan a las tareas. La figura C.11 muestra la lista de la definición de eventos que provee el modelador, esto permite que, un evento de inicio, puede estar vacío o recibir un mensaje.

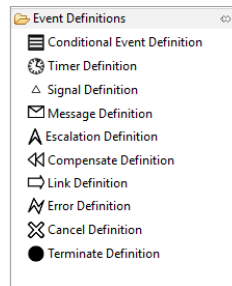


Figura C.11 Tipos de eventos

En definitiva, fue posible implementar todos los eventos requeridos por el caso de estudios, descrito en la sección C.2.

C.3.6.2. Actividades

El caso de estudio dado en la sección C.2 propone evaluar diferentes tipos de actividades. Tanto el modelador como el motor de Camunda soportan los siguientes tipos de actividades que muestra la figura C.12. Con respecto a las actividades de tipo *Service Task*, *Business Rule Task*, fue necesario desarrollar clases java. Ésta última además necesitó la integración con el motor de reglas de negocio Drools. Mientras que la *Script Task* solo requería el desarrollo de un script dentro de las propiedades, ver [34].

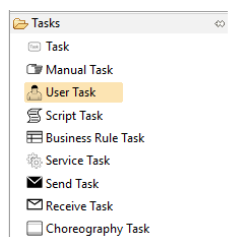


Figura C.12 Tipos de tareas

En síntesis, Camunda permite utilizar los tipos de actividades requeridos por el caso de estudio antes mencionado.

C.3.6.3. Compuertas

El modelado de compuertas en Camunda es muy sencillo. Los *Parallel Gateway* de apertura, permite indicar cual es el flujo por defecto. De la misma manera, en cada flujo de

la secuencia se debe definir cual es la condición que se debe cumplir para que el proceso prosiga por dicho flujo.

C.3.6.4. Comunicación entre *pools*

La herramienta provee dentro de la paleta de objetos que se pueden incluir en el modelo, *Pool*, *Line* y *Sub-process*, entre otros. Cada *pool* que se incluye en el modelo tiene la opción de ser ejecutable. Si es ejecutable, luego del *deploy*, el proceso que está dentro del *pool* queda disponible para ser ejecutado. Esto permitió definir dos *pool* dentro del modelo, marcando solamente como ejecutable el proceso que corresponde al trabajador.

Con respecto a los eventos entre los procesos, fue necesario implementar clases java para establecer la correlación entre los procesos, ver [34].

Finalmente, se implemento un subprocesso. Para ello, el modelador permite insertar un subprocesso dentro del modelo, dentro del cual se inserta el proceso. El modelador permite minimizar el proceso, facilitando la legibilidad del modelo, ver figura C.13

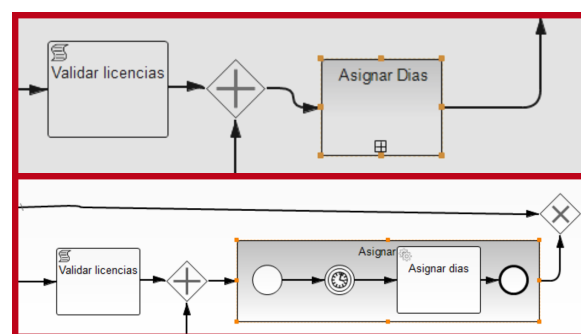


Figura C.13 Subprocesos

C.3.7. Orchestra

El software utilizado para la ejecución del caso de prueba fue:

- Navegador Chrome versión 49.
- Sistema operativo Windows 10.
- Base de datos PostgreSQL versión 9.2

C.3.7.1. Eventos

Orchestra permite eventos de inicio y de fin, de envío y recepción de mensajes y un evento de *timer*.

C.3.7.2. Actividades

Orchestra permite implementar 2 tipos de tareas: de script y humana. Por lo tanto no se pudieron ejecutar las tareas de regla de negocio ni la de servicio, ni la parte de escalarización. Tampoco permite ejecutar subprocesos.

C.3.7.3. Compuertas

Para la construcción del caso de estudio son necesarias 2 tipos de compuertas, paralelas y exclusivas. Orchestra cuenta con estas 2 por lo que en este sentido no hubo problemas para su construcción.

