

Universidad de las Ciencias Informáticas

FACULTAD 3



**Título: Herramienta para el manejo de flujos de
trabajos y su integración en una arquitectura
empresarial de JEE.**

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autores: Yunier Leyva Rivera

Roaldys García Robaina

Tutor: Ing. Yusbel Echemendía González

Junio del 2007

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de Junio del año 2008.

Yunier Leyva Rivera

Roaldys García Robaina

Yusbel Echemendía González

Firma del Autor

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO

Tutor: Yusbel Echemendía González

Especialidad de graduación: Ingeniería en Ciencias Informáticas

Categoría docente: Adiestrado

Categoría Científica: Ninguna

Años de experiencia: Ninguno

Año de graduado: 2007

AGRADECIMIENTOS

A mami y a mi papá Luis Ángel que con mucha paciencia y sabiduría me han guiado por el camino correcto y han aportado mucho en mi vida.

A mi papá Pedro, por todo el amor, educación, apoyo y ejemplo que me has brindado, con lo cuál he logrado todas las satisfacciones por las cuales hoy día, me siento orgulloso como persona.

A mis tíos, René, Fredy, Julio, Pedro, Nelson, Miguel que siempre han estado pendientes de mí.

A mi primo Leiner, por ser un hermano.

A toda mi familia y a todos aquellos que no he mencionado, pero que de sobra se deben dar por aludidos por haber contribuido de un modo u otro en mi formación y a la consumación de este sueño. A todos muchas gracias.

Yunier

A mi mamá por todo su cariño, esfuerzo, comprensión, por apoyarme y guiarme siempre por el buen camino, por ayudarme a ser todo lo que soy, por el amor que siempre me ha dado, por estar ahí donde siempre la he necesitado. No existe límite o dimensión que pueda describir el amor que siento por ti. Te quiero demasiado.

A mi papá por ser mi ejemplo a seguir. Como profesional, por el amor, la sabiduría y su desempeño en el desarrollo de este trabajo, sin el no se hubiera podido realizar. Por tenerlo ahí cuando lo he necesitado. Igual que mami papo, te quiero mucho.

En especial a mi hermanita, que aunque la vida no le permitió estar aquí hoy, sé que estaría orgullosa de mí. Por ser mi fiel amiga, guiarme por el buen camino, ayudarme cuando lo necesité, nunca te olvidaré. Como te extraño Grey, siempre estarás en mi corazón, te amo, te quiero, te adoro.

A mi tía Claribel, por ser mi segunda mamá, por su apoyo desde siempre, su inmenso cariño y por estar siempre pendiente de mis estudios.

A mis tías y primos que de una forma u otra estuvieron muy pendientes de mis estudios.

A mis amigas y amigos del preuniversitario y la universidad, por acompañarme en los buenos y malos momentos, por su apoyo y ayuda siempre que la necesité.

A todos los que de una manera u otra han contribuido a mi formación profesional.

A todos los que ya no se encuentran físicamente pero sí en los corazones y pensamientos de los seres queridos.

Roaldys

DEDICATORIA

A la memoria de mi abuela Esther, que aunque ya no este entre nosotros se hubiera sentido muy feliz en este día.

A mi abuela Nubia, que ha estado en todo momento de mi vida junto a mí brindando su amor y experiencia.

Yunier

A mi familia, en especial a mis padres y mi hermana.

Roaldys

RESUMEN

El presente Trabajo de Diploma se realiza un estudio detallado sobre los conocimientos de la tecnología para la automatización del flujo de trabajo (en lo adelante workflow) en organizaciones o empresas. Se estudia los principales aportes a la eficiencia en los procesos y en la estandarización del trabajo que brindan, se realiza un cuestionario a estudiantes de la Universidad de Ciencias Informática ¹(UCI) de la facultad 3 que participan en proyectos productivos acerca de esta tecnología workflow, de los cuales se arriba a la conclusión de que no cuentan con suficientes conocimientos sobre estos sistemas y no existe un procedimiento que estratégicamente contribuya a la materialización de la concepción teórica desde la profundización de una herramienta workflow. Por lo planteado anteriormente se realiza un estudio detallado sobre los principales conceptos y funcionalidades de esta tecnología. Se hace un análisis de las principales características de varias herramientas workflow desarrolladas en java² y de licencia libre, de las cuales se toma como referencia a Bonita Workflow por ofrecer altos niveles de funcionalidad y brindar soporte a la mayoría de los patrones workflow que se analizan. Se investiga a profundidad esta herramienta en cuanto a su arquitectura, requerimientos, comunidad, soporte y licencia, madurez y su puesta en marcha. El resultado obtenido es el procedimiento anteriormente descrito que permite de una forma estratégica y con varios momentos en el desarrollo, profundizar en el conocimiento de los sistemas workflow's como una solución para el mejor funcionamiento y organización de los procesos en los proyectos productivos.

PALABRAS CLAVE

Tecnología Workflow, Patrones workflow, Sistemas workflow.

¹ **Universidad de las Ciencias Informáticas.** Es un centro universitario de la educación superior cubana, docente- productor para la industria del software en Cuba.

² Ver Glosario.

TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	III
DEDICATORIA	IV
RESUMEN.....	V
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	2
1.1 INTRODUCCIÓN	2
1.2 CARACTERIZACIÓN DEL SOFTWARE LIBRE.....	2
1.2.1 <i>Software de código abierto (OPEN SOURCE)</i>	2
1.2.2 <i>Licenciamiento de Software Libre</i>	3
1.3 LA PLATAFORMA JEE	4
1.4 ESTÁNDARES Y ORGANIZACIONES.....	6
1.5 WORKFLOW	9
1.5.1 <i>Conceptos y definiciones</i>	11
1.5.2 <i>Tipos de workflow's</i>	14
1.5.3 <i>Adaptabilidad y workflow</i>	15
1.6 SISTEMA DE ADMINISTRACIÓN DE WORKFLOW	17
1.6.1 <i>Características</i>	17
1.6.2 <i>Funciones y beneficios de los sistemas workflow's.</i>	18
1.6.3 <i>Modelo de referencia de workflow.</i>	19
1.7 PATRONES DE WORKFLOW	23
1.7.1 <i>Patrones de flujo de control básicos.</i>	24
1.7.2 <i>Bifurcación avanzada y patrones de sincronización.</i>	25
1.7.3 <i>Patrones estructurales.</i>	27
1.7.4 <i>Patrones de múltiples instancias.</i>	27
1.7.5 <i>Patrones basados en estados.</i>	28
1.7.6 <i>Patrones de cancelación.</i>	30
1.8 CRITERIO DE COMPARACIÓN	30
1.9 HERRAMIENTAS PARA LA GESTIÓN DE FLUJO DE TRABAJO.	33

