

UNLP
Facultad de Informática
Tesina de Grado

Evaluación de plataformas que proveen
soporte BPM

*Facundo Martin Rodríguez,
Hector Gabriel Gotelli*

Director
Lic. Patricia Bazán
CoDirector
Lic. Claudia Queiruga

Índice

Prólogo	5
Parte I. Fundamentos	6
Capítulo 1 – BPM	7
Introducción.....	7
BPM (<i>Business Process Management</i>).....	7
BPMS y Ciclo de Vida de los Procesos.....	9
Ciclo de Vida de un Proceso en Oracle.....	10
Fases del Ciclo de Vida.....	11
BPMM (Business Process Maturity Model).....	14
¿Qué es BPMM?.....	16
Aplicación de BPMM.....	17
Niveles de Madurez.....	17
BPMM y la Estandarización de la OMG.....	18
Capítulo 2 – BPMN	19
Introducción.....	19
¿Por qué BPMN es importante?.....	19
¿A quién está dirigido BPMN?.....	19
¿Cómo se ajusta BPMN con UML?.....	20
Elementos del Lenguaje.....	21
<i>Flow Objects</i>	21
<i>Connecting Objects</i>	22
<i>Swimlanes</i>	24
<i>Artifacts</i>	26
Usos de BPMN.....	28
Procesos B2B Colaborativos.....	29
Procesos de Negocios Privados o Internos.....	29
Diferentes Niveles de Precisión.....	29
Ventajas de Modelar en BPMN.....	30
Capítulo 3 – Estándares de Soporte BPM	31
XPDL: Lenguaje de Intercambio.....	31
BPEL: Lenguaje de Ejecución.....	33
Parte II. Análisis de Herramientas	38
Capítulo 4- Casos de Uso	39
JBoss Enterprise SOA Platform.....	39
Caso de Uso: Ventas Web.....	39
Caso de Uso: Vacaciones.....	40
Bonita.....	40
Caso de Uso: Ventas Web.....	40
ConsistGPA.....	41
Caso de Uso: Solicitud de Bienes y Servicios.....	41
Oracle Business Process Architect.....	42
Caso de Uso: Cotización a Efectivo.....	42
Oracle SOA Suite.....	46
Caso de Uso: Flujo Crediticio.....	46
Caso de Uso: <i>Loan Flow Plus</i>	47
Capítulo 5 - JBoss Enterprise SOA Platform	49
Características y Componentes.....	49
jBPM.....	50
Licencia.....	51
Suscripción.....	51

Requerimientos Mínimos del Sistema	52
Requerimientos Mínimos de Hardware.....	52
Sistemas Operativos Soportados	52
Instalación y Configuración	52
Instalación de la Suite	52
Instalación del Ambiente de Desarrollo	53
Persistencia de Datos	53
Bases de Datos Soportadas.....	53
Configuración Inicial	53
Configuración Bajo MySql 5.0	53
Soporte de Tecnología Java	56
Análisis de la Suite y ciclo de vida BPM soportado	56
Componentes de la Herramienta	56
Modelado y Desarrollo.....	56
Despliegue.....	60
Ejecución y Monitoreo	61
Conclusión General de la Suite.....	61
Capítulo 6 - BONITA	64
Introducción	64
Arquitectura	64
Características Principales de la Herramienta	66
Principales Beneficios.....	66
Requerimientos de Hardware y Software	67
Hardware.....	67
Software	67
Sistemas Operativos Soportados	67
Licencia.....	67
Instalación y Configuración	68
Instalación del Bonita Eclipse Designer.....	68
Instalación del Bonita Console.....	68
Persistencia de Datos	68
Base de Datos Soportadas	68
Configuración Inicial	69
Soporte de Tecnología Java	69
Análisis de la Suite y ciclo de vida BPM soportado	69
Componentes de la Herramienta	69
Modelado e Implementación.....	70
Despliegue.....	74
Ejecución y Monitoreo	75
Conclusión General de la Suite	76
Capítulo 7 - ConsistGPA	78
Introducción	78
Características Técnicas.....	78
Arquitectura de la Suite	79
Herramienta de Diseño de Procesos.....	79
Motor de Workflow.....	80
Documentos Digitales	80
Integrador	81
Reportes y Monitoreo de Procesos.....	81
Información Gerencial	82
Dispositivos Móviles.....	82
Parametrización y Configuración de ConsistGPA	82
Seguridad y Auditoría	82
Beneficios	82
Procesos de Negocio y Casos en ConsistGPA	82
Automatización de Procesos de Negocio en Diversas Industrias	83
Licencia.....	83
Requerimientos de Hardware y Software	83
Microsoft Visio 2003	83

ConsistGPA.....	84
Instalación y Configuración	84
Persistencia de Datos	84
Bases de Datos Soportadas.....	84
Configuración Inicial	84
Soporte de Tecnología Java	85
Análisis de la Suite y ciclo de vida BPM soportado	85
Componentes de la Herramienta	85
Modelado e Implementación.....	85
Despliegue.....	89
Ejecución, Monitoreo y Optimización	90
Conclusión General de la Suite.....	92
Capítulo 8 - Suite Oracle.....	94
Introducción.....	94
Arquitectura	94
Oracle Business Process Analysis (BPA) Suite	94
Oracle SOA Suite.....	96
Requerimientos de Hardware y Software	98
Oracle Business Process Architect.....	98
Oracle SOA Suite.....	99
Oracle Application Server (SOA)	100
Instalación y Configuración de la BPA.....	101
Instalación de Oracle Business Process Architect (Windows).....	101
Instalación de JDeveloper Version 10.1.3.2 (Windows).....	102
Instalación y Configuración de la SOA Suite	102
Instalación de Oracle Application Server SOA Suite (Windows).....	102
Conexión de la SOA Suite con JDeveloper.....	104
Soporte de Tecnología Java	105
Análisis de la Business Process Architect (BPA) y fases del ciclo de vida BPM soportadas.....	106
Componentes de la Herramienta	106
Modelado y Diseño	106
Transformación del Modelo a BPEL	109
Generar los Modelos BPEL	110
Ciclo Continuo de Desarrollo.....	113
Análisis de la SOA Suite y fases del ciclo de vida BPM soportadas	113
Componentes de la Herramienta	113
Implementación y Despliegue.....	114
Ejecución y Monitoreo.....	117
Conclusión General de la Suite.....	119
Capítulo 9 – Conclusión Final.....	121
Integración de la Suite en una sola Herramienta.....	121
Soporte para todo el ciclo de vida de los procesos de negocio	122
Adherencia a los estándares	122
Licencia.....	123
Soporte para SOA	123
Integración con entornos de desarrollo y soporte JAVA	125
Persistencia de datos	126
Conclusión	126
Referencias	128

Prólogo

BPM (*Business Process Management*) es una metodología para modelar procesos de negocios que tiene como objetivo mejorar, a través de la gestión sistemática, la eficiencia de dichos procesos de negocios que se deben modelar, analizar, automatizar, integrar, monitorear y optimizar de forma continua.

Actualmente todos los sistemas IT soportan e implementan de una u otra manera procesos de negocios. Lo que hace único a BPM es la separación de la lógica de los procesos de negocios respecto del código de la aplicación. Cuando se logra la independencia entre la lógica de los procesos de negocios y el código de la aplicación, se obtiene un incremento en la productividad, se reducen los costos y se tiene una rápida respuesta ante los cambios.

A pesar de las ventajas que brinda la separación entre la lógica de negocios y la implementación, en la medida en que la solución BPM se ajuste mejor a la plataforma de desarrollo, redundante en facilidades para la documentación, depuración y mantenimiento posterior. En este sentido las aplicaciones basadas en Java adhieren a metodologías de diseño y paradigmas orientados a objetos. Cuentan con un *framework* estándar y ampliamente usado para el desarrollo de aplicaciones distribuidas como es JEE y todos los *frameworks* arquitecturales basados en JEE que facilitan el desarrollo de aplicaciones distribuidas. De esta manera se garantiza modularidad, extensibilidad, adaptación a cambios y otras cuestiones relacionadas a los requerimientos no funcionales de las aplicaciones distribuidas como la seguridad, alta disponibilidad y tolerancia a fallas.

BPM separa no solamente el proceso respecto de los datos sino que explicita el flujo del mismo y permite separarlo de la forma en que se ejecuta.

La motivación principal de este trabajo es aprender sobre esta metodología de resolución de problemas que es nueva para nosotros. A su vez, no sólo nos interesa conocer los conceptos teóricos de la misma, sino su parte práctica, ya que esta metodología está siendo cada vez más usada para el desarrollo de aplicaciones.

Los resultados obtenidos en la realización de esta tesina fueron presentados en el artículo “Entornos para usar BPM en aplicaciones JAVA: un análisis comparativo”, expuesto en WICC 2009 (*Workshop* de Investigadores en Ciencias de la Computación) realizado en San Juan los días 7 y 8 de Mayo.

Página de trabajos presentados:

http://www.wicc2009.com.ar/index.php?option=com_content&view=article&id=61&Itemid=74

Parte I. Fundamentos

La Parte I de este informe, introduce los conceptos más importantes que hemos encontrado a largo de esta tesina. Primero dando una introducción a BPM y luego profundizando en los estándares más importantes de esta metodología. Entre los que se encuentran BPMN como estándar de notación, XPDL como lenguaje para poder realizar intercambios de modelos entre diferentes herramientas de modelado y BPEL un lenguaje de ejecución de procesos de negocio basado en *web services*.

Capítulo 1 – BPM

Introducción

Un proceso de negocio es una actividad del mundo real que consta de un conjunto de tareas lógicamente relacionadas, que cuando se realizan en la secuencia apropiada y siguiendo las reglas del negocio, producen una salida válida para el negocio. Por ejemplo, realizar una transacción bancaria [1].

Actualmente, los procesos de negocios se están tornando ineficientes y obsoletos. Con el tiempo, los mismos se han ido automatizando, distorsionando y complejizando la estructura original del proceso. Por ende, la necesidad de modelizarlos y optimizarlos es cada vez más importante para las organizaciones [2].

El cambio de un proceso de negocios involucra examinar el mismo con el objetivo de reducir el número de actividades, eliminando las tareas de menor importancia y simplificando el proceso en general. Con el fin de realizar dicho cambio, es necesario adoptar un enfoque que permita rediseñar los procesos de negocios. En teoría esto parece ser una tarea simple, pero la tasa de éxito de un cambio en un proceso de negocio es menor que el 50% [2]. Una de las razones que contribuyen a este problema es la incapacidad de predecir los resultados de un cambio en los procesos. Consecuentemente, las áreas problemáticas de un cambio en los procesos, solo pueden ser identificadas una vez que los mismos han sido físicamente implementados. Por lo tanto, tener la capacidad de analizar y evaluar un cambio en los procesos antes de su implementación puede tener un impacto positivo sobre la tasa de éxito de los futuros cambios. Una posible forma de lograrlo es utilizando modelado dinámico de procesos de negocios. BPM es una metodología de modelado dinámico [2].

BPM (*Business Process Management*)

Se llama *Business Process Management* (BPM) a la metodología empresarial cuyo objetivo es mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio, que se deben modelar, automatizar, integrar, monitorear y optimizar de forma continua.

BPMS (BPM Suite) es el conjunto de servicios y herramientas que facilitan la administración de procesos de negocio. BPMS además contempla soporte para interacción humana, e integración de aplicaciones, y es aquí la diferencia fundamental con la tecnología de *workflow*¹ existente. BPM integra en los flujos a los sistemas.

Las soluciones del tipo *workflow* se limitan a definir el flujo de actividades humanas o de documentos, obteniendo solamente el seguimiento de los procesos. En los casos que un participante del proceso requiere como parte de sus actividades ingresar datos en una aplicación, debe salir del ambiente del *workflow*, levantar la aplicación y luego de terminada su operación volver al *workflow* y registrar el cambio de estado o finalización de la actividad. En BPM todo está integrado en el mismo flujo, lo que resulta más

¹ El flujo de trabajo (*workflow* en inglés) es el estudio de los aspectos operacionales de una actividad de trabajo. Entre sus aspectos más importantes se encuentran: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas. Extraído de: http://es.wikipedia.org/wiki/Flujo_de_trabajo

natural para un participante ya que él completa su actividad dentro del flujo BPM e inmediatamente se actualizan los sistemas que deban actualizarse. Por ende, el resultado de trabajar con una solución basada en BPM es más confiable y es menos propensa a errores.

En la práctica, un flujo BPM visualmente es muy parecido a un *workflow*, la diferencia está en que es posible diferenciar las actividades realizadas por personas y las sistematizadas (realizadas por sistemas) y ambas aparecen en el mismo flujo.

El “valor agregado” de BPM es que ofrece una solución completa, que abarca todo el ciclo de vida de un proceso de negocio: análisis, modelado, ejecución y monitoreo de los procesos.

En BPM el modelo del proceso se convierte en el núcleo de la implementación del proceso como solución tecnológica. El modelo del proceso de negocio (su diseño), realizado por el área de negocios de una empresa, es en sí lo que se ejecuta sobre el servidor de procesos (también conocido como motor de ejecución). Dicho en otras palabras: la lógica central de negocio que bajo la tecnología tradicional se debía programar y desplegar en un servidor de aplicaciones (tradicional), ahora se reemplaza por un modelo que se despliega en el servidor de procesos con mucha menos intervención del área de tecnología (menos programación).

En la práctica una buena solución BPM debería poder ejecutar un proceso modelado por el área de negocio, sin la necesidad que el área de tecnología de la organización tenga que programar una sola línea de código, y obteniendo como solución algo equivalente a un *workflow* tradicional (sin integración de sistemas). Luego, el área de tecnología debería tomar este “*workflow*” e implementar los formularios de entrada (de interacción con usuarios) y los “servicios” (las actividades automatizadas) para completarlo en un flujo BPM.

Hacer que un modelo se convierta en un proceso ejecutable requiere de varias tecnologías habilitantes. Cuando éstas se proveen juntas se le llama BPMS (*Business Process Management Suite*) y las principales son:

- **Motores de Orquestación:** coordinan la secuencia de actividades según los flujos y reglas del modelo de procesos.
- **Herramientas de Análisis y *business intelligence*:** analizan la información producida por la ejecución del proceso en tiempo real.
- **Motores de Reglas (*Rule Engines*²):** ejecutan reglas que permiten abstraer las políticas y decisiones de negocio de las aplicaciones subyacentes.
- **Repositorios:** mantienen los componentes y recursos de los procesos (definiciones, modelos, reglas, etc.) disponibles para su reutilización en múltiples procesos.

² El motor de reglas de negocio es un sistema que se configura para dar servicio a las necesidades de negocio a través de la definición de objetos y reglas de negocio, el software se rige por flujos que derivan responsabilidades a los distintos cargos de la empresa repartiendo así el trabajo equitativamente y cuantitativamente, cuando, quien y donde tiene que desempeñar la tarea asignada. Extraído de: http://en.wikipedia.org/wiki/Business_rules_engine

- **Herramientas de Simulación y Optimización:** comparan el nuevo diseño de procesos con el desempeño operacional actual.
- **Herramientas de Integración:** integran el modelo con otros sistemas, por ejemplo los sistemas legados de la empresa.

BPM también es considerada una disciplina de administración, que requiere que las organizaciones apliquen un pensamiento centrado en los procesos. Es un enfoque estructurado que emplea métodos, políticas, métricas, prácticas de administración y herramientas de *software* para mejorar la agilidad y el desempeño operacional [3].

En los últimos años, las técnicas de BPM han ayudado a reducir errores, minimizar costos e incrementar la productividad en las organizaciones, que van desde las empresas de fabricación de productos, proveedores de telecomunicaciones, empresas de seguros hasta departamentos gubernamentales.

BPM se complementa con otros enfoques para las mejoras en las organizaciones, en particular, se relaciona muy bien con la filosofía de diseño de *software* llamado SOA (*Service Oriented Architecture*³).

BPM ayuda a los directores de empresas u organizaciones a adaptarse a los cambios continuos de los procesos de negocios, mientras que SOA puede ayudar a los arquitectos de *software* a construir sistemas empresariales que se adapten a estos cambios continuos en los requerimientos.

Una de las claves del éxito de BPM es el conjunto de herramientas que incorpora para el análisis, comprensión y documentación de los procesos de negocios.

BPM es una especificación desarrollada por la OMG (*Object Management Group*). Este organismo internacional sin fines de lucro, es el encargado de desarrollar y mantener las especificaciones importantes de BPM [4].

BPMS y Ciclo de Vida de los Procesos

Las BPMS prestan apoyo en todo el ciclo de vida de los procesos de negocio, el cual se compone de las siguientes etapas:

- **Modelado de los Procesos de Negocio:** en esta etapa se crea o modela un proceso de negocio, también es aquí donde se definen mejoras, o cambios a los procesos para optimizarlos. En esta etapa el principal involucrado es el “**Analista de Negocios**”.

³ La Arquitectura Orientada a Servicios (en inglés Service Oriented Architecture), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio. Permite la creación de sistemas altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma estándar de exposición e invocación de servicios (comúnmente pero no exclusivamente servicios web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros. Extraído de: http://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios

- **Implementación:** en esta etapa se integran los componentes necesarios para implementar el proceso. El principal involucrado en esta etapa es el “**Ingeniero de IT**” para el caso de que los procesos se implementen como soluciones tecnológicas.
- **Ejecución de Procesos:** esta es la etapa en donde se despliega y ejecuta el proceso desarrollado previamente; en esta etapa los principales involucrados son los “**Participantes**” del proceso. Además aquí es cuando se recolecta la información para control, y seguimiento.
- **Control y Gestión:** esta es la etapa donde se le da seguimiento a los procesos, y donde se analiza la información de su ejecución, por ejemplo: indicadores de desempeño, cuellos de botella, caminos críticos, carga de trabajo, etc., su principal característica es que la información se analiza en tiempo real. En esta etapa los principales involucrados son los “**Supervisores, y la Gerencia**”.

Extraído de <http://soaagenda.com/journal/articulos/bpms-y-ciclo-de-vida-de-los-procesos/>

En este contexto y considerando la Suite Oracle (que será una de las estudiadas en esta tesina), se describe el ciclo de vida de los procesos mencionado desde el enfoque de dicha herramienta.

Ciclo de Vida de un Proceso en Oracle

El ciclo de vida BPM consiste de 3 fases:

- **Análisis de Procesos de Negocio:** captura, modela, simula y analiza los modelos de los procesos, con el objetivo de reducir riesgos y maximizar la eficiencia.
- **Ejecución de Procesos de Negocio:** implementa, despliega y testea el proceso de negocios.
- **Monitoreo de Procesos de Negocios:** gestiona procesos, monitorea métricas de negocios, analiza la *performance* y maneja alertas y excepciones.

Finalmente, para la mejora continua del proceso de negocio, las métricas de negocios recolectadas en la fase de **monitoreo** tienen que ser retroalimentadas en la fase de **análisis** para refinar los modelos de procesos. La Figura 1-1 muestra el ciclo de vida de un proceso de negocios usando las herramientas de Oracle. Vemos que todas las etapas del ciclo de vida están soportadas por alguna de las herramientas que componen la Suite.

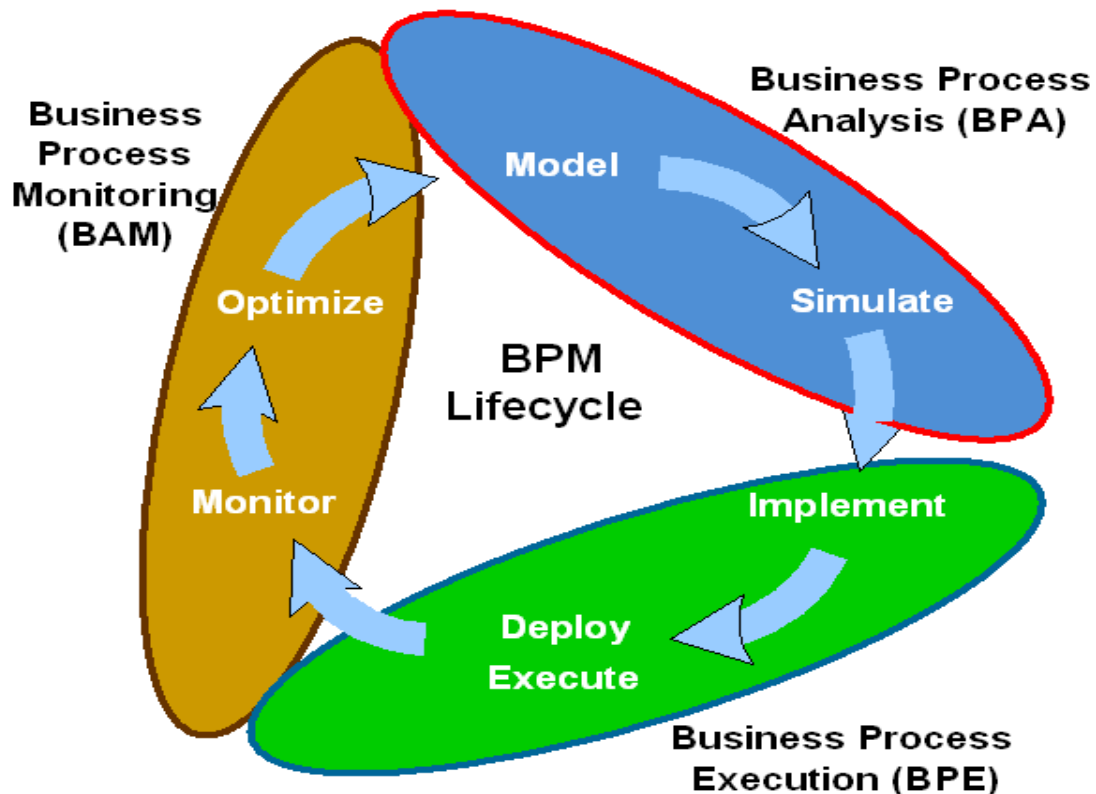


Figura 1 - 1: Gráfico del Ciclo de Vida de un Proceso y herramientas Oracle que las soportan. Extraída de: “Oracle Business Process Architect, Quick Start Guide Release 10.1.3.4”. Oracle Corporation. Pag 9.

Como ya lo hemos descrito Oracle BPA Suite es la herramienta de modelado destinada a los usuarios de negocios y la Oracle SOA Suite, es la plataforma de ejecución y monitoreo, destinada a los desarrolladores IT.

Los cambios en el modelo de procesos viven conjuntamente con el cambio frecuente de los requerimientos del negocio y el modelo de procesos debería ser un continuo reflejo de la implementación. Los cambios en el modelo de proceso de negocios deberían ser reflejados inmediatamente en la implementación y cualquier cambio en la implementación que afecte el flujo del negocio debería también reflejarse en el modelo de proceso de negocio. Oracle ha definido un metamodelo compartido entre las herramientas de análisis y las de desarrollo llamado *process blueprint* que permite sincronizar los modelos del negocio y los de IT a través del ciclo de vida del proceso de negocio. Esta solución facilita/alienta a los sectores de negocios e IT a trabajar en forma separada sobre una definición de proceso compartida.

Fases del Ciclo de Vida

1. Modelado del Proceso

Una vez que el proceso fue identificado, necesita ser documentado con precisión. Este es un paso importante y muchas veces determina el éxito de la implementación. Los Analistas de Negocio diseñan el modelo del proceso colaborando con los diferentes

interesados, tal como los usuarios de negocio, y los dueños de la línea de negocios, capturando los diferentes aspectos de un proceso.

Para empezar, se deben capturar las actividades claves del proceso en un Mapa de Proceso o Catálogo. Identificando y asociando las KPIs, SLAs, metas, riesgos y los objetivos de los procesos. Modelizar y organizar roles, datos y servicios para crear una librería compartida para la reutilización a través del proceso de negocio.

Luego, se deben descomponer los procesos principales en subprocesos y determinar los límites de los mismos. Enlazar subprocesos entre sí, donde la salida de uno se convierte en la entrada de otro. Utilizando una combinación de soluciones *top-down* y *bottom-up*, planeando los procesos en detalle. Los primeros niveles en un proceso son conceptuales o lógicos y nos conducen a flujos más detallados. Los últimos dos niveles son los destinados a la implementación del proceso.

A medida que el proceso iterativo de BPM se despliega, los trabajadores desarrollan un mayor conocimiento del flujo de sus procesos: el sector IT gana un entendimiento más sofisticado de los procesos de negocio que están implementando y el sector de Negocios desarrolla un mejor entendimiento de cómo la tecnología puede soportar sus necesidades, lo que incrementa sus habilidades de sugerir cambios destinados a la funcionalidad de las aplicaciones que soportan sus necesidades.

El contenido de un proceso es almacenado en el *business repository*. Múltiples usuarios pueden acceder y trabajar concurrentemente en el contenido del proceso, y los accesos simultáneos de varios usuarios al mismo modelo son provistos vía *check-in* y *check-out* implícitos.

2. Simulación y Análisis de un Proceso

El modelado es seguido de la simulación y el análisis de los modelos con el objetivo de optimizar el proceso de negocio para reducir riesgos e incrementar la flexibilidad. La simulación y análisis, es usualmente realizada por sofisticados analistas de negocios para producir modelos de procesos de negocios innovadores y óptimos. Con la simulación se puede medir la *performance* actual de los modelos, utilizando datos operacionales en tiempo real o histórico para medir costo y tiempo. También es posible analizar qué ocurre en los diferentes escenarios.

Los parámetros de tiempo pueden ser constantes o expresados en una distribución estadística. En adición a los parámetros de costo y tiempo para cada actividad, la simulación permite especificar la probabilidad para los diferentes caminos de procesos así como también la tasa en la que las instancias son esperadas. El componente de simulación también puede ser usado para realizar simulaciones basadas en roles y determinar la tasa de utilización de un rol a través de un conjunto de procesos de negocios. La simulación animada y gráfica asiste a los usuarios de negocios para determinar:

- **El tiempo del ciclo promedio:** tiempo promedio calculado a través de un conjunto de instancias.
- **El análisis total:** número de instancias procesadas en un tiempo dado.

- **El costo basado en actividades:** cantidad total de dinero gastado en producir y mantener un producto o servicio.
- **El camino crítico:** camino que tarda la mayor cantidad de tiempo.
- **El punto débil:** tarea en la que las instancias se encolan debido a la retención de recursos.
- **La utilización de recursos**

3. Publicación y Revisión de Modelos de Procesos

Los modelos de procesos de negocios que están destinados a revisión, pueden ser publicados en un portal seguro, en tiempo de diseño. Este portal provee acceso al contenido del proceso, incluyendo los resultados de la simulación, a toda la comunidad de negocios distribuida a través de diferentes ubicaciones geográficas. Este acceso está basado en una política de roles. Los diferentes participantes del negocio pueden revisar y proveer una retroalimentación sobre el contenido del proceso, y los analistas de negocios pueden incorporar esta retroalimentación nuevamente en el modelo de proceso.

4. Modelo de Negocio Compartido con IT

Los modelos de proceso de negocio son abstractos y tienen que ser implementados antes de ser utilizados. Estos modelos no sólo son para propósito de documentación, sino también son el punto de partida para la implementación. Una vez que se decide qué versión específica del modelo del proceso está lista para la implementación, un proceso *blueprint* es generado dentro de la arquitectura y es almacenado en el *business process repository*.

5. Generación e Implementación de un Modelo BPEL Ejecutable a partir de un Modelo de Proceso de Negocio Abstracto

Los desarrolladores IT acceden a los *blueprint* conectándose al *business repository* y lo utilizan como punto de partida para crear modelos BPEL ejecutables desde la herramienta IT (el componente de diseño BPEL de Oracle JDeveloper). El desarrollador IT dispone de dos vistas: la vista *blueprint* que muestra el proceso de negocio basado en BPMN y la vista BPEL que muestra el modelo ejecutable. El diseñador BPEL ofrece un diseño visual que puede ser almacenado en un formato BPEL estándar. Los artefactos BPEL son generados desde el *blueprint* promoviendo la automatización rápida del proceso y reduciendo los huecos de implementación en la traducción de los requerimientos de negocio directamente en una definición BPEL casi lista para desplegar. El paso siguiente es completar el esqueleto BPEL generado y desplegarlo en lo más alto del BPEL *process manager*, que es el motor en tiempo de ejecución de la Oracle SOA Suite. El desarrollador IT crea servicios de negocios, enlazándolos dentro de procesos de negocios finales y aplicaciones compuestas.

6. Desarrollo de Proceso Continuo

Compartir el modelo de procesos de negocio con el desarrollador IT no involucra un solo paso, sino que es un proceso de naturaleza iterativa. La gente de negocios puede

continuamente refinar y enriquecer el modelo de proceso, mientras que al mismo tiempo preservan los lineamientos negocios-IT a través de metadatos compartidos (*process blueprint*). El proceso ejecutable está en un bloqueo con respecto al modelo de proceso de negocios, a través del ciclo de vida del proceso. Los desarrolladores IT obtienen alertas en tiempo real sobre la creación de nuevas versiones del *process blueprint* y pueden ver y modificar estos procesos en paralelo utilizando el componente BPEL *process designer* de la Oracle SOA Suite. Los cambios a nivel de negocio pueden ser mezclados automáticamente con cualquier cambio realizado por los desarrolladores, para garantizar que los procesos implementados están en línea con las expectativas de los usuarios de negocios. El trabajo de implementación hecho por IT es preservado durante el proceso de mezclado, promoviendo el rápido desarrollo y prototipado. También, el desarrollador IT puede realizar cambios al flujo del negocio dentro de la vista *blueprint* y esto se ve reflejado en la herramienta *business architect* cuando el modelo BPEL ejecutable es guardado en el *business repository*. Los usuarios de negocios pueden revisar e incorporar estos cambios en el modelo de proceso de negocios para producir una nueva versión.

7. Despliegue y Testeo

El desarrollador IT despliega y testea los procesos ejecutables en el Oracle BPEL *process manager*. La plataforma Oracle SOA permite el testeo automático de los procesos ejecutables derivados del modelo de procesos, simulando servicios, proveyendo análisis de cobertura de código y soportando actividades de aserciones.

8. Gestión y Monitoreo

Los procesos que fueron desplegados y están en ejecución pueden ser gestionados utilizando Oracle *application server control* y monitoreados utilizando Oracle BAM. Esta última herramienta provee visualizar y capturar las métricas de negocios en tiempo real, desde los procesos de negocios, sistemas y otras fuentes, con el propósito de analizar *performance*. Esto les permite a los ejecutivos de negocios medir constantemente la *performance* actual, monitorear los acuerdos a nivel de servicio (SLAs) a través de varios servicios y procesos de negocios en la empresa. La herramienta permite la correlación de los indicadores clave de performance (KPIs) con la opción de profundizar en el proceso de negocios actual. Los gerentes de Negocios pueden entender los cuellos de botella y retardos de los procesos de forma muy efectiva utilizando *dashboards* y pueden tomar acciones correctivas si hay cambios en el entorno del negocio.

Extraído del paper: Oracle Business Process Architect, Quick Start Guide Release 10.1.3.4.

BPMM (Business Process Maturity Model)

El modelo de madurez de los procesos de negocios describe un camino de evolución que guía a las organizaciones a la maduración de los procesos de negocios. Proporciona un modelo de referencia para la evaluación de los procesos dentro de la empresa y ayuda a priorizar las mejoras a los mismos. También, se puede utilizar para evaluar los riesgos en el desarrollo y despliegue de nuevas aplicaciones empresariales que soportan procesos de negocios [4].

BPMM se basa en el *framework*⁴ de madurez de procesos de negocios de Watts Humphrey, que es también el responsable del ampliamente respetado CMMI⁵ (*Capability Maturity Model for Integration*) que se utiliza para ayudar a las organizaciones en los procesos de ingeniería de *software*. BPMM en contraste a CMMI provee un camino detallado para la mejora de los procesos de negocios. CMMI, define múltiples modelos de madurez, siendo muchos de ellos simples descripciones de los requerimientos que necesita una organización para estar en cada nivel sin detallar los pasos necesarios para pasar de un nivel a otro. BPMM organiza estos niveles logrando que las mejoras incorporadas en un nivel provean las bases para la construcción y el mejoramiento de los siguientes niveles. Por lo tanto, BPMM proporciona un camino para la mejora de los procesos, ayudando a identificar las deficiencias en los mismos [4].

La adopción de BPMM cubre las necesidades insatisfechas del mercado que apoya fuertemente la misión de la OMG y el uso de sus estándares en los productos y servicios brindados por los vendedores. Las organizaciones necesitan un método comprensivo para la evaluación de sus procesos de negocio y ambiente de organización, que los ayude a determinar los riesgos de realizar una implementación de sus aplicaciones empresariales [5]. Además de la simple identificación de los riesgos, este método proporcionará un camino para realizar las mejoras necesarias para incrementar el éxito y los beneficios de las aplicaciones empresariales. Para lograr esto, BPMM provee un *framework* que contribuye a los procesos de desarrollo e implementación de dichas aplicaciones empresariales [5].

La adopción del estándar BPMM apunta al menos cinco desafíos para el éxito de los sistemas empresariales:

1. Análisis de riesgos y debilidades de los procesos de negocios: las gerencias de las organizaciones cuentan con pocos estándares para valorar los *workflows* de sus procesos de negocios, necesitando métodos probados para analizar los riesgos y debilidades de los mismos.
2. Valoración de los modelos de procesos: las gerencias de las organizaciones cuentan con escasos métodos probados para valorar la fidelidad que existe entre cómo las tareas son realmente ejecutadas y cómo fueron descritas en los modelos de los procesos. Este problema compromete la validez de los requerimientos del sistema, la exactitud de los casos de uso y modelos de representación y la efectividad de la aplicación en uso.
3. Reducción de las aplicaciones empresariales: las gerencias de las organizaciones a menudo son inconscientes del grado de crecimiento de la organización, aumentando la complejidad de los procesos de negocio, dando como resultado varias maneras de ejecutar tareas similares. La creación de procesos estándares simplifica los requerimientos de las aplicaciones empresariales y como resultado se reduce la complejidad de las mismas.

⁴ Un *framework* básicamente es una estructura conceptual que se utiliza para resolver problemas complejos. Extraído de: <http://en.wikipedia.org/wiki/Framework>

⁵ CMMI es un modelo para la mejora de procesos que proporciona a las organizaciones los elementos esenciales para procesos eficaces. Extraído de: <http://en.wikipedia.org/wiki/CMMI>

4. Valoración de los servicios de proveedores: las gerencias de las organizaciones cuentan con pocos métodos probados para valorar la capacidad de los servicios prestados por proveedores en la tercerización de sistemas y otros servicios de negocios, y para determinar si cumplen o no con los parámetros acordados en los contratos.
5. Funcionamiento ágil de la organización y reducción de costos operacionales: las gerencias de las organizaciones necesitan guías que les permitan implementar los procesos de negocios requeridos para el funcionamiento ágil de la organización y para reducir los costos operacionales [5].

Existen muchas empresas u organizaciones que desperdician millones de dólares antes de terminar sus sistemas empresariales. Muchos de estos fracasos se deben a la tecnología utilizada así como también al estado en el que se encuentran estas organizaciones a la hora de realizar el despliegue de sus sistemas empresariales. Pero la mayoría de estos fracasos son atribuidos a las deficiencias a la hora de definir los procesos de negocios que son el objetivo de los sistemas empresariales. La predisposición de las organizaciones para el despliegue de tecnología es la motivación que hace que BPMM esté disponible como estándar [5].

¿Qué es BPMM?

Al final de la década del 80 Watts Humphrey comenzó a definir el *framework* de madurez de procesos, basándose en ideas que había desarrollado con un grupo de colegas en IBM. El *framework* de Humphrey fue implementado en el Modelo de Capacidad y Madurez para *software* (CMM, Paulk, et al., 1995) y más tarde en el Modelo de Capacidad y Madurez para la Integración (CMMI, Chrissis, et al., 2002). Este último se ha convertido en el estándar para evaluar la capacidad de desarrollo del *software* por parte de las organizaciones [5]

Los principios fundamentales de CMMI son los siguientes:

- **Evaluación de los atributos de un proceso de negocios:** los atributos de un proceso serán evaluados para determinar su capacidad para contribuir a los objetivos de la organización.
- **Madurez de la organización:** los procesos no pueden perdurar a no ser que la organización sea bastante madura para sostenerlos.
- **Mejora de los procesos:** la mejora de procesos es más acertada si se implementa como un programa de cambio organizacional guiado por etapas, permitiendo alcanzar de manera predecible los estados de madurez de la organización.
- **Niveles de Madurez:** cada etapa o nivel de madurez necesario sienta una base sobre la que pueden sostenerse e introducirse mejoras en el futuro.

El Modelo de Madurez de Procesos de Negocio (BPMM) sigue rigurosamente los principios del *framework* de madurez de procesos de Humphrey's y fue desarrollado por los co-autores de la CMM para *software* y CMMI [5].

Aplicación de BPMM

Hay cuatro formas principales en las se utiliza BPMM y cada una de ellas tiene diferentes requisitos para las evaluaciones.

- **Programas de mejoras de procesos de negocio:** BPMM fue diseñado para incorporar programas de mejoras y éste es su uso más frecuente. Los programas de mejoras deben ser iniciados con un análisis de la organización teniendo en cuenta sus fortalezas y debilidades.
- **Evaluación de riesgo para el desarrollo y despliegue de aplicaciones empresariales:** BPMM será usado para identificar los riesgos de la implementación de sistemas y proveerá una guía de las acciones que deben ser tomadas antes del despliegue del sistema.
- **Evaluación de la capacidad de los proveedores:** las organizaciones necesitan un estándar abierto y confiable mediante el cual puedan evaluar la capacidad de sus proveedores para conocer la calidad de sus servicios, precio y funcionalidad.
- **Pruebas de referencia:** la gerencia de la organización podría evaluar donde está posicionada la misma en relación a la madurez de los procesos de negocios en su segmento de industria [5].

Niveles de Madurez

Como todos los modelos de madurez guiados por el *framework* de madurez de procesos, BPMM está dividido en 5 niveles de madurez que representan los diferentes escenarios a través de los cuales una organización es transformada y sus procesos y capacidades son mejorados. Estos escenarios de madurez incluyen:

- **Nivel 1 – Inicial:** en este nivel los procesos de negocio son ejecutados muchas veces de forma inconsistente obteniendo resultados difíciles de predecir.
- **Nivel 2 – Gestionado:** en este nivel se estabiliza el trabajo de gestión local dentro de las unidades de trabajo, garantizando que pueda ser ejecutado de forma repetitiva y que cumpla los compromisos para grupos de trabajo primarios. Sin embargo, el desempeño de las unidades de trabajo que realizan tareas similares pueden utilizar diferentes procedimientos.
- **Nivel 3 – Estandarizado:** en este nivel los procesos son sintetizados hacia mejores prácticas, identificadas en los grupos de trabajo y son guiados para satisfacer las diferentes necesidades empresariales. Los procesos estándares proveen una economía de escala y una base para aprender de la experiencia y las medidas comunes.
- **Nivel 4 – Predecible:** en este nivel las capacidades habilitadas por los procesos estándares son explotadas y provistas en las unidades de trabajo. El rendimiento de los procesos es manejado estáticamente en todas partes del *workflow* para entender y controlar la variación de los procesos, permitiendo predecir los resultados desde los estados intermedios.

- **Nivel 5 – Innovación:** en este nivel las acciones de mejora tanto proactivas y oportunistas buscan innovaciones que permitan cerrar la brecha entre las capacidades actuales de la organización y las capacidades requeridas para lograr los objetivos del negocio.

Los niveles del 2 al 5 están compuestos de áreas de procesos que colectivamente habilitan la capacidad de lograr ese nivel. Cada área de proceso es designada para lograr objetivos específicos para la creación, soporte o sustentación del estado organizacional característico del nivel. Cada área de proceso consiste en una colección integrada por las mejores prácticas que indican “qué” se debería hacer pero no “cómo” debe hacerse. Así, las organizaciones tienen la libertad de definir sus propios métodos y soluciones para satisfacer las metas y objetivos de cada área de proceso [6].

BPMM y la Estandarización de la OMG

La OMG provee un ambiente comunitario y neutral para la creación de estándares abiertos que aceleran la aceptación del mercado hacia las nuevas tecnologías. Aunque los modelos de madurez y en particular el proyecto de BPMM es un nuevo tipo de norma de la OMG, será valioso para los consumidores de tecnología, y dará grandes beneficios para los estándares de desarrollo de sistemas y los métodos que implementan las técnicas estándar de la OMG. En efecto, como las empresas avanzan hacia una representación modelada de los procesos de negocios, la borrosa línea entre negocios y tecnología convergerá.

A través de su estrategia de BPM, la OMG está preparando el escenario para promover innovaciones en la aplicación de la tecnología. Sin embargo, los beneficios de estas aplicaciones se diluyen por la inmadurez de los procesos de negocios. En particular, el valor de las representaciones de los procesos de negocios mediante modelos está directamente relacionado con la madurez de los procesos que están siendo modelados. Mejorar los procesos de negocios a través de BPMM permitirá a las organizaciones aprovechar mejor la tecnología basada en estándares de la OMG y estándares relacionados con los procesos de negocios de otras organizaciones [5].

Capítulo 2 – BPMN

Introducción

BPMI (*Business Process Management Initiative*) es un organismo sin fines de lucro que ha desarrollado el estándar BPMN (*Business Process Modeling Notation*). La primera especificación de dicho estándar fue publicada en Mayo del 2004.

El objetivo primario de BPMN fue proveer una notación que sea legible y entendible para todos los usuarios de negocios, desde los analistas que realizan el diseño inicial de los procesos, hasta los responsables de desarrollar la tecnología que ejecutará estos procesos, y los gerentes de negocios encargados de administrar y realizar el monitoreo de los procesos. BPMN también soporta un modelo interno que permite generar ejecutables BPEL4WS. Así, BPMN crea un puente estandarizado para cubrir el hueco provocado por las diferencias entre el diseño de los procesos de negocios y su implementación.

BPMN define un diagrama de procesos de negocio (BPD) basado en una técnica adaptada de diagramas de flujo para la creación de modelos gráficos de operaciones de procesos de negocio. Un modelo de procesos de negocio, es una red de objetos gráficos que representan las actividades (por ejemplo tareas) y los controles de flujo que definen su orden de ejecución [7].

¿Por qué BPMN es importante?

El mundo de los negocios ha cambiado drásticamente en los últimos años. Los procesos son coordinados dentro de los límites naturales de las organizaciones y a su vez los mismos pueden interactuar con procesos de otras organizaciones. Actualmente un proceso de negocio abarca múltiples participantes y la coordinación puede ser compleja.

Hasta la aparición de BPMN no existía un estándar sobre técnicas de modelado desarrollado para estos fines. BPMN ha sido desarrollado para proveer a los usuarios de una notación estándar de forma análoga a como UML estandarizó el mundo de la ingeniería del *software* [3].

¿A quién está dirigido BPMN?

BPMN está dirigido a los analistas de negocios en el alto nivel y a los implementadores de procesos en el bajo nivel. Los analistas de negocio deberían fácilmente entender un diagrama de proceso de negocio en BPMN. Los implementadores de procesos deberían poder complementar el diagrama de proceso de negocio con mayor detalle con la intención de representar el proceso en una implementación ejecutable.

BPMN está dirigido a usuarios y proveedores de servicios que requieren comunicar los procesos de negocio de una forma estándar [3].

¿Cómo se ajusta BPMN con UML?

El advenimiento de BPMN, BPMS y sus lenguajes de ejecución no deja obsoleta la necesidad de desarrollos de sistemas, como los que se logran utilizando UML (*Unified Modeling Language*). Los desarrollos de sistemas siguen teniendo un rol importante en la arquitectura de procesos a nivel empresarial.

UML es un lenguaje que le facilita a los desarrolladores la especificación, visualización y documentación de modelos de sistemas de *software*. Está dirigido en líneas generales a los arquitectos de *software* e ingenieros de *software*. Fue desarrollado como un medio para mejorar el proceso de desarrollo de *software*, desde el diseño de la arquitectura hasta la implementación de la aplicación, para ser utilizado por personas con conocimientos técnicos (analistas de sistemas y programadores).

BPMN está dirigido a los analistas de negocio, arquitectos de sistemas e ingenieros de *software*. Fue desarrollado para mejorar todo el ciclo de vida del desarrollo de procesos desde el diseño de los mismos.

UML es desconocido para la mayoría de los analistas de negocio.