C.3.7.4. Comunicación entre *pools*

Orchestra no permite ni el diseño ni la ejecución de más de un *pool*. Por lo tanto no fue posible ejecutar 2 *pools* en el caso de estudio.

C.3.8. ProcessMaker

Para implementar el caso de estudio en esta herramienta se utilizaron las siguientes consideraciones:

- Una base de datos externa. Se utilizó PostgreSQL, versión 9.4. La base de datos tenía la tabla *Empleado*, la misma contenía información sobre los empleados (nombre, email y cantidad de días disponibles).
- Se utilizó el navegador Chrome, versión 49.

C.3.8.1. Eventos

El modelador permite incluir en el proceso eventos intermedios de envío de mensaje, pero no de recepción. También permite agregar un evento de inicio de recepción de mensaje y un evento de fin de envío de mensaje. Este tipo de eventos requieren de un *Message Type*, el cual se define dentro del modelo y queda disponible para todo el proceso, ver [37]. Sin embargo, el motor provisto para la versión *Community* no reconoce estos eventos. Por lo tanto, el caso de estudio para esta herramienta fue modificado, como muestra la figura C.14 eliminando los eventos de mensajes, además se modificó el inicio del proceso, dejando solamente el *pool* correspondiente a la organización, cambiando el evento de inicio con recepción de mensaje por un inicio vacío.

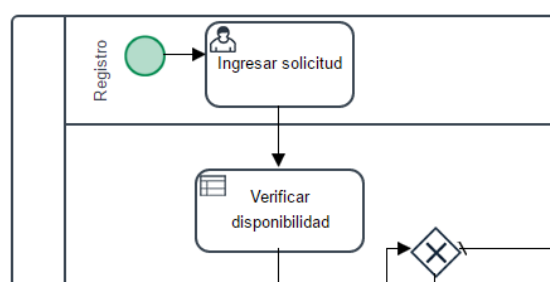


Figura C.14 Evento de inicio - Caso de estudio

Se presentó la misma situación al ejecutar el subproceso *Asignar Días*. El subproceso presentaba un evento *Timer Intermediate*, el cual provee el modelador, pero el motor no reconoce. Se procedió a modificar el subproceso como se observa en la figura C.15, eliminando el evento de *timer* y agregando en la tarea un *trigger* para indicar la duración de la tarea.

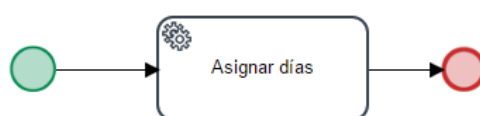


Figura C.15 Subproceso - Caso de estudio

C.3.8.2. Actividades

La versión *Community* de ProcessMaker provee un motor que no es capaz de distinguir los distintos tipos de actividades, tampoco provee un motor de reglas de negocio, por lo tanto no fue posible implementar la actividad *Verificar disponibilidad*. A pesar de esto, es posible simular el comportamiento esperado del proceso con los elementos provistos por el modelador.

Las actividades *Verificar disponibilidad* (regla de negocio), *Validar licencias* (script) y *Asignar días* (servicio) se implementaron a través de un *trigger*, el mismo permite realizar conexiones a servicios web, a bases de datos externas o realizar cálculos con variables del proceso.

C.3.8.3. Compuertas

El caso de estudio requería implementar dos tipo de compuertas, según una condición y paralelas. Las compuertas pueden implementarse sin problemas, teniendo en cuenta que luego de una compuerta paralela convergente debe haber una actividad.

C.3.8.4. Comunicación entre *pools*

Es posible modelar la comunicación entre *pools*. La herramienta provee un ejemplo, donde explica los pasos a seguir para lograr la comunicación, ver [37]. Si bien fue posible modelar la comunicación, no fue posible ejecutar el proceso por lo visto en la sección C.3.8.1

C.4. Resumen

Se brindó la definición del caso de estudio. En este sentido, se mostraron los modelos correspondientes al caso junto con la secuencia del proceso, donde se describió cada tarea.

Luego, se presentaron los análisis del caso de estudio para cada herramienta. En este sentido, para cada herramienta previo al análisis se brindaron consideraciones a tomar en cuenta, ya que no fue posible implementar todo el caso de estudio en todas las herramientas.

Anexo D

Proyecto web

D.1. Casos de uso

En esta sección se describen los casos de uso de la aplicación web.

