

Informe Final – Proyecto BioFlows



Tutores:

Dr. Ing. Raúl Ruggia Frick
Ing. María Laura González
López

Estudiantes:

Pablo Moleri Riva-Zucchelli
Ramiro Alejandro Patiño
Deambrosio
Pablo Martin Ranz Pino

Resumen

Este proyecto tuvo como principal objetivo el desarrollo de una solución que permita incorporar flujos de ejecución de experimentos al sistema de gestión de información de laboratorios (LIMS) a utilizarse en el "Institut Pasteur de Montevideo". Con tal finalidad, se planteó la posibilidad de diseñar y ejecutar workflows que permitieran modelar los flujos mencionados.

Durante la etapa de análisis la propuesta fue evolucionando para dar lugar a la solución final denominada BioFlows, la cual consiste en un sistema de diseño y ejecución de workflows adaptable a diversas instituciones con perfil biológico o de laboratorio. Esta adaptabilidad se logra mediante la configuración de interfaces externas que permiten interactuar con otros sistemas.

El sistema BioFlows aborda cinco grandes áreas: el modelado de los procesos biológicos como workflows, la seguridad de la aplicación, el repositorio de procesos definidos por los usuarios, la interfaz de usuario y por último, la parte central para la comunicación con sistemas externos, el componente de interacción con LIMS.

Para la construcción del sistema se utilizó la plataforma de desarrollo empresarial Java EE, optando por el servidor de aplicaciones JBoss. Para el diseño y ejecución de workflows se utilizaron las herramientas JaWE y jBPM respectivamente. Para la comunicación entre aplicaciones se utilizaron web services implementados mediante el motor Axis2. La autenticación y organización de usuarios se delegó a una aplicación externa mediante el protocolo LDAP. Para la persistencia se utilizó el servidor de bases de datos relacionales PostgreSQL, haciendo uso de la herramienta Hibernate para el mapeo del modelo orientado a objetos al relacional.

El sistema BioFlows se integró exitosamente con el LIMS a utilizarse en el "Institut Pasteur de Montevideo". La solución desarrollada cumplió el objetivo principal de permitir la gestión y ejecución de flujos de experimentos, presentando otros aportes como son: un sistema de alta usabilidad, utilización de estándares para facilitar la interoperabilidad e interacción configurable con sistemas externos.

Palabras Claves

Workflow, Procesos, PIMS, LIMS.

Agradecimientos

A nuestras familias por el apoyo incondicional y su cariño.

A Laura y Raúl por la paciencia y el apoyo.

A la facultad en general por habernos albergado e instruido durante estos seis años.

Tabla de Contenido

1. INTRODUCCIÓN.....	1
1.1. MOTIVACIÓN	1
1.2. OBJETIVOS	2
1.3. PROYECTO	3
1.4. ORGANIZACIÓN DEL DOCUMENTO	5
2. ESTADO DEL ARTE	6
2.1. ¿QUÉ ES UN WORKFLOW?	6
2.2. ESTÁNDARES EN WORKFLOWS	7
2.3. MOTORES DE WORKFLOW	8
2.4. DISEÑADOR DE PROCESOS	10
2.5. TECNOLOGÍAS DE DESARROLLO.....	12
3. PLANTEO DE LA SOLUCIÓN	20
3.1. INTRODUCCIÓN	20
3.2. DESCRIPCIÓN GENERAL	20
3.3. PROCESOS.....	21
3.4. SEGURIDAD	23
3.5. INTEGRACIÓN CON LIMS	25
3.6. REPOSITORIO DE PROCESOS.....	28
3.7. INTERACCIÓN CON EL USUARIO	28
3.8. DESARROLLO DEL SISTEMA.....	34
4. DISEÑO	35
4.1. ARQUITECTURA	35
4.2. DIAGRAMA DE DISTRIBUCIÓN	39
5. CASO DE ESTUDIO: PIMS.....	41
5.1. INTRODUCCIÓN	41
5.2. TRABAJO REALIZADO	41
5.3. RESULTADOS OBTENIDOS	51
6. CONCLUSIONES	52
7. TRABAJO A FUTURO.....	54
7.1. DISEÑADOR DE PROCESOS WEB	54
7.2. CONVERSIÓN TOTAL DE XPDL 2.0 A JPD.....	54
7.3. EXTENSIÓN DEL FRAMEWORK DE EJECUCIÓN DE WEB SERVICES.	54
7.4. SEGURIDAD	55
7.5. GESTIÓN DE ROLES	55
7.6. PERMISOS DE EJECUCIÓN DE PROCESOS	55
8. REFERENCIAS	56
9. ANEXOS	61

1. Introducción

Este proyecto de grado se desarrolló en el contexto del convenio LIMS que se lleva adelante en forma conjunta por el "Institut Pasteur de Montevideo" y el Instituto de Computación (In.Co.). Al inicio del proyecto, se planteó el requerimiento de agregar a una herramienta de gestión de información de laboratorios (LIMS [48]), la posibilidad de diseñar y ejecutar workflows [9] que permitieran modelar el flujo de ejecución de experimentos que se da en estos laboratorios. Esto implica contar con la capacidad de diseñar workflows, donde cada paso es un experimento y contar con un módulo de ejecución mediante el cual, quienes realizan los experimentos pueden consultar y completar la información necesaria para avanzar al siguiente paso (experimento).

Durante la etapa de análisis, la propuesta fue evolucionando para dar lugar a la solución final, la cual comprende un sistema de diseño y ejecución de workflows adaptable a diversas instituciones. Esta adaptabilidad se logra mediante la configuración de interfaces externas que permiten interactuar con otros sistemas, en particular el LIMS que se utilice en la institución en cuestión. Como resultado del proyecto se obtuvo una solución genérica que excede los objetivos iniciales.

1.1. Motivación

Como se indicó previamente, este proyecto de grado surge en el marco del convenio LIMS, en el cual se estudiaron diversos sistemas de gestión de laboratorios y se decidió tomar como base un sistema llamado PIMS [49]. Éste está diseñado para satisfacer las necesidades de laboratorios biológicos y se encuentra especializado en el tratamiento de proteínas.

Los LIMS vinculan múltiples elementos, entre los que se encuentran: datos públicos, investigadores, laboratorios, proyectos, protocolos, resultados, equipamientos, muestras, herramientas de análisis, etc. Una de las funcionalidades principales de estos tipos de sistemas es el registro de información relativa a la ejecución de experimentos. En general, cuando se efectúa un experimento se sigue cierto protocolo, que define la forma en que se debe efectuar el mismo. Cumpliendo dicho protocolo, el investigador ingresa al sistema LIMS datos relativos a la ejecución del experimento, como ser las condiciones en las que se efectúa, muestras que se utilizan, resultados intermedios, comentarios, etc. Cuando se hace referencia a los LIMS se está hablando de un tipo de sistemas dedicado al tratamiento de las necesidades de los laboratorios, mientras que el PIMS es un

sistema propiamente dicho que cumple con la definición de los sistemas de tipo LIMS, pero que está especializado en el tratamiento de proteínas como se explico previamente.

Es importante destacar, que si bien el PIMS cuenta con la capacidad de gestionar muestras y experimentos, no es posible utilizarlo para definir flujos de ejecución de experimentos (procesos), lo cual es una necesidad manifestada por los investigadores para integrar en un sistema toda su operativa de trabajo.

Los sistemas de workflow se definen como sistemas que ayudan a las organizaciones a especificar, ejecutar, monitorear y coordinar los pasos de un flujo de trabajo dentro de un ambiente de oficina distribuido. Al hablar de workflows de negocio, se hace énfasis en el modelado de la realidad a través de procesos que describen los pasos a seguir por la organización para completar una actividad.

En respuesta a la carencia que presenta el PIMS en cuanto a la gestión de procesos, se planteó el interés por el desarrollo de una solución que pudiera complementar al mismo en este aspecto. Se propuso modelar como procesos de negocio las secuencias de experimentos llevadas a cabo por un laboratorio, beneficiando a los investigadores con una herramienta capaz de organizar actividades que comprenden varios experimentos. El proyecto estuvo especialmente motivado por la investigación y utilización de tecnologías actuales, como medio para promover determinadas características, tanto del desarrollo como de la solución, tales como: velocidad de desarrollo, estandarización, integración, adaptabilidad, seguridad y usabilidad.

1.2. Objetivos

Este proyecto tuvo como objetivo principal proveer al sistema PIMS con la funcionalidad de workflow, brindando a sus usuarios un valor agregado capaz de aumentar la utilidad del sistema en el trabajo diario y a largo plazo. El proveer al PIMS con este tipo de funcionalidad implica por un lado el diseño de workflows y por otro la ejecución de los mismos.

El sistema debía ser de acceso web, de forma de poder ser utilizado desde distintos puntos de la organización sin requerir una instalación previa. Además debía estar implementado en Java [20] y de acuerdo a la infraestructura del instituto mencionado, se le dio preferencia al desarrollo sobre la plataforma JBoss [24], el DBMS PostgreSQL [34] y la autenticación a través del protocolo LDAP [33].

Se consideró que la usabilidad era un objetivo importante, ya que a través de ésta, los usuarios perciben la calidad del sistema y una alta usabilidad aporta a la adopción del mismo.

Otro aspecto importante era la seguridad del sistema, debiendo controlar el acceso en el diseño de los procesos y en la ejecución de los mismos.

1.3. Proyecto

En base a la realidad planteada, se realizó una etapa de análisis durante la cual se definió el alcance del sistema a desarrollar: **un sistema de workflow integrable con sistemas LIMS, que permita a los usuarios especificar procesos de su organización.**

Los principales aportes del proyecto son:

- **Manipulación de los procesos de la organización por parte de los usuarios.** El sistema permite crear, modificar y liberar procesos. También cuenta con la posibilidad de crear un proceso a partir de otros, haciendo una composición de los mismos. Además, permite vincular al diseño, información proveniente tanto del sistema LIMS que maneja la organización, como del gestor de usuarios y grupos de la misma. El sistema garantiza que solo un usuario puede manipular un proceso a la vez, permitiendo autorizar a otros usuarios mediante la funcionalidad de liberación.
- **Esquema de seguridad basado en roles e integrado con la organización.** El sistema cuenta con un esquema en el que los privilegios para acceder a distintas áreas del sistema se asignan a roles. Luego, a través de la integración con el sistema LDAP, se relacionan los roles del sistema a los grupos organizacionales. De este modo, se establecen los privilegios de cada usuario a través de los grupos a los que pertenece.
- **Entorno de acceso amigable a los usuarios y centralizado.** El sistema provee un único punto de entrada a través de una interfaz web, la misma presenta un diseño intuitivo y amigable.
- **Integración con el sistema tipo LIMS a partir de web services.** El sistema permite la configuración de web services a través de la interfaz web. La configuración incluye la ubicación del web service, selección de la operación y asignación dinámica de los parámetros. Esta funcionalidad constituye el eje central en la adaptabilidad a diversos LIMS.
- **Almacenamiento de la información.** Es centralizado, confiable y seguro mediante la utilización de una base de datos relacional. El almacenamiento centralizado permite a los usuarios trabajar desde cualquier terminal sin necesidad de transportar la información.

En la Figura 1 se puede ver de forma simplificada la distribución del sistema junto a las interacciones con el sistema LIMS y el servidor LDAP externos.

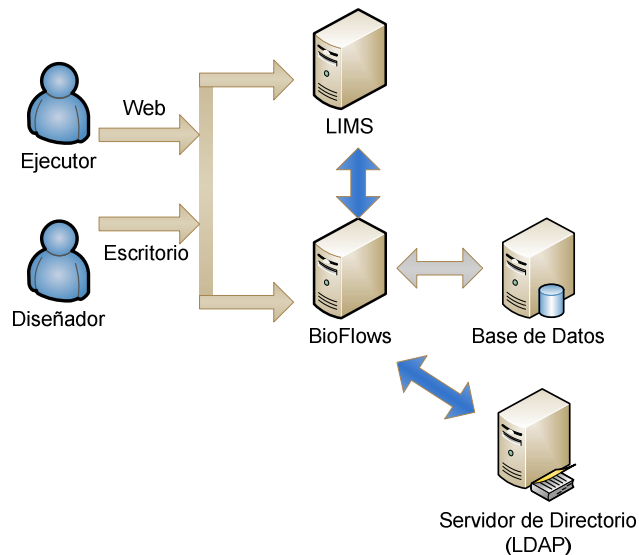


Figura 1. Distribución del Sistema.

El sistema desarrollado presenta dos interfaces de usuario, la primera es una interfaz web, la cual es el punto de entrada al sistema y permite administrar el mismo, ejecutar procesos, realizar actividades de supervisión y ejecutar el diseñador de procesos. Este último componente representa la segunda interfaz de usuario y consiste en una aplicación de escritorio implementada en Java que se ejecuta directamente desde la interfaz web utilizando Java Web Start [23].

El diseñador de procesos está basado en un editor de workflows open source llamado JaWE [17], al cual se le realizaron diversas extensiones para adaptarlo a las necesidades del proyecto. Para asistir la ejecución de procesos se utilizó el motor de workflows jBPM, este motor utiliza un lenguaje propietario para la definición de procesos (jPDL [7]). Debido a que el JaWE utiliza el estándar XPD [5] como lenguaje de descripción de procesos, fue necesario realizar una conversión de XPD a jPDL para cargar los procesos en el motor de workflows.

El componente central del sistema es el que provee los servicios para el resto de los componentes. Dicho componente está basado en la arquitectura J2EE [21] e implementado como EJB, haciendo fuerte énfasis en patrones de diseño. Por motivos de diseño y seguridad, este componente es el único con acceso al sistema de base de datos.

Como se mencionó anteriormente, el diseñador de procesos es ejecutado en el escritorio del usuario mediante Java Web Start. Para mantener unificado el

acceso a datos en el servidor, se implementó una interfaz de web services para brindar los servicios necesarios al diseñador de procesos, esto elimina la necesidad de un acceso directo desde el puesto cliente al servidor de bases de datos.

La gestión de usuarios es externa al sistema. Para esto se desarrolló un componente de interfaz con un servicio de directorio LDAP, delegando la autenticación y permitiendo adquirir la estructura organizacional del mismo. Siendo LDAP un estándar en servicios de autenticación, la interfaz desarrollada permite configurar rápidamente el sistema en cualquier organización que cuente con este tipo de servicio de directorio.

La solución presentada en este informe excede los objetivos iniciales, ya que se obtuvo un sistema genérico y adaptable, dado que no está acoplado a un único LIMS. Se logró una solución basada en tecnologías de punta, que demuestra la capacidad de configuración y adaptabilidad deseadas. Cabe resaltar que en el sistema existen funcionalidades que difícilmente son ofrecidas por aplicaciones de propósito similar, entre las que se destacan: la configuración dinámica de web services para integración con sistemas externos, el mapeo de roles con los grupos organizacionales del servidor LDAP y la amigabilidad de la interfaz gráfica para el diseño de procesos.

1.4. Organización del documento

En la sección 2 se aborda el estado del arte, presentando los conceptos y tecnologías investigadas al inicio del proyecto.

La sección 3 presenta la solución que se desarrolló cumpliendo con las especificaciones anteriormente definidas.

En la sección 4 se especifica el diseño de la solución para la construcción del sistema BioFlows.

En la sección 5 se profundiza en cómo se logró la integración con el PIMS (LIMS a utilizarse en el "Institut Pasteur de Montevideo"), que sirvió como caso de estudio del proyecto, utilizándose éste como sistema externo de integración.

En las secciones 6 y 7 se hace referencia a las conclusiones obtenidas a partir del proyecto y se presentan los posibles trabajos a futuro.

2. Estado del Arte

En esta sección se describen las tecnologías y estándares abordados. Éstos comprenden especificación de workflows, herramientas para diseñar procesos, motores de ejecución, frameworks para implementación de interfaces web y acceso a base de datos, protocolos de acceso a servicios de directorios de usuarios y tecnologías específicas de la plataforma de desarrollo.

2.1. ¿Qué es un workflow?

