

*Instituto de Computación,  
Facultad de Ingeniería,  
UdelaR*

---

**2014**

**ANEXOS INFORME -  
Construcción de Web  
Warehouse enfocados en la  
Calidad con BPMS**

Informe final de Proyecto de Grado 2014

Estudiantes:

Miguel Angel Mendiola

Fernando Alonso

Tutores:

Andrea Delgado

Adriana Marotta

05/2015 Montevideo, Uruguay

**Anexo A: Marco Teórico..... 3**

Data Warehouse..... 3  
Web Warehouse..... 9  
Procesos de Negocio. .... 14

**Anexo B: Manual de usuario..... 23**

Descripción de Proceso de Configuración ..... 23  
Descripción de Proceso de Carga ..... 38

**Anexo C: Manual técnico..... 42**

Modelo de Dominio..... 42  
Modelo de Base de Metadatos ..... 42  
Resumen ..... 44  
Especificación de funcionalidad y uso ..... 46  
Descripción de Proceso de Configuración ..... 46  
    Subproceso “Define Expected Schema and Mapping” ..... 46  
    Subproceso “Define Integrated Schema and Mapping” ..... 47  
    Subproceso “Define DW Schema and Mapping” ..... 47  
Descripción de Proceso de Carga ..... 51

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

## **Anexo A: Marco Teórico.**

A continuación en este anexo se presentarán los conceptos teóricos en los que se basa el proyecto de grado con una mayor profundidad.

### ***Data Warehouse.***

El Data Warehouse (abreviamos DW) es una disciplina dentro del Software que incluye herramientas y metodologías que se utilizan para la administración de información con el fin de analizarla y mejorar la gestión de cualquier organización.

Los objetivos del DW radican en automatizar los procesos de recopilación y generación de datos para que los usuarios puedan utilizarla. Tiene como principales actividades extraer los datos provenientes de sus fuentes, transformar dichos datos según las necesidades, integrarlos para generar más valor, refinarlos y centralizar la ubicación de todos esos datos útiles que la organización genera en su actividad diaria de negocio. A todo esto, se agregan los datos externos que provienen de interacciones que pueda tener la organización con terceros, como pueden ser proveedores, clientes o de otro tipo.

Dichas herramientas permiten el acceso y la manipulación de los datos, tanto actuales pero más que nada históricos, a través de un análisis en múltiples variables con el objetivo final de brindar información útil para la toma de decisiones en función de dichos datos.

El primer cuestionamiento que nos surge es por qué no realizar este análisis con las bases de datos tradicionales que hoy en día está presente en cualquier Sistema de Gestión de una organización. La respuesta es clara: los modelos de datos y las estrategias de diseño utilizadas por las bases de datos tradicionales apuntan a lo operacional o transaccional, esto implica que no sean adecuadas para soportar grandes cantidades de datos con consultas extremadamente costosas sobre datos que por lo menos, sean del meses anteriores. Por lo que antes estas necesidades surgen nuevas técnicas y estrategias de diseño para la construcción de DW.

Entonces, a modo de resumen los objetivos a los que apunta el DW son: la accesibilidad a los datos por parte de las personas de la organización, consistencia de los mismos entre los distintos sectores de una organización, ayudar a la toma de decisiones de los altos mandos, flexibilidad en la generación de datos, controlar el acceso a los mismos. [\[3\]](#)

Los principales servicios que un DW nos puede proveer son: administrar grandes volúmenes de datos, almacenamiento de datos históricos, condensa y agrega información de valor, integra y asocia datos provenientes de distintas fuentes.

En las organizaciones, el análisis para la toma de decisiones presenta distintos obstáculos, uno de ellos es la accesibilidad de las personas a la información, el tiempo de acceso a la misma, y la integración y calidad de los datos, por lo que un DW es de vital importancia si la organización pretende optimizar sus recursos, entender de donde viene la misma y a qué dirección se dirige. [\[3\]](#)

Por lo dicho, cualquier dato puede ser información vital, dependiendo del tipo de decisión a tomar, la persona que lo va a utilizar y de la calidad del mismo.

## ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

Como mencionábamos anteriormente, los sistemas tradicionales que utilizan las empresas (empresas privadas o públicas) hoy en día, como por ejemplos los ERP (Enterprise Resource Planning), son transaccionales, esto es, están orientados a las operaciones, asociados a procesos productivos de la organización y procesan grandes cantidades de transacciones, no son útiles para ser utilizados como medios de consulta masiva en donde se extraen millones de datos, por lo que no puede resolver las necesidades antes mencionadas. En la Tabla 6 mostramos algunas diferencias importantes entre ambos Sistemas, según distintos aspectos.

Tabla 1 Comparativa entre BD Operacionales y los DW por aspectos [3]

Aspect	Operational databases	Data warehouses
User type	Operators, office employees	Managers, high-ranking executives
Usage	Predictable, repetitive	Ad hoc, nonstructured
Data content	Current, detailed data	Historical, summarized data
Data organization	According to operational needs	According to the analysis problem
Data structures	Optimized for small transactions	Optimized for complex queries
Access frequency	High	From medium to low
Access type	Read, update, delete, insert	Read, append only
Number of records per access	Few	Many
Response time	Short	Can be long
Concurrency level	High	Low
Lock utilization	Necessary	Not necessary
Update frequency	High	None
Data redundancy	Low (normalized tables)	High (unnormalized tables)
Data modeling	ER model	Multidimensional model
Modeling and implementation	Entire system	Incremental

En contraparte, los sistemas de DW, atacan esta problemática, generando información para la toma de decisiones tomando como referencia algunas premisas: construir información a partir de datos objetivos, integrar diferentes fuentes de datos y ofrecer al usuario final mecanismos flexibles para el acceso a los mismos.

La información que gestionan los DW, refieren a los temas principales de la organización. Proviene de diferentes áreas de la misma y son de distintos tipos. Los aspectos a resolver en cuanto a la integración son la unificación de conceptos y la construcción de los datos integrados a partir de los datos fuentes. Como es de interés manejar los datos históricamente, se deben mantener con su referencia temporal y no ser volátiles, los mismos deben ser estables como para permitir el análisis de larga duración, para esto es necesario realizar una hostilización y planificación de carga.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

## Arquitectura de los Sistemas de Data Warehouse

Analizando la arquitectura típica de un DW como podemos ver en la Figura 1, en la capa más inferior tenemos lo que son las distintas fuentes de datos que podemos tener en un DW convencional, de tipo empresarial pueden ser:

- Datos históricos.
- Archivos de texto que contienen datos.
- Las (o la) bases de datos relacionales.
- Información geográfica almacenada de alguna manera, etc.

Una vez que definimos los distintos inputs de datos, utilizamos herramientas de ETL (Extracción, Transformación y Carga) que nos permiten extraer y explotar los datos de las fuentes antes mencionadas. Una vez que realizamos la extracción de los datos, los podemos alojar en un medio común, realizar las transformaciones necesarias para integrarlos, controlar la calidad de los mismos y cargar nuestro DW [\[3\]](#).

El DW final es el repositorio de los datos ya procesados, y organizados en tablas que pueden ser tablas de dimensiones (contienen los datos referentes a cada dimensión del DW) y las tablas de hechos (son registros que mencionan un hecho en particular e importante para nuestra organización que queremos estudiar cómo puede ser una venta en un comercio, una reserva en una agencia de viajes, la expedición de un documento en un organismo estatal o la realización de una multa en un ente municipal, etc.). Además, el DW contiene distintas medidas definidas por los técnicos y especialistas del negocio, que permiten realizar operaciones o estudios sobre los cruzamientos de dichas tablas, se pueden ver como un cálculo de algún tipo sobre los hechos, que de alguna manera nos interesa registrar.

Una vez que tenemos el DW cargado con sus tablas y medidas, la siguiente y última etapa es la manera en que se presenta la información al usuario. Para esto existen herramientas dedicadas a ello que se denominan Herramientas de Exploración y Análisis. Pueden ser herramientas de análisis multidimensional (herramientas OLAP) que nos permiten armar distintos “cubos” que cruzan la información, o también podemos obtener reportes planos de cierta complejidad o que impliquen consultas costosas; por lo lado tenemos el Data Mining, que es otro tipo de explotación de datos que en este documento no vamos a entrar en detalle.

Los usuarios idóneos (no técnicos) , especialistas del negocio de la organización utilizarán la herramienta y podrán realizar distintas operaciones de agrupación, cruzamiento, sumatorias, promedios, entre muchas otras, para poder tomar decisiones importantes en la organización en función de los datos del pasado. Estos usuarios finales son principalmente gerentes o de cargos altos cuyo trabajo refiere a la toma de decisiones y a la planificación.

## SDW - Arquitectura base

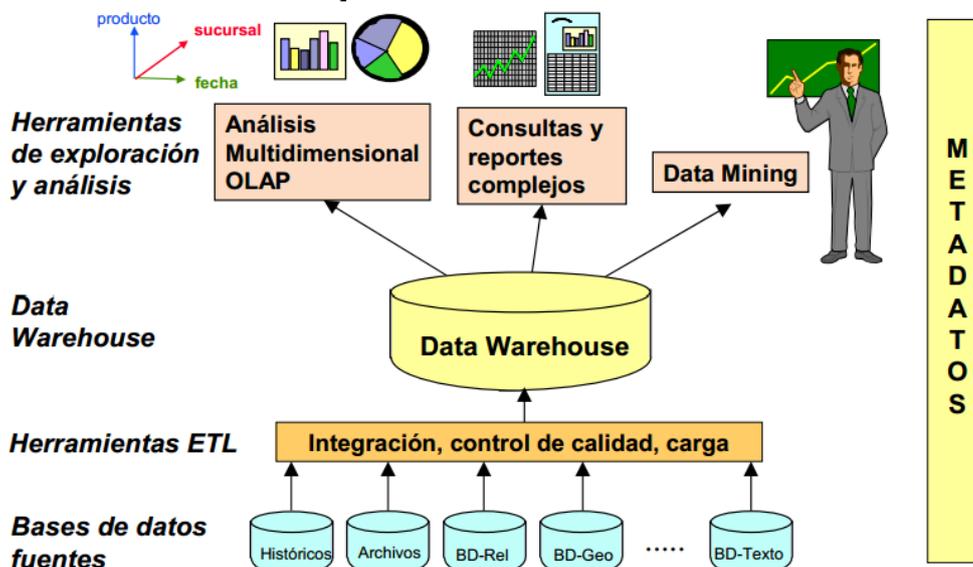


Figura 1 Diagrama de Arquitectura típica de un Sistema de DW [3]

### Desarrollo de los Sistemas de Data Warehouse

En cuanto al desarrollo de los Sistemas de DW, podemos identificar distintas fases [3]:

Especificación de Requerimientos.

Los técnicos deben relevar con los usuarios finales, las distintas necesidades que éstos últimos tienen sobre un sistema de este tipo, qué información necesitan saber periódicamente sobre la realidad del negocio de la empresa u organismo. Para esto es necesario que los técnicos realicen una especificación de los requerimientos, una documentación que represente lo que el DW debe exponer, qué operaciones debe soportar, y qué indicadores son de interés para el negocio en cuestión, entre otras cosas.

Diseño Conceptual.

El objetivo de la realización de un diseño conceptual, está enmarcado en poder visualizar los objetos que pertenecen al negocio a estudiar y las relaciones entre ellos. Poder obtener una descripción abstracta y completa del problema. Dicho modelo expresa la realidad en función de dimensiones y medidas. Se puede explicar sobre el tema en [4].

Podemos generar dicho diseño en función de los datos relevados en la especificación de requerimientos o estudiando las distintas fuentes de datos que tenemos a disposición.

En primera instancia se detectan los objetos relevantes para la problemática y se determina si son dimensiones o medidas.

A continuación, vamos a exponer el modelo CMDM para diseños conceptuales. [4]

## ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

En un esquema multidimensional tenemos las dimensiones que caracterizan a una actividad en particular, esto es, un hecho relevante a nuestro negocio y será un objeto de análisis. Cada dimensión estará compuesta por un conjunto de atributos que representan la misma.

Además, sobre estas dimensiones tendremos las jerarquías, esto es, atributos de la dimensión por el cual podemos agrupar la información relevante a dicha dimensión. Esto es, si la dimensión fuera "Vendedor", podríamos subir en la jerarquía por ejemplo por el "Sexo" o por su "Ciudad" de residencia.

Por ejemplo, si tuviéramos distintos vendedores de una empresa de autos y los mismos realizan ventas y tuviéramos las fechas de las mismas, podríamos tener siguientes dimensiones con sus jerarquías. (Ver Figura 2)

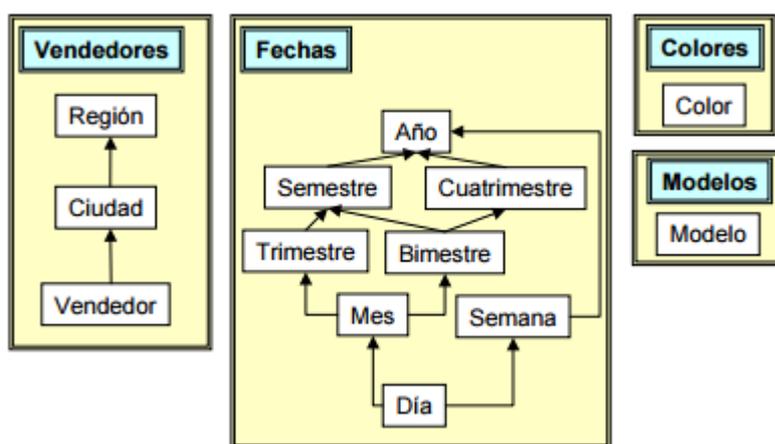


Figura 2 Diseño Conceptual para Vendedores de autos [3]

Seguramente queramos vincular estas dimensiones en lo que ya mencionamos como "hechos". Esta vinculación se representa como una "Relación Dimensional".

En dicha relación dimensional, es probable que queramos obtener información relevante sobre distintas variables y obtener distintos indicadores. Estos indicadores en el diseño son los que se llaman las "medidas".

Si continuamos con el ejemplo anterior, en la Figura 3 podemos ver que la cantidad de ventas de un vendedor en un período en particular puede ser un indicador interesante.



Figura 3 Medida "Cantidad" en la Rel. Dim. "Ventas" [3]

Dado un valor para cada dimensión se puede determinar un valor para la medida, se obtiene en la intersección de algún valor de las distintas dimensiones.

Según definiciones del curso de Diseño y Construcción de Data Warehouse de la Facultad de Ingeniería de UdelAR, las dimensiones se pueden ver como distintos criterios de análisis de los datos, macro objetos del

## ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

problema y/o variables independientes. Con respecto a las medidas, pueden ser valores o indicadores a analizar, datos asociados a relaciones entre objetos del problema, variables dependientes y/o variables en la intersección de las dimensiones.

### Diseño Lógico.

La etapa de diseño lógico obtiene como base el esquema conceptual antes mencionado y como resultado provee un esquema lógico relacional o multidimensional. Uno de los principales problemas en esta etapa es obtener un esquema lógico que satisfaga los requerimientos de performance en la realización de consultas complejas en el análisis de los datos, además de satisfacer los requerimientos básicos de los usuarios [3].

En [5], Kimball propone el modelo estrella, es una tabla principal que contiene información sobre los hechos y referencias a tablas más pequeñas con información sobre las dimensiones.

Las etapas que están incluidas en la construcción de un DW son:

- Extracción: la extracción de los datos se realiza sobre diferentes fuentes de distinta naturaleza, interfaces de diferentes tecnologías y de manera independiente a las transacciones operacionales de la organización. Para esto se realiza sobre procesos batch o Jobs, que se ejecutan en horarios de carga nula en la operativa.
- Transformación: Dado que los datos son provistos por diferentes fuentes, requieren una adecuación o normalización. Limpieza y optimización de los mismos orientado a la performance del DW.
- Carga: Se almacenan tablas pertenecientes al modelo multidimensional del DW, para que puedan ser utilizadas por herramientas dedicadas a la visualización y manipulación de la información.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

## **Web Warehouse**

A continuación vamos a definir un concepto fundamental en nuestro proyecto que es el Web Warehouse (abreviado WW). Es un DW pero tiene la particularidad que sus datos son extraídos de fuentes web y se agregan otros indicadores como la calidad de los servicios que nos proveen esos datos, nuevos obstáculos que son ajenos al WW, etc.

Con el fin de ser una herramienta útil, un WW debe tener en cuenta diferentes cualidades, como son la calidad de los datos que gestiona, y la calidad de servicios que proporcionan los datos de origen.

Además, teniendo en cuenta la alta volatilidad de los servicios web, la misma debe permitir una cierta flexibilidad y configurabilidad sobre los mismos.

### **Introducción.**

El proceso de construcción de un WW presenta grandes desafíos: como por ejemplo los distintos formatos que presentan los datos de la web y protocolos heterogéneos, como pueden ser HTML, XML, RSS, Web Services. Así como también la calidad de los datos que muy comúnmente presentan inconvenientes y problemas como errores de escritura de datos, valores perdidos, datos obsoletos e incompatibles.

El módulo del WW encargado de la extracción de los datos de la web, está compuesto por una cierta cantidad de servicios que se conectan al proveedor de los datos. Dichos servicios no forman parte del WW propiamente, pero sin ellos no sería posible generarlo. Estos servicios se denominan Web Data Service (WDS) y son implementados en su totalidad por desarrolladores de software.

Otro punto a considerar es la autonomía de estos WDS, en general conduce a la incertidumbre en cuanto a disponibilidad, costo, y los inesperados cambios en formatos y protocolos que utilizan para proporcionar los datos.

Un sistema de este tipo, permite medir la calidad de datos y servicios para seleccionar distintas fuentes e integrar los datos y proporcionar también valor agregado referente a la calidad a los consumidores de datos del WW.