Los casos de uso son una forma de definir las funcionalidades de un sistema.

No todos los usuarios pueden realizar todas las acciones, en este sentido se definen tres roles:

Administrador: es el usuario que administra el sistema. Un usuario con rol “Administrador” puede manejar usuarios, roles, categorías, características, subcaracterísticas, herramientas, versiones de herramientas. También puede ingresar resultados, realizar evaluaciones y comparaciones de herramientas.

Evaluator: es el usuario encargado de realizar las evaluaciones. Un usuario con este rol puede ingresar resultados. Además es capaz de manejar herramientas y versiones.

Cliente: este tipo de usuarios solamente pueden realizar comparaciones entre dos herramientas o ponderar los criterios de evaluación para obtener las herramientas que se ajusten más a dichas ponderaciones.

Nombre: Alta de categoría
Actores: Administrador
Descripción Registrar una nueva categoría en la base de datos del sistema.
Precondiciones: No hay.
Flujo principal: 1. El caso de uso se inicia cuando un usuario desea registrar una nueva categoría. 2. El usuario ingresa nombre de la categoría e indica si la misma es funcional o no funcional. 3. El sistema valida los datos ingresados. 4. Se crea la nueva categoría.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Baja de categoría
Actores: Administrador
Descripción: Eliminar una categoría existente en la base de datos del sistema.
Precondiciones: Existe al menos una categoría existente en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el administrador desea eliminar una categoría. 2. El usuario selecciona la categoría y realiza clic en el botón eliminar. 3. Se elimina la categoría del sistema.

Nombre: Modificación de categoría
Actores: Administrador
Descripción: Modificar los datos de una categoría en la base de datos del sistema.
Precondiciones: Debe existir al menos una categoría creada en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el usuario desea modificar una nueva categoría. 2. El usuario se posiciona sobre la categoría y realiza clic en el botón modificar. 3. El sistema muestra la pantalla de modificación de la categoría. 4. El usuario modifica el nombre o el tipo de la categoría. 5. El sistema valida los datos ingresados. 6. Se modifica la nueva categoría.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Listado de Categorías
Actores: Administrador
Descripción: Listar todas las categorías existentes en la base de datos del sistema.
Precondiciones: No hay.
Flujo principal: 1. El caso de uso se inicia cuando el administrador estando en el menú, realiza clic en la opción Categorías. 2. El sistema muestra la pantalla de listado de categorías mostrando el nombre de las categorías existentes.

Nombre: Alta de característica
Actores: Administrador
Descripción Registrar una nueva característica en la base de datos del sistema.
Precondiciones: Debe existir al menos una categoría en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando un usuario desea registrar una nueva característica. 2. El usuario ingresa nombre de la característica, selecciona la categoría a la que pertenece e ingresa una descripción de la característica. 3. El sistema valida los datos ingresados. 4. Se crea la nueva característica.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Baja de característica
Actores: Administrador
Descripción: Eliminar una característica existente en la base de datos del sistema.
Precondiciones: Existe al menos una característica en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el usuario desean eliminar una característica. 2. El usuario selecciona la característica y realiza clic en el botón eliminar. 3. Se elimina la característica del sistema.

Nombre: Modificación de característica
Actores: Administrador
Descripción: Modificar los datos de una característica en la base de datos del sistema.
Precondiciones: Debe existir al menos una característica creada en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el usuario desea modificar una característica. 2. El usuario se posiciona sobre la característica y realiza clic en el botón modificar. 3. El sistema muestra la pantalla de modificación de la característica. 4. El usuario modifica el nombre, la categoría o la descripción. 5. El sistema valida los datos ingresados. 6. Se modifica la característica.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Listado de características
Actores: Administrador
Descripción: Listar todas las características existentes en la base de datos del sistema.
Precondiciones: No hay.
Flujo principal: 1. El caso de uso se inicia cuando el usuario estando en el menú, realiza clic en la opción Característica. 2. El sistema muestra la pantalla de listado de características mostrando el nombre de las mismas y organizándolas por categorías.