Un workflow es una formalización de un proceso de negocio. Por proceso de negocio se refiere a actividades que se realizan en un tiempo, en lugares específicos, por personas, con principio y fin, y que cuentan con entradas y salidas bien definidas. Se pueden diferenciar dos grandes categorías de workflows:

- De negocios: Se centran principalmente, en la programación de tareas, incluyendo dependencias no solo sobre los datos sino también entre agentes humanos. En general, se puede decir que éstos son más genéricos, siendo capaces de representar cualquier estructura de tareas e igualmente aplicable a actividades, como la organización y seguimiento de documentos, tanto en papel como electrónicos.
- Científicos: Centran su ejecución en el procesamiento de datos a través de varios algoritmos, aplicaciones y servicios. Éstos encontraron gran aceptación en campos como la bioinformática, cumpliendo las necesidades de tener varias herramientas interconectadas, y manejo de grandes cantidades de datos en distintos formatos.

Los workflows se caracterizan por su representación en forma de grafo, esta representación los caracteriza por su interpretación intuitiva y de fácil comprensión. En la Figura 2 se muestra un grafo que representa el workflow de un proceso para crédito de cheques.

2.2. Estándares en workflows

2.2.1. BPMN - Business Process Modeling Notation

Se trata de una notación gráfica para modelar procesos genéricos mediante diagramas desarrollado por BPMI (Business Process Management Initiative) [4]. Si bien existen variadas notaciones gráficas, BPMN [2][3] surgió con el objetivo de proveer una notación que fuese fácilmente entendible por todos los usuarios del negocio, tiene especial importancia ya que fue adoptada por la Workflow Management Coalition (WfMC [6]) como su notación gráfica. Actualmente, BPMN se encuentra mantenido por el Object Management Group (OMG [1]) y representa un importante aporte para la estandarización de las notaciones de workflow.

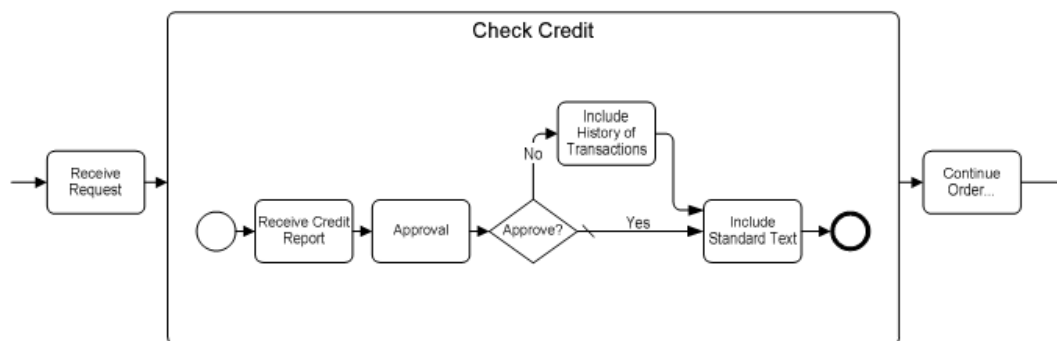


Figura 2. Ejemplo de uso de BPMN (Consulta de Crédito) [3].

2.2.2. XPDL - XML Process Definition Language

XPDL [5] es un estándar de la WfMC y actualmente, el lenguaje para definición de workflows más usado. A pesar de esto, gran parte de los motores de workflow continúan usando sus lenguajes propietarios. En la actualidad, la última versión existente es XPDL 2.0 la cual es una extensión compatible hacia atrás, pero con la gran virtud de ser capaz de representar cualquier diagrama BPMN. El lenguaje está basado en XML, lo cual lo hace comprensible para el ser humano y fácil de interpretar en cualquier lenguaje de programación. Los principales conceptos de definición de proceso que brinda XPDL se detallan en [61].

2.2.3. jPDL

jPDL [7] es el formato propietario definido por JBoss para la especificación de workflows. Este formato es interpretado por jBPM [13], el motor de workflows de la empresa JBoss, para la ejecución de los procesos. JBoss lo define como un lenguaje intuitivo para expresar gráficamente los procesos de negocio en términos de tareas, timers, acciones automáticas, estados en espera por acciones asincrónicas, etc.

2.3. Motores de workflow

2.3.1. jBPM

jBPM es un producto de JBoss que consiste en un motor de workflow y BPM [8] implementado como biblioteca Java. Presenta un modelo de programación simple con foco en el proceso. Para la definición de procesos utiliza un lenguaje propietario llamado jPDL, el cual es un lenguaje declarativo, no compatible con XPDL.

Si bien jBPM es una biblioteca, trae una interfaz web para utilizar las funcionalidades esenciales de ejecución. Para trabajar con jPDL incorpora un Designer, herramienta poco práctica para un usuario no técnico, principalmente porque está construida sobre Eclipse [36]. A esto se le suma que está dirigido a un usuario con conocimientos de programación, el cual puede automatizar tareas e interactuar con la API de jBPM embebiendo código Java en las propias tareas.

2.3.2. Bonita

Es una suite de workflow código abierto alojada en ObjectWeb y está implementada como aplicación J2EE. Cuenta con una interfaz web para el uso del workflow, y una interfaz Swing [41] para la edición de los mismos. Los procesos se representan en un formato XPDL extendido para cubrir ciertas necesidades. Apunta a la simplicidad de uso e instalación, haciendo foco en reducir la curva de aprendizaje, sacrificando performance por simplicidad. En versiones previas fue criticado por limitaciones en los procesos y problemas de escalabilidad, en la versión estudiada (versión 3.0), el equipo de Bonita [16] sostiene que estos obstáculos fueron superados. Dicha suite está principalmente orientada a procesos de larga duración, donde el foco está en los usuarios y no en tareas automatizadas.

La suite de Bonita comprende los siguientes componentes:

- Bonita workflow Server
 - Workflow Engine
- Bonita Graphical Tools
 - Workflow Console
 - xForms Editor
 - XPDL Editor (ProEd)

2.3.3. Enhydra.org workflow platform

Enhydra.org cuenta con dos proyectos de código abierto, por un lado, el motor de workflow Enhydra Shark [15] y por otro, un diseñador gráfico JaWE [17] [17] (Java workflow Editor). A su vez, existe un producto comercial basado en cada proyecto, estos productos comerciales agregan una gran cantidad de funcionalidades a los proyectos de código abierto. Una característica importante de estos proyectos es que están basados en XPDL puro (sin extensiones), lo que admite que las especificaciones puedan integrarse con herramientas que manejen XPDL. El motor de workflow está diseñado para ser fácilmente embebido en otras aplicaciones, lo cual lo hace apto para distintos tipos de plataformas como web o Swing. También provee interfaces Web para la administración y la gestión de las listas de trabajo, pudiendo estas interfaces ser personalizadas mediante xForms y XSLT.

2.3.4. OS workflow

Es un motor de workflow, sus creadores lo definen como una herramienta extremadamente flexible. Es considerado de bajo nivel debido a que la personalización de los procesos se consigue escribiendo XML y no mediante una interfaz gráfica. Por estas razones es una herramienta que debe ser administrada por usuarios técnicos. Una de las principales características de la herramienta es la posibilidad de definir acciones globales a todo un workflow de forma de reducir código necesario para la especificación. Otras de las características importantes es que provee una API para el acceso a las funcionalidades del motor y soporta invocaciones remotas mediante SOAP [27].

La definición de funciones soporta diferentes lenguajes:

- Java – based Functions (clases Java)
- Bean Shell Functions (lenguaje script)
- BSF Functions (perlscript, vbscript, Javascript)
- Utility Functions (son funciones predefinidas de la herramienta)

2.4. Diseñador de procesos

2.4.1. JaWE

Es un editor gráfico basado en Swing que sigue todas las especificaciones definidas por WfMC en el estándar. Permite la edición y visualización de cualquier workflow definido siguiendo el estándar XPDL y que no contenga ninguna extensión al mismo. La notación gráfica que maneja no sigue el estándar BPMN. Otra de las características con las que cuenta el editor gráfico es que puede utilizarse a través de Java Web Start.

2.4.2. Bonita

En la sección 2.3 se hizo referencia a la suite Bonita y se mencionó los componentes que conforman la interfaz gráfica para la especificación de procesos. Éstos se agrupan bajo el nombre Bonita Graphical Tools y son los siguientes: Workflow Console, xForms Editor, XPDL Editor (ProEd). Debido a que, el XPDL que genera contiene extensiones específicas de la herramienta, los modelos generados no son directamente interoperables con otros editores y motores de workflow que soportan XPDL.

2.4.3. OS Workflow

El proyecto OS Workflow [14] cuenta con una herramienta gráfica usada para la especificación de workflows, la misma se denomina "GUI Designer" y es una aplicación basada en Swing. Existen diversas formas de poder acceder a la misma, entre ellas se encuentra la posibilidad de ejecutar el editor a través de Java Web Start. Como se mencionó, el código de especificación de workflows para este proyecto es un XML propietario, por lo que los workflows generados por esta herramienta sólo podrán ser soportadas por el motor OS Workflow.

2.4.4. Taverna

Taverna [18] es una herramienta de código abierto para el diseño y ejecución de workflows, creada por myGrid Project [19]. Está compuesta por una herramienta de escritorio para la creación de workflows y motor de ejecución para los mismos. Los workflows son expresados en ScufI (Simple Conceptual Unified Flow language). Permite a los usuarios integrar diversas herramientas de software, por ejemplo, servicios provistos por National Center for Biotechnology Information [50], European Bioinformatics Institute (EBI) [51], etc. Para poder ejecutar Taverna se necesita tener instalada la versión 5 de Java y no es necesario tener ningún tipo de base de datos u otra aplicación.

Los creadores de la herramienta han definido que la mejor manera de entender a Taverna es usándolo. El sistema fue desarrollado para que pueda ser utilizado fácilmente por biólogos y bioinformáticos con conocimientos básicos en informática, y para soportar construcciones para análisis complejos obteniendo información tanto de fuentes públicas como privadas.

Taverna es una colaboración entre European Bioinformatics Institute (EBI), IT Innovation [52], The School of Computer Science [53], Newcastle Centre for Life [54], School of Computer Science at the University of Manchester [55] y Nottingham University Mixed Reality Lab [56].

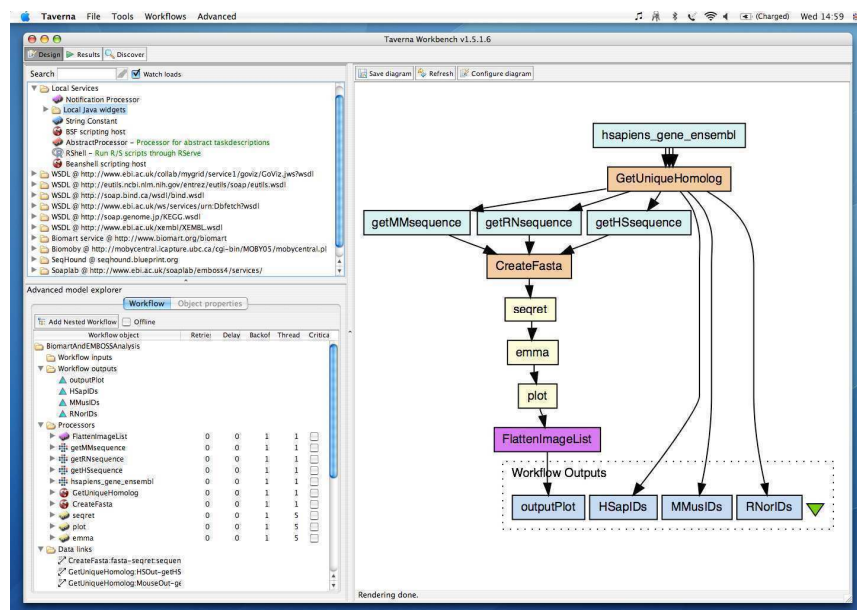


Figura 3. Taverna, Ejecución de un workflow [18].

2.5. Tecnologías de desarrollo

2.5.1. Tecnologías Base

2.5.1.1. J2EE

Java Platform Enterprise Edition es una plataforma de programación (parte de la plataforma Java) para desarrollar y ejecutar aplicaciones hechas en Java con arquitectura de n niveles distribuida, basándose en componentes de software modulares que ejecutan sobre un servidor de aplicaciones.

2.5.1.2. JBoss

Es un servidor de aplicaciones J2EE de código abierto, implementado en Java. JBoss puede ser utilizado en diversos sistemas operativos.

2.5.1.3. Swing

Es un framework para el desarrollo de interfaces gráficas. Tiene la característica de ser independiente de la plataforma en la que corre, esto se debe a que no usa los controles de GUI del sistema operativo en el que se ejecuta. Si bien esta característica permite que las aplicaciones se vean y comporten igual en todas las plataformas, tiene un costo en performance.

2.5.1.4. Hibernate

Hibernate [31] es un motor de persistencia que permite realizar la traducción entre modelos orientados a objetos y relacionales. Permite persistir los objetos realizados en el modelo orientado a objetos a un modelo relacional, sin perder ninguna de sus características (asociación, herencia, polimorfismo, composición y colecciones). Las consultas a estos objetos persistidos se realizan mediante un lenguaje SQL propio (HQL), así como en SQL nativo o propiamente manejando los conceptos del modelo orientado a objetos.

2.5.1.5. Jasper Report

Jasper Report [43] es una herramienta de código abierto que se compone de un conjunto de bibliotecas Java para facilitar la generación de reportes. Los reportes se definen a través de un formato xml. La salida de los mismos puede ser en diversos formatos, como lo son: pdf, csv, xml, txt, html, xls, rtf y jasper viewer.

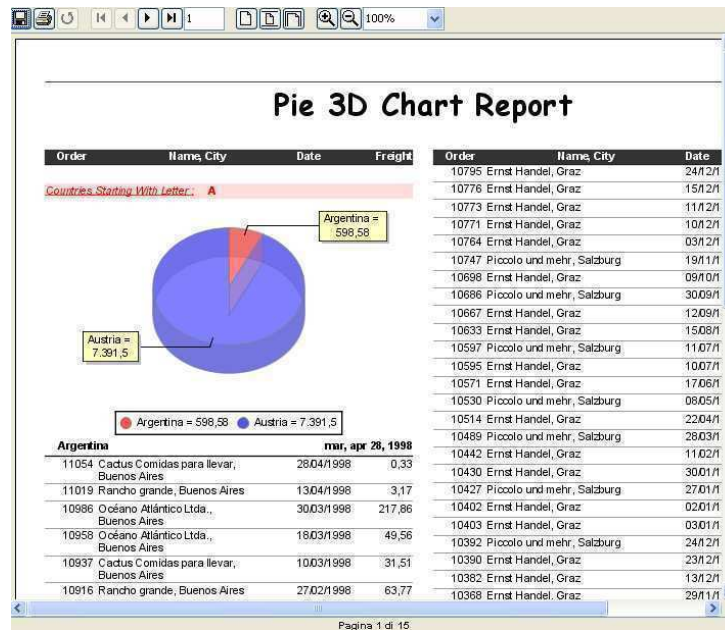


Figura 4. JasperReport, visualización de un reporte [43].

La definición de los reportes se puede realizar escribiendo directamente el xml o utilizando la herramienta IReport [42] para diseñarlos gráficamente, la cual viene incluida con el producto.