1.9.1 <i>OpenWFE</i>	33
1.8.2 <i>Enhydra Shark</i>	35
1.9.3 <i>Bonita</i>	36
1.9 ANÁLISIS DE LA SITUACIÓN ACTUAL ACERCA DEL CONOCIMIENTO DE LA TECNOLOGÍA WORKFLOW PARA DESARROLLAR EFICIENCIA EN LOS PROCESOS Y ESTANDARIZACIÓN DEL TRABAJO.....	37
1.10 CONCLUSIONES DEL CAPÍTULO	39
CAPÍTULO 2: SISTEMAS WORKFLOW'S (PROPUESTA)	40
2.1 INTRODUCCIÓN	40
2.2 FUNDAMENTOS DEL PROCEDIMIENTO PARA EL USO DE LAS HERRAMIENTAS QUE IMPLEMENTAN TECNOLOGÍA WORKFLOW.....	40
2.3 ANÁLISIS DE LOS SISTEMAS WORKFLOW'S: OPENWFE, BONITA Y ENHYDRA SARK.....	42
2.4 PROFUNDIZACIÓN EN EL ESTUDIO DE LA HERRAMIENTA BONITA WORKFLOW.....	45
2.4.1 <i>Arquitectura</i>	45
2.4.2 <i>Requerimientos</i>	47
2.4.3 <i>Patrones soportados</i>	48
2.4.4 <i>Soporte y licencia</i>	61
2.4.5 <i>Comunidad</i>	62
2.4.6 <i>Madurez</i>	64
2.5 GESTIÓN Y CONFIGURACIÓN DE BONITA WORKFLOW.....	66
2.5.1 <i>Ejecutar Bonita</i>	73
2.6 EVALUACIÓN DE LA PROPUESTA POR ESPECIALISTAS.....	80
2.7 CONCLUSIONES DEL CAPÍTULO	81
CONCLUSIONES GENERALES	82
RECOMENDACIONES	83
BIBLIOGRAFÍA	XI
GLOSARIO	XIII

TABLA DE FIGURAS

FIGURA 1.1 CAPAS DE ADAPTABILIDAD DE WORKFLOW.....	16
FIGURA 1.2 CARACTERÍSTICAS DE UN WORKFLOW.....	18
FIGURA 1.3 MODELO DE REFERENCIA PARA UN SISTEMA ADMINISTRADOR DE WORKFLOW.	20
FIGURA 1.4 CONSTRUCTORES BÁSICOS.....	24
FIGURA 1.5 ARQUITECTURA DE OPENWFE.	35
FIGURA 2.1 ARQUITECTURA DE BONITA.....	46
FIGURA 2.2 DEFINICIÓN DE LA VARIABLE DE AMBIENTE ANT_HOME.	67
FIGURA 2.3 INTERFAZ DE INSTALACIÓN DE JONAS 4.8.4.....	67
FIGURA 2.4 CONFIGURACIÓN DEL ANT Y BCEL (BYTE CODE ENGINEERING LIBRARY).....	68
FIGURA 2.5 SELECCIÓN DEL TIPO DE INSTALACIÓN.	68
FIGURA 2.6 SELECCIÓN DEL DIRECTORIO EN EL CUAL SE VA HA INSTALAR JONAS 4.8.4.	69
FIGURA 2.7 INTERFAZ DE EJECUCIÓN DE LA INSTALACIÓN DE JONAS 4.8.4.....	69
FIGURA 2.8 LEVANTAMIENTO DEL SERVIDOR JONAS 4.8.4.....	70
FIGURA 2.9 INTERFAZ DE BIENVENIDA DE JONAS 4.8.4.	70
FIGURA 2.10 APLICACIONES PUBLICADAS EN JONAS 4.8.4.....	71
FIGURA 2.11 DECLARACIÓN DE LAS VARIABLES DE AMBIENTE NECESARIAS PARA LA INSTALACIÓN DE BONITA 3.1.....	72
FIGURA 2.12 INTERFAZ DE INSTALACIÓN DEL SISTEMA BONITA WORKFLOW 3.1.	72
FIGURA 2.13 CONFIGURACIÓN DEL SISTEMA BONITA WORKFLOW 3.1.....	73
FIGURA 2.14 DEFINICIÓN DEL DIRECTORIO DE SALVAS DEL SISTEMA BONITA WORKFLOW 3.1.	73
FIGURA 2.15 LEVANTAMIENTO DEL SISTEMA BONITA WORKFLOW 3.1.	74
FIGURA 2.16 VERIFICACIÓN DE LA PUBLICACIÓN DEL SISTEMA DE BONITA WORKFLOW 3.1.....	74
FIGURA 2.17 INTERFAZ DE INICIO DEL SISTEMA BONITA WORKFLOW 3.1.	75
FIGURA 2.18 INTERFAZ DE LA HERRAMIENTA XFORMEDITOR.....	78

TABLAS

TABLA 1.1 CRITERIOS DE COMPARACIÓN.	32
TABLA 1.2 MUESTRAS SIGNIFICATIVA POR PROYECTOS.....	38
TABLA 2.1 EVALUACIÓN DE LAS CARACTERÍSTICAS DE LAS HERRAMIENTAS OPENWFE 1.7.1, ENHYDRA SHARK 2.0 Y BONITA 3.1.	43
TABLA 2.2 SOPORTE DE PATRONES EN OPENWFE 1.7.1, ENHYDRA SHARK 2.0 Y BONITA 3.1.....	45

INTRODUCCIÓN

En la actual Batalla de Ideas que desarrolla el pueblo cubano, se hace necesario continuar uniendo los esfuerzos que realizan organismos, organizaciones y entidades para lograr el constante desarrollo de la informática a través de la industria del software, principalmente la modalidad del software libre. Teniendo en cuenta que su empleo está abriéndose paso entre las principales aplicaciones en el mundo de la tecnología. Es imposible pasar por alto la presencia de estas soluciones. Existen muchas herramientas de licencia libre realizadas en JEE³ que brindan servicios de gestión.

El desarrollo priorizado de los programas de la Revolución van en función de que la universidad sea verdaderamente el centro productivo más importante de la comunidad, para este propósito se trabaja con vista a fortalecer la labor formativa, académica e investigativa de sus egresados. En el año 2002 se crea la UCI, la cual tiene como misión: formar profesionales comprometidos con su patria, altamente calificados en la rama de la informática y producir software y servicios informáticos a partir de la vinculación estudio-trabajo como modelo de formación.

La UCI fue creada sobre una fuerte base tecnológica y un amplio perfil productivo, ésta pretende convertir la industria cubana del software en un renglón fundamental de la economía, ser líder del desarrollo de las empresas de software en Cuba e insertarse en el mercado internacional.

La época actual, llamada la revolución científico técnica, necesita hombres capaces de consultar un gran cantidad de información en poco tiempo y utilizar ese caudal de conocimientos en la solución adecuada de los problemas que se plantean de forma creadora, la creatividad del hombre contemporáneo está a prueba cada día cuando debe enfrentar disímiles problemas sociales, científico-técnicos, económicos, ideológicos, entre otros.

Es así como la política de desarrollo sostenido obliga a la industria y a sus productos a competir en el mercado nacional e internacional, provocando la necesidad de introducir tecnología de vanguardia, lo que a su vez ha creado la exigencia de contar con profesionales con independencia cognoscitiva y creatividad.

³ Ver Glosario.

El desarrollo de las tecnologías y su incorporación en las actividades del ser humano ha cambiado y evolucionado de acuerdo a las necesidades de la vida diaria. Una de sus principales metas ha sido el intento por automatizar las actividades cotidianas. Es así como las organizaciones de diferentes sectores realizan sus procesos utilizando estas tecnologías con el fin de obtener dicha automatización y mejorar los resultados en sus procesos y objetivos de negocio.