UML define un número de diagramas que se pueden clasificar en las siguientes 3 categorías:

1. Estructura estática de la aplicación
2. Comportamiento dinámico
3. Administración y organización de soluciones de *software*

De estas tres categorías, el comportamiento dinámico es el utilizado para modelar los procesos de negocio; los diagramas asociados son el de actividad UML y los de casos de uso. BPMN está emparentado con UML por el hecho que ambos definen una notación gráfica para los procesos de negocio similar a los diagramas de comportamiento de UML. Sin embargo, BPMN y UML usan enfoques muy diferentes para modelar procesos de negocio.

UML ofrece un enfoque orientado a objetos para modelar aplicaciones, mientras que BPMN toma un enfoque centrado en los procesos. Este enfoque es mucho más natural e intuitivo para los analistas de negocios. Con BPMN, el control y los mensajes de flujo entre procesos son primeramente modelados. Luego, se definen implícitamente los modelos de objetos para los procesos en vez de hacerse explícitamente como en UML. BPMN también ofrece la opción de explicitar el modelado de objetos de negocio que pueden ser expuestos a través de servicios de negocio en el flujo del proceso.

UML no posee una vista de implementación de los modelos de negocio.

UML es un ensamblado de diagramas que conforman un conjunto de buenas prácticas colectivas. Desafortunadamente, esto significa que sus diagramas no fueron diseñados específicamente para trabajar conjuntamente unos con otros. Como consecuencia, los desarrolladores sólo pueden modelar una parte de sus aplicaciones con UML, el nivel de implementación detallado no está cubierto.

En contraste, BPMN define un único tipo de diagrama que posee múltiples vistas derivadas del mismo meta-modelo de ejecución del proceso. Como resultado la implementación es una vista lógica del proceso en un lenguaje de ejecución de procesos de negocio (BPML) [8].

Elementos del Lenguaje

Un diagrama de proceso de negocio está compuesto de un conjunto de elementos gráficos. Estos elementos permiten desarrollar fácilmente diagramas simples, que resultarán familiares a la mayoría de los analistas de negocios. Los elementos utilizados para construir los diagramas fueron elegidos para ser distinguibles unos de otros y utilizar las figuras que son familiares a la mayoría de los diseñadores. Por ejemplo, las actividades se representan mediante rectángulos y las decisiones mediante rombos. Cabe destacar que uno de los objetivos del desarrollo de BPMN fue crear un mecanismo sencillo para la creación de modelos de procesos de negocio y al mismo tiempo ser capaz de manejar la complejidad inherente de los procesos de negocio. El enfoque adoptado para manejar estos dos requisitos fue la organización de la gráfica de los aspectos de la notación en categorías específicas. Éste provee un pequeño conjunto de categorías de notación que permite al lector del diagrama de procesos de negocio reconocer fácilmente los elementos básicos y comprender el diagrama.

Dentro de las categorías básicas de elementos se pueden incluir variaciones adicionales o información para soportar requerimientos complejos sin tener un cambio drástico en la mirada y sentido básico del diagrama. Las 4 categorías básicas de elementos son:

- *flow objects* (objetos que representan el flujo)
- *connecting objects* (objetos conectores)
- *swimlanes* (andariveles)
- *artifacts* (artefactos)

Flow Objects

Un BPD (diagrama de procesos de negocio) tiene un conjunto de tres elementos principales, llamados *flow objects*, que permiten a los diseñadores no recordar un gran número de figuras diferentes. Estos tres objetos son:

- **Events (Eventos):** se representa con un círculo y es algo que sucede en el transcurso de un proceso de negocio. Estos eventos afectan el flujo de los procesos y usualmente tienen una causa (*trigger*) o un impacto (*result*). Los eventos son círculos con el centro vacío para permitir diferenciar causas o impactos. Hay tres tipos de eventos de acuerdo al momento en que afectan al flujo: Inicial, Intermedio, o Final (ver las figuras de abajo, respectivamente).



- **Activity (Actividad):** se representa mediante un rectángulo con las esquinas redondeadas y es un término genérico del trabajo que realiza la organización. Una actividad puede ser atómica, no atómica o compuesta. Los tipos de actividades son: Tareas y Sub-Procesos. Un Sub-Proceso se distingue por un signo “+” en la parte de abajo del centro de la figura.



- **Gateway (Bifurcación):** se representa mediante un rombo y se utiliza para controlar la divergencia y convergencia de los **flujos de secuencia**. Puede ser utilizado para representar decisiones tradicionales, así como bifurcaciones (*forking*), combinaciones (*merging*) y unión (*joining*) de caminos. Las marcas internas en el rombo indicarán de qué tipo de comportamiento se trata [7].



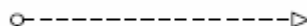
Connecting Objects

Los *flow objects* son conectados todos juntos en un diagrama para crear la estructura esqueleto básica de un proceso de negocio. Hay tres *connecting objects* que proveen esta función. Estos conectores son:

- **Sequence flow (Flujo de Secuencia):** es representado con una flecha llena y es utilizado para mostrar el orden (secuencia) de las actividades que serán ejecutadas en un proceso.



- **Message flow (Flujo de Mensaje):** es representado por una flecha con línea discontinua, con la punta vacía y un círculo al inicio de la línea. Es utilizado para mostrar el flujo de mensajes entre dos procesos participantes (entidades de negocios o roles de negocios) que envían y reciben dichos mensajes. En BPMN dos fondos (*pools*) separados en el diagrama representarán los 2 participantes.



- **Association (Asociación):** es representada por una flecha punteada y es utilizada para asociar datos, texto y otros artefactos con *flow objects*. Las asociaciones son utilizadas para mostrar las entradas y salidas de las actividades.



Para los diseñadores que requieren de poca granularidad de representación para crear modelos de procesos, los objetos y conectores antes mencionados proporcionan la habilidad de crear diagramas simples y entendibles. La Figura 2-1, ilustra un ejemplo sencillo de un modelo de proceso de negocio modelado usando los elementos básicos del lenguaje de notación.

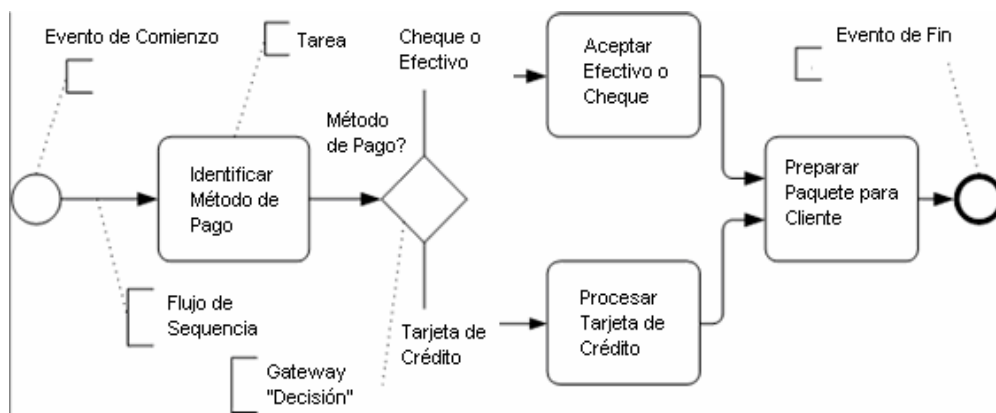


Figura 2 - 1: Ejemplo de un proceso simple. Extraída de: “*Introduction to BPMN*”. Stephen A. White. IBM Corporation. Pag 3, *Figure 1*.

Para los diseñadores que requieren de un mayor nivel de granularidad para crear modelos de procesos utilizados para análisis detallado o para ser manejados por un BPMS, necesitan incorporar más detalles a los elementos básicos [7]. La Figura 2-2 ilustra un modelo de un proceso de negocio con más grado de detalle.

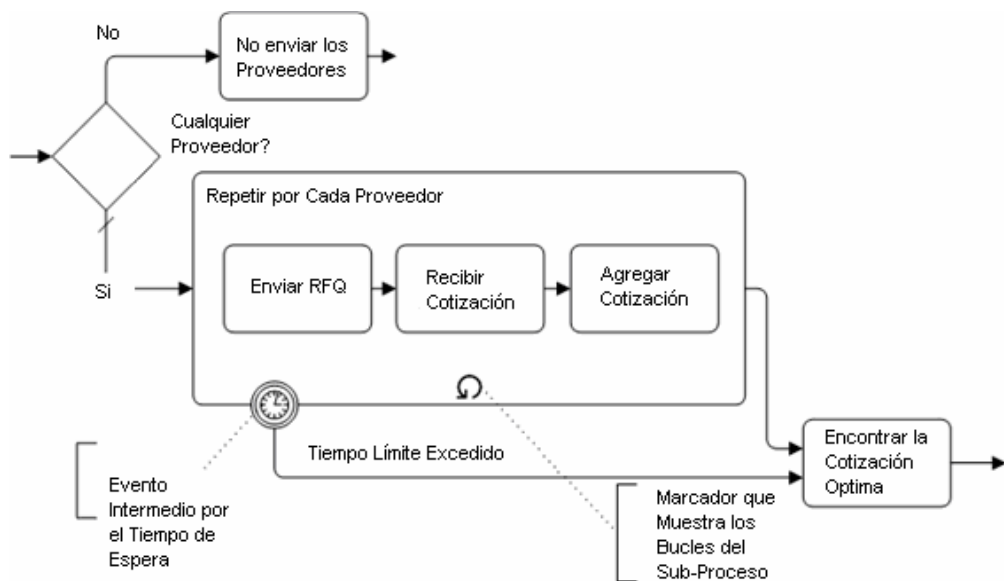


Figura 2 - 2: Fragmento de proceso que utiliza elementos detallados. Extraída de: “Introduction to BPMN”. Stephen A. White. IBM Corporation. Pag 4, Figure 2.

Swimlanes

Muchas metodologías de modelado de procesos utilizan el concepto de “swimlanes” como un mecanismo para organizar actividades en categorías visuales separadas, con el objetivo de ilustrar diferentes capacidades o responsabilidades funcionales. BPMN soporta swimlanes con dos construcciones principales. Los dos tipos de swimlanes provistos por BPMN son:

- **pool:** representa un participante en un proceso. También actúa como un contenedor gráfico para dividir un conjunto de actividades desde otros *pools*, usualmente en el contexto de situaciones B2B (*Business to Business*).



- **lane:** es una subpartición dentro de un *pool* que se extiende dentro de toda la longitud del mismo, pudiendo ser de forma vertical u horizontal. Las *lanes* suelen ser usadas para organizar y categorizar actividades.

Nombre	Nombre
	Nombre

Los *pools* son usados cuando el diagrama involucra dos entidades o participantes de negocios diferentes que están físicamente separados dentro del diagrama.

Las actividades dentro de los *pools* separados son consideradas procesos auto-contenidos. Así, el flujo de secuencia no puede cruzar el límite de un *pool*. El flujo de mensajes fue diseñado como un mecanismo para mostrar la comunicación entre dos participantes, siendo ésta la forma de comunicar dos *pools* (u objetos dentro de un *pool*). La Figura 2-3 ilustra dos *pools* que representan dos entidades diferentes, y estos se comunican entre si mediante el envío de mensajes dentro de un proceso de negocios.

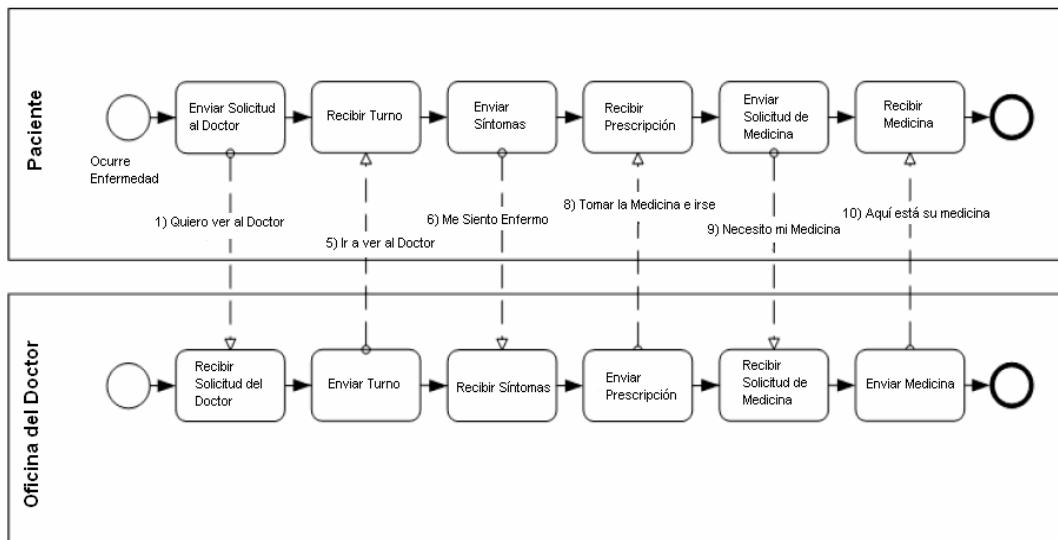


Figura 2 - 3: Dos *pools* que se comunican mediante flujos de mensajes. Extraída de: "Introduction to BPMN". Stephen A. White. IBM Corporation. Pag 5, Figure 3.

Las *lanes* suelen ser utilizadas para separar las actividades asociadas con una función o rol en una compañía. Los flujos de secuencia pueden cruzar los límites de las *lanes* dentro de un *pool* [7]. La Figura 2-4 muestra tres *lanes* diferentes que representan los roles de: Administración, Gestión y Servidor *web*.

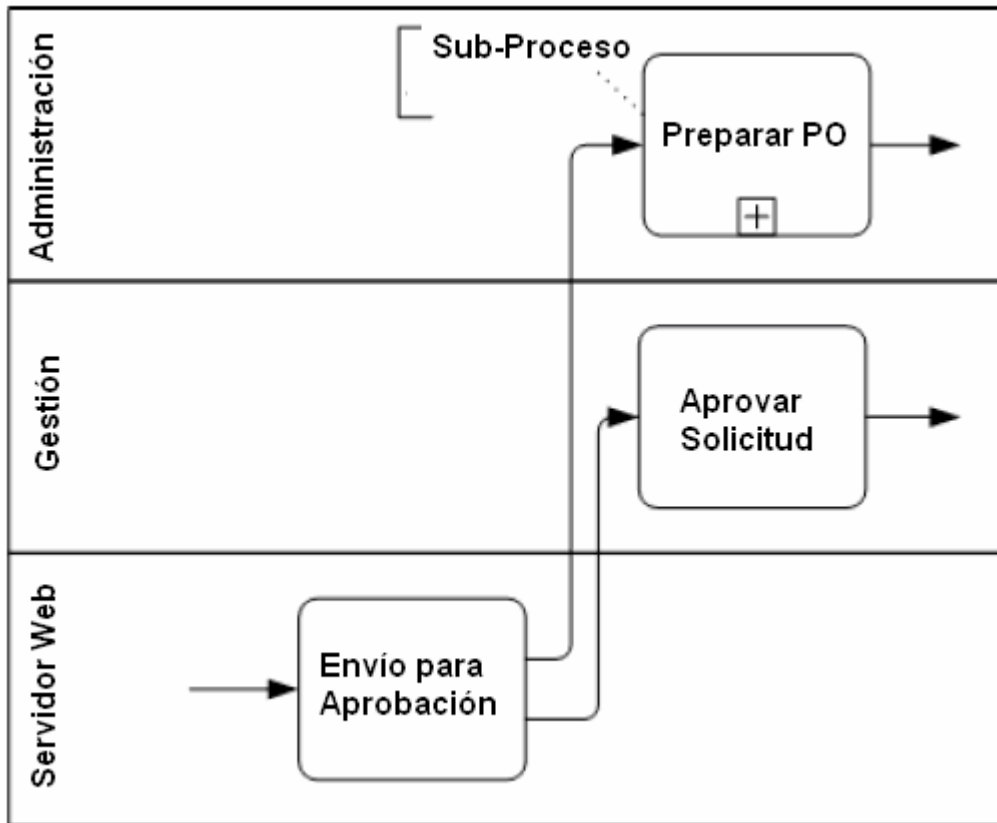


Figura 2 - 4: Tres lanes que representan diferentes roles de la organización. Extraída de: "Introduction to BPMN". Stephen A. White. IBM Corporation. Pag 5, Figure 4.

Artifacts

BPMN fue diseñado para permitir flexibilidad a los diseñadores y a las herramientas de modelado extendiendo la notación básica y proveyendo la habilidad para agregar un contexto adicional apropiado, en una situación específica, durante el modelado. Cualquier número de artefactos apropiados al contexto del proceso de negocios puede ser agregado al diagrama que está siendo modelado.

La especificación base de BPMN define solo tres tipos de BPD *artifacts*, que son:

- **Data object (Objeto de Datos):** proveen un mecanismo para mostrar como es requerido o producido un dato por las actividades. Se conectan a las actividades a través de las *associations*.



Nombre
[Estado]

- **Group (Grupo):** es representado por un rectángulo de esquinas redondeadas trazado con una línea discontinua. La agrupación puede ser utilizada para documentación o análisis, pero ésta no afecta al flujo de secuencia.



- **Annotation (Anotación):** proveen un mecanismo para proporcionar información adicional mediante un texto para ayudar al lector del diagrama.



Anotación de texto que permite al diseñador proveer información adicional.

Los diseñadores de modelos pueden crear sus propios artefactos para agregar más detalles de “cómo” el proceso es ejecutado; a veces para mostrar las entradas y salidas de una actividad dentro de un proceso. Igualmente, la estructura básica del proceso está dada por las actividades, bifurcaciones y flujos de secuencia, ya que dicha estructura no cambia con el agregado de los artefactos [7]. La Figura 2-5 ilustra el uso de los artefactos antes mencionados, para documentar el diagrama. Estos artefactos como antes mencionamos no afectan el flujo del proceso.

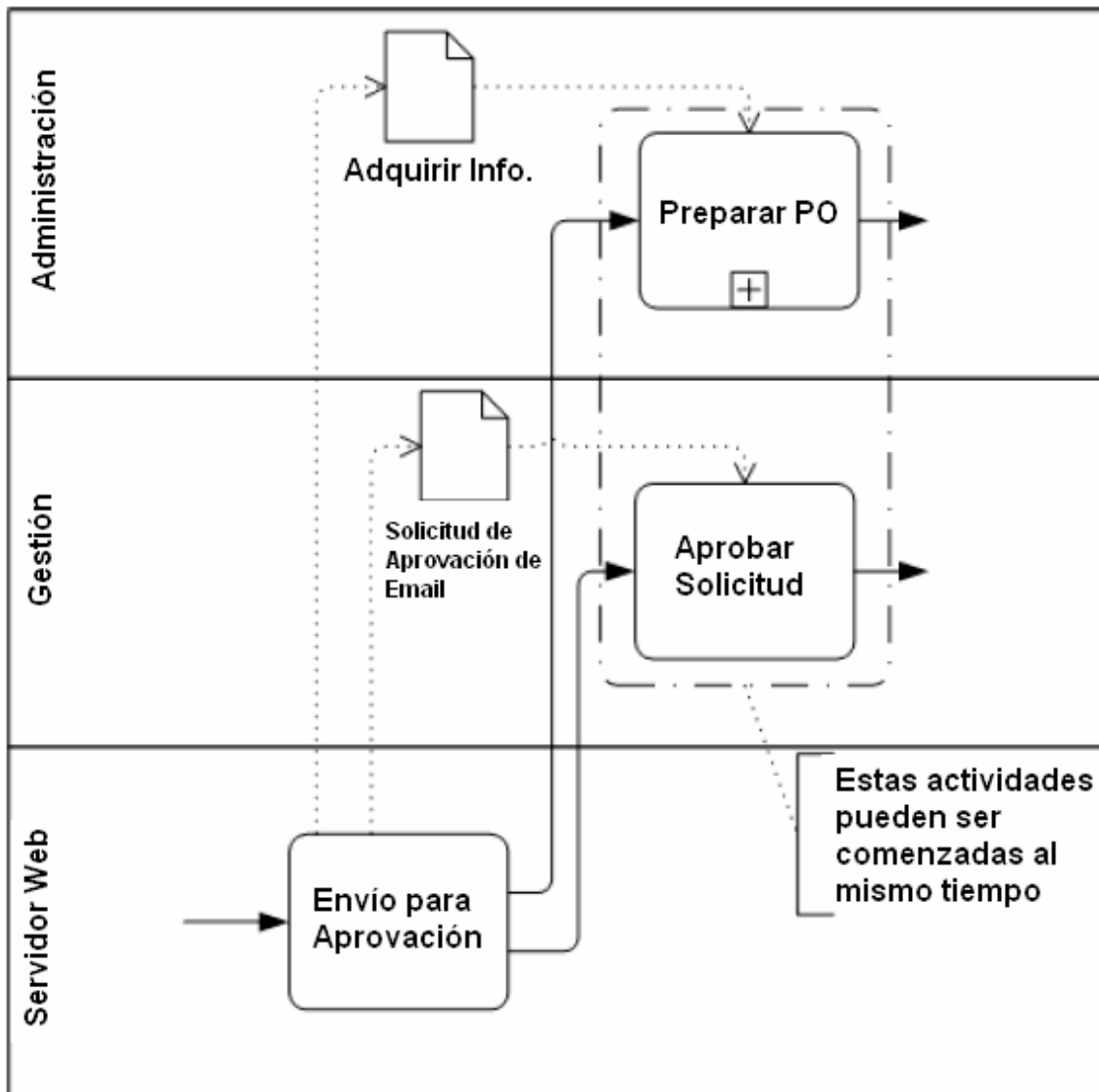


Figura 2 - 5: Ejemplo del uso de estos artefactos. Extraída de: “Introduction to BPMN”. Stephen A. White. IBM Corporation. Pag 7, Figure 5.

Usos de BPMN

El Modelado de Procesos de Negocios es utilizado para comunicar una amplia variedad de información de diferentes fuentes. BPMN fue diseñado para cubrir muchos tipos de modelos y permite la creación de segmentos de procesos, como también, el proceso de negocio completo, con diferentes niveles de fidelidad. Dentro de la variedad de objetivos de modelado de procesos, hay dos tipos básicos que pueden ser creados con un BPD:

- Procesos B2B Colaborativos.
- Procesos de Negocios Privados o Internos

Procesos B2B Colaborativos

Son procesos que muestran la interacción entre dos o más entidades de negocios. Los diagramas para estos tipos de procesos se ven generalmente desde un punto de vista global. Esto significa que no muestran en detalle ningún participante en particular, sino que se muestran las interacciones entre los mismos. Estas interacciones son descritas por una secuencia de actividades e intercambio de mensajes entre los participantes. El proceso define las interacciones de cada participante que son visibles para el público [7]. La Figura 2-6 ilustra dos *poles* que representan dos procesos B2B que colaboran mediante el envío de mensajes.

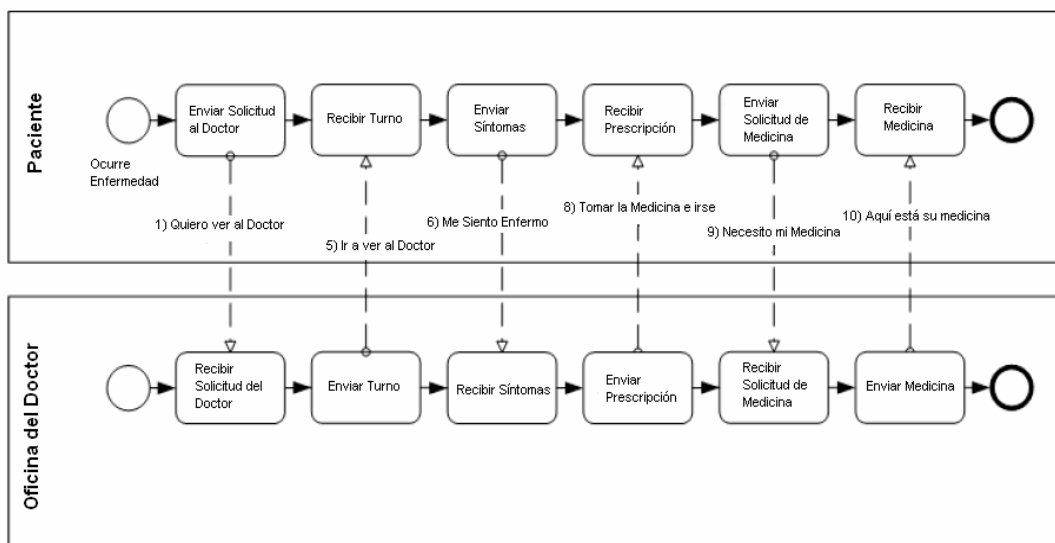


Figura 2 - 6: Diagrama que corresponde a Procesos B2B Colaborativos. Extraída de: "Introduction to BPMN". Stephen A. White. IBM Corporation. Pag 8, Figure 6.

Procesos de Negocios Privados o Internos

Un Proceso Interno de Negocio generalmente se centrará en una organización simple de negocio. Aunque los procesos internos a veces muestran interacciones con participantes externos, definen actividades que no son visibles al público, es decir, actividades privadas. Si se utiliza un *swimlane* entonces el proceso interno de negocio estará contenido en un *pool* simple. El Flujo de Secuencia del proceso estará contenido dentro del *pool* y no podrá cruzar los límites del mismo. Los flujos de mensajes podrán cruzar estos límites para mostrar las interacciones que existen entre dos procesos de negocio internos separados. Así, un solo diagrama de procesos de negocios puede mostrar varios procesos de negocios privados.

Diferentes Niveles de Precisión

El modelado de procesos de negocios generalmente empieza capturando las actividades de alto nivel y luego se explotan en niveles más bajos de detalle en diagramas separados. Podrían existir varios niveles de diagramas, dependiendo de la metodología usada para el desarrollo del modelo. Sin embargo, BPMN es independiente de cualquier metodología de modelado [7].

Ventajas de Modelar en BPMN

El desarrollo de BPMN fue un paso importante para reducir la fragmentación que existía entre las herramientas de modelado de procesos y las notaciones. El BPMI *notation working group* se basó en la experiencia de diferentes notaciones, buscando consolidar las mejores ideas de estas notaciones divergentes en una notación estándar más simple. Algunos ejemplos de notaciones que fueron revisadas son: UML *activity diagram*, UML EDOC *business processes*, IDEF, ebXML BPSS, *activity-decision flow (ADF) diagram*, RosettaNet, LOVeM, and *event-process chains (EPCs)*. Esta fragmentación de notaciones ha dificultado la amplia difusión y adopción de sistemas de administración de procesos de negocios interoperables. Una notación estándar bien constituida, reducirá la confusión entre los negocios y los usuarios finales de IT.

Otro factor que llevó al desarrollo de BPMN es que, históricamente, los modelos de procesos de negocios desarrollados por gente de negocios han sido separados técnicamente de la representación de procesos requerida por los sistemas diseñados para la implementación y ejecución de los mismos. Así, existía la necesidad de traducir manualmente el modelo original de los procesos de negocios hacia los modelos de ejecución. Dichas traducciones estaban sujetas a errores y hacían difícil la comprensión de la evolución y rendimiento de los procesos de negocios desarrollados por los propietarios de los mismos.

Para ayudar a disminuir la brecha técnica en el modelado, un objetivo clave para el desarrollo de BPMN fue crear un puente entre la notación para el modelado de procesos de negocios y los lenguajes de ejecución que implementarán los procesos en un BPMS. Los objetos gráficos de BPMN, soportados por un conjunto rico de atributos, han sido asociados al *business process execution language for web services (BPEL4WS v1.1)*, el estándar de facto para la ejecución de procesos [7].

Capítulo 3 – Estándares de Soporte BPM

XPDL: Lenguaje de Intercambio

La versión de XPDL (*XML Process Definition Language*) 2.0 fue creada por la WfMC (*Workflow Management Coalition*) y está destinada a ser usada como un formato estándar de archivos para BPMN. El propósito original de la primera versión de XPDL (1.0) fue dar soporte a la importación y exportación de la definición de un proceso de negocios. Esta característica es mantenida y mejorada por la segunda versión de XPDL.

Las especificaciones XPDL y BPMN resuelven el mismo problema de modelado desde diferentes implementaciones: XPDL provee un archivo en formato XML que puede ser usado para intercambiar modelos de procesos entre herramientas mientras que BPMN provee una notación gráfica para facilitar la comunicación humana entre usuarios de negocios y usuarios técnicos, de procesos de negocios complejos [9].

Una gran variedad de herramientas pueden ser utilizadas para analizar, modelar, describir y documentar procesos de negocios. La interfase de definición de procesos de negocios define un formato de intercambio común que soporta la transferencia de definiciones de procesos entre productos diferentes. La interfase también define una separación formal entre el desarrollo y los ambientes de ejecución, habilitando a una definición de procesos generada por una herramienta de modelado, poder ser usada como una entrada a un número de diferentes productos de ejecución de procesos.

La definición de procesos, generada por una herramienta de construcción (*build-time tool*) es capaz de ser interpretada por diferentes productos de ejecución (*run-time products*). Las definiciones de procesos transferidas entre estos productos o almacenadas en un repositorio separado, son accesibles a través del formato de intercambio común XPDL.

Para proveer un método común que acceda y describa definiciones de procesos, se estableció un modelo de meta-datos. Este modelo de meta-datos identifica las entidades utilizadas más comúnmente en una definición de procesos. Una variedad de atributos describe las características que posee este conjunto limitado de entidades. Basándose en este modelo, el fabricante de herramientas específicas puede transferir modelos vía un formato de intercambio común.

Uno de los elementos más importantes de XPDL es el constructor genérico que soporta atributos específicos de los fabricantes para el uso dentro de las representaciones comunes. La gramática de XPDL está directamente relacionada con los objetos y atributos contenidos dentro de las definiciones de los procesos. Esta solución necesita dos operaciones que deberán ser provistas por los fabricantes:

- Importación de la definición de procesos desde XPDL.
- Exportación de la definición de procesos desde la representación interna del vendedor a XPDL.

XPDL forma un estándar de intercambio común que habilita a los productos a continuar con el soporte de una representación interna arbitraria de la definición de procesos, con una función importar/exportar para mapear hacia/desde el estándar dentro del producto.

Una gran variedad de mecanismos diferentes pueden ser usados para transferir datos de la definición de procesos entre sistemas, acorde a las características de los diferentes escenarios de negocios. En todos los casos, la definición de los procesos debe ser expresada en una forma consistente, derivada desde el conjunto común de objetos, relaciones y atributos que expresan estos conceptos subyacentes [9]. La Figura 3-1 describe básicamente el concepto de intercambio de definiciones de procesos entre herramientas, pudiendo luego el archivo XPDL generado ser ejecutado por diferentes motores que soporten este estándar. El “meta modelo” describe las entidades de alto nivel contenidas dentro de la definición de procesos, sus relaciones y atributos. Algunos de estos atributos pueden ser definidos para propósitos de simulación y monitoreo. Este “meta-modelo” también define algunas convenciones para el agrupamiento de definiciones de procesos dentro de modelos de procesos relacionados y el uso de datos de definición común, a través de un número de diferentes definiciones de procesos o modelos [9].

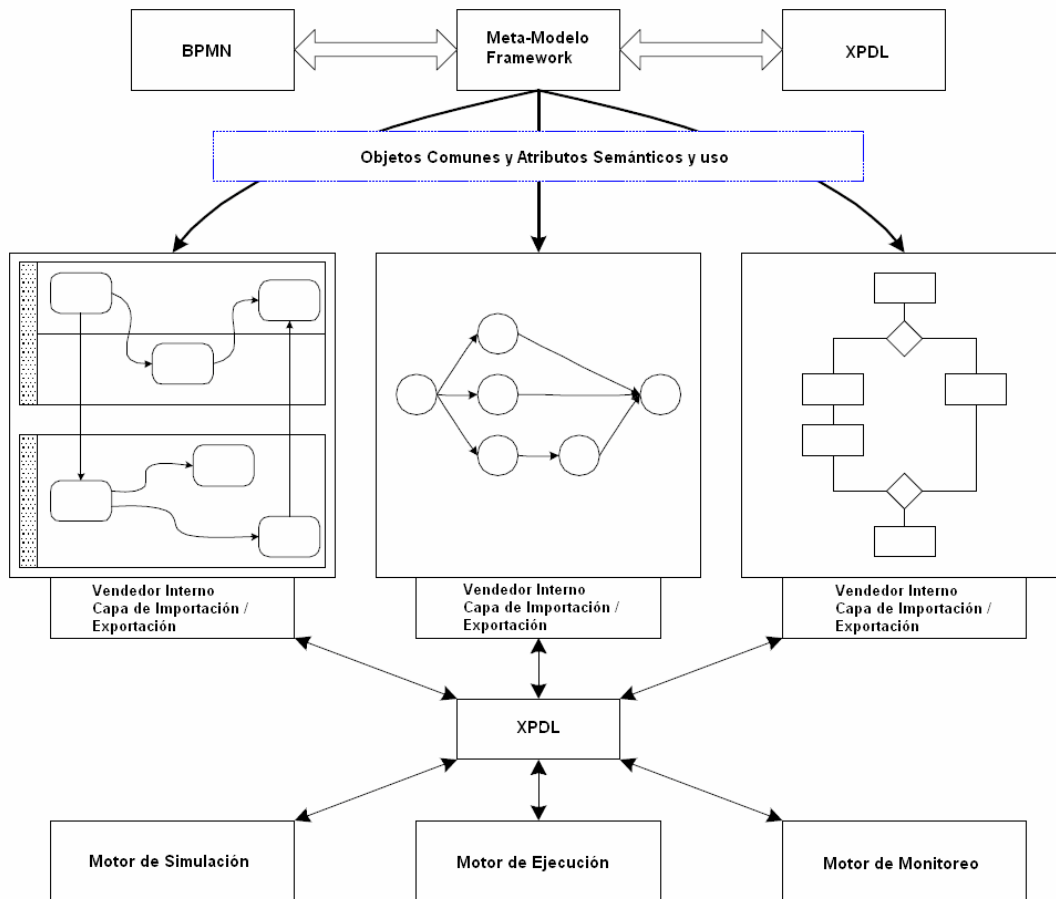


Figura 3 - 1: Intercambio de la definición de procesos. Extraída de: “*Process Definition Interface - XML Process Definition Language, Versión 2.00*”. Workflow Management Coalition (WFMC). Pag 9, *Figure 5-1*.

BPEL: Lenguaje de Ejecución

El objetivo de los *web services*⁶ es lograr la interoperabilidad entre aplicaciones, usando estándares *web*. Los *web services* utilizan un modelo de integración de bajo acoplamiento que permite una integración flexible de sistemas heterogéneos en una gran variedad de dominios incluyendo B2C (*Business to Customers*⁷), B2B (*Business to Business*) e integración de aplicaciones empresariales.

Los protocolos definidos para *web services* son los siguientes: SOAP (*Simple Object Access Protocol*), UDDI (*Universal Description Discovery and Integration*), WSDL (*Web Services Description Language*). SOAP define un protocolo de mensajes basado en XML para una interoperabilidad de servicios básica. WSDL introduce una gramática común para describir servicios. UDDI provee la infraestructura requerida para publicar y describir servicios de una manera sistemática. Todas estas especificaciones juntas permiten a las aplicaciones encontrarse unas a otras e interactuar siguiendo un modelo independiente de la plataforma y con bajo acoplamiento.

La integración de sistemas requiere más que la habilidad de interactuar utilizando protocolos estándares. Todo el potencial de *web services* como una plataforma de integración, será aprovechado solamente cuando las aplicaciones y los procesos de negocios sean capaces de integrar sus complejas interacciones utilizando un modelo de integración de procesos estándar. El modelo de interacción que es soportado directamente por WSDL es esencialmente un modelo sin estado de *request-response*⁸ o de interacciones de una vía.

Los modelos para la integración de negocios típicamente asumen secuencias de intercambios de mensajes punto a punto, ambos *request-response* y *one-way*, con manejos de estados e interacciones de larga duración involucrando dos o más partes. Para definir tales interacciones de negocios se necesita una descripción formal de los protocolos de intercambio de mensajes usados por procesos de negocios en sus interacciones.

WS-BPEL (*Web Services Business Process Execution Language*) define un modelo y una gramática para describir el comportamiento de un proceso de negocios basado en la interacción entre el proceso y entidades externas. La interacción con cada entidad externa, se realiza a través de interfases de *web services* y la estructura de la relación en el nivel de interfase es encapsulada en lo que se conoce como *partnerlink*. El proceso WS-BPEL define múltiples interacciones de servicio con estas entidades externas que se coordinan para lograr el objetivo de negocio, así como también el estado y la lógica necesaria para esta coordinación. WS-BPEL también introduce mecanismos sistemáticos para el tratamiento de excepciones de negocios y fallas en el

⁶ Un servicio web (en inglés *web service*) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios web. Extraído de: http://es.wikipedia.org/wiki/Servicio_Web

⁷ B2B se refiere a las transacciones efectuadas entre empresas: (comunicaciones de comercio electrónico) de empresa a empresa. Extraído de: <http://es.wikipedia.org/wiki/B2B>

⁸ *request-response*, también conocido como *request-reply*, es un patrón de intercambios de mensajes en el cual el solicitante envía un mensaje a un sistema receptor, el cual recibe el mensaje y procesa el pedido y por último retorna la respuesta. Es muy utilizado en arquitecturas cliente/servidor. Extraído de: <http://en.wikipedia.org/wiki/Request-response>

procesamiento. Además, introduce un mecanismo para definir cómo actividades individuales o compuestas dentro de una unidad de trabajo serán compensadas en casos donde ocurran excepciones. En la Figura 3-2 podemos apreciar un proceso WS-BPEL el cual se comunica con dos *PartnerLinks*: Cliente, es el encargado de iniciar la comunicación con el proceso WS-BPEL y también recibe los resultados de la ejecución. *CreditingRatingService*, es un *Web Services* que dado un Id de cliente, retorna una tasación crediticia para el cliente.

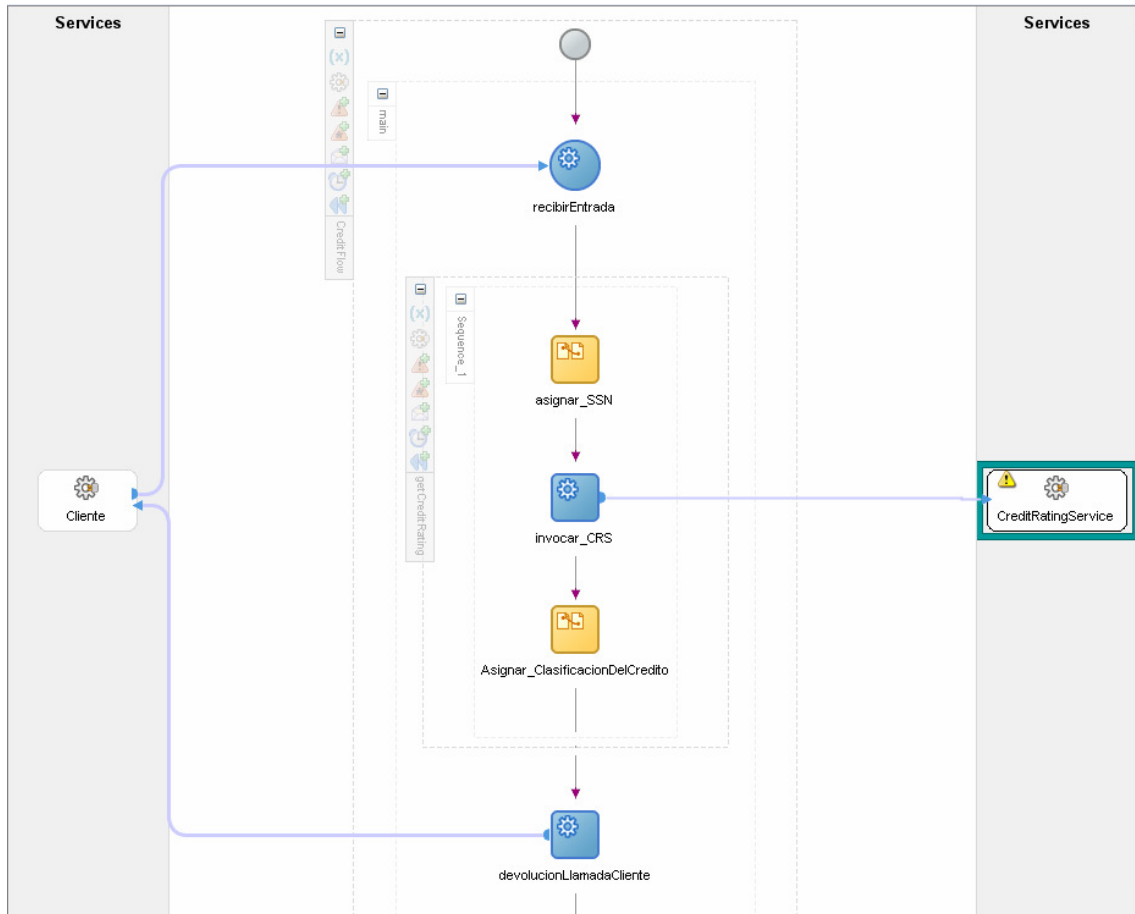


Figura 3 - 2: Ejemplo de proceso WS-BPEL desde la IDE jDeveloper.

El siguiente código WS-BPEL corresponde al XML que representa el proceso WS-BPEL anterior. El código muestra la sección de *PartnerLinks* donde se configuran las entidades externas *Cliente* y *CreditingRatingService* que interactúan con el proceso.

```
<!--
////////////////////////////////////
////////////////////////////////////
PARTNERLINKS
List of services participating in this BPEL process

////////////////////////////////////
////////////////////////////////////
-->
<partnerLinks>
  <!--
  The 'client' role represents the requester of this service. It is
  used for callback. The location and correlation information associated
```

```

with the client role are automatically set using WS-Addressing.
-->
  <partnerLink name="Cliente" partnerLinkType="client:CreditFlow"
    myRole="CreditFlowProvider"
    partnerRole="CreditFlowRequester"/>
  <partnerLink name="CreditRatingService"
    partnerLinkType="nsl:CreditRatingService"
    myRole="CreditRatingServiceProvider"
    partnerRole="CreditRatingServiceProvider"/>
</partnerLinks>

```

WS-BPEL utiliza varias especificaciones XML: WSDL 1.1

(<http://www.w3.org/TR/wsdl>), XML Schema 1.0 (<http://www.w3.org/TR/xmlschema-0/>), XPath 1.0 (<http://www.w3.org/TR/xpath>) y XSLT 1.0 (<http://www.w3.org/TR/xslt>). Los mensajes WSDL y los tipos de definición XML *schema* proveen un modelo de datos usados por procesos WS-BPEL. XPath y XSLT proveen soporte para la manipulación de datos. Todos los recursos externos son representados como servicios WSDL. WS-BPEL provee extensibilidad para adaptar futuras versiones de este estándar, específicamente el XPath y estándares relacionados usados en la computación XML. El siguiente código basado en el ejemplo anterior, muestra los componentes *Assign* del lenguaje que utilizan la operación *copy* de XPath, para copiar el valor de una variable a otra.

```

<scope name="getCreditRating">
  <sequence name="Sequence_1">
    <assign name="asignar_SSN">
      <copy>
        <from variable="inputVariable" part="payload"
          query="/client:CreditFlowProcessRequest/client:ssn"/>
        <to variable="invoke_CRS_process_InputVariable"
          part="payload" query="/nsl:ssn"/>
      </copy>
    </assign>
    <invoke name="invocar_CRS" partnerLink="CreditRatingService"
      portType="nsl:CreditRatingService" operation="process"
      inputVariable="invoke_CRS_process_InputVariable"
      outputVariable="invoke_CRS_process_OutputVariable"/>
    <assign name="Asignar_ClasificacionDelCredito">
      <copy>
        <from variable="invoke_CRS_process_OutputVariable"
          part="payload" query="/nsl:rating"/>
        <to variable="outputVariable" part="payload"
          query="/client:CreditFlowProcessResponse/client:creditRating"/>
      </copy>
    </assign>
  </sequence>
</scope>

```

Junto a cada proceso WS-BPEL se define un WSDL que lo describe. Dentro de este WSDL se definen los WSDL *port types*, que describen la API o interface del *web service*. Un *port type* es una lista de operaciones con entradas y salidas. Cada una de estas es un mensaje WSDL que debe ser previamente definido. En el siguiente código se muestra el ejemplo de la definición de un *portType* que representa la operación *process*, junto a sus entradas y salidas.