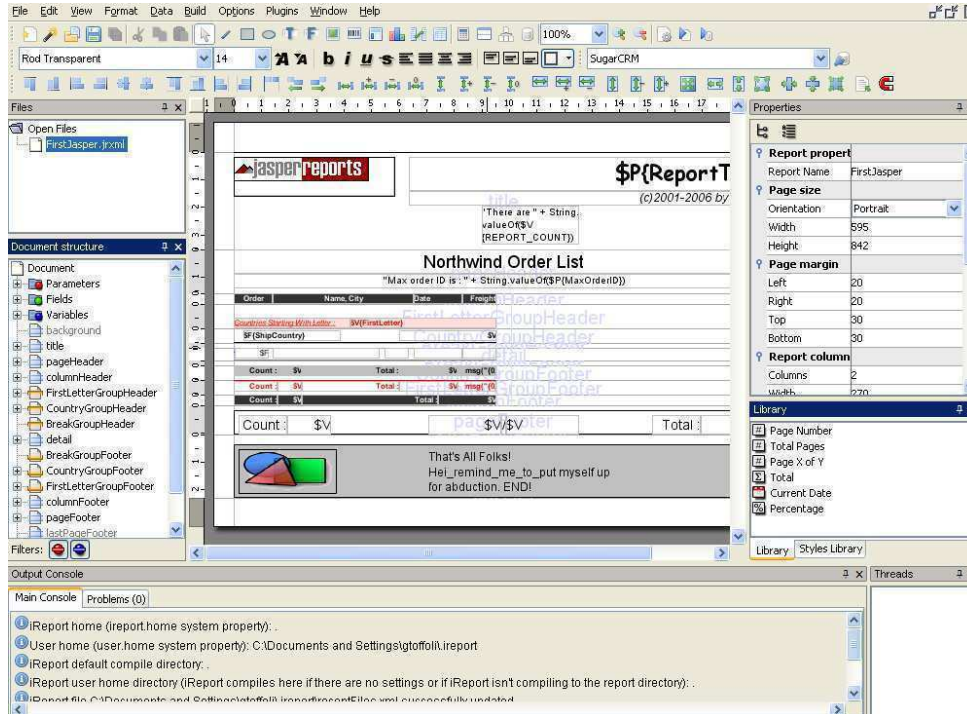


Figura 5. IReport, herramienta para la creación de reportes [42].

2.5.2. Tecnologías Web

2.5.2.1. JSF

JavaServer Faces (JSF) [40] es un framework para aplicaciones web Java, el cual simplifica el desarrollo de interfaces J2EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también puede usar otras tecnologías.

JSF provee:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Un conjunto por defecto de componentes para la interfaz de usuario.
- Dos bibliotecas de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- Beans administrados.

2.5.2.2. RichFaces

RichFaces [38][39] es una biblioteca de componentes para JSF que permitir integrar fácilmente componentes Ajax en el desarrollo de aplicaciones.

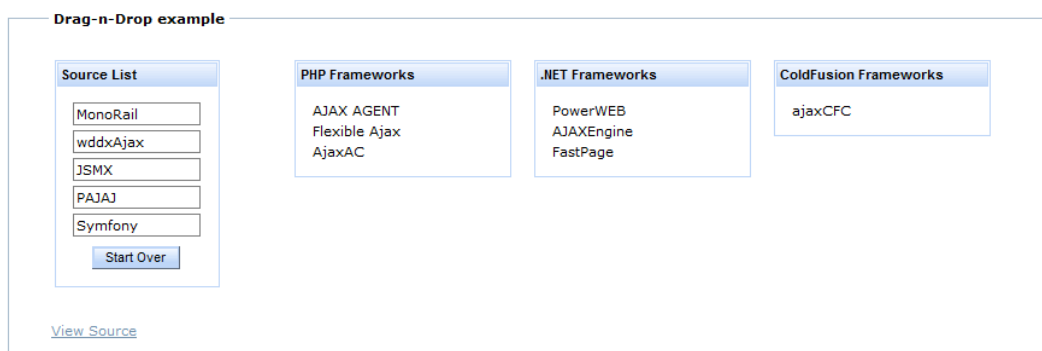


Figura 6. Drag-Drop Support, Drop Support [39].

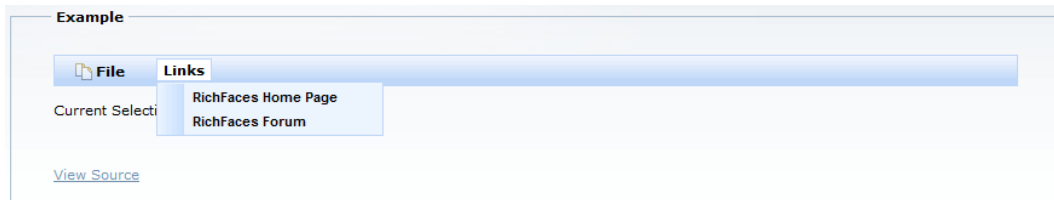


Figura 7. Rich Menu, Drop Down Menu [39].

Incluye un fuerte apoyo a la skinnability de aplicaciones JSF, esto significa que es sumamente fácil cambiar la apariencia de toda la aplicación. Uno de los objetivos de RichFaces es aprovechar al máximo los beneficios de JSF incluyendo el ciclo de vida, la gestión de recursos estáticos y dinámicos de las aplicaciones.

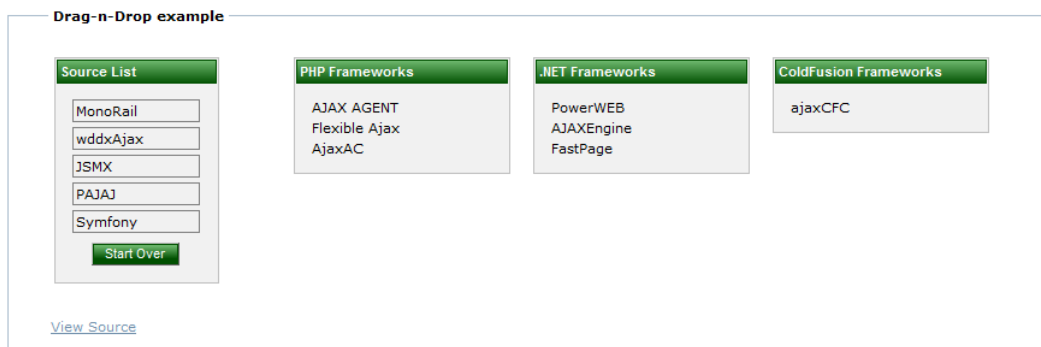


Figura 8. Cambio de skin [39].

2.5.2.3. Web services

La tecnología de web services consiste en un conjunto de protocolos y estándares que sirven para el intercambio de información entre diversas aplicaciones. Permite que los sistemas que intercambiaran información sean desarrollados en diversos lenguajes de programación y ejecutados en diferentes plataformas. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Hoy en día existen organizaciones responsables de la arquitectura y los reglamentos de los web services. Algunos de los estándares son:

- XML (Extensible Markup Language) [29], formato estándar para intercambiar información.
- SOAP (Simple Object Access Protocol) [27], protocolo sobre el cual se establece el intercambio de información.
- WSDL (Web Services Description Languages) [28], lenguaje de interfaz pública para los web services. Éste consiste en una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los web services que describe.

Algunas de las ventajas que representan estos estándares son las siguientes: aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que corren; fomentan los estándares y protocolos basados en texto, lo que hacen más fácil acceder a su contenido y entender su funcionamiento.

La principal desventaja es su bajo rendimiento en comparación con otros modelos de comunicación distribuida, tales como RMI (Remote Method Invocation). Éste es uno de los inconvenientes derivados de adoptar un formato basado en texto, elección que se debe a que XML es un formato diseñado para ser fácilmente comprensible por personas.

Axis2

Es el nuevo motor de web services de Apache. Su arquitectura ha sido diseñada desde cero, teniendo en cuenta los errores cometidos en Axis1. Cabe destacar su mejor rendimiento, pero sobretodo su sistema de módulos que permite, de forma sencilla, añadir nuevas funcionalidades y soportar futuras especificaciones sobre webservices. Axis2 [28] puede utilizar SOAP 1.1, SOAP 1.2 y REST.

2.5.2.4. Java Web Start

Java Web Start [23] es la implementación de la especificación JNLP (Java Networking Launching Protocol) [24] y está desarrollada por Sun Microsystems. Esta especificación permite centralizar en un servidor web una aplicación Java de escritorio para poder ser utilizada por diversos clientes que tengan acceso al servidor, de esta manera se evitan problemas de instalación y distribución. Todos los recursos necesarios por parte de la aplicación se encuentran centralizados en el servidor, cuando un cliente accede a la misma a través de JWS (Java Web Start) por ejemplo a través de un link en un pagina web HTML, el servidor se encarga de proveer los recursos necesarios para su ejecución.

Otra de las características de esta tecnología es que asegura que la versión que se ejecuta en el cliente es la última, ya que ésta es actualizada si en el servidor se encuentra una nueva. La única restricción que tiene el cliente para poder ejecutar correctamente una aplicación a través de JWS es tener instalada la Java Virtual Machine. A partir de la versión 1.4 de JRE (Java Runtime Enviroment), JWS se encuentra incluido en el mismo.

Un ejemplo del funcionamiento es, un servidor web donde se encuentra una página web HTML con links a aplicaciones Java, donde cada uno de éstos apunta a archivos .jnlp que indican la ruta de la aplicación en éste u otro servidor. Al

momento de acceder a uno de estos links, se inicia automáticamente JWS, se comprueba la seguridad, se actualiza la última versión de ser necesario y se ejecuta la copia local de la aplicación obtenida del servidor.

2.5.3. Servicio de directorio

2.5.3.1. LDAP

LDAP significa *Lightweight Directory Access Protocol*, es un protocolo que provee servicios de directorio ordenado y distribuido, organizando la información de forma muy similar a como lo hace un sistema de archivos, o el servicio de nombres de dominio (DNS) en Internet, tal como podemos ver en la Figura 9.

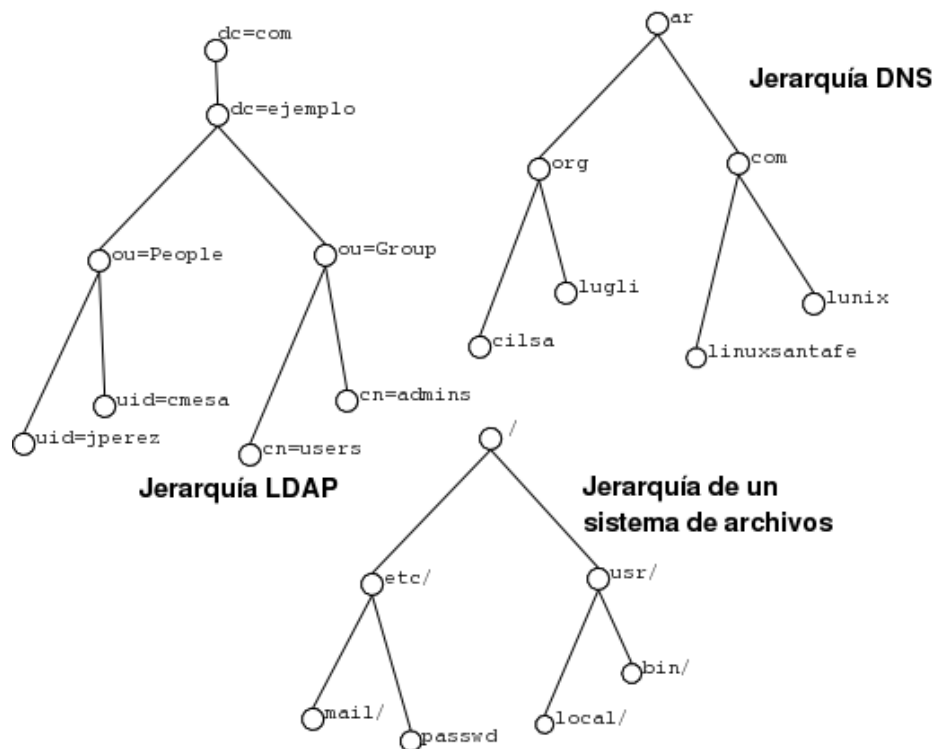


Figura 9. Ejemplo de definición de jerarquías [34].

LDAP funciona como una base de datos, está optimizado para operaciones de lectura y búsqueda. Por otro lado, no posee soporte para ingreso de datos por transacciones ni *rollback*, funcionalidades que se encuentran en los motores de base de datos relacionales. Habitualmente, almacena la información de login (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados, etc.).

La arquitectura cliente-servidor y estructura en forma de árbol que utiliza para almacenar su información, tiene algunas ventajas interesantes, como:

- Evita la duplicación de datos. La estructura de datos obliga a que no exista el mismo dato en dos lugares diferentes del esquema.
- Permite la distribución de la administración. Al igual que el servicio de DNS, la responsabilidad en la administración de los datos de un árbol se puede separar entre distintos equipos si es necesario.
- Acepta niveles de acceso bien detallados. Es posible definir políticas de seguridad por cada nodo.

OpenLDAP

Es un proyecto de código abierto que tiene como objetivo el desarrollo de una suite que implemente el protocolo LDAP, también tiene la posibilidad de soportar múltiples esquemas por lo cual puede conectarse a cualquier otro LDAP. Se define como un esfuerzo conjunto para desarrollar un grupo de aplicaciones y herramientas de desarrollo. En el proyecto se marca como objetivo el interés de que tenga un alto grado de robustez. La gestión del mismo es llevada a cabo por una comunidad de voluntarios que se interrelacionan a través de Internet para planificar y desarrollar el proyecto.

Netscape Directory SDK

Netscape Directory SDK [32] es una biblioteca Java que permite desde una aplicación tener acceso a un LDAP. Dentro de las funcionalidades soportadas están: recuperación, modificación, borrado e insertado de datos.

2.5.4. LIMS

Los LIMS (Laboratory information management system) [48] es un software utilizado en laboratorio para la gestión de muestras, usuarios, instrumentos, estándares y otras funcionalidades como pueden ser la facturación o automatización de procesos. Los LIMS están fuertemente relacionados con los LIS (Laboratory Information System), la principal diferencia es que éstos últimos se usan generalmente en Clínicas u Hospitales, mientras que los LIMS están más orientados a la investigación y análisis.

2.5.4.1. PIMS

El PIMS (Protein Information Management System) [49] es un proyecto que tiene como objetivo la construcción de un sistema de tipo LIMS para satisfacer las necesidades de laboratorios biológicos. El resultado del mismo es un sistema denominado de la misma forma, con las características de uso libre para toda la comunidad científica. Éste ha sido construido pensando en la flexibilidad de implantación en todos los ambientes científicos y en el acceso distribuido al mismo,

por esto es que se diseñó un sistema web de forma de poder cumplir con estos objetivos.

La primera versión liberada del producto fue la "PIMS Version 0.01" el 18/03/2005, actualmente, el producto se encuentra en la versión "PIMS Version 2.0", liberada el 15/01/2008. Hoy en día, no se cuenta con un plan de futuros trabajos para el proyecto. El sistema está hecho con tecnología Java y HTML, puede ejecutarse en diversos servidores de aplicaciones como Tomcat [26] y JBoss entre otros.

Entre los conceptos principales del PIMS se encuentran los experimentos, que representan los trabajos realizados por un investigador en el laboratorio. Los experimentos se definen basados en protocolos, estos últimos constituyen la definición abstracta de un experimento. El protocolo define, entre otras cosas, los lineamientos que deberán seguirse para la ejecución del experimento, así como sus entradas y salidas. La manipulación de experimentos consiste en la gestión (creación, edición y borrado) de experimentos, el cambio de estado y la edición de entradas y salidas, entre otras características.

El PIMS también cuenta con otras funcionalidades, como pueden ser la generación de reportes y búsquedas en el sistema.

The screenshot displays the main menu of the Protein Information Management System 2.0 (PIMS). The interface is organized into several functional panels. At the top, there is a navigation bar with links for Home, Log out demo, Target, Experiment, People & Places, History, and Help. The main content area is divided into several sections: 'Active samples older than 7 days' (listing various sample IDs), 'Active samples ready for use' (showing 'None'), 'History' (listing recent items like 'Demo PIMS', 'Plate (from order) 63102', and 'SSPF Purification 1'), 'Leads construct management' (with a 'Load Primer Order Construct Management' link), 'Samples assigned to demo' (showing 'None'), 'Constructs no progress for 7 days' (listing 'TTX 1565.N', 'Tx 1576.N', and 'VC0717.N'), and a 'Search by barcode' section with an input field and a 'Search' button. A 'Perspective: standard' dropdown is visible in the top right corner.

Figura 10. Menú principal del PIMS [49].

3. Planteo de la Solución

3.1. Introducción

La búsqueda de la solución comenzó a partir del objetivo principal, que es la necesidad de proveer al PIMS con la capacidad diseñar y ejecutar procesos. Finalmente, se obtuvo una solución lo suficientemente genérica para que fuera adaptable tanto al contexto de referencia, como así también a otros contextos similares.