Con el fin de alcanzar la flexibilidad y configurabilidad de este tipo de plataformas, deberíamos realizar un diseño orientado a servicios. Las diferentes tareas que se realizan para el procesamiento de datos que alimentaran el WW y luego los cubos multidimensionales se implementan como datos y servicios de información especializada.

Estos servicios son seleccionados dinámicamente por el sistema de acuerdo con la necesidad de procesamiento de datos que puede variar en cada carga de WW. [\[6\]](#)

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

## Ciclo de vida de un Web Warehouse

Debido a la complejidad de este tipo de sistemas es importante especificar las diferentes etapas que componen su ciclo de vida, así como los diferentes roles que participan activamente en su instalación y operación. [6]

En la Figura 4 podemos ver un ejemplo de arquitectura de un WW, en la cual se plantean tres principales componentes que son: Extracción de Datos (DSI), Integración de Datos (Integrator) y Transformación y carga de los datos al DW a través de diferentes módulos (ETL).

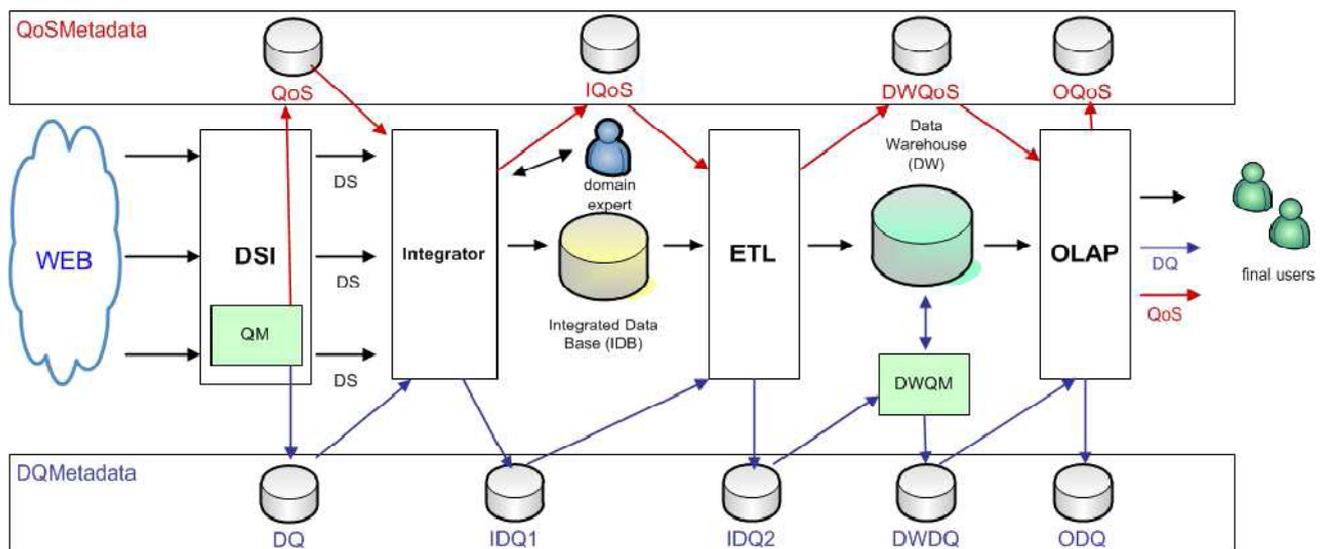


Figura 4 Arquitectura propuesta para la Construcción de un Web Warehouse. [6]

DSI:

El módulo Data Services Infrastructure (DSI) proporciona un entorno de ejecución para implementar, almacenar y ejecutar Web Data Services, con el fin de acceder y obtener los datos de las fuentes Web.

Tres principales responsabilidades:

- Proporcionar los mecanismos para extraer datos de diferentes tipos de WDS. Y exponerlos como DS homogéneos utilizando tecnologías estándar.
- Monitorear los WDS y periódicamente medir la calidad. La medición de la calidad de datos y servicios se efectúan por servicios específicos que son invocados por el módulo de gestión de calidad.
- Proporcionan mecanismos de adaptación en tiempo de ejecución, en el proceso de extracción para reaccionar a la posible degradación de calidad de los WDS.

Integrator:

El módulo Integrator que invoca los WDS realiza la integración de datos y proporciona datos en un formato normalizado, generando la base de datos integrada (IDB, Integrated Data Base).

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

Sus principales responsabilidades son:

- Selección orientada a la calidad de los WDS.
- La integración de datos.
- Generación de metadatos de calidad asociados a los datos integrados.

El Integrator interactúa con los usuarios expertos que realizan las principales decisiones de integración de datos, tales como la identificación de objetos y resolución de conflictos. Esta interacción se logra a través de una interfaz de usuario. [6]

## ETL y OLAP

Los módulos de ETL y OLAP llevan a cabo las funciones clásicas en el contexto de DW.

Además de esto, tiene que construir metadatos de calidad de datos y de calidad de servicios, en la arquitectura: DQ y Qos.

A su vez, DWQM (manejador de calidad del DW) obtiene la medida de la calidad de los datos, recibiendo la calidad de datos de los metadatos como entrada y combinándolas con nuevas medidas realizadas sobre el DW a través de la invocación de servicios de medida de calidad específicos del mismo.

Una vez implementada la plataforma del WW, deben llevarse a cabo algunas tareas antes de que pueda ser puesto en funcionamiento.

En primer lugar, deben ser seleccionados un conjunto de sitios web del dominio de interés, a continuación la plataforma debe ser instalada y configurada con el fin de crear una instancia para el dominio en particular. Después de eso, la carga del WW puede ser ejecutada, esto incluye la extracción de datos web, integración, transformación y carga del DW.

Por último el WW puede ser explotado por el usuario final, todas estas etapas componen el ciclo de vida del WW.

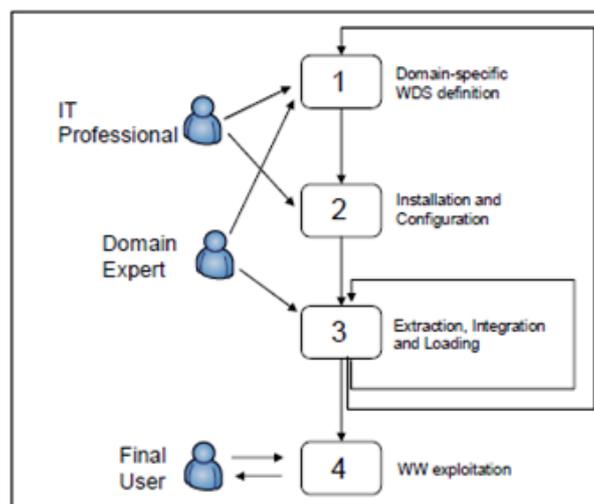


Figura 5 Perfiles y sus actividades [6]

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

La Figura 5 muestra las diferentes etapas en el ciclo de vida del WW con la participación de diferentes roles. Definimos las siguientes 4 etapas:

1. Definición de un conjunto de fuentes de datos web de dominio específico. En esta etapa se selecciona un primer conjunto de fuentes de datos de la web, teniendo en cuenta las características de calidad, tales como el costo y la reputación. Los llamamos candidatos Web Data Source.
2. La instalación y configuración del sistema. En esta etapa la plataforma de WW es instanciada para un dominio y requerimientos en particular. Esto implica las siguientes tareas principales:
  - a. Diseño y definición del DW (incluye el esquema y los procesos ETL).
  - b. Diseño del esquema el cual se alimenta de los Data Service arrojados por la DSI.
  - c. Desarrollo de los Data Services.
  - d. Selección de los Data Services, se seleccionan los Data Services que cargarán el WW, teniendo en cuenta las características de calidad que se deducen de la ejecución sucesiva de ellos como el rendimiento o la accesibilidad de contenidos.
3. Extracción, integración, transformación y carga con la gestión de calidad, en esta etapa es cuando se carga el DW de los datos que provienen de la web. Esta ejecución puede ocurrir periódicamente o cada vez que se produce una actualización de los datos en el WDS. Como se muestra en (Ver Figura 5), a partir de esta etapa es posible volver a la etapa 1, si se quiere cambiar el candidato Web Data Source por ejemplo agregando un nuevo sitio web como fuente. Durante este proceso, se consideran los metadatos de calidad para la invocación de los Data Services, para tareas de integración y además se enriquece y se propaga al DW.
4. La explotación del WW. El WW es explotado a través de las herramientas OLAP que permiten la navegación de datos y la aplicación de las operaciones multidimensionales sobre los datos para su análisis. Los cubos de datos construidos también puede ser publicados en la Web que está disponible para cualquier usuario.

Durante el ciclo de vida descrito, interactúan diferentes tipos de usuarios con el sistema. El profesional de IT debe participar de las etapas 1 y 2, siendo necesario en todas las tareas que son anteriores a la carga y la explotación del WW. El experto de dominio debe ayudar con su conocimiento cuando son seleccionados los candidatos Web Data Source (etapa 2 del ciclo de vida), así como en el momento de selección de Data Service y la integración de los datos (etapa 3 del ciclo de vida). Durante la integración de datos este usuario ayudará con la resolución de conflictos y la identificación de problemas en objetos. Por último el usuario final tendrá el beneficio del WW construido. [\[6\]](#)

## Gestión en la calidad de un WW.

En un WW existen dos grandes pilares en cuanto a la calidad. El primero de ellos apunta a la calidad de los servicios que nos proveen los datos y el otro son los factores relacionados con la calidad de los datos propiamente.

Principales dimensiones de calidad de los servicios:

- Nivel de servicio.
- Interoperabilidad.
- Seguridad.
- Valor de Negocio

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

- Usabilidad
- Estabilidad

Principales dimensiones de calidad de los datos:

- Exactitud
- Integridad
- Actualización (datos viejos?)
- Unicidad
- Coherencia
- Fiabilidad

A lo largo del proceso de selección y extracción no todos los factores de calidad puede ser relevantes para una tarea específica, depende de que tarea que factores de calidad son relevantes:

1. Selección de candidatos Web Data Source: los factores relevantes son los que se utilizan para decidir si vale la pena construir un contenedor de WDS (DEFINIR CONTENEDOR CON OTRAS PAALBRAS).
2. Selección de DS para carga de WW: los factores pertinentes son los que se utilizan por el usuario especializado para seleccionar el DS a usar.
3. La propagación de los valores de calidad de los WW: los factores relevantes son los útiles para los usuarios finales del WW.

Es importante observar que este análisis considera un escenario muy general. Se espera que la relevancia de un factor de calidad pueda variar de acuerdo al contexto o dominio específico. [\[7\]](#)

## Plataforma Orientada a Servicios

La arquitectura propuesta aprovecha el enfoque orientado a servicios para llevar a cabo sus tareas principales.

Selección y Extracción de datos Web.

Este proceso se compone de cuatro pasos principales:

- El primer paso consiste en seleccionar los candidatos Web Data Source, esta selección se realiza teniendo en cuenta los factores de calidad mencionados.
- En el segundo paso, consiste en el desarrollo y publicación de los Data Services. Con el fin de proporcionar un acceso homogéneo a los distintos tipos de Web Data Source, se propone la implementación de los Data Service para encapsular la lógica de extracción. Para el desarrollo del Data Service el módulo DSI ofrece un entorno de ejecución que permite medir la calidad del Data Service.
- El tercer paso es la selección de los DS, que es ayudado por 2 componentes: el Data Service Registry (DSR) y el componente de Data Service Quality Management (QM). El DSR ofrece funcionalidades para buscar y descubrir las DS publicadas. El componente QM supervisa continuamente la calidad del DS. La información de calidad calculada por el módulo QM queda disponible para el componente DSR y el módulo integrador, por lo que se puede propagar al WW.

## ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

- El último paso es la invocación de DS. Dado que la calidad de los datos puede cambiar con el tiempo el módulo DSI provee un mecanismo de adaptación de la calidad con el objetivo de mantener cierto nivel en la misma.

### Procesos de Negocio.

Existen varias definiciones para procesos de negocio, algunas iniciales de la década de los noventa y otras más actuales que involucran nuevos elementos. Un proceso de negocios es:

- “Un conjunto de tareas relacionadas realizadas para alcanzar una salida del negocio definida para un cliente o mercado particular” (Davenport, 1993) [2].
- “Una colección de actividades que toma una o más tipos de entrada y crea una salida que es de valor para el cliente” (Hammer & Champy, 1993) [2].
- “Un conjunto de actividades realizadas en coordinación en un entorno organizacional y técnico, para alcanzar un objetivo del negocio” (Weske, 2007) [2].

Este conjunto de actividades tienen como objetivo ayudar a alcanzar un determinado objetivo de negocio. Cada proceso de negocio es realizado por una única organización, pero puede interactuar con procesos de otras organizaciones.

En la Figura 6 podemos ver los principales elementos que son las entradas y salidas, que están formadas por los requerimientos de clientes, información, servicios, productos, etc. Los recursos destinados para realizarlos con las personas, materiales y un conjunto de objetivos de la organización y los procesos de negocio específicos.

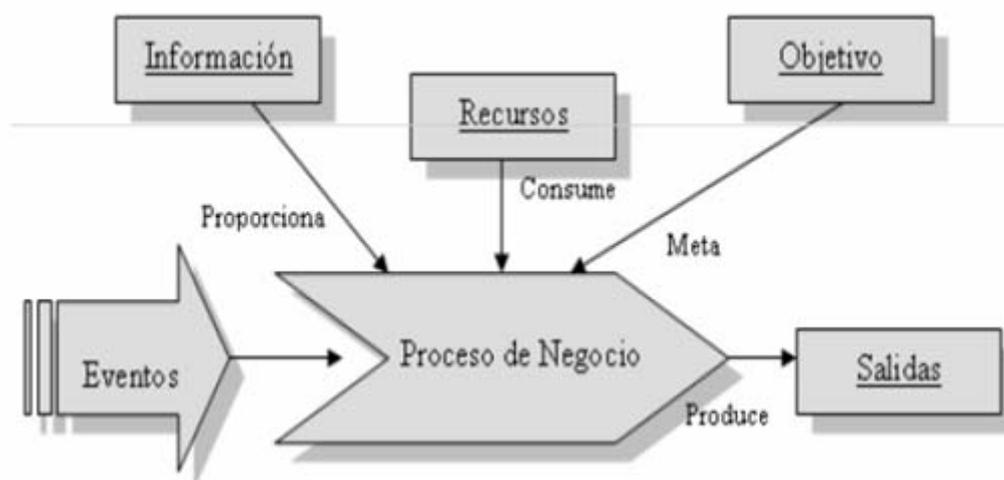


Figura 6 Elementos participantes en un Proceso de Negocio [2]

Existen también elementos específicos en la construcción de un Proceso de Negocio, por ejemplo los participantes internos de la organización (secciones, áreas, roles), participantes externos (clientes, socios, proveedores), actividades, tareas, datos, información y las relaciones entre ellas (secuencia, reglas de negocio, bifurcaciones, restricciones).

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

Los procesos de negocio se caracterizan por ser grandes, complejos y muy dinámicos. Además involucran varias secciones o áreas de una organización y organizaciones distintas y tienen duraciones prolongadas a lo largo del tiempo, o sea una ejecución puede durar semanas, meses o años. Estos procesos son específicos de Cada dominio: salud, bancarios, etc., pueden ser tanto manuales, automatizados o ambos y en general son difíciles de explicitar (implícitos en sistemas) [2].

En la Figura 7 podemos ver un diagrama que nos permite entender las relaciones de los distintos objetos en una organización junto con la interacción con el Sistema y la importancia de los Procesos de Negocio.

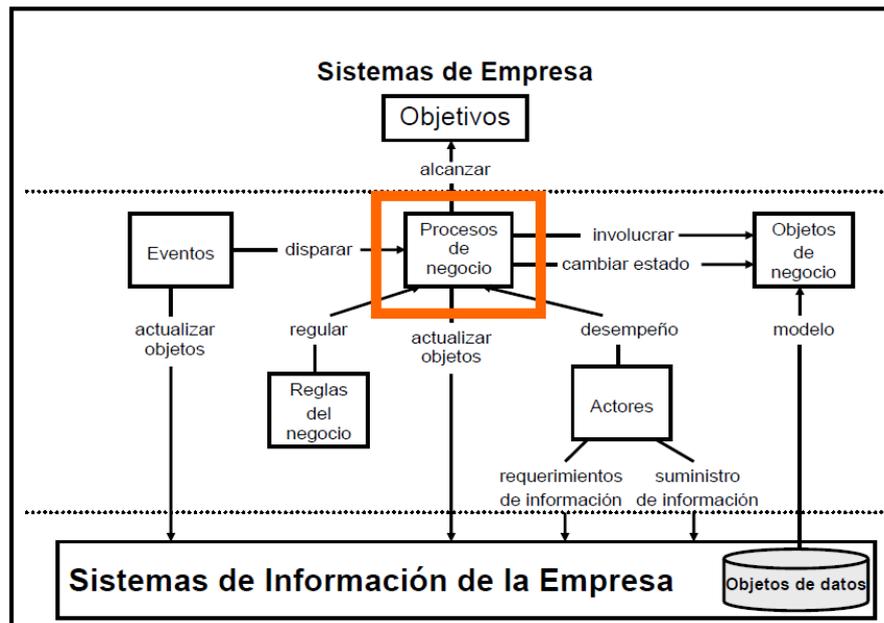


Figura 7 Uso, interacción con el Sistema y la importancia de los Procesos de Negocio [1]

## Modelos de Procesos de Negocio

Son una representación abstracta (gráfica) de los procesos de una organización, que muestran principalmente cómo y por quién son llevadas a cabo las actividades que generan valor para la organización. Además muestran también otro tipo de información que es de valor agregado para la empresa como ser los actores involucrados en los procesos (roles, áreas), cuáles son las actividades operativas distinguibles, que actividades son ejecutables y por quien, cuales son las entradas y salidas de actividades, cuál es la secuencia de las actividades, los recursos consumidos, y los eventos que dirigen el proceso.