(Párrafo extraído de

http://www.eclipse.org/tptp/platform/documents/design/choreography_html/tutorials/ws_bpel_tut.html)

```

<!-- ~~~~~
PORT TYPE DEFINITION - A port type groups a set of operations into
a logical service unit.
~~~~~ -->

<!-- portType implemented by the CreditRatingService BPEL process -->
<portType name="CreditRatingService">
  <operation name="process">
    <input message="tns:CreditRatingServiceRequestMessage"/>
    <output message="tns:CreditRatingServiceResponseMessage"/>
    <fault name="NegativeCredit" message="tns:CreditRatingServiceFaultMessage" />
  </operation>
</portType>

```

Podemos decir que un proceso WS-BPEL es una definición reusable que puede ser desplegada de diferentes formas y en diferentes escenarios, mientras se mantenga un comportamiento uniforme en el nivel de aplicación a través de todos ellos [10].

Entre las características principales para utilizar WS-BPEL desde el punto de vista del potencial de los *web services*, se encuentran:

- Reutilizar servicios independientemente de la plataforma en la que fueron desarrollados.
- Desarrollar herramientas y otras tecnologías que aumentan el nivel de automatización y en consecuencia decrementan los costos en el establecimiento de procesos de negocios automatizados.

Los conceptos básicos de WS-BPEL pueden ser aplicados de dos formas: **abstracto** o **ejecutable**. Un **proceso abstracto** puede ser utilizado para describir un comportamiento de intercambio de mensajes observable para cada una de las partes involucradas, sin revelar su representación interna. Hay dos buenas razones para separar los aspectos públicos del comportamiento de un proceso de negocios de su aspecto privado o interno: una de ellas es que los negocios no quieren revelar sus decisiones internas y administración de datos a sus compañeros de negocios y la otra es que, el hecho de separar los procesos públicos de los privados, provee la libertad de cambiar los aspectos privados de la implementación de los procesos sin afectar el comportamiento observable. Este último, debe ser descripto claramente en una plataforma independiente. Un proceso abstracto WS-BPEL, es un proceso parcialmente especificado que no está destinado a ser ejecutado y debe ser explícitamente declarado como abstracto (*'abstract'*). Considerando que los procesos ejecutables son completamente especificados y así pueden ser ejecutados, un proceso abstracto podría ocultar alguno de los detalles operacionales requeridos que expresan los procesos ejecutables.

Todas las construcciones de procesos ejecutables están disponibles para procesos abstractos, en consecuencia los procesos WS-BPEL ejecutables y abstractos, comparten el mismo poder expresivo. En adición a estas características disponibles en los procesos ejecutables, los procesos abstractos poseen dos mecanismos para ocultar los detalles operacionales:

- El uso de marcas especiales
- La omisión

Los procesos abstractos cumplen un rol descriptivo, con más de un caso de uso. Uno de estos casos de uso podría usarse para describir el comportamiento observable de algunos o todos los servicios ofrecidos por un proceso ejecutable. Otro caso de uso, podría usarse para definir una “plantilla del proceso” que representa las mejores prácticas para un dominio específico. Esa “plantilla del proceso” podría capturar la lógica esencial del proceso de una forma compatible con una representación en tiempo de diseño mientras que excluye los detalles de ejecución que serán completados cuando se realice el mapeo a un proceso ejecutable.

A pesar del caso de uso específico, todos los procesos abstractos comparten una base sintáctica común. Ellos poseen diferentes requerimientos para el nivel de opacidad y restricciones sobre qué partes de la definición de un proceso podría ser omitida u ocultada. Los usos adaptados de los procesos abstractos tienen diferentes efectos en las restricciones de consistencia y en la semántica de los procesos. Algunas de estas restricciones requeridas no son manejables por el esquema XML. La base sintáctica común de los procesos abstractos especifica las características que definen el universo sintáctico de los procesos abstractos. Dada esta base, un perfil de uso provee la especialización necesaria y la semántica WS-BPEL ejecutable para un uso particular de un proceso abstracto.

Es posible el uso de WS-BPEL para definir un proceso de negocios ejecutable. Mientras que una definición abstracta de proceso WS-BPEL no requiere ser completamente especificada. El lenguaje efectivamente define un formato portable para los procesos de negocios que confían exclusivamente en los recursos de los *web services* y los datos XML. Además, tales procesos ejecutan e interactúan con sus entidades externas en una forma consistente a pesar de la plataforma de soporte y el modelo de programación utilizado.

La continuidad del modelo conceptual básico entre los procesos abstractos y ejecutables en WS-BPEL, hace posible importar y exportar los aspectos públicos incorporados en los procesos abstractos como procesos o plantilla de roles, mientras se mantiene la intención y la estructura del comportamiento observable. Esto aplica incluso a aquellos casos en que la implementación de los aspectos privados usa funcionalidad dependiente de la plataforma [10].

Parte II. Análisis de Herramientas

En la Parte II se detalla el análisis de herramientas que tienen soporte para BPM. Se eligieron dos herramientas *open source*: JBoss SOA Platform y Bonita, y dos herramientas de licencia propietaria: ConsistGPA y Oracle Suites (Oracle BPA, Oracle SOA Suite)

Para realizar dicho análisis se desarrollaron varios casos de uso, los cuales están descritos en el Capítulo 4. Luego hay un capítulo para cada una de las herramientas con el análisis de las mismas. Finalmente en el Capítulo 9 realizamos una comparación de las herramientas analizadas.

Capitulo 4- Casos de Uso

A continuación detallaremos algunos de los casos de uso que utilizamos para analizar cada una de las herramientas.

JBoss Enterprise SOA Platform

Caso de Uso: Ventas Web

Este caso de uso consiste de un proceso en el cual un cliente y un comercio están involucrados en un requerimiento de venta.

El cliente realiza el pedido el cual pasa a un empleado para su revisión. En esta revisión se puede solicitar más información al cliente. Una vez realizada dicha revisión se ejecutan dos tareas paralelas. Una de ellas es la entrega de los ítems de la venta y la otra el pago y el registro de la venta en el sistema. La Figura 4-1 muestra el diagrama de flujo antes explicado, donde podemos visualizar la comunicación entre los nodos y actividades pertenecientes al proceso.

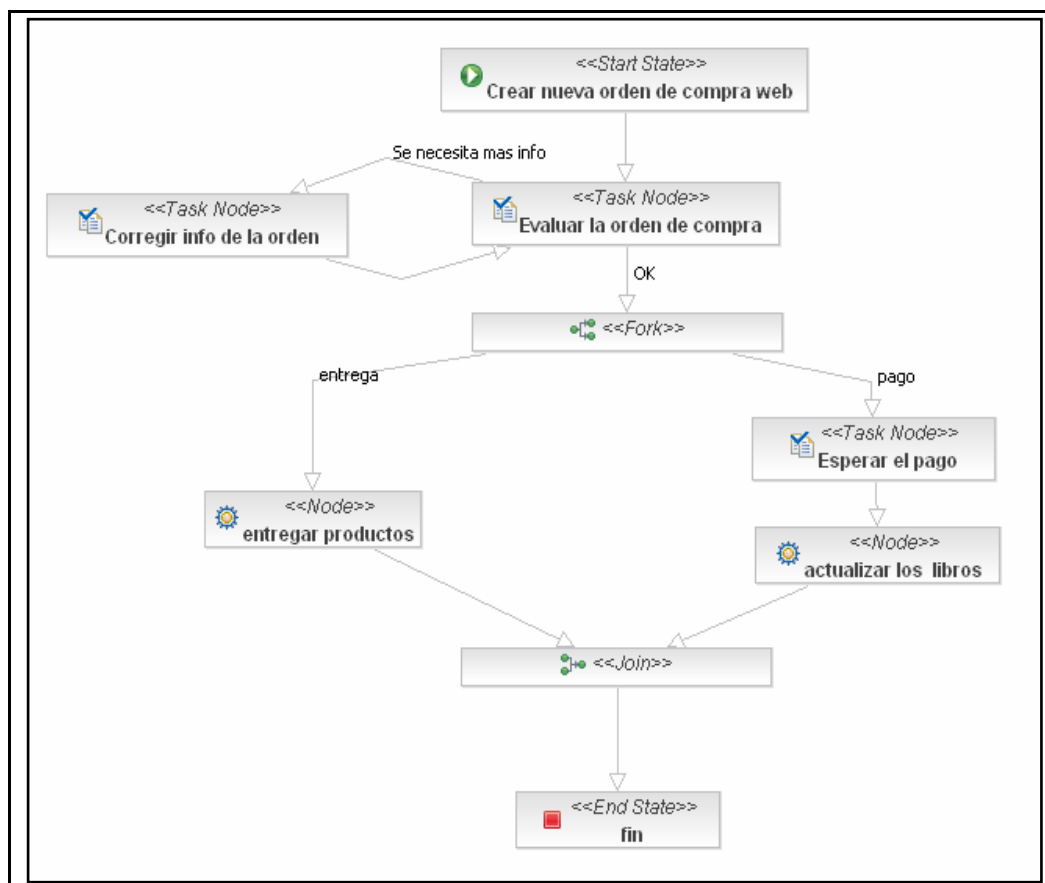


Figura 4 - 1: Proceso ventas *web*, desarrollado en la herramienta JBoss Enterprise SOA Platform.

Este caso de uso fue extraído de los ejemplos que trae la herramienta *jboss soa platform standalone* versión 4.2.0.

Caso de Uso: Vacaciones

Se trata de un caso de uso en el que el usuario es un empleado que registra un pedido de vacaciones llenando un formulario de datos. Una vez realizado el pedido se pasa a la evaluación del mismo donde se puede solicitar información adicional hasta que se decide si se aprueba o no la solicitud. La Figura 4-2 muestra el diagrama del caso de uso que mencionamos anteriormente.

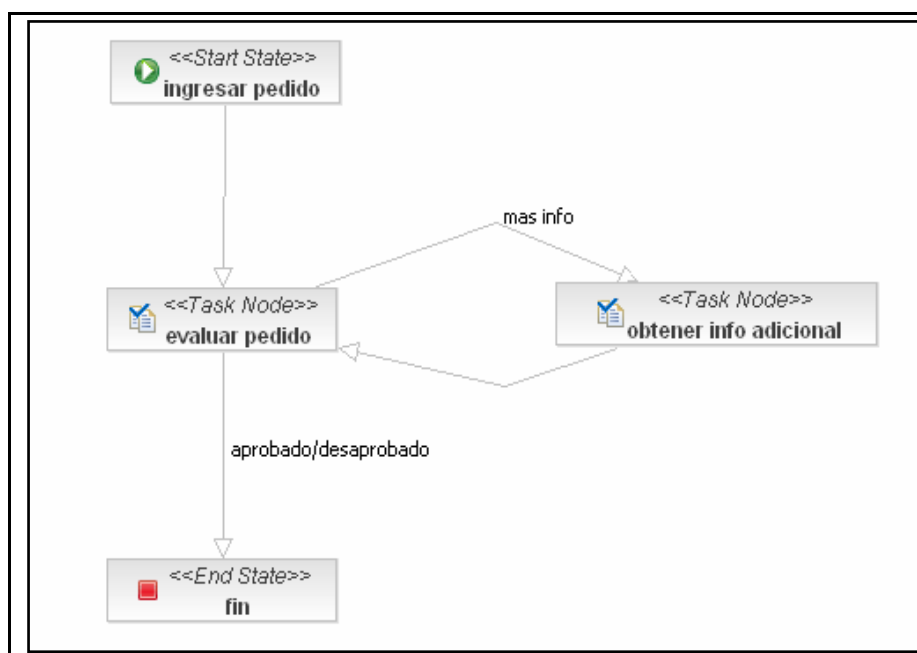


Figura 4 - 2: Proceso vacaciones, desarrollado con la herramienta JBoss Enterprise SOA Platform.

Este caso de uso fue extraído de: <http://docs.jboss.com/jbpm/v3/demos/movies/jbpm-overview-800x600.htm>.

Bonita

Caso de Uso: Ventas Web

Este caso de uso consiste de un proceso en el cual un cliente y un comercio están involucrados en un requerimiento de venta.

El cliente realiza el pedido el cual pasa a un empleado para su revisión. En esta revisión se puede solicitar más información al cliente. Una vez realizada dicha revisión esta puede ser rechazada, o en caso contrario pasa a realizarse el pago de la misma. Una vez realizado el pago, se puede solicitar una entrega “Express” que es opcional y luego se archiva la venta. En la Figura 4-3 tenemos el diagrama que representa el caso de uso Ventas *web*.

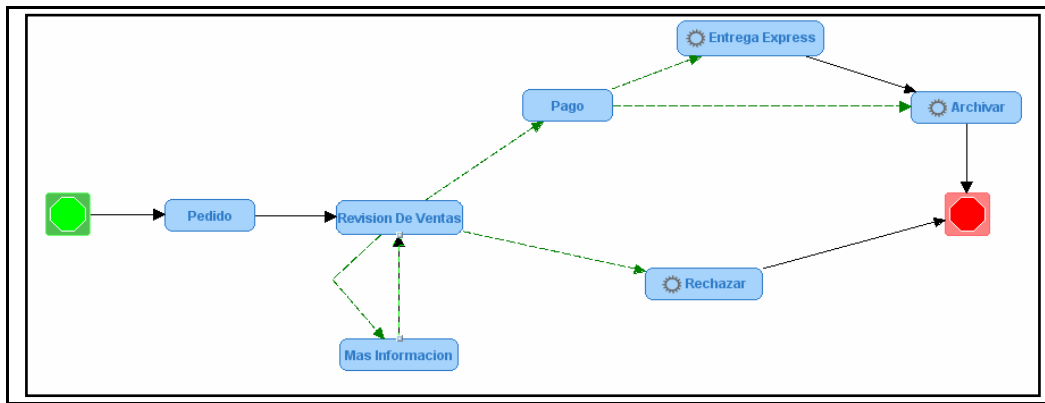


Figura 4 - 3: Proceso ventas *web*, desarrollado con la herramienta Bonita.

Este caso de uso fue extraído de los ejemplos que trae la herramienta Nova Bonita Versión 4.0.

ConsistGPA

Caso de Uso: Solicitud de Bienes y Servicios

Este caso de uso consiste en realizar una solicitud de compras de bienes y servicios. El solicitante llena una solicitud de compra por un conjunto de bienes que pasa a la unidad de suministro la cual verifica la existencia de stock. En caso de tener stock, se realiza la entrega de los bienes requeridos. En caso de no tener stock, se ejecuta un proceso de compras más complejo en el que intervienen varias unidades organizacionales y en el que se crea una Carpeta de Compras que puede ir o no a licitación. Finalizado este ciclo de compras, se procede a la entrega de los bienes solicitados. En la Figura 4-4 se muestra el proceso Solicitud de Bienes y Servicios desarrollados con la herramienta Consist GPA. Se visualizan las *lanes* representando los roles de la organización.

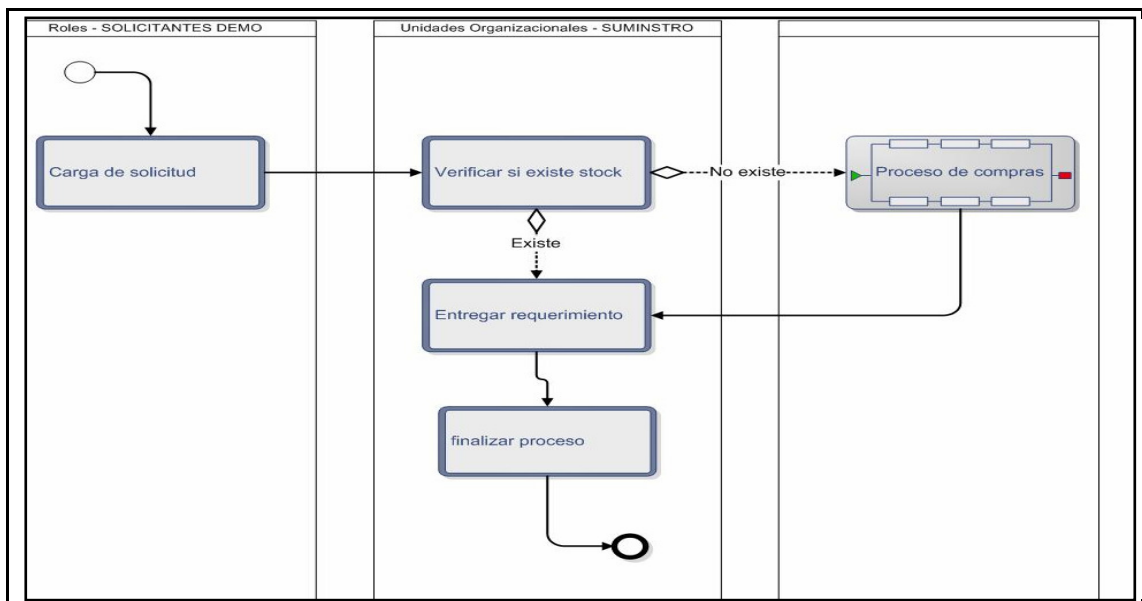


Figura 4 - 4: Proceso de solicitud de bienes y servicios desarrollado con la herramienta Consist GPA

En la Figura 4-4 del proceso solicitud de bienes y servicios, podemos ver que si no existe stock se ejecuta el proceso de compras antes mencionado.

Este caso de uso fue extraído de los ejemplos que trae la herramienta Consist GPA Versión 3.

Oracle Business Process Architect

Caso de Uso: Cotización a Efectivo

Este caso de uso se basa en una compañía que está buscando mejorar su proceso de Cotización a Efectivo. La cotización a efectivo toma un largo tiempo para completarse y el objetivo está centrado en convertir la mayoría de los pedidos de cotización en órdenes de venta.

El proceso de Cotización en Efectivo consiste de dos subprocessos: el proceso Cotización seguido por el proceso Orden. En el proceso Orden, el documento de ventas es generado por el Representante de Ventas y luego es aprobado por el Ejecutivo de Ventas. La cotización es luego enviada al cliente para su aceptación. En el proceso Orden, la orden es generada desde la Cotización, para ello la información de crédito del cliente es verificada y la orden es completada. La Figura 4-5 muestra el proceso principal Cotización a Efectivo que contiene a los dos subprocessos que interactúan.

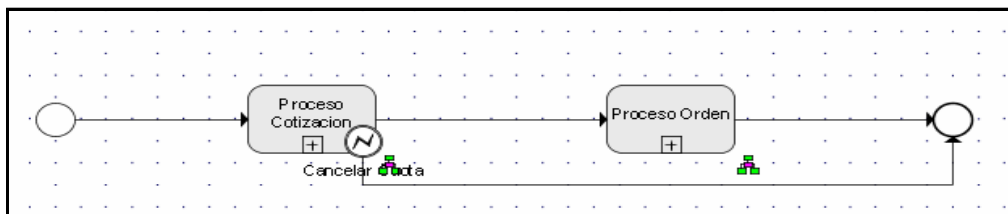


Figura 4 - 5: Proceso cotización a efectivo modelado en la herramienta Oracle BPA.

Participantes del Proceso

Los siguientes participantes están involucrados en el Proceso de Cotización a Efectivo:

- Clientes: los clientes generan pedidos de ventas, ellos aceptan o rechazan las cotizaciones de venta y reconocen el envío de la orden. Las expectativas de los clientes son la rápida generación de las cotizaciones de venta y el cumplimiento de las órdenes. También, ellos necesitan ser notificados dónde sus cotizaciones u órdenes están demoradas.
- Representante de Ventas: es el responsable de generar una cotización en el sistema, realizando un seguimiento con el equipo de ventas para refinar la cotización y asegurar la aprobación por parte del ejecutivo de ventas y luego convertir las cotizaciones en órdenes de venta.
- Ejecutivo de Ventas: es el responsable de aprobar el precio y acordar la estructura de la cotización.
- Supervisor: es el responsable de la aprobación de las órdenes de venta.

- Empleado: es el responsable de empaquetar y entregar la orden. También, es responsable de manejar las excepciones en el proceso Orden.

Servicios

- Servicio Cliente: servicio de negocios que abarca operaciones como: (1) Crear cliente, (2) Obtener detalles del cliente dado el nombre o identificador del mismo, (3) Actualizar cliente y borrar cliente.
- Servicio de calificación crediticia: servicio de negocios que retorna la calificación crediticia dado el número de seguro social del cliente.
- Servicio de Procesamiento de Pedidos: servicio de negocios que abarca operaciones tales como: (1) Crear orden, (2) Actualizar orden, (3) Borrar orden y (4) Cancelar orden.

Interfases de Usuarios de Negocios

- Interfase de Procesamiento de Pedidos: el empleado utiliza esta interfase para manipular las órdenes en el evento de error durante el procesamiento de pedidos.
- Interfase de Cotización: el representante de ventas, así como el ejecutivo de ventas, acceden a esta interfase para crear, refinar y aprobar las cotizaciones.

Datos

- Pedido de Cotización: el mensaje de Petición de Cotización del Cliente pone en marcha el “Proceso Cotización”.
- Cotización: el registro cotización es generado y luego aprobado en el proceso “Proceso Cotización”. Este contiene la estructura del trato, información del precio, así como también, información del cliente y el representante de ventas.
- Orden: es una lista de productos pedidos por el cliente y además contiene información del cliente como ser el nombre y dirección de entrega y facturación. El registro de la orden es generado y completado en el “Proceso Cotización”.
- Cliente: el registro cliente contiene detalles como nombre, identificador, información de la tarjeta de crédito, número de seguro social y dirección de entrega y facturación.

Flujo del Proceso Cotización

1. El documento de cotización de ventas es generado por el representante de ventas después de recibir un mensaje de pedido de cotización del cliente. Este es un paso manual. La cotización contiene detalles tales como nombre del cliente, dirección de entrega, etc.
2. El gerente de ventas aprueba la estructura del trato y el precio de la cotización de ventas. Este es un paso manual. El representante de ventas tiene que rehacer la cotización si esta no es aprobada por el gerente de ventas y reenviarla para su aprobación.

3. Una vez que la cotización ha sido aprobada, el representante de ventas envía la cotización al cliente que luego la aprueba o rechaza.
 - a. Si el cliente la aprueba, un pedido de orden es generado y el “Proceso Orden” es invocado.
 - b. Si el cliente la rechaza, la cotización es eliminada y el cliente es notificado sobre la cancelación.

La Figura 4-6 muestra el diagrama del proceso Cotización que interactúa con el cliente mediante el pasaje de mensajes. El proceso está representado por tres *lanes* que muestran los diferentes roles de la organización.

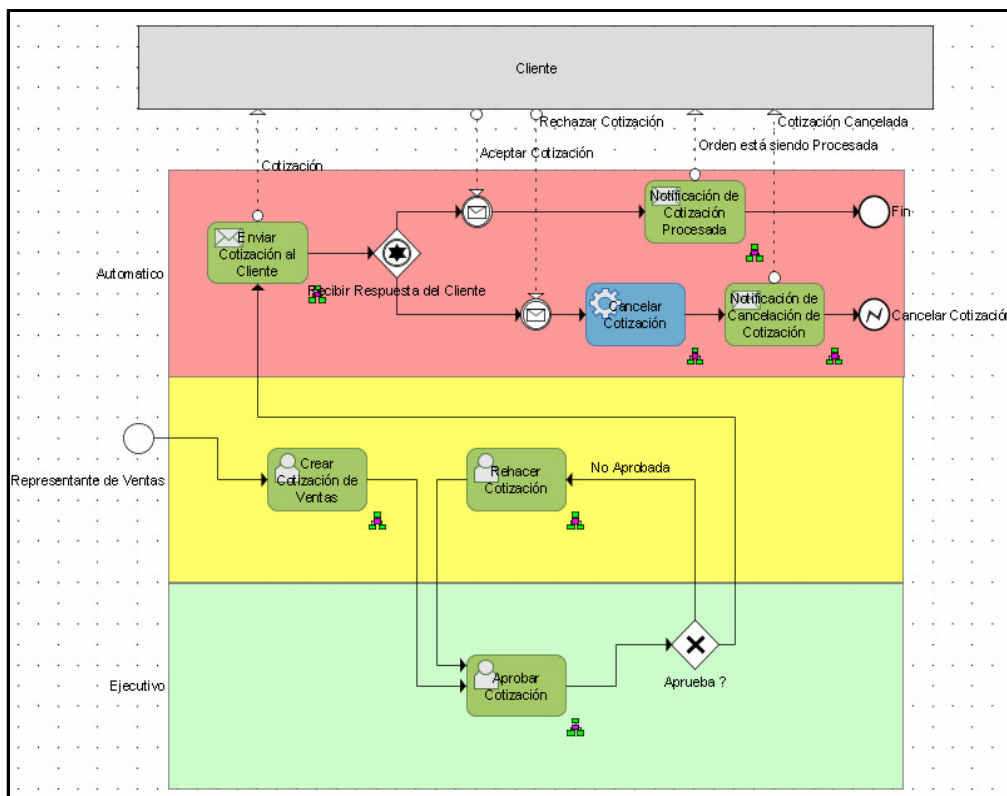


Figura 4 - 6: Proceso Cotización, modelado en la herramienta Oracle BPA.

Flujo del Proceso Orden

1. El pedido de orden desde el proceso Cotización pone en marcha el flujo del "Proceso Orden".
2. Una orden es creada e insertada en el sistema de procesamiento de órdenes. Este es un paso automático
 - a. El estado de la orden es cambiado a “pendiente”.
 - b. La entrada es la “cotización” y la salida es la “orden”.
 - c. El empleado ingresa la orden manualmente en el sistema de procesamiento de órdenes si la invocación del servicio de procesamiento

de órdenes falla. Esto puede pasar si dicho servicio está caído o si el servicio retorna error de “Datos Incorrectos”.

- d. La información del cliente en la cotización es asignada en los campos cliente en la orden.
3. El registro cliente completo es recuperado por la invocación del Servicio Cliente.
 - a. La entrada es el nombre del cliente y la salida es el registro cliente.
 - b. La información del cliente, es manualmente ingresada luego por el empleado si la invocación del Servicio Cliente falla. Esto puede ocurrir si el Servicio Cliente está caído o si el registro cliente no existe en el sistema cliente.
 4. La verificación del crédito es ejecutada luego por la invocación del Servicio de Calificación Crediticia.
 - a. La entrada es el registro del cliente
 - b. La salida contiene la información de la calificación crediticia.
 5. La orden es manualmente aprobada por el supervisor con posterioridad.
 6. Si la orden es aprobada, ésta es enviada a Cumplimiento de Pedidos
 - a. Una vez que la orden es completada, el proceso asigna el estado de la orden como completa, y envía una notificación al cliente vía email con la información de la orden de compra.
 - b. Si la orden no es aprobada, se cancela y se envía una notificación al cliente para la cancelación de la misma.

En la Figura 4-7 podemos visualizar el proceso orden que es el encargado de efectivizar el pedido.

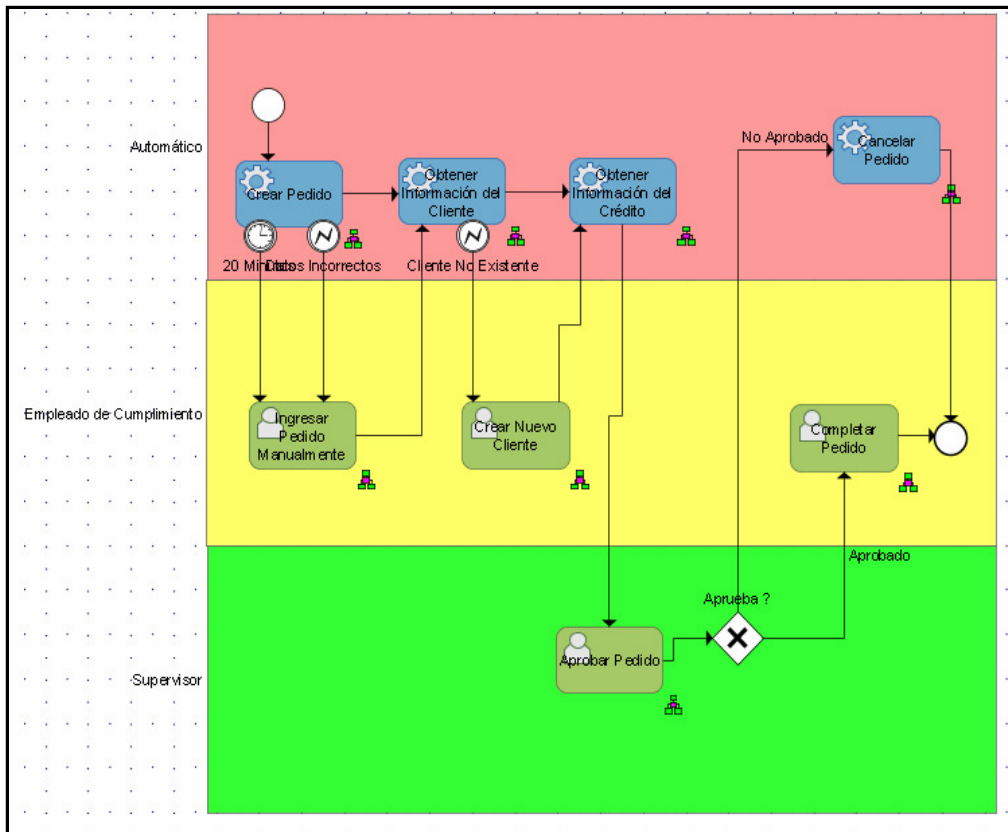


Figura 4 - 7: Proceso Orden, modelado en la herramienta Oracle BPA.

Este caso de uso fue desarrollado a partir de la documentación *Oracle Business Process Architect Quick Start Guide Versión 10.1.3.4*.

Oracle SOA Suite

Caso de Uso: Flujo Crediticio

Este caso de uso consiste de un proceso asincrónico, en el cual se realiza la invocación a un servicio simple. El servicio al que se invoca se llama “Servicio Sincrónico de Tasación de Crédito”. Al ser este proceso asincrónico, se definen dos *portType* en el proceso BPEL, cada uno con una operación de una vía: una para inicializar el proceso asincrónico y la otra para que el servicio invocado pueda entregar la respuesta asincrónica.

La Figura 4-8 muestra la vista BPEL del proceso *credit flow* en la IDE jDeveloper. Se pueden apreciar la comunicación entre el proceso y sus *partnerLinks* Cliente y ServicioDeClasificacionCrediticia.

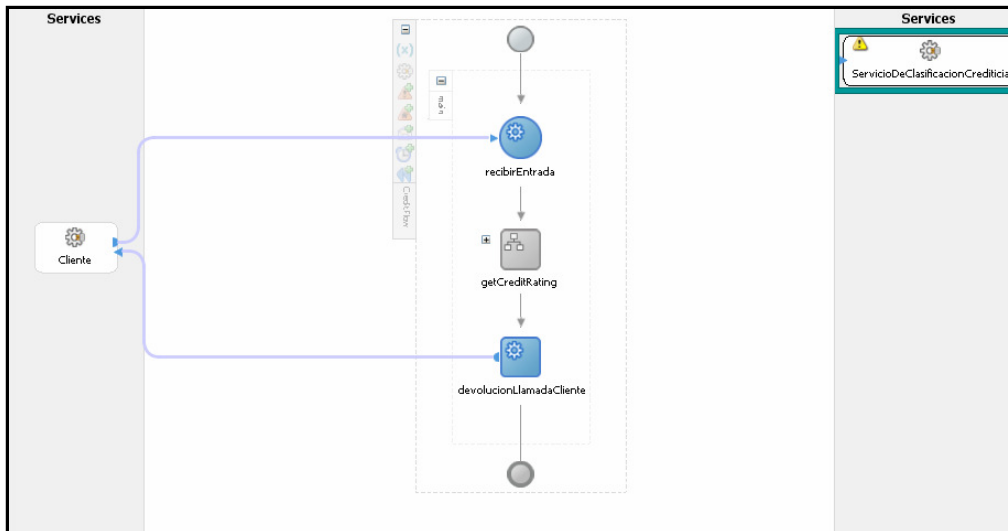


Figura 4 - 8: Proceso *crédit flow*, desarrollado desde la IDE jDeveloper y ejecutado luego en la herramienta Oracle SOA Suite.

Este caso de uso fue extraído de los ejemplos que trae la herramienta *Oracle SOA Suite* 10.1.3.1.

Caso de Uso: *Loan Flow Plus*

Este caso de uso consiste de un proceso asincrónico complejo, en el cual se muestran entre otras cuestiones: manejo de fallas en interacciones con servicios asincrónicos, tareas de usuario/manuales, integración de código Java/JEE, etc. Estos requerimientos son claves en los flujos de negocios complejos.

Loan Flow Plus es un proceso con un tiempo alto de ejecución que toma un documento de préstamo como entrada y luego asincrónicamente devuelve una oferta de préstamo seleccionada y aprobada. Cuando comienza la ejecución del proceso, asigna un número de seguridad social para el cliente desde un servicio y luego realiza una llamada asincrónica al servicio CreditRating obteniendo una tasación (clasificación crediticia) para el cliente. El proceso está codificado para manejar la falla CreditNegativo que puede ser lanzada si el cliente tiene una tasación de crédito negativo (por estar en quiebra por ejemplo). Si esa excepción se lanza, el proceso encola una tarea para un empleado que manualmente manejara la excepción y luego esperará a que la tarea se complete. El empleado puede usar una aplicación UI (*user interface*) para ver sus tareas y especificar una tasación de crédito o elegir cancelar cada aplicación.

Una vez que se obtiene una tasación de crédito, a través de cualquier tarea (manual o automática), el proceso comienza un procesamiento automático que envía el documento de préstamo a dos procesadores de préstamo, quienes pueden tomar una cantidad de tiempo arbitrario en devolver la oferta. Como los servicios pueden tener un tiempo alto de ejecución, fueron implementados como servicios asincrónicos y el proceso BPEL los invoca en paralelo. El proceso, espera hasta que sean entregadas las dos ofertas de crédito y luego utiliza una lógica muy simple para elegir la mejor oferta (una sentencia condicional IF que obtiene la tasa de interés más baja). Finalmente, otra tarea de usuario permite al cliente revisar y aprobar la selección de la oferta de préstamo. Una vez que

Capítulo 5 - JBoss Enterprise SOA Platform

JBoss enterprise SOA platform incluye una arquitectura orientada a servicios (SOA) y está compuesta por las siguientes componentes *open source*⁹: *JBoss enterprise service bus* (ESB), *JBoss jBPM*, *JBoss rules* y *JBoss enterprise application platform* para integrar aplicaciones, servicios, transacciones y componentes de negocios, dentro de procesos de negocios automatizados [11].

Características y Componentes

- ***JBoss Enterprise Service Bus***: este componente actúa como intermediario entre las aplicaciones empresariales, servicios de negocios, componentes de negocios y *middleware*¹⁰, para integrar y posibilitar la automatización de procesos de negocios. Soporta varios productos de mensajería para transporte, componentes SOA y transformación de datos para una comunicación confiable. También, provee un registro para el descubrimiento e integración de servicios. Está diseñado para gestionar *software* de forma simple o compleja. Por su flexibilidad y su arquitectura abierta, permite a otros productos incorporar suplementos para extender la plataforma.
- ***JBoss Rules***: es un motor de reglas rápido y eficiente que facilita al desarrollador, analista de negocios o auditor la visualización y modificación de sus reglas de negocios en la infraestructura de las aplicaciones IT, para implementar políticas de negocios evolutivas. También, soporta una gran variedad de lenguajes y tablas de decisión, facilitando una rápida modificación de las políticas de negocios para responder a los constantes cambios en las mismas.
- ***JBoss jBPM***: posibilita a las empresas la creación y automatización de procesos de negocios para coordinar personas, aplicaciones y servicios. Brinda automatización de procesos a un amplio conjunto de problemas de negocios, que van desde flujos embebidos a orquestación de procesos de negocios empresariales.
- ***JBoss Enterprise Application Platform***: es una plataforma para aplicaciones Java que integra un servidor de aplicaciones con la siguiente generación de *frameworks* de aplicaciones en un solo producto. Permite resolver la automatización de negocios, la integración de aplicaciones empresariales y la orquestación de procesos [11] [12].

⁹ Código abierto (en inglés open source) es el término con el que se conoce al software distribuido y desarrollado libremente. Extraído de: http://es.wikipedia.org/wiki/Codigo_abierto

¹⁰ El middleware es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red). El middleware nos abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas. Dependiendo del problema a resolver y de las funciones necesarias, serán útiles diferentes tipo de servicios de middleware. Extraído de: <http://es.wikipedia.org/wiki/Middleware>

jBPM

JBoss JBPM es un *framework* flexible y extensible para lenguajes de procesos. JPDL es un lenguaje de procesos que está construido por sobre el *framework*. Este lenguaje se utiliza para expresar procesos de negocios de forma gráfica en términos de tareas, estados de espera para comunicación asincrónica, contadores de tiempo y demás operaciones. También, el *framework* permite la creación del proceso mediante XML. Mientras otras soluciones se enfocan en un lenguaje particular, JBPM soporta múltiples lenguajes de procesos. En la Figura 5-1 se muestran los componentes más importantes de la arquitectura del *framework* jBPM.



Figura 5 - 1: Arquitectura del componente jBPM provisto por la plataforma *JBoss enterprise SOA*. Extraída de: http://www.jboss.org/jbossjbpm/jbpm_overview/

JPDL

JPDL es el lenguaje de proceso principal de JBPM. A la vez que es directamente ejecutable, provee capacidades de modelado. Este aprovecha la PVM (Máquina Virtual de Procesos) y el componente identidad, administración de tareas y componentes *Enterprise*.

BPEL

Es el estándar para la orquestación de servicios. JBPM provee una implementación de este estándar. Se utiliza para escribir *scripts* que crean *web services*.

GPD

El Diseñador Gráfico de Procesos (GPD) es un conjunto de plugins¹¹ construidos para Eclipse. Los propósitos de GPD son múltiples: creación y edición gráfica de procesos JPDL, construcción e inclusión de operaciones personalizadas, dando soporte para la creación de un nuevo lenguaje gráfico.

La Máquina Virtual de Procesos (PVM)

Las partes comunes como la ejecución del *workflow* y la administración de estados son incluidas en la Máquina Virtual de Procesos (PVM). Los lenguajes de procesos extienden la PVM para que puedan correr nativamente como una sola tecnología.

Dicha PVM, es una librería Java para construir y ejecutar los *workflows*.

¹¹ Un complemento (o plugin en inglés) es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. Extraído de: <http://es.wikipedia.org/wiki/Plugin>

Los lenguajes de procesos son un conjunto de tipos de actividades construidos usando las características provistas por la PVM.

Pageflow

Mientras no esté disponible como un sub-proyecto separado, *pageflow* es un lenguaje de procesos que es soportado por jBPM. Ofrece la posibilidad de usar jBPM *pageflow* como el mecanismo de navegación entre páginas *web*. La creación de estos *pageflow* es soportada por el GPD.

Identity

La administración de usuarios, grupos y permisos es comúnmente conocida como Administración de Identidad. jBPM actualmente incluye un componente opcional de Identidad que puede ser reemplazado fácilmente por el manejo de identidad que posee la compañía [13].

Licencia

JBoss Soa Platform tiene una licencia LGPL (*Lesser General Public License*), la cual lo hace libre para descargarlo, usarlo, embeberlo, y distribuirlo.

Suscripción

Se puede suscribir a *JBoss enterprise middleware*, con las opciones de 4 CPU o 32 CPU, esta suscripción incluye:

- **Acceso al Producto:** código fuente y binario, documentación.
- **Actualizaciones:** actualizaciones regulares que proveen mejoras generales, nuevas características, etc.
- **Revisiones:** acceso a los últimos defectos corregidos y erratas de seguridad.
- **Mejoras:** derechos a las nuevas versiones de *JBoss enterprise middleware* liberadas durante el tiempo de vida de la suscripción.
- **Flexibilidad:** la suscripción no es específica a una versión particular, el acceso y soporte para versiones previas también es soportado.
- **JBoss Customer support portal:** conexión directa con el personal de soporte se JBoss.
- **Soporte:** soporte múltiple de servicios proveyendo cobertura ilimitada para incidentes con una hora de respuesta [11].

Para más información sobre las suscripciones, visitar <http://www.redhat.com/jboss/benefits/>.

Requerimientos Mínimos del Sistema

Requerimientos Mínimos de Hardware

- 768 Mb de RAM
- 800 Mb de espacio en el disco rígido para la versión full (*platform*) o 220 Mb de espacio en disco rígido para la versión *standalone* [14].

Sistemas Operativos Soportados

Como *JBoss Soa platform* es una aplicación 100% Java, es ínter operable con la mayoría de los sistemas operativos que son capaces de correr una máquina virtual Java (JVM), incluyendo Windows, Linux, Unix y otros [11].

Instalación y Configuración

Instalación de la Suite

Lo primero que se debe hacer es instalar una máquina virtual Java, si es que no se tiene ninguna instalada, cualquier versión de la JDK 1.5.x alcanza.

Luego se debe exportar al camino y cambiar la variable de entorno JAVA_HOME, con el directorio de instalación de la máquina virtual.

Para poder descargar el producto se debe crear un usuario en JBoss Red Hat desde la página

https://www.redhat.com/apps/webform.html?event_type=simple_form&eid=1302

Cuando se realiza la suscripción, se recibe un mail con la URL desde donde se puede descargar *JBoss enterprise SOA platform*.

Existen dos versiones para el mismo producto: la *standalone* que es la versión liviana con la funcionalidad principal del ESB y la *full* que incluye un servidor de aplicaciones Java completo y un *framework* de aplicaciones [14].

Con solo descomprimir el zip queda instalado el producto. Una vez arrancado el servidor (ejecutando *run* en directorio bin del servidor), podemos comprobar que la herramienta está funcionando accediendo a <http://localhost:8080/> desde el navegador *web*.

Para acceder a cada uno de los componentes se necesita un usuario y *password*. Se deben modificar los archivos *soa-user.properties* para administrar usuarios y *soa-roles.properties* para administrar los roles de cada usuario.

Instalación del Ambiente de Desarrollo

Nuestro ambiente de desarrollo consistirá de la herramienta IDE Eclipse, con el plugin JBPM *Graphical Process Designer* (descargable desde http://www.eclipse-plugins.info/eclipse/plugin_details.jsp?id=1028), el cual nos permite crear los procesos de negocio de forma gráfica.

Bajamos un Eclipse de <http://www.eclipse.org/downloads/>. Para esta versión de la suite necesitamos la versión 3.3, con la edición estándar es suficiente.

Debemos bajar el *plugin JBoss jBPM JPDL designer - Versión 3.1.0.SP1*, ubicar la carpeta *designer* dentro de la carpeta *jbpm-jpdl* de la plataforma.

Luego de descomprimir el Eclipse, hay que trasladar la carpeta eclipse con el entorno, pisando la carpeta eclipse dentro del directorio *designer*. Esto realiza un combinado reemplazando e instalando el *plugin* dentro del entorno eclipse. Una vez realizado este paso, queda todo funcionando. Para comprobarlo solo debemos hacer clic con el botón derecho en el *package explorer* del Eclipse, *new->other* y ahí vamos a poder crear un *process definition* o un *process project* dentro de la carpeta JBoss jBPM.

Persistencia de Datos

Bases de Datos Soportadas

JBoss Enterprise SOA platform utiliza JDBC para realizar el acceso a la base de datos, con lo cual, es ínter operable con cualquier base de datos que sea compatible con JDBC, entre las que se destacan IBM DB2, MySQL, Oracle *database*, Microsoft SQL Server y otras [11].

Configuración Inicial

Por defecto la configuración de JBoss Enterprise SOA platform trabaja con una Base de Datos preconfigurada llamada HyperSonic. Debido a que esta base de datos virtual (reside en memoria RAM), es útil para desarrollo y *testing*, pero no lo es para producción, se puede configurar el *framework* para que utilice otros motores de Base de Datos [15].

Configuración Bajo MySql 5.0

En el directorio DB dentro de la carpeta donde reside el *framework*, podemos encontrar un conjunto de *scripts* para la creación de la base de datos en varios motores de base de datos que soporta el *framework* (entre ellos MySQL, SQL *server*, Oracle, PostgreSQL, etc.). Este *script* crea la misma estructura de base de datos que la HyperSonic.

Para cambiar la base de datos HyperSonic por alguno de estos motores, se debe realizar lo siguiente:

- Instalar el motor de base de datos y correr el servidor.

- Crear la Base de Datos con un nombre cualquiera, por ejemplo JBPM, a través de la herramienta de administración del motor de Base de Datos.
- Correr el *script* que corresponde en la base de datos previamente creada. Este *script* contiene sentencias para eliminar las restricciones y tablas de la misma por si antes fue creada. Si es la primera vez que se crea esta base de datos, se deben eliminar esas sentencias del *script*
- Luego de haber ejecutado el *script*, se necesitan cargar unos registros en la base, que corresponden a usuarios del *framework*. Esto se realiza ejecutando las siguientes sentencias SQL:

```

insert into JBPM_ID_USER (ID_, CLASS_, NAME_, EMAIL_, PASSWORD_)
    values ('1', 'U', 'user', 'sample.user@sample.domain', 'user');

insert into JBPM_ID_USER (ID_, CLASS_, NAME_, EMAIL_, PASSWORD_)
    values ('2', 'U', 'manager', 'sample.manager@sample.domain', 'manager');

insert into JBPM_ID_USER (ID_, CLASS_, NAME_, EMAIL_, PASSWORD_)
    values ('3', 'U', 'shipper', 'sample.shipper@sample.domain', 'shipper');

insert into JBPM_ID_USER (ID_, CLASS_, NAME_, EMAIL_, PASSWORD_)
    values ('4', 'U', 'admin', 'sample.admin@sample.domain', 'admin');

```

Verificar que el campo ID_ no sea autoincremental en la configuración de la tabla. Si es así, hay que eliminar ese campo en las sentencias de inserción anteriores.

El siguiente paso consiste en actualizar la configuración del servidor JBoss para que trabaje con la nueva base de datos creada. Los pasos para realizar esta configuración son los siguientes:

- Crear un nuevo *datasource* para el JBoss jBPM *server*. En el siguiente camino ``${JBPM_SDK_HOME}/jbpm-server/server/jbpm/deploy`. Debemos reemplazar el contenido del archivo `jbpm-ds.xml` por el siguiente código:

```

<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>JbpmDS</jndi-name>
    <connection-url>
      jdbc:mysql://localhost:3306/jbpm
    </connection-url>
    <driver-class>com.mysql.jdbc.Driver</driver-class>
    <user-name>Database_User</user-name>
    <password>User_Pass</password>
    <metadata>
      <type-mapping>MySQL</type-mapping>
    </metadata>
  </local-tx-datasource>
</datasources>

```

Este código sólo sirve para el motor MySQL, en caso de utilizar otro motor hay que cambiar los parámetros de este XML por los correspondientes al nuevo motor. Inicialmente este archivo contiene la configuración necesaria para que funcione con la base de datos HyperSonic.

- El siguiente paso es instalar el controlador del dispositivo de JDBC correspondiente al motor elegido. En nuestro caso es el controlador del dispositivo de MySQL 5.0 que se puede descargar desde la siguiente URL:
<http://dev.mysql.com/downloads/connector/j/5.0.html>
- Una vez descargado el controlador del dispositivo, se lo debe instalar y luego copiar los jar correspondientes (en nuestro caso *mysql-connector-java-5.0.8-bin.jar*) en el siguiente camino: $\${JBPM_SDK_HOME}/jbpm-server/server/jbpm/lib$.
- Luego, se debe realizar la configuración de Hibernate. Hibernate permite a la aplicación *web* conectarse a cualquier base de datos usando solo el *datasource* escogido, además de poder serializar la base de datos a objetos POJO [15]. Para configurar Hibernate se deben realizar los siguientes pasos:
 1. Editar el archivo *hibernate.cfg.xml* en el paquete de la aplicación: $\sim/jbpm-jpdl-3.2.1/server/server/jbpm/deploy/jbpm-console.war$
 2. La conexión al *datasource* debe ser especificada en la propiedad *hibernate.connection.datasource* y el dialecto de la base de datos debe ser modificado en la propiedad *hibernate.dialect*. En el caso de MySQL 5.0, el fragmento de código quedaría así:

```
<hibernate-configuration>
<session-factory>

    <!-- hibernate dialect -->
    <property name="hibernate.dialect">
        org.hibernate.dialect.MySQLInnoDBDialect
    </property>

    <!-- JDBC connection properties (begin) ===
    <property
        name="hibernate.connection.driver_class">org.hsqldb.jdbcDriver</property>
    >
    <property
        name="hibernate.connection.url">jdbc:hsqldb:mem:jbpm</property>
    <property name="hibernate.connection.username">sa</property>
    <property name="hibernate.connection.password"></property>
    ===== JDBC connection properties (end) -->

    <property name="hibernate.cache.provider_class">
        org.hibernate.cache.HashtableCacheProvider
    </property>

    <!-- DataSource properties (begin) -->
    <property name="hibernate.connection.datasource">
        java:./JbpmDS
    </property>
    <!-- DataSource properties (end) -->
```

Soporte de Tecnología Java

Para realizar el diseño e implementación de los procesos de negocios, se utiliza el *plugin* de Eclipse JBPM Graphical Process Designer (http://www.eclipse-plugins.info/eclipse/plugin_details.jsp?id=1028) que permite crear proyectos jBPM con la estructura necesaria para su desarrollo en Java. Además trae un conjunto de herramientas visuales para el desarrollo de los diagramas que representan a los procesos de negocio.

Para realizar la implementación de una actividad automática se utilizan clases Java, que deben implementar una interfase específica, la cual se encuentra en la librería de jBPM.

Análisis de la Suite y ciclo de vida BPM soportado

Componentes de la Herramienta

Eclipse con el plugin JBPM Graphical Process Designer

El ambiente de modelado y desarrollo está compuesto por el IDE Eclipse y el *plugin* JBPM *Graphical Process Designer*. Este conjunto de herramientas nos permiten realizar el modelado, desarrollo y despliegado de los procesos de negocios en JPDL.

JBoss SOA platform

La plataforma consta de dos componentes principales:

- **jBPM – JPDL:** este componente consta de un conjunto de librerías que dan soporte a la creación de procesos de negocios mediante JPDL. Estas librerías se pueden agregar a los proyectos Java para implementar procesos de negocios.
- **JBoss application server:** es un servidor de aplicaciones JEE que trae soporte para la ejecución y administración de procesos de negocios JPDL.

Además, trae incorporado un *enterprise service bus*, mediante el que es posible incorporar y gestionar servicios.

Modelado y Desarrollo

La etapa de modelado y desarrollo se realiza en el ambiente Eclipse utilizando el *plugin* JBPM *Graphical Process Designer*. Este componente nos permite crear los proyectos jBPM; dentro de estos proyectos podemos agregar los diferentes procesos de negocios BPM.

El diseñador consta de dos vistas:

- Una vista gráfica con una paleta de componentes para realizar los diagramas de procesos. Dicha paleta trae los componentes básicos con posibilidad de agregar nuevos elementos. Los componentes visuales antes mencionados son propios del lenguaje JPDL. En la Figura 5-2 tenemos un ejemplo de la vista gráfica de un proceso de negocios.

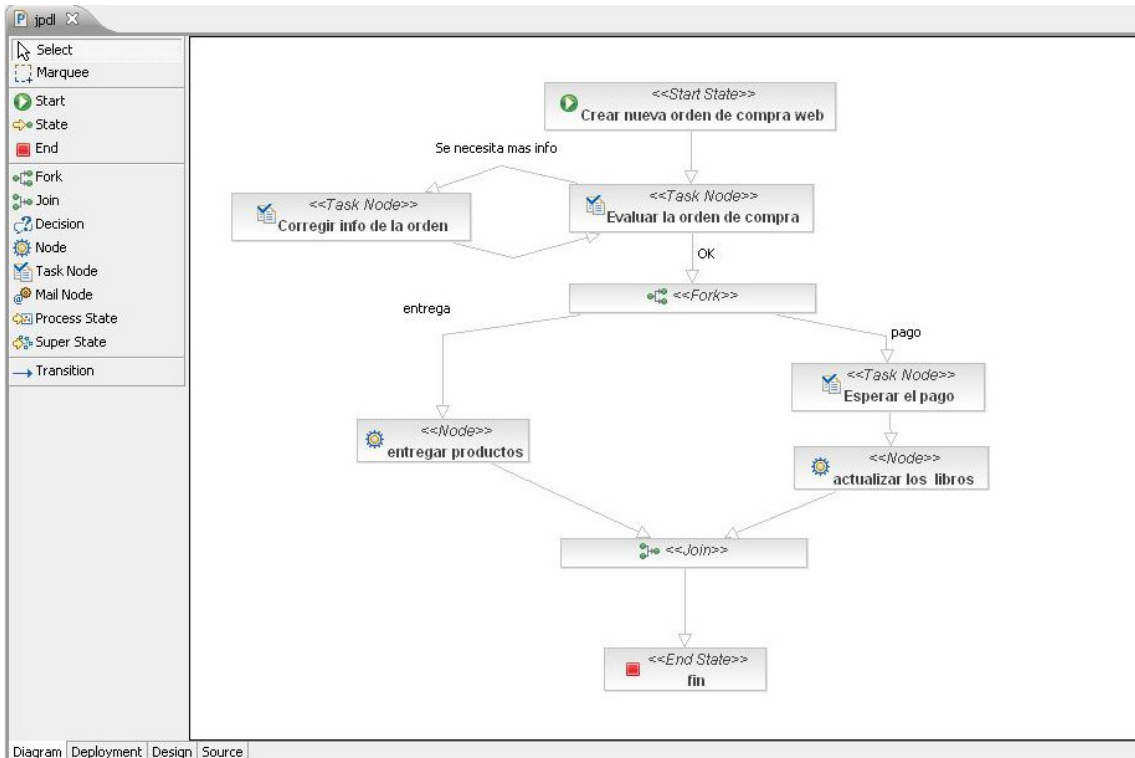


Figura 5 - 2: Vista gráfica del plugin JBPM Graphical Process Designer

- Una vista XML donde podemos ver el código fuente que se genera en el lenguaje JPDL. En la Figura 5-3 podemos visualizar la vista XML de un proceso.

```

xmins="urn:jbpm.org:jpdl-3.2"

<!-- SWIMLANES (= process roles) -->
<swimlane name="comprador" />
<swimlane name="vendedor">
  <assignment actor-id="manager" />
</swimlane>

<swimlane name="contador">
  <assignment actor-id="shipper" />
</swimlane>

<swimlane name="empresa de entrega">
  <assignment actor-id="admin" />
</swimlane>

<!-- NODES -->

<start-state name="Crear nueva orden de compra web">
  <task swimlane="buyer" />
  <transition to="Evaluar la orden de compra" />
</start-state>