Para el desarrollo de la solución se puso foco, principalmente, en los siguientes puntos:

- Usabilidad y amigabilidad del sistema por parte de los usuarios no técnicos.
- Seguridad en el acceso a los datos.
- Adaptabilidad con distintos LIMS.
- Se utilizaron tecnologías para realizar un diseño más simple y disminuir los tiempos de desarrollo.

3.2. Descripción General

El sistema implementado denominado BioFlows, es una Plataforma Integral para la Gestión de Procesos en Sistemas Biológicos. En la Figura 11 se presenta un diagrama general del sistema y sus interacciones con aplicaciones externas. Entre sus principales características se pueden destacar su capacidad de adaptarse a distintos LIMS y su cualidad de ser un producto integral, ya que provee todo lo necesario para trabajar con procesos biológicos, contando con:

- Diseñador de procesos.
- Interfaz de ejecución de procesos.
- Interfaz administrativa de configuración.
- Reportes sobre estado de procesos.

El sistema delega el manejo de usuarios, debido a que la autenticación se realiza a través un servidor LDAP. La integración con el LIMS se realiza mediante web services configurables (tanto a nivel de url, como de los parámetros que reciben) a través de la interfaz administrativa. El sistema está concebido para tener un acceso web.

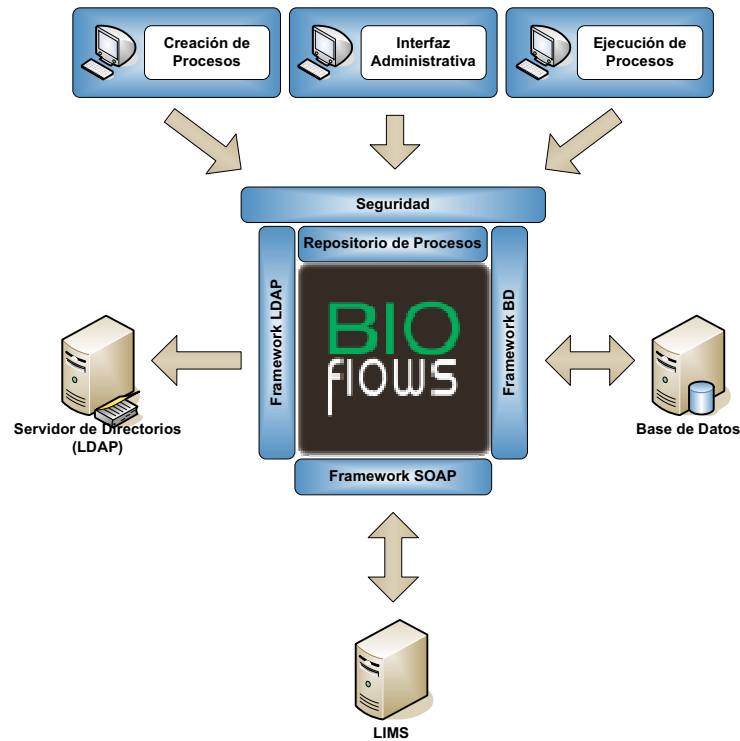


Figura 11. Descripción general del sistema BioFlows.

Para la construcción de la solución se abordaron cinco grandes áreas:

- Procesos.
- Seguridad.
- Integración con LIMS.
- Repositorio de Procesos.
- Interacción con el usuario.

3.3. Procesos

Como primera área clave se describe el modelo que se realizó para los procesos, luego se desarrolla como se implementó dicho modelo.

3.3.1. Modelado de Procesos

La primera actividad fue modelar un proceso, el mismo es una sucesión de experimentos, cada uno de éstos tiene entradas, salidas y datos internos. Esta sucesión tiene cierto orden y bifurcaciones, pudiendo ejecutar distintos experimentos bajo ciertas circunstancias, tanto del nodo actual como de la sucesión transcurrida. Este orden de ejecución es responsabilidad del usuario del sistema. Debido a la semejanza de este funcionamiento con el de un workflow, se investigó

la posibilidad de modelar un proceso de esta forma, comprobando finalmente, que esto era viable. Para representar los flujos de trabajo en un laboratorio se desarrolló el siguiente modelo:

- Cada nodo es una actividad de propósito general o un experimento.
- Las relaciones que unen dos nodos, son los posibles experimentos (o actividades de propósito general) a ejecutar en un próximo paso a partir del nodo actual.
- La navegación entre los distintos nodos es responsabilidad del usuario utilizando las relaciones entre éstos.
- Cada nodo tiene su estado, el mismo está determinado por sus entradas, datos internos y salidas.

3.3.2. Motor de Workflow

Una vez modelado el proceso como un workflow se inició la búsqueda de un motor de éstos. Los criterios utilizados para la búsqueda fueron los requerimientos no funcionales, la solución debía ser implementada en J2SE-J2EE [22] y que el servidor de aplicaciones fuera JBoss. El primer paso que se tomó fue la búsqueda de distintos motores de workflow en la comunidad. Luego de una selección inicial a partir de requerimientos no funcionales y limitaciones de los distintos productos como ser, funcionalidades poco claras, mala o nula documentación y aceptación en la comunidad, se iniciaron las pruebas sobre dos motores de workflow, Bonita de ObjectWeb y jBpm de JBoss. En la Tabla 1 se puede ver un cuadro comparativo entre ambos.

	jBPM	Bonita
Lenguaje de definición de procesos	jPDL	XPDL
Performance	Buena	Criticada
Deploy en JBoss	Si	En versión Beta

Tabla 1. Cuadro comparativo entre jBPM y Bonita.

Si bien ambos son similares en lo que atañe al funcionamiento, cada uno presenta sus particularidades. En ambos, los workflows se definen mediante un XML, la diferencia radica en que Bonita utiliza XPDL que es un estándar de la WfMC (si bien lo utiliza extendido para dar soporte a algunas funcionalidades extras), mientras que jBPM utiliza jPDL que es un lenguaje propietario del mismo.

En lo que respecta a las funcionalidades que ambos presentan, realmente son muy similares, ambos tienen una interfaz de ejecución web, debido a su complejidad está muy lejos de ser una interfaz de usuario que ejecute procesos, ya que está orientada a usuarios técnicos.

Respecto al diseño de procesos, Bonita presenta una aplicación Java Web Start y jBPM presenta un plug-in de Eclipse, ambas son poco adecuadas para ser utilizadas por usuarios que diseñen procesos.

Otro punto que se consideró fue la performance, como no fue posible realizar pruebas de carga por falta de infraestructura, se buscó en la comunidad los comentarios sobre JBPM y Bonita, en éste aspecto. Bonita fue reiteradamente criticado por su mala performance, mientras que por el contrario jBPM no presentaba este tipo de críticas.

Por último se tuvo en cuenta que el ambiente de ejecución iba a estar basado en JBoss, dado que jBPM es de la misma compañía se simplifica la interacción e integración de estos.

Debido a los problemas de performance y la dificultad para hacer correr sobre JBoss, Bonita fue descartado y se eligió jBPM de JBoss, a pesar de que jBPM utiliza un lenguaje propietario (jPDL) para la definición de sus workflows y se busca utilizar un lenguaje estándar para la definición de procesos. Ésto agrega una problemática cuya solución será desarrollada más adelante en este documento.

3.4. Seguridad

Siguiendo con las áreas claves de la solución elaborada, en esta sección se describen los componentes y decisiones tomadas en la construcción del motor de seguridad de la aplicación.

3.4.1. Usuarios y Roles

Como se explicó anteriormente, el sistema delega el manejo de Usuarios y Roles. La aplicación cuenta con cuatro roles estáticos:

- Administrador
- Diseñador de Procesos
- Ejecutor de Procesos
- Supervisor

Cada uno de estos roles tiene asociado un conjunto de permisos también estáticos. En la Figura 12 se observan las responsabilidades de cada uno de los roles.

El sistema autentica contra un servidor LDAP, por otro lado, se contempla que cada usuario pertenece a uno o más grupos dentro de este servidor. Dado que el sistema maneja las autorizaciones a nivel de los roles descritos previamente, es necesario relacionar los usuarios a estos roles, para lo cual se desarrolló una funcionalidad que permite asignar roles del BioFlows a cada grupo del LDAP. De esta manera, cuando se quiere conocer los permisos de un usuario, se consultan los grupos a los que pertenece y a través de éstos se establecen los roles que tiene. Una vez que se tienen los roles del usuario, se puede verificar si el usuario está autorizado a utilizar el recurso que solicita.

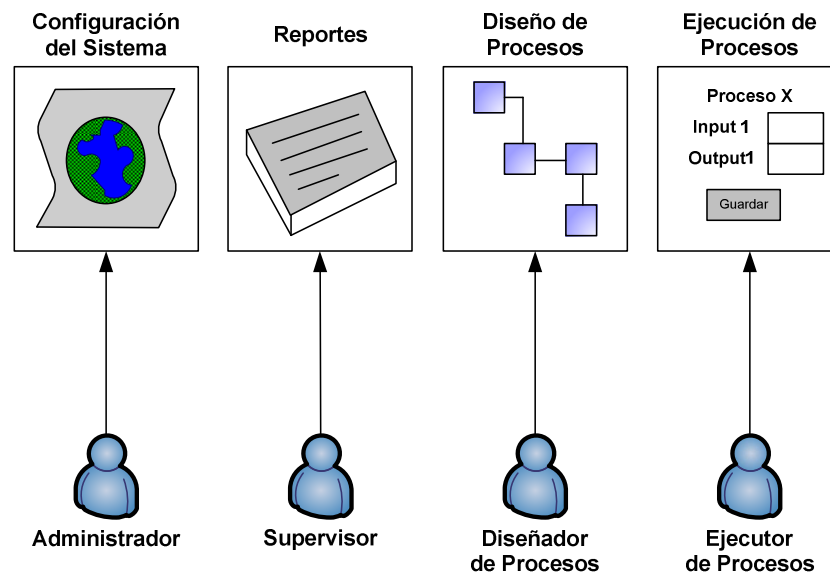


Figura 12. Roles del sistema.

3.4.2. Seguridad y Auditoría

El esquema de seguridad y validación es una parte muy importante del servidor. Se realiza control de permisos a nivel de servicios del sistema y a nivel de presentación, de acuerdo al siguiente comportamiento:

- El usuario se loguea en el sistema.
- Se valida su usuario y contraseña contra el servidor LDAP.
- Si la validación es correcta se registra la sesión del usuario en el sistema.
- Se asigna un identificador y un vencimiento a la misma (cada cierto tiempo se eliminan las sesiones vencidas).
- Toda esta información es devuelta como resultado del login, quedando la misma disponible en la sesión HTTP del contenedor para cada usuario.

La seguridad a nivel de servicios del sistema se realiza de la siguiente forma:

- El usuario invoca una operación en el sistema.
- Ésta se intercepta por el manejador de seguridad.
- Se corrobora que existe una sesión abierta y no vencida (si está vencida se elimina la sesión abierta) para el identificador de sesión del usuario.
- En caso de ser esto positivo, se procede a obtener los grupos a los cuales pertenece dicho usuario en el servidor LDAP y a través de los mapeos se obtienen los roles del BioFlows.
- Se verifica si el usuario está autorizado a partir de los roles, la interfaz del sistema que se está intentando acceder y la operación ejecutada.

La seguridad a nivel de presentación se realiza sólo en la parte del sistema cuya interfaz es web. La misma consiste en interceptar todas las peticiones de páginas del sistema, la autorización se verifica de forma análoga a la seguridad de servicios. Cabe acotar que la validación de seguridad sobre páginas web es una operación dentro del sistema BioFlows y que todos los roles tienen acceso a ésta.

La auditoría se registra a nivel de la base de datos, por cada invocación que se realiza al sistema, ya sea servicio del sistema o página web, se registra usuario, IP, fecha-hora y si obtuvo el recurso o no.

3.5. Integración con LIMS

Para lograr integrar el sistema BioFlows con el LIMS se desarrolló un framework, que permite ejecutar operaciones de integración de forma dinámica, estas operaciones se configuran en tiempo de ejecución por los usuarios del sistema BioFlows que cuentan con permisos administrativos.

El framework desarrollado cuenta con un componente estático en el que se tienen configurados los servicios de integración disponibles en el sistema y los parámetros disponibles para utilizar en la invocación. Por otra parte, se tiene un componente dinámico en el que el usuario especifica la url del web service, la operación a ejecutar, que parámetros de los disponibles se van a utilizar y el orden de éstos.

Cuando se ejecuta un web service de integración simplemente se especifica el nombre y se envían todos los parámetros disponibles. En este punto el framework utiliza la información configurada por el usuario para ejecutar el web service con los parámetros seleccionados. Por último, se devuelve un objeto que representa una abstracción XML del resultado.

El framework cuenta con algunos utilitarios que permiten consumir el WSDL para obtener las operaciones de éste, como también para poder obtener los parámetros de una operación y el tipo de éstos, ya sean básicos o complejos. En el caso de los complejos se desglosa hasta obtener los tipos básicos.

El sistema cuenta con seis servicios de integración, algunos obligatoriamente deben ser configurados y otros no. La lista completa de servicios de integración se puede visualizar en la Tabla 2.

Nombre	Descripción	Obligatorio
getInputFromProtocol	Obtiene las entradas de un protocolo.	Sí
getOutputFromProtocol	Obtiene las salidas de un protocolo.	Sí
getProtocols	Obtiene una lista de protocolos.	Sí
setTaskInputValue	Persiste el valor de una entrada para un experimento.	No
setTaskOutputValue	Persiste el valor de una salida para un experimento.	No
finishExperiment	Finalización de un experimento.	No

Tabla 2. Servicios disponibles en el sistema BioFlows.

En las siguientes figuras se describe cómo se resuelve la integración con el LIMS, los servicios se dividen en dos grupos, los que se utilizan para el diseño de procesos (Figura 13) y los que se realizan para la ejecución de procesos (Figura 14).

En el diagrama de la Figura 13 se presentan cuales son los pasos en la creación de un proceso en el sistema BioFlows y como es la interacción con los sistemas de tipo LIMS. Los pasos en los cuales hay una interacción con sistemas externo son:

- **Obtener Protocolos:** En este paso se comunica con el sistema externo recuperando los protocolos existentes, el usuario puede seleccionar uno y este se insertará como una nueva actividad en el proceso que se esta diseñando.
- **Visualizar Entradas:** En este paso se recupera la información sobre los inputs que tiene definido un protocolo, previamente se debe seleccionar una de las actividades definidas dentro del proceso, está actividad debe ser del tipo de protocolo.

- Visualizar Salidas: Análogamente al paso Visualizar Entradas se presentan cuales son los outputs del protocolo seleccionado.

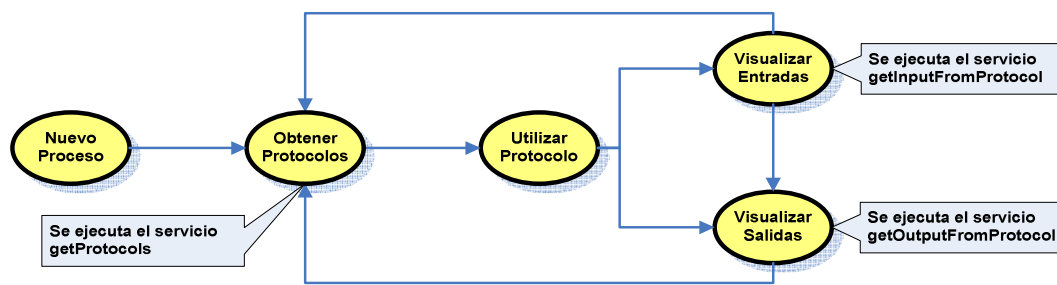


Figura 13. Interacción BioFlows - LIMS para el diseño de procesos.