A lo largo del tiempo los modelos para una mejor gestión de los flujos de trabajo han adquirido vital importancia en la industria del software. En los últimos años estas herramientas han ido evolucionando de forma significativa. Existen organizaciones que a menudo fallan por no comprender que su efectividad puede mejorar drásticamente si los procesos que soportan su operación están bien gestionados y automatizados. El desarrollo de estos motores de gestión facilita los procesos de negocio en una organización, dando como beneficios una notable reducción de costo, mejor ejecución y calidad en los servicios.

La automatización de los procesos es de vital importancia para alcanzar una meta común donde la documentación, información o tareas necesarias para realizar los procedimientos son utilizadas por todos de acuerdo a un conjunto de reglas previamente establecidas que garantice la integración adecuada de todos los recursos lo que permite una verdadera agilidad en el desarrollo de la organización.

Uno de los problemas que se encuentran habitualmente en el desarrollo de aplicaciones para empresas, es que las tareas o procesos que se desarrollan en el entorno laboral de las mismas quedan inmersos en el código de la aplicación que resuelve la problemática de la empresa. Por lo que la gran mayoría de los usuarios no tienen conocimiento de estas tareas, pues están ocultas y se realizan automáticamente.

Entre las nuevas tecnologías para facilitar el crecimiento y mejora de las organizaciones se encuentran los workflow, que son representaciones computarizadas de un proceso de negocio donde se especifican las actividades que están relacionadas con el proceso, el control y orden en que deben ser ejecutadas, el flujo de datos e informaciones y los agentes necesario para su ejecución.

Autores como Wil van der Aalst señalan que el workflow se relaciona con la automatización de los procedimientos donde los documentos, la información o tareas son pasadas entre los

participantes del sistema de acuerdo a un conjunto de reglas previamente establecidas. El fin de lo anterior es llegar a culminar una meta común impuesta por la empresa (1).

Del análisis anterior se infiere que el workflow abarca un conjunto de métodos y tecnologías que ofrece las facilidades para modelar y gestionar los diversos procesos que ocurren dentro de una organización.

La Workflow Management Coalition (en lo adelante WFMC) lo define como: “automatización de un proceso de negocio, de forma completa o en parte, en donde documentos, información o tareas son pasadas desde un participante a otro para que tome acción, de acuerdo a un conjunto de reglas procedurales” (2).

En la definición se evidencia que para la implementación de un proceso de negocio hay que tener en cuenta un grupo de conceptos claves tales como:

Automatización: Mediante una tecnología poder organizar de forma automatizada determinados aspectos del proceso de negocio.

Proceso de negocio: Se define como: “conjunto de uno o más procedimientos o actividades directamente ligadas, que colectivamente realizan un objetivo del negocio, normalmente dentro del contexto de una estructura organizacional que define roles funcionales y relaciones entre los mismos” (2).

Tareas: La unidad lógica de trabajo.

Participantes: Pueden ser usuarios humanos de la aplicación o no.

Acciones: Son las que toman los participantes para poder lograr el objetivo del negocio.

Reglas: Las que rigen los procesos automatizados.

El empleo de la tecnología workflow es importante por los beneficios que aporta, entre ellos:

- El trabajo no queda atascado o extraviado.
- Eficiencia en los procesos y estandarización del trabajo, lo que conduce a una reducción de costos dentro de una empresa:
 - La estandarización de los procesos lleva a tener un mayor conocimiento de los mismos, lo que a su vez conduce a eliminar partes innecesarias en su secuencia y obtener una mejor calidad de estos.

- Con el control de los procesos es posible monitorear el estado actual de las tareas, así como también observar como evolucionan los planes de trabajo realizados.
- La asignación de tareas se realiza mediante la definición de roles dentro de la empresa, eliminando la tediosa tarea de asignar los trabajos caso por caso.
- Se asegura que los recursos de información (aplicaciones y datos) van a estar disponibles para los trabajadores cuando ellos los requieran.
- Mejora del rendimiento y productividad del trabajo de todos los integrantes del proyecto corporativo y reduce las etapas de cada proceso.
- Aumento de la sinergia entre la gestión documental y los flujos de trabajo por la integración entre ambos. Las organizaciones pueden manejar y hacer viajar los contenidos a lo largo de su ciclo de vida por la empresa gracias al workflow para posteriormente archivarlos y acceder a ellos mediante el gestor documental.
- Paralelismo en ejecución de tareas que permite la realización de varias tareas simultáneamente. En muchos procesos resulta necesaria esta simultaneidad.

Toda empresa para mantener e incrementar su cuota de mercado debe adaptarse y evolucionar constantemente, además debe hacerlo con la mayor rapidez posible. Desarrollar el flujo de trabajo a código en los sistemas informáticos tiene el efecto de restarle el dinamismo y agilidad que toda empresa necesita, además de incrementar los costos de mantenimiento.

La tecnología workflow tiene como primer requisito que el proceso automatizado debe ser fácil de modificar sin ayuda de un programador de forma que la barrera del cambio disminuya. Esta incorpora descripciones gráficas de los procesos y la posibilidad de modificar el proceso sobre la marcha.

Muchas empresas desean desarrollar sus propios workflow con todo el esfuerzo que eso conlleva. Siempre existe la tentación de desarrollar soluciones desde cero, confiando en las habilidades que tienen. No aprecian los beneficios que tienen los productos que ya han sido probados y que permiten el desarrollo de funcionalidades que desean sus aplicaciones.

Mantener una independencia es un lujo, toda empresa que han optado este método están ahora tratando de cambiarlos por aquellos estándares de la industria.

Las obras científicas y documentos analizados anteriormente tienen una gran significación en la comprensión de la automatización de un proceso de negocio. A su vez, brindan especial importancia a la optimización del trabajo de una organización empresarial, sin embargo, en la UCI se denotan insuficiencias en el conocimiento de las herramientas para el uso de la tecnología workflow. Si bien se reconoce el uso de los softwares libre como una necesidad, no se aprecian evidencias de que exista una concepción teórica y práctica que refleje la relación del empleo de sistemas workflow y una guía para su posible instalación, encaminada a desarrollar eficiencia en los procesos y estandarización del trabajo.

El diagnóstico preliminar realizado a estudiantes vinculados a proyectos productivos de la facultad 3 corrobora lo anterior. Por otro lado, pocas investigaciones en la UCI se han dedicado explícitamente al estudio de la tecnología workflow.

En consecuencia algunas de las cuestiones que motivaron este trabajo se resumen en los siguientes elementos:

- No es suficiente la información sobre los motores de flujo de trabajo de licencia libre desarrollados en JEE para potenciar eficiencia en los procesos y en la estandarización del trabajo.
- No es factible que las organizaciones tengan que desarrollar sus workflow desde cero, ya que esto implica más tiempo de producción, costo en mantenimiento y generalmente no presentan una buena escalabilidad.
- Los proyectos productivos de la facultad 3 no cuentan con conocimientos generalizados para gestión del flujo de trabajo, algunos de ellos realizan su propia gestión y control de forma empírica, trayendo como consecuencia atrasos en la entrega e insatisfacción del cliente.