La definición de Modelo de Proceso de Negocio es:

*"Representación del conjunto de actividades y de las restricciones de ejecución entre ellas"* (Weske, 2007) [2].

Una instancia es un caso concreto en una organización, formado por instancias de actividades, cada modelo de proceso actúa como una plantilla para un conjunto de instancias de proceso, cada modelo de actividad actúa de forma similar para un conjunto de instancias de actividad.

En la Figura 8 se muestra un ejemplo de esta representación gráfica para "hacer pan". Se puede observar el *lane* que es el "Panadero", un evento de inicio, seguido de las tres actividades manuales por la que está formado el proceso y termina con el evento de fin. [1]

## ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

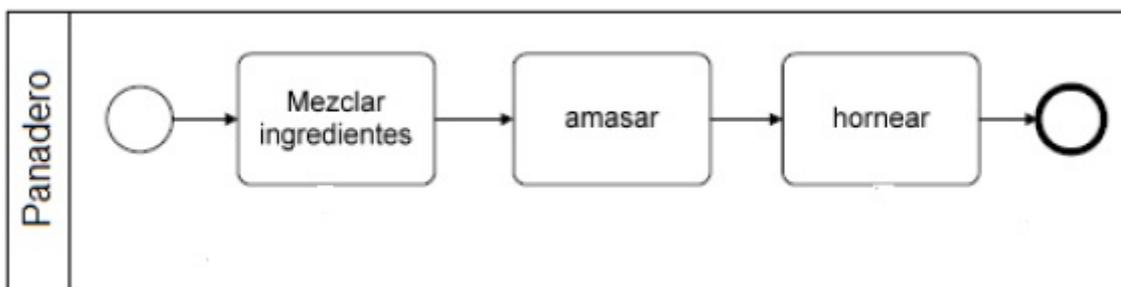


Figura 8 Ejemplo de modelo para hacer pan.

Estos modelos permiten razonar sobre posibles re-diseños, evaluando distintas posibilidades (flujos, interacciones), permiten además realizar un análisis basado en simulación o técnicas analíticas, predecir performance, utilización de recursos, cuellos de botella y son la base para incorporación de un BPMS, automatización de actividades, interacción humana, envío y recepción de mensajes.

Algunos posibles errores al realizar el modelado de Procesos de Negocio podrían ser: describir sólo una idealización de la realidad, concentrarse en el flujo “normal” ó “deseable” y dejar afuera flujos alternativos que puedan causar problemas, modelar a un nivel de abstracción erróneo, muy abstracto y general, no permite “ver” el proceso, muy detallado se vuelve muy complejo y difícil de entender, no elegir correctamente la notación a utilizar, no todos los involucrados pueden “entender” lo que está modelado (dificulta la comunicación).

### Gestión de Procesos de Negocio

Las nuevas tecnologías de gestión se han introducido en las organizaciones como una herramienta para solucionar problemas con clientes, proveedores o empleados, y además fomentan la colaboración entre los actores, la Gestión de Procesos de Negocio se podría definir como:

*“Un conjunto de actividades que realizan las organizaciones para optimizar o adaptar sus Procesos de Negocio a las nuevas necesidades organizacionales”* (BPMI, 2005) [2].

*“Incluye conceptos, métodos y técnicas para soportar el diseño, administración, configuración, ejecución y análisis de los Procesos de Negocio en las organizaciones”* (Weske, 2007) [2].

*“Soporte para Procesos de Negocio utilizando métodos, técnicas y software para diseñar, ejecutar, controlar y analizar procesos operacionales involucrando humanos, organizaciones, aplicaciones, documentos y otras fuentes de información”* (Van der Aalst, 2003) [2].

Los elementos principales son: optimizar, adaptar, mejorar los Procesos de Negocio en las organizaciones, procesos alineados con objetivos organizacionales, incluye conceptos, métodos, técnicas y software (BPMS). Además soporta el diseño, análisis, configuración, ejecución, control, evaluación de los Procesos de Negocio (Ciclo de vida de los Procesos de Negocio), involucrando humanos, organizaciones, aplicaciones, documentos y otras fuentes de datos.

### BPM

En la Figura 9 y la Figura 10 podemos ver un antes y un después en la estructuración de las organizaciones con o sin BPM respectivamente, antes la gente de negocios (procesos, roles, personas, ...) y los técnicos (sistemas, máquinas, datos, ...) hablaban en términos de la función que ocupan en la organización, en

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

cambio ahora con BPM es posible que todos hablen de lo mismo, la tecnología BPMS permite salvar la distancia con los sistemas, máquinas y aplicaciones que automatizan los PN [1].

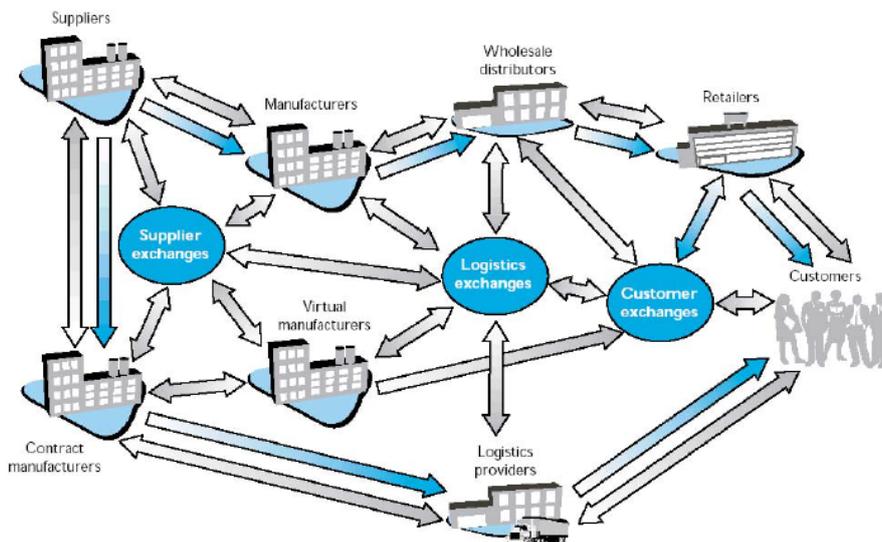


Figura 9 La organización y la comunicación antes del BPM [1]

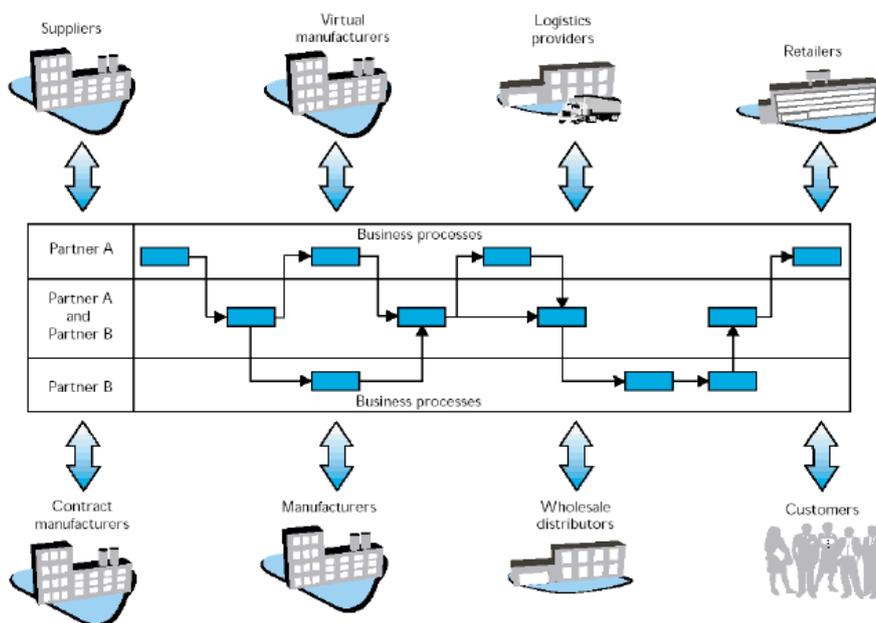


Figura 10 Estructura y comunicación en una organización luego del BPM [1]

BPM incluye conceptos, métodos y técnicas para dar soporte al diseño, administración, configuración, realización y análisis de procesos de negocio. Algunas de las definiciones más comunes dicen que un BPMS es:

- "Un sistema de software genérico, guiado por representaciones explícitas de procesos con sus actividades y restricciones para coordinar la ejecución de procesos de negocio" (Weske, 2007) [2].

## ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

- “Un sistema de software genérico guiado por diseños explícitos de procesos para ejecutar y gestionar procesos de negocio operacionales” (Van der Aalst 2003) [2].

- “Un sistema de software genérico que permite coordinar la realización (ejecución) de procesos de negocio en base a representaciones de proceso explícitas (modelos)” (Weske, 2007) [2].

O sea los BPMS son sistemas “conscientes” de los procesos de negocio (modelos) los cuales guían su ejecución (puede ser analizados, mejorados, etc.). Un BPMS tiene incluir un motor de procesos.

Una vez definido (modelado), un PN puede ser sujeto a análisis, mejora o realización.

En la Figura 11 mostramos los componentes más importantes de la arquitectura típica de un BPMS.

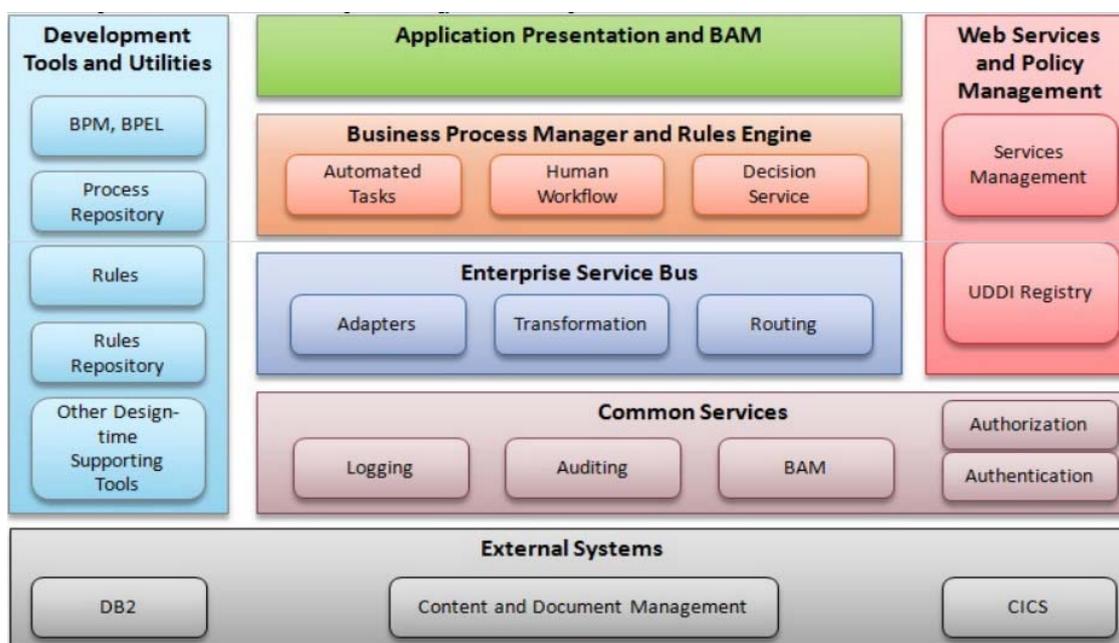


Figura 11 Arquitectura típica y componentes de un BPMS [2]

Los BPMS soportan el ciclo de vida de los procesos de negocio. En los BPMS convergen y se integran diversas tecnologías (middleware), que ya están maduras a nivel de mercado como por ejemplo: servidores de aplicaciones, EAI (Enterprise Application Integration), WorkFlows, ERP, CRM, E-Business, E-Commerce, EDI (Electronic Data Interchange), servicios web, reglas de negocio (Rules Management), inteligencia de negocio (Business Intelligence), cuadros de mando (Business Activity Monitoring). Un BPMS, también llamado BPM Suite, puede incluir múltiples partes. (Ver Figura 12)

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

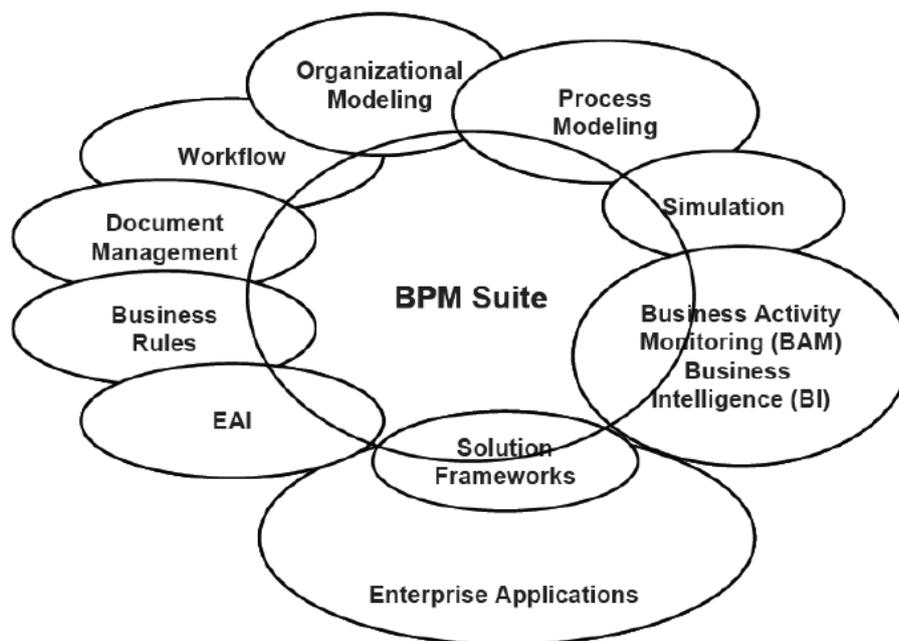


Figura 12 Posibles tecnologías que vincula un BPMS [1]

## Ciclo de Vida de Procesos de Negocio

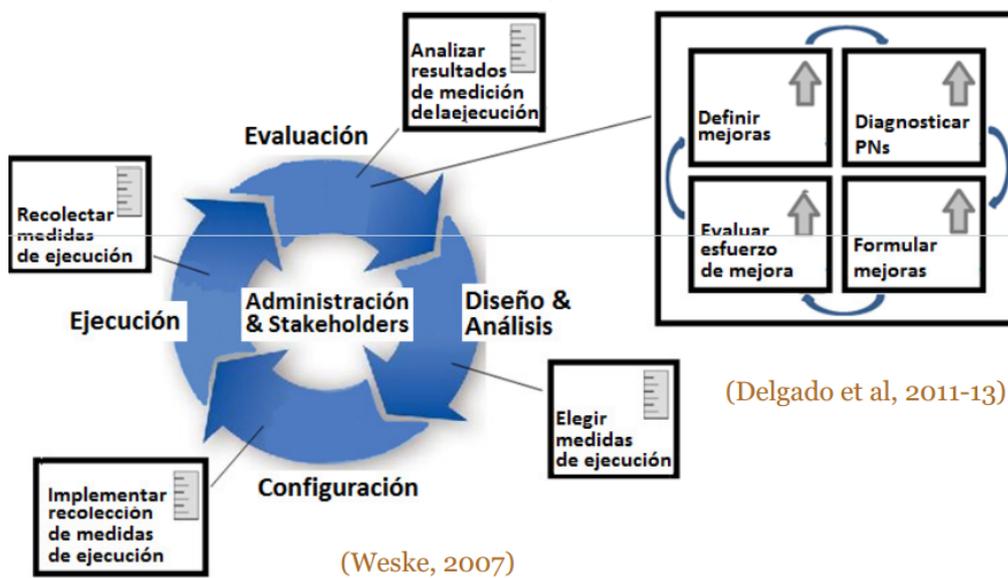


Figura 13 Ciclo de vida de los Procesos de Negocio [2]

En la Figura 13 y la Figura 14 podemos ver las distintas etapas en el ciclo de vida de los Procesos de Negocio.

En la fase Diseño y Análisis, el objetivo es modelar y validar los procesos de negocio en la organización, las tareas principales en esta etapa son: analizar los procesos de negocio y su entorno organizacional y tecnológico, representarlos en modelos de procesos de negocio con notación adecuada (BPMN 2.0) y buenas prácticas (Workflow patterns) y finalmente verificar y validar los modelos especificados.

## ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

A continuación en la fase de Configuración, el objetivo es implementar, testear y desplegar los procesos de negocio en la organización, en la que se destacan como tareas principales seleccionar plataforma, tecnologías y lenguajes de implementación, implementar procesos de negocio y software (servicios), reglas de negocio, formularios de usuario, integración, etc., y realizar testing y despliegue del sistema en la organización para su operación (capacitación, migración de datos, etc.).

Luego en la fase de Ejecución, el objetivo es ejecutar los procesos de negocio y registrar datos asociados a la ejecución, para la cual las principales tareas son ejecutar los procesos de negocio según el modelo de proceso de negocio definido y las restricciones, reglas de negocio, etc. Asociadas, registrar datos de la ejecución en logs de ejecución (ejemplo: secuencia de ejecución de actividades, tiempos asociados, recursos involucrados, datos manejados, etc.) y monitorear la ejecución de los procesos de negocio, generalmente con un componente de Business Activity Monitor (BAM).