En el diagrama de la Figura 14 se presentan cuales son los pasos en la ejecución de un proceso en el sistema BioFlows y como es la interacción con los sistemas de tipo LIMS. Los pasos en los cuales hay una interacción con los sistemas de tipo LIMS son:

- Salvar Entradas: En este paso se pueden actualizar los valores que tienen las entradas del experimento (instancia del protocolo definido en el proceso) cuando se guardan estos valores se actualiza la información del experimento creado en el sistema de tipo LIMS.
- Salvar Salidas: Análogo al paso Salvar Entradas con la diferencia que se actualizan los valores de las salidas del experimento.
- Ejecutar Transición: Al ejecutar este paso se cambia el estado del experimento en el PIMS a "OK".

Cabe aclarar que el comportamiento de las operaciones es específico al desarrollo hecho para este proyecto, como se explica en la sección 5 este se puede modificar.

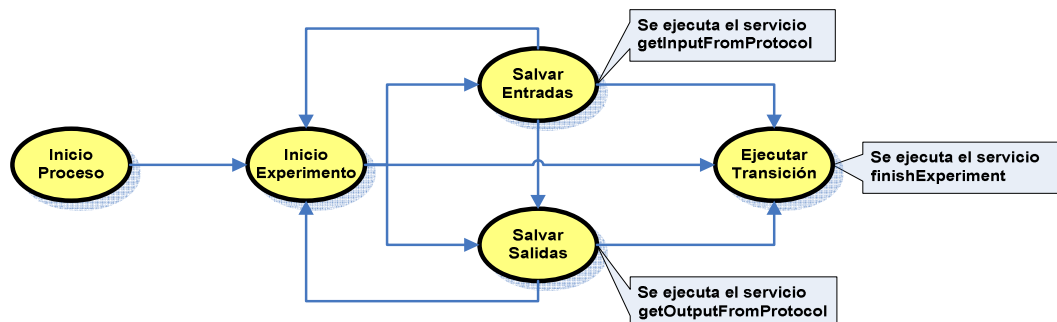


Figura 14. Interacción BioFlows - LIMS para la ejecución de procesos.

3.6. Repositorio de Procesos

Para el repositorio de procesos se buscó la forma de tener éstos de forma centralizada en el sistema BioFlows y no en el sistema de archivos de cada usuario, debido a los problemas de seguridad que esto conlleva. La solución fue persistir los procesos en la base de datos del sistema BioFlows, para lo cual se definió una política de uso que se explica a continuación.

Existen tres formas para salvar un proceso:

- Sólo para el usuario.
- Para determinados grupos
- Para todos los usuarios del sistema.

Cuando se quiere abrir un proceso, se le presentan al usuario sólo aquellos procesos a los cuales tiene permiso (los dados al salvar un proceso). Cabe acotar que la apertura de un proceso se puede realizar tanto para escritura como para lectura. Sólo se permite salvar un proceso existente en el sistema, si el usuario realizó una apertura del proceso para escritura.

3.7. Interacción con el Usuario

Como última área clave se desarrolla la parte visible del sistema por la cual se interactúa con los usuarios. Esta área marca la visión que los usuarios tienen sobre el sistema, por esta razón se hizo gran énfasis en ella.

3.7.1. Diseñador de Procesos

Para diseñar los procesos se buscó un diseñador de procesos código abierto para saber si había que desarrollarlo o se podía utilizar alguno existente. Luego de frustradas búsquedas se encontró el diseñador gráfico JaWE (Java workflow Editor).

El JaWE es un editor gráfico basado en Swing que maneja todos los conceptos de XPD. Como este diseñador no genera los procesos en formato jPDL, se debió realizar una transformación de uno a otro, aspecto que será profundizado más adelante. Como se mencionó anteriormente, ésta es una aplicación basada en Swing y por lo tanto de escritorio, para lograr la integración con la interfaz web se decidió realizar la invocación mediante Java Web Start.

En base a las necesidades del proyecto se optó por recortar las funcionalidades del JaWE, esta decisión se tomó, debido a algunas de las características ofrecidas por la herramienta no aplicaban a la realidad a resolver. Los cambios realizados sobre el código original fueron los de quitar estas características e incorporar la integración con sistemas externos. El código

agregado se encuentra en módulos independientes a los originales del JaWE, pero siguiendo las mismas características de implementación de éste. En la Figura 15 se puede visualizar el diseñador de procesos.

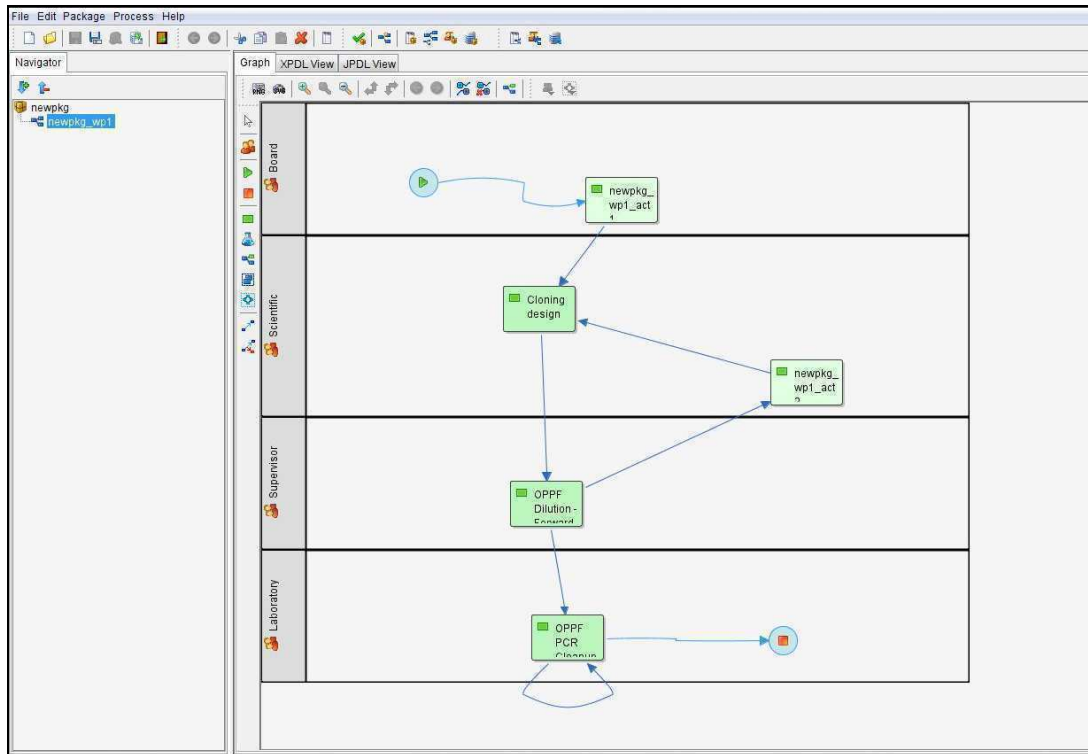


Figura 15. Interfaz de usuario del Diseñador de procesos.

Para iniciar este componente mediante Java Web Start se necesita el archivo JNLP, el cual es generado en forma dinámica por parte del servidor BioFlows. Esto es necesario debido a que toda invocación que realice este componente sobre el servidor BioFlows se efectúa a través de web services, utilizando para esta ejecución la información de sesión del usuario. En el archivo JNLP se agregan los parámetros que el usuario recibe al momento de realizar el login y la url del servidor a donde debe ejecutar los web services. Los datos de usuario son encriptados en el archivo por motivos de seguridad.

En las invocaciones que se realizan contra el servidor BioFlows mediante web services, se encriptan los datos de usuario utilizando un algoritmo de clave simétrica.

Conversión XPDL-jPDL

Como se indicó anteriormente, en jBPM es necesario utilizar el lenguaje jPDL para realizar la importación de los procesos, mientras que el JaWE es un editor basado en XPDL, por este motivo fue necesario hacer una conversión entre dichos lenguajes.

Durante la fase de estudio del estado del arte, no se encontraron estudios ni herramientas abiertas que permitan hacer esta transformación, sólo se encontraron algunas herramientas comerciales de diseño de workflows que permiten la exportación a jPDL.

Debido a que ambos lenguajes fueron diseñados como modelos de realidades similares, existen muchos conceptos comunes, a pesar de que están representados de forma distinta. Esto fue un factor determinante al estudiar la viabilidad de la conversión y permitió llevar a un mínimo la cantidad de información perdida tras la transformación. Debido al alcance del proyecto se consideró suficiente un subconjunto de los elementos existentes en el modelos XPDL, los mismos se presentan en al Tabla 3.

XPDL
Package
WorkflowProcess
DataField
Participant
Activity
Transition
Extended Attributes

Tabla 3. Elementos del modelo XPDL considerados al realizar la transformación.

Para los elementos presentados en la Tabla 3 existen 2 tipos de mapeos al modelo jPDL estos son, mapeo directo de los elementos de ambos modelos y mapeo indirecto. En el último caso se tienen consideraciones para poder realizar la conversión de un elemento de un modelo a otro. En la Tabla 4 se presenta que tipo de mapeo fue utilizado para cada uno de los elementos presentados en la Tabla 3. En el anexo Conversión XPDL – JPDL [61] se especifica en detalle la conversión realizada entre ambos modelos.

XPDL	Mapeo
Package	Indirecto
WorkflowProcess	Directo
DataField	Indirecto
Participant	Directo
Activity	Indirecto
Transition	Indirecto
Extended Attributes	Indirecto

Tabla 4. Tipo de mapeo utilizado para los elementos del modelo XPDL.

Una vez realizada la transformación, se integró al JaWE de manera de generar en tiempo real el jPDL, el cual se muestra en una pestaña del Diseñador de Procesos.

3.7.2. Interfaz Web

Es la aplicación web que representa el punto de entrada al sistema. La misma fue desarrollada utilizando JSF y RichFaces para lograr una interfaz intuitiva y atractiva al usuario. En la Figura 16 se presenta la página de inicio del sistema BioFlows.



Figura 16. Página de inicio del Sistema BioFlows.

Las funcionalidades presentadas por esta aplicación están divididas de acuerdo a los roles del sistema, presentando un submenú para cada rol, el cual estará visible sólo si el usuario conectado pertenece a algún grupo asignado a dicho rol. Adicionalmente, se presenta un submenú de usuario para cerrar la sesión o ir a la página de inicio.

Los submenús son los siguientes:

- Usuario: permite cerrar la sesión o ir a la página de inicio.
- Administración: se utiliza para realizar tareas de configuración del sistema.
- Diseño de Procesos: presenta la opción de iniciar el diseñador de procesos.
- Ejecución: ofrece las opciones de bandeja de entrada e inicio de procesos.
- Supervisión: muestra listados de procesos en curso e historial de ejecución.

En las Figuras que aparecen a continuación (Figura 17, 18, 19 y 20) el primer menú que se muestra es el mencionado anteriormente como el menú de usuario, en el mismo aparece el nombre del usuario que se encuentra logueado, para estas Figuras es el usuario board.

3.7.2.1. Administración



Figura 17. Menú del sistema BioFlows para el rol Administrador.

El menú de administración presenta tareas de configuración del sistema, apunta a un usuario de perfil técnico, al cual se le brinda la posibilidad de ajustar parámetros generales, de comunicación y de seguridad.

La configuración general apunta a la parametrización del sistema, permitiendo editar parámetros del sistema BioFlows y del LDAP.

La Administración de Roles presenta una pantalla donde el usuario puede, de forma sencilla y amigable, asociar los distintos grupos organizacionales almacenados en el LDAP, con los roles del sistema.

La Administración de Web Services presenta una pantalla donde se pueden elegir los web services configurables del sistema y crear una configuración. En dicha configuración se indica la ubicación, operación y los parámetros necesarios para la invocación del mismo.

3.7.2.2. Diseño de Procesos

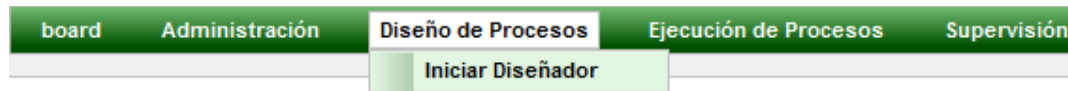


Figura 18. Menú del sistema BioFlows para el rol Diseñador de Procesos.

En este menú se presenta una única opción, la cual permite iniciar el Diseñador de Procesos mediante Java Web Start, a través del cual los usuario podrán definir nuevos procesos o modificar los ya existentes y realizar el deploy en el servidor en tiempo de ejecución para que aquellos usuario con privilegios de ejecución los puedan utilizar, como se explica en el punto siguiente.

3.7.2.3. Ejecución de Procesos



Figura 19. Menú del sistema BioFlows para el rol Ejecutor de Procesos.

Este menú presenta las actividades centrales del sistema, aquí los usuarios ejecutores pueden acceder a su bandeja de entrada, donde se muestran las tareas del usuario, tanto asignadas como candidatas. Una vez que un usuario ingrese a su bandeja de entrada se le mostraran las actividades en las que se encuentran los procesos que actualmente se están ejecutando diferenciándose en 2 grupos, estos son, las actividades de los procesos que no han sido asignados a ningún usuario y sobre los cuales él tiene acceso, o sea que el usuario pertenezca al rol para el cual ha sido definida la actividad en la que se encuentra el proceso, a estas actividades se las denominará candidatas, el usuario tiene la opción de poder asignárselas; por otra parte se presentan aquellas actividades de procesos que se encuentran asignadas al usuario, por lo explicado anteriormente, estas actividades el usuario debió asignárselas. El usuario cuenta con la funcionalidad de desasignarse las actividades que se tiene, en caso de desasignarse alguna de estas la misma volverá a la lista de las actividades candidatas del rol que tienen asociado la actividad en el proceso.

En el punto inicio de procesos, se listan los procesos del sistema, permitiendo iniciar cualquiera de ellos. En el caso que se inicie un nuevo proceso su primera actividad se listara en las actividades candidatas de los usuarios que pertenecen al rol que tiene asociado la actividad dentro del proceso.

3.7.2.4. Supervisión

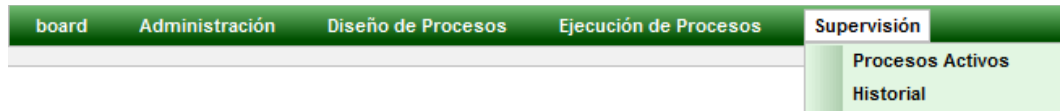


Figura 20. Menú del sistema BioFlows para el rol Supervisor.

El submenú de Supervisión apunta a usuarios con un perfil gerencial sobre el sistema. Permite obtener reportes (utilizando JasperReport) en formato pdf sobre los procesos activos y sobre la historia de los procesos ejecutados. En ambos reportes es posible especificar una serie de filtros para acotar los resultados listados.

3.8. Desarrollo del sistema

En este punto se indicará el tipo de desarrollo de cada uno de los componentes que conforman el sistema BioFlows presentados previamente, estos se presentan en la Tabla 5.

Componente	Tipo de Desarrollo
Motor de Workflow	Reutilización total
Usuario y Roles	Desarrollo
Seguridad y Auditoria	Desarrollo
Framework de integración con LIMS	Desarrollo
Repositorio de Procesos	Desarrollo
Diseñador de Procesos	Reutilización parcial
Interfaz Web	Desarrollo

Tabla 5. Tipo de desarrollo de Componentes.

Cabe aclarar que la reutilización parcial del diseño de procesos que se indica en la Tabla 5 se debe a, como se explico en la sección 3.7.1 el diseñador de procesos del sistema BioFlows se basa en la herramienta de código abierto JaWE, pero debido a las necesidades del sistema a desarrollar, la versión original de la herramienta no contaba con todas las funcionalidades requeridas, por lo cual hizo falta extender la misma para adaptarse 100% a las necesidades del sistema.

4. Diseño

En esta sección se presenta el diseño y las decisiones tomadas en la construcción del sistema BioFlows.

4.1. Arquitectura

Para el planteo de la arquitectura se puso gran énfasis en el uso de patrones de diseño y patrones arquitectónicos [44]. El uso de estos ayudó para el desarrollo simple y extensible de varios componentes del sistema.

El sistema BioFlows, está conformado por un conjunto de componentes, un grupo de interfaces de usuario y varios puntos de extensión para la comunicación con sistemas externos. En la Figura 21 se presentan los componentes que formar parte del sistema, así como también las distintas capas que este presenta.