Al respecto se plantean las siguientes interrogantes: ¿Se dispone de suficiente preparación para el uso de la tecnología workflow? ¿Es necesario implementar desde cero una herramienta para la gestión de los flujos de trabajos que se encargan de la automatización de los procesos? ¿No es más factible la integración de herramientas estandarizadas que ya han

sido probadas por organizaciones y han dado resultados satisfactorios, permitiendo la adaptación a diferentes cambios que se puedan presentar?

En consecuencia se ha definido como **problema científico**: ¿Cómo contribuir al uso de la tecnología workflow en la facultad para potenciar la eficiencia en los procesos y estandarización del trabajo?

En correspondencia con lo planteado se propone como **objetivo general**: elaborar un procedimiento que estratégicamente contribuya a la materialización de la concepción teórica desde la profundización de un sistema workflow para potenciar eficiencia en los procesos y estandarización del trabajo.

Para ello se han trazado los siguientes **objetivos específicos**:

- Obtener un estudio del estado del arte de los principales motores de flujo de trabajo existentes en la plataforma JEE y sus características.
- Obtener una propuesta de un sistema workflow para su utilización en la facultad.
- Proponer un procedimiento para un mejor conocimiento y utilización de la tecnología workflow propuesta.

Se asume como **objeto de investigación**: las herramientas para la gestión del flujo de trabajo. Como **campo de acción**: Los motores de flujos de trabajo existentes en la plataforma JEE específicamente que sean de licencia libre.

La investigación sostiene como **hipótesis** que si se realiza una propuesta de una herramienta que contribuya a la materialización de la concepción teórica desde la profundización de un sistema workflow, desarrollada sobre la plataforma JEE específicamente que sea de licencia libre, se puede contribuir a la formación profesional sobre el conocimiento de la tecnología workflow en la facultad 3 de los estudiantes y a la integración de esta herramienta en los proyectos productivos.

El alcance del objetivo planteado exigió las siguientes **tareas científicas**:

- Establecimiento de los fundamentos teóricos necesarios para el desarrollo de la tecnología workflow.
- Caracterización del estado actual del problema.

- Elaboración sustentada en una concepción teórica de un procedimiento para el uso de una herramienta que implementa tecnología workflow para potenciar eficiencia en los procesos y en la estandarización del trabajo.
- Valoración del grado de factibilidad del trabajo presentado mediante la consulta de especialistas.

En el desarrollo de la investigación se parte de la dialéctica materialista como método general de investigación y se emplearon otros en el orden teórico, empírico y estadístico, los que estuvieron dirigidos en las siguientes direcciones:

Nivel teórico:

El método de **análisis y síntesis**, se utilizó a lo largo de toda la investigación y sirvió especialmente para procesar la información de las diferentes fuentes bibliográficas y revelar sus aportes y limitaciones esenciales, lo cual permitió sistematizar los fundamentos teóricos – metodológicos de la investigación.

La **inducción - deducción**, para establecer los razonamientos más generales y particulares del objeto de investigación y el campo de acción, así como concretar la elaboración teórica para llegar a nuevas conclusiones.

Del nivel empírico

Observación: presente en las etapas de la investigación para la obtención de información en cuanto al conocimiento de la tecnología workflow, mediante el diagnóstico realizado.

Encuestas: Las encuestas tienen por objetivo averiguar el conocimiento de los estudiantes que participan en proyectos productivos acerca de la tecnología workflow. Se realiza un procedimiento de muestreo y son aplicadas a una parte de los estudiantes. Una de sus ventajas es la enorme rapidez con que se obtienen los resultados.

Cuestionario: elaboración de preguntas para determinar opiniones y puntos de vistas de estudiantes acerca del uso de la tecnología workflow.

Preguntas cerradas: consiste en proporcionar al encuestado una serie de opciones para que escoja una como respuesta. Tienen la ventaja de que pueden ser procesadas más fácilmente y su codificación se facilita.

Métodos y técnicas del nivel estadístico: De la estadística descriptiva se emplearon, la distribución de frecuencias (tablas) y el análisis porcentual, utilizados para el procesamiento de los resultados de diagnóstico.

La **actualidad** del trabajo se corresponde con las necesidades de la universidad y la sociedad cubana de lograr una cultura general integral acorde con los objetivos del país en circunstancias complejas.

El **aporte práctico o resultados esperados** de esta investigación se materializan en la generalización de un conocimiento para mejorar la eficiencia en la gestión y control de los proyectos productivos, dado los problemas que están presentes en el desarrollo de los mismos.

El presente trabajo se encuentra estructurado por dos capítulos, conclusiones, recomendaciones, bibliografía y anexos.

Capítulo 1: Fundamentación teórica. En este apartado se pretende proporcionar los aspectos principales sobre concepto y definiciones, así como el estudio del arte referente a las distintas herramientas para la gestión de flujo de trabajo. Se hace una valoración del resultado obtenido del cuestionario realizado.

Capítulo 2: Herramienta para la gestión de flujo de trabajo (Propuesta).

Se hace un estudio de los sistemas workflow's mencionados en el capítulo anterior, obteniendo la herramienta propuesta. Se desarrolla una guía para un mejor entendimiento e implementación de la herramienta propuesta.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se fundamentan los principales aspectos de los motores de flujos de trabajo existentes en la plataforma JEE específicamente que sean de licencia libre, así como las definiciones y conceptos relacionados con esta tecnología y el análisis de varios productos de este tipo, además se realiza una valoración del estado actual del problema.

1.2 Caracterización del software libre.

El Software Libre (en lo adelante SL) constituye una mirada distinta a la industria del software, donde el modelo predominante es el modelo de software propietario (o privativo) y que se caracteriza por la utilización de licencias de software pagadas, como modelo de negocios, estas licencias pueden costar importantes sumas de dinero, solamente dan derecho al uso de los programas bajo una serie de importantes restricciones, especialmente en lo referente a la redistribución y modificación de los programas. Además estos softwares privativos son desarrollados por empresas de distintos tamaños que van desde pequeñas empresas locales hasta grandes corporaciones internacionales.

Por el contrario, el SL posee un modelo de licenciamiento sumamente distinto, donde se entrega una gran cantidad de derechos al usuario, especialmente en lo referente a modificación y redistribución, donde los programas se inician en función de necesidades particulares de los propios usuarios y se van desarrollando en forma distribuida en distintos lugares del mundo a través de Internet. Más formalmente, el SL se puede definir como: “Software Libre (en inglés free software) el que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El SL suele estar disponible gratuitamente en Internet, o a precio del costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y aunque conserve su carácter de libre, puede ser vendido comercialmente” (3).

1.2.1 Software de código abierto (OPEN SOURCE)

El concepto de código abierto (Open Source) surge de una reunión de estrategia efectuada en Palo Alto, California, como resultado del anuncio realizado por parte de la empresa Netscape, en enero de 1998, acerca de la liberación del código fuente del programa “Navigator”. El

objetivo de este grupo era aprovechar esta oportunidad de la liberación del código fuente del Navigator para liberarse de las connotaciones ideológicas y confrontacionales del término “free software”. Netscape licenció y dejó disponible el código fuente bajo Open Source, con el nombre de Mozilla. En febrero de 1998, Eric S. Raymond y Bruce Perens crean la Open Source Initiative (OSI). Con el respaldo de cerca de 20 años de experiencia sobre desarrollo de software cerrado contra el desarrollo de software abierto, la OSI continúa hasta el día de hoy incentivando y promocionando el Open Source principalmente en los entornos empresariales, su labor se centra en destacar y dar un perfil más alto a los beneficios de contar libremente con el código fuente y buscan incorporar a las grandes empresas de software y otras industrias de alta tecnología al Open Source.

