

2. Antecedentes

2.1. Tecnología de flujos de trabajo

2.1.1. Definición de flujos de trabajo

Un flujo de trabajo es una herramienta que permite la implementación técnica de un proceso de negocio a través de la definición de los aspectos funcionales y operacionales propios de la actividad. Para llevarla a cabo será necesaria la asociación coordinada entre documentos, información y tareas; todas ellas relacionadas con los distintos participantes y sujetas a una serie de reglas definidas por el entorno sobre el que se modela. La secuencia correlativa de los elementos, la sincronización entre tareas o la información relacional entre procesos son algunos de los aspectos operacionales más representativos a tener en cuenta en la definición de estos flujos de trabajo.

Para lograr la correcta definición de un flujo de trabajo dentro de un área específica de negocio se requiere no sólo la presencia de un experto en modelado, sino también un conocimiento exhaustivo de la actividad a modelar. Mediante el adecuado uso de los flujos de trabajo se podrán definir procesos fidedignos, capaces de trasladar el comportamiento real del sistema a proceso modelado y conseguir, de este modo, el posible control automatizado del mismo.

Actualmente, las nuevas tendencias en la regulación de organizaciones y empresas muestran que los flujos de trabajo son comúnmente utilizados como herramientas de gestión. De esta forma se consigue mejorar sustancialmente la agilidad y el rendimiento de tareas tanto administrativas como comerciales. Por otro lado, la constante evolución en las tecnologías asociadas a estos flujos de trabajo está permitiendo incrementar notablemente la efectividad de la automatización y del control de estos procesos.

La tecnología de los flujos de trabajo está cimentada en la creencia de que algunas tareas se realizan con mayor efectividad de forma automática. Por ello se delega el control de este tipo de tareas a computadoras. De este modo se obtiene mayor fiabilidad y rapidez en el procesamiento de la información, facilitando además el manejo de los elementos asociados al flujo y proporcionando así mejoras sustanciales en la ejecución.

2.1.2. Representación e Interpretación de Flujos de Trabajo

Al margen de la actividad que se pretenda modelar, la definición y representación de procesos de negocio mediante flujos de trabajo está sujeto a una serie de reglas. En la siguiente enumeración se establece una breve descripción sobre los elementos más comunes presentes en este tipo de modelado:

- ✓ **Tareas:** las tareas se relacionan con el conjunto de actividades o acciones desempeñadas en un proceso. Generalmente son realizadas por una única persona y están englobadas dentro de una competencia específica o rol.
- ✓ **Usuarios:** se corresponden con el conjunto de participantes incluidos en el proceso.
- ✓ **Roles:** define cada una de las distintas competencias existentes en el sistema con independencia de los usuarios a los que vayan a ser asignados. De este modo, un participante podrá tener más de un rol en el flujo.
- ✓ **Rutas:** representa la secuencia de pasos establecida en el sistema para garantizar la consecución de los objetivos del mismo. Para ello, todos los usuarios del proceso deben poder formar parte de esta secuencia. Se definen además varios tipos de ruta, apareciendo las rutas fijas y las rutas condicionales. En las rutas fijas, el camino establecido es unívoco de forma que siempre permanece establecida la misma secuencia de tareas. Por otro lado, las rutas condicionales dependerán de la evaluación de diversas condiciones o variables del sistema para decidir el siguiente salto en su secuencia de tareas.
- ✓ **Reglas de transición:** son un conjunto de reglas lógicas que determinan la siguiente acción a realizar. La definición de estas reglas puede llegar a ser compleja, contemplando múltiples opciones, variaciones o excepciones.
- ✓ **Datos:** se corresponde con toda la información incluida dentro del flujo ya sean documentos, imágenes, archivos, etc. Se Distinguen dos categorías de datos:
 - **Datos de control:** conjunto de datos útiles para la lógica del sistema.
 - **Datos relevantes:** datos utilizados para las tareas de cálculo de rutas correspondientes al siguiente salto en el flujo.
- ✓ **Eventos:** establece una interrupción al comportamiento secuencial del flujo. Suele contener información asociada al evento, además posee un origen y puede tener

uno o más destinatarios. Los eventos pueden ser lanzados voluntariamente por los usuarios o de forma automática por el sistema.

- ✓ **Plazos:** conjunto de tiempos asociados a ciertos elementos del sistema. Se permite la asociación de eventos a la finalización de estos plazos o deadlines.
- ✓ **Políticas y Procesos:** definen la nomenclatura y la forma de actuar de los distintos flujos sobre los que se pretenda realizar el modelado. Además se establecen las interacciones y las sentencias para e el manejo de los distintas tareas contenidas en el flujo de trabajo.

Desgraciadamente, el principal problema que aparece a la hora de modelar procesos de negocio reside en el uso de las metodologías apropiadas para el análisis y la interpretación de los requisitos reales del sistema. Bajo esta premisa aparecen diferentes niveles de comprensión del proceso a partir de los cuales se deberán formular las condiciones aparentes que modelen el comportamiento final del sistema modelado. Por este motivo y como puede observarse en la siguiente figura, el desarrollo obtenido para el modelo software se originará como resultado de la yuxtaposición de una serie interpretaciones llevadas a cabo por distintos actores concedores del proceso.

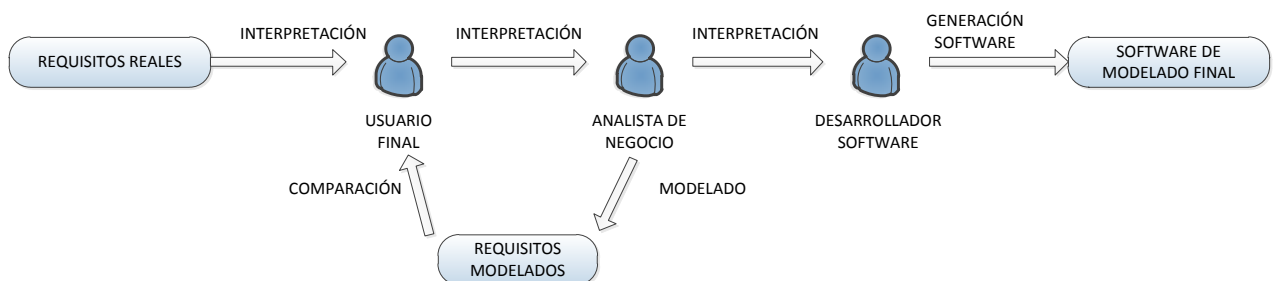


Figura 1: Esquema conceptual de desarrollo para el modelado de un proceso de negocio

En el diagrama anterior se observa la aparición del analista de negocio como figura intermedia entre el usuario final y el desarrollador software. Este actor interpreta un papel fundamental en el modelado encargándose de la traducción de las condiciones y los requisitos del sistema entre ambos extremos del diagrama. De este modo, el analista de negocio deberá caracterizar el modelo de negocio para que la diferencia entre los requisitos reales y los requisitos modelados sea mínima. Para lograr este grado de compromiso, el analista deberá no sólo conocer en profundidad el proceso a modelar, sino que también deberá realizar un estudio exhaustivo de las distintas actividades a realizar por los actores o participantes del proceso. Por último, el analista deberá además ser capaz de transmitir este conocimiento al

desarrollador software y trasladarlo así al lenguaje de modelado de flujos de trabajo requerido por el proyecto.