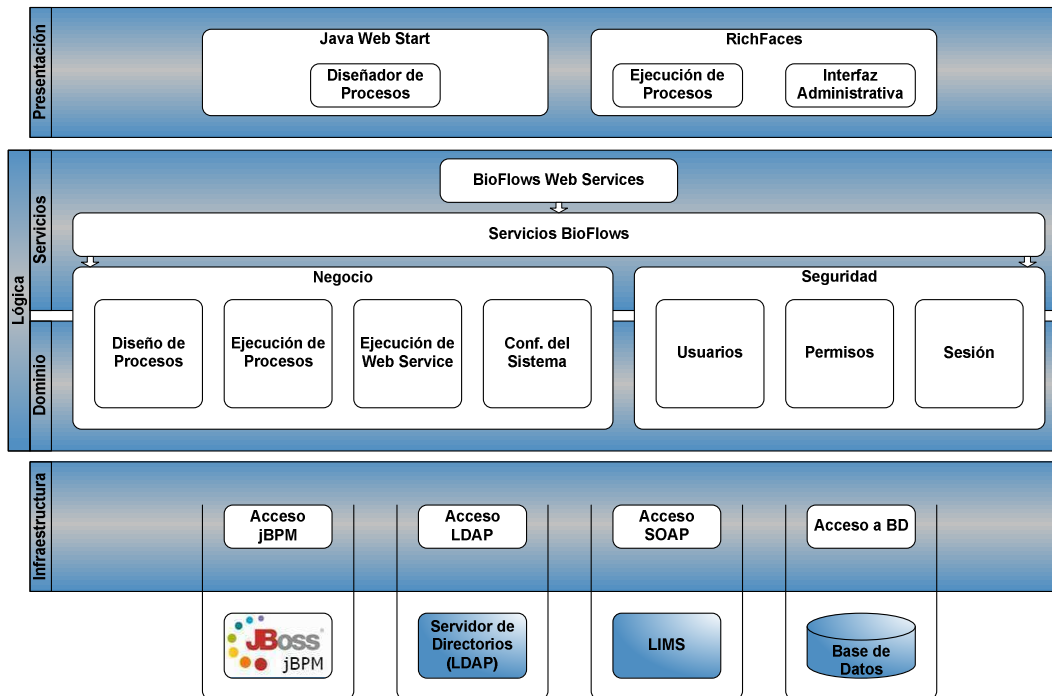


Figura 21. Arquitectura y componentes del sistema BioFlows.

4.1.1. Presentación

La capa de presentación tiene como objetivo el manejo de la interfaz gráfica de usuario, concebida para ser completamente web. Está compuesta por páginas RichFaces, módulos que encapsulan la lógica de los casos de uso y una aplicación desarrollada en Swing que es presentada en forma web mediante Java Web Start. Este componente está dividido en dos subcomponentes, el diseñador de procesos y la interfaz web que permite la ejecución de procesos y configuración del sistema.

El componente diseñador es una interfaz desarrollada cien por ciento en Swing, es presentada en forma web mediante Java Web Start. Este componente está basado en el editor gráfico de workflow (código abierto) JaWE, de ObjectWeb.

El componente de ejecución y configuración, está desarrollado utilizando páginas RichFaces.

4.1.2. Lógica

La capa lógica es donde residen los servicios del sistema y la implementación de estos. También es la encargada de controlar los permisos de usuario y llevar la auditoría de los pasos estos.

Para realizar un acceso más ordenado a la infraestructura del sistema, se decidió dividir esta capa en dos subcapas, las mismas son Servicios y Dominio.

4.1.2.1. Servicios

La capa de servicios es el punto de entrada a la capa lógica. En esta capa se distinguen dos componentes el primero se denomina servicios privados (Servicios BioFlows) y el segundo servicios públicos (BioFlows Web Services).

Los servicios públicos son un conjunto de operaciones, publicados como web service, los cuales son consumidos por el diseñador de procesos. Los servicios privados son aquellas operaciones utilizados por las páginas web. Ambos tipos de servicios no mantienen ningún tipo de estado.

Para hacer ordenado el acceso a la lógica, todos los servicios del sistema son implementados a través de Session Bean [21] del tipo StateLess. El mismo implementará el pattern Session Facade, será este quien verifique los permisos del usuario. Para resolver la problemática del negocio utilizará Business Delegate, buscando las entidades de dominio a través de un Service Locator.

La verificación de permisos se realiza cada vez que se intente ejecutar algún servicio sobre el sistema BioFlows, o se pretenda acceder a alguna página del mismo. Para cada recurso (página o servicio del sistema) del BioFlows que se

pretenda acceder se persistirá: usuario, fecha y hora de ejecución, recurso y resultado (Permitido, Denegado), como información de auditoría.

Cada operación de los servicios tiene su propia sesión sobre la base de datos. Se controlará la transaccionalidad a este nivel y se realizará commit o rollback de toda la operación. Todas las operaciones recibirán como parámetro el nombre de usuario y el identificador de sesión recibido al momento del log-in, para poder realizar la verificación de permisos.

4.1.2.2. Dominio

La capa de dominio le brinda a la capa de servicios una forma transparente de acceder a los distintos elementos que componen la infraestructura del negocio. Es también la encargada de manipular los distintos objetos de negocio (Procesos, Usuarios, etc.).

4.1.2.3. Componentes

Para tener una visión menos compleja del sistema BioFlows, se decidió realizar una primera división en dos grandes subcomponentes, los mismos son Seguridad y Negocio.

Seguridad

Realiza el control de seguridad del sistema. Entre sus responsabilidades se destacan, autenticación de usuarios, auditoría (de los pasos llevados por los distintos usuarios en el sistema), gestión de usuarios, roles y permisos. A partir de las responsabilidades antes descriptas se realiza una segunda descomposición en subcomponentes de este componente, dando como resultado los componentes de Usuario, Sesión y Permisos.

Usuarios

Este componente se encarga de brindar las funcionalidades para el manejo de mapeos de roles. También se encarga de autenticar usuarios utilizando LDAP.

Permisos

Este componente se encarga de comprobar permisos de usuario y hacer auditoría las acciones de éste.

Sesión

Este componente se encarga del mantenimiento de las sesiones de usuario en el sistema BioFlows. Creando nuevas sesiones al realizar el login y eliminándolas al realizar logout, así como también dando de baja las sesiones vencidas. En cada invocación a operaciones del sistema BioFlows, se chequea que el usuario que realizó dicha invocación tenga una sesión abierta en el sistema.

Negocio

Es donde se maneja la lógica del sistema y sus funcionalidades principales. Entre ellas se destacan, ABM de procesos, configuración de servicios de integración y configuración de parámetros del sistema. A partir de las responsabilidades antes descritas se realiza una segunda descomposición en subsistemas, dando como resultado los componentes Diseñador, JBPM, Web Service y Configuración.

Diseñador

Este componente se encarga de brindar las funcionalidades para el manejo del repositorio y manipulación de Procesos.

JBPM

Este componente se encarga de brindar las funcionalidades para la ejecución de procesos y tareas. Es el único encargado de interactuar con la biblioteca jBPM.

Configuración

Este componente es el encargado de registrar todas las configuraciones de parámetros (LDAP, etc.) que puede realizar el usuario Administrador, también obtener los valores registrados para dichos parámetros.

Web Service

Este componente es el encargado de configurar y ejecutar todos los web services de integración.

4.1.3. Infraestructura

La infraestructura contiene todos los módulos necesarios para realizar la comunicación. Está compuesto por cuatro módulos:

- El modulo de acceso a la base de datos es el responsable de realizar la comunicación con la base de datos, se utiliza el framework Hibernate para resolver este cometido.
- El modulo de acceso al LDAP es el responsable de realizar la comunicación con el LDAP, se utiliza el framework NetScape Directory SDK para este cometido.
- El modulo de acceso SOAP es el responsable de realizar la comunicación con los web services de integración, para la resolución de esta comunicación se utiliza Axis2.
- El modulo de acceso jBPM es el responsable de interactuar con el motor de workFlow (jBPM), se utiliza la API del mismo para este cometido.

4.2. Diagrama de Distribución

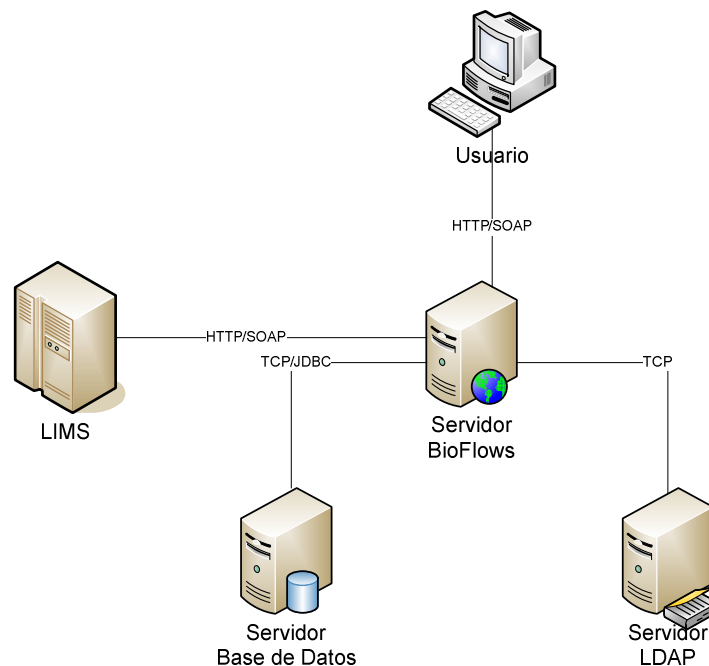


Figura 22. Diagrama de distribución del sistema BioFlows.

4.2.1. Nodo Usuario

En este nodo se encuentra el browser utilizado para acceder a las páginas web que provee el sistema y la aplicación Java Web Start.

4.2.2. Servidor BioFlows

En este nodo se encuentra el servidor web que permite la interacción de los actores con el sistema por medio de una interfaz Web, el servidor de aplicaciones que maneja la lógica del sistema, transaccionalidad, etc. y el motor de workflow. En este nodo reside la interfaz de servicios del sistema y la implementación de los mismos.

4.2.3. Servidor Base de Datos

En este nodo se encuentra el manejador de base de datos utilizado para almacenar la información.

4.2.4. Servidor LDAP

En este nodo se encuentra el repositorio de usuarios y grupos que se encuentran en el dominio de la aplicación.

4.2.5. LIMS

En este nodo se encuentra el LIMS con los distintos servicios publicados con los cuales se interactúa.

5. Caso de Estudio: PIMS

5.1. Introducción

El presente proyecto se realizó con la finalidad de complementar el sistema a utilizarse en el "Institut Pasteur de Montevideo" para la gestión de proteínas y trabajos realizados en el laboratorio. El sistema optado para cumplir con estos requisitos es el PIMS, por lo cual el mismo tiene un rol significativo en este proyecto. Debido a que en un inicio el objetivo del proyecto era complementar al sistema PIMS con funcionalidades de creación y ejecución de procesos, se pensaba desarrollar una solución completamente acoplada al mismo. Con la evolución de los objetivos del proyecto el PIMS pasó de ser el centro de atención en la aplicación, al caso de estudio para mostrar el correcto funcionamiento del sistema desarrollado y la obtención de los objetivos planteados.

5.2. Trabajo Realizado

El objetivo del sistema realizado es poder integrar un sistema de workflow a diferentes sistemas externos, siendo el PIMS uno de éstos. La integración entre BioFlows y el PIMS permite realizar consultas sobre distintos conceptos del último, en este caso puntal, brinda la posibilidad de obtener los protocolos y la información sobre las muestras de entrada y salida que maneja cada uno de éstos. Esta información es utilizada en la construcción de nuevos procesos de la organización a través del diseñador de workflows. Otra de las funcionalidades que permite esta integración, es la creación de experimentos durante la ejecución de un proceso, siendo estos experimentos instancias de los protocolos definidos en el mismo. Las funcionalidades que se están mencionando son específicas del desarrollo realizado para este caso de estudio.

Como se hizo mención en la sección 2.5.4.1 los protocolos y experimentos son conceptos dentro del PIMS. Los protocolos son la definición abstracta de los experimentos y especifican los lineamientos que deberán seguirse en su ejecución, así como la definición de las entradas y salidas de las muestras que lo componen. Por su parte los experimentos representan los trabajos realizados por un investigador en el laboratorio, la manipulación de experimentos consiste en su gestión (creación, edición y borrado), así como el cambio de estado, la edición de entradas y salidas, entre otras características.

5.2.1. Interfaz de Web Services

Como se mencionó anteriormente la integración con otros LIMS se realiza mediante web services, lo que hace que las funcionalidades con sistemas externos sean configurables y permita una fácil evolución. En el caso del PIMS la premisa de interactuar con una interfaz de web services fue imposible, ya que éste no cuenta con dicha funcionalidad. Por lo tanto, se desarrollaron web services que interactúan localmente con la API que provee el PIMS. Por cada uno de los seis servicios configurables en el sistema BioFlows, se implementó un web service sobre el PIMS. En la Figura 23 y la Tabla 6 se puede ver una lista y descripción de los mismos.

IPimsWebService	
+ getProtocols (String c1, String c2, String c3)	: List<Protocol>
+ getInputFromProtocol (String idProtocol)	: List<IOProtocol>
+ getOutputFromProtocol (String idProtocol)	: List<IOProtocol>
+ updateExperimentInput (String idProtocol, String name, String value, String expName, String userId)	: String
+ updateExperimentOutput (String idProtocol, String name, String value, String expName, String userId)	: String
+ finishExperiment (String idExperiment)	: String

Figura 21. Interfaz de web service desarrollada.

Servicio	Semántica
getProtocols	Obtiene una lista de protocolos.
getOutputFromProtocol	Obtiene las salidas de un protocolo.
getInputFromProtocol	Obtiene las entradas de un protocolo.
setTaskInputValue	Persiste el valor de una entrada para un experimento. Si el experimento no existe, lo crea y deja el estado del mismo en "en proceso". Si el usuario que está ejecutando el experimento no existe en el PIMS lo crea.
setTaskOutputValue	Persiste el valor de una salida para un experimento. Si el experimento no existe, lo crea y deja el estado del mismo en "en proceso". Si el usuario que está ejecutando el experimento no existe en el PIMS lo crea.
finishExperiment	Cambia el estado del experimento a "ok".

Tabla 6. Servicios desarrollados para el PIMS.

En la Figura 24 se presenta como fueron configurados los servicios del BioFlows respecto la interfaz de web services desarrollada para el PIMS.

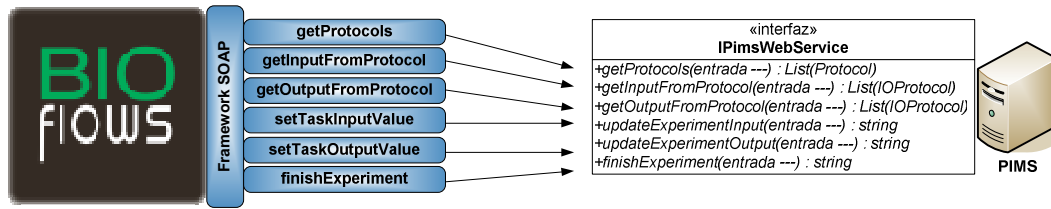


Figura 24. Configuración de web services de integración para el PIMS.

5.2.2. Ejemplo de integración

En este punto se presenta un ejemplo del funcionamiento de la integración entre el sistema BioFlows y el sistema de tipo LIMS, en este caso puntual el PIMS. Una vez que se cree un nuevo proceso, se deben incluir los roles que van a ser responsables de las actividades que se definan dentro del proceso. En la Figura 25 se definió que solamente va a participar el rol Laboratory, únicamente podrán manipular las actividades de los procesos inicializados aquellos usuarios que pertenezcan al rol que tiene asignada la actividad en el proceso. El primer punto de integración entre los sistema es el modelado de procesos de la organización, cuando se quiere insertar un nuevo protocolo se realiza la comunicación entre los sistemas mediante web services como se explico previamente, en la Figura 25 se muestra como es la interfaz que presenta a los protocolos que se recuperan del PIMS. Una vez que se selecciona uno de los protocolos desplegados (en el caso de la Figura 25 el OPPF PCR), se insertará en el proceso una nueva actividad.