Finalmente en la fase de Evaluación, cuyo objetivo es evaluar la ejecución de los procesos de negocio para mejorar los modelos y la implementación, las actividades principales son procesar los logs de ejecución de procesos de negocio y proveer información, evaluar la ejecución real de procesos de negocio con técnicas de Business Intelligence (BI), medidas de tiempos de ejecución, recursos utilizados, costos, pacientes atendidos, etc. (Key Performance Indicators, KPI) y descubrir modelos, compararlos con ejecución, extenderlos con datos reales (Process mining, minería de procesos) [2].

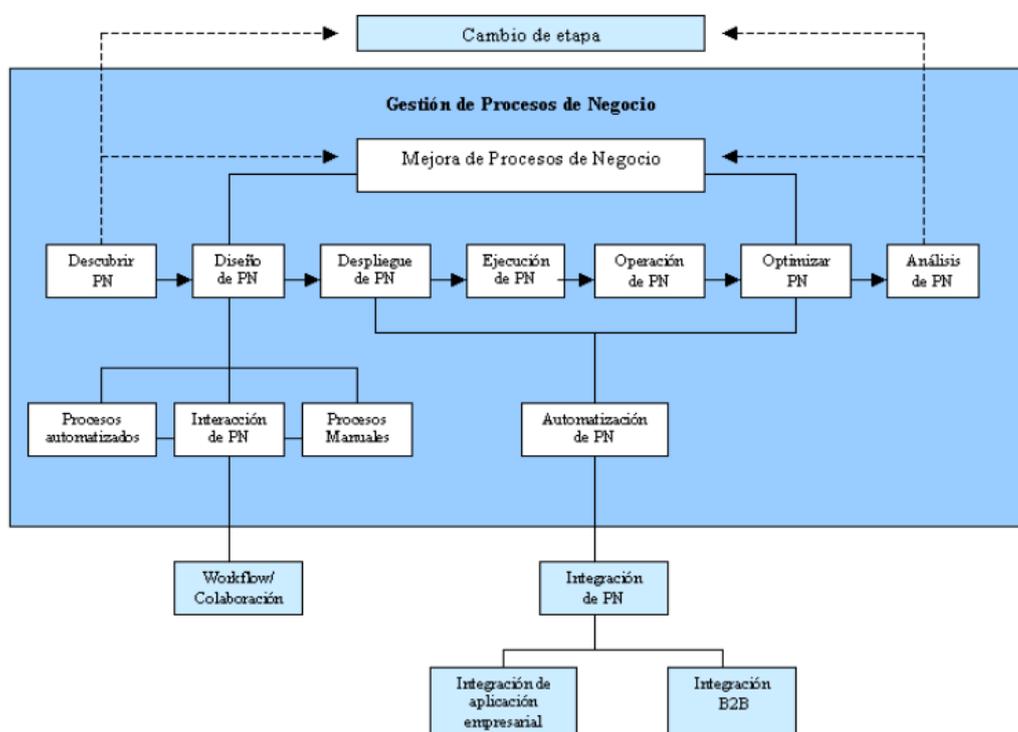


Figura 14 Visión del Ciclo de Vida un poco más extendida. [1]

La fase de Descubrimiento es hacer explícita la manera en que se hacen las cosas (frente a cómo se deberían hacer).

Darse cuenta explícitamente de cómo se hacen las cosas realmente, frente a cómo se dice que se hacen.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

Implica conocer los PN (flujo de eventos, flujo de mensajes y flujo de órdenes) desde la perspectiva de todos los participantes (personas o sistemas). Permite obtener una imagen clara sobre cómo trabajan interna y externamente los PN. [1]

La fase de Diseño consiste en modelar, simular y reestructurar el PN, elaborar un modelo de cada PN que pueda ser comprensible por todos los involucrados, pudiendo reutilizar de conocimiento (subprocesos). Generar un repositorio de procesos (en papel y en formato electrónico), reutilización, generalización y especialización de elementos de procesos aplicando buenas prácticas. [1]

La fase de Despliegue tiene como principal objetivo implantar un nuevo PN a todos los participantes (personas, sistemas, otros procesos).

En la fase de Ejecución el objetivo es asegurar que el nuevo PN es llevado a cabo por todos los participantes. A continuación en la fase de Interacción, la principal actividad es permitir a las personas gestionar el interfaz entre procesos automáticos y manuales. [1]

Finalmente en la fase de Operación y Mantenimiento, se podrían identificar como tareas principales la intervención para resolver excepciones, reasignar participantes. La optimización del PN, cambiarlo para mejorarlo, la mejora de procesos debe ser un esfuerzo continuo, en ciclos de diseño-despliegue-ejecución-operación-optimización.

Y el posterior análisis (medir el rendimiento del PN e idear estrategias de mejora) y automatización, que se realiza durante las etapas de despliegue, ejecución, operación y optimización [1].

Los principales roles participantes son [1]:

- Jefe de Procesos (Chief Process Officer)
- Ingeniero de Procesos (Business Engineer)
- Diseñador (Process Designer)
- Participante (Process Participant)
- Trabajador de Conocimiento (Knowledge Worker)
- Responsable del Proceso (Process Responsible)
- Arquitecto de Sistemas (System Architect)
- Desarrollador (Developer)

## **BPMN (Business Process Modelling Notation)**

Es una notación gráfica (diagramas) para el modelado conceptual de procesos de negocio.

Se ha desarrollado buscando: proveer una notación que sea fácilmente entendida por todos los usuarios: analista de negocio, desarrollador técnico, o la propia gente del negocio, crear un puente estandarizado para el vacío existente entre el diseño del proceso de negocio y su implementación tecnológica y asegurar que los lenguajes para la ejecución de los procesos de negocio puedan ser visualizados con una notación común.

En particular BPMN 2.0 además de proveer lo mencionado anteriormente permite visualizar los procesos de negocio de la organización y todos los elementos que intervienen en el mismo (actividades, flujo, participantes, bifurcaciones, datos y subprocesos), intercambiar modelos realizados en distintas herramientas de modelado y permite la ejecución de procesos, en los nuevos motores de ejecución BPMN 2.0 (ejemplo: Activiti, JBPM5) [2]. La Figura 15 contiene todos los elementos mencionados en esta sección, como por ejemplo, Actividades, Compuertas, Eventos, etc.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

**BPMN 2.0 - Business Process Model and Notation** <http://bpmb.de/poster>

Traducido por: **Idelfonso Montano, Luciano Garcia Buñales, Martin Dumas**

**Actividades**

- Tarea:** Una Tarea es una unidad de trabajo, el trabajo a realizar. Cuando aparece con el símbolo indica un Subproceso, una actividad que puede ser reutilizada.
- Transacción:** Una Transacción es un conjunto de actividades relacionadas lógicamente, administradas a un proceso transaccional particular.
- Subproceso de Evento:** Un Subproceso de Evento se sitúa en el interior de otro Subproceso. Este subproceso indica la secuencia de actividades que se ejecutará cuando se produzca un evento. El subproceso de evento puede tenerse en cuenta al analizar el proceso que lo contiene.
- Actividad de Llamada:** Una Actividad de Llamada es una referencia a un Subproceso o Tarea definido de forma global que se reutiliza en el proceso actual.

**Tipos de Tareas:** Los tipos especifican la naturaleza de las tareas que se desea llevar a cabo.

- Subproceso
- Ciclo
- Instancias Múltiples en Paralelo
- Instancias Múltiples en Secuencia
- Ad Hoc
- Compartición

**Flujo de Secuencia:** Define el orden de ejecución entre dos actividades.

**Flujo por Defecto:** Cambio a seguir si las condiciones de los caminos alternativos evalúan a falso.

**Flujo Condicional:** Tiene una condición asociada que permite decidir si el camino será activado o no.

**Compuertas**

- Exclusiva:** En un punto de bifurcación, selecciona exactamente un flujo de secuencia de entre las alternativas existentes. En un punto de convergencia, la compuerta espera a que un flujo incidente complete para activar el flujo saliente.
- Paralela:** Esta compuerta siempre será seguida por eventos o tareas de recepción, y sólo activará un flujo saliente dependiendo del evento que ocurre en primer lugar.
- Inclusiva:** En un punto de bifurcación, al menos un flujo es activado. En un punto de convergencia, espera a que todos los flujos incidentes completen antes de activar el flujo saliente.
- Exclusiva Basada en Eventos (generador de instancias):** En la ocurrencia de uno de los eventos subsiguientes se crea una nueva instancia del proceso, activada para activar el siguiente.
- Paralela Basada en Eventos (generador de instancias):** En la ocurrencia de uno de los eventos subsiguientes se crea una nueva instancia del proceso, activada para activar el siguiente.

**Contenedores**

- Los Contenedores o los Contenedores de Mensajes representan a las entidades responsables de las actividades en un proceso (p.e., una organización, un rol o un departamento).
- El Flujo de Mensajes transmite la información que fluye a través de las organizaciones. Este flujo puede cooperar con el intercambio de mensajes que se especifica mediante la combinación de...

**Diagrama de Colaboración**

**Diagrama de Conversación**

**Diagrama de Coreografía**

**Eventos**

**Datos**

Figura 15 Todas las herramientas disponibles para BPMN 2.0 [21]

## Anexo B: Manual de usuario

En el presente apartado se explicara y detallara el uso del sistema para realizar la configuración y posterior carga de un WW.

### *Descripción de Proceso de Configuración*

A continuación se presentaran las pantallas de cada una de las actividades con el fin de facilitar el entendimiento del proceso de configuración por parte del usuario.

#### **SELECT DOMAIN**

Seleccionamos el dominio con el cuál queremos ejecutar la Configuración, mediante un combo con opciones precargadas como lo describe la Figura 16.

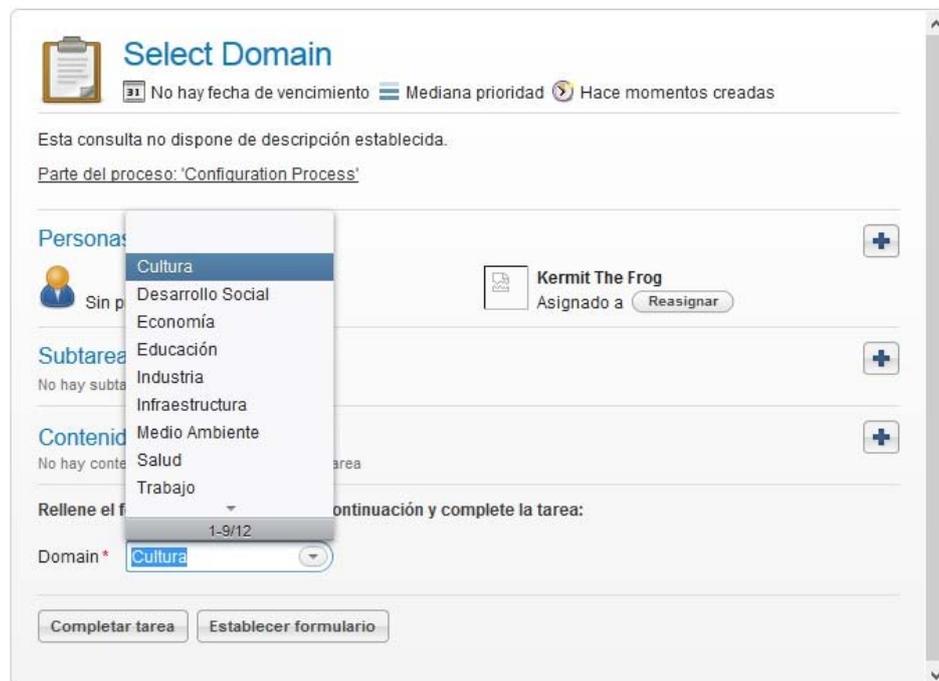


Figura 16 Select Domain

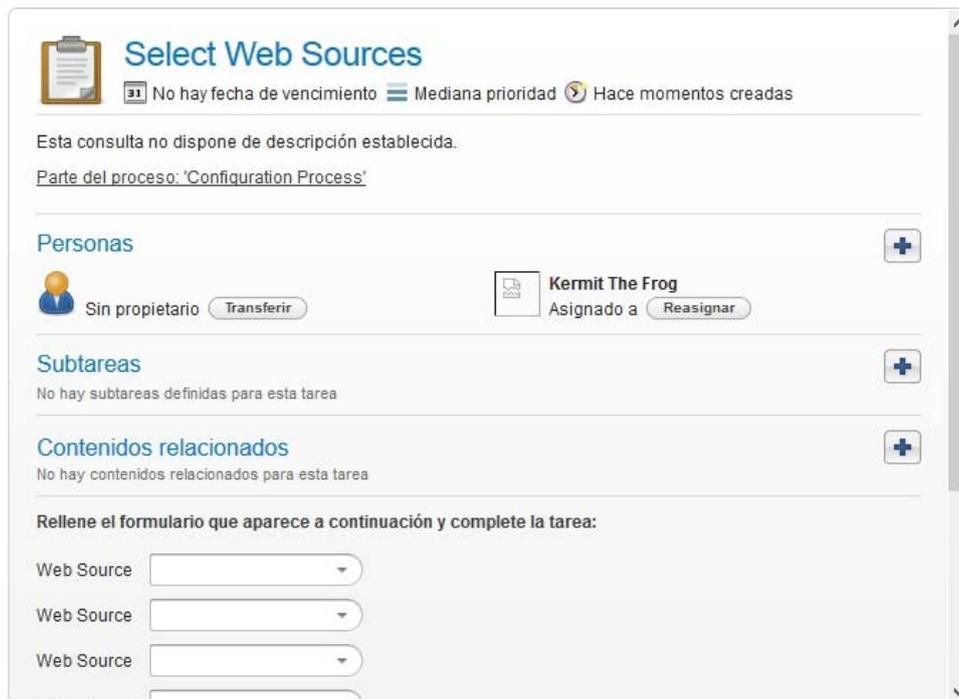
# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

## SELECT WEB SOURCE

En esta actividad, vamos a seleccionar Web Sources que ya existan y estén habilitadas para el dominio con el que estamos trabajando. El formulario que nos permite realizar esto se describe en la Figura 17.

Las Web Source son las fuentes web de donde vamos a extraer los datos.



**Select Web Sources**

No hay fecha de vencimiento Mediana prioridad Hace momentos creadas

Esta consulta no dispone de descripción establecida.

Parte del proceso: 'Configuration Process'

**Personas**

Sin propietario Transferir

Kermit The Frog  
Asignado a Reasignar

**Subtareas**

No hay subtareas definidas para esta tarea

**Contenidos relacionados**

No hay contenidos relacionados para esta tarea

Rellene el formulario que aparece a continuación y complete la tarea:

Web Source

Web Source

Web Source

Web Source

Figura 17 Select Web Sources

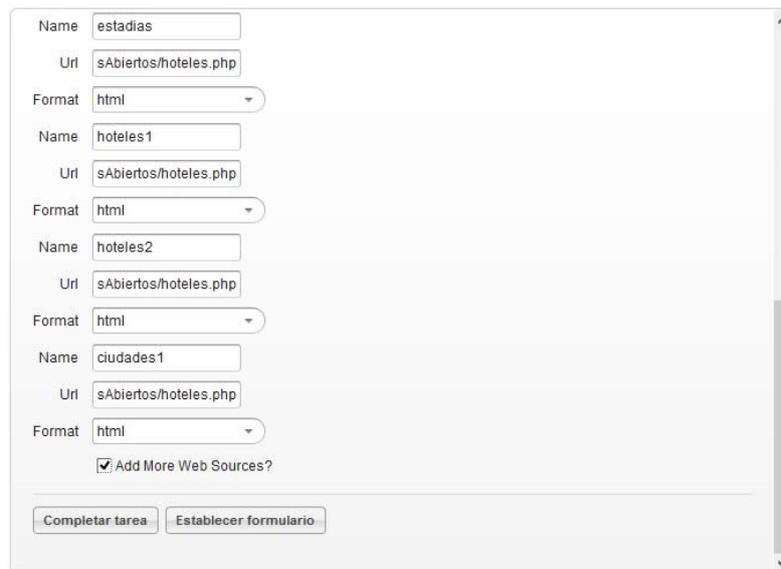
## CREATE WEB SOURCE

Esta tarea es muy similar a la anterior, pero presenta la diferencia de que las Web Sources a utilizar se deben dar de alta, es decir que no se usaron anteriormente (Ver Figura 18).

Esto nos permite en futuras ejecuciones de este proceso, tener ingresadas las Web Sources para determinado dominio.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---



The screenshot displays a web interface for creating web sources. It features a list of four entries, each with a 'Name' field, a 'Uri' field, and a 'Format' dropdown menu. The entries are:

- Name: estadias, Uri: sAbiertos/hoteles.php, Format: html
- Name: hoteles1, Uri: sAbiertos/hoteles.php, Format: html
- Name: hoteles2, Uri: sAbiertos/hoteles.php, Format: html
- Name: ciudades1, Uri: sAbiertos/hoteles.php, Format: html

At the bottom of the list, there is a checkbox labeled 'Add More Web Sources?' which is checked. Below the list are two buttons: 'Completar tarea' and 'Establecer formulario'.

Figura 18 Create Web Sources

## DEFINE EXPECTED SCHEMA

La siguiente actividad que la ilustramos con la Figura 19 y Figura 20, determina de qué forma queremos crear o seleccionar el Expected Schema para cada Web Source.

Las opciones son:

- Crear el esquema manualmente, y para ello debe activar "CreateSchemaManually". Se explicará en siguientes tareas.
- Seleccionando un Expected Schema creado anteriormente.
- Mediante un *json*, que nos permite de manera automática y con un formato determinado, generar toda la estructura referente al Expected Schema obteniendo el mismo resultado que si lo hubiéramos creado manualmente.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

Esta consulta no dispone de descripción establecida.