Nombre: Alta de subcaracterística
Actores: Administrador
Descripción Registrar una nueva subcaracterística en la base de datos del sistema.
Precondiciones: Debe existir al menos una característica en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando un usuario desea registrar una nueva subcaracterística. 2. El usuario ingresa nombre de la subcaracterística, una pregunta para la subcaracterística, selecciona la característica a la que pertenece e ingresa una descripción de la subcaracterística. 3. El sistema valida los datos ingresados. 4. Se crea la nueva subcaracterística.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Baja de subcaracterística
Actores: Administrador
Descripción: Eliminar una subcaracterística existente en la base de datos del sistema.
Precondiciones: Existe al menos una subcaracterística en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el usuario desea eliminar una subcaracterística. 2. El usuario selecciona la subcaracterística y realiza clic en el botón eliminar. 3. Se elimina la subcaracterística del sistema.

Nombre: Modificación de subcaracterística
Actores: Administrador
Descripción: Modificar los datos de una subcaracterística en la base de datos del sistema.
Precondiciones: Debe existir al menos una subcaracterística creada en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el usuario desea modificar una subcaracterística. 2. El usuario se posiciona sobre la subcaracterística y realiza clic en el botón modificar. 3. El sistema muestra la pantalla de modificación de la subcaracterística. 4. El usuario modifica el nombre, la pregunta, la característica o la descripción. 5. El sistema valida los datos ingresados. 6. Se modifica la subcaracterística.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Listado de subcaracterísticas
Actores: Administrador
Descripción: Listar todas las subcaracterísticas existentes en la base de datos del sistema.
Precondiciones: No hay.
Flujo principal: 1. El caso de uso se inicia cuando el usuario estando en el menú, realiza clic en la opción Subcaracterística. 2. El sistema muestra la pantalla de listado de subcaracterísticas mostrando el nombre de las mismas y organizándolas por categoría y característica.

Nombre: Alta de herramienta
Actores: Administrador, Evaluador
Descripción Registrar una nueva herramienta en la base de datos del sistema.
Precondiciones: No hay.
Flujo principal: 1. El caso de uso se inicia cuando un usuario desea registrar una nueva herramienta. 2. El usuario ingresa nombre de la herramienta. 3. El sistema valida los datos ingresados. 4. Se crea la nueva herramienta.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Baja de herramienta
Actores: Administrador, Evaluador
Descripción: Eliminar una herramienta existente en la base de datos del sistema.
Precondiciones: Existe al menos una herramienta en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el usuario desean eliminar una herramienta. 2. El usuario selecciona la herramienta y realiza clic en el botón eliminar. 3. Se elimina la herramienta y todas sus versiones del sistema.

Nombre: Modificación de herramienta
Actores: Administrador, Evaluador
Descripción: Modificar los datos de una herramienta en la base de datos del sistema.
Precondiciones: Debe existir al menos una herramienta creada en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el usuario desea modificar una herramienta. 2. El usuario se posiciona sobre la herramienta y realiza clic en el botón modificar. 3. El sistema muestra la pantalla de modificación de herramienta. 4. El usuario modifica el nombre. 5. El sistema valida los datos ingresados. 6. Se modifica la herramienta.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Listado de herramientas
Actores: Administrador, Evaluador
Descripción: Listar todas las herramientas existentes en la base de datos del sistema.
Precondiciones: No hay.
Flujo principal: 1. El caso de uso se inicia cuando el usuario estando en el menú, realiza clic en la opción Herramientas. 2. El sistema muestra la pantalla de listado de herramientas mostrando el nombre de las mismas y en caso de tener, muestra las versiones de la misma.

Nombre: Alta de versión de una herramienta
Actores: Administrador, Evaluador
Descripción Registrar una nueva versión de una herramienta en la base de datos del sistema.
Precondiciones: Existe al menos una herramienta en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando un usuario desea registrar una nueva versión para una herramienta. 2. El usuario ingresa número de la versión y la fecha. 3. El sistema valida los datos ingresados. 4. Se crea una nueva versión para la herramienta seleccionada.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Baja de versión de herramienta
Actores: Administrador, Evaluador
Descripción: Eliminar una versión de una herramienta existente en la base de datos del sistema.
Precondiciones: Existe al menos una herramienta con una versión asociada en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el usuario desean eliminar una versión de una herramienta. 2. El usuario selecciona la herramienta y la versión y realiza clic en el botón eliminar. 3. Se elimina la versión de la herramienta.