<task-node name="Evaluar la orden de compra">
  <task swimlane="salesman">
    <timer due-date="20 seconds" repeat="10 seconds">
      <action class="org.jbpm.websale.RemindActor">
        <swimlaneName>salesman</swimlaneName>
      </action>
    </timer>
  </task>

```

Figura 5 - 3: Vista XML del plugin JBPM Graphical Process Designer

Ambas vistas son bidireccionales, esto quiere decir que al modificar una de ellas, se actualiza la otra automáticamente.

Cuando se crea un proyecto jBPM, automáticamente se crea toda la estructura de directorios con las librerías necesarias. La Figura 5-4 muestra la estructura de directorios y archivos generados con el *plugin JBPM Graphical Process Designer*.

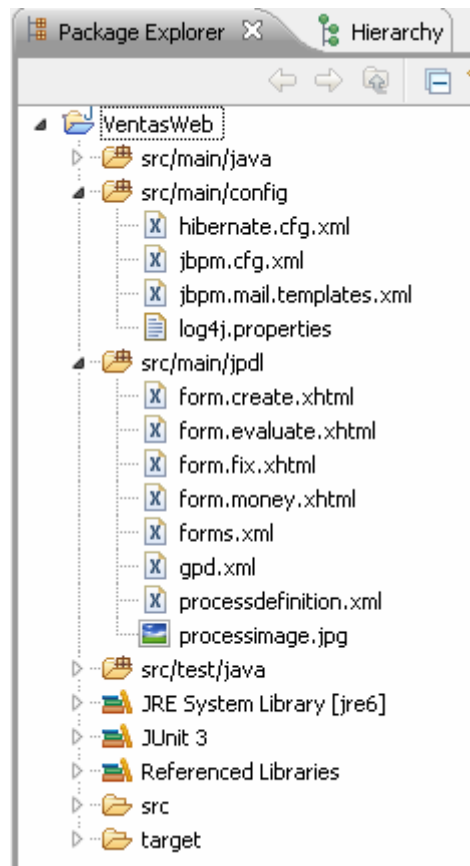


Figura 5 - 4: Estructura de directorios generada.

Los modelos de procesos no adhieren al estándar de notación BPMN, y como mencionamos anteriormente, utilizan un lenguaje propietario llamado JPDL. En la Figura 5-5 podemos ver resaltados los nodos *fork* y *join* (elementos propios de JPDL) que no forman parte del estándar de notación BPMN.

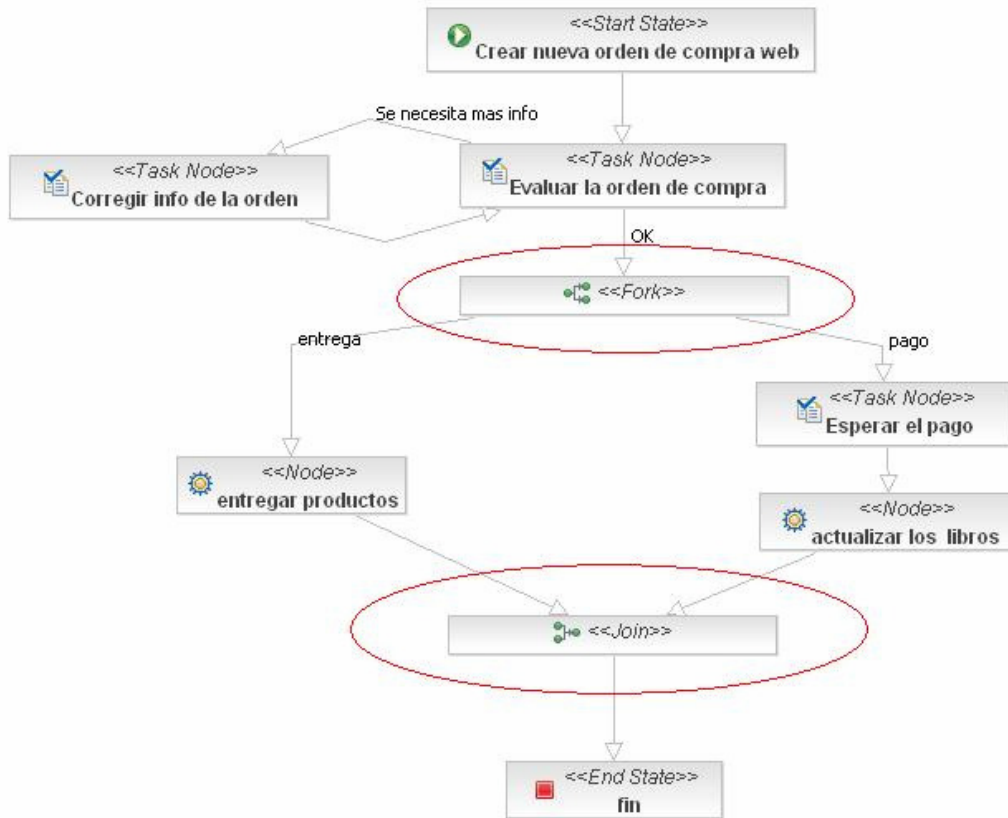


Figura 5 - 5: Algunas diferencias entre JPDL y BPMN.

Entre los componentes disponibles en JPDL tenemos Tareas Manuales (*task node*) y Automáticas (*node*), Nodos de Estado (*state*), Transiciones (*transition*) para conectar componentes, Nodos Mail (*mail node*) para realizar envío de mails, ESB *service* para agregar servicios que están disponibles en el ESB, Nodos de Decisión (*decision node*) y Nodos de Bifurcación (*fork* y *join*) entre otros. Hay que destacar que si bien JPDL es un lenguaje propietario, los componentes no tienen grandes diferencias con los del estándar BPMN y el código JPDL es muy sencillo de comprender y modificar, siendo entendible por cualquier persona.

El diseñador de procesos es muy intuitivo y simple de utilizar. Cada componente agregado al diagrama, tiene un conjunto de propiedades que lo describen, siendo fácil la modificación de las mismas. Para el caso de tareas manuales, el diseñador nos da la posibilidad de crear formularios simples que podrán ser utilizados por el rol encargado de la tarea, para completar las variables asociadas al mismo cuando el proceso está en ejecución. Además, permiten controlar el flujo de ejecución de los procesos con botones que indican que transición se elige dentro del flujo. También, es posible agregar de forma sencilla validaciones a los campos del formulario. Estos formularios tienen una extensión xhtml y se pueden generar fácilmente desde el mismo IDE. Para realizar el mapeo de actividades manuales con los formularios existentes, existe un archivo `forms.xml` en el cual se especifica esta información.

En las tareas automáticas, se especifica que clase Java será la encargada de ejecutar la misma. Estas clases implementan una interfase propia de la librería JPDL, la interfase tiene un método principal que se debe completar para dar implementación a la tarea.

Despliegue

Una vez finalizado el diseño y desarrollo de los procesos, se realiza la etapa de despliegado. Esta etapa se puede realizar de dos maneras.

- En la IDE Eclipse, en el archivo donde tenemos el diseño del proceso (processdefinition.xml), tenemos disponible la solapa *deployment*. Desde esta vista, podemos configurar y testear la conexión al servidor. Luego, se deben seleccionar los archivos, directorios, clases y recursos que componen el proceso. Presionando el botón “*Deploy Process Archive*”, podemos desplegar el proceso directamente al servidor JBoss. También, tenemos la posibilidad de grabar en alguna ubicación un archivo que contendrá toda la información del proceso sin realizar el despliegado. En la Figura 5-6 se ve el formulario que nos permite realizar el despliegue en el servidor de los procesos diseñados con el JBPM *Graphical Process Designer*.

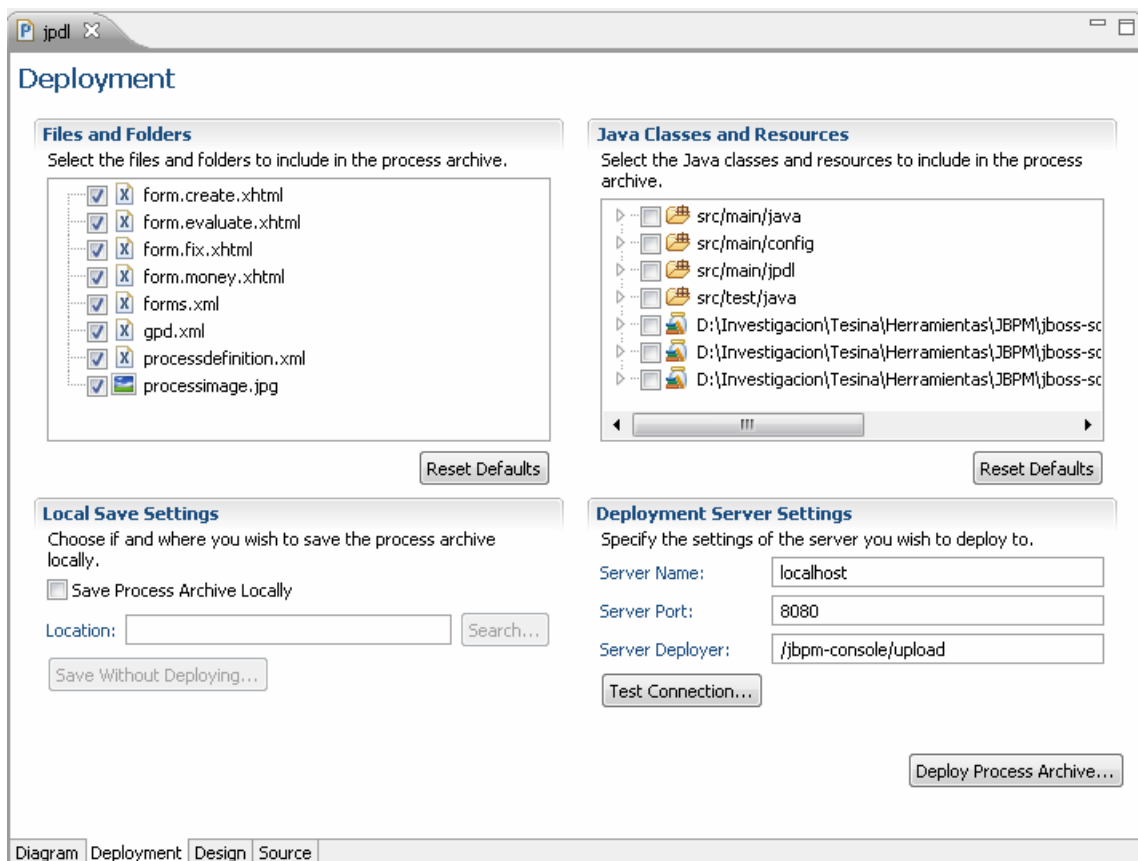


Figura 5 - 6: Desplegado desde la IDE Eclipse.

- La otra forma de desplegar el proceso es desde la consola de administración jBPM que trae la suite. Vamos a la solapa procesos y elegimos la opción “*deploy new process*”, luego seleccionamos la ubicación del archivo generado previamente desde la solapa *deployment* en el Eclipse. En la Figura 5-7 vemos el formulario de despliegue de un proceso jBPM en el servidor JBoss.

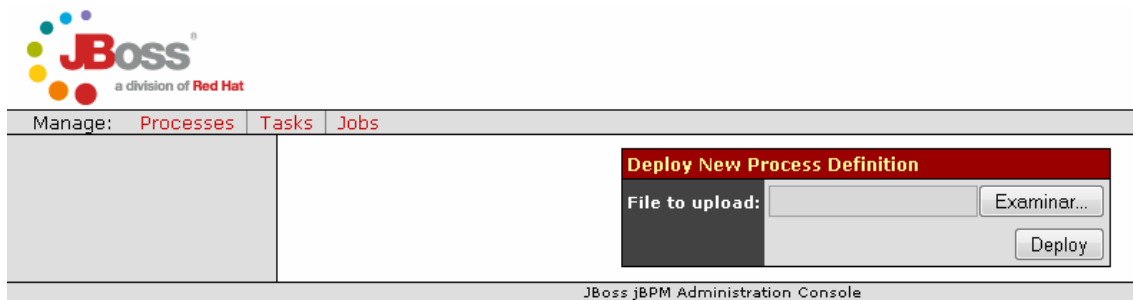


Figura 5 - 7: Despliegado desde la consola de administración.

Ejecución y Monitoreo

Esta etapa se realiza desde la *jBPM console*, accesible desde el navegador *web*. Desde la solapa de Procesos, se listan los procesos desplegados actualmente en el servidor. Además, como se explicó en la etapa anterior, podemos desplegar nuevos procesos y eliminar los existentes. También tenemos un buscador de procesos con filtros que nos permite ubicar fácilmente un proceso en particular.

Tenemos la posibilidad de acceder a información detallada de un proceso, desde el botón Examinar. Esta vista nos brinda información del proceso como por ejemplo: nombre y versión, instancias creadas, estado de las mismas, imagen del flujo del proceso, acceso a las tareas en ejecución junto a otras opciones de administración de las instancias (crear, eliminar). Notar que el manejo de versiones lo realiza automáticamente el servidor.

En la solapa Tareas se pueden ver todas las tareas de todos los procesos en el servidor. Se incluye un buscador de tareas con filtros para buscar una tarea en particular. Por cada tarea de la lista podemos ver el estado en el que se encuentran, cambiar el estado de las mismas, suspenderlas o acceder a información más detallada. Dentro de cada tarea, podemos ver el formulario asociado (si dispone de uno), el cual se puede completar con la información adecuada para continuar con el flujo del proceso. También podemos ver el listado de las variables responsables de la tarea, reasignar la tarea a otro rol, visualizar fecha y hora de inicio y creación junto a otros datos de interés.

En la solapa *jobs*, podemos ver un *log* con los errores que se van produciendo en tareas que están en ejecución. Este *log* es muy útil para detectar problemas en los procesos desplegados, ya que muestra información detallada de las excepciones que fueron lanzadas cuando ocurre algún error.

Conclusión General de la Suite

El conjunto de herramientas que componen la suite permiten cubrir las etapas básicas en el desarrollo de procesos de negocios: modelado, implementación, ejecución y monitoreo.

Estas herramientas no dan soporte a etapas como: Simulación y Optimización que son muy importantes a la hora de realizar un desarrollo BPM a nivel empresarial. Estas etapas son claves para detectar en forma temprana fallas y errores, permitiendo depurar y optimizar los procesos, reduciendo costos y tiempo de desarrollo.

Si bien afirmamos que la herramienta soporta modelado de procesos, y a su vez son fáciles de comprender, sólo existe un diagrama del proceso en todo el ciclo de vida. Sus componentes y notación están muy ligados a la implementación de los procesos, haciendo más difícil su comprensión para la gente de negocios.

La existencia de una única herramienta para el modelado y la implementación, permite que la gente de negocios y la gente de IT tengan una mayor comunicación. Igualmente creemos que es necesario perfiles intermedios que hagan de mediadores de esta comunicación.

En cuanto a los requerimientos de *hardware*, todas las herramientas analizadas, no requieren de un *hardware* potente y pueden correr en una PC hogareña.

En cuanto a la instalación, es de muy bajo nivel y requiere una gran cantidad de pasos ya que la *Jboss SOA platform* no trae incorporadas herramientas de desarrollo, haciendo necesario la configuración e instalación de *plugins* para el ambiente Eclipse. Existe una herramienta *Jboss developer studio*, bajo licencia GNU *public license* v2 (GPL) que trae todas estas funcionalidades integradas.

Con respecto a la documentación asociada a la herramienta es abundante y sencilla de comprender. Además, existe una comunidad con varios foros que brindan soporte para resolver un amplio conjunto de problemas.

Las herramientas analizadas no traen soporte para el lenguaje de ejecución BPEL. La empresa JBoss y su socio *Active endpoint* tienen un conjunto de herramientas por separado para dar sustento a este lenguaje. Este conjunto de herramientas está compuesto por:

- **Eclipse BPEL designer:** *plugin* para eclipse que permite diseñar procesos BPEL, disponible desde <http://www.eclipse.org/bpel/>
- **JBoss AS 4.2.2.GA:** dentro de este servidor de aplicaciones tendrán que ser instalados de forma separada los componentes ESB y JBossWS.
- **Active BPEL engine:** esta herramienta es un motor de BPEL *open source* desarrollado por la empresa *Active endpoints* que debe ser descargado desde la página de la empresa (<http://www.activevos.com/community-open-source-engine-download.php#final50>) y luego instalado dentro de una instancia de Tomcat.

Es claro que JBoss no trae soporte directo a BPEL, ni tampoco un paquete de herramientas unificado para lograr esto. Igualmente, la comunidad JBoss en un futuro piensa incorporar el soporte para este lenguaje dentro de la *JBoss SOA platform* [16].

A nuestro criterio, este conjunto de herramientas es adecuado para pequeños y medianos desarrollos de *software*, ya que carece de componentes para cubrir etapas importantes en el desarrollo de procesos, lo cual es de vital importancia en grandes desarrollo cuando el tamaño y la cantidad de procesos crece. Igualmente cabe destacar que este conjunto de herramientas es *open source*, con lo cual su utilización es libre. Además, las constantes actualizaciones de versiones e incorporación de componentes, hace de este conjunto de herramientas una muy buena apuesta a futuro.

La solución que brinda este conjunto de herramientas, apunta a una solución BPM que puede ser implementada bajo una plataforma SOA (*JBoss enterprise SOA platform*), con servicios que estén disponibles en el ESB (*JBoss enterprise service bus*).

Capítulo 6 - BONITA

Introducción

Es una solución *BPM/workflow* liviana que provee soporte a XPDL 1.0. Nova Bonita posee un módulo de extensión XPDL, una API de *workflow* completa, soporte para múltiples tipos de variables, instanciación de actividades múltiples, versionado de procesos y un conjunto de servicios.

La tecnología Nova está basada en la Máquina Virtual de Procesos (PVM). Esta máquina virtual define un motor de proceso genérico que da soporte para múltiples lenguajes de procesos tales como BPEL, XPDL, etc. Esta solución de diseño del motor de procesos tiene la característica de otorgarle al analista de negocios libertad a la hora de modelar. Adicionalmente, permite al desarrollador utilizar tecnología de procesos embebida en una aplicación Java [17].

Arquitectura

Los componentes de la arquitectura de Nova Bonita son los siguientes:

- Un Motor de Procesos Genérico: la Máquina Virtual de Procesos (PVM).
- Soporte para el estándar XPDL 1.0.
- Consola de Procesos *web* 2.0¹².
- Un Diseñador de Procesos Gráficos XPDL (en 2 versiones: *plugin* para Eclipse o aplicación de escritorio).
- Contenedor de Servicios, tales como, Persistencia, Seguridad, Identidad, Conexión, *log* entre otros.

En la Figura 6-1 visualizamos los componentes que forman parte de la arquitectura Nova Bonita.

¹² El término Web 2.0 fue acuñado por Tim O'Reilly en 2004 para referirse a una segunda generación en la historia de la Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs, los wikis o las folcsonomías, que fomentan la colaboración y el intercambio ágil de información entre los usuarios. Extraído de: http://es.wikipedia.org/wiki/Web_2.0

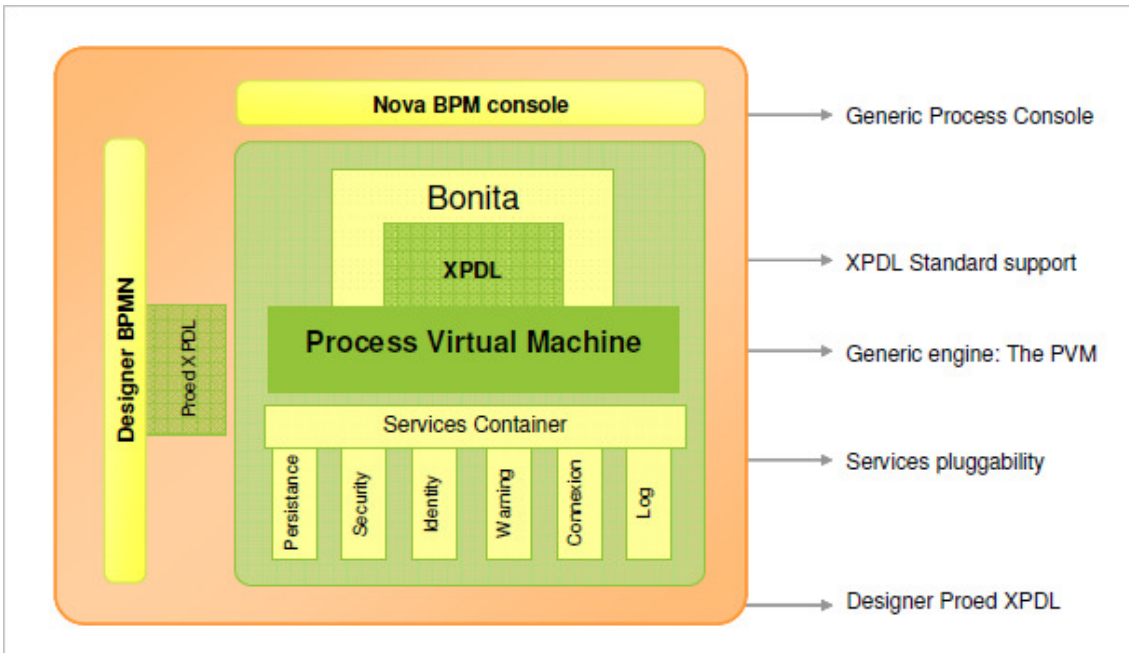
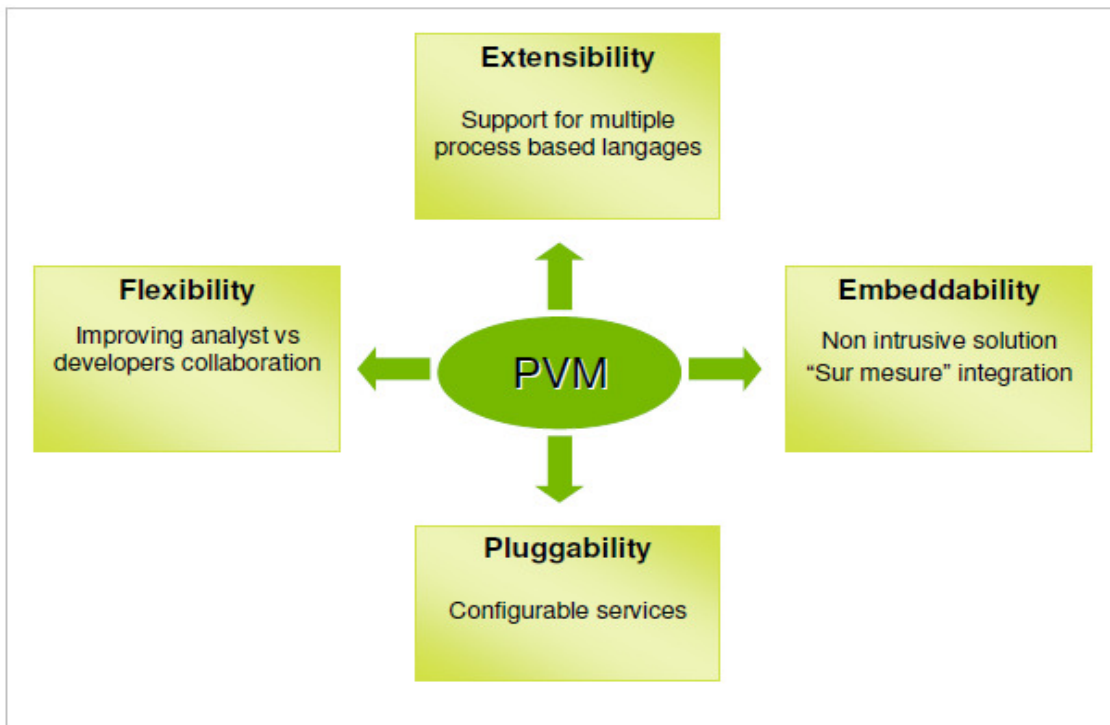


Figura 6 - 1: Arquitectura de la herramienta Nova Bonita. Extraída de: “*Bonita The Open Source Workflow Project*”. OW2 Consortium y Bull. Pag 1.

La Máquina Virtual de Procesos (PVM) es una tecnología para aplicaciones basadas en procesos. Como implementación por defecto, Bull y JBoss han entregado un motor genérico de procesos que permite construir *workflow* y soluciones BPM extensibles. La Figura 6-2 muestra las características principales que posee la PVM de esta herramienta.



The Process Virtual Machine

Figura 6 - 2: Características más destacables de la PVM. Extraída de: “*Bonita The Open Source Workflow Project*”. OW2 Consortium y Bull. Pag 2.

Características Principales de la Herramienta

- Potente API de *workflow* que cubre el despliegue, definición, ejecución e historial de datos de *workflow*.
- Soporte para actividades XPDL 1.0: *join, split, activity*.
- Soporte para los elementos principales de XPDL 1.0: *datafield, dataType, participant, transition*.
- Soporte para Entidades/Recursos avanzados: *hooks, mappers, performer assignments* e instanciación múltiple de actividades (a través de los atributos extendidos de XPDL).
- Ejecución Persistente a través de un servicio de persistencia configurable (por defecto este servicio lo provee Hibernate).
- Soporte de Subprocesos.
- Soporte de Ciclos e Iteraciones.
- Módulos Diario e Historial configurables.
- Capacidades avanzadas de despliegado de procesos, incluyendo archivos “.bar”.
- Versionado de procesos y paquetes.
- Ciclo de vida unificado para la ejecución de actividades *workflow*, manejando sincronización con el ciclo de vida de las tareas, conocidas como actividades manuales.

Nova Bonita incluye soporte para elementos definidos en el estándar XPDL 1.0. Las siguientes versiones agregarán soporte para el estándar XPDL 2.0 cubriendo las nuevas características: Ejecución de Actividades Asíncronas, Modificación de Instancias, entre otras [17].

Principales Beneficios

- Reducción de costos y riesgos, automatizando procesos sistema a sistema y persona a persona. Los procesos pueden abarcar organizaciones y localidades geográficas.
- Manejo eficiente de situaciones inesperadas: Bonita permite en forma dinámica y segura, modificar la definición de un proceso en ejecución.
- Eficiencia: capacidad para cambiar un proceso en progreso, evolución de procesos a futuro y adaptación al contexto del cliente.
- Entorno gráfico basado en *web 2.0*.

- Arquitectura flexible en la cual todos los servicios de negocios se pueden conectar: Persistencia, Transacciones, Seguridad, Identidad, Mensajes Asincrónicos o contadores de tiempo y también Administración de Tareas, BI o BAM¹³.
- Integración con el Proyecto *open source* Orchestra para dar una solución integrada BPM agregando soporte para los estándares XPDL y BPEL.
 - Integración no Intrusiva
 - Versión Liviana: desplegable en cualquier aplicación existente Tomcat, Aplicaciones *swing*, etc.
 - Versión Empresarial: desplegable en cualquier servidor Java EE (Jonas, JBoss, Weblogic, etc).
 - Versión “A Medida”: Adaptada a cualquier entorno particular del cliente [18].

Requerimientos de Hardware y Software

Hardware

Procesador de 1 Ghz, es recomendado, con un mínimo de 512 de RAM. Los usuarios Windows pueden evitar ajustar el archivo de *swap* y mejorar la performance usando 1 Gb o más de RAM [17].

Software

Para funcionar requiere la instalación de la *Java virtual machine* (jdk 1.5 o 1.6).

Sistemas Operativos Soportados

- Solaris-10 (SunOS 5.10) x86
- GNU/Linux kernel 2.6.25-2 x86 Debian
- Windows XP [17]

Licencia

La herramienta Nova Bonita se rige bajo la licencia LGPL. Esta licencia de *software* creada por la *free software foundation*, permite utilizar, compartir y modificar dicho *software* por el usuario final [18] [19].

¹³ Es un software que ayuda a monitorear las actividades de negocio, donde estas actividades están implementadas en un programa de computadora. Extraído de: http://en.wikipedia.org/wiki/Business_activity_monitoring

Instalación y Configuración

Instalación del Bonita Eclipse Designer

Para realizar esta instalación, hay que descomprimir el zip que contiene el plugin llamado bonita-eclipseDesigner (se puede descargar desde http://forge.ow2.org/project/showfiles.php?group_id=56) en cualquier IDE Eclipse con versión superior a la 3.2.

Este zip tiene una carpeta plugins, por lo cual debe ser descomprimido dentro de la carpeta raíz de la IDE Eclipse. Una vez descomprimido se debe reiniciar el Eclipse para terminar la instalación. Para comprobar que se instaló correctamente se debe verificar que en el menú “file->new” dentro de la IDE, disponemos de la opción “Bonita workflow” [20].

Instalación del Bonita Console

Este componente trae embebido una versión integrada del motor de ejecución (*runtime*), y de la consola de administración. Todos estos componentes están montados en un servidor de aplicaciones Tomcat.

Existe la posibilidad de instalar de forma separada el motor de ejecución en un ambiente distribuido (Cliente/Servidor) o en aplicaciones *web* personalizadas.

La instalación de la consola se realiza descomprimiendo el archivo zip de la consola en cualquier directorio. Esto creará una carpeta bonita-console-XX. Para arrancar la consola de Nova Bonita hay que dirigirse al directorio /bin dentro de la carpeta antes instalada, y ejecutar el comando “bpm run”. Esto automáticamente inicia el motor de ejecución y despliega la consola *web* dentro del Tomcat. El acceso a la consola se realiza ejecutando la siguiente URL dentro del navegador *web*:

<http://localhost:8080/portal/>. El usuario por defecto para acceder en la consola es *root/bpm* (usuario administrador) [20].

Persistencia de Datos

Base de Datos Soportadas

- Mysql-server 5.0.51a-6
- Postgresql 8.3.3-1
- Oracle 11.1.0
- H2 1.0.76
- HSQL 1.8.0.7 [17]

Configuración Inicial

La distribución Nova Bonita está configurada por defecto para correr con una base de datos embebida HSQL (*hyperthreaded structured query language database*), se trata de un motor de base de datos *open source* escrito en Java [21]). La configuración para HSQL está disponible en el directorio `/conf` de la distribución (archivo *hibernate.core.properties*). Esta configuración se utiliza para persistir la información de definición y ejecución (*runtime*) de los componentes. Para complementar esto se utiliza otro archivo llamado *hibernate-history.properties* que almacena la configuración de la base de datos historial.

Para utilizar otro motor de base de datos en Nova Bonita se deben realizar los siguientes pasos:

- Copiar el controlador del dispositivo de la base de datos a utilizar dentro del directorio `/lib` de la distribución (por ej. MySQL.jar, Oracle.jar).
- Configurar el archivo *hibernate.properties* con los datos de la base de datos a utilizar. Si se quiere configurar la base de datos del historial hay que cambiar también el archivo *hibernate-history.properties*.
- Acceder al directorio principal de la instalación y ejecutar el comando “*ant init-db*”. Este comando inicializa la base de datos con el esquema de Nova Bonita. Se debe seleccionar qué base de datos se quiere inicializar (principal o historial) [17].

Soporte de Tecnología Java

Como ya lo hemos mencionado, para realizar el diseño e implementación de los procesos de negocios, se utiliza un plugin de Eclipse que permite crear proyectos Bonita con la estructura necesaria para el desarrollo en JAVA y que a su vez provee un conjunto de herramientas visuales para el desarrollo de los diagramas que representaran los procesos de negocio.

Un Hook de Bonita es un componente que nos permite asociar las actividades automáticas con clases Java para realizar la implementación de las mismas. De forma similar a jBPM, un Hook es una interfase que tiene un único método donde se debe completar con el código Java correspondiente para implementar la actividad.

Análisis de la Suite y ciclo de vida BPM soportado

Componentes de la Herramienta

La suite está compuesta por tres componentes diferentes que pueden ser utilizados de forma separada o en conjunto. Se permiten varias configuraciones para estos componentes: JSE (Estándar) vs JEE (enterprise). Estos componentes son:

- **Motor de Ejecución (*runtime*) Nova Bonita:** este componente es el motor de procesos. Los procesos pueden ser desplegados, ejecutados y monitoreados a través de una API que provee servicios BPM.
- **Consola Nova Bonita:** interfase gráfica basada en *web 2.0* que potencia la experiencia del usuario durante las fases del desplegado, ejecución y monitoreo.
- **Diseñador Gráfico Nova Bonita:** es el entorno de desarrollo BPM que permite de forma gráfica definir procesos, como así también, conectores a los sistemas de información existentes [20].

Modelado e Implementación

El modelado y la implementación, se realizan desde la IDE Eclipse debidamente instalado con el *plugin* ProedXPDL (disponible para su descarga desde http://forge.objectweb.org/project/download.php?group_id=56&file_id=8040).

Para comenzar, se debe crear un nuevo “Bonita project” desde la IDE, esto nos creará la estructura de directorios con las librerías necesarias. A continuación, se debe crear en la carpeta “src\xpdl” un “Bonita XPDL file”, el cual contendrá la definición del proceso. Dentro de la estructura de directorios, se encuentra la carpeta “BAR”, la cual contendrá un archivo .bar que es la definición del proceso a desplegar y sus conectores, que se irá actualizando a medida que se vayan guardando los cambios en el archivo XPDL que contiene el diseño del proceso. La Figura 6-3 muestra la estructura de archivos y directorios generada al crear un proyecto Nova Bonita en el plugin de Eclipse.

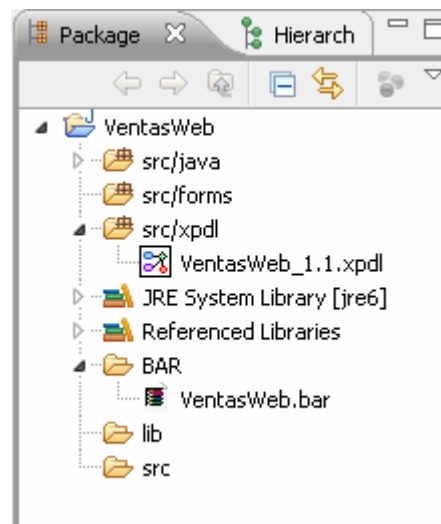


Figura 6 - 3: Estructura de directorios generados por el plugin ProedXPDL.

El archivo XPDL tiene 3 vistas diferentes:

- **Vista *activity*:** muestra el diagrama principal en notación XPDL 1.0, el flujo del proceso, las actividades y sus conexiones. La Figura 6-4 ilustra la vista *activity* de un proceso de negocios modelado en Nova Bonita.

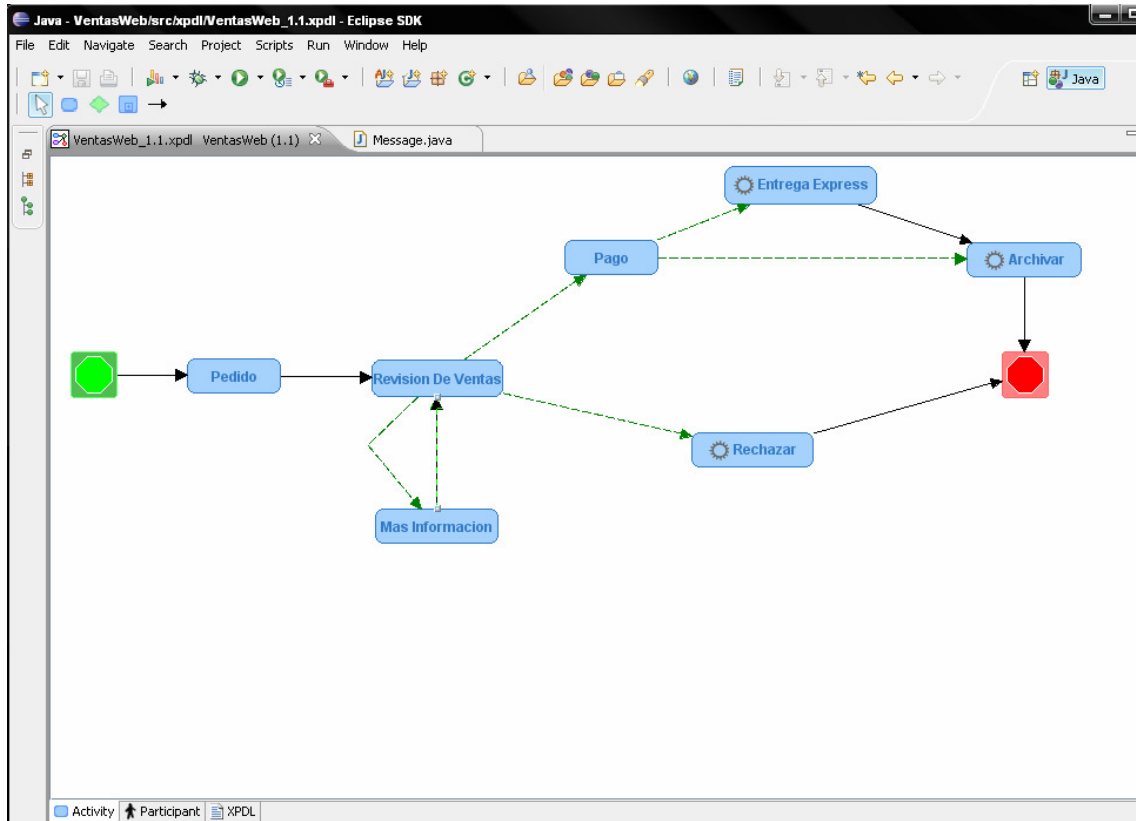


Figura 6 - 4: Vista *activity*.

- **Vista *participant***: muestra el mismo flujo anterior desde el punto de vista de las actividades que realiza cada rol participante del proceso (*swimlanes*). La Figura 6-5 visualiza la vista *participant* de un proceso de negocios, siendo posible distinguir a qué rol pertenece cada actividad.

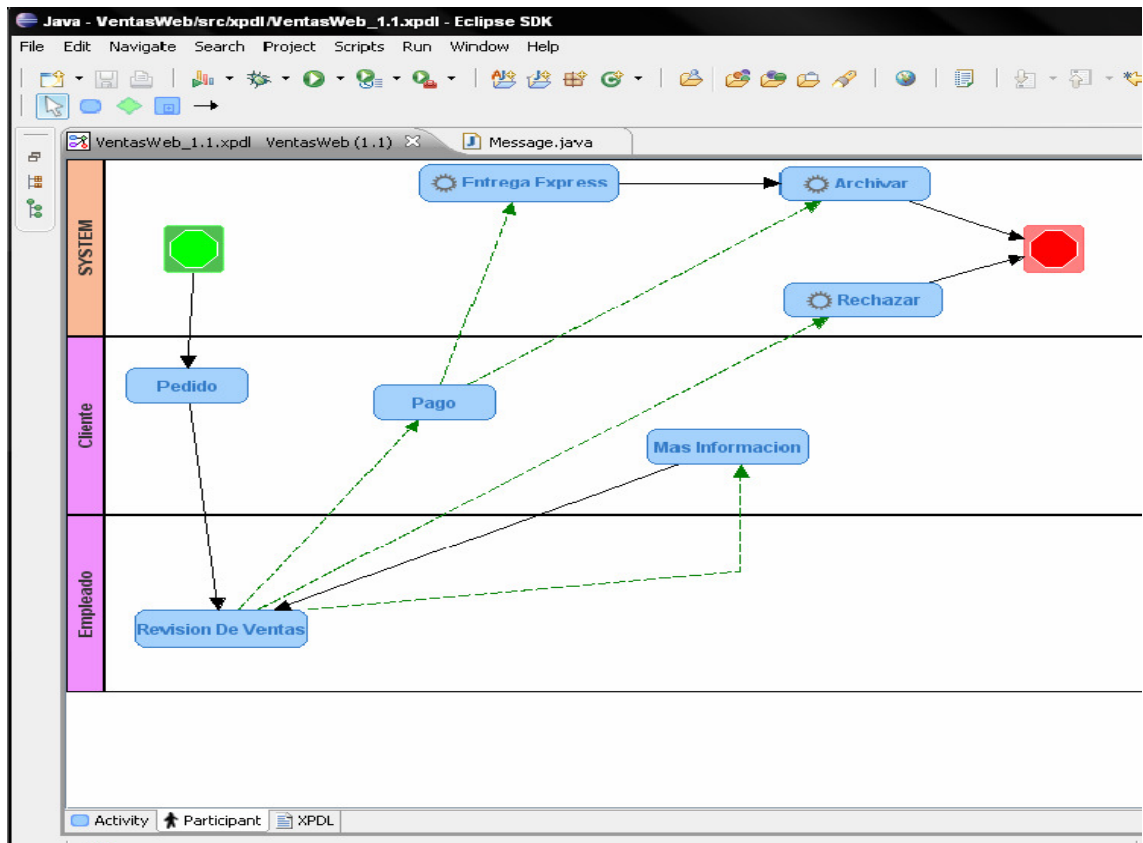


Figura 6 - 5: Vista *participant*.

- **Vista XPDL:** muestra el código XPDL que se genera automáticamente a medida que se completa el gráfico. La Figura 6-6 describe el código XPDL que se genera a medida que se modela un proceso de negocio.

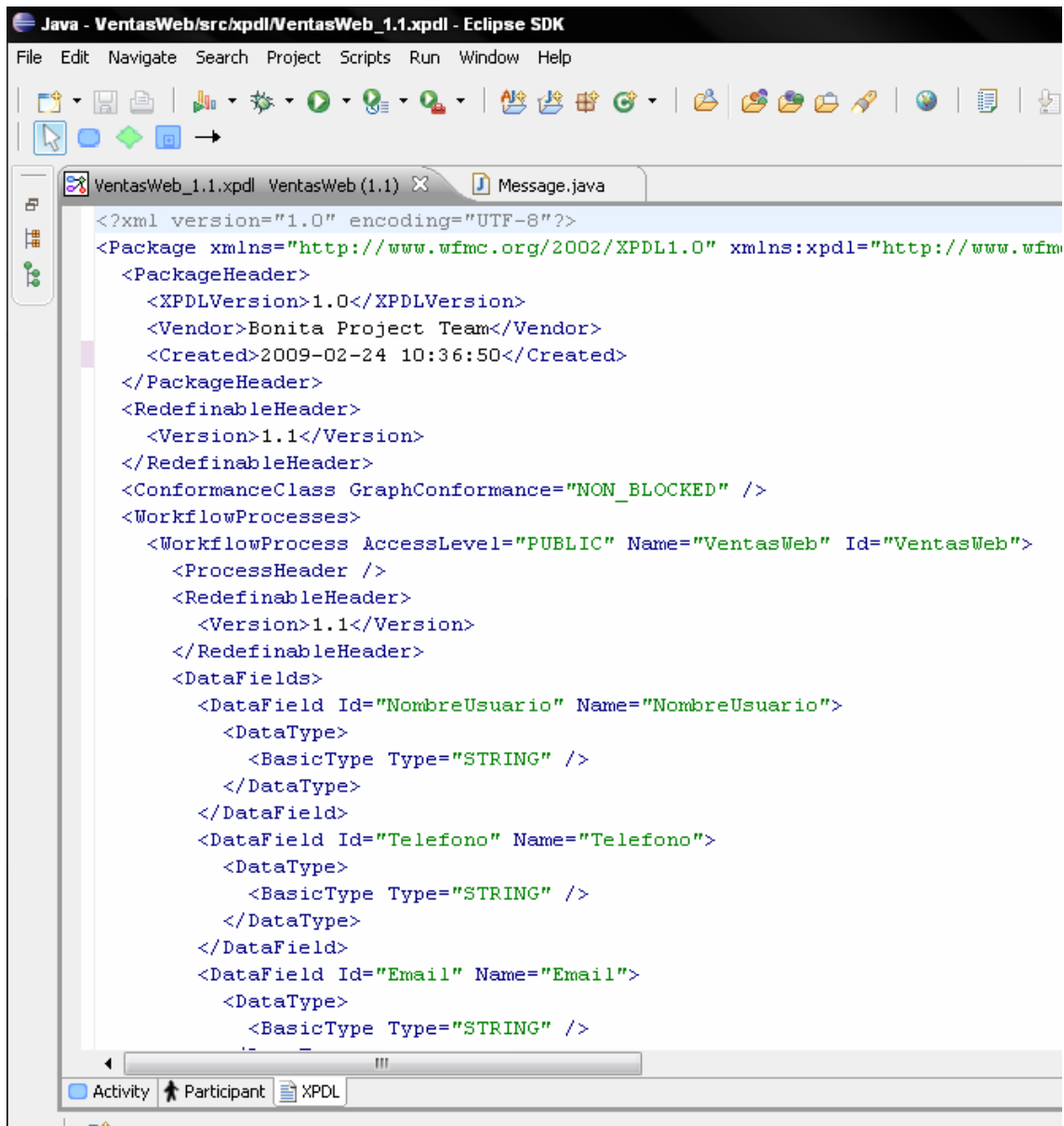


Figura 6 - 6: Vista XPDL.

Estas vistas están sincronizadas, un cambio en cualquiera de ellas se refleja en las vistas restantes.

Con respecto a los componentes que se pueden utilizar en el diseñador, están disponibles sólo los componentes básicos como actividades manuales y automáticas, transiciones, eventos de inicio y fin, y *swimlanes* entre otros. Estos componentes tienen un conjunto rico de propiedades que permiten configurar su comportamiento. También cabe destacar, que estos componentes respetan el estándar BPMN, haciendo el diagrama fácilmente comprensible para la gente de negocios.

El conector más importante de Nova Bonita es el *hook*; este componente permite enlazar las actividades automáticas con clases Java para realizar la implementación de las mismas. El hecho de agregar un *hook* a una actividad, convierte a la misma en automática. A bajo nivel un *hook*, es una clase Java que implementa una interfase, que

define un método *execute()* que se debe completar con la funcionalidad que corresponda.

Con respecto al lenguaje de definición de los procesos, como mencionamos anteriormente, el utilizado es XPDL. Este es un lenguaje estándar que si bien es fácil de comprender, requiere de un conocimiento previo del mismo.

El *plugin* de Eclipse cuenta con una gran cantidad de asistentes que facilitan la tarea a la hora del modelado e implementación del proceso, por ejemplo para la creación de *hooks*, XPDL, *role mapper*, etc.

El diseño de los procesos, se realiza en forma intuitiva y sencilla, respetando el estándar BPMN en casi todos sus componentes. Además, no presenta ningún detalle técnico en el diagrama, ocultando los mismos dentro de las propiedades de los componentes, logrando un diagrama mucho más legible.

Despliegue

La fase de despliegue se realiza de forma única, desde la consola Nova Bonita, yendo a la ventana “BPM *management*”. Desde esta ventana, se debe ubicar el archivo *.bar* que contiene la información de despliegado del proyecto. La Figura 6-7 muestra el formulario para desplegar procesos de negocios en el servidor de Bonita.

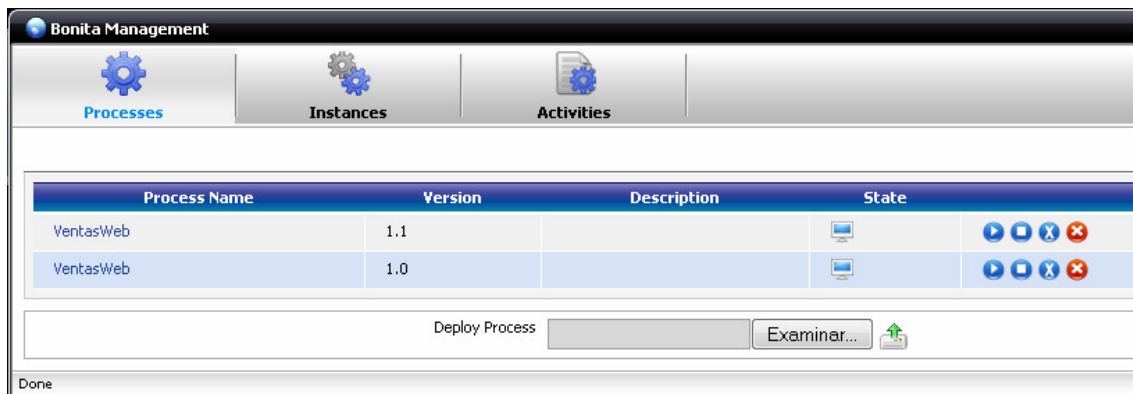


Figura 6 - 7: Ventana donde se realiza el despliegue de un proceso de negocio.

Si no se encontraron errores en la definición de los procesos durante el despliegue, los mismos estarán disponibles para ser ejecutados.

La administración de las versiones de los procesos desplegados se maneja manualmente desde el *plugin* de Eclipse. Cuando se quiere modificar la versión del proceso, hay que utilizar la opción “*save as*” sobre el archivo XPDL que define el proceso, luego se muestra un campo con la versión actual. La única forma de cambiar la versión es guardarla con un número mayor a la versión actual.

El *plugin* de la IDE Eclipse, carece de alguna opción para realizar el despliegado directamente desde el ambiente, haciendo necesario realizar el despliegado de forma manual como se mencionó anteriormente.

Ejecución y Monitoreo

La ejecución y monitoreo de los procesos se realiza desde la consola Nova Bonita. Esta consola se basa en un entorno visual *web* 2.0, lo que la hace muy agradable visualmente para los usuarios. Tiene un menú basado en iconos para facilitar el acceso a las diferentes ventanas y aplicaciones de la consola. Además, trae un conjunto de *portlets*¹⁴ que pueden ser incorporados a gusto del usuario para agregar funcionalidad extra (Calculadoras, Calendarios, etc).

Desde la ventana *BPM management* se puede gestionar y monitorear los procesos del sistema. Tenemos tres solapas en esta ventana:

- **Processes:** podemos ver la lista de procesos desplegados en el sistema y también desplegar nuevos procesos. De cada proceso de la lista es posible conocer la versión, el estado actual y un menú con botones para realizar acciones de arranque, detención, pausado y eliminación del mismo.
- **Instances:** podemos visualizar la lista de instancias creadas de cada proceso. Cada una tiene una versión asociada que es manejada por el servidor. También, podemos ver por cada instancia el estado actual, fecha de comienzo y fin de la instancia, rol que la inició y un conjunto de botones para administrarla.
- **Activities:** nos muestra la lista de actividades de cada instancia junto con el estado actual, el tipo de actividad, a qué instancia pertenece, fecha de comienzo y fin, junto a un conjunto de botones para administrarla.

Desde la ventana *users worklist* podemos visualizar la lista de tareas para un determinado usuario. Esta ventana tiene dos solapas: una lista de tareas que muestra las actividades pendientes de ejecución que son responsabilidad del usuario y una lista de tareas realizadas que muestra un historial de actividades realizadas por el mismo usuario. Si el usuario tiene permisos puede crear instancias de procesos y administrar o completar actividades. En la Figura 6-8 podemos apreciar el entorno gráfico de la consola de administración de procesos de Nova Bonita.

¹⁴ Los portlets son componentes modulares de interfaz de usuario gestionada y visualizada en un portal web. Extraído de: <http://es.wikipedia.org/wiki/Portlets>

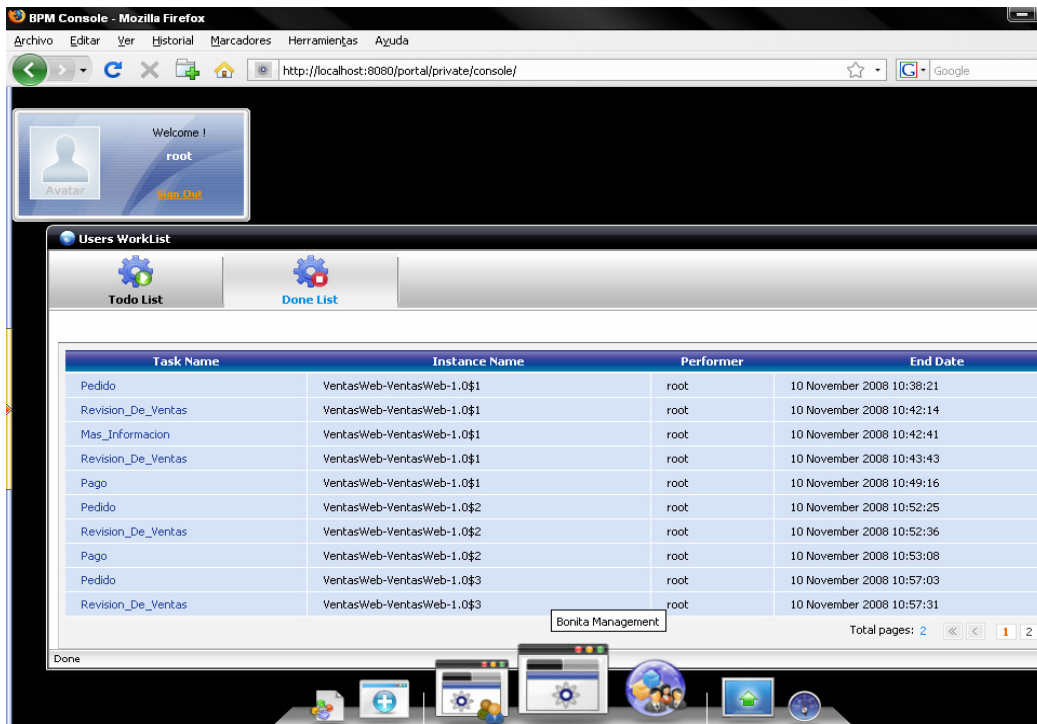


Figura 6 - 8: Imagen que muestra la consola *web 2.0* de Bonita, se puede apreciar que el entorno gráfico es agradable.

En lo que respecta a los formularios, la herramienta los genera basándose en la detección automática de las variables y sus tipos definidos para cada actividad manual definida en el diseñador. La consola realiza verificaciones de tipos para las variables del formulario y muestra las mismas de forma particular según el tipo de dato de la variable (Ej. los tipos *date* se muestran con un calendario *web 2.0*).

Conclusión General de la Suite

El conjunto de herramientas que componen la suite permiten cubrir las etapas básicas en el desarrollo de procesos de negocios: modelado, implementación, ejecución y monitoreo.

Estas herramientas no dan soporte a etapas como Simulación y Optimización que son muy importantes a la hora de realizar un desarrollo BPM a nivel empresarial. Estas etapas son claves para detectar fallos y errores de forma temprana, permitiendo depurar y optimizar los procesos, reduciendo costos y tiempo de desarrollo.

El diseño de procesos está inspirado en BPMN ya que soporta la versión XPD 1.0. La versión 2.0 trae soporte pleno para el estándar BPMN y va a ser incorporada en futuras versiones de Bonita. El modelo brinda varias vistas que son fáciles de comprender por la gente de negocios, haciendo mucho más sencilla la participación de los mismos en el diseño de procesos. El modelado es muy sencillo ya que el conjunto de componentes es reducido.

Los requerimientos de *hardware* no son exigentes ya que las herramientas son livianas, esto hace que la suite se pueda ejecutar en cualquier PC hogareña.

La instalación y configuración de las herramientas es muy sencilla, necesitando muy pocos pasos para completar la misma. La documentación asociada al producto es muy completa y fácil de comprender.

Las herramientas no dan soporte a BPEL, ya que el lenguaje de ejecución asociado a las mismas es XPDL. El soporte a BPEL está dado por otro proyecto *open source* paralelo a la empresa Bull que se llama Orchestra, y está basado en la misma Máquina Virtual de procesos.

La herramienta de diseño e implementación no trae ningún componente que de soporte para incluir servicios en los procesos diseñados, ni tampoco permite realizar conexiones de actividades con servicios. Por lo tanto, si se quiere efectuar esto se debe realizar manualmente (generar el cliente del servicio en Java y luego invocarlo en algún *hook*).

Debido a la carencia de un ESB y de alguna facilidad para la invocación de servicios dentro del flujo de los procesos de negocio, este conjunto de herramientas desde nuestro punto de vista está destinado a proveer una solución BPM que no requiera una plataforma SOA ni tampoco invocaciones a servicios, siendo ideal cuando el proyecto está codificado en el lenguaje Java, ya que la codificación de las actividades se realiza mediante este lenguaje. Consideramos también que es adecuado para pequeños y medianos desarrollos de *software*, ya que carece de soporte para cubrir etapas importantes en el desarrollo de procesos.

Capítulo 7 - ConsistGPA

Introducción

ConsistGPA - BPM Suite, es un Administrador de Procesos de Negocios de empresas que permite diseñar, controlar e integrar los procesos de negocio con las aplicaciones existentes.

ConsistGPA fue especialmente diseñado con características que permiten establecer una relación productiva entre el negocio y la tecnología, de manera tal de resolver el desafío de los ejecutivos de negocio automatizando e integrando sus procesos. Estas características permitirán realizar lo siguiente:

- Potenciar a los responsables de procesos, para diseñar y controlar sus propios procesos con agilidad y flexibilidad.
- Permitir que los participantes del proceso, como son los empleados, gerentes, clientes y socios trabajen más eficientemente.
- Brindar al personal de informática las capacidades para maximizar su infraestructura existente de IT, integrando los procesos de negocio con las aplicaciones existentes y extenderlos a nivel de empresa, clientes y proveedores [22].

ConsistGPA basa su potencia en los tres pilares fundamentales de la tecnología de BPM, como son *workflow*, *enterprise application integration* (EAI) y *business intelligence*¹⁵ (BI). ConsistGPA incorpora innovaciones tecnológicas tales como administración de documentos electrónicos, firma digital, xml, *e-mails*, *web services*, internet, etc [23].

Características Técnicas

Las características técnicas de esta suite son las siguientes:

1. Desarrollo orientado a objetos.
2. Desarrollo en tres capas: *user interfase*, *business task* y *business object*.
3. La aplicación podrá correr en intra / extra / o internet.
4. Administración centralizada.
5. Soporte para cualquier RDBMS.
6. Integración de correo electrónico en la aplicación.

¹⁵ Se denomina inteligencia empresarial o inteligencia de negocios o BI (del inglés *business intelligence*) al conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa. Extraído de: http://es.wikipedia.org/wiki/Business_intelligence

7. Integración vía XML.
8. Almacenamiento de documentos en servidor XML Nativo.
9. Tecnología abierta.
10. Herramientas de consultas y reportes.
11. Auditoría de movimientos.
12. Multiplataforma: Unix, Linux, Windows y otros [22].

Arquitectura de la Suite

En la Figura 7.1 se muestra la representación de todos los módulos y componentes de la suite, junto a la comunicación de los mismos a través del motor de ejecución. Pieza clave de la suite ConsistGPA [24].

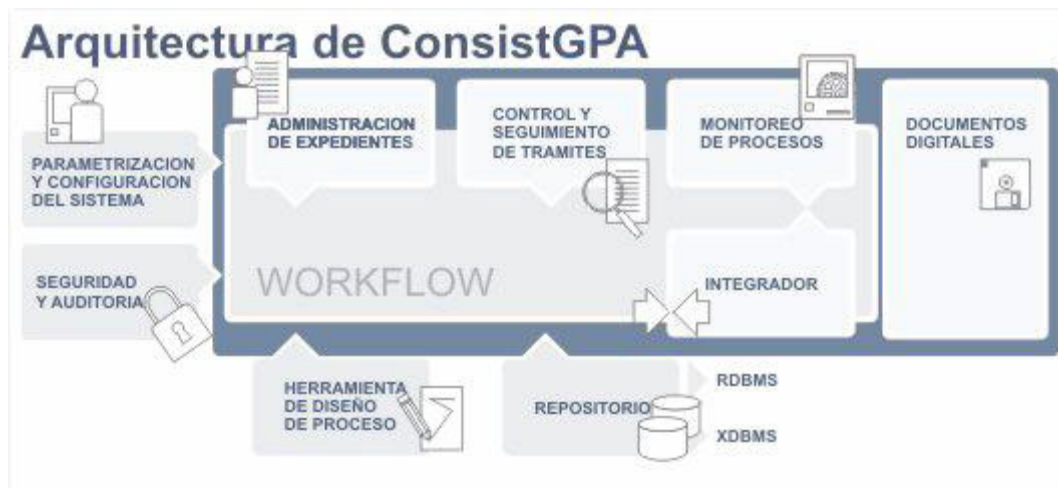


Figura 7 - 1: Módulos y componentes de la suite ConsistGPA. Extraída de: “Manual de Usuario Herramienta de Definición de Procesos”. Consist. Pag 4.

En particular, la herramienta de diseño de proceso interactúa con el motor de ejecución para poder realizar cada una de las operaciones sobre un proceso y acceder a las bases de datos [24].

Herramienta de Diseño de Procesos

Esta componente gráfica permite diseñar los procedimientos administrativos a implementar en la organización usando conceptos de interacción con el usuario como arrastrar y soltar, deshacer, copiar y pegar. Esta herramienta, interactúa con el motor de *workflow*, a través de XML sobre HTTP, permitiendo validar las definiciones realizadas, mantener diferentes versiones de un proceso y su estado, indicando si está en etapa de diseño, de *testing*, en uso productivo, etc. A su vez, permite documentar cada uno de los pasos y exportar e importar definiciones de procesos en formato XML [22].

Motor de Workflow

Alineada con los estándares definidos por la *workflow management coalition* (WfMC), su objetivo es controlar que se respeten los procedimientos administrativos definidos en la organización, guiando a los usuarios en la realización y el cumplimiento de etapas y objetivos, de los distintos documentos.