Parte del proceso: 'Define Expected Schema and Mapping'

**Personas** +

Sin propietario **Transferir** Kermit The Frog Asignado a **Reasignar**

**Subtareas** +

No hay subtareas definidas para esta tarea

**Contenidos relacionados** +

No hay contenidos relacionados para esta tarea

Rellene el formulario que aparece a continuación y complete la tarea:

Web Source: hoteles1

JSON: [Empty field]

Name Schema: Subir JSON

Create Schema Manually

**Completar tarea** **Establecer formulario**

Figura 19 Define Expected Schema and Mapping

Esta consulta no dispone de descripción establecida.

Parte del proceso: 'Define Expected Schema and Mapping'

**Personas** +

Sin propietario **Transferir** Kermit The Frog Asignado a **Reasignar**

**Subtareas** +

No hay subtareas definidas para esta tarea

**Contenidos relacionados** +

No hay contenidos relacionados para esta tarea

Rellene el formulario que aparece a continuación y complete la tarea:

Web Source: hoteles2

JSON: [Empty field]

Name Schema: Subir JSON

Create Schema Manually

**Completar tarea** **Establecer formulario**

- Subir JSON
- ciudades\_expected
- clientes\_expected
- estadias\_expected
- hoteles\_expected

Figura 20 Define Expected Schema and Mapping

## CREATE EXPECTED SCHEMA AND ATRIBUTES

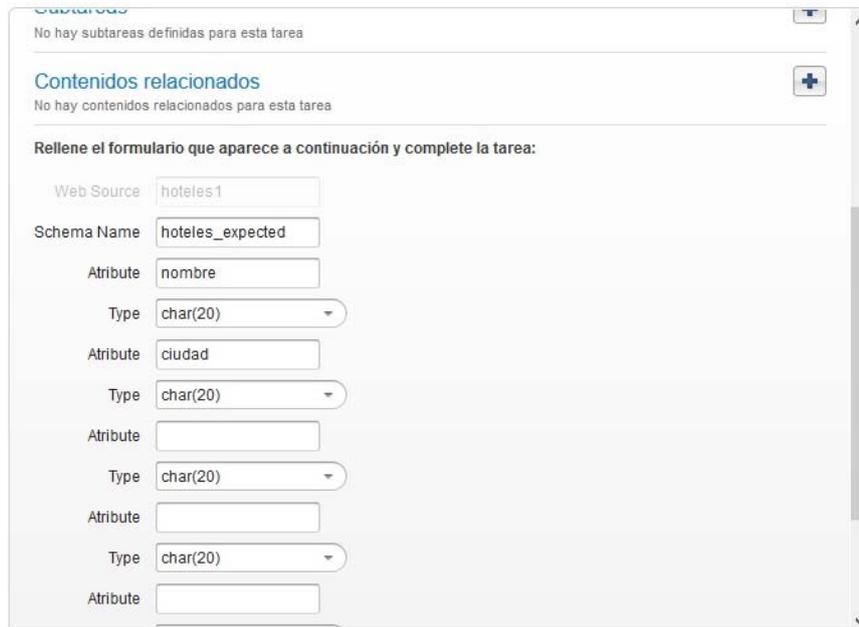
Como mencionamos en la tarea anterior, la presente actividad se ejecuta sólo si seleccionamos antes la manera manual de crear el Expected Schema.

Aquí ingresamos los datos de cada Expected Schema (nombre de la tabla, nombre y tipo de cada atributo).

Una opción que presenta esta tarea pero no se visualiza en la imagen, es "Add More Atributes" y nos permite seguir agregando atributos para el Expected Schema que estamos creando, invocando a la actividad

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

“Add more Atributes for Expected Schema”. Podemos ver en la Figura 21 la actividad "Create Expected Schema and Atributes".

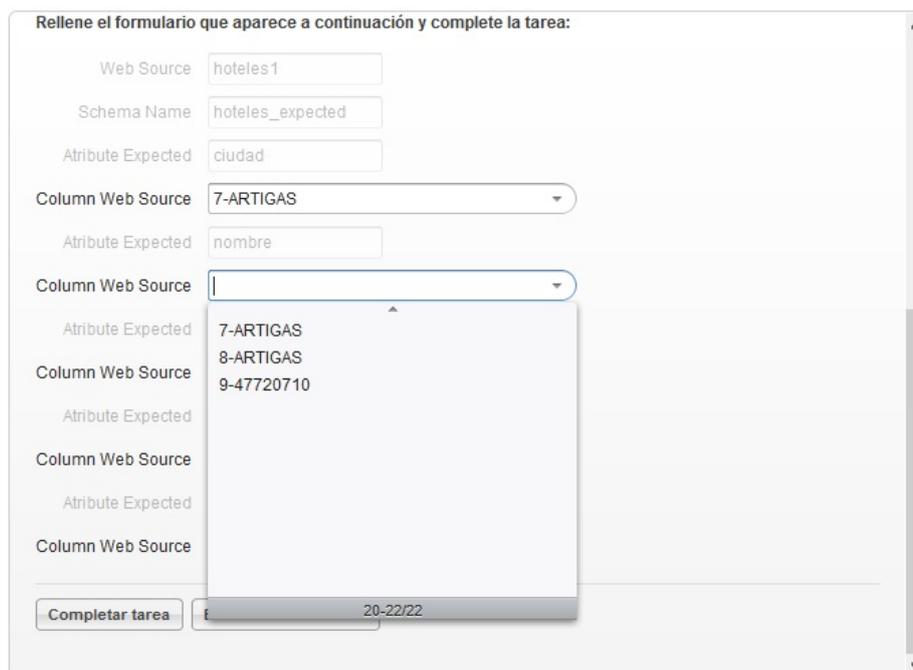


The screenshot shows a web form titled "Contenidos relacionados" with a plus icon. Below the title, it says "No hay contenidos relacionados para esta tarea". The main instruction is "Rellene el formulario que aparece a continuación y complete la tarea:". The form contains several input fields: "Web Source" with the value "hoteles1", "Schema Name" with "hoteles\_expected", and a series of "Attribute" and "Type" pairs. The first attribute is "nombre" with type "char(20)", the second is "ciudad" with type "char(20)", and there are three more empty attribute fields, each with a "char(20)" type dropdown.

Figura 21 Add more Atributes for Expected Schema

## MAPPING EXPECTED SCHEMA AND WEB SOURCE

Realizamos el mapeo entre los atributos del Expected Schema con las columnas de la Web Source que ya fueron extraídas como comentamos anteriormente.



The screenshot shows a web form with the instruction "Rellene el formulario que aparece a continuación y complete la tarea:". The form has several fields: "Web Source" (hoteles1), "Schema Name" (hoteles\_expected), "Attribute Expected" (ciudad), "Column Web Source" (7-ARTIGAS), "Attribute Expected" (nombre), and another "Column Web Source" field which is currently empty. A dropdown menu is open for the second "Column Web Source" field, showing three options: "7-ARTIGAS", "8-ARTIGAS", and "9-47720710". Below the form, there is a "Completar tarea" button and a date "20-22/22".

Figura 22 Mapping Expected Schema and Web Source.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

La Figura 22 muestra el formulario antes mencionado, donde podemos ver los distintos campos a cargar mediante combos cargados desde base de metadatos.

## UPLOAD DATA SERVICES

Ingresamos la url del Data Services que provee la extracción de los datos, como valor por defecto se despliega el del formato de la Web Source y en caso que el usuario desee cambiar el servicio lo podrá hacer (Ver Figura 23).

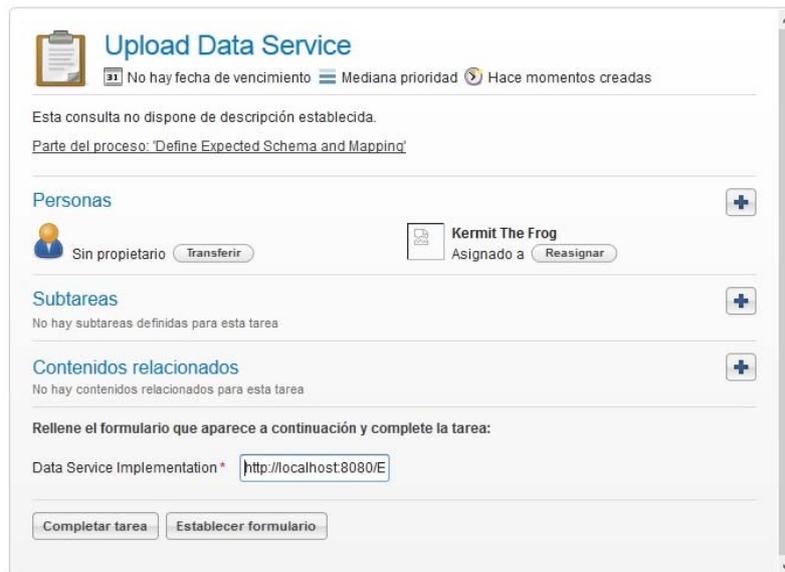


Figura 23 Upload Data Service

## DEFINE INTEGRATEDSCHEMA

La Figura 24 muestra la actividad "Define Integrated Schema" que determina de qué forma queremos crear el Integrated Schema.

Las opciones son:

- Crear el esquema manualmente, y para ello debe activar "CreateSchemaManually". Se explicará en siguientes tareas.
- Mediante un *json*, que nos permite de manera automática y con un formato determinado, generar toda la estructura referente al Integrated Schema obteniendo el mismo resultado que si lo hubiéramos creado manualmente.

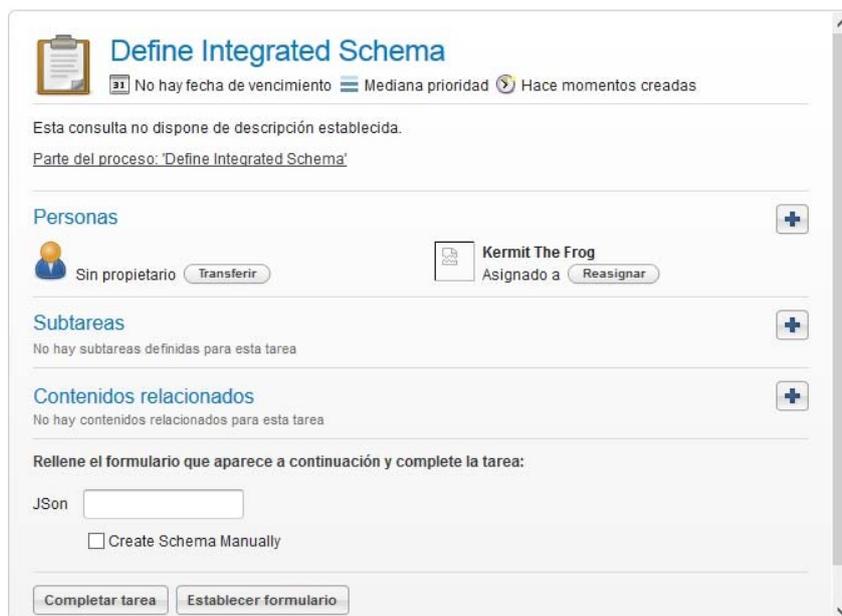


Figura 24 Define Integrated Schema.

## CREATE TABLE AND ATRIBUTES FOR INTEGRATED SCHEMA

Como mencionamos en la tarea anterior, la actividad "*Create Table and Atributes for Integrated Schema*" de la Figura 25 se ejecuta sólo si seleccionamos antes la manera manual de crear el Integrated Schema.

Aquí ingresamos los datos de las tablas del "*Integrated Schemas*" (nombre de la tabla, nombre y tipo de cada atributo y cuál es la *primary key*)

Una opción que presenta esta tarea pero no se visualiza en la imagen, es "*Add More Atributes*" y nos permite seguir agregando atributos para la tabla del IntegratedSchema que estamos creando, invocando a la actividad "*Add more Atributes for Integrated Schema*". También tenemos la opción de seguir agregando tablas al IntegratedSchema y volvemos a ejecutar la tarea "*Create Table and Atributes for Integrated Schema*" seleccionando el check "*Add More Tables*".

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

The screenshot shows a form for creating a table named 'hoteles\_integrated'. It lists several attributes, each with a text input for the attribute name, a dropdown for the data type (all set to 'char(20)'), and a checkbox for 'Is Primary Key'. The first attribute, 'nombre', has its 'Is Primary Key' checkbox checked. The other attributes are 'ciudad', and three unnamed attributes.

Figura 25 Create Table and Atributes for Integrated Schema

## SHOW INTEGRATED SCHEMA

La siguiente actividad no requiere un acción del usuario, simplemente nos permite visualizar como ha quedado nuestro Integrated Schema. La Figura 26 nos muestra como el Sistema presenta al usuario esta información.

The screenshot shows a task interface titled 'Personas'. It includes a user profile for 'Kermit The Frog' with a 'Reasignar' button. Below this are sections for 'Subtareas' and 'Contenidos relacionados', both indicating no items are present. A section titled 'Rellene el formulario que aparece a continuación y complete la tarea:' contains a scrollable list of tables and their attributes: 'ciudades\_integrated (dpto, nombre)', 'clientes\_integrated (ci, ciudad, nombre)', 'estadias\_integrated (ci\_cliente, nom\_hotel)', and 'hoteles\_integrated (ciudad, nombre)'. At the bottom are buttons for 'Completar tarea' and 'Establecer formulario'.

Figura 26 Show Integrated Schema

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

## DEFINE FOREIGN KEYS INTEGRATED SCHEMA

En esta actividad vamos a determinar claves foráneas entre las tablas del Integrated Schema.

Una vez completado el formulario, el mismo nos permite seguir agregando *foreign keys* activando “Define More Foreign Keys”.

Como muestra la Figura 27, en un combo se muestran los atributos de la tabla que estamos definiendo y en el otro las *primary keys* del resto de las tablas.

Esta actividad se ejecutará para cada tabla del Integrated Schema, teniendo la posibilidad para cada una de ellas de no determinar ninguna *foreign key*.

Rellene el formulario que aparece a continuación y complete la tarea:

Table Name: hoteles\_integrated

Attribute: ciudad

Foreign Attribute: [empty]

Attribute: ciudades\_integrated - nombre

Foreign Attribute: clientes\_integrated - ci

Attribute: estadias\_integrated - ci\_cliente

Foreign Attribute: estadias\_integrated - nom\_hotel

Attribute: [empty]

Foreign Attribute: [empty]

Attribute: [empty]

Foreign Attribute: [empty]

Define More Foreign Keys?

Completar tarea Establecer formulario

Figura 27 Define Foreign Key for Integrated Schema

## MAPPING INTEGRATED SCHEMA AND EXPECTED SCHEMA

Realizamos el mapeo entre los atributos del Integrated Schema con los atributos del Expected Schema, al final podemos decidir si queremos seguir agregando mapeos y la tarea se volverá a ejecutar.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

**Mapping Integrated Schema and Expected Schema**

No hay fecha de vencimiento Mediana prioridad Hace momentos creadas

Esta consulta no dispone de descripción establecida.

Parte del proceso: 'Mappinq Integrated Schema'

Rellene el formulario que aparece a continuación y complete la tarea:

Table Name: Alojamiento

Attribute Integrated: direccion

Attribute Expected: HotelesExp - Direccio

Attribute Integrated: nombre

Attribute Expected: HostelsExp - Nombre

Attribute Integrated: tel

Attribute Expected: [Dropdown menu open]

Attribute Expected options: CiudadesExp - Depto, CiudadesExp - Nombre, HostelsExp - Ciudad, HostelsExp - Direccion, HostelsExp - Nombre, HotelesExp - Ciudad, HotelesExp - Direccion, HotelesExp - Nombre

Completar tarea

Figura 28 Mapping Integrated Schema and Expected Schema.

La Figura 28 nos permite ver como es la actividad "Mapping Integrated Schema and Expected Schema", donde indicamos los mappings entre atributos del Integrated Schema y Expected Schema como ya mencionamos.

## DEFINE JOINS INTEGRATED SCHEMA

En la tarea de "Mapping Integrated Schema and Expected Schema" que muestra la Figura 28, se mapean distintas tablas del Expected Schema para conformar las tablas del esquema integrado. Esto implica que debemos indicarle al sistema como vamos a *joinear* esas tablas del Expected para cargar las tablas del Integrado. En este formulario debemos para cada tabla del Expected Schema de cada tabla del Integrated Schema, indicar cómo y con qué campo/s joinear entre ellas. Entonces seleccionamos el campo de la tabla de Expected que estamos iterando, en el combo "Expected Attribute", y luego seleccionamos un campo de las restantes tablas del Expected Schema en el combo "Expected Schema and Attribute" que ya no hayan sido procesados sus joins. Esto lo podemos ver en la Figura 29.

Observación: una vez que procesamos varios joins entre las tablas del Expected, es posible que nos encontremos con una tabla (que aún no procesamos) que ya fueron indicados sus joins por procesamientos de tablas anteriores. Para ellos existe el check "Done" que nos permite marcar la tabla como "ya procesada" sin indicar ningún join en la actividad.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

Define Joins Integrated Schema

No hay fecha de vencimiento Mediana prioridad Hace momentos creadas

Rellene el formulario que aparece a continuación y complete la tarea:

Integrated Schema: Alojamientos

Expected Schema: HostelsExp

Expected Attribute: Nombre

Expected Schema and Attribute: HotelesExp - Nombre

Expected Attribute: HotelesExp - Ciudad, HotelesExp - Direccion, HotelesExp - Nombre

Expected Schema and Attribute: [Empty]

Expected Attribute: [Empty]

Expected Schema and Attribute: [Empty]

Expected Attribute: [Empty]

Expected Schema and Attribute: [Empty]

Expected Attribute: [Empty]

Expected Schema and Attribute: [Empty]

Add More Joins?

Done

Completar tarea Establecer formulario

Figura 29 Define Joins Integrated Schema.