Nombre: Modificación de versión de herramienta
Actores: Administrador, Evaluador
Descripción: Modificar los datos de una versión en la base de datos del sistema.
Precondiciones: Debe existir al menos una herramienta con una versión asociada creada en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el usuario desea modificar una versión de una herramienta. 2. El usuario se posiciona sobre la versión de una herramienta y realiza clic en el botón modificar. 3. El sistema muestra la pantalla de modificación de versión. 4. El usuario modifica el número o la fecha. 5. El sistema valida los datos ingresados. 6. Se modifica la versión de la herramienta.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Listado de versiones de herramientas
Actores: Administrador, Evaluador
Descripción: Listar todas las versiones de las herramientas existentes en la base de datos del sistema.
Precondiciones: No hay.
Flujo principal: 1. El caso de uso se inicia cuando el usuario estando en el menú, realiza clic en la opción Herramientas y luego en la opción Versiones. 2. El sistema muestra la pantalla de listado de versiones de herramientas mostrando todas las herramientas del sistema junto con sus versiones.

Nombre: Alta de usuarios
Actores: Administrador, Evaluador, Cliente
Descripción Registrar un nuevo usuario en el sistema.
Precondiciones: No hay.
Flujo principal: 1. El caso de uso se inicia cuando se desea registrar un nuevo usuario en el sistema. 2. Se ingresa nombre, apellido, email y contraseña. 3. El sistema valida los datos ingresados. 4. Se crea el nuevo usuario en el sistema.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Baja de usuario
Actores: Administrador
Descripción: Eliminar un usuario existente en el sistema.
Precondiciones: Existe al menos un usuario creado en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el administrador desea eliminar un usuario. 2. El administrador selecciona el usuario a eliminar. 3. Se elimina el usuario de la herramienta.

Nombre: Modificación de usuario
Actores: Administrador
Descripción: Modificar los datos de un usuario en la base del sistema.
Precondiciones: Debe existir al menos un usuario en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el administrador desea modificar los datos de un usuario. 2. El usuario administrador se posiciona sobre el usuario que desea modificar y realiza clic en el botón modificar. 3. El sistema muestra la pantalla de modificación de usuario. 4. El usuario modifica el nombre o apellido. 5. El sistema valida los datos ingresados. 6. Se modifica el usuario seleccionado.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Listado de usuarios
Actores: Administrador
Descripción: Listar todos los usuarios de la base de datos del sistema.
Precondiciones: No hay.
Flujo principal: 1. El caso de uso se inicia cuando el usuario administrador estando en el menú, realiza clic en la opción Usuarios. 2. El sistema muestra la pantalla de listado de usuarios.

Nombre: Alta de roles
Actores: Administrador
Descripción Registrar un nuevo rol en el sistema.
Precondiciones: No hay.
Flujo principal: 1. El caso de uso se inicia cuando se desea crear un nuevo rol en el sistema. 2. Se ingresa nombre. 3. El sistema valida los datos ingresados. 4. Se crea el nuevo rol en el sistema.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Baja de rol
Actores: Administrador
Descripción: Eliminar un rol existente en el sistema.
Precondiciones: Existe al menos un rol creado en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el administrador desea eliminar un rol. 2. El administrador selecciona el rol a eliminar. 3. Se elimina el rol de la herramienta.

Nombre: Modificación de rol
Actores: Administrador
Descripción: Modificar los datos de un rol en la base del sistema.
Precondiciones: Debe existir al menos un rol en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el administrador desea modificar los datos de un rol. 2. El usuario administrador se posiciona sobre el rol que desea modificar y realiza clic en el botón modificar. 3. El sistema muestra la pantalla de modificación de rol. 4. El usuario modifica el nombre. 5. El sistema valida los datos ingresados. 6. Se modifica el rol seleccionado.
Flujo alternativo: 3. 1 Los datos ingresados son incorrectos. 2 El sistema indica un error. 3 Finaliza el caso de uso.

Nombre: Listado de roles
Actores: Administrador
Descripción: Listar todos los roles de la base de datos del sistema.
Precondiciones: No hay.
Flujo principal: 1. El caso de uso se inicia cuando el usuario administrador estando en el menú, realiza clic en la opción Roles. 2. El sistema muestra la pantalla de listado de roles.