Como cabe esperar esta tarea será de vital importancia para la futura implementación del proceso. Por este motivo, cualquier inconsistencia a la hora de trasladar las características del sistema así como su posterior modelado repercutirá significativamente en la solución final obtenida.

Aunque el concepto de automatización asociado a los flujos de trabajo se encuentra estrechamente ligado con la optimización de procesos de negocio, la aplicabilidad real de esta tecnología resulta bastante limitada. Por este motivo, el desarrollo del proyecto se orientará hacia otra tecnología asociada, conocida como Business Process Management y presentada en la siguiente sección de la memoria.

2.1.3. Business Process Management (BPM)

Se denomina BPM al conjunto de herramientas y servicios que facilitan la administración de un proceso de negocio. Como se comentó en el cuadro anterior, el principal objetivo perseguido por este conjunto se centra en la mejora del desempeño organizativo de los procesos, buscando optimizar la eficiencia y eficacia de los mismos. De este modo, las competencias fundamentales de BPM abarcarán el diseño, el modelado, la organización, la documentación y la optimización de toda la actividad, de una forma estable y controlada. A través de estas capacidades aparece un ciclo continuo y cerrado capaz de desarrollar, en cada iteración, la propia implementación del proceso. Por todo ello, el sistema desarrollado por BPM ambicionará la continua mejora del proceso, permitiendo a los desarrolladores tener un acercamiento mayor a la actividad real sobre la que se desarrolla. Este ciclo aparece gráficamente representado en la figura que se muestra a continuación.

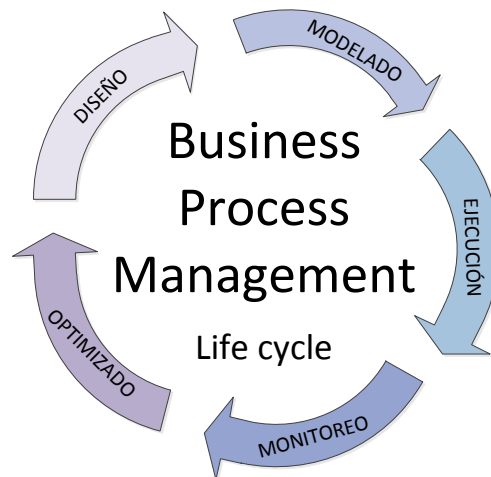


Figura 2: Ciclo de vida para un gestor de procesos de negocio

El **DISEÑO** es una etapa que incluye la identificación de los procesos existentes y el diseño de futuras líneas de desarrollo para nuevos procesos. Dentro de esta etapa se incorporará la representación de los flujos además de otros factores como alertas y notificaciones, procedimientos de operación estándar, acuerdos de nivel de servicio y mecanismos de entrega de tareas, entre otros. Cabe destacar que un diseño correcto y eficiente reducirá sustancialmente el número de problemas durante el tiempo de vida del proceso.

Durante la etapa de **MODELADO** se realizará la correspondiente traducción del diseño realizado en la etapa anterior al lenguaje requerido para su ejecución. Para completar esta etapa se incluirán una serie de variables adicionales. De este modo se buscará extender la operación del proceso con total independencia de las circunstancias en las que se encuentre.

Para la **EJECUCION** se hará uso de aplicaciones que realicen el seguimiento secuencial del proceso definido en las etapas anteriores. Por medio de esta ejecución se obtendrá una aproximación controlada sobre la automatización general del proceso en cuestión. La variedad de las aplicaciones utilizadas con este fin adoptará diferentes niveles de complejidad en relación con los requisitos de ejecución estipulados durante esta etapa.

Durante la **MONITORIZACION** se capturarán los distintos procesos que se encuentren en ejecución. A través de ellos se almacenará la información concerniente a su estado real, así como datos relativos a su operación específica. El grado de monitorización dependerá directamente tanto de la información del negocio que se quiera evaluar y analizar, como de las características propias del proceso que se esté considerando

La última etapa se corresponde con la **OPTIMIZACIÓN**. En esta etapa se recuperará la información sobre el rendimiento del proceso proveniente de la etapa anterior. A partir de ella se analizará en busca de posibles deficiencias, cuellos de botella u otros aspectos que puedan limitar el funcionamiento optimizado del proceso. Por otro lado, ciertos aspectos como la reducción de costes, o mejoras en algunas de las etapas del proceso de negocio pueden también hacerse patentes, permitiendo así mejorar el rendimiento general del mismo.

2.1.4. Business Process Management Notation (BPMN)

BPMN es el acrónimo de Business Process Management Notation. Este acrónimo se corresponde con un estándar para la notación gráfica de los procesos de negocio utilizados en la plataforma JBPM. Esta notación permite unificar la representación de los procesos de negocio facilitando, de forma intuitiva, la organización gráfica de sus elementos. Los diagramas y gráficos usados en BPMN se integran perfectamente a las tecnologías de la información aplicadas comúnmente en las empresas dentro del formato establecido para los flujos de trabajo. El principal objetivo perseguido por esta notación consiste en estandarizar la representación de los procesos y facilitar así su utilización entre los distintos participantes involucrados. De este modo se busca crear un lenguaje de comunicación fiable, capaz de servir de enlace entre el diseño y la implementación final de estos procesos.

Aunque el estándar perseguido por BPMN se encuentra bastante extendido, en la actualidad existe una amplia variedad de lenguajes y herramientas destinadas a esta representación. Al margen de ello, para el objetivo perseguido por este proyecto este lenguaje de representación no sólo aporta interesantes características de rendimiento y aplicabilidad sino que también añade otros factores a tener en cuenta. Entre estos factores, resulta interesante destacar los siguientes:

- BPMN representa un estándar internacional de modelado de procesos consolidado y aceptado por la comunidad.
- BPMN resulta completamente independiente de la metodología utilizada en el modelado de procesos.
- BPMN permite crear un puente que limite la brecha existente entre el proceso de negocio y su implementación.

- BPMN consigue modelar los procesos de forma unificada y estandarizada, consiguiendo un cómodo entendimiento entre los distintos participantes implicados.

2.1.5. Herramientas BPMN

En el mercado existen múltiples herramientas dedicadas al modelado de procesos en BPMN. Debido al éxito en la implementación de procesos con esta notación han ido surgiendo a lo largo de los años numerosas herramientas que dan soporte a esta nomenclatura. La mayoría de ellas ofrecen un entorno de diseño bastante depurado y de fácil uso, presentando además interfaces de usuario estéticas y cómodas.