El motor de *workflow* interpreta las definiciones realizadas mediante la herramienta de Definición de Procesos, reconociendo conceptos como procesos estrictos y flexibles, subprocesos, versiones, actividades manuales y automáticas, diferentes tipos de participantes, roles, pre y post condiciones, flujos obligatorios y/o opcionales, contextos de ejecución, actividades contextuales, etc. El motor de *workflow* mantiene cada caso con toda la información de control que necesita para asegurar que el trámite administrativo respete las definiciones realizadas, y además, brinda información sobre cómo y cuándo se realizó cada actividad y en qué contexto.

El usuario podrá registrar la cancelación o finalización de los distintos casos (a las instancias de los procesos de negocio en esta herramienta se los conoce como casos) y/o actividades, e inclusive su suspensión o detención ante condiciones excepcionales no previstas, tanto de casos como de actividades puntuales [22].

Documentos Digitales

ConsistGPA dispone de una herramienta de generación de formularios para definir cualquier formulario en la organización, logrando que el mismo quede totalmente integrado al ambiente de trabajo del usuario y sus procesos de negocio.

A partir de la definición de un formulario HTML, utilizando el editor que se desee, ConsistGPA lo incorpora al catálogo de formularios de la aplicación, para luego realizar la generación del mismo y dejarlo disponible para su utilización dentro del ambiente de trabajo de usuarios. Estos formularios electrónicos basan su interfase de presentación en plantillas HTML, con lo cual la personalización de los mismos se realiza a través de cualquier herramienta de diseño de HTML. También pueden establecerse reglas de validación y presentación de interfase sobre cada campo ingresado, permitiéndose la integración con datos (internos y externos) u otras aplicaciones de la organización. Para cada formulario el usuario puede salvaguardar diferentes versiones, pudiendo luego consultar la conformación histórica del mismo. Con el fin de potenciar y maximizar los beneficios de los formularios XML, se implementan tecnologías de firma digital para este tipo de documentos. Pueden definirse diferentes niveles de acceso, para el usuario creador del formulario, sus compañeros de área y el resto de los usuarios de la organización.

ConsistGPA, posibilita además trabajar con cualquier otro documento electrónico, ya sean contenidos generados por manejadores de contenidos, generados por herramientas de Microsoft Office, imágenes, PDF, u otros objetos de negocio de diferentes aplicaciones.

Los documentos digitales pueden estar almacenados en el repositorio propio de ConsistGPA o pueden ser accedidos desde diferentes repositorios como es el caso de los contenidos de ConsistCM [22].

Integrador

ConsistGPA provee un conjunto de herramientas y servicios que permiten que las aplicaciones de la organización o de otras organizaciones interactúen en forma bidireccional con los procesos definidos en ConsistGPA. Estas aplicaciones a integrar podrían ejecutarse en plataformas heterogéneas como ser Windows, diversos Unix, Linux, AS/400 o *mainframes*¹⁶, o podrían estar desarrolladas en diversos lenguajes y con diferentes fuentes de datos.

ConsistGPA, haciendo uso de los servicios estándares o productos tales como *activity connect*, permite acceder virtualmente a aplicaciones o fuentes de datos ya existentes, tanto para operaciones de lectura como de lectura-escritura.

El Integrador se encarga de controlar y regular el avance de los casos dentro y fuera de la organización, basándose en la definición del proceso de negocio asociado a cada tipo de documento.

Bajo el control del motor de *workflow*, el usuario transita el proceso administrativo a través del sucesivo pase entre actividades. ConsistGPA informa en cada paso sobre la tarea administrativa concreta que corresponde realizar y controla las condiciones establecidas sobre su realización (duración máxima de la tarea, documentos digitales que deben ser vinculados al caso, etc.) La circulación de documentos físicos, de alta importancia entre distintas oficinas, utiliza según se halla definido, un esquema de envío y aceptación, basado en el uso de constancias escritas. Se dispone en forma inmediata tanto de la información actual como histórica, que permite conocer la situación real del caso, visualizando las actividades, los estados y propiedades de las mismas, el contenido digital que se le ha ido incorporando, etc.

Se brindan herramientas que permiten ver fácilmente los documentos que ingresan al sector y distribuyen la carga de trabajo entre los empleados. De esta forma cada usuario puede conocer en forma inmediata los documentos sobre los cuales tiene que trabajar, contando con alertas y avisos sobre tareas demoradas o vencidas, provistas por el Motor de *workflow* [22].

Reportes y Monitoreo de Procesos

ConsistGPA brinda un conjunto de reportes operativos que permiten analizar la iniciación, el avance y la finalización de los procedimientos administrativos. Pueden obtenerse reportes cuantitativos y/o detallados, considerando un cierto período de tiempo, una o varias áreas de la organización, determinados procesos de negocio, etc.

Cada usuario puede por si mismo solicitar la ejecución de los reportes que le están autorizados y continuar con su trabajo. Cuando lo crea pertinente puede obtener los resultados en pantalla (en formato PDF) para su consulta y/o impresión. Existen herramientas de auditora sobre los procesos ejecutados y la posibilidad de recuperar reportes históricos [22].

¹⁶ Una computadora central (en inglés mainframe) es una computadora grande, potente y costosa usada principalmente por grandes compañías para el procesamiento de una gran cantidad de datos; por ejemplo, para el procesamiento de transacciones bancarias. Extraído de: <http://es.wikipedia.org/wiki/Mainframe>

Información Gerencial

Se dispone de potentes herramientas de análisis y navegación de estructuras multidimensionales que permiten mostrar al nivel directivo de la organización la información consolidada sobre la ejecución de los distintos procesos de negocio implementados, apoyando de esta manera a la toma de decisiones [22].

Dispositivos Móviles

La necesidad de tomar decisiones en los momentos y tiempos exactos o el acceso a información vital para la toma de decisiones, muchas veces surge cuando no tenemos una PC a mano, o peor aún, estando fuera de las instalaciones de la organización, debido a esto ConsistGPA cuenta con interfase WML (*wireless markup language*) y acceso vía WAP (*wireless application protocol*) para ser operado desde cualquier dispositivo móvil, como ser un teléfono celular o una *palm-top* [22].

Parametrización y Configuración de ConsistGPA

Esta componente permite acceder y trabajar sobre diferentes objetos de configuración del sistema, adecuando o actualizando la parametrización básica de la aplicación. Por ejemplo, se administra la estructura organizacional, manteniendo el árbol jerárquico de las unidades. Para cada unidad se puede registrar su titular, dirección de correo electrónico, etc. También pueden definirse unidades externas a la organización, que intervienen en los procesos de negocio. La nómina de usuarios habilitados, manteniendo los datos personales y laborales de cada uno de ellos. Si se requiere, también es posible definir nuevas tablas de configuración, por ejemplo, para ser usadas en los formularios digitales [22].

Seguridad y Auditoría

Este módulo controla el acceso a cada una de las funciones del sistema, verificando que el usuario tenga autorización para ejecutarla. Mediante la definición de perfiles de seguridad se logra definir el grupo de operaciones que cada usuario puede realizar. Por otra parte, esta componente mantiene pistas de auditoría de cada modificación realizada sobre los datos, permitiendo conocer exactamente cuándo y quién modificó determinada información.

Si en la organización se utilizan servicios de directorio activo (LDAP), ConsistGPA puede interactuar con los mismos de manera no invasiva [22].

Beneficios

Procesos de Negocio y Casos en ConsistGPA

Los procesos de negocios involucran documentos digitales y físicos, para estos últimos ConsistGPA cuenta con funcionalidad específica que permite el registro y mantenimiento de la información relativa al mismo, a través del cual se transforma un proceso de negocio genérico, en un caso real y puntual. El mismo también identifica unívocamente, y engloba, toda la información, tanto digital como física, perteneciente a una instancia particular del proceso de negocio o caso.

Para cada caso es posible registrar información sobre:

- El proceso que va a recorrer, dependiendo del objetivo por el cual el documento se encuentra en la organización.
- Las personas físicas y/o jurídicas que están relacionadas con el caso, pudiéndose definir múltiples tipos de relaciones o vínculos.
- El contenido digital, que se conforma incorporándole todo tipo de documento digital (archivos Word, Excel, imágenes, Formularios Electrónicos, etc).

Para cada caso es posible utilizar las herramientas específicas para:

- Localización de casos por diferentes y múltiples criterios accediendo a la consulta integral de la información.
- Unión de casos pertenecientes a procesos de negocio exactamente iguales o totalmente distintos.
- División de casos en diferentes subprocesos paralelos, regulados bajo las especificaciones de los casos originales y visualizados conjuntamente.
- Administración de Archivos físicos internos o externos a la organización.

Automatización de Procesos de Negocio en Diversas Industrias

ConsistGPA es capaz de automatizar una amplia gama de procesos de negocio de diferentes industrias. Estos procesos pueden involucrar un alto volumen de trabajo rutinario, transacciones en las que es necesaria la intervención humana, administración de contenidos digitales y documentos físicos, integración con aplicaciones existentes, etc [22].

Licencia

La licencia de esta herramienta esta dividida en dos partes:

- **Licencia Microsoft Visio:** se debe obtener la licencia de Microsoft Visio, ya que es una parte clave de ConsistGPA.
- **Licencia ConsistGPA:** la licencia de la herramienta está orientada a la cantidad de usuarios que la utilizan, dependiendo la cantidad de usuarios varía el precio. Para obtener mas información acceder a <http://www.consist.com.ar>

Requerimientos de Hardware y Software

Microsoft Visio 2003

Los requerimientos de *hardware* mínimos para utilizar este componente son:

- **Procesador:** PC Pentium 233 MHz. o superior Pentium III recomendado.

- **Memoria:** 128 MB de RAM (o superior) recomendado para el sistema operativo
- **Disco Rígido:** 245 MB de espacio disponible
- **Sistema operativo:** Microsoft Windows 2000 con *service pack 3* (SP3) o Microsoft Windows XP o un sistema operativo posterior [25].

ConsistGPA

La aplicación ConsistGPA está desplegada en un servidor de aplicaciones Tomcat 5, el cual no necesita de un *hardware* potente para funcionar. El *hardware* necesario para correr la aplicación dependerá de la cantidad de usuarios concurrentes que hagan uso de la consola.

Instalación y Configuración

Los pasos para realizar la instalación de la suite ConsistGPA en el entorno Windows son los siguientes:

1. Registrar el servicio de Tomcat para ejecutar la aplicación ejecutando [unidad de disco]\GPAOnline\services\GPA-Install.bat
2. Arrancar el servicio de Tomcat con el servicio generado que se llama "Consist GPA", en el caso que no esté iniciado. Esto se puede ver desde Panel de Control -> Herramientas Administrativas -> Servicios.
3. Iniciar la configuración de ConsistGPA desde el navegador *web* con la URL:
http://[nombre del servidor]:8089/gpa300ES/SLConfigTomcat
4. Configurar los parámetros en función de la descripción presentada en la pantalla. (posee opciones *default*). Estos parámetros incluyen la parametrización de la base de datos.
5. Acceder al producto con http://[nombre del servidor]:8089/gpa300ES/SLLogin [24].

Persistencia de Datos

Bases de Datos Soportadas

La suite ConsistGPA soporta cualquier base de datos relacional, entre las más destacadas están Oracle, SQL Server, MySql y Adabas D.

Configuración Inicial

Para poder utilizar la suite se debe definir una base de datos y cargar el esquema con sus datos para que la misma pueda funcionar. Para realizar esto se tienen 2 opciones:

- Restaurando un back up de Adabas D, Oracle o Sql Server existente en la siguiente ubicación: [unidad de disco]\GPAOnline\db

- Se puede crear vía SQL en función del diccionario de datos [26]

Soporte de Tecnología Java

La herramienta ConsistGPA no posee un ambiente de desarrollo, ofrece una plantilla de modelado sobre Microsoft Visio.

El punto de conexión con la tecnología Java es mediante los adaptadores, que permiten asociar objetos externos ligados al lenguaje (*web services*, Bases de Datos vía JDBC, Java wrappers, Clases Java) que pueden ser utilizados como implementación en una actividad automática.

Análisis de la Suite y ciclo de vida BPM soportado

Componentes de la Herramienta

Esta suite está compuesta por 2 componentes principales:

- **Microsoft Visio 2003:** permite realizar el diseño de los procesos y el despliegado de los mismos mediante una plantilla incorporada que provee acceso a estas funcionalidades.
- **Consola ConsistGPA:** es una aplicación *web* desarrollada en Java que maneja la ejecución, optimización y monitoreo de los procesos desplegados.

Modelado e Implementación

Como explicamos antes, las fases de Modelado e Implementación se realizan desde la herramienta Visio. Esta herramienta es un entorno gráfico de diseño con conceptos de arrastrar y soltar, deshacer, copiar y pegar, que interactúa con el motor de *workflow*, a través de mensajes XML sobre el protocolo http, permitiendo validar las definiciones realizadas, mantener diferentes versiones de un proceso y su estado (en uso, en desarrollo, etc.).

Eligiendo la plantilla *workflow*, y luego accediendo al servidor, podemos empezar a trabajar con esta herramienta. La plantilla agrega un nuevo menú en la parte superior, que tiene todas las operaciones para realizar la implementación y despliegue del proceso.

Además, tenemos disponible una paleta de componentes para diseñar los procesos. Los componentes respetan el estándar de notación BPMN. En la Figura 7-2 se puede apreciar a la izquierda la paleta de componentes y sobre la derecha uno de los procesos de negocio modelados en esta herramienta.

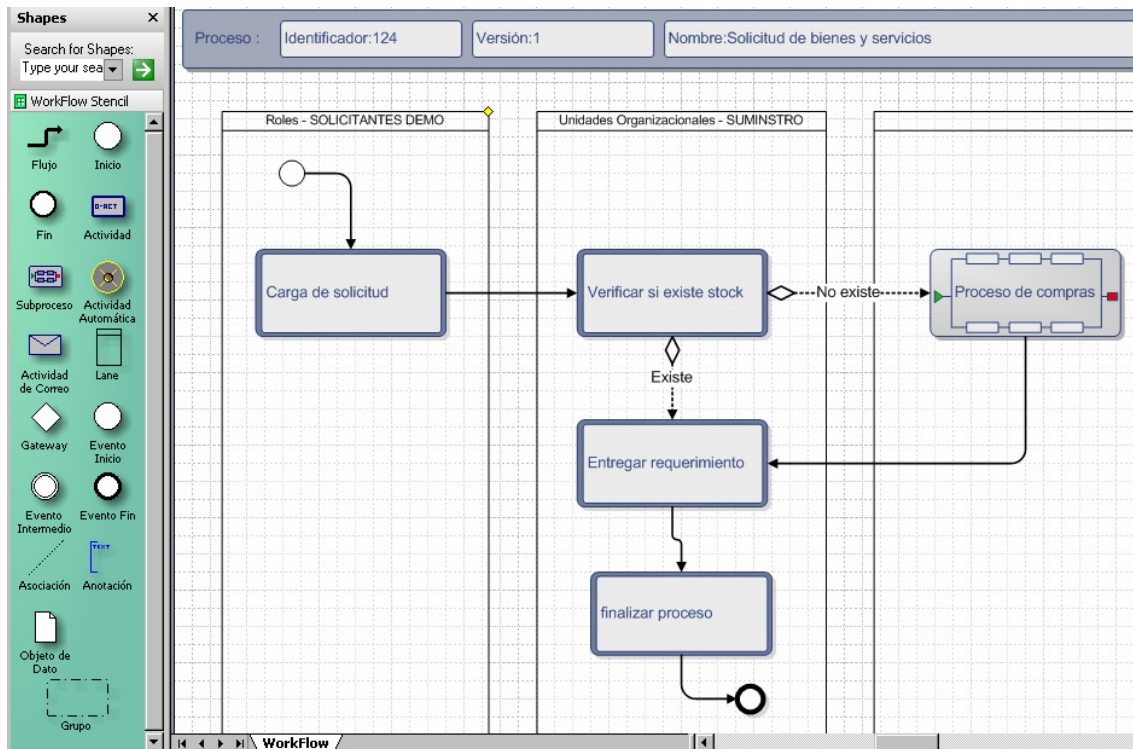


Figura 7 - 2: Modelado de procesos en Microsoft Visio.

Para comenzar a modelar se deben definir previamente los *lanes* que se van a utilizar, cada uno con un responsable (Usuario, Rol, Unidad Organizacional, Agente), ya que la consola en fases posteriores es muy restrictiva en lo relacionado a los responsables de cada actividad.

La paleta de componentes trae todos los componentes necesarios para poder construir diagramas en BPMN. Cada componente tiene un conjunto amplio de propiedades que permiten configurar su comportamiento, inclusive algunos de ellos cambian su semántica y aspecto visual (por ej, un flujo puede ser obligatorio, opcional y excluyente) dependiendo de la configuración interna que posean. La implementación en esta herramienta es visual, no se necesita codificar las actividades automáticas ni el control del flujo del proceso. Todo se configura desde el diseñador de procesos, ya que la suite no trae una IDE de desarrollo.

Para asociar objetos externos al sistema se utilizan adaptadores. Los tipos de adaptadores existentes son:

- *web services*
- Bases de Datos vía JDBC
- *Entire X Java wrappers*
- Clases Java

Como podemos ver la mayor parte de los tipos de adaptadores están ligados al lenguaje Java. En la definición del adaptador se especifican el tipo de adaptador, nombre y definición, y luego las propiedades específicas según el tipo elegido. Una vez definido

el Adaptador que permite identificar el objeto a integrar, la integración se completa definiendo el Componente. En el Componente se configura el comportamiento del mismo, es decir, cómo se realizará la integración.

Un Componente define un tipo y una interfase de comunicación con el sistema que lo utiliza. Este tendrá un conjunto de parámetros de entrada y salida. Desde el punto de vista de su utilización puede ser visto como una caja negra, a la que se le proporcionan valores en los parámetros de entrada, se le ordena ejecutar y se toman los valores que la ejecución deja en los parámetros de salida. Varios componentes pueden utilizar el mismo adaptador, aunque pueden efectuar diferentes operaciones sobre el mismo [24].

Una vez creado el componente, el mismo se puede referenciar a través de una actividad automática.

La herramienta permite crear y administrar formularios fácilmente desde el menú, eligiendo la opción “Administración de Formularios”. Cuando se crea un formulario, se debe especificar el conjunto de campos del mismo, a los cuales se les pueden asociar validaciones pre-definidas dependiendo del tipo de dato que se use. A su vez, es posible crear el formulario a partir de un HTML existente.

El conjunto de tipos de datos posibles es muy completo, inclusive se pueden asociar tablas a los campos para generar combos de datos y utilizar campos auto-incrementales como identificadores. Los campos de los formularios formarán partes de las variables globales del proceso, y se pueden referenciar desde cualquier actividad o flujo del mismo (por ejemplo, para orientar el control del flujo dependiendo de un determinado valor de alguna variable). Una vez creados, los formularios se pueden reutilizar en los diferentes procesos de la organización; notar que esto se configura desde el mismo formulario. Cuando el formulario es asociado a una actividad, en la misma se pueden configurar los comportamientos de cada campo (por ejemplo si un campo es editable, visible, etc). En la Figura 7.3 se muestra la administración de formularios y podemos ver que el campo seleccionado es un combo que está asociado a una tabla de la cual extrae sus valores.

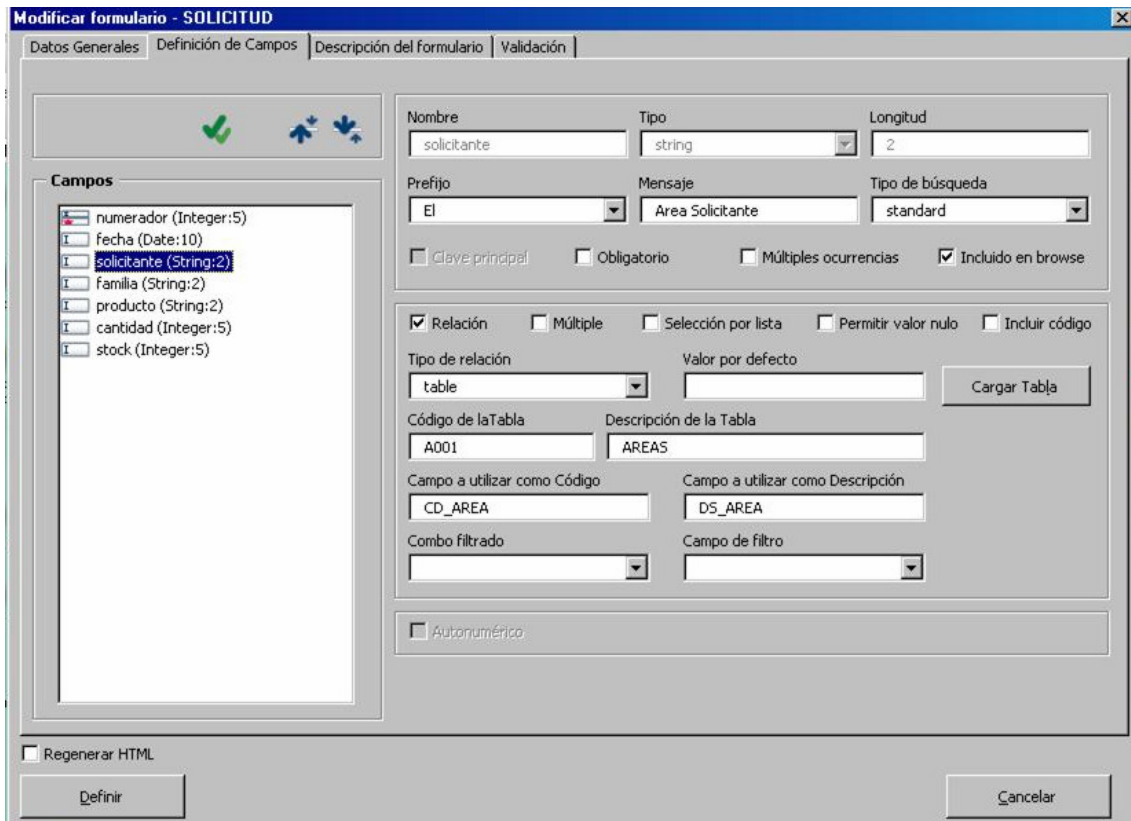


Figura 7 - 3: Pantalla de administración de formularios

Desde el menú principal de la herramienta se puede acceder a la opción “Definición de Métricas y Dimensiones”. Esta función tiene por objetivo permitir la definición de las Métricas y Dimensiones a partir de las cuales se creará el cubo (representación multidimensional del modelo de negocio) en el motor OLAP de TM1. A partir de esta definición se realizará el monitoreo del proceso, permitiendo al usuario de negocio analizar los aspectos más importantes del proceso de manera eficaz y dinámica mediante el uso de herramientas como por ejemplo MS Excel. Esto es útil para realizar un análisis de la información de negocio a nivel gerencial permitiendo tanto la optimización del proceso como la toma de decisiones para la empresa [27]. La Figura 7.4 muestra la interfase gráfica para realizar la definición de métricas y dimensiones de un proceso existente. A la izquierda tenemos el conjunto de variables que se pueden utilizar para definir el cubo. A la derecha se muestran las métricas y dimensiones definidas para el cubo actual.

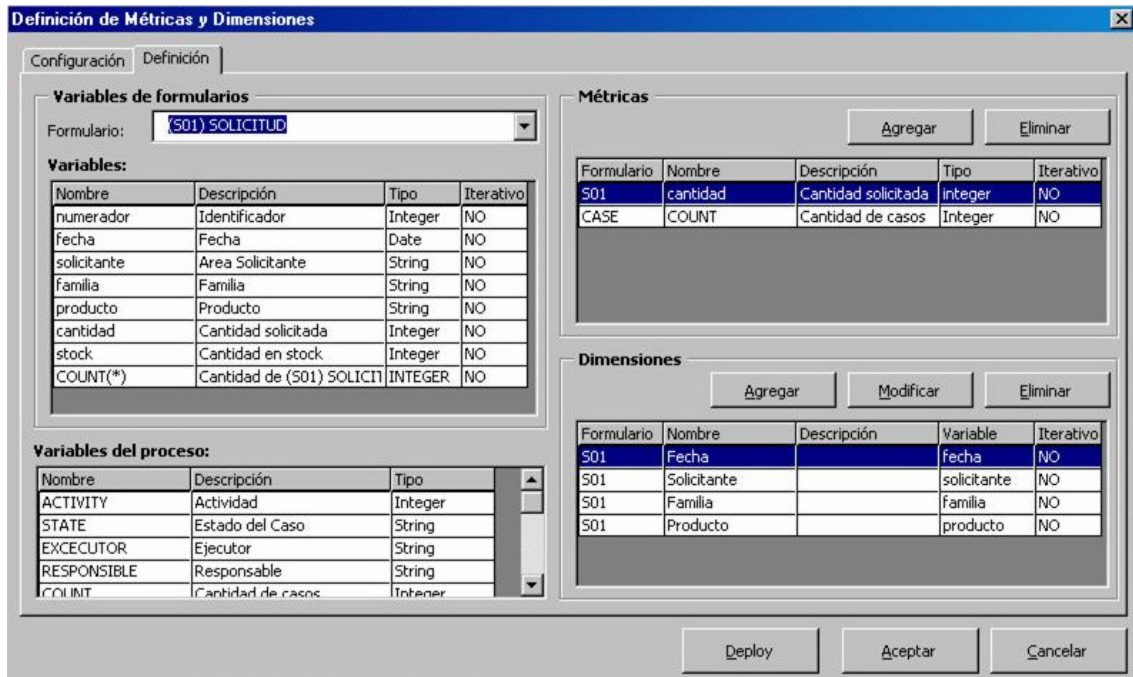


Figura 7 - 4: Definición de métricas y dimensiones de un proceso

Una vez definido el proceso, el mismo se puede exportar a un archivo XML con formato propietario en forma de *backup*.

Despliegue

La fase de despliegue se realiza desde la herramienta Microsoft Visio, eligiendo la opción "Salvar Modelo en el Motor". Para realizar esta operación el usuario logueado en la herramienta debe ser el responsable del proceso a desplegar, de lo contrario el motor dará error. La Figura 7-5 muestra la opción del menú de la plantilla Visio, para realizar el despliegue de un proceso de negocios.

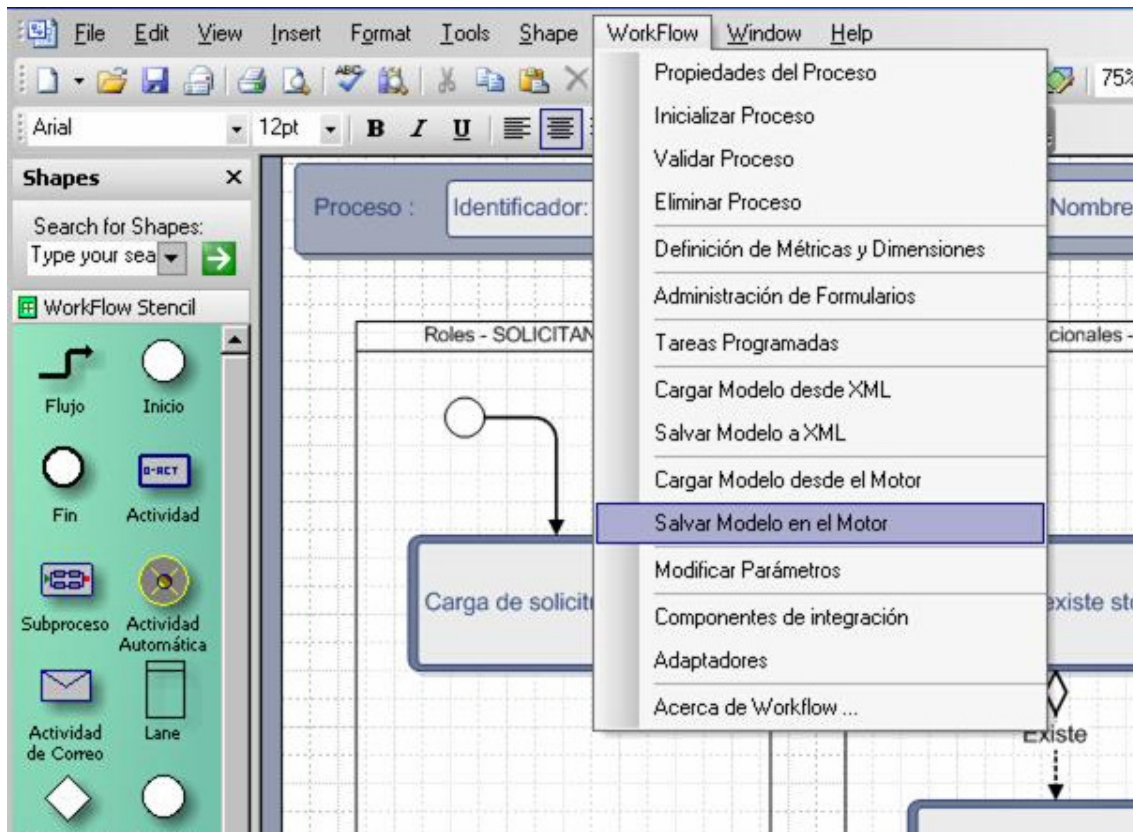


Figura 7 - 5: Despliegue de un proceso en ConsistGPA

Esta operación realiza una validación en el modelo, y si todo está correcto el proceso estará disponible para ser utilizado desde la consola ConsistGPA. Si existen errores, se mostrará una ventana con la lista de errores que ocurrieron. Además, podemos validar los procesos de negocios antes de ser desplegados con la opción “Validar Proceso”.

Ejecución, Monitoreo y Optimización

La fase de ejecución se realiza en la consola *web* ConsistGPA. La herramienta es muy restrictiva con respecto a la seguridad y al rol que posee el usuario que se identifica. Cuando accede un usuario, se acota el menú de opciones dependiendo de sus permisos y además este puede ejecutar solo los procesos y actividades que están bajo su responsabilidad (por ej bajo su Unidad Organizacional, Rol o Usuario).

El menú principal se divide en varios módulos distribuidos en diferentes solapas, entre las que se encuentran:

- **Casos:** permite acceder al conjunto de operaciones relacionadas con las instancias de los procesos de negocio. Entre las opciones más importantes se encuentran: Iniciar Caso, Búsqueda de Casos y Tareas por diferentes criterios, Eliminar Casos, Administración de documentos anexos a un caso, entre otras.
- **Formularios:** permite realizar la administración de los formularios. Entre las opciones principales tenemos: Administración, Generación y Baja de Formularios. La generación de formularios desde la consola es similar a la de la plantilla de Microsoft Visio.

- **Reportes:** permite acceder al módulo de Reportes que tiene las siguientes operaciones disponibles: Ejecución de Reportes, Consulta Múltiple de Repotes Ejecutados y Consulta de BAM. Notar que el usuario logueado solo podrá administrar los reportes que le hayan sido autorizados.
- **Configuración:** permite realizar todas las operaciones relacionadas a la configuración y administración de roles, usuarios, unidades organizacionales, permisos de acceso y perfiles. Además, se pueden administrar las tablas auxiliares que se utilizan para completar los *combo-boxes* de datos en los formularios. También se pueden visualizar los *logs* del sistema y administrar las tareas programadas.

En la Figura 7.6 podemos ver la consola *web* que trae la herramienta, junto con las solapas antes mencionadas.

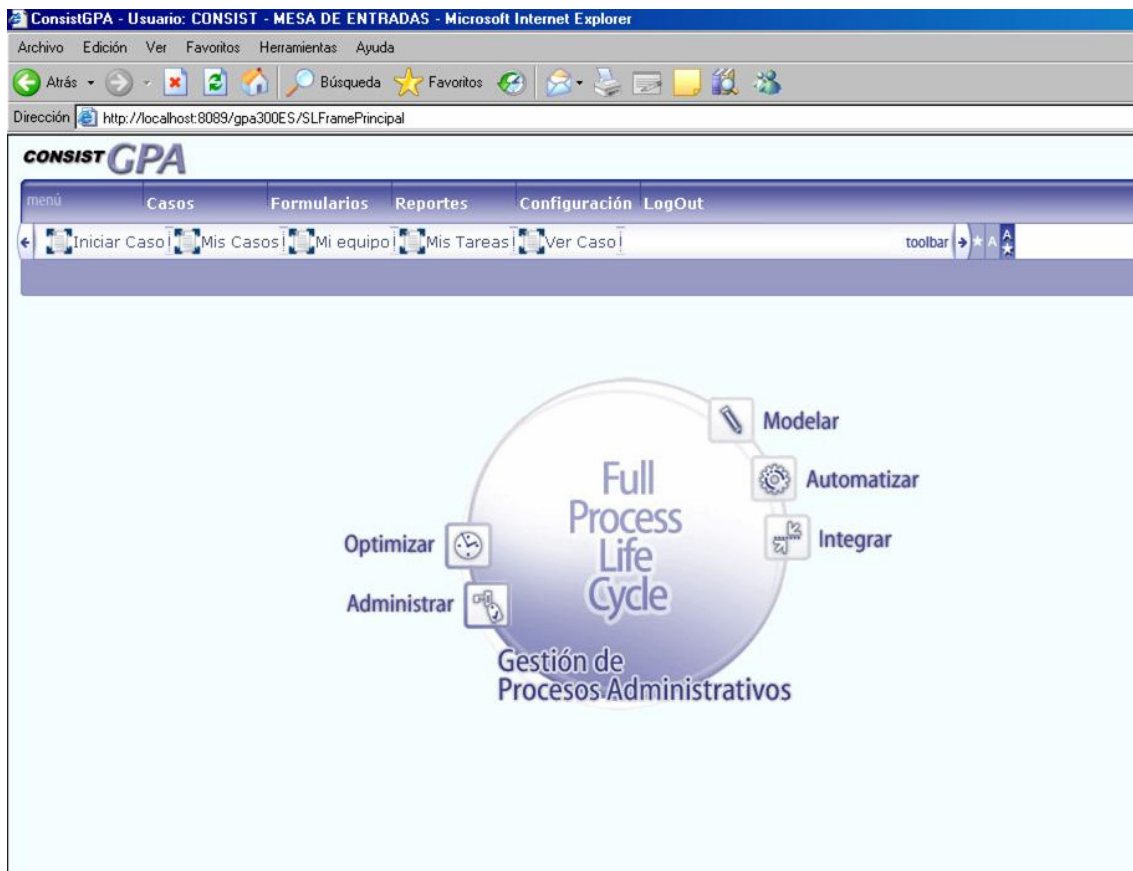


Figura 7 - 6: Consola *web* de ConsistGPA

El módulo de Casos es muy completo y fácil de utilizar. En el mismo se realiza el monitoreo de actividades y procesos para obtener información sobre el estado de los mismos y poder continuar con la ejecución del flujo. Contiene un amplio conjunto de buscadores por diferentes criterios para localizar los casos o tareas que están bajo la responsabilidad del usuario.

Desde el módulo de Reportes podemos acceder a la consulta de BAM. El módulo de BAM permite asociar un cubo al proceso, en cual se definen métricas y dimensiones en la etapa de diseño. Luego en dicho módulo, podemos visualizar los datos estadísticos

que se generaron a partir de las diferentes corridas del proceso. Existe la posibilidad de visualizar el cubo con diferentes tipos de diagramas (por ejemplo gráfico de barras, gráfico de torta) y grillas. Tenemos la posibilidad de guardar los resultados estadísticos en una vista que se almacenara en el servidor OLAP TM1. En la Figura 7-7 podemos visualizar el *software* OLAP TM1 donde se puede ver la estructura de los cubos y dimensiones.

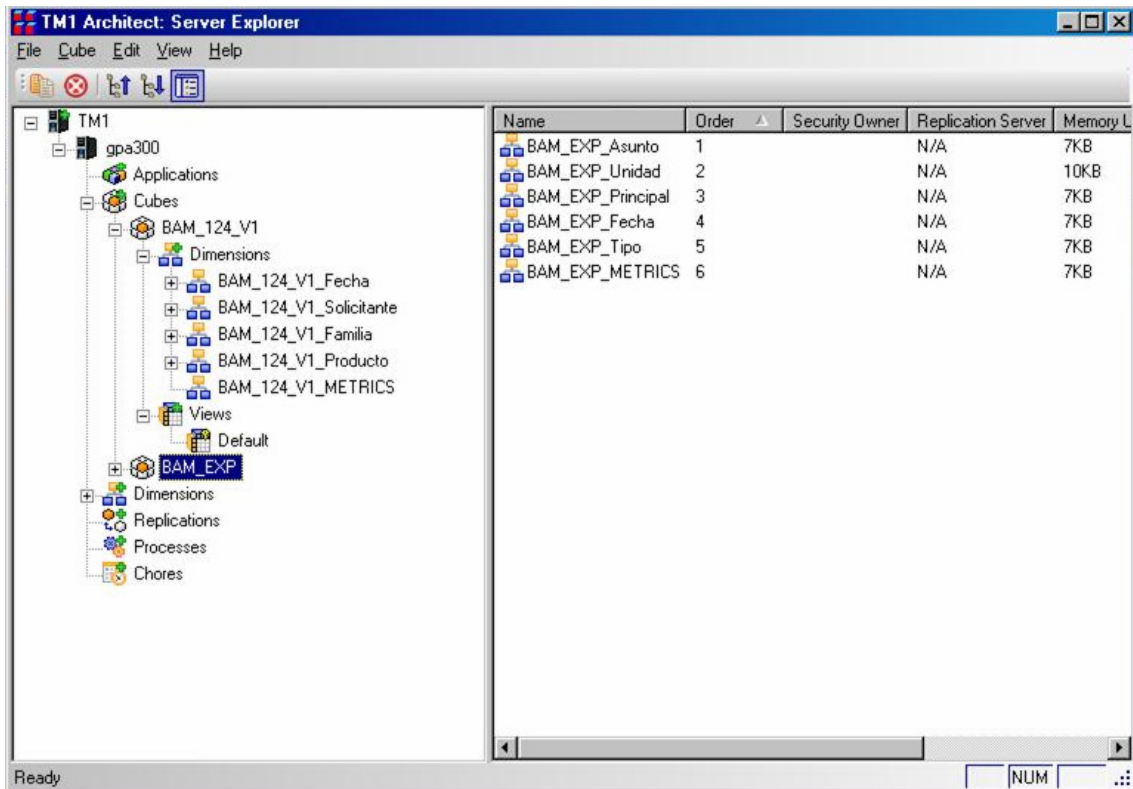


Figura 7 - 7: Servidor OLAP TM1

En el Servidor OLAP TM1 se realiza la administración de los cubos y dimensiones definidas. Este servidor es una aplicación externa a la consola *web*.

También el módulo de Reportes brinda la posibilidad de ejecutar reportes sobre los procesos con distintos criterios. Estos reportes se pueden almacenar en diferentes formatos de archivo en el disco (PDF, TXT, RTF).

La fase de optimización comienza cuando se tienen los resultados estadísticos de los procesos de negocio de la organización a través del modulo de BAM. Estos resultados sirven de soporte a la toma de decisiones para el sector gerencial de la organización, permitiendo descubrir falencias en los procesos analizados.

Conclusión General de la Suite

El conjunto de herramientas de esta suite permite cubrir el ciclo de vida básico del desarrollo de los procesos de negocios. Carece de herramientas que permitan realizar simulaciones en las etapas de diseño. En cuanto a la optimización, es posible realizarla en la etapa de ejecución a través de los resultados obtenidos en el módulo de BAM. A nuestro entender la optimización en esta etapa del ciclo de vida tiene un costo mucho

mayor que la optimización realizada en la etapa de diseño, ya que los errores o deficiencias del proceso se terminan detectando cuando este ya está implementado y desplegado, incrementando el costo de los mismos.

La utilización de la herramienta de diseño Microsoft Visio está destinada tanto para las personas del sector IT como para los analistas de negocios y gerentes. El analista de negocios realiza el diagrama del proceso que después será completado con los datos técnicos por el sector IT. Un aspecto positivo del uso de esta herramienta de modelado es que es una herramienta madura, testeada y conocida por los usuarios.

Con respecto a la consola de administración está destinada a la gente de negocios y a los responsables de las distintas áreas organizacionales de la empresa. Esta herramienta está pensada para desarrollo de soluciones BPM de mediana y gran escala. Si bien carece de un Bus de Servicios (ESB), la herramienta brinda la posibilidad de invocar servicios *web* a través de adaptadores y componentes.

Las herramientas están sujetas a dos licencias separadas, una para la GPA y otra para Microsoft Visio.

Los requerimientos de *hardware* no son exigentes, las herramientas corren en cualquier PC de hogar de última generación, permitiendo ejecutar ambas herramientas en el mismo sistema. La consola *web* corre bajo cualquier sistema operativo (cualquiera que pueda correr el Tomcat 5) y la herramienta de diseño Microsoft Visio solo corre bajo Windows.

La instalación y configuración de las herramientas es sencilla y su uso es muy intuitivo y fácil de aprender.

Con respecto a los estándares, solo adhiere al estándar de notación BPMN. Los XML generados para exportar los diagramas son propietarios, no adhiriendo al estándar XPDL.

La etapa de implementación se realiza a través de componentes u objetos de negocios propios de la plantilla Visio. No se dispone de un lenguaje de implementación para representar el flujo del proceso, se realiza configurando las propiedades y componentes asociados al proceso.

La herramienta soporta BAM dentro de la consola de ejecución. Este es un aspecto muy importante a la hora de la optimización y el monitoreo de los procesos. También, posee manejo de reportes por proceso para realizar el seguimiento de las actividades.

En lo relacionado a la documentación es acotada y carece de detalles técnicos y ejemplos. Solo tenemos disponibles lo que brinda Consist. En la *web* hay muy poca información adicional.

Capítulo 8 - Suite Oracle

Introducción

Oracle BPM permite el modelado, ejecución, optimización y administración de procesos de negocios. Incluye herramientas para las áreas de negocio e IT, profundizándose en ambas áreas, basándose en un circuito de ida y vuelta sin fisuras sobre una innovadora solución de un meta-modelo compartido.

Oracle BPM permite la integración de diferentes fuentes de información, como por ejemplo, aplicaciones legadas y empaquetadas, contenido estructurado y no estructurado, información basada en papel e ideas basadas en *business intelligence (BI)* [28].

Arquitectura

Oracle BPM está formada por los siguientes productos:

Oracle Business Process Analysis (BPA) Suite

Oracle BPA Suite es una herramienta sofisticada para el modelado y análisis de los modelos de procesos de negocios. Consta de los siguientes componentes:

- **Oracle *Business Process Architect*:** es el componente de modelado y simulación. Provee un entorno de modelado gráfico rico e intuitivo, que se ajusta a los usuarios de negocios para definir mapas de procesos y flujos detallados de procesos consistentes de actividades humanas o automatizadas, junto a un manejo de reglas que se extiende a través de los límites de la organización. Además del modelado de procesos de negocios, también soporta el modelado de datos (modelos UML), modelado organizacional, sistemas IT, análisis de impacto y generación de reportes. Puede ser utilizada para realizar análisis de procesos corriendo diferentes simulaciones en distintos escenarios. Estas simulaciones pueden ser utilizadas para ejecutar análisis de rendimiento, análisis de costo y tiempo de las actividades. Esta es una herramienta de diagnóstico para descubrir caminos críticos, cuellos de botellas en los recursos (tanto humanos como del propio sistema) y problemas estructurales de los procesos. A través de la simulación se puede determinar rápidamente la *performance* de un proceso ante ciertas condiciones hipotéticas. Es una herramienta versátil que puede ser utilizada por usuarios de negocios que tengan diferentes habilidades. La herramienta provee un conjunto predefinido de perspectivas (conocidas como filtros), pudiéndose fácilmente crear nuevas perspectivas.
- **Oracle *Business Process Publisher*:** es utilizado para publicar modelos de procesos de negocios en un portal de procesos y ofrece acceso seguro y basado en roles para el contenido del proceso. Fomenta la colaboración a través de varios miembros de un equipo y permite compartir y revisar los procesos de negocios a escala empresarial.

- **Oracle *Business Process Repository* y Oracle *Business Process Repository Server*:** el primero es utilizado para almacenar metadatos de procesos y el segundo para proveer acceso concurrente al repositorio de negocios. Promueve el desarrollo colaborativo de modelos de procesos, destacándose el acceso basado en roles, versionado y balanceo de carga.

Beneficios

- Facilita el entendimiento y documentación de procesos existentes.
- Promueve la reutilización de mejores prácticas y asegura que haya consistencia en las iniciativas de procesos a través de la empresa.
- Agrega dinamismo al diagrama de proceso ya que puede ser cambiado en respuesta a las condiciones de cambio del negocio. El diagrama de procesos es más que un dibujo estático.
- Permite analizar el impacto sobre cambios en los recursos y procesos existentes.
- Permite determinar los cuellos de botella y optimizar los procesos de negocios vía simulación.
- Incrementa la velocidad de desarrollo mediante la traducción automática de modelos de procesos de negocios en un modelo BPEL ejecutable.
- Colabora con diferentes interesados de negocios vía el portal de proceso.
- Elimina las brechas entre Negocios e IT a través de una integración ida y vuelta sin fisuras con el entorno de ejecución. El modelo de proceso en BPA Suite se convierte en la herramienta de comunicación entre el Negocio e IT.
- Satisface el variado conjunto de habilidades de los diferentes tipos de usuarios de negocios y puede ser configurada para presentar múltiples perspectivas a diferentes usuarios [28].

En la Figura 8-1 podemos ver los componentes de la arquitectura de la BPA de Oracle.

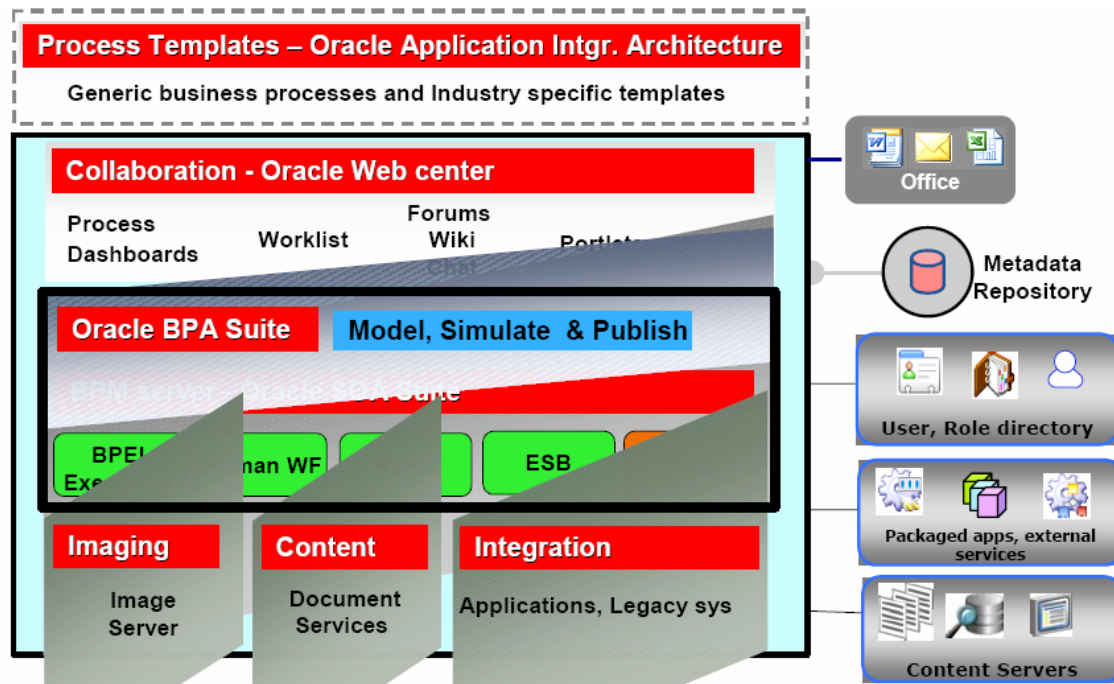


Figura 8 - 1: Arquitectura de la herramienta *business process architect* (BPA). Extraída de: “Oracle Business Process Architect, *Quick Start Guide Release 10.1.3.4*”. Oracle Corporation. Pag 8.

Oracle SOA Suite

Oracle SOA Suite es la plataforma de ejecución y monitoreo para la solución BPM de Oracle.

- **Oracle BPEL Process Designer (implementa y despliega):** es utilizada para desarrollar y desplegar ejecutables basados en los procesos BPEL. Está integrado con el Oracle *business process repository* y usa los modelos de procesos de negocios desarrollados con Oracle *business process architect* (BPA) como un punto de comienzo para generar artefactos BPEL. Es usado para desarrollar servicios y orquestarlos dentro de aplicaciones compuestas y procesos de negocios. La herramienta puede ser usada para testear los procesos BPEL y desplegar los mismos sobre la plataforma de ejecución (Oracle BPEL *process manager*).
- **Oracle BPEL Process Manager (ejecuta):** es el motor de ejecución de procesos. Provee una solución basada en estándares y fácil de usar para la creación, despliegado y gestión de procesos de negocios, bajo una arquitectura orientada a servicios. Su soporte nativo para los estándares como BPEL, XML, XSLT, XPATH, JMS, JCA y Servicios *web*, hace a esta una solución ideal para la creación integrada de procesos de negocios que son verdaderamente portables entre plataformas. El Oracle BPEL *process manager* soporta:

Interacciones Humanas: el componente de ejecución de procesos de Oracle provee soporte para la participación humana en los procesos fin-a-fin. Las características del flujo de trabajo humano incluye una aplicación de lista de trabajo, personalizable y basada en roles, fuera de la caja de patrones de flujo de

trabajo para ruteo sofisticado y asignación, notificaciones de apoyo extensas y políticas escalables, y fácil para usar modelado declarativo. Los servicios de tarea creados son reusables a través de múltiples procesos de negocios. La aplicación de lista de trabajo puede generar varios reportes para monitorear y optimizar la ejecución de trabajo. Los reportes pueden incluir ítems de trabajo que necesitan atención, actuales cargas de trabajo y distribución, tiempos de ciclo y productividad.

Los reportes incluyen datos basados en el perfil de usuario. Los reportes del supervisor incluyen datos para su personal, etc. Además del enrutamiento del modelado en el diagrama del proceso, Oracle BPM soporta patrones de enrutamiento adaptados para los flujos de trabajo humano a humano. Tales soportes incluyen la habilidad de los participantes para invitar a otros participantes, solicitar la información de otros participantes, delegar o reasignar tareas a otros participantes. Estas capacidades disponibles en tiempo de ejecución podrían ser restringidas en el modelo del proceso.

Interacciones de Contenido: Oracle BPM habilita interacciones estructuradas y no estructuradas. Los participantes podrían agregar comentarios y archivos adjuntos para una tarea, que luego, estarán disponibles para todos los otros participantes. Documentos en el Sistema de Gestión de Contenido podrían también ser usados para comunicación estructurada. Oracle BPM también trae soporte para foros de discusión y otros tipos de comunicaciones estructuradas como el *chat*, *voip*, etc.