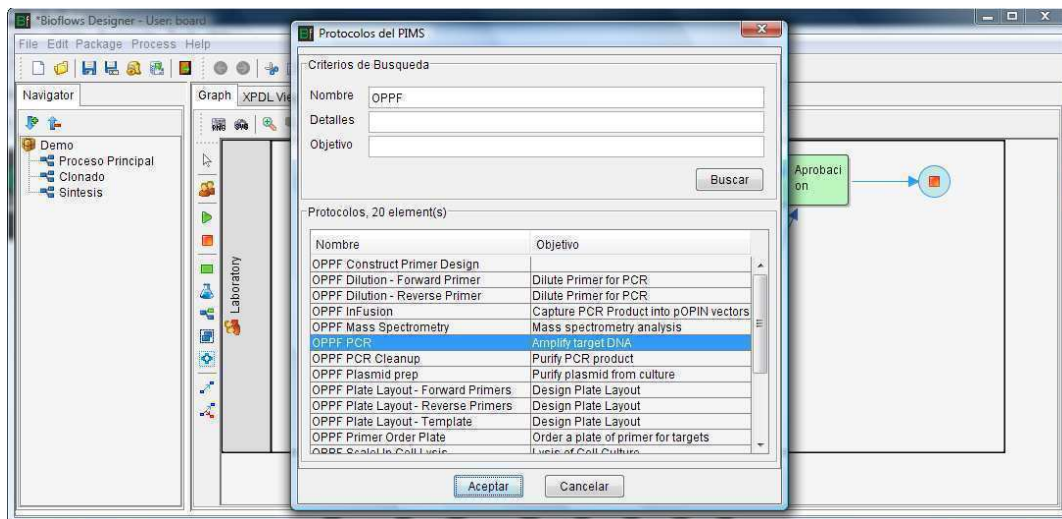


Figura 25. Obtención de protocolos del sistema PIMS.

Una vez que se haya insertado algún protocolo (obtenido del PIMS) se puede visualizar tanto los inputs y outputs que el mismo tiene definido en el PIMS. En la Figura 26 se puede visualizar los outputs del protocolo seleccionado en el Figura 25,

también se puede ver que los inputs se muestran en la pestaña continua a los outputs. La información referente tanto a los inputs y outputs es recuperada del PIMS de la misma forma que se recupero la información sobre los protocolos existentes en éste.

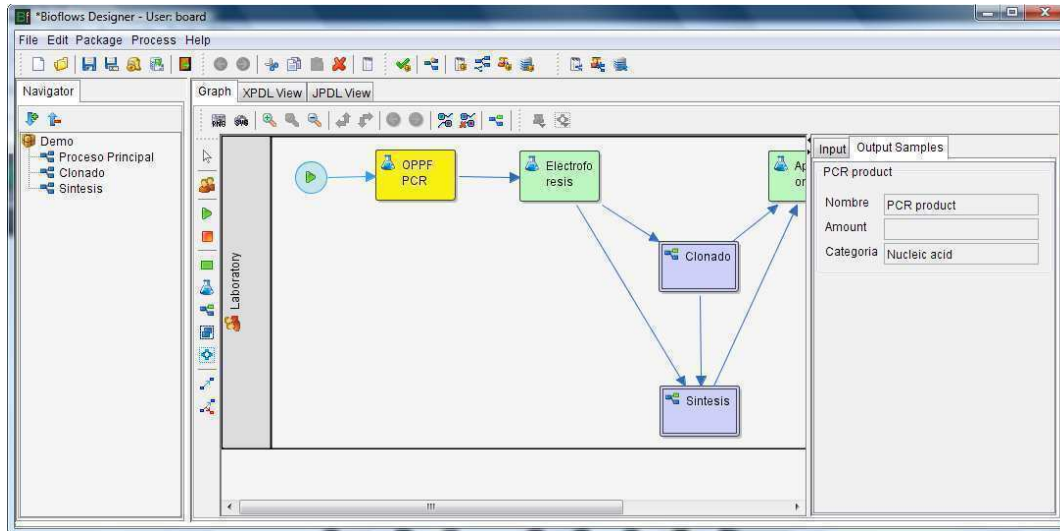


Figura 26. Visualización de inputs y outputs de un protocolo perteneciente al proceso.

Cuando se haya terminado de modelar uno o varios procesos, la forma de publicarlos para que los usuarios puedan ejecutarlos es utilizando la funcionalidad de deploy. La misma permite publicar los procesos al servidor del BioFlows de forma que los usuarios los puedan inicializar. La interfaz para esto se presenta en la Figura 27. De todos los procesos disponibles se puede elegir a cuales se les quiere realizar el deploy, como se puede visualizar en la Figura 27, estos serán dejados disponibles a los usuarios en el BioFlows.

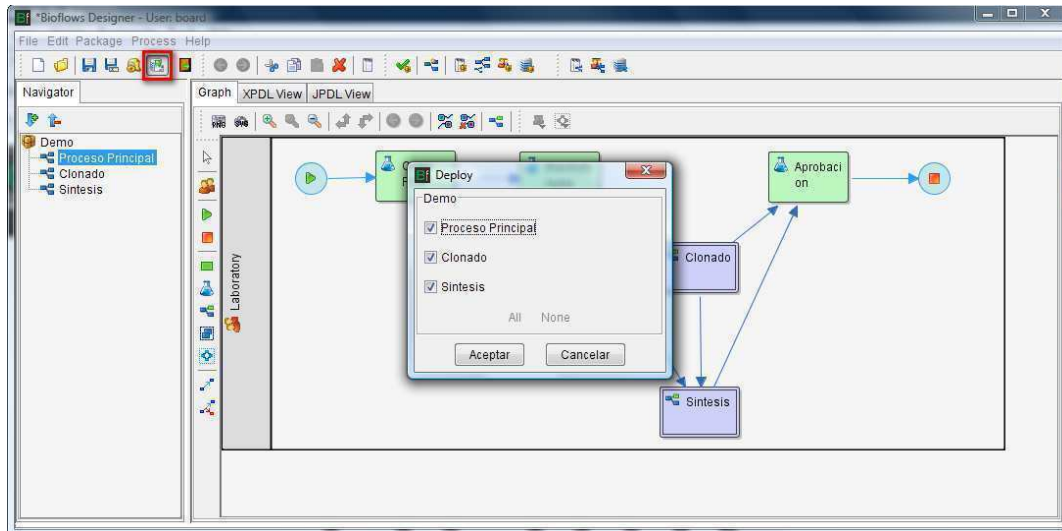


Figura 27. Deploy de procesos.

Después de publicar los procesos deseados en el BioFlows, como se indicó anteriormente, un usuario que pertenezca al rol ejecutor tiene la posibilidad de ingresar al menú **Inicio de Procesos** dentro del menú **Ejecución de Procesos** en el sistema, acá se podrá visualizar todos los procesos disponibles del sistema como se muestra en la Figura 28.

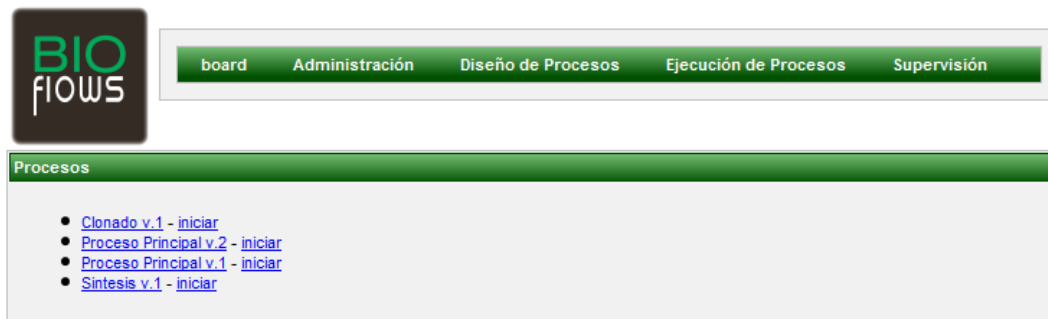


Figura 28. Procesos disponibles para inicializar.

Cuando el usuario se encuentre en el menú que se presentó en la Figura 28 cuenta con dos opciones, éstas son, visualizar la definición del proceso dando click sobre le nombre del mismo, el resultado se muestra en la Figura 29, o inicializar el proyecto a través de la opción **iniciar** que se encuentra al lado del nombre del proceso deseado.

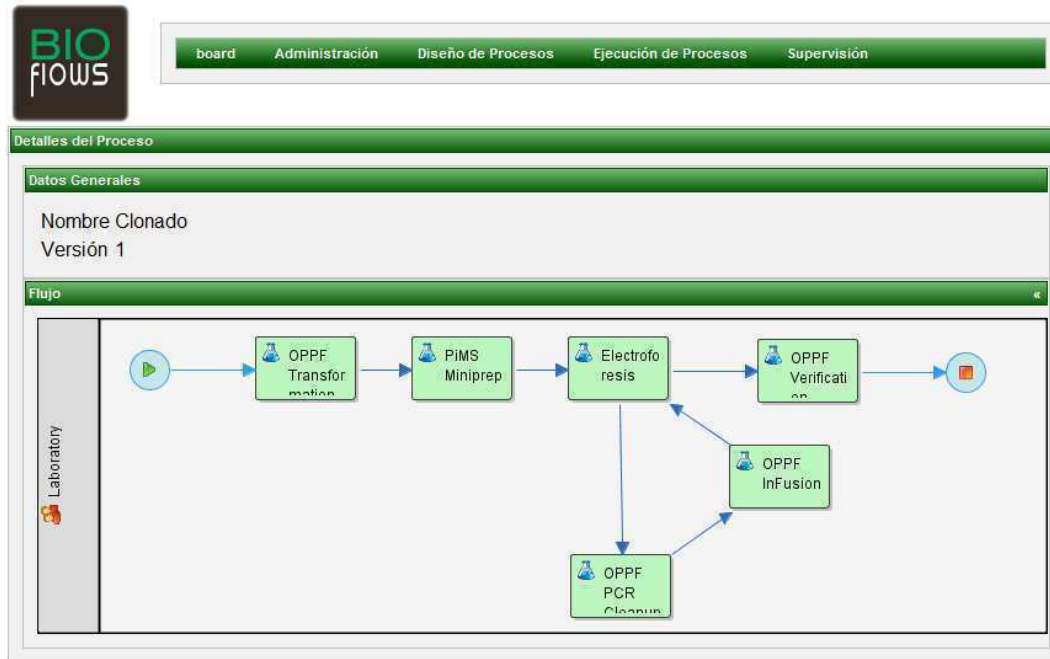


Figura 29. Visualización de proceso deployado.

Otra de las alternativas que tiene un usuario con el rol ejecutor es la de entrar al menú Bandeja de Entrada dentro del menú Ejecución de Procesos en el sistema, en la Figura 30 se visualizan cuales son las actividades candidatas que tiene el usuario para asignarse. Cabe aclarar que el usuario solamente podrá ver aquellas actividades que estén definidas en el proceso dentro de un rol al cual él pertenece. Dando click sobre alguno de los nombre de las actividades candidatas el usuario se asignara ésta.

The screenshot shows the BioFlows interface with a 'lab' user selected in the 'Ejecución de Procesos' menu. Below the navigation bar, there are two tables: 'Experimentos Asignados' and 'Experimentos Candidatos'. The 'Experimentos Candidatos' table contains one entry:

Tarea	Descripción	Comienzo	Vencimiento	Proceso
OPPF PCR	Amplify target DNA	2008-05-25 14:30:51.067		Proceso Principal

Figura 30. Visualización de actividades candidatas para el usuario lab.

En la Figura 31 se presentan aquellas actividades que el usuario se asignó, como se explicó anteriormente. En dicha figura se puede visualizar la opción de **desasignar**, para elegir dicha funcionalidad se deberá realizar click en el link

desasignar que tiene la actividad, y la misma volverá a las actividades candidatas. Se visualizará la misma en forma análoga a la Figura 30.



The screenshot displays the BioFlows interface for a user named 'lab'. At the top left is the 'BIO FLOWS' logo. To the right is a green header bar with the text 'lab Ejecución de Procesos'. Below this, there are two main sections:

Experimentos Asignados

Tarea	Descripción	Comienzo	Vencimiento	Proceso	
Electroforesis		2008-05-25 14:36:34.241		Proceso Principal	desasignar

Experimentos Candidatos


Tarea	Descripción	Comienzo	Vencimiento	Proceso
-------	-------------	----------	-------------	---------

Figura 31. Visualización de actividades asignadas al usuario lab.

Otra funcionalidad sobre las actividades que un usuario tiene asignadas, es la de modificar sus propiedades, ya sean los valores de las variables globales, los inputs, los outputs e incluso finalizarla, navegando a otra de las actividad disponibles a través de las transiciones que se definieron en el proceso al que pertenece.

Para abrir una actividad se debe dar clic sobre el nombre de la actividad asignada, de esta manera se mostrara la información de la misma como se muestra en la Figura 32. Una vez que se encuentre dentro de las propiedades de la actividad se puede modificar los valores de variables globales, inputs y outputs, los mismos aparecen en las secciones Variables Globales, Entradas y Salidas respectivamente. En la sección de Transiciones se indica a que otras actividades se puede navegar, en caso que sea la actividad final se podrá navegar al fin del proceso.

Cuando se salven los nuevos valores para los inputs y/o outputs se tiene otra de las interacciones con el sistema externo PIMS. Al salvar cada uno de estos se actualizará la información existente en el experimento creado en el PIMS, en caso que sea la primera vez que se salva esta información se crea el experimento para poder actualizar la información. Cuando se cambie de actividad se actualizará la información del experimento creado previamente en el PIMS cambiando su estado a OK, en caso que el mismo todavía no existiera primero se crea el mismo.



lab Ejecución de Procesos

Datos Generales

Nombre Electroforesis
 Descripción
 Fecha de Inicio 2008-05-25 14:36:34.241
 Fecha Limite

Variables Globales

Entradas

Buffer TAE
 Bromuro de Etidio

Salidas

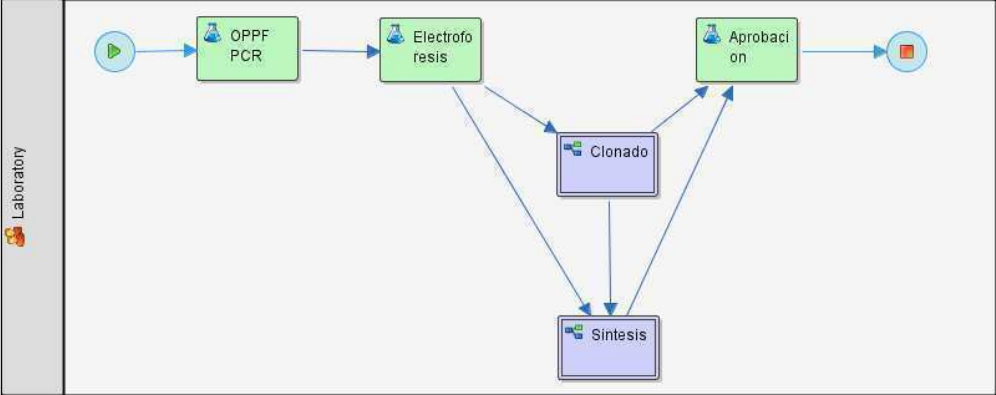
Buffer

Transiciones

- [134 Principal_tra3_destino:Sintesis](#)
- [133 Principal_tra2_destino:Clonado](#)

Flujo

Laboratory



```

graph LR
    Start((Start)) --> OPPF[OPPF PCR]
    OPPF --> Electroforesis[Electroforesis]
    Electroforesis --> Clonado[Clonado]
    Electroforesis --> Sintesis[Sintesis]
    Clonado --> Aprobacion[Aprobacion]
    Sintesis --> Aprobacion
    Aprobacion --> End((End))
    
```

Figura 32. Visualización de la actividad Electroforesis asignada al usuario lab.