Nombre: Alta de resultados
Actores: Administrador, Evaluador
Descripción Registrar un resultado de una evaluación en el sistema.
Precondiciones: Debe existir al menos una herramienta con una versión en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando se desea registrar un nuevo resultado en el sistema. 2. Se selecciona la versión de la herramienta y para cada subcaracterística se debe indicar, en caso de pertenecer al módulo Técnico-Funcional se debe ingresar el nivel de cumplimiento, nivel de soporte y factor. En caso de pertenecer al módulo Técnico-No Funcional se debe seleccionar el nivel de cumplimiento. 3. Se crea el nuevo resultado en el sistema.

Nombre: Baja de resultado
Actores: Administrador, Evaluador
Descripción: Eliminar un resultado existente en el sistema.
Precondiciones: Existe al menos un resultado creado en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el usuario desea eliminar un resultado. 2. El usuario selecciona el resultado a eliminar. 3. Se elimina el resultado de la versión de la herramienta.

Nombre: Modificación de resultado
Actores: Administrador, Evaluador
Descripción: Modificar los datos de un resultado en la base del sistema.
Precondiciones: Debe existir al menos un resultado en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el usuario desea modificar los datos de un resultado. 2. El usuario se posiciona sobre el resultado que desea modificar y realiza clic en el botón modificar. 3. El sistema muestra la pantalla de modificación de resultados. 4. El usuario modifica los valores seleccionados para las subcaracterística. 5. Se modifica el resultado seleccionado.

Nombre: Listado de resultados
Actores: Administrador, Evaluador
Descripción: Listar todos los resultados de la base de datos del sistema.
Precondiciones: No hay.
Flujo principal: 1. El caso de uso se inicia cuando el usuario estando en el menú, realiza clic en la opción Resultados. 2. El sistema muestra la pantalla de listado de resultados.

Nombre: Alta de evaluación
Actores: Administrador, Evaluador, Cliente
Descripción Registrar una nueva evaluación de un conjunto de herramientas en el sistema.
Precondiciones: Debe existir al menos una versión de herramienta con un resultado en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando se desea crear una nueva evaluación en el sistema. 2. Se ingresa el nombre de la evaluación, el umbral mínimo y máximo y las versiones de herramientas que desea evaluar. 3. El usuario indica la importancia para cada subcaracterística. 4. El sistema muestra el resultado de la evaluación para cada una de las herramientas seleccionadas. 5. Se crea la nueva evaluación en el sistema.

Nombre: Baja de evaluación
Actores: Administrador, Evaluador, Cliente
Descripción: Eliminar una evaluación existente en el sistema.
Precondiciones: Existe al menos una evaluación creada en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el usuario desea eliminar una evaluación. 2. El usuario selecciona la evaluación a eliminar. 3. Se elimina la evaluación del sistema.

Nombre: Modificación de evaluación
Actores: Administrador, Evaluador, Cliente
Descripción: Modificar el nombre de una evaluación en la base del sistema.
Precondiciones: Debe existir al menos una evaluación en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando el usuario desea modificar el nombre de una evaluación. 2. El usuario se posiciona sobre la evaluación que desea modificar y realiza clic en el botón modificar. 3. El sistema muestra la pantalla de modificación de evaluación. 4. El usuario modifica el nombre de la evaluación. 5. Se modifica la evaluación seleccionada.

Nombre: Listado de evaluaciones
Actores: Administrador, Evaluador, Cliente
Descripción: Listar todos las evaluaciones de la base de datos.
Precondiciones: No hay.
Flujo principal: 1. El caso de uso se inicia cuando el usuario estando en el menú, realiza clic en la opción Evaluaciones. 2. El sistema muestra la pantalla de listado de evaluaciones.

Nombre: Comparación de herramientas
Actores: Administrador, Evaluador, Cliente
Descripción Comparar los valores de los resultados ingresados entre dos versiones de herramientas.
Precondiciones: Deben existir al menos dos resultados en el sistema.
Flujo principal: 1. El caso de uso se inicia cuando se desea comparar dos resultados de versiones de herramientas. 2. Se seleccionan las versiones de las herramientas. 3. El sistema muestra los resultado ingresados en cada subcaracterística de las versiones de las herramientas seleccionadas.