Para la realización de este proyecto se optó inicialmente por la utilización del software Bizagi Process Modeler. Esta herramienta de modelado desarrollada por BizAgi permite un diseño rápido y completo de procesos de negocio basándose en el estándar de modelado BPMN. Adicionalmente, este software aportaba una interfaz de usuario clara y atractiva. A pesar de ello, la elección de este programa en concreto fue debida principalmente a otros motivos: a la **facilidad** de uso, ya que resulta bastante intuitivo, y a la **rapidez** en su aprendizaje, ya que existían mucha documentación asociada.

En la siguiente captura de pantalla se recoge la interfaz de trabajo de este software de modelado. Como puede observarse ofrece una interfaz de modelado cómoda y sencilla. Los distintos elementos utilizables permanecen recogidos en la columna izquierda del editor mientras que el entorno de edición representa la mayor parte de la ventana. Esta captura se corresponde con el diagrama de modelado completo para el proceso de elaboración de vino. Este diagrama, así como los subprocesos incluidos en él, se presentarán con mayor profundidad en el anexo correspondiente al modelado del proyecto mediante esta herramienta.

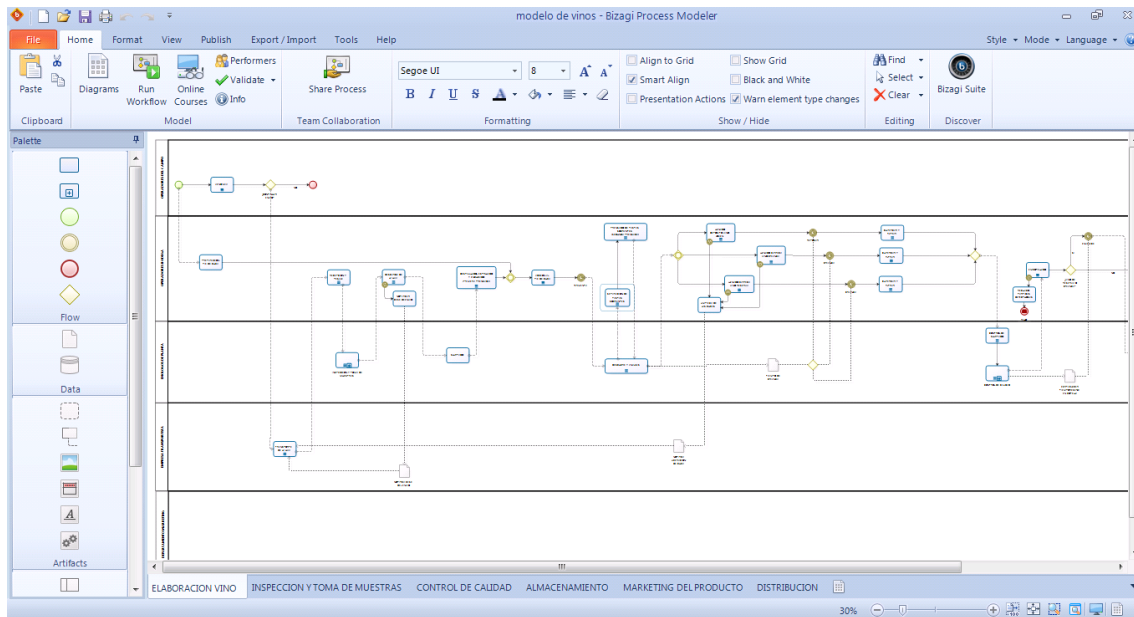


Figura 3: Modelado del proceso general de elaboración de vino mediante el software de Bizagi Modeler

Por otro lado, el software presentado anteriormente ofrece además una compatibilidad directa con otro programa de la misma compañía llamado Bizagi BPM Suite. Este programa, al igual que otras “suites” de BPM, permite ejecutar, monitorizar y gestionar los distintos procesos definidos en el programa de modelado. Una vez integrados ambos programas, el conjunto obtenido permitiría generar una solución plausible y adecuada a los objetivos principales que se abordan en este proyecto. Sin embargo, durante la integración del mismo (entre el proceso definido en el Bizagi Modeler y el software Bizagi BPM Suite) aparecieron numerosos problemas de compatibilidad entre sus versiones. Ese inconveniente unido a la gran cantidad de fallos recogidos por el BPM Suite en su versión gratuita provocó un cambio en la orientación del proyecto hacia otros entornos más estables y libres. Gracias a este cambio se decidió finalmente por la utilización del entorno de desarrollo Eclipse a través de un “plugin” específico para la definición de procesos mediante el estándar BPMN. En secciones posteriores de la memoria se verá de forma detallada tanto el proceso de instalación como la configuración de este entorno a partir del cual se ha desarrollado principalmente este proyecto.

Antes de continuar con el siguiente apartado de la memoria conviene hacer una breve introducción del entorno de desarrollo Eclipse y, más concretamente, del componente asociado a la definición de procesos en BPMN. Los correspondientes pasos necesarios para

llevar a cabo la instalación y la configuración del entorno de trabajo se encontrarán recogidos en el anexo titulado “Manual de configuración”.

Una vez desplegado correctamente el entorno de desarrollo se podrá iniciar la definición gráfica de procesos siguiendo, para ello, los pasos que se describen a continuación. El primero de estos pasos consistirá en la creación de un nuevo proyecto incluido dentro del desplegable correspondiente a los proyectos jBoss jBPM. Al seleccionar esta casilla aparecerá otra ventana dedicada, en este caso, a nombrar el nuevo proyecto. Ambas ventanas están recogidas en las capturas de pantalla presentadas a continuación.

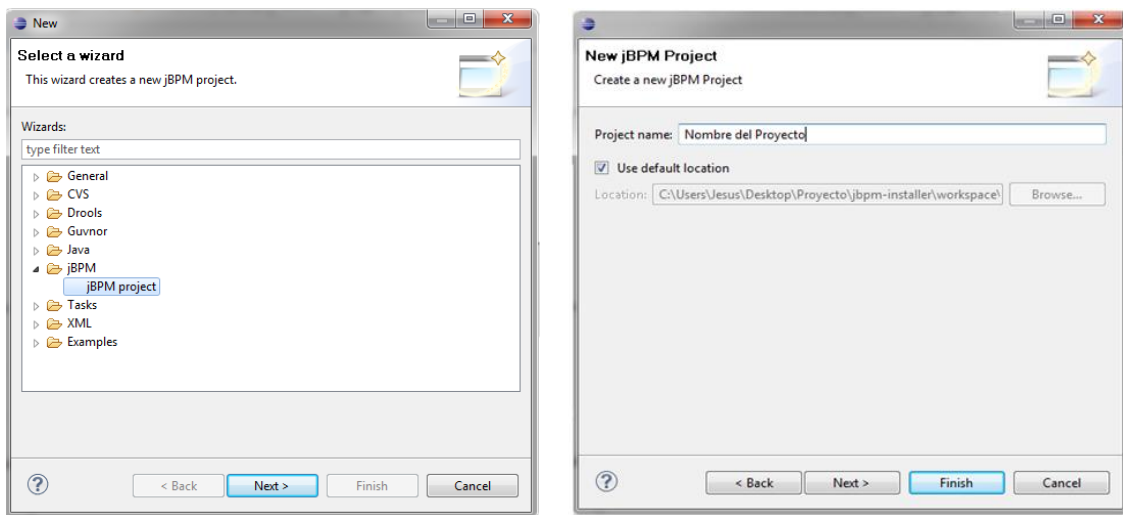


Figura 4: Creación de un nuevo proyecto jBoss jBPM mediante el componente instalado en Eclipse Indigo.