En términos prácticos, Open Source y SL, no presentan diferencias de importancia, salvo aspectos más de tipo filosófico, principalmente respecto a la “libertad” del software y el uso comercial de los programas. Por esta razón, se ha popularizado el uso de los conceptos como FOSS⁴ (Free and Open Source Software) y FLOSS⁵ (Free/Libre/Open Source Software) (4).

1.2.2 Licenciamiento de Software Libre

De acuerdo con Richard Stallman y la Free Software Foundation⁶, las licencias de software deben cumplir con cuatro libertades para ser calificados como “libres” (5):

- La libertad de ejecutar el programa para cualquier propósito.
- La libertad para estudiar y modificar el programa.
- La libertad para copiar el programa, y así, poder ayudar a tu vecino.
- La libertad para mejorar el programa y dejar disponibles las mejoras al público, de forma que toda la comunidad se beneficie.

Las libertades 1 y 3 requieren de acceso al código fuente, puesto que el estudio y la modificación del software sin su código fuente son en extremo difícil, altamente ineficiente y

⁴ Ver Glosario

⁵ Ver Glosario

⁶ Free Software Foundation: La Fundación para el Software Libre (FSF) está dedicada a eliminar las restricciones sobre la copia, redistribución, entendimiento, y modificación de programas de computadoras. Con este objeto, promueve el desarrollo y uso del software libre en todas las áreas de la computación.

usualmente imposible en la práctica, el acceso al código fuente comentado resuelve este problema.

Por su parte OSI define como Open Source los programas que aseguran los siguientes derechos y obligaciones (4):

- No se debe exigir pago o cualquier otro tipo de cobro por la redistribución.
- Disponibilidad del código fuente.
- Derecho a realizar modificaciones y crear versiones derivadas.
- Puede requerir que las versiones modificadas sean distribuidas como la versión original más parches.
- No pueden existir discriminaciones hacia personas o grupos.
- No pueden existir discriminaciones hacia ciertos ámbitos o entornos de uso.
- Todos los derechos debe fluir hacia las versiones redistribuidas.
- La licencia se aplica al programa como un todo y para cada uno de sus componentes.
- La licencia no puede restringir otro programa, con lo que se permite la distribución de programas de código abierto y código cerrado en forma conjunta.

La (5) y (4) publican listas de las licencias que ellos han analizado y cumplen con sus definiciones de SL y Open Source respectivamente. La mayor parte de los SL disponibles utilizan un conjunto pequeño de licencias, las más populares son: la GNU General Public License (GPL), la GNU Lesser General Public License (LGPL), la BSD License, la Mozilla Public License, la MIT License y la Apache License.

1.3 La Plataforma JEE

Para conocer acerca de JEE hay que tener una idea de lo que es java, esta tecnología se crea como una herramienta de programación en una pequeña operación denominada “*The Green Project*” en Sun Microsystems⁷, en el año 1991. El equipo llamado “Green Team” estaba compuesto por trece personas y dirigido por James Gosling (6). Hoy día java está presente en numerosos sectores de la actualidad y millones de programadores lo utilizan.

Debido a su portabilidad, seguridad y eficiencia está presente en una gran cantidad de dispositivos, ordenadores y redes. Java se caracteriza por compilar a un código intermedio o

⁷ Ver Glosario.

bytecode, el cual es interpretado por una máquina virtual de java. La máquina virtual de java permite que una aplicación que haya sido implementada en java se ejecute en cualquier sistema operativo con soporte para la máquina virtual. También reúne todas las características de un ambiente orientado a objetos: es sencillo, cuenta con capacidad de generación de aplicaciones distribuidas, robustas, seguras, de arquitectura neutral, portables, multihilo, dinámicas y de alto rendimiento.

Numerosas empresas desarrollan para la plataforma java, entre ellas se encuentran IBM⁸, Sun, Oracle, JBOSS las cuales aportan componentes.

Unas de sus brillantes características es cuando Sun hace open source sus implementaciones java, trayendo como consecuencia un alto nivel de nuevas oportunidades en el mercado, implicando el incremento de distintas innovaciones.

Cuando una persona o entidad cree que es necesaria la presencia de una determinada tecnología dentro de las plataformas basadas en java, lo que hace es crear un JSR (Java Specification Request) y presentarlo para su aprobación. El contenido del documento describe porque es necesaria dicha tecnología y en caso de aprobarse se crea una especificación, un documento en el cual se describe dicha tecnología, sus partes, las relaciones entre las mismas y los roles de las personas que usarán dicha tecnología. Además el equipo encargado del desarrollo de la especificación ha de proporcionar un test de compatibilidad y una implementación de referencia.

Para brindar la posibilidad a todo el mundo de participar en la evolución de java y de todas las plataformas que se basan en java se creó JCP, el cual es un organismo formado por más de 500 empresas y asociaciones que aseguran la evolución de las plataformas basadas en java y además es el encargado de decidir que especificaciones se aprueban y de controlar las fases por las que pasan.

Se define a JEE como una especificación, un JSR (concretamente el JSR-151), que precisa una plataforma de desarrollo empresarial, a la que se llama la plataforma JEE. Esta plataforma está formada por varios componentes:

- Conjunto de especificaciones.
- Test de compatibilidad, el JEE Compatibility Test Suite (CTS).
- Implementación de referencia de JEE.

⁸ Ver Glosario.

- Conjunto de guías de desarrollo y de prácticas aconsejadas denominadas JEE BluePrints.

La plataforma JEE define un modelo de programación encaminado a la creación de aplicaciones basadas en n-capas. La lógica de la aplicación se divide en componentes de diferentes funciones que componen una aplicación JEE y que están distribuidos en dependencia de la capa en el ambiente multi-capas JEE al cual la aplicación pertenece.

Entre las ventajas de aplicar dicho modelo tenemos que al tener las capas separadas existirá poco acoplamiento entre las mismas, facilitando las modificaciones entre ellas y obteniendo mejoras en cuanto a mantenibilidad, extensibilidad y reutilización de componentes.

1.4 Estándares y organizaciones

Una de las estrategias que siguen actualmente las empresas para mejorar la calidad que brindan, es rediseñar sus procesos. Esta metodología es denominada Reingeniería de los Procesos de Empresas (Business Process Re-engineering) que puede ser causada por los cambios organizacionales, legislativos o cambios en los objetivos del negocio. En esta situación, muchas veces es necesario relacionarse con otras organizaciones. Pero para poder hacer esto debe existir la posibilidad de que los productos de un vendedor puedan comunicarse con los de otro, pues es claro que cada empresa u organización comprará los productos que crea conveniente para su caso.

Se estima que actualmente los distintos productos de workflow que hay en el mercado sobrepasan los cien. Cada uno de ellos se enfoca sobre distintos aspectos funcionales, herramientas de diseño visual, en las cuales se ofrecen ciertos diagramas para representar la realidad, otras se enfocan en la integración de los datos con las aplicaciones.