## DEFINE DATA WAREHOUSE SCHEMA

La siguiente actividad determina de qué forma queremos crear el DW Schema, esto lo podemos ver en la Figura 30.

Las opciones son:

- Crear el esquema manualmente, y para ello debe activar "CreateSchemaManually". Se explicará en siguientes tareas.
- Mediante un *json*, que nos permite de manera automática y con un formato determinado, generar toda la estructura referente al DW Schema obteniendo el mismo resultado que si lo hubiéramos creado manualmente.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

Esta consulta no dispone de descripción establecida.  
Parte del proceso: 'Define DW Schema And Mapping'

**Personas** +

Sin propietario **Transferir** Kermit The Frog  
Asignado a **Reasignar**

**Subtareas** +  
No hay subtareas definidas para esta tarea

**Contenidos relacionados** +  
No hay contenidos relacionados para esta tarea

Rellene el formulario que aparece a continuación y complete la tarea:

JSON

Name Schema

Create Schema Manually

**Completar tarea** **Establecer formulario**

Figura 30 Define DW Schema and Mapping

## CREATE TABLE AND ATRIBUTES FOR DATA WAREHOUSE SCHEMA

Como mencionamos en la tarea anterior, la actividad "Create Table and Atributes for Data Warehouse Schema" se ejecuta sólo si seleccionamos antes la manera manual de crear el DW Schema.

Aquí ingresamos los datos de las tablas del "DWSchemas" (indicando el nombre de la tabla junto con el nombre y tipo de cada atributo con su clave primaria y si es tabla de hechos o no).

Una opción que presenta esta tarea pero no se visualiza en la imagen, es "Add More Atributes" y nos permite seguir agregando atributos para la tabla del DW Schema que estamos creando, invocando a la actividad "Add more Atributes for Data Warehouse Schema".

También tenemos la opción de seguir agregando tablas al Data Warehouse Schema y volvemos a ejecutar la tarea "Create Table and Atributes for Data Warehouse Schema" seleccionando el check "Add More Tables". Todas las características de esta actividad la podemos ver en la Figura 31.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

Nombre Sistema: datawarehouse

Name Table: Hoteles

Atribute: nombre

Type: char(20)

Is Primary Key?

Atribute: ciudad

Type: char(20)

Is Primary Key?

Atribute:

Type: char(20)

Is Primary Key?

Atribute:

Type: char(20)

Is Primary Key?

Atribute:

Type: char(20)

Is Primary Key?

Is Fact Table

Figura 31 Create Table and Atributes for Data Warehouse Schema.

## SHOW DW SCHEMA

La siguiente actividad no requiere un acción del usuario, simplemente nos permite visualizar como ha quedado nuestro DW Schema (Ver Figura 32).

Personas

Sin propietario [Transferir](#)

Kermit The Frog  
Asignado a [Reasignar](#)

Subtareas  
No hay subtareas definidas para esta tarea

Contenidos relacionados  
No hay contenidos relacionados para esta tarea

Rellene el formulario que aparece a continuación y complete la tarea:

- Ciudades (departamento, nombre)
- Clientes (ci, ciudad, nombre)
- Estadias (ci\_cliente, nombre\_hotel)
- Hoteles (ciudad, nombre)

[Completar tarea](#) [Establecer formulario](#)

Figura 32 Show DW Schema.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

## DEFINE FOREIGN KEYS OF FACT TABLES FOR DATA WAREHOUSE SCHEMA

En esta actividad vamos a determinar claves foráneas entre las tablas del DW Schema.

Como muestra la Figura 33, en un combo se muestran los atributos de la tabla que estamos definiendo y en el otro las *primary keys* del resto de las tablas de dimensiones.

Una vez completado el formulario, el mismo nos permite seguir agregando *foreign keys* activando “Define More Foreign Keys”.

Esta actividad se ejecutará para cada tabla de hechos (*fact table*) del DW Schema.

Contenido relacionado

No hay contenidos relacionados para esta tarea

Rellene el formulario que aparece a continuación y complete la tarea:

Fact Table: Estadias

Attribute Fact Table: ci\_cliente

Foreign Attribute: Clientes-ci

Attribute Fact Table: Ciudades-nombre

Foreign Attribute: Clientes-ci

Attribute Fact Table: Hoteles-nombre

Foreign Attribute: [Empty]

Attribute Fact Table: [Empty]

Foreign Attribute: [Empty]

Attribute Fact Table: [Empty]

Foreign Attribute: [Empty]

Add More Foreign Keys

Completar tarea Establecer formulario

Figura 33 Define Foreign Keys of Fact Tables for Data Warehouse Schema

## MAPPING DATA WAREHOUSE SCHEMA AND INTEGRATED SCHEMA

Realizamos el mapeo entre los atributos del DW Schema con los atributos del Integrated Schema. Ilustramos esto con la Figura 34.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

Rellene el formulario que aparece a continuación y complete la tarea:

Name Schema	<input type="text" value="datawarehouse"/>
Name Table	<input type="text" value="Hoteles"/>
Atribute DataWarehouse	<input type="text" value="ciudad"/>
Atribute Integrated	<input type="text" value="hoteles_integrated-ciudad"/>
Atribute DataWarehouse	<input type="text" value="nombre"/>
Atribute Integrated	<input type="text" value=""/>
Atribute DataWarehouse	<input type="text" value=""/>
Atribute Integrated	<input type="text" value=""/>
Atribute DataWarehouse	<input type="text" value=""/>
Atribute Integrated	<input type="text" value=""/>
Atribute DataWarehouse	<input type="text" value=""/>
Atribute Integrated	<input type="text" value=""/>
Atribute DataWarehouse	<input type="text" value=""/>
Atribute Integrated	<input type="text" value=""/>

ciudades\_integrated-dpto  
ciudades\_integrated-nombre  
clientes\_integrated-ci  
clientes\_integrated-ciudad  
clientes\_integrated-nombre  
estadias\_integrated-ci\_cliente  
estadias\_integrated-nom\_hotel  
hoteles\_integrated-ciudad  
hoteles\_integrated-nombre

Figura 34 Mapping Data Warehouse Schema and Integrated Schema.

## DEFINE JOINS DW SCHEMA

En la tarea de “Mapping Data Warehouse Schema and Integrated Schema” se mapean distintas tablas del Integrated Schema para conformar las tablas del esquema de DW. Esto implica que debemos indicarle al sistema como vamos a *joinear* esas tablas del Integrated para cargar las tablas del DW Schema. En este formulario (ver Figura 35) debemos para cada tabla del Integrated Schema de cada tabla del DWSchema, indicar cómo y con qué campo/s joinear entre ellas. Entonces seleccionamos el campo de la tabla de Integrated que estamos iterando, en el combo “Integrated Atribute”, y luego seleccionamos un campo de las restantes tablas del Integrated Schema en el combo “Integrated Schema and Atribute” que ya no hayan sido procesados sus joins.

Observación: una vez que procesamos varios joins entre las tablas del Integrated, es posible que nos encontremos con una tabla (que aún no procesamos) que ya fueron indicados sus joins por procesamientos de tablas anteriores. Para ellos existe el check “Done” que nos permite marcar la tabla como “ya procesada” sin indicar ningún join en la actividad.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

**Define Joins DW Schema**  
No hay fecha de vencimiento | Mediana prioridad | Hace momentos creadas

Esta consulta no dispone de descripción establecida.  
Parte del proceso: 'Configuration Process'

**Personas**  
Sin propietario (Transferir) | Kermit The Frog (Asignado a) (Reasignar)

**Subtareas**  
No hay subtareas definidas para esta tarea

**Contenidos relacionados**  
No hay contenidos relacionados para esta tarea

Rellene el formulario que aparece a continuación y complete la tarea:

DW Schema Table	<input type="text" value="turismo"/>
Integrated Schema Table	<input type="text" value="actividades_integrated"/>
Integrated Atribute	<input type="text" value="ciudad"/>
Integrated Schema Table and Atribute	<input type="text" value="alojamientos_integrated - ciudad"/>
Integrated Atribute	<input type="text" value="dpto"/>
Integrated Schema Table and Atribute	<input type="text" value="alojamientos_integrated - dpto"/>
Integrated Atribute	<input type="text"/>

Figura 35 Define Joins DW Schema

## Descripción de Proceso de Carga

### SELECT CONFIGURATION

Todas las ejecuciones del proceso de Configuración, se guardan en la base de metadatos con un identificador. Como ya dijimos en la tabla "Configuration".

Apenas comienza el proceso de Carga, debemos seleccionar cuál de ellas vamos a utilizar (ver Figura 36).

**Select Configuration**  
No hay fecha de vencimiento | Mediana prioridad | Hace momentos creadas

Esta consulta no dispone de descripción establecida.  
Parte del proceso: 'Feeding Process'

**Personas**  
Sin propietario (Transferir) | Kermit The Frog (Asignado a) (Reasignar)

**Subtareas**  
No hay subtareas definidas para esta tarea

**Contenidos relacionados**  
No hay contenidos relacionados para esta tarea

Rellene el formulario que aparece a continuación y complete la tarea:

Seleccione la configuración que desea cargar \*

(dropdown menu with options 1, 2, 3)

Figura 36 Select Configuration

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

## PROBLEMS SOLVING

Esta tarea será implementada en próximas versiones, la idea de esta tarea es resolver los problemas que surjan de la extracción tomando en cuenta los metadatos de calidad obtenidos en el proceso de configuración.

## CONFLICTS RESOLUTION FOR EXPECTED TABLES

Esta tarea será implementada en próximas versiones, la idea de esta tarea es resolver los conflictos que surjan en la carga de los Expected Schemas tomando en cuenta los metadatos de calidad obtenidos en el proceso de configuración.

## CONFLICTS RESOLUTION FOR INTEGRATED TABLES

Esta tarea será implementada en próximas versiones, la idea de esta tarea es resolver los conflictos que surjan en la carga de los Integrated Schemas tomando en cuenta los metadatos de calidad obtenidos en el proceso de configuración.

## CONFIRM QUERY TO LOAD DW FACT TABLES

Esta tarea muestra al usuario como se cargara cada tabla de hechos, teniendo la posibilidad de modificarla. La Figura 37 muestra la actividad "Confirme Query to Load DW Fact Tables".

**Confirm Query to Load DW Fact Tables**  
No hay fecha de vencimiento Mediana prioridad Hace momentos creadas

Esta consulta no dispone de descripción establecida.  
Parte del proceso: 'Feeding Process'

**Personas** +

Sin propietario Transferir Kermit The Frog Asignado a Reasignar

**Subtareas** +  
No hay subtareas definidas para esta tarea:

**Contenidos relacionados** +  
No hay contenidos relacionados para esta tarea:

Rellene el formulario que aparece a continuación y complete la tarea:

Fact Table turismo

```
INSERT INTO turismo (cod_actividad,cod_alojamiento,cod_servicio,fecha)
SELECT
actividades_integrated.nombre,alojamientos_integrated.codigo,transportes_integrated.codigo,fechas_integrated.
```

Completar tarea Establecer formulario

Figura 37 Confirm Query to Load DW Fact Tables

## COMMIT LOAD OF DATA WAREHOUSE

Finalmente, esta tarea permite al usuario decidir si la carga de cada tabla de hechos es correcta y desea que sea definitiva o desea cargarla nuevamente. (Ver Figura 38)

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

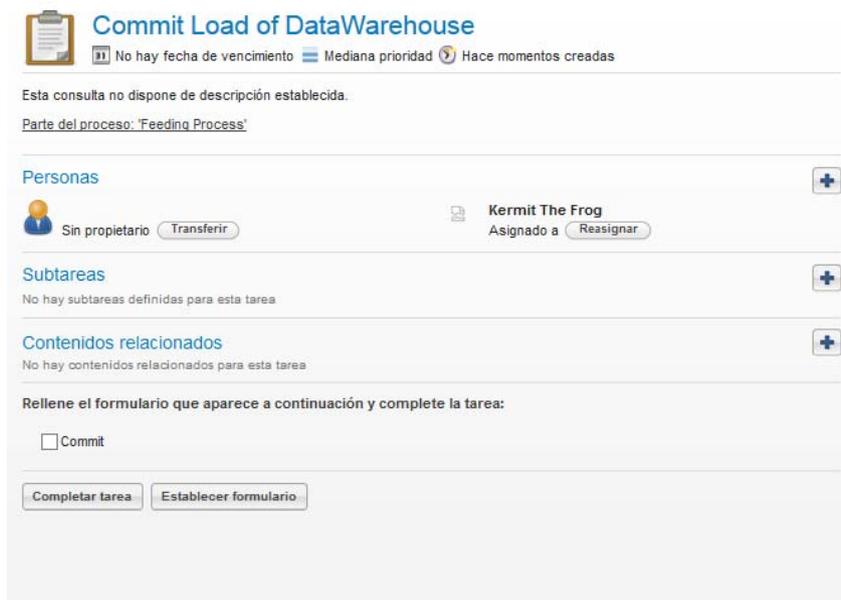


Figura 38 Commit Load of Data Warehouse.

A continuación, a modo de ejemplo y para facilitar el uso para los usuarios se presentan ejemplos de json para la carga de los esquemas de los “Expected Schema”, “Integrated Schema” y “DW Schema”:

“Expected Schema”  

```
{ "nombre": "alojamientos_expected", "atributos": [ { "nombre": "codigo", "tipo": "char(100)" }, { "nombre": "nombre", "tipo": "char(100)" }, { "nombre": "cod_ciudad", "tipo": "char(100)" }, { "nombre": "nom_dpto", "tipo": "char(100)" }, { "nombre": "direccion", "tipo": "char(100)" }, { "nombre": "telefono", "tipo": "char(100)" } ] }
```

“Integrated Schema”  

```
{ "nombre": "esquema_integrado", "tablas": [ { "nombre": "alojamientos_integrado", "atributos": [ { "nombre": "codigo", "tipo": "char(100)", "pk": true }, { "nombre": "nombre", "tipo": "char(100)", "pk": false }, { "nombre": "ciudad", "tipo": "char(100)", "pk": false }, { "nombre": "dpto", "tipo": "char(100)", "pk": false }, { "nombre": "poblacion_dpto", "tipo": "char(100)", "pk": false }, { "nombre": "direccion", "tipo": "char(100)", "pk": false }, { "nombre": "telefono", "tipo": "char(100)", "pk": false } ] }, { "nombre": "actividades_integrado", "atributos": [ { "nombre": "nombre", "tipo": "char(100)", "pk": true }, { "nombre": "ciudad", "tipo": "char(100)", "pk": false }, { "nombre": "dpto", "tipo": "char(100)", "pk": false }, { "nombre": "duracion", "tipo": "int", "pk": false }, { "nombre": "fecha", "tipo": "char(100)", "pk": false }, { "nombre": "tipo", "tipo": "char(100)", "pk": false } ] }, { "nombre": "transportes_integrado", "atributos": [ { "nombre": "codigo", "tipo": "char(100)", "pk": true }, { "nombre": "empresa", "tipo": "char(100)", "pk": false }, { "nombre": "ciudad_origen", "tipo": "char(100)", "pk": false }, { "nombre": "dpto_origen", "tipo": "char(100)", "pk": false }, { "nombre": "ciudad_destino", "tipo": "char(100)", "pk": false }, { "nombre": "dpto_destino", "tipo": "char(100)", "pk": false }, { "nombre": "hora_salida", "tipo": "char(100)", "pk": false }, { "nombre": "hora_llegada", "tipo": "char(100)", "pk": false } ] }, { "nombre": "fechas_integrado", "atributos": [ { "nombre": "fecha", "tipo": "char(100)", "pk": true }, { "nombre": "mes", "tipo": "char(100)", "pk": false }, { "nombre": "anio", "tipo": "char(100)", "pk": false } ] } ] }
```

“DW Schema”  

```
{ "nombre": "esquema_dw", "tablas": [ { "nombre": "alojamientos", "atributos": [ { "nombre": "codigo", "tipo": "char(100)", "pk": true }, { "nombre": "nombre", "tipo": "char(100)", "pk": false }, { "nombre": "direccion", "tipo": "char(100)", "pk": false }, { "nombre": "telefono", "tipo": "char(100)", "pk": false }, { "nombre": "ciudad", "tipo": "char(100)", "pk": false } ] } ] }
```

## ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

```
alse},{\"nombre\":\"departamento\",\"tipo\":\"char(100)\",\"pk\":false}},\"is_fact\":false},{\"nombre\":\"actividades\",\"atributos\":{\"nombre\":\"codigo\",\"tipo\":\"char(100)\",\"pk\":true},{\"nombre\":\"nombre\",\"tipo\":\"char(100)\",\"pk\":false},{\"nombre\":\"tipo\",\"tipo\":\"char(100)\",\"pk\":false}},\"is_fact\":false},{\"nombre\":\"transportes\",\"atributos\":{\"nombre\":\"codigo\",\"tipo\":\"char(100)\",\"pk\":true},{\"nombre\":\"empresa\",\"tipo\":\"char(100)\",\"pk\":false},{\"nombre\":\"origen\",\"tipo\":\"char(100)\",\"pk\":false},{\"nombre\":\"destino\",\"tipo\":\"char(100)\",\"pk\":false},{\"nombre\":\"hora_salida\",\"tipo\":\"char(100)\",\"pk\":false},{\"nombre\":\"hora_llegada\",\"tipo\":\"char(100)\",\"pk\":false}},\"is_fact\":false},{\"nombre\":\"turismo\",\"atributos\":{\"nombre\":\"cod_alojamiento\",\"tipo\":\"char(100)\",\"pk\":true},{\"nombre\":\"fecha\",\"tipo\":\"char(100)\",\"pk\":true},{\"nombre\":\"cod_actividad\",\"tipo\":\"char(100)\",\"pk\":true},{\"nombre\":\"cod_servicio\",\"tipo\":\"char(100)\",\"pk\":true}},\"is_fact\":true},{\"nombre\":\"fechas\",\"atributos\":{\"nombre\":\"fecha\",\"tipo\":\"char(100)\",\"pk\":true},{\"nombre\":\"mes\",\"tipo\":\"char(100)\",\"pk\":false},{\"nombre\":\"anio\",\"tipo\":\"char(100)\",\"pk\":false}},\"is_fact\":false}}
```

## Anexo C: Manual técnico

En este anexo se explicara y detallara la implementación de la solución del sistema planteado.