Tras crear correctamente el nuevo proyecto se abrirá la interfaz del entorno de desarrollo para la edición gráfica de diagramas en BPMN. Este editor, que se puede observar en la siguiente captura de pantalla, distingue tres ventanas principales además de una completa barra de herramientas localizada en la parte superior del conjunto.

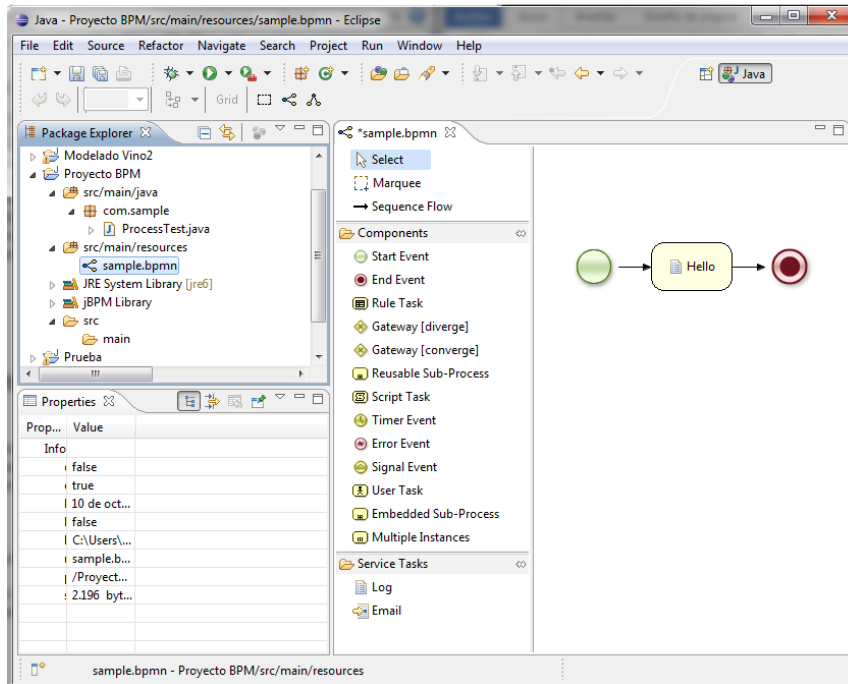


Figura 5: Interfaz de desarrollo gráfico para el modelado en BPMN.

La primera de estas ventanas, denominada “Package Explorer”, hace referencia al explorador del entorno de trabajo de Eclipse. Para los proyectos creados como jBoss jBPM, la estructura de directorios es tal y como se representa en la captura de pantalla. Los elementos principales de este árbol de directorios serán el correspondiente a “src/main/java” donde se almacenan las clases java definidas en el proyecto; el asociado a “src/main/resource” que incorpora todos los flujos BPMN fijados en el proyecto y, por último, todas las librerías incluidas, ya sean las definidas por el Java Runtime Environment (JRE); las recogidas por la jBPM o cualquier otra librería que se quiera referenciar.

La segunda ventana a tener en cuenta en el entorno de desarrollo se denomina “Properties”. Esta ventana, como su nombre indica, reflejará las propiedades intrínsecas de los elementos seleccionados en el entorno. A través de ella, será posible modificar y definir las características internas de los distintos nodos utilizados en el editor gráfico BPMN.

La tercera ventana a destacar hace referencia a la nombrada con el propio nombre del archivo a editar, en este caso “simple.bpmn”. Dentro de esta interfaz se hace posible la edición de los diagramas BPMN por medio de la elección de los nodos suministrados en la barra localizada a su izquierda. Entre todos los elementos incorporados por esta barra de edición, se pueden distinguir la gran mayoría de los facilitados en el anexo correspondiente al estándar de notación gráfica BPMN.

2.2. Servidor de aplicaciones JBoss

Las siglas JBoss se refieren al servidor de aplicaciones de código abierto implementado en su totalidad en Java. Este servidor, al ser de código abierto, fundamenta su desarrollo en una red mundial de colaboradores y testadores organizados bajo la gestión interna de los foros de www.jboss.org. Mientras que, por un lado, esta red de colaboradores representa un pilar fiable sobre el que plantear las posibles dudas de los desarrolladores, por el otro, la licencia abierta permite a los usuarios descargar, utilizar y distribuir su contenido sin ninguna restricción asociada.

Entre las características más atrayentes a tener en cuenta en la elección de este servidor, resulta interesante destacar las siguientes:

- Rápido, ligero y escalable
- Ofrece una arquitectura modular
- Resulta flexible y consistente
- Permite la administración de dominios
- Cumple con los estándares
- Aporta servicios middleware para cualquier objeto Java
- Incorpora plataformas de alto rendimiento para aplicaciones empresariales
- Permite despliegue en paralelo
- Administración elegante por medio de interfaz JMX
- Resulta bastante fiable en entornos empresariales

A raíz de ello, este servidor ha podido fácilmente desarrollarse y extenderse por todo el panorama internacional convirtiéndose, actualmente, en uno de los servidores más populares del mercado.

2.2.1. Arquitectura del servidor

En cuanto a su funcionamiento interno, JBoss implementa todo el paquete de servicios definidos en la plataforma J2EE (Enterprise Edition), la cual se dedica al desarrollo y la ejecución de software de aplicaciones. A través de este paquete de servicios se incorporan los estándares necesarios para usar JBoss como contenedor de componentes, ya sea Java Servlet Pages (JSP), Java Message Service (JMS) o Enterprise JavaBeans (EJB), entre otros. Por medio

de estos contenedores se pueden definir además diferentes capas en la aplicación (gestión de sesiones, acceso remoto a base de datos, interfaces usuario, etc.) de forma centralizada reduciendo, de este modo, la complejidad en el desarrollo de las aplicaciones y servicios.