El desarrollo de estándares para la interoperabilidad de las diversas herramientas, permite la elección de los mejores productos, según el enfoque que se le da a la aplicación. Por ejemplo, se puede comprar una herramienta que ofrezca un entorno amigable para realizar el análisis y diseño del problema y por otro lado comprar un componente que resuelva la auditoría de los datos y poder trabajar en forma integrada con los dos componentes.

Es necesario entonces estandarizar la forma de comunicación entre los distintos componentes de un producto de workflow para poder tener flexibilidad a la hora de operar con distintos

productos. Esta necesidad se justifica además por las proyecciones que se tienen actualmente, sobre la tecnología de workflow en el mercado. Por esto se debe atacar el problema de incompatibilidades de antemano, y no cuando existan miles de productos en esta área, cada uno con sus particularidades.

La **WfMC** es una agrupación compuesta por compañías, vendedores, organizaciones de usuarios, y consultores. El objetivo de esta agrupación es ofrecer una forma de “diálogo” común a todos. De esta forma las diferentes herramientas que se implementen en esta área podrán tener cierto nivel de interoperabilidad, es decir, podrán comunicarse entre ellas para poder realizar las distintas tareas involucradas en un sistema de workflow (2).

Asociación Internacional de Reingeniería y Workflow (WARIA).

La organización WARIA, establecida en 1992 tiene como misión interpretar y analizar el desarrollo de los workflow's y el comercio electrónico, compartir experiencias, evaluaciones de productos, la educación y conexión entre los usuarios y vendedores. Esta organización trabaja estrechamente con la WFMC.

También identifica y aclara las publicaciones que son comunes para los usuarios de workflow, comercio electrónico y los que están en el proceso de reingeniería en sus organizaciones. La asociación facilita oportunidades para que los miembros puedan comunicarse y compartir experiencias libremente (7).

El **Grupo de Gestión de Objetos (OMG)** se formó en 1989, con el propósito de crear una arquitectura estándar para objetos distribuidos en redes (componentes). En octubre de 1989 comenzó a funcionar como una corporación independiente y sin ánimo de lucro. El compromiso asumido por el OMG busca el desarrollo de especificaciones para la industria del software que sean técnicamente “excelentes”, comercialmente viables e independientes del vendedor. Hoy en día el consorcio incluye unos 800 miembros (compañías en la industria del software). La primera arquitectura resultante de esta colaboración fue CORBA⁹ (Common Object Request Broker Architecture). El elemento central de esta arquitectura es el ORB (Object Request Broker). Un ORB permite que un objeto cliente haga una petición a un

⁹ CORBA: es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

servidor sin conocer la interfaz ni la localización del objeto servidor. OMG define gestión de objetos como el desarrollo de software que modela el mundo real mediante su representación como objetos. Estos objetos son la encapsulación de atributos, relaciones y métodos de componentes de software identificables (8).

La **Notación de Modelado de Procesos de Negocio (BPMN)** implementa una notación de modelo para procesos, concretamente el conjunto original de especificaciones propuestas por Business Process Management Initiative (BPMI¹⁰), ahora parte del OMG. Se trata de una notación gráfica de los pasos y actividades de un proceso de negocio. Modela tanto la secuencia de actividades como los datos o mensajes intercambiados entre los distintos participantes. BPMN no está pensado para modelar aplicaciones, sino procesos que correrán dentro de dichas aplicaciones. Por ello, la salida de BPMN necesita ser expresada en algo que no sea un lenguaje programático. Es aquí donde entra en juego BPML (metalenguaje desarrollado por BPMI). Así, a través de BPML disponemos de la traducción de un formato gráfico (para ser leído por personas) a un formato legible por máquinas (9). En ausencia de BPML, nos encontramos con BPEL y XPD L como soluciones a esa traducción.

El **Lenguaje de Ejecución de Procesos de Negocio (BPEL)**, es un lenguaje ejecutable con sus variables y operaciones. Las operaciones permiten enviar y recibir mensajes SOAP¹¹ y tiene un gran aporte para XML y transformaciones XML. Además, permite de manera fácil invocar múltiples servicios web al mismo tiempo y sincronizar los resultados. BPEL no tiene soporte gráfico, es decir, no especifica como deben ser los diagramas interpretados de los procesos que define. El objetivo de BPEL es brindar una forma de orquestar servicios web, la secuencia de interacciones subyacente y el flujo de datos punto a punto (10).

El **Lenguaje de Definición de Procesos XML (XPD L)**, especifica un formato de diseño para los procesos y permite una representación gráfica de los procesos incluyendo coordenadas X

¹⁰ Ver Glosario.

¹¹ (SOAP) Simple Object Access Protocol. Este protocolo deriva de un protocolo creado por David Winer, XML-RPC en 1998. En su sitio web <http://www.userland.com> se puede encontrar multitud de documentación acerca de este primer protocolo de comunicación bajo http mediante XML.

e Y para cada nodo implementado. Además los nodos pueden especificar atributos tales como roles, descripción de actividades o llamadas a servicios web. Suele ser preferido cuando se trata de implementar procesos o workflows con interacciones humanas, ya que en esto supera a BPEL (para soportar interacciones humanas, BPEL necesita de añadidos a la especificación, lo cual dificulta su portabilidad ya que suelen ser implementaciones propietarias de cada proveedor). El objetivo de XPD L es almacenar y permitir el intercambio de diagramas de procesos y ofrecer una manera estándar para representar procesos de tal manera que puedan ser importados/exportados por cualquier editor que implemente el estándar (11).

1.5 Workflow

Entre las nuevas tecnologías que han surgido en los últimos años, una de las que más ha contribuido a facilitar el crecimiento y mejora de las organizaciones ha sido la de workflow.

Habitualmente los sistemas de gestión de las organizaciones dan el soporte administrativo a las tareas que se desarrollan en ellas, escondiendo los procesos dentro del código de los sistemas informáticos. El nivel de dinamismo de una empresa moderna hace que los sistemas basados en la tecnología tradicional tengan un alto costo de mantenimiento y desarrollo, ya que al tener que recurrir a la codificación para ajustar sus cambios operativos, hace que esta tarea sea no solo costosa, sino también lenta y difícil.

También los ocultamientos de los procesos en los sistemas de gestión hacen que los usuarios tengan poco conocimiento de que las tareas que están realizando es parte de un proceso definido por la organización.

La solución a este problema, es automatizar las diversas tareas procesadas en la organización y conectarlas en un proceso que sea claro para los usuarios y fácil de modelar y gestionar para la organización. La tecnología que hace esto posible, es la tecnología de workflow.

Los sistemas de flujo de trabajo, por su propia naturaleza son soluciones que deben ser capaces de integrar en el flujo de información la diversidad de datos que la empresa almacena en diferentes sistemas y de mostrarlos al usuario desde una interfaz única. Las tecnologías de flujo de trabajo han surgido como consecuencia de la general evolución hacia sistemas distribuidos y del desarrollo de herramientas de definición de automatización de procesos. No

significan como tal una nueva tecnología, sino una nueva forma de emplear los sistemas informáticos para trabajar con los procesos de la empresa y comunicar sistemas aislados (12). Los sistemas de flujo de trabajo integran los aspectos de "datos" de la información junto con los de "procesos". El flujo de trabajo no automatiza solamente las tareas que se están realizando, sino que también incide en la mejora y reordenación de las mismas. Los sistemas de flujo de trabajo se pueden implantar en forma de pequeños proyectos de relativo bajo coste y resultados observables a corto plazo, lo que les convierte en soluciones muy adecuadas para planteamientos de reorganización de métodos de trabajo en la organización.