### Modelo de Dominio

La Figura 39 contiene el diagrama del Modelo de Dominio de la solución propuesta. En ella podemos ver las distintas entidades, las relaciones entre ellas, etc. Todos estos objetos refieren a los metadatos generados por el proceso de Configuración y dichas entidades serán de utilidad cuando queramos ejecutar el proceso de Carga.

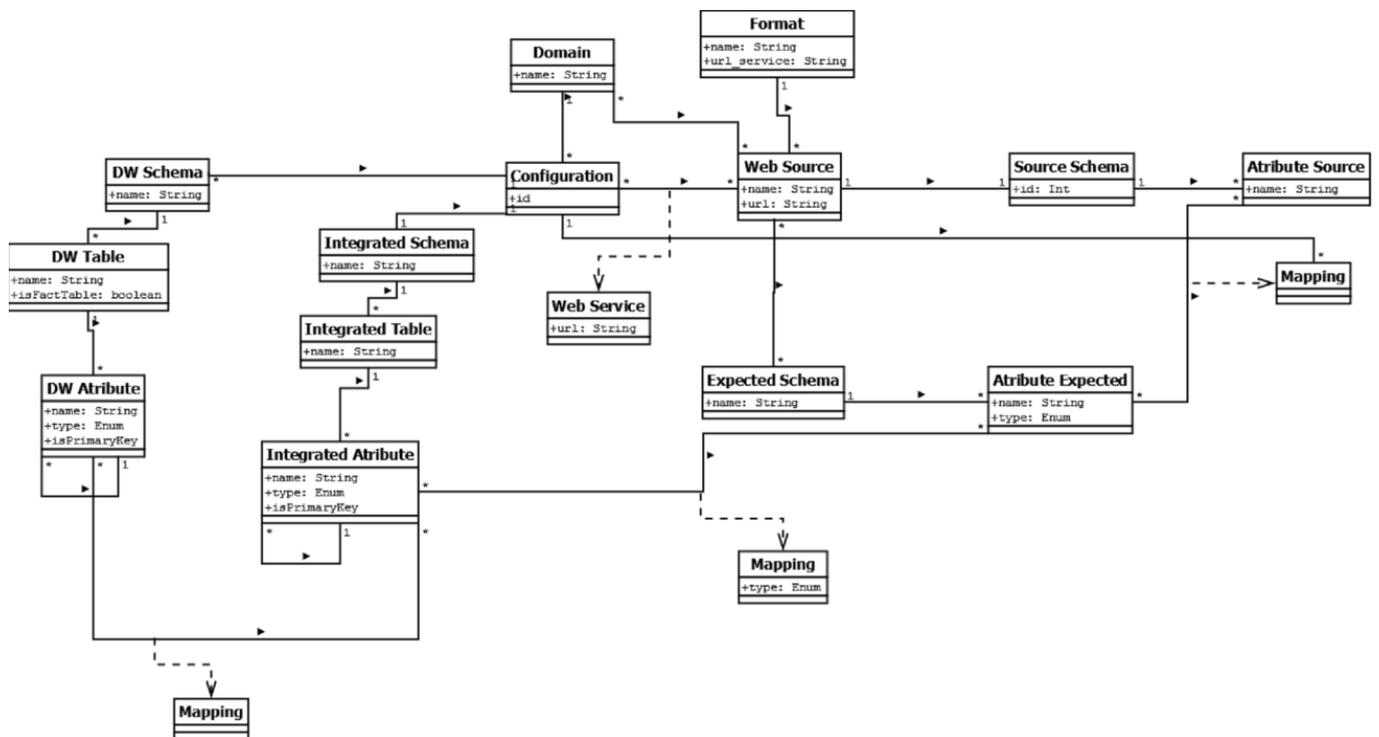


Figura 39 Modelo de Dominio de la solución propuesta.

### Modelo de Base de Metadatos

La Figura 40 contiene el diagrama del Modelo de Base de Metadatos de la solución propuesta. En el mismo se especifican las tablas pertenecientes a la misma, los atributos de cada una de ellas (con sus tipo de dato) y las relaciones entre las tablas. Toda esta estructura será de constante consulta cuando estemos cargando nuestro DW, ya que la forma de cargarlo estará determinada por el contenido de estas tablas, esto es, los metadatos mencionados anteriormente serán los alojados en estas estructuras.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

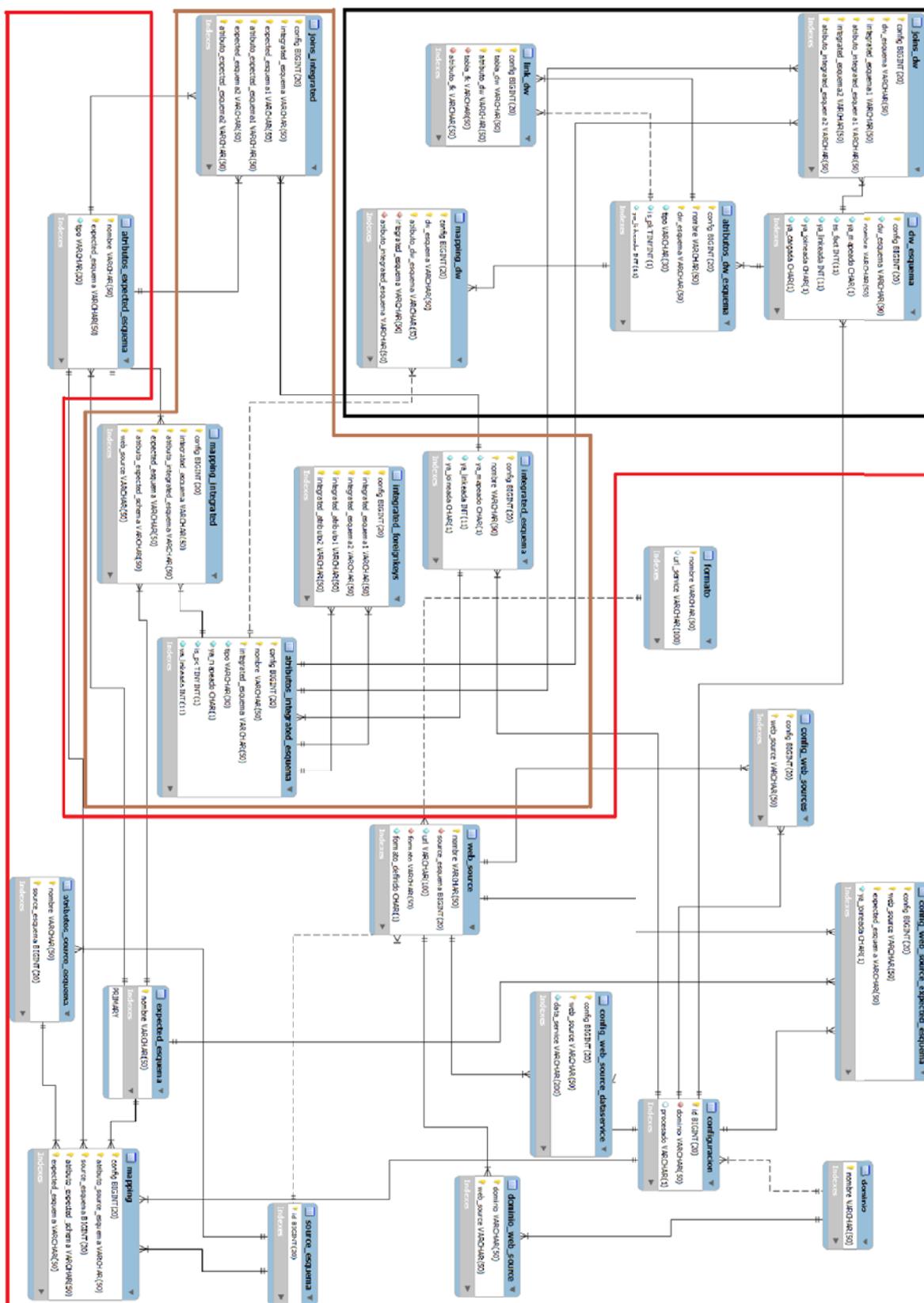


Figura 40 Modelo de Base de Metadatos de la solución propuesta.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

## **Resumen**

La solución del sistema está compuesta por un proyecto dividido en 6 paquetes que son los siguientes:

- 1) *"base"* que contiene una clase llamada *"ControladorBD"* la cual encapsula todas las llamadas a las bases de datos (la base de metadatos del sistema y las bases de datos correspondientes a los WW construidos).
- 2) *"constant"* que contiene la clase llamada *"FlowConstant"*, esta almacena todas las variables de los flujos de los procesos y subprocessos utilizados en la solución.
- 3) *"customTypes"*, en este paquete se encuentran todos los tipos de datos creados para usar como combos o como campos de textos extensos dentro del Activiti, los cuales explicaremos más en detalle.
- 4) *"dtos"* contiene los datatypes utilizados.
- 5) *"procesos"*, este paquete tiene todas las clases que resuelven las acciones realizadas por cada actividad de los procesos
- 6) *"services"* que contiene los servicios utilizados en la solución, como ser la extracción de la primera fila de los datos publicados en la web con su tipo correspondiente (sea la fila de títulos o la primera fila de datos en caso que no existan títulos).

A continuación explicaremos generalidades de la implementación para luego ver proceso por proceso y actividad por actividad que se realiza en cada una.

El servidor utilizado para realizar la solución es el apache Tomcat 8.0, Eclipse Kepler con los plugins necesarios para utilizar Activiti (Activiti Eclipse BPMN 2.0 Designer 5.15). El motor de base de datos es MySQL 5.5.

Existe una configuración previa para el funcionamiento del Activiti que es la configuración de la base de datos y de los tipos de datos. Esto es en la carpeta webapps donde se *deployan* los *war*, en el archivo *"db.properties"* (activiti-explorer\WEB-INF\classes) se configura la conexión a la base del Activiti:

```
db=mysql
```

```
jdbc.driver=com.mysql.jdbc.Driver
```

```
jdbc.url=jdbc:mysql://localhost:3306/activiti
```

```
jdbc.username=root
```

```
jdbc.password=root
```

Por otro lado *"activiti-standalone-context.xml"* (activiti-explorer\WEB-INF) en el *tag* *"property"* se detallan los tipos creados por el desarrollador de la siguiente manera:

```
<property name="customFormTypes">
```

```
<list>
```

## ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

```
<bean class="customTypes.UrlFormType" />
```

```
</list>
```

```
</property>
```

Y finalmente en el archivo "activiti-ui-context.xml" (activiti-explorer\WEB-INF) en el tag "property" se especifica el *Renderer* del tipo *customizado* de la siguiente manera:

```
<property name="propertyRenderers">
```

```
<list>
```

```
<bean class="customTypes.UrlFormPropertyRenderer" />
```

```
</list>
```

```
</property>
```

Como explicamos anteriormente la clase "ControladorBD" contiene todas las operaciones que acceden a las bases de datos, tanto de metadatos generados por el proceso de configuración como los datos de las bases correspondientes a los DW construidos.

La clase "FlowConstant" contiene las variables del flujo que van a ser accedidas luego desde cada clase que implementan las actividades.

En cuanto al paquete de tipos de datos, cada uno de estos cuenta con una clase que extiende "AbstractFormType" el cual contiene una constante llamada "TYPE\_NAME" que indica el nombre del tipo que será utilizado luego en el modelo del proceso. Y además contiene otra clase que de la misma forma extiende "AbstractFormPropertyRenderer" y sobrescribiendo el método "getPropertyField" de forma de *customizar* nuestro tipo deseado (estas clases son las que se parametrizan de la manera descrita anteriormente).

Por otra parte, en el paquete "services" tenemos clases de servicios que se utilizan en la implementación, las más importantes son: "ExtractSourceSchema" (esta clase extrae de la fuente web el título de sus columnas o en su defecto la primer tupla de datos) y "ParseExpectedSchema" (el cual se encarga de *parsear* el *json* con las definiciones de los esquemas)

Finalmente el paquete "procesos" contiene las clases utilizadas en las actividades, además de las clases que resuelven la funcionalidad de cada actividad, hay una clase llamada *LimpiarDatosListener* la cual es llamada al completar cada actividad que se encarga de resetear los valores de las variables que correspondan.

Las clases utilizadas en actividades manuales implementan *TaskListener* y *ExecutionListener* y las utilizadas en tareas automáticas implementan *JavaDelegate* sobrescribiendo los métodos que correspondan en cada caso.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

Otro punto a tener en cuenta, luego de generar el *jar* del proyecto y el bar del modelo (esto es desde la vista de Activiti del eclipse click derecho en el proyecto y "Create deployment artifacts") este *jar* se debe poner en activiti-explorer\WEB-INF\lib y en la sección de "Gestionar" -> "Despliegue" -> "Cargar nueva" se sube el .bar correspondiente a los modelos generados.

## Especificación de funcionalidad y uso

A continuación explicaremos en cada proceso y subprocessos que se realiza en cada actividad, teniendo en cuenta que las principales características de las actividades se encuentran en la pestañas General (id y nombre), *Form* (los campos con sus tipos y sus propiedades) *Listeners* (las clases invocadas y en qué momento, por ejemplo *complete*).

## Descripción de Proceso de Configuración

La Figura 41 muestra el proceso de Configuración completo, con sus subprocessos y actividades principales.

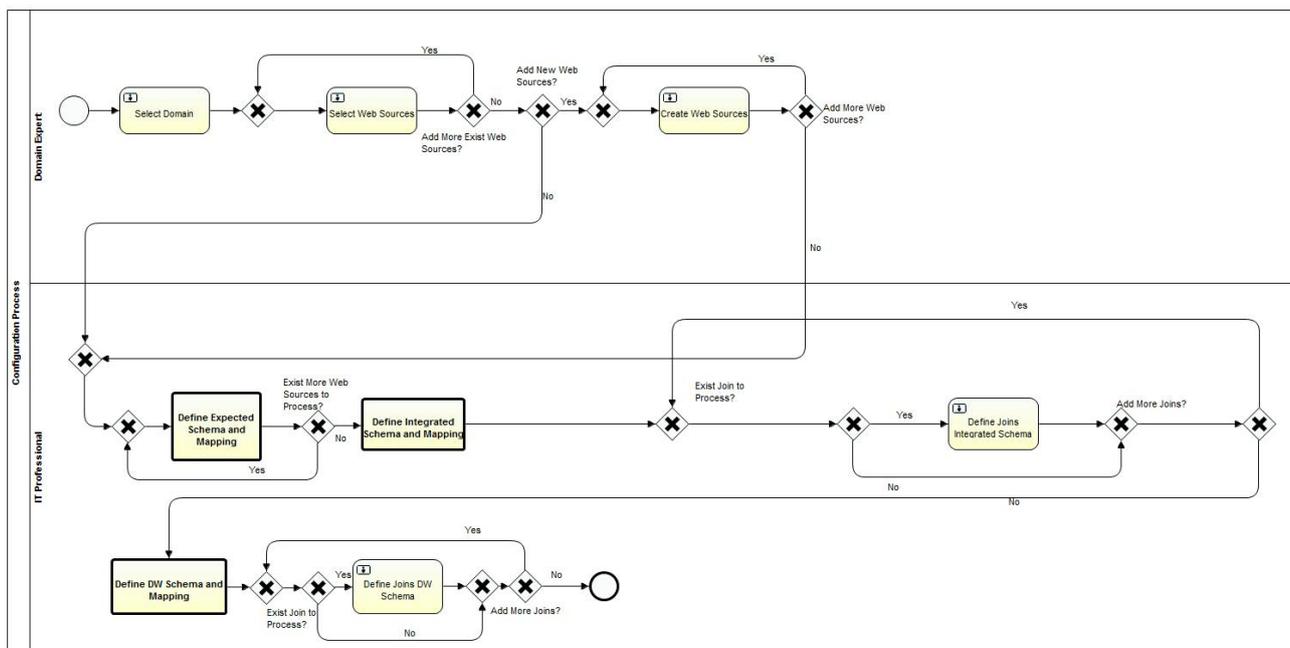


Figura 41 Proceso de Configuración.

## Subproceso "Define Expected Schema and Mapping"

La Figura 42 muestra el proceso "Define Expected Schema and Mapping", contiene una actividad automática y el resto actividades manuales, no contiene subprocessos. Este sub proceso es invocado desde el proceso de Configuración (Ver Figura 41)

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

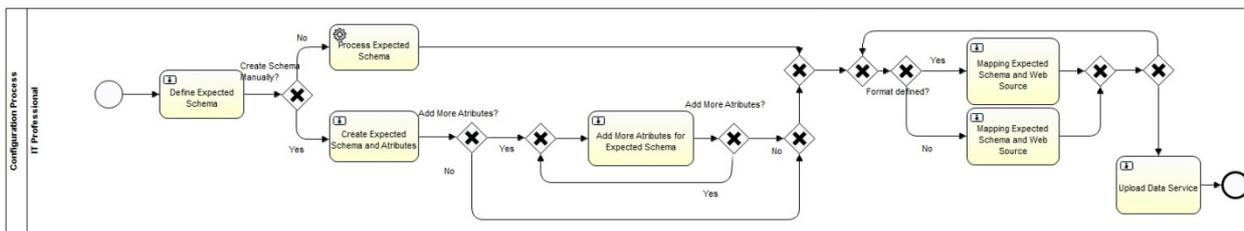


Figura 42 Define Expected Schema and Mapping.