Interacciones del Sistema: Oracle BPM soporta la integración de sistemas en procesos fin-a-fin. Provee una infraestructura *plug-and-play*¹⁷ basada en estándares para la exposición de sistemas como servicios y la orquestación de los mismos dentro de flujos de proceso. Puede ser usado para entregar aplicaciones compuestas (orquestación de *web services*, flujos de proceso J2EE) y aplicaciones de integración de datos a través de soluciones de conectividad +300 y con otras aplicaciones a través de conectores B2B.

- **Oracle Business Activity Monitoring (Monitor):** permite monitorear servicios y disparar eventos. Permite visualizar en tiempo real el estado de la empresa, procesos de negocios, personas y sistemas. El componente BAM le permite a los ejecutivos de negocios la habilidad para monitorear acuerdos de nivel de servicio (SLAs) a través de varios servicios y procesos de negocios de la empresa, permitiéndole correlacionar indicadores clave de rendimiento (KPIs)¹⁸ para el proceso de negocio actual, y lo más importante es que le permite cambiar los procesos de negocios mucho mas rápidamente y eficientemente para tomar acciones correctivas si el entorno del negocio cambia o para mejorar la eficiencia. Los usuarios de negocios pueden rápidamente crear *dashboards* sumamente efectivos y reportes que muestran medidas de negocios críticas y KPIs que se actualizan en tiempo real con capacidad de profundizar dentro de

¹⁷ Plug-and-play (conocida también por su abreviatura PnP) es la tecnología que permite a un dispositivo informático ser conectado a un ordenador sin tener que configurar (mediante jumpers o software específico (no controladores) proporcionado por el fabricante) ni proporcionar parámetros a sus controladores. Extraído de: <http://es.wikipedia.org/wiki/Plug-and-play>

¹⁸ KPI, del inglés Key Performance Indicators, o Indicadores Clave de Desempeño, miden el nivel del desempeño de un proceso, enfocándose en el "como" e indicando que tan buenos son los procesos, de forma que se pueda alcanzar el objetivo fijado. Extraído de: http://en.wikipedia.org/wiki/Key_performance_indicator

información detallada. El usuario puede también modelar condiciones de alerta y reglas que podrían ser usadas para advertir cuando las ciertas condiciones se cumplen.

- **Oracle Business Rules:** se trata de una herramienta de diseño de reglas y un motor de inferencia para capturar políticas de negocios. El Componente de Reglas de Negocios de la plataforma Oracle BPM es un motor de reglas basado en reglas de inferencia de alta performance y habilita las reglas de negocios para ser explícitamente especificadas y gestionadas por usuarios de negocios en una manera declarativa (declaraciones versus lógica procedural) [28].

El gráfico de la Figura 8-2 visualiza todos los componentes, previamente explicados, de la arquitectura de la SOA Suite de Oracle.

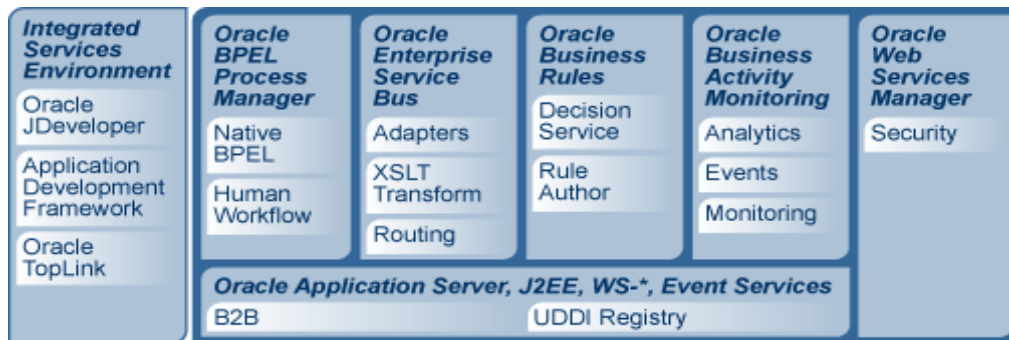


Figura 8 - 2: Arquitectura de la herramienta SOA Suite. Extraída de: “Oracle SOA Suite Quick Start Guide 10g (10.1.3.1.0)”. Oracle Corporation. Pag 8, Figure 1–1.

Requerimientos de Hardware y Software

Oracle Business Process Architect

Requerimientos Mínimos de Hardware

Procesador Intel Pentium III 450 Mhz, para la instalación cliente sin sistema de bases de datos local. Se necesita 256 MB RAM, para la instalación cliente con sistema de bases de datos local, disco duro de 375 MB, tarjeta gráfica SVGA con resolución de pantalla de 800 x 600 y 256 colores, ancho de banda de 56 k, son posibles de utilizar anchos de banda menores, sin embargo, esto puede resultar en una performance reducida.

Requerimientos Recomendados de Hardware

Procesador Intel Pentium IV 2.4 Ghz, 1 GB RAM para instalación cliente con y sin sistema de base de datos local, tarjeta gráfica con resolución de 1024 x 768 y al menos 256 colores y un ancho de banda de 128 k.

Sistema Operativo

Windows XP Profesional.

Oracle SOA Suite

Versiones JRE y JDK

SUN Microsystems garantiza compatibilidad entre los niveles de parche JRE. Por lo tanto, las aprobaciones Oracle BPA Suite para JREs son válidas para todos los niveles de parche de una versión.

Si JRE 1.5.0_02 y sus actualizaciones son aprobados para un producto Oracle BPA Suite, la aprobación será válida para JRE 1.5.0_02 y JRE 1.5.0_05 y cualquier nivel subsecuente de actualización. Si se tiene instalado *Oracle Business Repository*, se utilizará automáticamente una versión ya instalada de la JDK.

Servidores de Base de Datos

Siempre se debe observar la versión provista por el fabricante de la base de datos ya que podría ser posible que solo un nivel de parche específico de la base de datos sea aprobado para operar con el servidor que se esta usando.

Los *setup* de 32 y 64 bit están disponibles para Oracle *business process repository*. Ejecutando el *setup* de 64 bits en una computadora que tiene una edición de Windows 2003 de 64 bits, las opciones de sistema de 64 bits podrán ser usadas al máximo.

Si se corre el *setup* de 32 bits bajo la edición de Windows 2003 de 64 bits, un mensaje será mostrado notificando que existe un programa de instalación separado para este sistema. Sin embargo, se puede decidir correr la instalación de 32 bits. En este caso, las opciones de 64 bits no serán utilizadas, lo que resultará en una performance reducida.

Mientras el *setup* de 32 bits puede ser corrido bajo la edición de Windows 2003 de 64 bits, no es posible correr el sistema de 64 bits sobre un sistema de 32 bits.

Sistema Operativo

- Windows *server* 2000 *standard edition* + *service pack* 4.
- Windows *server* 2003 *standard edition* + *service pack* 1 [30].

Requerimientos Mínimos de *Hardware*

Procesador Intel Pentium IV 2.4 Ghz, 512 MB RAM, Tarjeta gráfica SVGA con resolución de pantalla de 800 x 600 y 256 colores.

Requerimientos Recomendados de *Hardware*

Procesador Intel Pentium IV 3.0 Ghz, 2 GB RAM,

Tarjeta gráfica con resolución de 1024 x 768 con al menos 256 colores. Los requerimientos de procesador dependen del número de usuarios y el tamaño de la base de datos (1 GB RAM para 50 usuarios).

Si se utiliza Oracle BPA Suite *converter*, se necesita un adicional de 256 MB RAM y 512 MB de espacio de disco duro para convertir una base de datos.

Se necesita una red de alta velocidad (100 Mbits) entre el servidor de base de datos y Oracle *business process repository*.

Espacio de Disco

El espacio de disco dependerá del tipo de instalación elegida:

- **Instalación Básica:** Servidor J2EE y Oracle SOA Suite: 800 MB
- **Instalación Avanzada:** Servidor J2EE, Servidor *web* y Oracle SOA Suite: 875 MB
- **Instalación Avanzada:** Servidor J2EE y Servidor *web*: 500 MB
- **Instalación Avanzada:** Servidor J2EE: 475 MB
- **Instalación Avanzada:** Servidor *web*: 325 MB

Navegador *web*

Microsoft Internet *Explorer* versión 6.0 + *service pack 2* y Mozilla Firefox versión 1.5.

Oracle Application Server (SOA)

Sistema Operativo

Los sistemas operativos soportados por esta herramienta son:

- Windows 2000 con *service pack 3* o superior.
- Windows *server* 2003 (32 bits) con *service pack 1* o superior.
- Windows *server* 2003 *release 2*.
- Windows XP Profesional con *service pack 2* o superior.

Los sistemas operativos de 64 bits soportados son:

- Windows XP Profesional *x64 edition*.
- Windows *server* 2003, *standard x64 edition*.
- Windows *server* 2003, *enterprise x64 edition*.
- Windows *server* 2003, *datacenter x64 edition*.

Capacidad de Red

La herramienta no requiere conexión a red para poder funcionar.

IP

Se puede instalar la herramienta en una computadora que utiliza IP estática o IP basada en DHCP.

Hostname

Es importante asegurarse que el nombre del servidor no tenga más de 255 caracteres.

Procesador

Recomendado un procesador Intel Pentium de 300 Mhz o superior.

Memoria

512 MB de memoria RAM es suficiente para instalar y correr Oracle *application server*. Para sitios de producción, se debe configurar al menos 1 GB de memoria física. Para sitios con mucho tráfico de red, aumentar la cantidad de memoria mejora la *performance*.

Para aplicaciones Java se debe incrementar la memoria *heap* máxima asignada para procesos OC4J o configurar procesos OC4J adicionales para utilizar esta memoria.

Sistema de Archivos

Se recomienda NTFS por sobre FAT32 o FAT porque este incluye características de seguridad.

Monitor

Pantalla de 256 colores [31].

Instalación y Configuración de la BPA

La versión a instalar es la Oracle BPA Suite 10.1.3.4 *developer version*.

Instalación de Oracle Business Process Architect (Windows)

1. Asegurarse que los requerimientos mínimos del sistema se cumplen para todos los productos y las funcionalidades.
2. Ejecutar el archivo Setup.exe desde el CD de instalación o desde el directorio donde está descomprimida la BPA, en caso de haberla descargado en un zip.
3. Se acepta el acuerdo de licencia.
4. Se elige el directorio destino donde será instalado el producto. Si ya está instalado, este paso será omitido y se usará el directorio donde ha sido instalada previamente.

5. Se debe seleccionar que base de datos Oracle se va a utilizar. Las opciones son Oracle Lite o *express edition*. Tener en cuenta que la base de datos Lite consume menos recursos que la *express edition*
6. Elegir el idioma de la instalación. Luego, comienza la instalación [32].

Instalación de JDeveloper Version 10.1.3.2 (Windows)

1. Descargar el producto desde la siguiente url:
<http://www.oracle.com/technology/products/jdev/index.html>
2. Se debe tener instalada una jdk 1.5.0 para poder correr este producto.
3. Una vez descargado el zip, descomprimirlo en una carpeta a elección.
4. Ejecutar el archivo JDeveloper.exe de jdeveloper para arrancar el producto.
5. Al ejecutar el JDeveloper.exe, muestra una ventana donde nos pide el camino completo donde se encuentra el java.exe de la jdk 1.5.0.

Instalación y Configuración de la SOA Suite

La versión a instalar es la Oracle *Application Server* SOA Suite 10.1.3.1.0.

Instalación de Oracle Application Server SOA Suite (Windows)

1. Asegurarse que los requerimientos mínimos del sistema se cumplen.
2. Ejecutar el archivo Setup.exe desde el CD de instalación o desde el directorio donde está descomprimida la SOA Suite, en caso de haberla descargado en un zip.

En Figura 8-3 se muestra la pantalla de instalación.

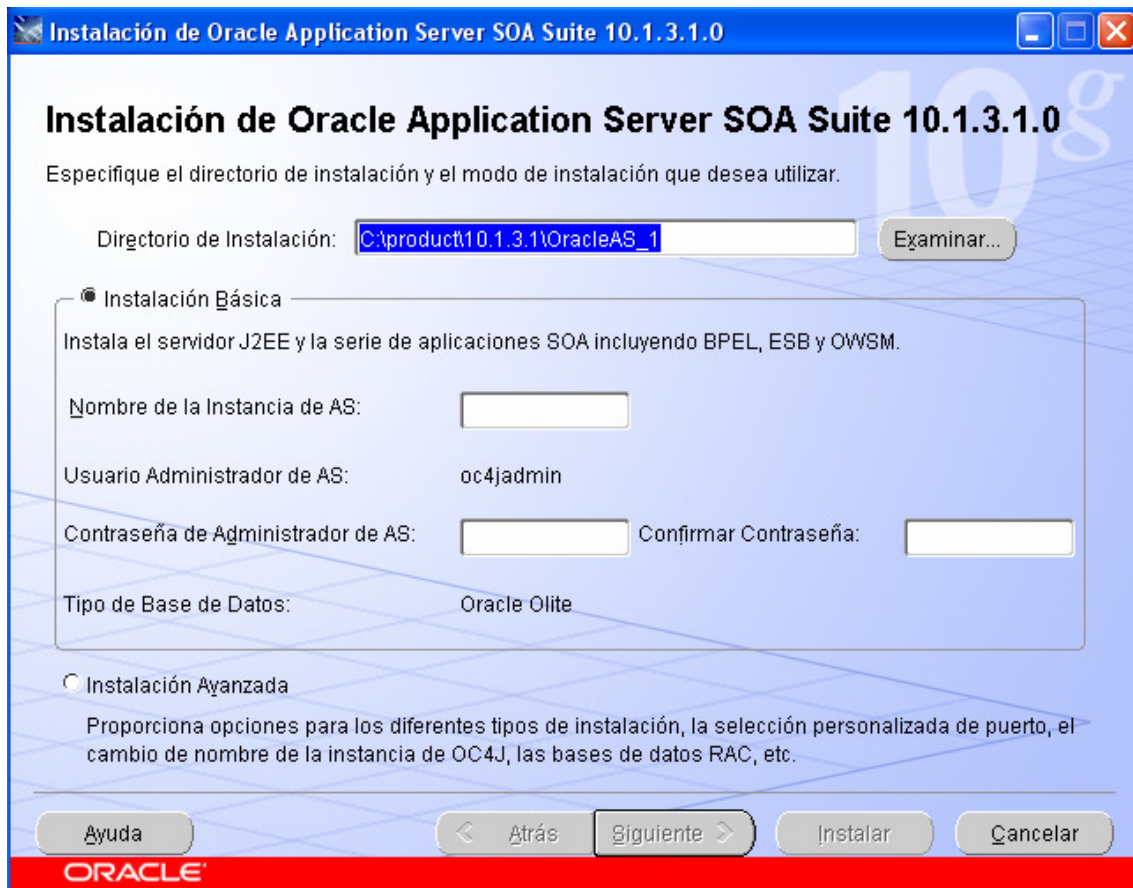


Figura 8 - 3: Pantalla de instalación del Oracle Application Server SOA Suite

- **Directorio de Instalación:** ingresar el directorio donde se desea instalar el Oracle *application server*.
 - **Seleccionar Instalación Básica.**
 - **Nombre de la instancia AS:** el nombre que identifica esta instancia del Oracle *application server*. Si se tiene más de una instancia del Oracle *application server* en el mismo servidor, las instancias deben tener un nombre único.
 - **Usuario Administrador de AS:** el nombre del usuario administrador para la instancia Oracle *application server* está configurado como oc4jadmin y no puede ser cambiado. Para gestionar la instancia del Oracle *application server* usando Oracle *enterprise manager* 10g, hay que acceder con el usuario oc4jadmin.
 - **Contraseña de Administrador AS y Confirmar Contraseña:** ingresar la clave para el usuario oc4jadmin.
 - **Tipo de Base de Datos:** la Oracle *database* Lite (Olite) se instala junto con la instalación básica. Si se desea utilizar una base de datos Oracle existente, se debe realizar una instalación avanzada.
3. Se ingresan los datos de la pantalla anterior y se presiona el botón Instalar. En este momento se inicia la instalación del producto.

4. Una vez finalizada la instalación se muestra una pantalla con la información del resultado de la instalación y algunos links para poder acceder a la página de bienvenida.
5. Podemos acceder a la página de bienvenida mediante la siguiente URL en el navegador *web*: `http://localhost:8888/`.
6. Por defecto, todos los componentes del Oracle application server instalados tienen la misma clave que la ingresada en el momento de la instalación. Para reforzar la seguridad se deberían cambiar las claves de dichos componentes [31].

Conexión de la SOA Suite con JDeveloper

Para realizar la conexión a los servidores de la SOA Suite desde el JDeveloper se tienen que seguir los siguientes pasos:

Establecer una conexión con la base de datos Oracle Lite desde el JDeveloper

Utilizando el JDeveloper necesitamos asegurarnos que se puede establecer la conexión a la base de datos Oracle Lite [33]. Para hacer esto se deben seguir los siguientes pasos:

1. En el pestaña **conexiones** del JDeveloper, crear una nueva conexión en la carpeta *database* (clic derecho en la carpeta, elegir la opción *new database connection*).
2. En la página de bienvenida de dicha conexión, presionar **next**. Luego se va a mostrar la página de tipo de conexión.
3. En el nombre de la conexión ingresar Olite y seleccionar Oracle Lite en el combo de tipo de conexión.
4. Presionar **next**. Luego se muestra la página de autenticación de la conexión.
5. Especificar los siguientes valores y presionar **next**.
 - *username: system*
 - *password: any*
6. En el siguiente paso se especifican los datos propios de la conexión.
7. Especificar el controlador del dispositivo correcto para la conexión. Presionando el botón **browse** accedemos a la lista de librerías disponibles.
8. Agregar una nueva librería para este controlador del dispositivo, presionando el botón **add entry** buscamos el jar de la librería que se encuentra en el directorio de instalación de la SOA Suite, más precisamente en `Oracle_home/integration/esb/lib/olite40.jar`. Luego elegimos el botón **aceptar**.
9. Ingresar el nombre del servidor, puerto, y SID con la siguiente información:

- a. *host name*: localhost
 - b. JDBC *port*: 1531
 - c. SID: OraESB
10. Presionar el botón **next** y luego aparecerá la ventana de testeo de conexión. Presionando el botón **test connection**, si la conexión es correcta aparecerá el mensaje **success!** Luego presionar el botón **finish** [34].

Establecer la conexión con el *application server* desde el JDeveloper

Una vez que la conexión con la base de datos Oracle Lite fue exitosa, debemos establecer la conexión con el *application server*.

1. Presionar el botón derecho del *mouse* en **application server** y elegir **new connection**.
2. Presionar el botón **next** y completar los campos *username/password* con el *login* del usuario oc4jadmin. Seleccionar el *checkbox deploy password*.
3. Presionar el botón **next** y completar:
 - *host name*: localhost
 - RMI *port*: 12401
4. Presionar **next** y testear la conexión. Si fue exitosa nos mostrará el mensaje **success!** [33].

Establecer la conexión con el BPEL *process manager server* desde el JDeveloper

1. Presionar el botón derecho del *mouse* en **integration server** y seleccionar **new connection**.
2. Ingresar un nombre para la conexión y presionar el botón **next**.
3. Elegir la conexión del *application server* creado en la sección anterior, desde el combo **application Server**. En el campo **port number**, escribir el puerto HTTP utilizado por la SOA Suite.
4. Presionar el botón **next**, y testear la conexión. Esto puede tardar varios segundos. Si la conexión fue exitosa nos mostrará en la pantalla con todos los servidores en OK [33].

Soporte de Tecnología Java

La implementación de los procesos de negocio se realiza usando el IDE JDeveloper. El lenguaje de ejecución de estos procesos es BPEL, por lo tanto se utilizan *web services* para proveer de funcionalidad a los mismos. La IDE JDeveloper cuenta con un amplio

soporte y un conjunto de herramientas para la implementación de estos *web services* en el lenguaje Java.

Análisis de la Business Process Architect (BPA) y fases del ciclo de vida BPM soportadas

Componentes de la Herramienta

La herramienta está compuesta por múltiples módulos que tienen diferentes propósitos. A continuación se describen los componentes de la herramienta BPA:

- **Explorador:** permite crear, restaurar, iniciar y detener el conjunto de Bases de Datos. En esta suite los proyectos se denominan Bases de Datos. Una vez iniciada una Base de Datos en este módulo podemos ver toda la estructura de carpetas, diagramas y objetos que componen dicho proyecto.
- **Diseñador:** permite crear y modificar los modelos asociados a cada Base de Datos. Este diseñador soporta una gran cantidad de tipos de diagramas (BPMN, UML, etc).
- **Editor de Matrices:** permite mostrar las relaciones que existen entre una gran variedad de objetos dentro de una matriz, sin importar el tipo del modelo.
- **Administración:** permite administrar las bases de datos de la BPA Suite, y modificar o agregar ítems en la configuración del Oracle *business process repository*. Además, este módulo ayuda a simplificar la administración de bases de datos, usuarios y grupos de usuarios.
- **Editor de Comandos:** permite escribir *scripts*, por ejemplo para evaluación. Se requiere un conocimiento previo de los lenguajes de programación JavaScript y Oracle BPA Suite *script* (basado en JavaScript con objetos y métodos adicionales).
- **Simulación:** permite realizar simulaciones de procesos obteniendo datos estadísticos basados en diagramas, para un posterior análisis y optimización de los mismos [35].

Modelado y Diseño

El diseñador de procesos respeta el estándar BPMN para modelado de los procesos de negocios. Tiene una paleta de objetos predefinidos que incluye todos los componentes BPMN, dando la posibilidad de agregar más objetos. Además de contener los objetos básicos del estándar, trae un conjunto de objetos que permiten modelar excepciones, contadores de tiempo, eventos y transacciones que enriquecen el modelo.

El diseño de los procesos se realiza de una manera muy intuitiva y flexible. Permite un agregar, modificar y eliminar los objetos del diagrama, en forma muy simple. Cada objeto del diagrama posee un gran conjunto de propiedades y atributos para configurar dicho objeto. Simplemente haciendo *click* con el botón derecho del *mouse*, sobre un

objeto, se puede acceder al menú de propiedades que nos muestra las propiedades y atributos divididos por categorías.

El diseñador de procesos permite reacomodar elementos u objetos del diagrama, manteniendo el orden con respecto a los otros objetos. Permite también la selección de objetos del diagrama por diferentes criterios (por tipo de elemento, por clase, etc).

Cuando se tienen elementos compuestos o subprocesos dentro de otros procesos, el diseñador permite acceder fácilmente al elemento interno mediante un *link* gráfico. Esto último facilita la navegación dentro de diagramas complejos. La Figura 8-4 muestra un diagrama modelado en la BPA. Se ve claramente que respeta la notación estándar de BPMN.

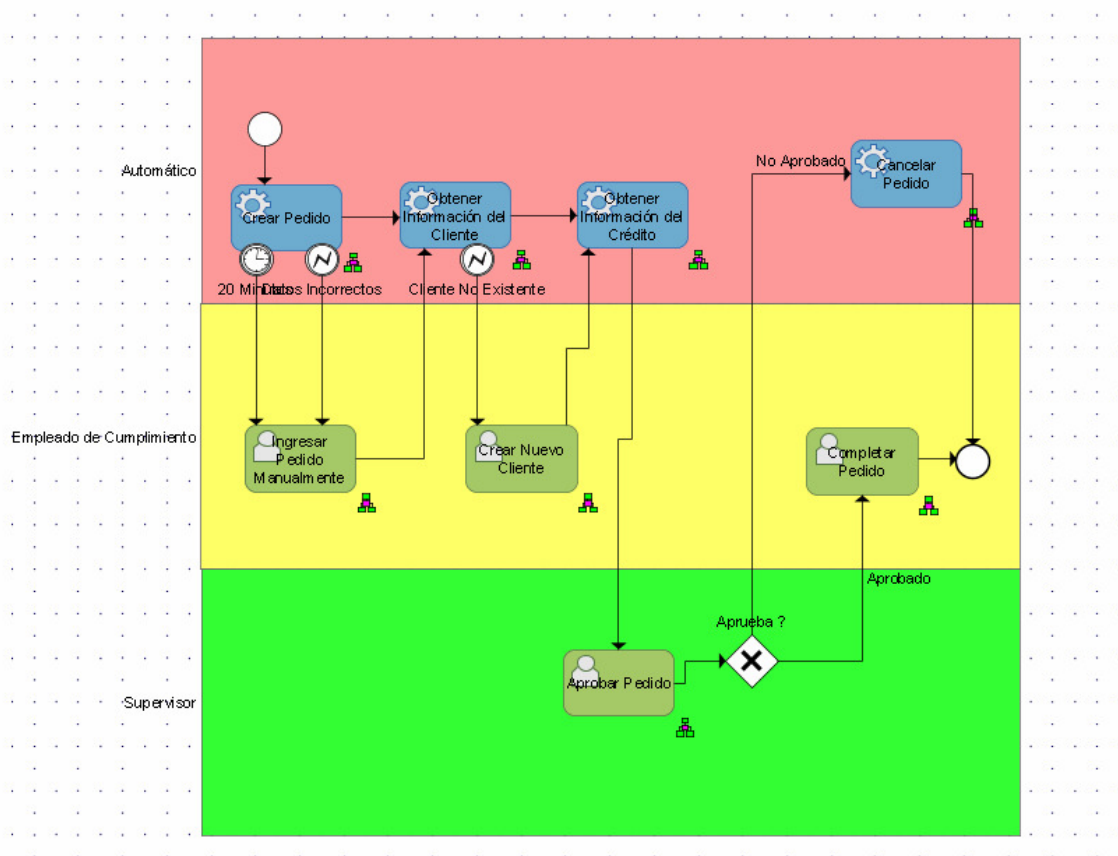


Figura 8 - 4: Diagrama de un proceso en BPMN usando BPA

A simple vista se puede comprobar que los diagramas son intuitivos y descriptivos. Cada elemento creado en cada uno de los diagramas tiene una representación mediante un objeto que se guarda en el módulo explorador.

Algunas falencias de *refactoring* encontradas en el diseñador, son:

1) respecto a la eliminación de objetos del diagrama, cuando se elimina algún objeto, su correspondiente en el explorador no se elimina. Esto trae como consecuencia que se almacenen varios objetos que no están siendo utilizados en el diagrama. Para evitar esto hay que tener la precaución de eliminar también el objeto dentro del explorador para

que quede consistente el diagrama con respecto a los objetos almacenados en el proyecto.

2) respecto al copiar y pegar objetos entre diagramas o en un mismo diagrama, ya que el diseñador crea una referencia a dicho objeto y no una nueva instancia. Si la intención es crear un nuevo objeto, lo que se debe hacer es tomarlo desde la paleta de objetos y arrastrarlo hacia el diagrama.

Respecto a la asociación de los roles responsables de una actividad humana, cuando se quiere configurar este tipo de asociación, la herramienta no brinda la posibilidad de elegir entre los roles existentes en el proyecto y obliga a crear la referencia manualmente.

Simulación

En esta etapa se puede realizar la simulación de los procesos creados. Esto permite entre otras cosas evaluar el rendimiento de los procesos y hallar cuellos de botella, puntos débiles, cantidad de recursos requeridos, y luego realizar la optimización de los mismos.

La BPA tiene un módulo para realizar las simulaciones, que permite obtener estadísticas sobre el costo, tiempos de ejecución y uso de recursos, de cada una de las actividades que componen el diagrama de proceso analizado.

En el módulo diseñador se editan las propiedades de cada actividad para así configurar la simulación. Por cada objeto del diagrama se tiene un conjunto amplio de propiedades que se tendrán en cuenta a la hora de la simulación. Por ejemplo, en los eventos de inicio se puede configurar la cantidad de instancias del proceso a ejecutar; en las actividades podemos cambiar la duración y los costos (total de materiales, de energía, de personas, etc); en las decisiones (*gateway*) se puede configurar la probabilidad de tomar un camino u otro; para cada rol se puede configurar el número de personas encargadas. También tenemos parámetros que se configuran para poder ejecutar la simulación, como por ejemplo el tiempo de espera entre la ejecución de un subproceso y otro, la duración total de la simulación y los tipos de estadísticas que se quieren como resultado de la simulación.

Cuando se ejecuta la simulación, el motor de simulación realiza un chequeo sintáctico y semántico del proceso objetivo, lo que permite descubrir errores en el diseño, por ejemplo, un camino de ejecución no llega al evento de finalización, esto es un error que el motor obliga a corregir para poder continuar con la simulación. Generalmente, los mensajes de error arrojados por el motor, son muy descriptivos y vienen acompañados de una solución posible.

La simulación brinda la posibilidad de mostrar una animación en la que se visualizan claramente las actividades por las que atraviesa el flujo de la simulación, así como también la cantidad de veces que pasa por cada actividad. La simulación animada es opcional, ya que consume muchos recursos y hace más lenta la simulación.

Una vez finalizada la simulación se muestran todas las estadísticas en diferentes solapas organizadas de forma tabular. Seleccionando los valores de alguna columna y haciendo *click* sobre el botón derecho del *mouse* podemos generar diferentes gráficos (barras, circular, líneas) para visualizar los resultados de una forma más clara. También, se

pueden ordenar los valores de las columnas en forma ascendente y descendente. En la Figura 8-5 podemos ver el resultado de una simulación de un proceso desde la BPA. En la parte derecha se muestra un gráfico de barras calculado a partir de la columna Activaciones.

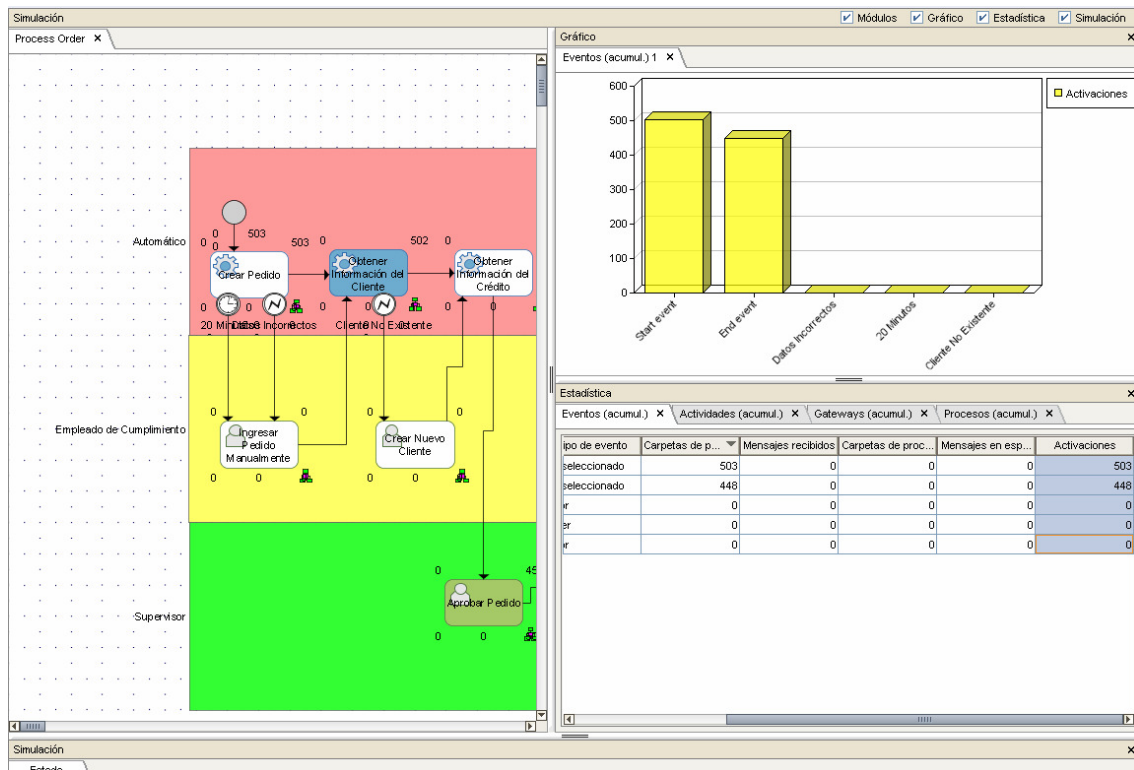


Figura 8 - 5: Resultado de la simulación de un proceso usando la herramienta BPA

Consideramos que este módulo es muy útil en la etapa de análisis y diseño de procesos ya que está orientado a gerentes y analistas y permite mediante las estadísticas obtenidas, optimizar y depurar el diagrama del proceso. Los resultados de esta etapa permiten retroalimentar la etapa de análisis y diseño, obteniendo un ciclo de mejora continua, hasta que se obtiene una versión del proceso optimizada y lista para pasar al sector IT.

Transformación del Modelo a BPEL

Una vez realizados y simulados los diagramas tenemos un proceso abstracto que no se puede ejecutar. Por lo tanto, el siguiente paso es convertir los diagramas en procesos *blueprint* (meta-modelo compartido entre las herramientas de análisis y las herramientas de desarrollo). Esta tarea se debe realizar por cada diagrama diseñado. Como resultado cada meta-modelo generado será consumido por el sector IT para transformarlo en un proceso ejecutable BPEL [28]. Esta transformación se realiza desde el módulo diseñador de la BPA: se elige un diagrama particular, luego desde el menú SOA tenemos la opción de **compartir el modelo con IT** y de esta manera es posible realizar los chequeos semánticos necesarios para detectar errores en el diagrama. Luego, mediante un menú asistente se configura dicha transformación. La Figura 8-6 muestra la opción del menú SOA de la BPA que permite **compartir el diseño con IT** para así pasar un proceso desde esta herramienta hacia el jDeveloper.

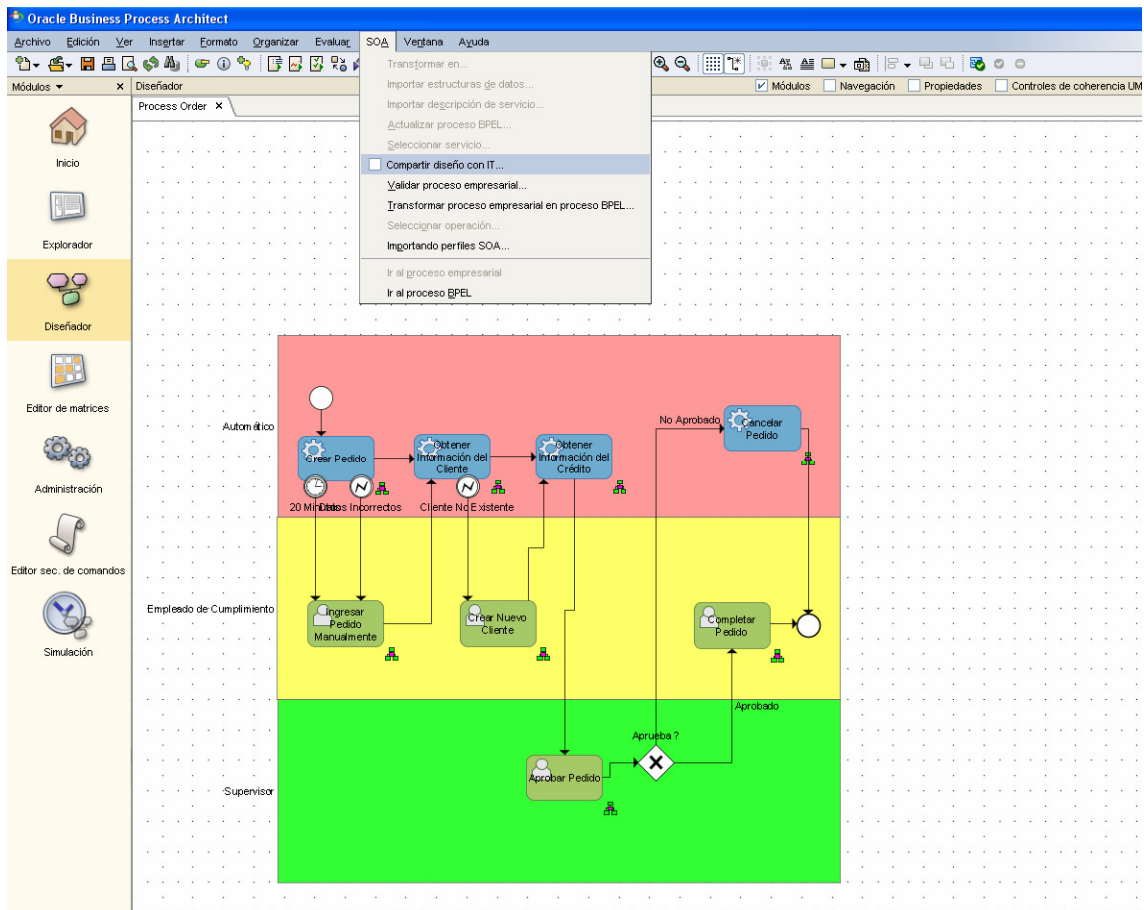


Figura 8 - 6: Procedimiento para compartir un proceso modelado en la BPA con el grupo de IT.

Al finalizar la transformación, la herramienta nos abre un editor de texto con el resultado de la operación y las advertencias o errores que ocurrieron. El resultado de la transformación se guarda en el proyecto y se puede ver la estructura generada desde el explorador. Asimismo desde el diseñador podemos acceder a los procesos *blueprint* generados.

Generar los Modelos BPEL

Para realizar esta etapa se utiliza el IDE JDeveloper. Previamente se debe configurar esta herramienta para que nos permita conectarnos al repositorio de la BPA. Eso se lleva a cabo agregando el *plugin* que se encuentra en la carpeta *addons* del directorio donde está instalada la BPA.

Dentro del JDeveloper, en el menú *help -> check for updates*, seleccionamos la opción *install from local file* y luego elegimos la ruta donde se encuentra el archivo **pcbpel_bundle.zip**. Esto agregará el *plugin* necesario. Algo importante a tener en cuenta es que existen nuevas versiones de JDeveloper que tienen ya instalado dicho *plugin*, en ese caso no es necesario realizar la configuración antes mencionada.

Los desarrolladores IT pueden conectarse al *business repository* desde el JDeveloper (en este caso es la herramienta para Diseñar los Procesos BPEL) y acceder a los modelos de los procesos de negocios compartidos previamente. El modelo del proceso

de negocios es utilizado como punto de partida para generar el modelo de implementación BPEL.

Para crear nuevos procesos BPEL basados en los *blueprint* previamente compartidos, desde el JDeveloper se debe crear una aplicación y luego se deben crear proyectos BPEL por cada *blueprint* compartido. Para hacer esto hay que realizar una conexión al BPA *repository server*. Es importante tener en cuenta que a la hora de realizar la conexión, el idioma de la misma debe ser el mismo que el utilizado en la BPA. En caso contrario, la importación del proceso será inconsistente y los diagramas no contendrán los nombres para cada componente. Tras realizar la importación de los *blueprint*, en el JDeveloper se obtienen dos vistas de cada proceso: la **Vista de Negocios** (también llamada “Vista *blueprint*”) y la **Vista BPEL**. La **Vista de Negocios**, muestra el modelo BPMN del proceso; los atributos BPMN definidos en la BPA Suite son incorporados como información que puede ser utilizada por IT. Dentro de estos diagramas, las actividades tienen candados que indican que el desarrollador IT no puede eliminarlas. Lo que puede hacer con ellas es agregar detalles de implementación. La Figura 8-7 muestra el proceso *blueprint* generado cuando se comparte un modelo desde la BPA hacia el JDeveloper.

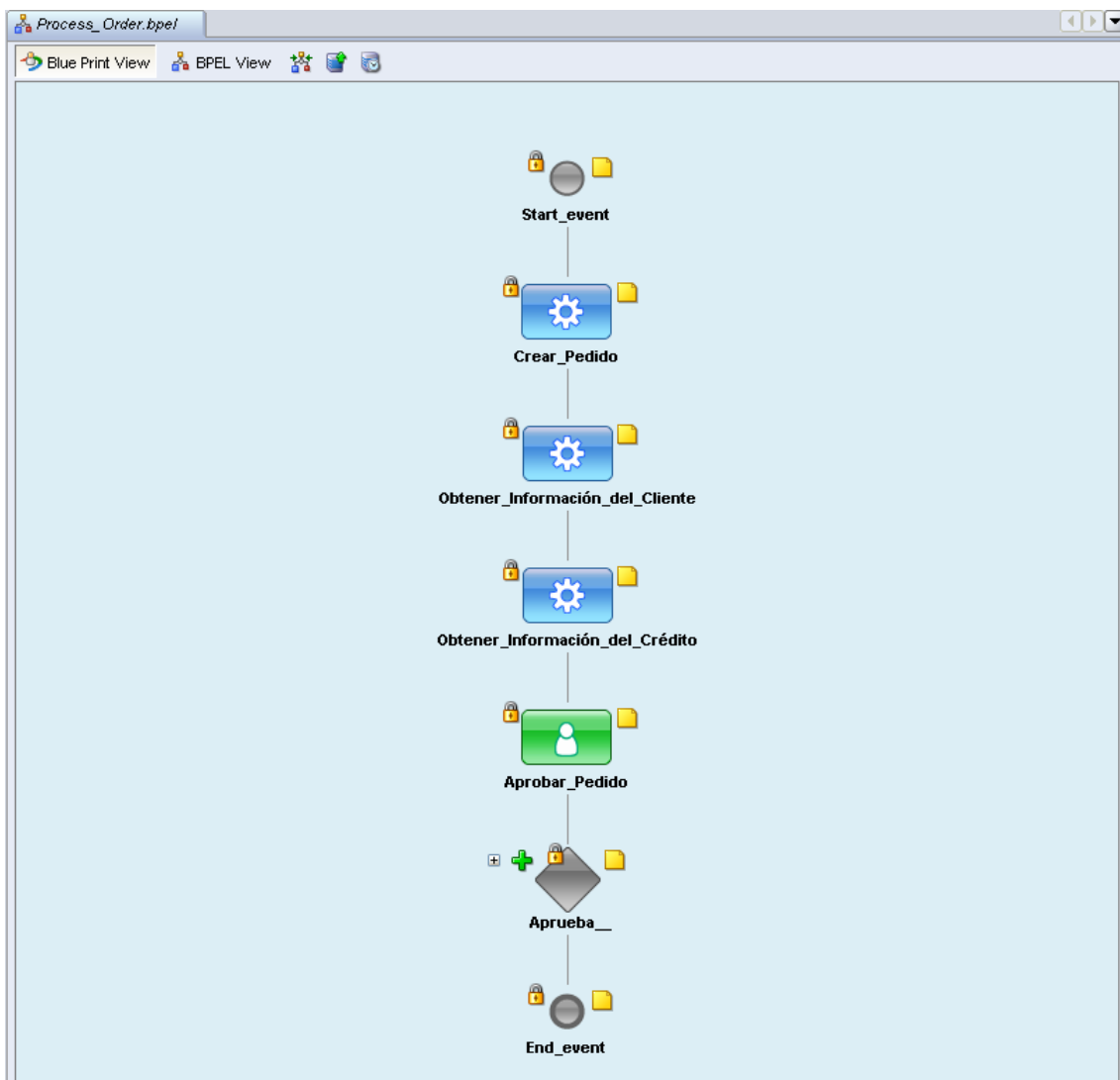


Figura 8 - 7: Vista de Negocios o *blueprint* de un proceso en el JDeveloper.

La **Vista BPEL** nos muestra el conjunto de artefactos BPEL que fueron generados automáticamente. Es importante considerar que los procesos BPEL generados no están completos y no pueden ser desplegados. El desarrollador IT tiene que enriquecer los modelos BPEL generados y tiene que crear los servicios de negocios faltantes, los mapeos de datos y las ligaduras y sensores para coleccionar métricas de negocio y manejar las excepciones del sistema. La Figura 8-8 muestra la **Vista BPEL** de un proceso de negocios. En los costados izquierdo y derecho podemos ver los *partnerLinks* que se comunican con el proceso.

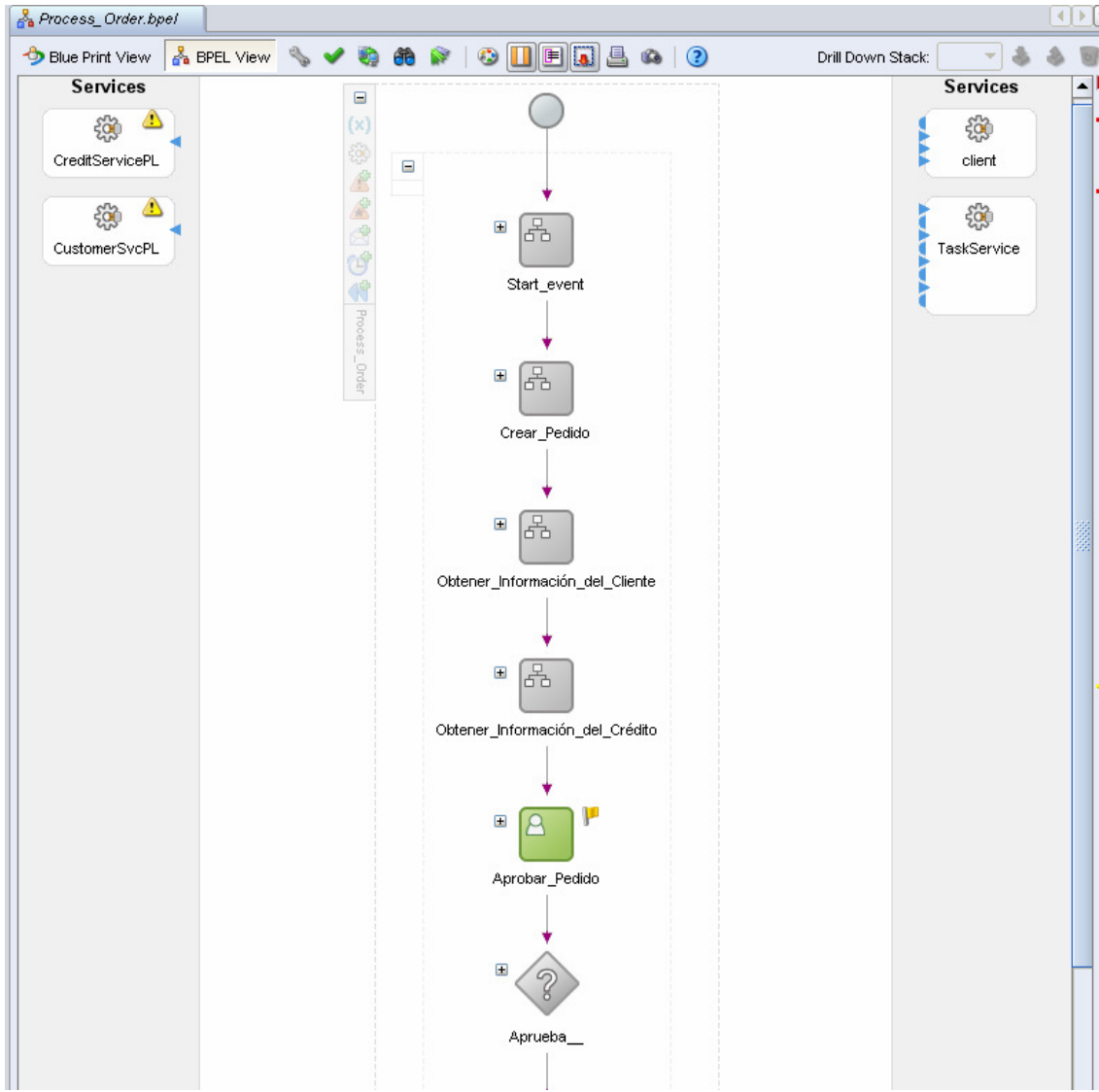


Figura 8 - 8: Vista BPEL de un proceso en el JDeveloper.

También, el JDeveloper muestra los archivos XML correspondientes a los procesos BPEL generados.

La IDE JDeveloper presenta un entorno amigable que permite fácilmente integrar los sectores de IT y de negocios. Las vistas previamente descritas facilitan esta integración. Si bien la herramienta está destinada al sector de desarrollo, la **vista de negocios** permite que los ejecutivos participen en esta etapa ya que dicha vista muestra los diagramas utilizando una notación BPMN.

La IDE presenta un conjunto de asistentes que facilitan la conexión al repositorio, creación de proyectos, importación de *blueprint* y la conexión de las actividades con los servicios que las van a implementar. Además, permite realizar validaciones del BPEL para encontrar cuales son los errores y partes del proceso que es necesario implementar. Para cada error encontrado se despliegan sus posibles soluciones. Todo esto se muestra en una ventana de *logs* con descripciones fáciles de comprender.

Ciclo Continuo de Desarrollo

La Oracle BPA Suite y la Oracle SOA Suite fueron integradas para que el sector de IT y el de negocios puedan realizar desarrollos en paralelo. Una vez que tenemos el modelo de un proceso y éste fue compartido con IT, es muy probable que surjan nuevos cambios desde el sector de negocios que es necesario reflejar en los correspondientes procesos BPEL. La Suite de Oracle brinda la posibilidad de realizarlo de una forma natural.

Cuando se realizan los cambios en el modelo de un proceso desde la BPA, éste se tiene que compartir nuevamente con IT. Recordemos que esto se hace desde el menú SOA con la opción “Compartir Diseño con IT”. Si la opción está seleccionada, se debe quitar el *check* y volverla a seleccionar. Una vez realizado esto, desde el JDeveloper podemos abrir la **vista *blueprint*** del proceso modificado y ésta nos va a indicar mediante un mensaje que una nueva versión del proceso ha sido detectada. Todo es realizado en tiempo real. En esa misma vista tenemos un botón llamado “Actualizar desde el Servidor de BPA”, que realiza una comparación entre la versión de la BPA y la versión del JDeveloper, mostrando en una ventana las diferencias encontradas entre ambos modelos. Seleccionando el botón “Diferencias Visuales” podemos ver las versiones y el combinado resultante de una forma visual. Presionando el botón OK, se realiza el combinado correspondiente.

Análisis de la SOA Suite y fases del ciclo de vida BPM soportadas

Componentes de la Herramienta

La herramienta está compuesta por varios módulos separados que tienen diferentes propósitos. Estos módulos son los siguientes:

- **Oracle BPEL *Process Manager***: provee al usuario una solución para desplegar y gestionar procesos de negocios BPEL.
- **Oracle *Enterprise Service Bus (ESB)***: el ESB mueve datos a través de múltiples sistemas, dentro y fuera de una empresa. Es la infraestructura subyacente de la arquitectura SOA y de la de manejo de eventos EDA. Permite monitorear y gestionar los datos de negocios con un impacto mínimo en las aplicaciones existentes [36].
- **Oracle *Web Services Manager (OWSM)***: es un componente utilizado para brindar seguridad a los *web services* y para monitorear las actividades realizadas sobre los *web services* protegidos [37].

- **Oracle Business Rules:** las reglas de negocios son sentencias que describen políticas de negocios. Este componente permite a los analistas de negocios cambiar políticas que están expresadas como reglas de negocios, con poco o nulo conocimiento de programación. Las aplicaciones que utilizan este componente soportan el cambio continuo que permita a las aplicaciones adaptarse a las nuevas regulaciones de la empresa o cambios de política [38].
- **Oracle JDeveloper:** es un entorno integrado de desarrollo (IDE) para construir aplicaciones y *web services* usando Java, XML y el estándar SQL. Soporta el ciclo de vida completo de desarrollo con características integradas para diseñar, codificar, *debuggear*, testear y desplegar aplicaciones [39].
- **Oracle Application Server:** provee un conjunto de componentes de la infraestructura de servicios para crear, desplegar, y gestionar arquitecturas orientadas a servicios. Consiste de Oracle HTTP *server* (basado en Apache), Oracle *containers* para J2EE (OC4J o J2EE Server) que permite desplegar aplicaciones basadas en J2EE y Oracle *enterprise manager 10g application server control* que permite realizar las tareas de administración y monitoreo del servidor [31].

Implementación y Despliegue

Desde la IDE JDeveloper se realiza la implementación de los procesos BPEL. Anteriormente, explicamos que los procesos BPEL pueden ser importados desde la BPA. Cuando se importa un proceso desde la BPA, se obtiene el beneficio de tener todo el flujo BPEL creado, basándose en el diagrama BPMN modelado previamente. Otra forma de realizar esto es crear un nuevo proyecto BPEL desde cero mediante el diseñador del JDeveloper. Los BPEL pueden ser sincrónicos o asincrónicos dependiendo del caso de uso a resolver y el tipo de comunicación que se necesita. Para ambos existen plantillas ya desarrolladas que nos proveen una estructura de directorios, archivos XML configurados y un modelo BPEL inicial el que debe ser enriquecido con componentes para implementar el proceso de negocios deseado. La tarea de IT consistirá en conectar los componentes BPEL que corresponden con los *web services* o *task* definidos agregando los *partnerlinks* (entidades que se comunican con el proceso BPEL) al diagrama.

La paleta de componentes BPEL predefinidos es muy completa y cada componente tiene asociado un conjunto de funciones predefinidas e implementadas, que están listas para ser utilizadas a la hora de la implementación. Existen funciones para manejos de *strings*, funciones matemáticas y asignación de valores entre otras, utilizando el lenguaje XPath (lenguaje que permite construir expresiones que recorren y procesan un documento XML [40]) para realizar la invocación de las mismas.

En la Figura 8.9 se puede ver la IDE JDeveloper, en la parte izquierda la estructura de directorios y archivos generados, en el centro el proceso BPEL que se está desarrollando y a la derecha la paleta con los componentes para esta vista.

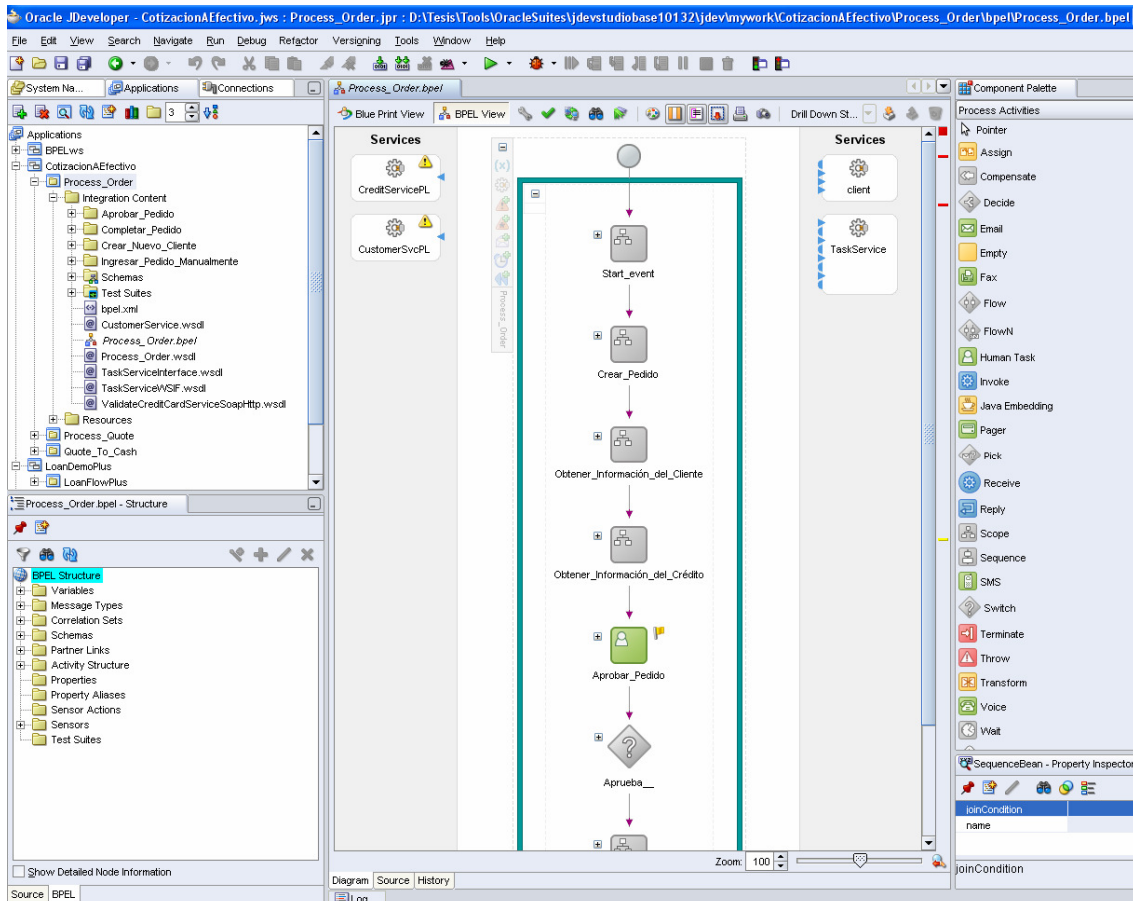


Figura 8 – 9: IDE JDeveloper

Algo importante a tener en cuenta es que JDeveloper es un entorno completo que nos permite diseñar *web services*, aplicaciones *web* y demás componentes que pueden ser conectados a los BPEL. Esto nos evita tener que utilizar otros entornos o herramientas para implementar los elementos que dan soporte a los BPEL.