El término workflow es definido por la WfMC como: "automatización de un proceso de negocio, de forma completa o en parte, en donde documentos, información o tareas son pasadas desde un participante a otro para que tome acción, de acuerdo a un conjunto de reglas procedurales" (13).

En esta investigación los autores se afilian a este concepto que precisa que un workflow es el flujo automatizado de un proceso de negocio, en el cual se definen un conjunto de actividades bien relacionadas y estructuradas entre sí, que son realizadas por miembros de la organización ya sean personas o aplicaciones con el fin de lograr un objetivo común siguiendo unas reglas de negocio preestablecidas.

Algunos aspectos que se destacan en la definición:

- Cualquier conjunto de actividades: Amplia gama de actividades relacionadas con el negocio y su administración.
- Objetivo común: Los individuos que participan en un flujo de trabajo deben tener un objetivo común para que se pueda desarrollar las actividades de la mejor forma posible. No pueden trabajar en proyectos independientes.
- Reglas de negocio: Si un proceso no sigue unas reglas y ruta preestablecidas, no se trata de workflow, sino de un proceso de colaboración.

La automatización de los procesos.

Es importante profundizar en las razones por las cuales se automatiza un proceso entre las que se encuentran: reducción de costos, incremento de eficiencia, menores oportunidades de error, mejor control, calidad en beneficio de todos.

Por ser inherentes a la administración de toda organización los procesos se consideran como si tuvieran "costo cero". Nada más alejado de la realidad; cuesta el papel, el tiempo, el espacio, la administración logística y la oportunidad que sus colaboradores estratégicos se dediquen a mejores maneras de lograr sus objetivos. Las empresas que han automatizado sus procesos administrativos han descubierto nuevas fuentes de ahorro y oportunidades de mejorar la calidad de su gestión y la satisfacción de las expectativas de calidad de sus clientes:

- Disminuyendo costos asociados al papel, producción, almacenamiento y transporte de formas, formularios y documentos.
- Reduciendo el tiempo de procesamiento, ahorrando en horas hombre y obteniendo resultados en menor tiempo.
- Disminuyendo las posibilidades de incumplimiento, error y fallas por pérdida o desaparición de papeles.
- Mejorando la calidad y oportunidad de la información necesaria para la realización de actividades fundamentales del negocio.
- Permitiendo a la gerencia concentrarse en lo que es realmente productivo para la organización.

1.5.1 Conceptos y definiciones

Para un mejor entendimiento sobre esta tecnología workflow se hará un breve recorrido por algunos de sus conceptos más importantes los cuales servirán para entender la importancia de esta tecnología y como el éxito o el fracaso de un sistema workflow se basa en la claridad de la definición de sus procesos.

Caso: Es un elemento físico o abstracto el cual es producido o modificado por algún tipo de trabajo. Los siguientes son sinónimos de un caso: Trabajo, Job, Producto, Servicio, Item. Un caso siempre tiene un inicio y un fin.

Proceso. Un proceso consiste de un número de tareas que necesitan ser realizadas y un conjunto de condiciones que determinan el orden de las tareas.

Tarea. Las tareas son unidades de trabajo formadas por un conjunto de acciones o actividades, que son realizadas por uno o más recursos (un recurso puede ser una persona o una máquina) en un intervalo de tiempo predeterminado. Una tarea es atómica, lo cuál quiere decir que no es divisible en tareas más pequeñas (14). Es importante este concepto de tarea ya que para la fase de modelado es necesario tener previamente identificadas las tareas, esto se logra a través del análisis del proceso como un flujo de trabajo.

Las tareas pueden ser manuales, automáticas y semiautomáticas. Las tareas manuales son realizadas exclusivamente por personas. Por el contrario en una tarea automática no existe participación alguna de personas. Mientras que en las tareas semiautomáticas, se tiene una combinación de las dos anteriores.

Enrutamiento. En un proceso de workflow la información fluye a través de los distintos elementos que conforman el proceso. Es así como la información puede tomar diferentes rutas hasta llegar a completarse. Lo anterior es conocido como flujo de control del workflow; para representar esas posibles rutas se han definido los siguientes constructores básicos en el modelo de referencia de workflow de la WfMC.

Secuencia. Este constructor modela la situación más natural y simple dentro de un proceso workflow, modela una tarea que se ejecuta sólo hasta que la tarea que le precede ha sido ejecutada.

- OR-Split. Es un punto en el workflow donde al final de un único hilo de control se toma una decisión con la cual se bifurcará cuando se encuentra con múltiples alternativas.
- AND-Join. Es un punto en el workflow donde dos o más actividades que se ejecutan en forma paralela convergen, entonces gracias a esa convergencia, otra actividad puede ser iniciada.
- OR-Join. Es un punto en el workflow donde dos o más alternativas para continuar dentro del workflow re-convergen en una simple actividad como el siguiente paso a seguir en el workflow.

Datos. Los datos son los documentos, archivos, imágenes, registros de la base de datos, y otros utilizados como información para llevar a cabo el trabajo. Entre los datos manejados por el workflow encontramos:

- Datos de control: son los datos internos manejados por la lógica del sistema de workflow.
- Datos relevantes: son aquellos datos utilizados para determinar el ruteo de las distintas tareas del sistema.
- Datos de la aplicación: estos datos son específicos de la aplicación, no son accedidos por la lógica del workflow.

En cuanto a los datos, es muy utilizada la noción de documento como recipiente de información que se transmite de una tarea a otra. Por esta razón, cuando se hace referencia a datos manejados por el sistema, es posible referirse a estos datos como documentos. Existen ciertas propiedades que se le pueden asociar a un documento, tales como: la definición de los derechos de acceso a los mismos; las vistas definidas sobre ellos; manejo de accesos concurrentes (es decir, que dos personas o procesos puedan acceder al documento simultáneamente); también es posible definir formas de relacionar datos provenientes de fuentes externas al documento, tales como datos de la aplicación o de la base de datos.

Personas. Como se explicó anteriormente las tareas son realizadas por recursos, ya sean máquinas o seres humanos, es así como las personas dentro de una organización son parte importante en lo que a cumplimiento de los procesos se refiere. En muchos procesos las tareas son realizadas en un orden establecido previamente por ciertas personas de acuerdo a su capacidad o rol que desempeñe dentro de la empresa, además de realizarse en base a las condiciones y reglas de negocio. Una persona es conocida como un actor dentro del escenario de un workflow, sin embargo como actor también puede referirse a una máquina.

Roles. Cada rol define las distintas competencias potenciales que existen en el sistema. Se definen independientemente de las personas físicas a las cuales se les van a asignar dichos roles. Una persona puede tener más de un rol, siendo un rol un conjunto de características de un actor. En el caso de actores personas, un rol se puede referir a determinadas habilidades, responsabilidades o autoridad que ésta posea.