Por otro lado, este servidor también incorpora otros servicios más allá de los propiamente definidos por la plataforma J2EE. A partir de ellos se asegura extender las capacidades de todo el conjunto hacia un entorno más completo y avanzado. Entre todos los servicios incluidos en JBoss, algunos de los más representativos quedan representados en el diagrama propuesto a continuación:

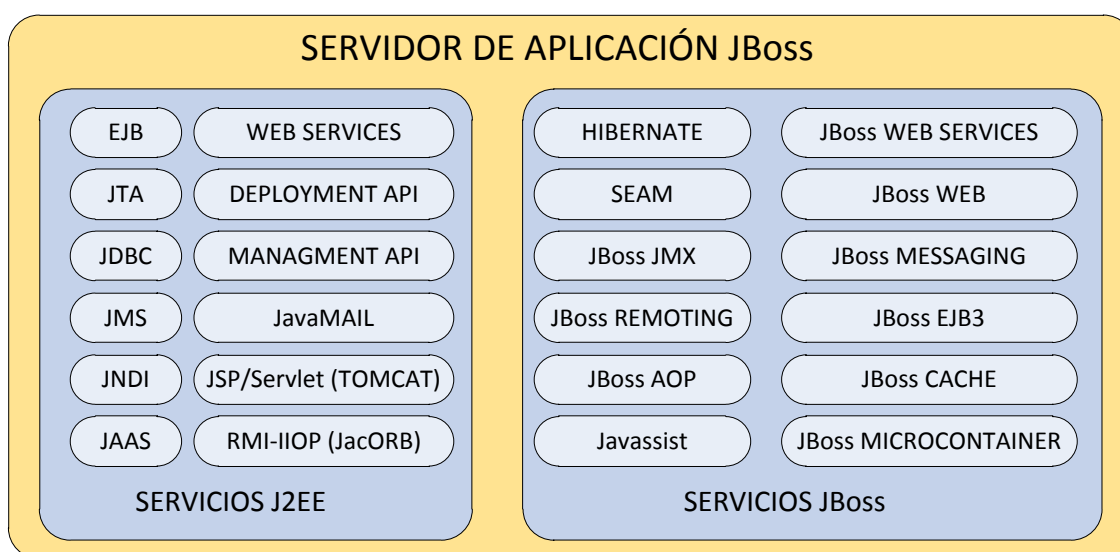


Figura 6: Elementos y servicios incorporados en el servidor de aplicaciones JBoss

Bajo la agrupación de todos estos elementos, el núcleo del servidor JBoss se encuentra implementado sobre Java Managment eXtension (JMX). Este núcleo, construido como una aplicación J2EE, permite el despliegue de servicios Mbeans para la gestión dinámica de dependencias y parámetros configurables en los distintos servicios del servidor.

A partir del modelo JMX se consigue, entre otras funciones, modificar una determinada configuración, gestionar el comportamiento de una aplicación, conocer las estadísticas o los cambios de estado de un servicio concreto, interoperar con otras tecnologías, etc.

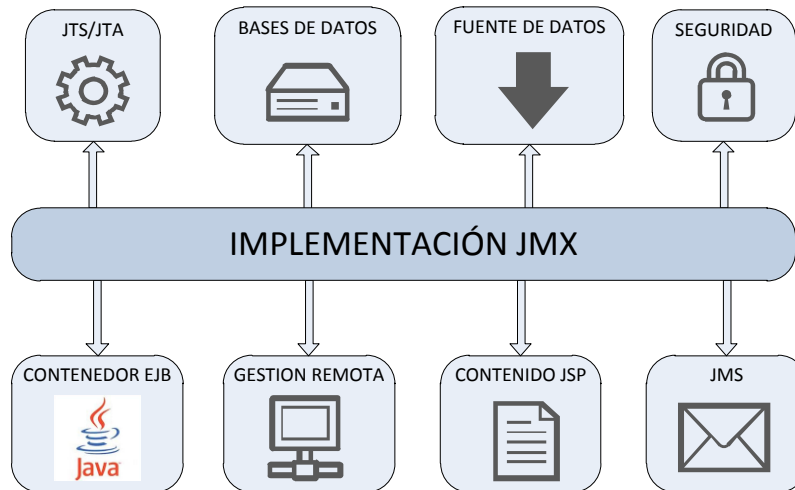


Figura 7: Integración entre JMX y los componentes estándar de JBoss

De entre todas sus acciones, algunas de las más interesantes vienen recogidas a continuación:

- ✓ Mostrar el árbol JNDI (Java Naming and Directory Interface)
- ✓ Forzar un volcado de memoria
- ✓ Mostrar el uso de la memoria
- ✓ Gestionar el escáner de despliegues
- ✓ Redespargar una aplicación
- ✓ Detener JBoss
- ✓ Conocer el estado de los EJB (Enterprise JavaBeans) desplegados e instanciados
- ✓ Acceder a la base de datos Hypersonic (HSQL).

Para la navegación a través de la interfaz, JBoss incorpora una consola accesible desde varias URLs. De este modo y por medio de un navegador, se mostrarán todas las MBeans de acceso público manipulables por los usuarios en el servidor. El contenido mostrado por estas URLs (tanto por <http://localhost:8080/jmx-console/> como por <http://localhost:8080/web-console/>) está, a modo de ejemplo gráfico, recogido en las siguientes capturas de pantallas.

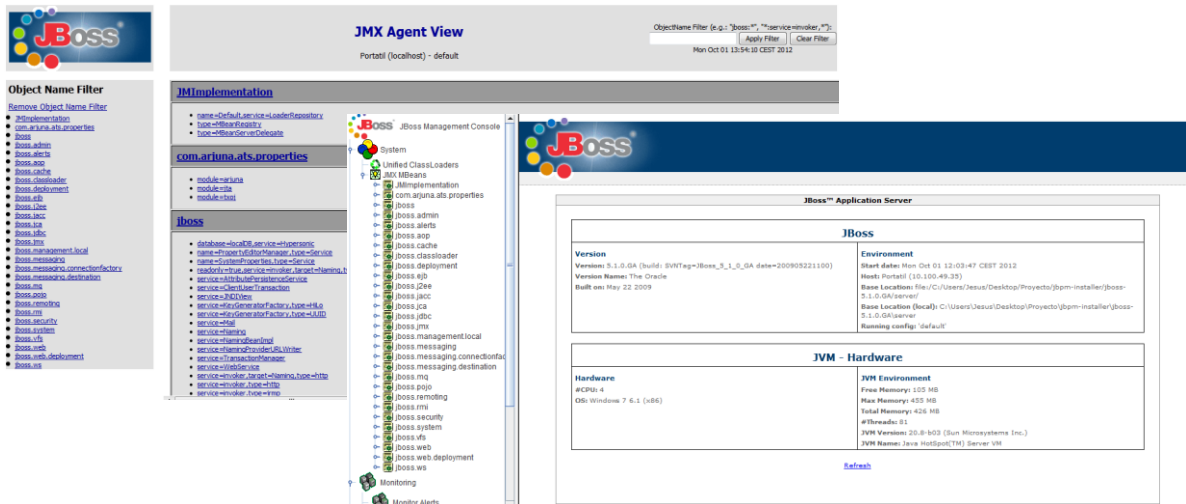


Figura 8: Capturas de pantalla de las consolas de acceso JMX para el servidor JBoss