Con respecto al diseñador de los procesos, es de muy fácil uso y bastante intuitivo facilitando el desarrollo visual y así limitándonos a arrastrar componentes al diagrama, y luego completar las propiedades que corresponden.

Una vez que los procesos BPEL están implementados, es necesario realizar la compilación de los mismos con la opción *make* del menú contextual que se despliega presionando el botón derecho del *mouse* sobre el proyecto. El resultado de la compilación se muestra en una consola de *log*. Si la compilación es exitosa, se crea una carpeta *output* dentro del proyecto que contendrá un archivo en formato **jar** que es el proyecto BPEL listo para ser desplegado. En caso de existir errores de compilación, los mismos se muestran en la consola, con una descripción detallada y una posible solución.

Existen dos formas de desplegar el proyecto BPEL en la SOA Suite: una consiste en entrar a la aplicación *BPEL manager* de la SOA Suite y elegir la opción de “Deployar Proceso”. En éste caso se busca el archivo **jar** previamente generado en la carpeta *output* del proyecto. La Figura 8-10 muestra el formulario para realizar el despliegue de un proceso desde la SOA Suite.

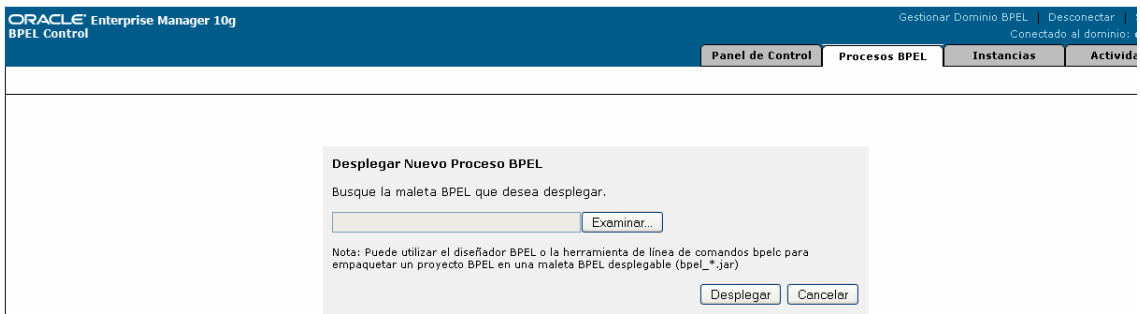


Figura 8 - 10: Despliegue de un proceso desde la SOA Suite

La otra forma de desplegar el proyecto BPEL en la SOA Suite es desde la IDE JDeveloper mediante la opción *deploy* -> *BPEL process deployer* del menú contextual del proyecto. La Figura 8-11 muestra la opción del menú que permite desplegar un proceso BPEL desde la IDE jDeveloper.

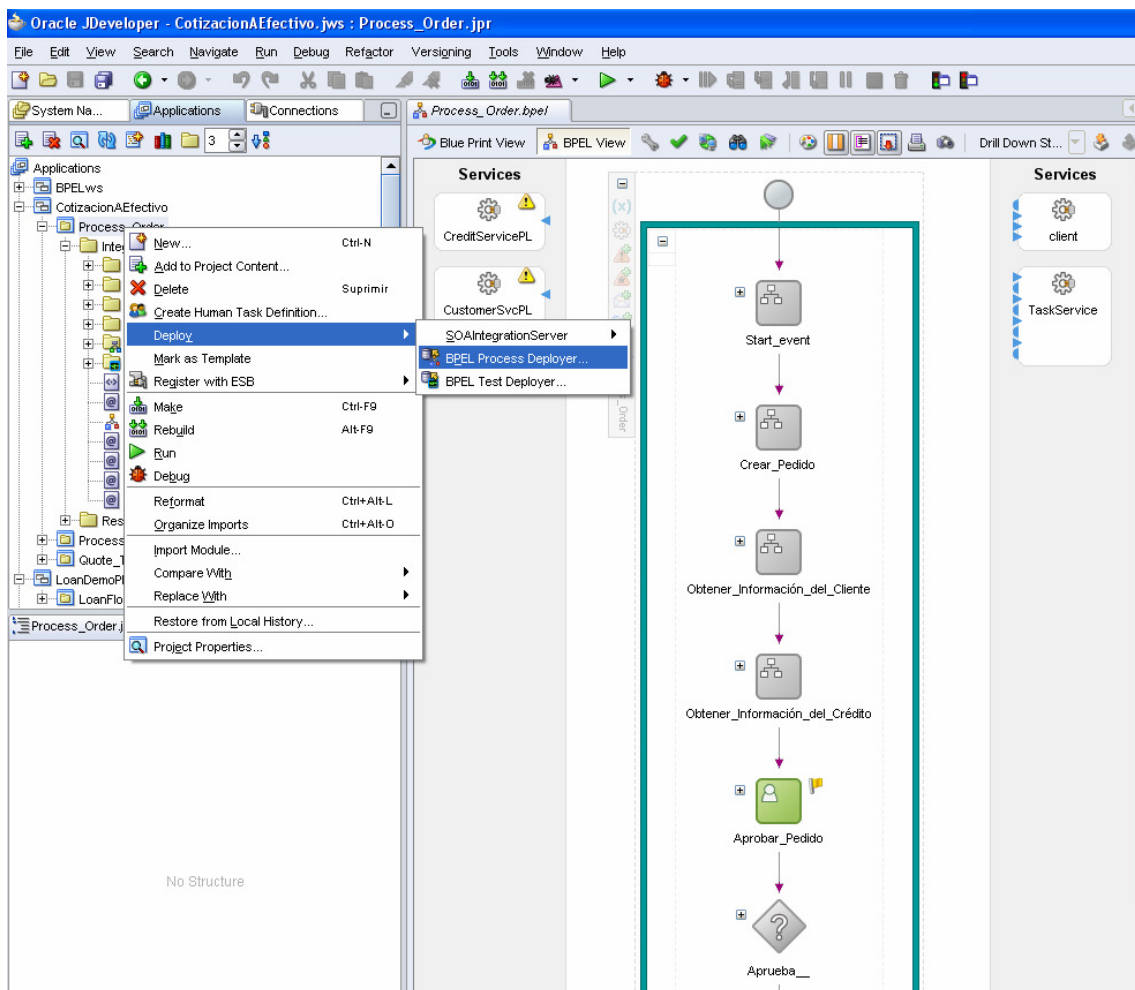


Figura 8 - 11: Despliegue de un proceso BPEL desde la IDE JDeveloper.

El despliegue de un proceso BPEL desde el jDeveloper hará un *make* automático para compilar el proyecto y si no hubo errores lo despliega directamente en el BPEL *manager* de la SOA con un número de versión asociado. Es importante tener en cuenta que para realizar esto último necesitamos tener una conexión entre el jDeveloper y los servidores de la SOA Suite (ver “Conexión de la SOA Suite con JDeveloper” en este

documento). Los procesos BPEL cuando se despliegan se publican automáticamente como servicios *web* para que otros procesos BPEL u otras aplicaciones puedan iniciarlos fácilmente.

Ejecución y Monitoreo

La etapa de ejecución se realiza y gestiona desde la componente BPEL *manager* de la SOA Suite. Esta componente nos permite realizar la administración y seguimiento de todos los procesos BPEL desplegados y de sus instancias en ejecución.

El BPEL *manager* es una herramienta *web* que nos permite:

- Visualizar todos los procesos desplegados y también desplegar nuevos procesos. Estas funcionalidades están disponibles desde la pestaña **Panel de Control**. También, tenemos una vista de las instancias de los procesos en ejecución y otra vista de los procesos terminados recientemente.
- Gestionar detalladamente cada proceso desplegado. Las operaciones más importantes que tenemos disponibles sobre los procesos son: **Anular Despliegue, Activar o Desactivar proceso** (un proceso desactivado no se puede instanciar), **Iniciar proceso** (provee un formulario para los datos de entrada), **Ver información sobre el WSDL del BPEL, Visualizar el XML que define al BPEL** y **Ver informes** de las instancias del proceso. Todas estas operaciones son accesibles desde la pestaña **Procesos BPEL**.
- Gestionar todas las instancias de los procesos. Disponemos de un buscador de instancias con varios filtros para localizar una instancia específica. Seleccionando una instancia, podemos ver la descripción de la misma (estado, fecha de modificación, prioridad, etc). A su vez, es posible gestionar dicha instancia (abortarla o suprimirla). También, podemos visualizar gráficamente el flujo de ejecución hasta el momento, teniendo la posibilidad de auditar cada actividad mostrando el XML asociado a cada una, con los datos enviados o recibidos. Es posible visualizar las interacciones asociadas a cada instancia, pudiendo de esta manera ver las actividades completadas y pendientes del flujo. Todas estas funcionalidades están disponibles a partir de la pestaña **Instancias**.
- Visualizar las actividades asociadas a todas las instancias. Disponemos de un buscador de actividades con filtros, que nos permite localizar alguna actividad en particular. Aquellas actividades que están pendientes, se pueden auditar para obtener una descripción de cual es el motivo por el que aún no han terminado. Estas funcionalidades están disponibles a partir de la pestaña **Actividades**.

La Figura 8.12 ilustra la consola *web* de la herramienta BPEL *manager*. En la parte izquierda de la pantalla se pueden ver los diferentes procesos desplegados y a la derecha el conjunto de solapas que fueron mencionadas anteriormente.

The screenshot shows the Oracle BPEL v10.1.3.1.0 web console interface. The main content area displays a table with the following data:

Procesos BPEL Desplegados			
Nombre	Instancia	Proceso BPEL	Fecha de Modificación ↑
CreditRatingService	30003 : Instance #30003 of StarLoan	StarLoan (v. 1.0)	15/10/08 08:40:10
LoanFlowPlus	30001 : Instance #30001 of LoanFlowPlus	LoanFlowPlus (v. 2.0)	15/10/08 08:40:00
StarLoan	20035 : Instance #20035 of StarLoan	StarLoan (v. 1.0)	13/10/08 17:27:22
TaskActionHandler	20033 : Instance #20033 of LoanFlowPlus	LoanFlowPlus (v. 2.0)	13/10/08 17:27:19
TaskManager			
UnitedLoan			
Instancias del Proceso BPEL Terminadas Recientemente (Más...)			
	30004 : Instance #30004 of UnitedLoan	UnitedLoan (v. 1.0)	15/10/08 08:40:02
	30002 : Instance #30002 of CreditRatingService	CreditRatingService (v. 1.0)	15/10/08 08:39:58
	20036 : Instance #20036 of UnitedLoan	UnitedLoan (v. 1.0)	13/10/08 17:27:20
	20034 : Instance #20034 of CreditRatingService	CreditRatingService (v. 1.0)	13/10/08 17:27:18

At the bottom left of the table area, there is a link: [Desplegar Nuevo Proceso](#). At the bottom right, the version number "Consola de Oracle BPEL v10.1.3.1.0" is displayed.

Figura 8 - 12: Consola *web* de la herramienta BPEL *manager*

A nuestro criterio esta herramienta es muy útil para los analistas de negocios porque le permite realizar el seguimiento de los procesos e instancias en ejecución. Incorpora un conjunto importante de opciones que permiten auditar todas las actividades que se ejecutan en cada una de las instancias de los procesos.

En la Figura 8-13 podemos ver la auditoria de una actividad de un proceso ejecutado en la SOA Suite.

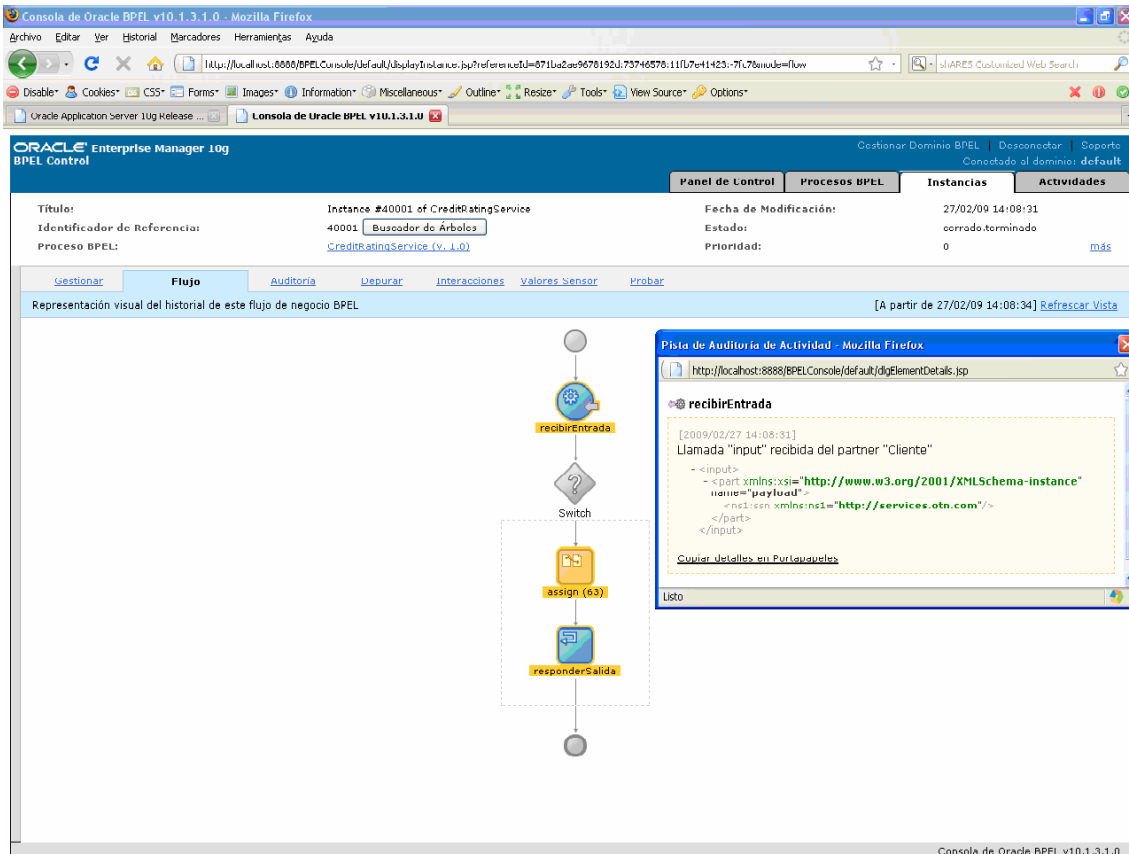


Figura 8 - 13: Proceso en ejecución y la auditoria de una de sus actividades usando la herramienta BPEL manager

Otra ventaja de esta herramienta es que es accesible vía web, esto nos da la posibilidad de poder controlar y auditar los procesos desde cualquier punto o PC conectada a Internet.

Conclusión General de la Suite

El conjunto de herramientas de Oracle para BPM (BPA, jDeveloper y SOA Suite), permiten cubrir todo el ciclo de vida del desarrollo de los procesos de negocios. La BPA y la SOA Suite están pensadas para ser utilizadas por los analistas de de negocios y el jDeveloper específicamente para el sector de IT.

A nuestro criterio, sería importante disponer de nuevos perfiles que permitan acortar la brecha entre los conocimientos técnicos del sector de IT y el conocimiento de la gente de negocios, facilitando así la comunicación entre los mismos.

Esta suite de herramientas resuelve y da soporte a aplicaciones empresariales de gran escala. El uso de las herramientas está afectado por una licencia paga que tiene un monto elevado, lo que hace que su utilización deba estar acorde a la envergadura del proyecto sobre el cual se aplicará.

En lo relacionado a requerimientos de hardware, la BPA y el jDeveloper corren en una PC hogareña sin requerir un hardware potente. Para la SOA Suite se necesita un servidor dedicado con altas prestaciones de hardware. Igualmente, esto depende de la cantidad y tamaño de las aplicaciones que se corren en el mismo.

Si bien las herramientas son muy completas e intuitivas, una gran desventaja es que se necesitan tres herramientas separadas para cumplir con todo el ciclo de vida de BPM, además de tener una base de datos Oracle en entornos productivos para poder persistir la información que necesitan las herramientas para funcionar (la base de datos Oracle Lite no es recomendable en este tipo de entornos). Esto trae como consecuencia tener que pagar muchas licencias y tener que configurar las mismas para que funcionen en conjunto. A la hora de realizar cambios en los procesos, estos deben pasar por varias herramientas. Esto exige una coordinación entre los diferentes sectores de la empresa. Para mitigar esta cuestión, las herramientas traen asistentes que ayudan a realizar la sincronización de una manera ordenada.

Una gran ventaja que tiene la solución de Oracle es que provee una plataforma de implementación para BPM basada en servicios, dándonos la posibilidad de aprovechar las ventajas que ofrecen los mismos. La SOA Suite incorpora varias herramientas para administrar, monitorear y brindar seguridad a todos los servicios dentro y fuera de la organización.

Además las herramientas adhieren a los estándares más importantes que tiene BPM como ser BPMN y BPEL.

Capítulo 9 – Conclusión Final

En este capítulo realizaremos una comparación de las herramientas analizadas, considerando las siguientes métricas:

- Integración de la Suite en una sola herramienta.
- Soporte para todo el ciclo de vida de los procesos de negocio.
- Adherencia a los estándares.
- Licencia.
- Soporte para SOA.
- Integración con entornos de desarrollo y soporte JAVA.
- Persistencia de Datos.

Integración de la Suite en una sola Herramienta

Existen varias ventajas a la hora de tener toda la funcionalidad para cumplir el ciclo de vida del desarrollo de procesos de negocios en una sola herramienta.

1. Facilita la realización de las distintas etapas del ciclo de vida de los procesos, ya que todo se realiza en un mismo entorno, evitando el pasaje del modelo de un proceso de una aplicación a otra; esto evita también inconsistencias entre los diagramas.
2. Simplifica la instalación y configuración, evitando problemas de compatibilidad entre múltiples componentes y sin tener que configurar la comunicación entre las mismas.
3. Ahorro en pago de licencias, en el caso de herramientas propietarias, ya que se paga una sola licencia y no varias (en el caso de necesitar varias herramientas).

En el caso de las herramientas analizadas, ninguna tiene sus componentes integrados en una sola herramienta. Todas necesitan dos o tres herramientas para cumplir el ciclo de vida del desarrollo de procesos completo.

En el caso de **Oracle**, la suite está dividida en tres herramientas: BPA para el diseño de procesos, jDeveloper (IDE) para el desarrollo de procesos y servicios y SOA Suite para ejecución, administración y monitoreo.

jBPM, Bonita y ConsistGPA se apoyan en dos herramientas para implementar el ciclo de vida de los procesos: una IDE de diseño y desarrollo de los procesos y un servidor de ejecución de procesos con una consola de administración y monitoreo.

Soporte para todo el ciclo de vida de los procesos de negocio

Como ya lo hemos descrito en este informe, las etapas en el desarrollo de procesos de negocios son: Modelado, Simulación, Implementación, Ejecución, Optimización y Monitoreo. Cada etapa es muy importante para el desarrollo de un proceso de negocios. La carencia de alguna de ellas hace que la herramienta tenga limitaciones.

Las etapas de modelado, implementación, ejecución y monitoreo están cubiertas de distinta manera por las cuatro herramientas analizadas, siendo las etapas de simulación y optimización las que presentan mayor interés en este análisis debido a su prestación heterogénea.

De las herramientas analizadas la única que cubre todas las etapas del ciclo de vida es la suite de **Oracle**. Cuenta con simulación en etapa de diseño, permitiendo depurar errores o ineficiencias en etapas tempranas del desarrollo, logrando así la obtención de un proceso optimizado antes de su implementación.

ConsistGPA, **jBPM** y **Bonita** no cuentan con soporte para la simulación en ninguna etapa del desarrollo.

ConsistGPA es la única herramienta que trae soporte para BAM dentro de la suite, permitiendo un monitoreo más específico y orientado a la optimización de los procesos. En el caso de **Oracle** el soporte a BAM lo da a través de otra herramienta separada (*Oracle business activity monitoring*).

El resto de las suites BPM no traen soporte para BAM, sino que brindan un monitoreo básico de procesos y auditoría a través de sus consolas de administración *web*.

En lo que respecta a la fase de optimización, **ConsistGPA** permite hacer una optimización de procesos basándose en los resultados obtenidos en las estadísticas y reportes del módulo de BAM. Esta optimización se realizaría en etapas de ejecución. En el caso de **Oracle** como mencionamos anteriormente, brinda soporte a optimización en etapa de diseño, simulando la ejecución de procesos, obteniendo estadísticas sobre las variables de interés. **jBPM** y **Bonita** no traen ninguna facilidad o herramienta para realizar optimización dentro de la suite.

Adherencia a los estándares

Dentro de los estándares más importantes que están relacionados con BPM tenemos BPMN (<http://www.bpmn.org/>), XPD L (<http://www.wfmc.org/xpdl.html>) y BPEL (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel). Cada uno tiene su función dentro del desarrollo de procesos de negocios. Si bien no son imprescindibles dentro de BPM, el soporte de los mismos dentro de las herramientas tiene un valor agregado, permitiendo estandarizar el desarrollo de los procesos, facilitando la comunicación, compresión y portabilidad con otras herramientas que también adhieran a los mismos.

La suite de **Oracle**, **ConsistGPA** y **Bonita** traen soporte para BPMN. Su notación y los componentes para el diseño de procesos respetan este estándar. **jBPM** trae una

notación propietaria (JPDL), que si bien se asemeja a BPMN, está más cercana a la implementación de los procesos.

Con respecto a XPD, la única herramienta que trae soporte para este estándar es **Bonita**. Esto permite exportar los diagramas de los procesos a otra herramienta que soporte este estándar. Las restantes herramientas utilizan un xml propietario para realizar la definición de sus procesos.

En el caso del estándar BPEL, la única herramienta que trae soporte es **Oracle**. Las restantes herramientas no adhieren a este estándar.

Licencia

Este aspecto indica si el licenciamiento de las herramientas adoptadas es de código fuente abierto o propietario. La decisión que la organización tome en este sentido tiene que estar alineada con la estrategia global de la empresa u organismo ya que estaría en juego el mantenimiento y responsabilidad a futuro de las aplicaciones.

Las soluciones de código fuente abierto implican mayores libertades pero también exigen en los equipos responsabilidades diferentes y una metodología de trabajo que garantice la actualización de versiones y participación activa en comunidades FLOSS. Por otro lado soluciones propietarias implican la necesidad de contratos de mantenimientos que aseguren que las herramientas se actualicen y no se vuelvan obsoletas o peligrosas desde el punto de vista de vulnerabilidades de seguridad. Desde el punto de vista económico, las soluciones propietarias exigen el pago de licencias para adquisición y uso mientras que las soluciones de código abierto no exigen pago de licencia pero si responsabilidades especiales en el equipo de trabajo e inversión en recursos humanos normalmente más especializados.

Como lo describimos anteriormente, las herramientas **Bonita** y **jBPM** tienen licencia LGPL (*GNU Lesser General Public License*). En contraste las herramientas de Oracle (**BPA**, **SOA Suite** y **jDeveloper**) son propietarias con una licencia paga OTN (*Oracle Technology Network License Agreement*) al igual que **ConsistGPA** (Licencia ConsistGPA).

En cuanto al soporte a usuarios de las herramientas analizadas, observamos que la suite de **Oracle** cuenta con un grupo de especialistas que se dedican a resolver problemas que aparecen con el uso intensivo de las mismas, además de hacer disponibles foros de consulta de rápida respuesta como el foro de SOA Suite <http://forums.oracle.com/forums/forum.jspa?forumID=320>, entre otros.

Esto mismo se observa en **jBPM** que cuenta con una comunidad dinámica de usuarios (http://www.jboss.org/jbossjbpm/jbpm_community/) y un conjunto de herramientas colaborativas como *wikis* y foros que facilitan la rápida solución de problemas encontrados durante el desarrollo.

Soporte para SOA

Los conceptos de SOA y BPM resultan ortogonales. Así es posible implementar BPM con o sin SOA, pero una solución de BPM sin SOA es muy frágil ya que cualquier

cambio en la implementación impacta directamente en los procesos de negocio, siendo una solución muy propensa a errores.

Generalmente, la mayoría de las organizaciones tienen diversas aplicaciones implementadas en variadas tecnologías. Típicamente existen múltiples silos de aplicaciones y compartir información entre éstos no es una tarea sencilla ya que existen diferencias de tecnología, plataforma y modelos de datos [41]. Utilizar SOA y *web services* introduce una capa de servicios que consiste en una línea de servicios de negocios alineada a un dominio de negocio particular. De esta manera obtenemos, **servicios reusables** de negocios, que pueden ser compartidos por dominios múltiples de negocio y, **una plataforma de *web services***, que permite que los servicios sean utilizados independientemente de la aplicación y la plataforma tecnológica que los implementa [41].

Entre las cuatro herramientas analizadas, la que mejor se ajusta a los estándares de SOA son las suites de **Oracle** (Oracle BPA y Oracle SOA Suite), ya que utilizan como lenguaje de ejecución a BPEL, el estándar de facto en la orquestación de *web services*. Esto facilita el despliegue de servicios desde los lenguajes notacionales de procesos, favorece la modificación y actualización de los procesos y reduce la brecha entre la lógica de los procesos y la de los servicios. Los procesos se modelan en la BPA y se transfieren al jDeveloper para ser implementados, generando automáticamente un esqueleto del proceso en lenguaje BPEL. El desarrollador realiza la implementación de este proceso utilizando *web services* (la IDE provee amplio soporte para el desarrollo de *web services* en lenguaje JAVA). A su vez, provee soporte para el lenguaje XPath que aporta un gran conjunto de operaciones para manipular los diferentes xml. Además, la IDE jDeveloper realiza chequeos de sintaxis de los procesos BPEL, permite compilarlos y desplegarlos en la SOA Suite con la ayuda de asistentes. La SOA Suite trae un motor de ejecución que permite ejecutar y administrar los procesos BPEL. También, trae soporte para el despliegue y administración de todos los *web services* que se utilicen en los procesos de negocios. De lo anterior se deduce que todas las herramientas de Oracle analizadas apuntan a dar soporte directo a BPEL en todo el ciclo de desarrollo de los procesos.

ConsistGPA no provee soporte para BPEL. Posee una tecnología de adaptadores muy sencilla de definir y un componente para poder ejecutar un *web service* dentro de una actividad automática. La tecnología de adaptadores no se ajusta a los estándares de la industria como lo es BPEL, y de esta manera hace más costosa la integración y la interoperabilidad.

jBPM no trae soporte directo a BPEL, ni tampoco un paquete de herramientas unificado para lograr esto. La manera de obtener una implementación SOA con jBPM es a través de los componentes “*ESB service*” dentro del diseñador de procesos en la IDE Eclipse. Existen varias ventajas en el uso de los ESB, entre ellas podemos mencionar estandarización, bajo acoplamiento, escalabilidad y confiabilidad [42].

Bonita no provee soporte alguno para SOA, no cuenta con ninguna facilidad para conectar una actividad con un servicio, teniendo el desarrollador que implementar un cliente en Java para poder utilizar el mismo en una actividad del proceso de negocio.

Integración con entornos de desarrollo y soporte JAVA

Es indiscutible hoy la necesidad de contar con un IDE de desarrollo que facilite y agilice la implementación de los procesos de negocio. En el transcurso de esta tesina vimos básicamente dos IDE, Eclipse y jDeveloper de Oracle.

La IDE Eclipse es utilizada con un *plugin* específico por las herramientas **jBPM** y **Bonita**. Su instalación y configuración no tiene demasiados problemas y puede ser realizada fácilmente.

La integración de jDeveloper fue descrita en el Capítulo 8 y consiste en asociar los diferentes servidores de Oracle: servidor de procesos y servidor de aplicaciones para despliegues de los servicios. Esta tarea tampoco es demasiado compleja pero requiere la administración y configuración de dos servidores.

En cuanto a **ConsistGPA**, la herramienta no posee una IDE de desarrollo y en cambio ofrece una plantilla asociada al Microsoft Visio, que cubre la etapa de modelado. La principal ventaja es la facilidad de uso, permitiendo que personas no vinculadas a IT puedan rápidamente modelar procesos de negocio. Sin embargo, esto tiene la limitación propia del uso de una plantilla que no incorpora toda la simbología de la notación y que no puede traducirse en servicios, requiriendo de otra herramienta que interprete esa plantilla. En lo que respecta a la implementación, la plantilla no trae soporte para desarrollar código en ningún lenguaje de programación, esto limita las posibilidades de agregar funcionalidad a los procesos de negocios. Para implementar los procesos, la plantilla provee un conjunto de formularios que permiten configurar todas las funcionalidades de los elementos de un diagrama de procesos de negocios; esto se puede complementar asociando objetos externos mediante adaptadores y componentes. La plantilla agrega un menú de opciones que permite realizar el despliegue de los procesos de negocios de una manera sencilla, como así también, importar y exportar los modelos a un xml propietario.

En lo que respecta a la administración de servicios, en todas las herramientas que proveen soporte de servicios, los mismos se administran fuera de las IDE de desarrollo. En las IDE de desarrollo solo se realiza la integración con los mismos, o sea, se asocian los servicios a los nodos o actividades de los procesos. Para ello, los servicios deben estar disponibles en el servidor para realizar esta asociación. La única herramienta que permite el despliegue de servicios hacia el servidor es la herramienta de **Oracle** mediante un asistente del jDeveloper que realiza esta operación. En el caso de **jBPM** y **ConsistGPA**, los servicios se deben desplegar manualmente en el servidor.

Con respecto a la tecnología Java, todas las herramientas que hemos analizado soportan Java. En el caso de **jBPM** y **Bonita**, el *plugin* de Eclipse permite implementar las tareas automáticas mediante clases Java. En el caso de **ConsistGPA**, se utilizan los adaptadores y componentes para asociar objetos externos implementados en Java que proveen funcionalidad a las actividades automáticas. Por último en el caso de **Oracle**, jDeveloper facilita la implementación de *web services* en Java, siendo la tecnología más flexible para la implementación de los servicios que provienen del motor de BPEL.

Si bien SOA-BPM es una metodología despegada de la implementación, en la medida que las soluciones se ajusten mejor a la plataforma de desarrollo redundando en facilidades para la documentación, depuración y mantenimiento posterior. En este sentido las

aplicaciones basadas en JAVA adhieren a metodologías de diseño y paradigmas orientados a objetos, cuentan con un *framework* estándar y ampliamente usado como es JEE [43] y todos los *frameworks* arquitecturales basados en JEE que facilitan el desarrollo de aplicaciones distribuidas y mantienen la adherencia a los estándares. De esta manera se garantiza modularidad, extensibilidad, adaptación a cambios y otras cuestiones relacionadas a los requerimientos no funcionales de aplicaciones distribuidas como es la seguridad, alta disponibilidad y tolerancia a fallas.

De esta manera, se puede concluir que en la actualidad el soporte para Java provisto por las suites para BPM aporta un importante valor agregado a la hora de evaluar su factibilidad de uso y posible adopción.

Persistencia de datos

En este punto, analizamos la persistencia de los datos internos requeridos para el funcionamiento de los motores de procesos, no cubriendo los aspectos vinculados con la persistencia de datos como un recurso administrado por las aplicaciones y los servicios.

Cada una de las herramientas que analizamos necesitan persistir datos internos a la misma, para que estas pueda funcionar. Por ejemplo, el motor de ejecución de procesos necesita mantener el estado de los procesos que se están ejecutando, junto con todas las instancias de los mismos. Para persistir estos datos internos, las herramientas utilizan bases de datos relacionales. Ya describimos cuales son las bases de datos soportadas por cada una de las herramientas, las cuales inicialmente están configuradas con bases de datos de desarrollo y que luego, en ambientes productivos deberán ser reemplazadas por bases de datos de más envergadura o de producción.

jBPM, **ConsistGPA** y **Bonita** ofrecen la posibilidad de utilizar diversos motores de bases de datos y proveen *scripts* que generan el esquema de base de datos para el motor elegido, esto es debido a que el lenguaje SQL si bien está estandarizado no es directamente portable entre los distintos motores de bases de datos.

Las herramientas más limitadas en cuanto a la persistencia de datos, son las suites de Oracle, ya que éstas obligan a utilizar bases de datos Oracle que agrega el pago de otra licencia a la hora de puesta en marcha de un desarrollo BPM con este conjunto de herramientas.

Conclusión

El enfoque orientado a procesos explicita el conocimiento muchas veces oculto acerca de cómo, quién y cuándo se realizan las actividades de una organización. La gestión de los procesos de negocios BPM permite cubrir todo el ciclo de vida de los procesos de negocios desde su modelado, hasta su monitoreo, pasando por la implementación, ensamblado y el despliegue.

A lo largo de esta tesina hemos aprendido mucho sobre BPM y sus estándares, analizando el manejo de cuatro herramientas diferentes muy aceptadas en la actualidad. Este estudio exhaustivo nos permitió adquirir habilidades en el uso y en la adopción de la metodología BPM, según lo ofrecido por cada herramienta.

Asimismo, nos permitió recorrer el ciclo de vida de los procesos de negocios que es totalmente diferente al ciclo de vida del desarrollo del *software* con el que estamos familiarizados los tecnólogos. Esta experiencia nos brindó la posibilidad de enfocar los problemas desde una perspectiva más amplia que la del aspecto meramente tecnológico, teniendo en cuenta los distintos roles y actores intervinientes en una organización moderna donde los cambios son muy frecuentes y se requiere de una gran capacidad de adaptación a los mismos. La siguiente tabla muestra una comparativa de las principales métricas analizadas de cada una de las herramientas:

	Hardware	Licenciamiento	Ciclo de vida		Soporte SDA	Integración con IDEs JAVA
JBPM	768 MB de RAM 800 MB de espacio en el disco rígido para la versión full (platform) o 220 Mb de espacio en disco rígido para la versión standalone	Lesser General Public License (LGPL)	Modelado ✓	Ejecución/ Despliegue ✓	Monitoreo ✓	Cuenta con componentes "ESB service" dentro del diseñador de procesos para utilizar servicios que estén disponibles en el ESB Eclipse con plug-in http://www.eclipse-plugins.info/eclipse/plugin_details.jsp?id=1028 http://sourceforge.net/projects/jbpm
Bonita	Procesador de 1 GHz es recomendado, con un mínimo de 512 de RAM	Lesser General Public License (LGPL)	Modelado ✓	Ejecución/ Despliegue ✓	Monitoreo ✓	No provee Eclipse con plug-in http://forge.objectweb.org/project/showfiles.php?group_id=56
GPA	Procesador: PC Pentium 233 MHz o superior Pentium III recomendado. 128 MB de RAM (o superior) Disco Rígido: 245 MB de espacio disponible	Licencia Consistent-GPA	Modelado ✓	Ejecución/ Despliegue ✓	Monitoreo ✓	Posee un mecanismo propio para definir un adaptador y un componente para ejecutar un servicio web No posee un entorno de desarrollo. Ofrece una plantilla de modelado sobre Microsoft Visio: http://office.microsoft.com/en-us/visio/default.aspx
Oracle Business Process Analysis Suite 10g Oracle SOA Suite 10g	Procesador Intel Pentium IV 3.0 Ghz. 2 GB RAM. Tarjeta gráfica con resolución de 1024 x 768 con al menos 256 colores Red de alta velocidad (100 Mbit) entre el servidor de base de datos y Oracle business process repository	OTN License Agreement	Modelado ✓ Oracle BPA 10g	Ejecución/ Despliegue ✓ JDeveloper 10g y Oracle SOA Suite 10g	Monitoreo ✓	Utiliza como lenguaje de ejecución de ejecución a BPCL. JDeveloper 10g

Figura 9 - 1: Tabla comparativa extraída del poster presentado en el congreso WICC 2009.

Referencias

- [1] Patricia Bazán. “BPEL: una propuesta para el uso de Web Services”. CACIC. 2007.
- [2] Nayna Patel y Vlatka Hlupic. “Dynamic Business Process Modelling (BPM) For Business Process Change”. Department of Information Systems and Computing, Brunel.
- [3] BPM Group Snoop Consulting. “BPM Group”. Junio 2008. Usuario: facundo.rodriguez ; Clave:leonidas. <http://tiki.snoopconsulting.com/tiki-index.php?page=BPM&highlight=bpm>
- [4] Object Management Group (OMG). “Business Process Management with OMG specifications”. 140 Kendrick street Building a suite 300 Needham, MA 02494 U.S.A.
- [5] Bill Curtis y John Alden. “The Business Process Maturity Model (BPMM): What, Why and How”. Febrero 2007.
- [6] Object Management Group (OMG) . “Business Process Maturity Model (BPMM) v1.0 - Beta 2”. Noviembre 2007.
- [7] Stephen A. White. “Introduction to BPMN”. IBM Corporation.
- [8] Martin Owen and Jog Raj, Popkin Software. “BPMN and Business Process Management, Introduction to the New Business Process Modeling Standard”. 2003.
- [9] Workflow Management Coalition (WFMC). “Process Definition Interfase - XML Process Definition Language, Versión 2.00”. Octubre 2005.
- [10] OASIS. “Web Services Business Process Execution Language Version 2.0”. Abril 2007.
- [11] Red Hat JBoss Community. “JBoss Enterprise SOA Platform datasheet”. Julio 2008.
- [12] Red Hat JBoss Community. “JBoss Enterprise Application Platform datasheet”. Abril 2007.
- [13] Red Hat JBoss Group. “jBPM Overview”.
http://www.jboss.org/jbossjbpm/jbpm_overview/
- [14] Red Hat JBoss Community. “Getting Started Guide JBoss Enterprise SOA Platform 4.2”. Julio 2008.
- [15] Red Hat JBoss Group. “JBoss jBPM jPDL 3.2 jBPM jPDL User Guide”. Febrero 2006. <http://docs.jboss.org/jbpm/v3/userguide/>
- [16] Red Hat JBoss Community. “JBoss Enterprise SOA Platform Answers to Frequently Asked Questions”. <http://www.jboss.com/products/platforms/soa/faqs/>
- [17] BonitaTeam. “Nova Bonita Workflow – Reference Guide”. Septiembre 2008.

- [18] OW2 Consortium y Bull. “BONITA THE OPEN SOURCE WORKFLOW PROJECT”.
- [19] Wikipedia Community. “GNU Lesser General Public License”. Octubre 2008. http://es.wikipedia.org/wiki/Licencia_p%C3%BAblica_general_limitada_de_GNU
- [20] BonitaTeam. “Nova Bonita Workflow - Quick Start Guide”. Septiembre 2008.
- [21] Wikipedia Community. “HSQLDB”. Septiembre 2008. <http://es.wikipedia.org/wiki/HSQLDB>
- [22] Consist (Business Information Technology). “Consist GPA”. http://www.consist.com.ar/pub2/PublicNew/sitio/ccm_pagina.jsp?codcanal=consistgpa_PY&pais=paraguay&country=PY&language=es.
- [23] Consist (Business Information Technology). “Gestión y Automatización de Procesos de Negocio”.
- [24] Consist (Business Information Technology). “Manual Usuario Herramienta de Definición de Procesos”.
- [25] Microsoft Corporation. “Requerimientos del sistema para Visio 2003”. Octubre 2003. <http://www.microsoft.com/latam/office/visio/prodinfo/sysreq.msp>
- [26] Consist (Business Information Technology). “Archivo Readme de la Instalación”.
- [27] Consist (Business Information Technology). “Manual Usuario Definicion Metricas y Dimensiones”.
- [28] Oracle Corporation. “Oracle Business Process Architect, Quick Start Guide Release 10.1.3.4”. Abril 2008.
- [29] Oracle Corporation. “Oracle® SOA Suite Quick Start Guide 10g (10.1.3.1.0)”. Septiembre 2006.
- [30] Oracle Corporation. “Oracle Business Process Architect, Administration Guide Release 10.1.3.4”. Marzo 2008.
- [31] Oracle Corporation. “Oracle Application Server Installation Guide 10g Release 3 (10.1.3.1.0) for Microsoft Windows”. Octubre 2006.
- [32] Oracle Corporation. “Oracle Business Process Architect, Installation Guide Release 10.1.3.4”. Marzo 2008.
- [33] Anil Passi. “Connect jDeveloper to SOA Servers”. Enero 2007. <http://oracle.anilpassi.com/connect-jdeveloper-to-soa-servers-2.html>
- [34] Oracle Corporation. “Oracle® Enterprise Service Bus Quick Start Guide 10g (10.1.3.1.0)”. Septiembre 2006. http://download-uk.oracle.com/docs/cd/B31017_01/integrate.1013/b28212/buildendtoend.htm

[35] Ayuda Oracle Business Process Architect 10.1.3.4 “Explorador, Diseñador, Editor de Matrices, Simulador, Administración, Editor de Secuencias de Comandos”.

[36] Oracle Corporation. “Oracle® Enterprise Service Bus Quick Start Guide 10g (10.1.3.1.0)”. Septiembre 2006.
http://download.oracle.com/docs/cd/B31017_01/integrate.1013/b28212/intro_qs.htm#BABFGJFE

[37] Atul in owsm, soa. “Oracle Web Services Manager Architecture (OWSM)”. Diciembre 2007. <http://onlineappsdba.com/index.php/2007/12/15/oracle-web-services-manager-architecture-owsm/>

[38] Oracle Corporation. “Oracle® Business Rules User's Guide 10g (10.1.3.1.0)”. Septiembre 2006.
http://download.oracle.com/docs/cd/B31017_01/web.1013/b28965/intro.htm#BHCBEBC

[39] Oracle Corporation. “Oracle® BPEL Process Manager Quick Start Guide 10g (10.1.3.1.0)”. Septiembre 2006.
http://download.oracle.com/docs/cd/B31017_01/integrate.1013/b28983/intro.htm#sthref9

[40] Wikipedia. “XPath”. Octubre 2008. <http://es.wikipedia.org/wiki/XPath>

[41] Eric Newcomer, Greg Lomow. “Understanding SOA with Web Services”. Abril 2005.

[42] Saul Caganoff. “The benefits of an ESB”. Enero 2008.
<http://www.soabloke.com/2008/01/31/the-benfits-of-an-esb/>

[43] Sun Microsystems Java EE 5. “Sitio Oficial de JEE”. 2009.
<http://java.sun.com/javaee/technologies/javaee5.jsp>