## Subproceso “Define Integrated Schema and Mapping”

La Figura 43 nos permite visualizar el subproceso "Define Integrated Schema and Mapping", este subproceso es invocado desde el proceso de Configuración. (Ver Figura 41)

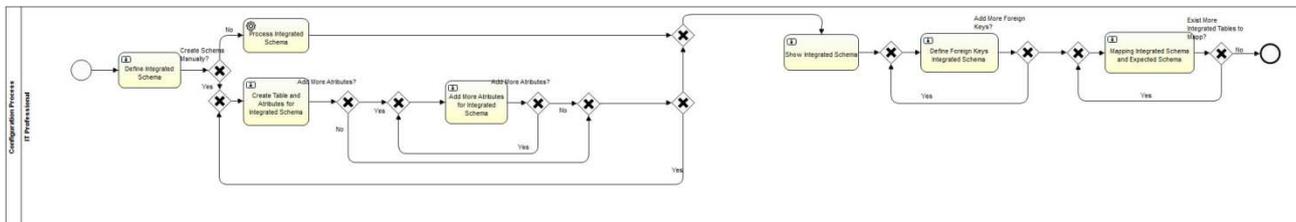


Figura 43 Define Integrated Schema and Mapping.

## Subproceso “Define DW Schema and Mapping”

En la Figura 44 podemos ver completamente el subproceso "Define DW Schema and Mapping" invocado desde el proceso principal de Configuración. La primera parte consiste en la definición propiamente del esquema y la segunda parte corresponde al mapping con el Integrated Schema.

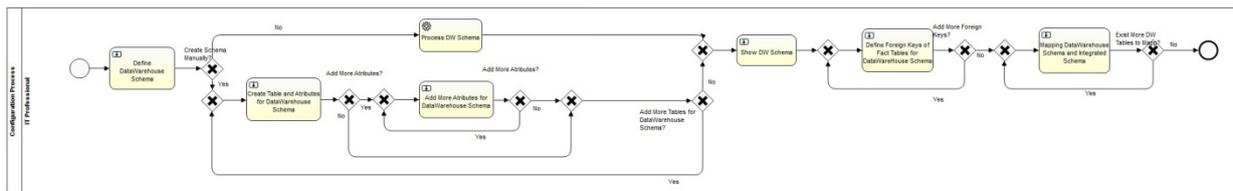


Figura 44 Subproceso Define DW Schema and Mapping.

Ahora explicaremos en que consiste técnicamente cada actividad, teniendo en cuenta la lectura previa del manual de usuario de modo de entender conceptualmente que se realiza en cada actividad.

### SELECT DOMAIN

Cuando la tarea se completa, se inserta en la tabla “Configurations” una nueva configuración y una referencia al dominio seleccionado.

### SELECT WEB SOURCE

La tabla que guarda los registros asociados es “WebSources”, de esta forma, generamos una relación entre nuestro proceso de configuración y las Web Source ya existentes de ejecuciones anteriores.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

Dicha relación se inserta en la tabla "ConfigWebSources".

## CREATE WEB SOURCE

Se graban las tablas "WebSources" para alojar las fuentes web y "DomainWebSources" que indica la relación entre éstas y el dominio. Esto nos permite en futuras ejecuciones de este proceso, tener precargadas las Web Sources para determinado dominio.

Al igual que en la actividad anterior, insertamos en "ConfigWebSources" para vincularlas.

Además de esto, se realiza un pequeño procesamiento de los datos provistos por cada web, en donde guardamos los nombres de cada columna de datos si es que estos existen, o de lo contrario, la primer fila de datos. Esto se utilizara posteriormente en el mapeo con las columnas de cada Web Source.

Para esto, se grabaran las tablas "SourceSchemas" y "AttributesSourceSchemas".

## DEFINE EXPECTED SCHEMA

Las opciones son:

- Crear el esquema manualmente, y para ello debe activar "CreateSchemaManually". Se explicará en siguientes tareas. En este caso no se realizan cambios en la base de metadatos, sólo es una condición de flujo.
- Seleccionando un Expected Schema creado anteriormente. En este caso se realiza una inserción en la tabla "ConfigWebSourceExpectedSchemas".
- Mediante un *json*, que nos permite de manera automática y con un formato determinado, generar toda la estructura referente al Expected Schema obteniendo el mismo resultado que si lo hubiéramos creado manualmente.

## CREATE EXPECTED SCHEMA AND ATRIBUTES

Aquí ingresamos los datos y se guardaran en las tablas "Expected Schemas", "AttributesExpectedSchemas" y "ConfigWebSourceExpectedSchemas".

Una opción que presenta esta tarea pero no se visualiza en la imagen, es "Add More Atributes" y nos permite seguir agregando atributos para el Expected Schema que estamos creando, invocando a la actividad "Add more Atributes for Expected Schema". Este flujo se puede visualizar en las imágenes de los procesos y subprocesos.

## MAPPING EXPECTED SCHEMA AND WEB SOURCE

Para esto se grabará la tabla "MappingSourceExpected" por cada mapeo.

## UPLOAD DATA SERVICES

Como valor por defecto, se despliega el valor "url\_services" del formato de la Web Source.

En caso que el usuario desee cambiar el servicio lo podrá hacer, pero en todos los casos deja registro del valor de la url en la tabla "ConfigWebSource".

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

## DEFINE INTEGRATED SCHEMA

Las opciones son:

- Crear el esquema manualmente, y para ello debe activar "CreateSchemaManually". Se explicará en siguientes tareas. En este caso no se realizan cambios en la base de metadatos, sólo es una condición de flujo.
- Mediante un *json*, que nos permite de manera automática y con un formato determinado, generar toda la estructura referente al Integrated Schema obteniendo el mismo resultado que si lo hubiéramos creado manualmente.

## CREATE TABLE AND ATRIBUTES FOR INTEGRATED SCHEMA

Aquí ingresamos los datos y se guardaran en las tablas "IntegratedSchemas", "AtributesIntegratedSchemas". En donde se insertarán las tablas y atributos del esquema respectivamente.

Una opción que presenta esta tarea pero no se visualiza en la imagen, es "Add More Atributes" y nos permite seguir agregando atributos para la tabla del IntegratedSchema que estamos creando, invocando a la actividad "Add more Atributes for Integrated Schema". Este flujo se puede visualizar en las imágenes de los procesos y subprocessos.

También tenemos la opción de seguir agregando tablas al IntegratedSchema y volvemos a ejecutar la tarea "Create Table and Atributes for Integrated Schema" seleccionando el check "Add More Tables"

## SHOW INTEGRATED SCHEMA

La siguiente actividad no requiere un acción del usuario, simplemente nos permite visualizar como ha quedado nuestro Integrated Schema sin modificarse nada en la base de metadatos.

## DEFINE FOREIGN KEYS INTEGRATED SCHEMA

Dichas referencias se grabarán en la tabla "IntegratedForeignKeys".

Una vez completado el formulario, el mismo nos permite seguir agregando *foreign keys* activando "Define More Foreign Keys".

## MAPPING INTEGRATED SCHEMA AND EXPECTED SCHEMA

Para esto se grabará la tabla "MappingExpectedIntegrated" por cada mapeo.

## DEFINE JOINS INTEGRATED SCHEMA

En la tarea de "Mapping Integrated Schema and Expected Schema" se mapean distintas tablas del Expected Schema para conformar las tablas del esquema integrado. Esto implica que debemos indicarle al sistema como vamos a *joinear* esas tablas del Expected para cargar las tablas del Integrado. En este formulario debemos para cada tabla del Expected Schema de cada tabla del Integrated Schema, indicar cómo y con qué campo/s joinear entre ellas. Entonces seleccionamos el campo de la tabla de Expected que estamos iterando, en el combo "Expected Attribute", y luego seleccionamos un campo de las restantes tablas del Expected Schema en el combo "Expected Schema and Attribute" que ya no hayan sido procesados sus joins.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

Luego de ejecutar esta actividad se grabará una tabla en la base de metadatos que se llama "Joins\_Integrated" donde cada registro representará un join de los mencionados.

## DEFINE DATA WAREHOUSE SCHEMA

Las opciones son:

- Crear el esquema manualmente, y para ello debe activar "CreateSchemaManually". Se explicará en siguientes tareas. En este caso no se realizan cambios en la base de metadatos, sólo es una condición de flujo.
- Mediante un *json*, que nos permite de manera automática y con un formato determinado, generar toda la estructura referente al DW Schema obteniendo el mismo resultado que si lo hubiéramos creado manualmente.

## CREATE TABLE AND ATRIBUTOS FOR DATA WAREHOUSE SCHEMA

Aquí ingresamos los datos y se guardaran en las tablas "DWSchemas", "AtributosDWSchemas". En donde se insertarán las tablas y atributos del esquema respectivamente.

Una opción que presenta esta tarea pero no se visualiza en la imagen, es "Add More Atributes" y nos permite seguir agregando atributos para la tabla del DW Schema que estamos creando, invocando a la actividad "Add More Atributes for Data Warehouse Schema". Este flujo se puede visualizar en las imágenes de los procesos y subprocesos.

En este formulario, debemos determinar si la tabla del DW Schema es FactTable en el mismo.

También tenemos la opción de seguir agregando tablas al DW Schema y volvemos a ejecutar la tarea "Create Table and Atributos for Data Warehouse Schema" seleccionando el check "Add More Tables".

## SHOW DW SCHEMA

La siguiente actividad no requiere un acción del usuario, simplemente nos permite visualizar como ha quedado nuestro DW Schema, sin modificarse nada en la base de metadatos.

## DEFINE FOREIGN KEYS OF FACT TABLES FOR DATA WAREHOUSE SCHEMA

Dichas referencias se grabarán en la tabla "DWForeignKeys".

Una vez completado el formulario, el mismo nos permite seguir agregando *foreign keys* activando "Define More Foreign Keys".

Esta actividad se ejecutará para cada tabla de hechos del DW Schema.

## MAPPING DATA WAREHOUSE SCHEMA AND INTEGRATED SCHEMA

Para esto se grabará la tabla "MappingIntegratedDW" por cada mapeo.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

## DEFINE JOINS DW SCHEMA

En la tarea de “Mapping Data Warehouse Schema and Integrated Schema” se mapean distintas tablas del Integrated Schema para conformar las tablas del esquema de DW. Esto implica que debemos indicarle al sistema como vamos a *joinear* esas tablas del Integrated para cargar las tablas del DW Schema. En este formulario debemos para cada tabla del Integrated Schema de cada tabla del DWSchema, indicar cómo y con qué campo/s joinean entre ellas. Entonces seleccionamos el campo de la tabla de Integrated que estamos iterando, en el combo “Integrated Attribute”, y luego seleccionamos un campo de las restantes tablas del Integrated Schema en el combo “Integrated Schema and Attribute” que ya no hayan sido procesados sus joins.

Luego de ejecutar esta actividad se grabará una tabla en la base de metadatos que se llama “Joins\_dw” donde cada registro representará un join de los mencionados.

## Descripción de Proceso de Carga

Describimos completamente el proceso de Carga mediante la Figura 45. En ella podemos ver las actividades automáticas y manuales que le pertenecen.

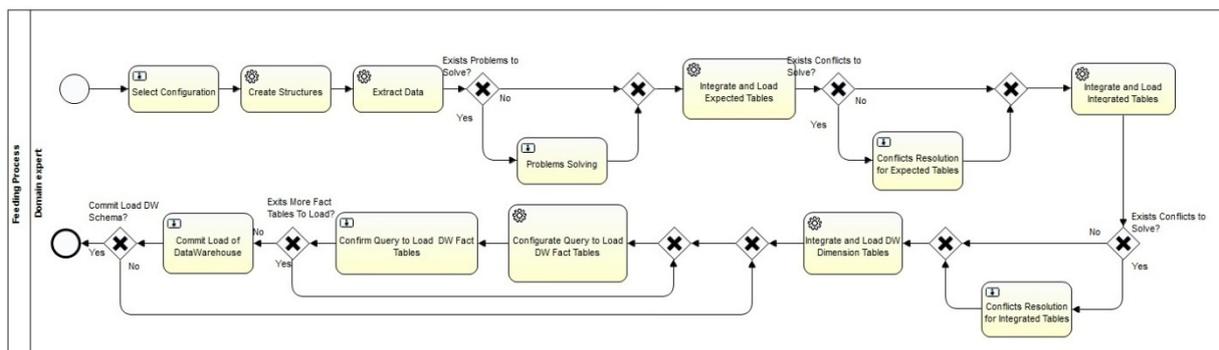


Figura 45 Proceso de Carga.

## SELECT CONFIGURATION

Todas las ejecuciones del proceso de Configuración, se guardan en la base de metadatos con un identificador. Como ya dijimos en la tabla “Configuration”.

Apenas comienza el proceso de Carga, debemos seleccionar cuál de ellas vamos a utilizar.

Lo único que hace la tarea en el sistema, es grabar la variable de flujo correspondiente a esta selección.

## CREATE STRUCTURES

Una vez ejecutada la tarea automática, se crean las estructuras necesarias definidas como metadatos en el proceso de Configuración.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

Más en detalle:

- Se crea la base de datos que alojará todos los datos del proceso de Feeding. Es importante destacar que el proceso de Configuration trabaja en una base de metadatos exclusiva, y cada proceso de Carga tendrá también una base de datos independiente. El nombre de la base de datos será "base\_id" donde id corresponde al identificador del proceso de Configuración.
- Para cada Web Source seleccionado o creado en el proceso de Configuración, se creará una tabla de nombre "expectedEsquema\_webSource", donde webSource es el nombre de la Web Source y expectedEsquema es el nombre del ExpectedSchema asociado a la misma en el proceso de Configuración. La extracción de datos se alojará en cada una de estas tablas, para cada Web Source a procesar. Los atributos de estas tablas serán las columnas especificadas en el mapeo del Web Source.
- Para cada ExpectedSchema asociados a la Configuración seleccionada, se crearán una tabla por cada uno de ellos, donde el nombre de la tabla será el nombre del ExpectedSchema. Los campos de estas tablas, serán los atributos definidos en el ExpectedSchema.
- Se crearán las tablas del IntegratedSchema, con sus atributos definidos.
- De manera similar se crearán las tablas del DW Schema, con sus atributos.
- Por último, se marcará la Configuración como ya procesada.
- Observación: si al inicio de esta tarea automática la Configuración ya figura como procesada, no se creará ni destruirá ninguna estructura, sólo se borrarán registros de las mismas. Eso sucede cuando la ejecución de la carga no es la primera.

## **EXTRACT DATA**

Esta actividad también automática, se encargará de consumir el servicio de extracción de datos, para cada Web Source. Los datos obtenidos serán alojados en las tablas creadas a tales fines, mencionadas anteriormente cuyo nombre es la concatenación de los nombres del ExpectedSchema y del Web Source. Una vez terminada la tarea, todos los datos web habrán sido extraídos y alojados en sus correspondientes tablas.

## **PROBLEMS SOLVING**

Si bien esta tarea no está implementada, queda prevista en el modelo, la idea de esta tarea es resolver los problemas que surjan de la extracción tomando en cuenta los metadatos de calidad obtenidos en el proceso de Configuración.

## **INTEGRATE AND LOAD EXPECTED TABLES**

La actividad de carga de las tablas de Expected, se encargará de unir los datos alojados en las tablas de Web Source, que corresponden al mismo Expected Schema y los cargará en las mencionadas tablas. El fin es de agrupar los datos que corresponden al mismo Expected Schema en su correspondiente tabla.

## **CONFLICTS RESOLUTION FOR EXPECTED TABLES**

Esta tarea no está implementada, queda prevista en el modelo, la idea de esta tarea es resolver los conflictos que surjan en la carga de los Expected Schemas tomando en cuenta los metadatos de calidad obtenidos en el proceso de Configuración.

# ANEXOS INFORME - Construcción de Web Warehouse enfocados en la Calidad con BPMS

---

## **INTEGRATE AND LOAD INTEGRATED TABLES**

En función de los mapeos de Expected Schema e Integrated Schema, y con la definición de joins correspondientes, cargaremos las tablas del Esquema Integrado, joineando las tablas correspondientes del Expected Schema.

## **CONFLICTS RESOLUTION FOR INTEGRATED TABLES**

Esta tarea no está implementada, queda prevista en el modelo, la idea de esta tarea es resolver los conflictos que surjan en la carga de los Integrated Schemas tomando en cuenta los metadatos de calidad obtenidos en el proceso de Configuración.

## **INTEGRATE AND LOAD DW DIMENSION TABLES**

En función de los mapeos de Integrated Schema y DWSchema, y con la definición de joins correspondientes, cargaremos las tablas de dimensiones del Esquema DW, joineando las tablas correspondientes del Integrated Schema.

## **CONFIGURATE QUERY TO LOAD DW FACT TABLES**

Al igual que en la tarea anterior y en función de los mapeos de Integrated Schema y DWSchema, y con la definición de joins correspondientes, cargaremos las tablas de hechos del Esquema DW, joineando las tablas correspondientes del Integrated Schema.

## **CONFIRM QUERY TO LOAD DW FACT TABLES**

Esta tarea muestra al usuario la *select* con la que va a cargar cada tabla de hechos, teniendo la posibilidad de modificarla con la *select* que desee cargarla.

## **COMMIT LOAD OF DATA WAREHOUSE**

Finalmente, esta tarea permite al usuario decidir si la carga de cada tabla de hechos es correcta y desea que sea definitiva, en caso afirmativo no se realiza nada, en caso contrario se borra los datos de la tabla de hechos y se vuelve a la actividad anterior ("*Confirm query to load DW Fact Tables*").