2.2.2. Base de datos HypersonicSQL y conectores JDBC

Con objeto de aclarar las tecnologías más importantes utilizadas en este proyecto resulta interesante incidir brevemente sobre las características principales de la base de datos empleada. En concreto, esta base de datos se corresponde con HyperSQL (HSQL), un motor de bases de datos ligero, de código abierto, estable y completamente implementado en Java. Entre sus características principales se pueden destacar las siguientes:

- ✓ Soporta la sintaxis SQL estándar.
- ✓ Posee un cómodo sistema gestor de bases de datos.
- ✓ Todos sus procedimientos están almacenados en Java
- ✓ Permite la utilización de restricciones y disparadores con operaciones de INSERT, DELETE o UPDATE
- ✓ Soporta la integridad referencial (claves foráneas)
- ✓ Tiempo de arranque optimizado (mínimo)
- ✓ Gran velocidad en las operaciones sobre la base de datos: SELECT, INSERT, DELETE y UPDATE.
- ✓ Almacenamiento en disco hasta 8GB
- ✓ Al estar escrito en Java ofrece portabilidad
- ✓ Ofrece un controlador JDBC fácilmente utilizable de forma embebida en una aplicación

Tal y como se ha comentado en la enumeración anterior, este motor de base de datos incorpora un gestor cómodo y fiable. Este elemento, que puede invocarse desde la interfaz JMX definida anteriormente, ha sido de vital importancia a la hora de evaluar la interacción de los procesos definidos en el proyecto y su base de datos correspondiente. En la siguiente captura de pantalla pueden observarse dos ventanas superpuestas que corresponden tanto a la pantalla para la configuración de conexión con la base de datos como a la propia interfaz de operación del gestor.

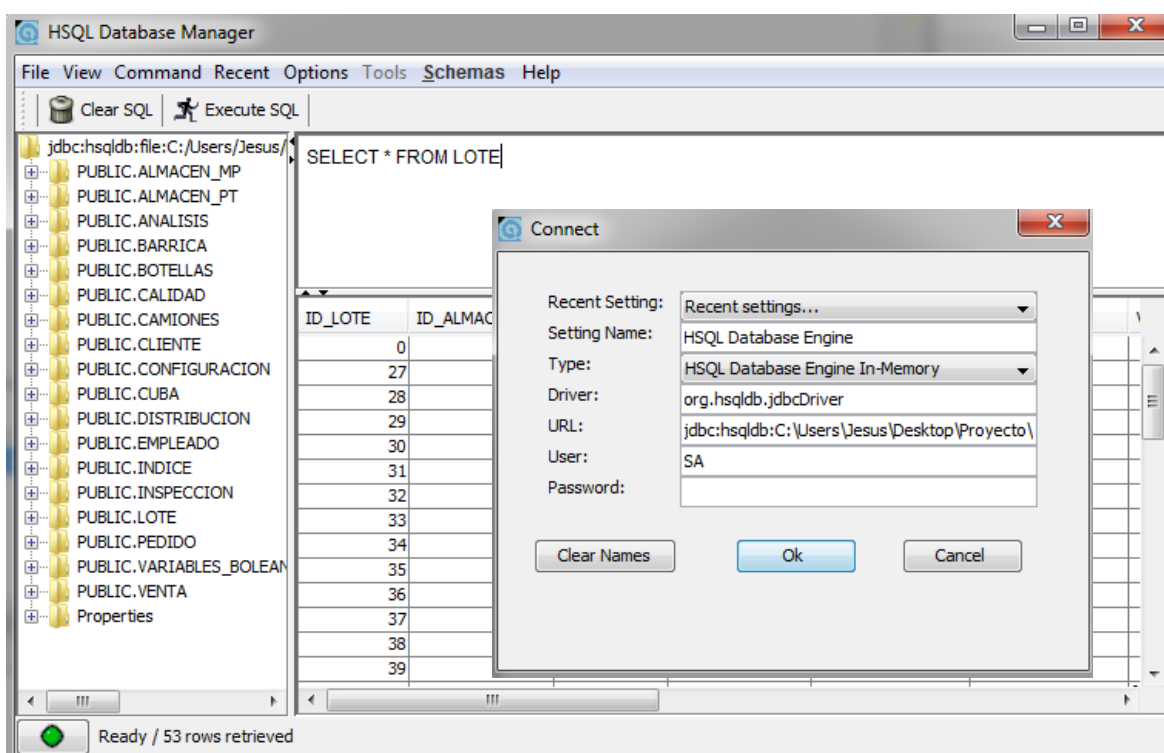


Figura 9: Capturas de pantalla para el sistema gestor de base de datos para HSQL y correspondiente ventana de conexión

En la primera de ellas, titulada “Connect”, se establecen los parámetros de configuración para la conexión entre el sistema gestor y la base de datos. Los principales parámetros los representan el tipo de conexión, el driver utilizado, y la URL específica con la ubicación de la base de datos en memoria. Para mayor seguridad se pueden asociar nombre de usuario y contraseña a la base de datos y evitar así posibles accesos indebidos o no autorizados.

La segunda pantalla, titulada “HSQL Database Manager”, se corresponde con la interfaz de usuario para el sistema gestor de bases de datos. Esta interfaz se encuentra distribuida en tres partes. La primera, localizada en la parte izquierda de la pantalla, hace referencia a un árbol esquemático donde están representadas todas las tablas creadas en la

base de datos. Las otras dos partes se corresponden con la línea de comandos, situada en la parte superior, y con el resultado de la operación, ubicada en la inferior.