En la Figura 33 se muestra la información general del experimento que fue creado en el PIMS para la actividad Electroforesis presentada en la Figura 32.

The screenshot displays the PIMS web interface for editing experiment details. The page title is "Experiment: 200". The navigation bar includes "Home", "Log out administrator", "Target", "Experiment", "People & Places", "History", and "Help". A "Perspective: standard" dropdown is visible in the top right. On the left, a sidebar lists "Click for Experiments:" with links for "Purification", "Cloning", "DNA purification", and "PCR". The main content area has tabs for "Details", "Parameters", "Inputs", "Output Samples", and "Files". The "Details" tab is active, showing the following fields:

- Name: 200
- Type: [Purification](#)
- Status: In_process
- Protocol: [Electroforesis](#)
- Start date: 25/05/2008
- End date: 25/05/2008
- Construct / Target: [none]
- Scientist: [lab Add/Remove](#)
- Details: (empty text area)

A "Save" button is located at the bottom of the form.

Figura 33. Visualización de las propiedades generales del experimento generado en el PIMS para la actividad de la Figura 32.

En la Figura 34 se visualiza los valores de los inputs del experimento creado para la actividad del proceso, como se explicó anteriormente. Se puede ver que los valores para los inputs en la Figura 32 y 34 son iguales.

The screenshot shows the PIMS web interface for 'Experiment: 200'. The top navigation bar includes 'Home', 'Log out administrator', 'Target', 'Experiment', 'People & Places', 'History', and 'Help'. A 'Perspective: standard' dropdown is visible in the top right. On the left, a sidebar lists 'Click for Experiments:' with links for 'Purification', 'Cloning', 'DNA purification', and 'PCR'. The main content area has tabs for 'Details', 'Parameters', 'Inputs', 'Output Samples', and 'Files'. The 'Inputs' tab is active, displaying a table with the following data:

Role	Sample / Stock	Amount
Buffer TAE	Search/Add samples	1.0
Bromuro de Etidio	Search/Add samples	2.0

Below the table is a 'Save' button.

Figura 34. Visualización de los inputs para el experimento presentado en la Figura 33.

5.3. Resultados Obtenidos

Debido a los objetivos del proyecto y como se está haciendo referencia en este documento, el BioFlows queda completamente desacoplado del PIMS, la integración mediante web services configurables permite cambiar el sistema a integrar o cambiar las funcionalidades que se quieren ejecutar. Si el sistema externo ya contara con web services para ejecutar la funcionalidad buscada, simplemente se configura el servicio correspondiente en BioFlows para que se cumpla la interfaz deseada. Por el contrario, si el sistema externo no cuenta con dicha interfaz, es necesario modificar o reescribir los web services externos para soportar las nuevas funcionalidades. De esta forma se puede lograr la integración con diversos sistemas externos (LIMS) o la modificación de las funcionalidades que se estén utilizando del sistema que se encuentre configurado. Por ejemplo si en un futuro otra área del "Institut Pasteur de Montevideo" tuviera la necesidad de trabajar sobre targets del PIMS en vez de protocolos, para lo cual se reescribirían los web services, de modo que trabajen sobre los targets en vez de hacerlo sobre los protocolos.

Con la solución planteada a la problemática de integración se ha obtenido una gran flexibilidad para poder utilizar funcionalidades de sistemas externos de tipo LIMS. En el marco de este proyecto se contó con la problemática que la arquitectura del PIMS no era ideal para verificar esta integración debido a que éste carece de una interfaz de web services, la misma fue desarrollada desde cero como un módulo externo al sistema, lo cual puede condicionar el diseño de la solución para satisfacer esta realidad particular. Pese a lo explicado anteriormente, se considera que fue posible verificar la adaptabilidad del framework de integración desarrollado y por ende del sistema BioFlows en sí mismo.

6. Conclusiones

Este proyecto surge vinculado al PIMS, un sistema de gestión de información de laboratorios (LIMS), que permite mantener control de las muestras y experimentos que se llevan a cabo en los laboratorios. Al inicio del mismo, los objetivos se centraban en complementar las carencias del PIMS en cuanto a la utilización de flujos de ejecución de experimentos (procesos). Se buscaba una solución que incluía al PIMS como componente central.

Como complemento al PIMS se planteó la integración de una herramienta de workflow que fuera capaz de especificar y ejecutar un flujo de ejecución de experimentos. Esto produjo cierta incertidumbre en lo referente a las tecnologías y desafíos técnicos que se debían afrontar, lo cual condujo a una etapa importante de investigación sobre los aspectos mencionados, para evaluar la viabilidad del proyecto, estándares a adoptar y tecnologías para asistir el desarrollo.

Una vez culminada la etapa de análisis, se realizó un primer prototipo para mitigar algunos riesgos, principalmente la conversión de XPDL a jPDL y el deploy de procesos sobre un servidor jBPM. En base al análisis y el prototipo exitoso, se definió la solución a desarrollar, la cual no debía tener al PIMS como un componente del sistema, sino que debía interactuar con él. Esta interacción debía ser configurable para permitir la integración con otro sistema, logrando una solución genérica.

Como resultado de este proceso, se desarrolló el sistema BioFlows, una Plataforma para la Gestión de Procesos en Sistemas Biológicos. Este sistema está desarrollado en base a cinco pilares básicos: adaptabilidad, estandarización, usabilidad, integración con directorio de usuarios (LDAP) e integración con sistemas LIMS.

Adaptabilidad

El sistema está diseñado para adaptarse a distintas organizaciones. Esto se logra principalmente, mediante la integración con sistemas externos y la utilización de estándares.

Estandarización

El sistema contempla estándares en diversos puntos: LDAP en el acceso a información de usuarios, web services en la integración con otros sistemas y XPDL en la definición de procesos. Estos puntos promueven la adaptabilidad del sistema.

Usabilidad

En diversos puntos del sistema se tomaron decisiones tomando en cuenta la usabilidad, principalmente al considerar el acceso web, las tecnologías utilizadas en el desarrollo de la interfaz web y en la elección del diseñador de procesos.

Integración con directorio de usuarios (LDAP)

El sistema cuenta con la capacidad de integrarse fácilmente a una organización que ya utiliza un sistema de directorio con acceso a través de LDAP. En el caso de que la organización no contara con este servicio, LDAP es el protocolo de gestión de usuarios más utilizado, por lo que existen diversas aplicaciones, tanto comerciales como gratuitas, capaces de cumplir con este objetivo.

Integración con sistemas LIMS

El sistema está diseñado para comunicarse con sistemas externos mediante SOAP, siendo éste un protocolo ampliamente utilizado para la integración de sistemas. Este diseño permite la comunicación con el sistema LIMS mediante un protocolo estándar. La configuración de los web services a consumir se realiza de una forma sencilla mediante la interfaz web.

El desarrollo basado en las áreas mencionadas dio como resultado un sistema capaz de integrarse con el PIMS. Esta integración demostró tanto el cumplimiento de los objetivos iniciales como la adaptabilidad del sistema.

Se considera que tanto la solución alcanzada como el desarrollo del proyecto cumplieron los objetivos y expectativas de los integrantes del grupo, permitiendo aplicar y enriquecer los conocimientos adquiridos académica y profesionalmente.

7. Trabajo a Futuro

En esta sección se presentan sugerencias de posibles trabajos a realizar. Los puntos aquí planteados pueden considerarse como una guía para la evolución del Sistema BioFlows.

7.1. Diseñador de Procesos Web

El diseñador de procesos es una herramienta de escritorio que funciona sobre la máquina virtual Java, esto crea restricciones a nivel de comunicación con el servidor y requerimientos de software y hardware en el puesto cliente.

En este punto se propone la implementación de la herramienta de diseño en un ambiente web, con las ventajas que éste presenta referentes a los puntos ya explicados, sin perder las facilidades que presenta la herramienta de escritorio (drag & drop, dibujo de transiciones, etc.). Se sugiere utilizar como representación gráfica la notación estándar BPMN.

7.2. Conversión total de XPDL 2.0 a jPDL

Debido a que la herramienta de diseño utilizada soporta la versión de XPDL 1.0 y los nuevos agregados de la versión 2.0 no se consideraron relevantes para la definición actual del sistema, se optó por no extender esta herramienta para darle soporte a la nueva versión.

Se propone la extensión del Diseñador de Procesos para soportar la nueva versión de XPDL y una conversión total de un lenguaje a otro, de manera de aumentar la flexibilidad de la herramienta para representar procesos complejos de distintas organizaciones.

7.3. Extensión del Framework de ejecución de Web Services.

Debido a como está desarrollado el framework, no se permite el uso de servicios de integración con un sistema LIMS cuando alguno de los parámetros de entrada no es de un tipo básico (string, int, etc). Si bien no fue necesario este tipo de operaciones en el contexto del proyecto, son consideradas de utilidad para hacer del sistema, una aplicación más integrable.

7.4. Seguridad

Si bien los web services que ejecuta el diseñador de procesos, encriptan los datos del usuario a través de una clave simétrica, no se cuenta con un mecanismo de seguridad a nivel global del servicio. En este punto se propone una extensión del sistema para utilizar HTTPS o WS Security [57] como mecanismo de seguridad en la comunicación mediante web services y HTTPS en la interfaz web del sistema.

7.5. Gestión de Roles

El sistema permite el mapeo de roles a grupos organizacionales definidos en el LDAP, pero estos roles están definidos de una forma estática, lo cual indica que no se cuenta con la capacidad de crear o eliminar roles.

Para que esta funcionalidad estuviera completa se requeriría la gestión de permisos sobre los roles. En el sistema actual esta configuración es dinámica, pero no existe forma de cambiarla sin acceso directo a la base de datos.

7.6. Permisos de ejecución de procesos

En el sistema actual existe un esquema de seguridad para la manipulación de procesos, pero no se manejan restricciones en la ejecución de los mismos. Si bien cada tarea está restringida a grupo en particular, no hay ninguna limitación para iniciar la ejecución del proceso.

8. Referencias

- [1] OMG
<http://www.omg.org/> Septiembre 2007
- [2] BPMN
<http://www.bpmn.org/> Septiembre 2007
- [3] BPMN
http://www.omg.org/news/meetings/workshops/soa-bpm-mda-2006/00-T4_White.pdf Septiembre 2007
- [4] BPMI
<http://www.bpmi.org/> Marzo 2008
- [5] XPDL
<http://www.wfmc.org/standards/xpdl.htm> Enero 2008
- [6] WFMC
<http://www.wfmc.org/> Septiembre 2007
- [7] jPDL
<http://docs.jboss.com/jbpm/v3/userguide/> Marzo 2007
- [8] BPM
http://en.wikipedia.org/wiki/Business_Process_Management Septiembre 2007
- [9] BPEL
<http://en.wikipedia.org/wiki/BPEL> Abril 2008
- [10] BAM
http://en.wikipedia.org/wiki/Business_Activity_Monitoring Abril 2008
- [11] ESB
http://en.wikipedia.org/wiki/Enterprise_service_bus Mayo 2008
- [12] Workflow
<http://www.e-workflow.org/> Agosto 2007
- [13] jBPM
<http://www.jboss.com/products/jbpm> Marzo 2008
- [14] OS workflow

<http://www.opensymphony.com/osworkflow/> Septiembre 2007

[15] Enhydra

<http://www.enhydra.org/workflow/> Septiembre 2007

[16] Bonita

<http://bonita.objectweb.org/> Septiembre 2007

[17] JaWE

<http://www.enhydra.org/workflow/jawe/index.html> Marzo 2008

[18] Taverna

<http://taverna.sourceforge.net/> Marzo 2008

[19] myGrid

<http://www.mygrid.org.uk/> Abril 2008

[20] Java

<http://java.sun.com> Enero 2008

[21] J2EE

Advanced J2EE Platform Development: Applying Integration Tier Patterns

Ed: Prentice Hall, Inc. (2003) ISBN: 9780130449122

Enero 2008

[22] J2SE, J2EE

<http://www.javaworld.com/> Marzo 2008

[23] Java Web Start

<http://java.sun.com/products/javawebstart> Marzo 2008

[24] JNLP

<http://es.wikipedia.org/wiki/JNLP> Marzo 2008

[25] JBoss

<http://www.jboss.org> Febrero 2008

[26] Tomcat

<http://tomcat.apache.org/> Febrero 2008

[27] SOAP

<http://www.w3.org/TR/soap> Marzo 2008

[28] WSDL

<http://www.w3.org/TR/wsd/> Marzo 2008

[29] XML

<http://www.w3.org/XML> Marzo 2008

[30] Axis2

<http://ws.apache.org/axis2> Diciembre 2007

[31] Hibernate

<http://www.hibernate.org> Diciembre 2007

[32] NetScape Directory SDK

<http://www.mozilla.org/directory/javasdk.html> Diciembre 2007

[33] LDAP

http://www.javahispano.org/contenidos/es/acceso_a_ldap_mediante_netscape_directory_sdk/ Diciembre 2007

[34] LDAP

<http://dns.bdat.net/documentos/ldap/> Diciembre 2007

[35] PostgreSQL

<http://www.postgresql.org> Agosto 2007

[36] Eclipse

<http://www.eclipse.org/> Agosto 2007

[37] MyEclipse

<http://www.myeclipseide.com/> Enero 2008

[38] JBoss RichFaces

<http://labs.jboss.com/jbossrichfaces/> Marzo 2008

[39] JBoss RichFaces

<http://livedemo.exadel.com/richfaces-demo/index.jsp> Marzo 2008

[40] JavaServer Faces

Mastering JavaServer Faces

Bill Dudley, Jonathan Lehr, Bill Willis, LeRoy Mattingly

Ed: Wiley, Inc. (2004) - ISBN: 9780471462071

Marzo 2008

[41] Swing

<http://java.sun.com/javase/6/docs/technotes/guides/swing/index.html>

Marzo 2008

[42] IReport

http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/ireport/ Marzo 2008

[43] JasperReport

http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/jasperreports/ Marzo 2008

[44] UML y Patrones

Una introducción al análisis y diseño orientado a objetos y al proceso unificado. LARMAN, CRAIG. Segunda Edición.

Ed: Prentice Hall, Inc. (2003) - ISBN: 8420534382.

Septiembre 2007

[45] Proyecto Batuta

<http://www.fing.edu.uy/~pgsoasem/> Abril 2008

[46] Interacción persona computadora

<http://www.fing.edu.uy/inco/cursos/inpercom/> Diciembre 2007

[47] Taller de Sistema de Información

<http://www.fing.edu.uy/inco/cursos/tsi/TSI2/> Septiembre 2007

[48] LIMS

http://en.wikipedia.org/wiki/Laboratory_Information_Management_System

Agosto 2007

[49] PIMS

<http://www.pims-lims.org/> Marzo 2008

[50] National Center for Biotechnology Information

http://eutils.ncbi.nlm.nih.gov/entrez/query/static/esoap_help.html Marzo 2008

[51] European Bioinformatics Institute (EBI)

<http://xml.ddbj.nig.ac.jp/wsd/index.jsp> Marzo 2008

[52] IT Innovation

<http://www.it-innovation.soton.ac.uk/> Marzo 2008

[53] The School of Computer Science

<http://www.cs.ncl.ac.uk/> Marzo 2008

[54] Newcastle Centre for Life

<http://www.life.org.uk./> Marzo 2008

[55] School of Computer Science at the University of Manchester

<http://www.cs.man.ac.uk/> Marzo 2008

[56] Nottingham University Mixed Reality Lab

<http://www.mrl.nott.ac.uk/> Marzo 2008

[57] Web Services Security

<http://www.ibm.com/developerworks/library/ws-secure/> Marzo 2008

9. Anexos

[58] Descripción de la Arquitectura.

[Anexo - Descripcion de la Arquitectura.doc](#)

[59] Manual de Usuario.

[Anexo - Manual de Usuario.doc](#)

[60] Interacción.

[Anexo - Interaccion.doc](#)

[61] Conversión jPDL-XPDL.

[Anexo - Conversion XPDL-JPDL.doc](#)