Respecto a la conexión con la base de datos, se utilizarán conectores JDBC. JDBC es un API (Application Programming Interface) que define una librería estándar para el acceso a bases de datos. En el siguiente cuadro de código se puede observar un acceso a la base de datos a través de un conector JDBC.

```
global java.lang.String sql
global java.sql.Connection conn
global java.sql.Statement st
ResultSet rst1 = null;
try { // Cargamos el controlador JDBC
    Class.forName("org.hsqldb.jdbcDriver").newInstance();
}
catch (Exception ex)
{
    System.err.println("Se ha producido un error al cargar el controlador JDBC");
}
Try
{ // Se realiza la conexión JDBC
    conn = DriverManager.getConnection(url_db,"sa","");
    st = conn.createStatement();
}
catch (Exception ex)
{
    System.out.println(ex);
}
try
{ // Se ejecuta una sentencia SQL
    st.executeUpdate("insert into almacen_mp (id_almacen,localizacion,esp_libre,estado,
    c_max) values (" +id+ "," +string2+ ", "+num1_f+ ", true, "+num1_f+"");
}
catch (Exception e)
{
    System.out.println(e);
}
try
{ // Se cierra la conexión JDBC
    st.executeUpdate("SHUTDOWN");
}
catch (Exception e)
{
    e.printStackTrace();
}
```

Figura 10: Código correspondiente a un acceso a la base de datos HSQLDB mediante conectores JDBC

Aunque el código anterior resulta a la par simple y autoexplicativo merece la pena destacar que tanto las posibles dudas que pudieran surgir como otros aspectos de interés

(librerías, clases, variables, etc.) quedarán recogidos con mayor profundidad en capítulos posteriores de la memoria.

La elección de esta base de datos concreta no está sujeta a ningún criterio específico, sino que se fundamenta en el hecho de estar incluida dentro del paquete descargable para el servidor JBoss. A partir de diversas pruebas de uso, así como de su correspondiente testeo durante la ejecución del propio proceso general, se determinó que esta herramienta cumplía convenientemente con los principales requisitos perseguidos por este proyecto.

2.3. jBPM y JPD

2.3.1. JBoss jBPM

Se conoce como JBoss jBPM a una plataforma de desarrollo de código abierto e implementada en Java que modela, gestiona y ejecuta procesos de negocio mediante el estándar BPMN 2.0. Este sistema, basado en una programación orientada a gráficos (GOA), representa un elemento clave para la implantación de los flujos de trabajo en entornos de negocio empresarial. Mediante la utilización de los servicios ofrecidos por esta plataforma se conseguirá implementar los distintos procesos de negocio coordinando, para ello, personas, servicios y aplicaciones.

A través de JBoss jBPM se permitirá cumplir con las necesidades actuales de las organizaciones empresariales, asegurando además su adaptabilidad frente a los requisitos tanto de los clientes como de los mercados. De forma adicional, la escalabilidad garantizada por su motor de procesos admitirá su utilización en toda clase de entornos, con total independencia de la complejidad inherente al mismo. El resultado final será una plataforma flexible, capaz de crear, coordinar y monitorizar procesos de negocio de forma ágil, flexible y fiable.

Los elementos principales incorporados por esta herramienta vendrán determinados por el editor de procesos y el motor de ejecución de procesos. En nuestro caso, para el editor se utilizará un diseñador instalado como componente adicional al software de desarrollo Eclipse IDE Indigo.

El motor de procesos utilizado se corresponde con el componente por defecto definido en la JBoss Enterprise SOA Platform para jBPM. Este motor se encarga de la interpretación y la

ejecución de los procesos definidos en la etapa de modelado. Para lograr su cometido, el motor de procesos también podrá hacer uso de diversas bases de datos asociadas así como de sus numerosos registros locales. A partir de ellos se podrán efectuar las peticiones y consultas pertinentes requeridas durante la ejecución.

Si bien se pretende obtener una breve visión de conjunto para el funcionamiento interno de jBPM, resulta de interés aclarar algunas de las funciones adicionales que este sistema nos ofrece. Entre estas funciones se destacan:

- ✓ Verificación del estado actual del sistema por medio de la validación de los comandos utilizados en la ejecución de una tarea específica.
- ✓ Autenticación de los usuarios del proceso, estableciendo además la accesibilidad controlada a las distintas tareas definidas en el flujo.
- ✓ Elección del camino correcto por medio de la evaluación de condiciones y reglas en tiempo de ejecución. Para esta evaluación se pueden incorporar mecanismos de toma de decisiones basados en el motor de reglas de jBPM, a través de la definición de archivos específicos o Drools.

Como cabe esperar, el motor de procesos representa un elemento determinante en la automatización de los flujos de trabajo. Este elemento tomará como entrada los distintos procesos definidos durante la etapa de modelado. Dichos elementos, desarrollados gráficamente por el diseñador instalado en Eclipse, establecerán el flujo secuencial de trabajo a partir del cual originará el orden estricto de ejecución.

Por este motivo, esta herramienta ofrece el puente de unión entre los modelos gráficos de negocio, y su correspondiente implementación software; o, desde otro punto de vista, entre los analistas de negocio y los desarrolladores de BPM. Este concepto, unido a otros comentados en capítulos anteriores, ponen de manifiesto la estrecha relación entre todos los actores implicados en la automatización de los procesos de negocio, ya sean los propios clientes, los administradores y empleados de la empresa, los analistas del negocio o los desarrolladores de software.

Una vez ejecutado el proceso de negocio, el motor será el encargado de determinar todas las conexiones necesarias para llevar a cabo las distintas tareas requeridas, ya sea con sus participantes humanos o con servicios y aplicaciones externas. A modo de recordatorio,

jBPM distinguirá entre dos tipos de actividades dependiendo de su modo de ejecución. Atendiendo a esta clasificación podremos encontrar actividades automáticas donde la ejecución se realiza de forma directa y automática por el software de aplicación; y actividades con tiempos de espera, donde su ejecución dependerá del resultado obtenido de servicios automáticos externos o de participantes humanos. Esta variedad de actividades, así como un desglose interno de la plataforma jBPM, se representa de forma esquemática en la siguiente figura.

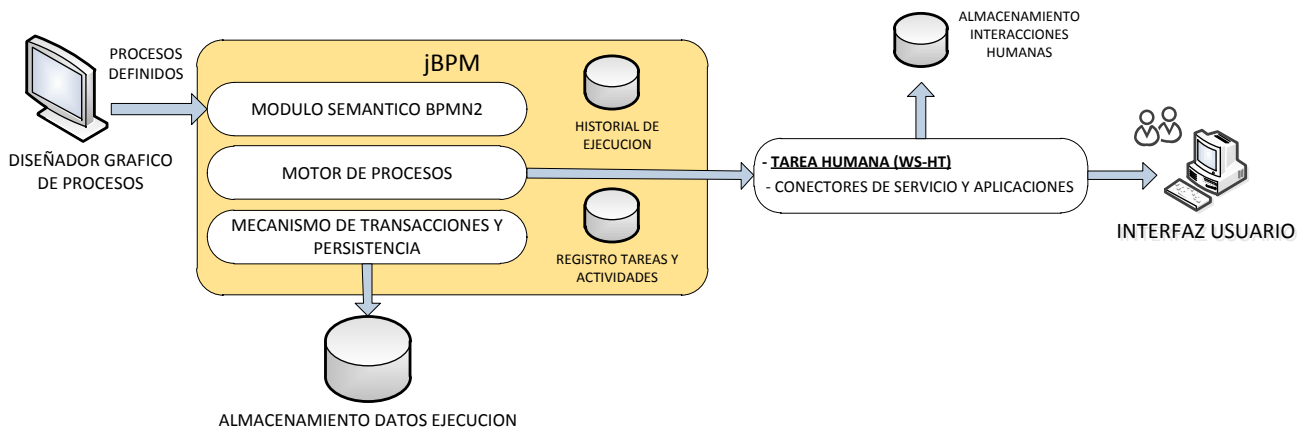


Figura 11: Esquema interno de la plataforma jBPM

Aunque jBPM como herramienta permite la conexión con servicios y aplicaciones externas, este proyecto se centra principalmente en la interacción coordinada con actores humanos. Para esta interacción, jBPM utiliza un estándar denominado Web Service Human Task (WS-HT). Mediante este componente de establecen, mantienen y administran los ciclos de vida asociados a la tareas de interacción humana. A través de este estándar no sólo se definen los posibles estados alcanzables en su ciclo de vida, sino también las estructuras de datos manejables por ellas.

Por otro lado, en el esquema anterior también está recogida una subdivisión interna de varios componentes dentro del conjunto jBPM. Estos elementos se encuentran agrupados según la funcionalidad específica de cada uno de ellos. Dentro de esta agrupación se distinguen tipos esenciales:

- ✓ Módulo semántico BPMN2: este componente será el encargado de interpretar los procesos definidos en el programa de modelado gráfico. Para ello, el

proceso deberá cumplir con las especificaciones establecidas en el estándar de notación BPMN2.

- ✓ Núcleo del motor de procesos de negocio: este elemento se ocupa de crear instancias de los procesos interpretados por el módulo semántico y ejecutar sus definiciones. Además, el motor de jBPM proporciona un conjunto de APIs de Java para su integración con otras aplicaciones.
- ✓ Mecanismos de transacción y persistencia: este módulo es el encargado de almacenar y mantener el estado de cualquier proceso de negocio ejecutado en el sistema.
- ✓ Historial de ejecución: se corresponde con un módulo de memoria dedicado al almacenamiento de toda información adicional relativa a la ejecución de los procesos de negocio.

Dentro del motor de procesos jBPM se establece también la posibilidad de ejecución múltiple de procesos. Esta característica, que permite tener activos varios procesos a la vez de forma concurrente, se plantea como una interesante ventaja a tener en cuenta por los desarrolladores atraídos por esta plataforma. A partir de esta ejecución múltiple se conseguirá establecer instancias independientes de los procesos para una ejecución autónoma y simultánea. Debido a la importancia operacional de esta característica, este aspecto será discutido con mayor profundidad en posteriores apartados de la memoria.

2.3.2. JPDL (jBPM Process Definition Language)

Como se comentó en el capítulo anterior, la definición de procesos se realiza mediante un editor gráfico instalado como componente del software de desarrollo Eclipse. A través de este componente se generarán los distintos archivos correspondientes al proceso en cuestión sirviendo, posteriormente, de referencia en ejecución al motor central de jBPM. Aunque esta aclaración es correcta, no resulta del todo completa. Para completarla es necesario puntualizar que el motor de proceso jBPM no entiende de entornos gráficos, siendo necesaria una correspondencia entre el modelo gráfico y el correspondiente lenguaje de programación empleado.

Bajo esta premisa nace el concepto de JPDL. Estas siglas, que hacen referencia a jBPM Process Definition Language, se asocian a una notación para el lenguaje de definición de

procesos de acuerdo con un XML Schema. Este lenguaje, equivalente por tanto a XML, representa específicamente lo editado de forma gráfica mediante el componente de diseño gráfico de Eclipse. Si bien existe una relación casi total entre lo editado gráficamente en Eclipse y lo definido en JPDL, existen también ciertas entidades que no pueden ser representadas en el componente gráfico aunque sí definidas por JPDL. Al margen de ello, la relación entre ambos métodos de definición, ya sea el método gráfico o el programado, resulta bastante directa y completa. De este modo, se puede apreciar que el editor gráfico no sólo es capaz de leer e interpretar el fichero definido en JPDL, sino que también puede generar de forma automática los cambios editados gráficamente en el correspondiente lenguaje de definición.

A continuación se añade el esquema genérico de definición de procesos estableciendo, para ello, la correspondencia entre el modelado gráfico y la definición en JPDL.

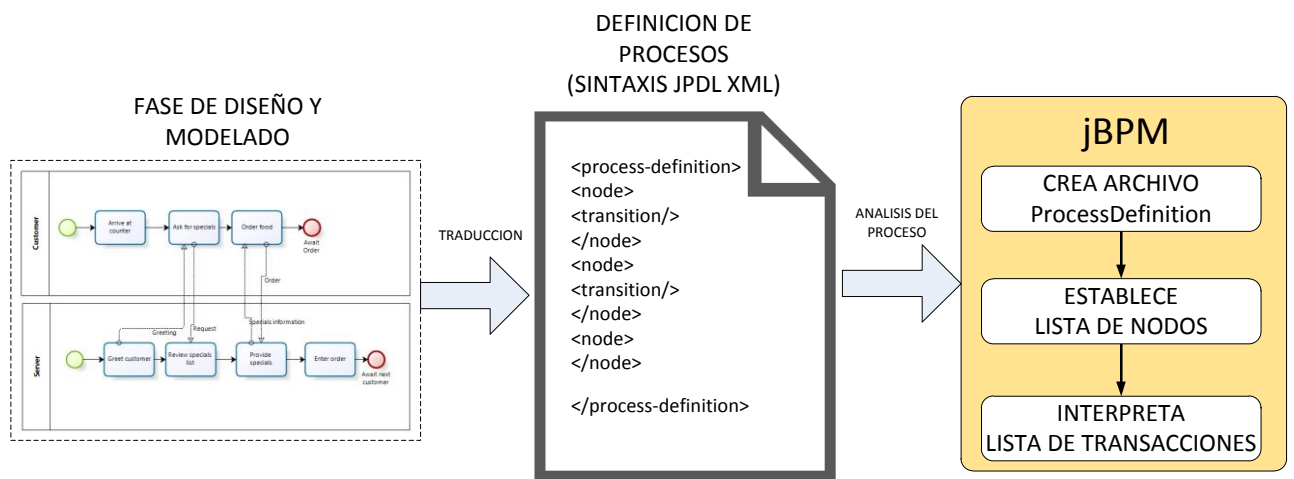


Figura 12: Asociación entre los métodos de definición de procesos y el comportamiento interno de jBPM

Como cabe esperar, en la definición de procesos en JPDL se hará uso de las correspondientes etiquetas XML. Todas estas etiquetas permanecen recogidas en un esquema XML fácilmente accesible en la red. Para localizar cualquier información acerca de este esquema, sólo es necesario navegar a través de páginas como la siguiente:

<http://docs.jboss.com/jbpm/v4/schemadocs/index.html>.