Grupos. Es un conjunto de actores que realizan el mismo rol. Cuando se asignan tareas a un grupo de actores en vez de a un solo actor, se puede tener mayor flexibilidad en la selección de quien puede realizar la tarea, teniendo así una división de las labores más eficiente.

Eventos. Un evento es una interrupción que contiene información, éste tiene un origen y uno o más destinatarios. La información contenida en el mensaje que se produjo por el evento puede ser implícita o dada por el usuario. Los eventos pueden ser disparados voluntariamente por el usuario; o en forma implícita durante un proceso según el estado de los datos o de decisiones tomadas por el usuario; o en forma automática. Por ejemplo, cuando un gerente de un banco hace una consulta sobre ciertos datos para hacer una auditoria, se dispara un evento que le devuelve la información sobre dicha consulta.

Plazos. Son los tiempos que se le asignan a ciertas tareas para que éstas sean realizadas o estén listas antes de tomar una determinación diferente a la normal, como podría ser su cancelación o la finalización del proceso mismo. Podemos considerar los siguientes ejemplos de plazos: el tiempo máximo que se le asigna a una tarea para que sea terminada, el tiempo máximo para recorrer una ruta; terminar una tarea antes de cierta fecha, entre otras más.

A los plazos podemos asignarles eventos, de forma que con el incumplimiento del plazo se dispare dicho evento de forma automática.

1.5.2 Tipos de workflow's

Debido a la diversidad de procesos de negocio que existen dentro de las empresas, se tiene la siguiente clasificación para los workflow's (15):

- De Producción.
- De Colaboración.
- Administrativo.

El **workflow de producción** es frecuentemente llamado workflow de transacciones debido a que la transacción en una base de datos es considerada la clave de todo proceso. Este tipo de workflow es el más común en el mercado y se caracteriza por automatizar procesos de negocios que tienden a ser repetitivos, bien estructurados y con gran manejo de datos.

Las aplicaciones de workflow que resuelven procesos de negocios donde participa gente para lograr una meta común, son llamadas workflow de colaboración.

Los **workflow de colaboración** estructuran o semi-estructuran procesos de negocios donde participan personas, con el objetivo de lograr una meta común. Típicamente involucran documentos, los cuales son los contenedores de la información. Se sigue la ruta de estos

paso a paso, además de las acciones que se toman sobre ellos. Los documentos son la clave, y por lo tanto es esencial para la solución de workflow mantener la integridad de dichos documentos.

El **workflow administrativo** como lo dice su nombre es aquel que involucra procesos de administración en una empresa tales como órdenes de compra, reportes de ventas. Estos workflow's se emplean cuando existe una gran cantidad de procesos administrativos dentro de la empresa y es necesaria la distribución de soluciones a diferentes usuarios. Una solución de workflow administrativo difiere para cada organización, y los cambios son frecuentes. Por esto, la posibilidad de poder hacer cambios de diseño es muy importante.

Además de estos tres tipos de workflow debemos hacer mención sobre un workflow caótico que definimos como **ad hoc workflow** y que podemos interpretar como que algún usuario puede definir o tomar decisiones dentro de la aplicación en cualquier momento, esto es, en una empresa los empleados pueden mandar información a clientes en forma desorganizada y no estructurada, por ejemplo, vía correo electrónico.

Hay quienes consideran que ad hoc es un tipo de workflow ligado al workflow administrativo o como simplemente una propiedad que puede o no tener una aplicación de workflow, sea el workflow de producción, el de colaboración o el de administración.

1.5.3 Adaptabilidad y workflow

Las actividades de negocio son dinámicas, sujetas a una evolución constante debido a requerimientos que incrementan la competitividad de las empresas, rediseño y optimización de los procesos de negocios existentes.

La figura 1.1 muestra diferentes capas de adaptabilidad de workflow propuestas en (16).

En un nivel inferior se encuentra los cambios en la infraestructura de los sistemas, seguido de los cambios en los recursos tales como los componentes de software, modelo de datos y los recursos de la organización de negocio. Los sistemas también pueden sufrir cambios en cuanto a los procesos que ejecutan y también en el dominio en donde funcionan.

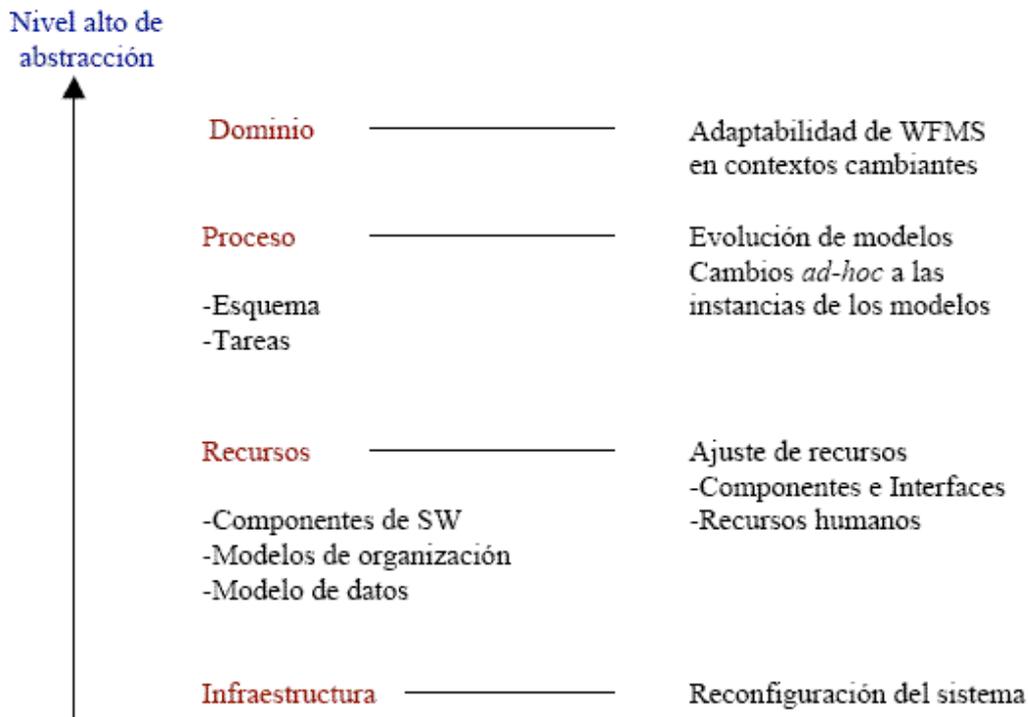


Figura 1.1 Capas de adaptabilidad de workflow.

Dominio

En el nivel de dominio un sistema workflow puede ser considerado como un solo componente afectado por los cambios que sufra el contexto en donde se encuentre, lo que da como resultado una serie de requerimientos de adaptabilidad dentro del sistema (definición de procesos, modelo de datos).

Procesos

El nivel de adaptabilidad de procesos trata con los cambios relativos a los modelos implementados por los workflow's. Modelos que les permite definir procesos. En este nivel los cambios de los modelos se deben llevar a cabo paralelamente al de los procesos de negocios.

Recursos

La modificación de recursos se refiere a los cambios y reajuste que sufren los soportes del sistema workflow como la sustitución y modificación de los componentes de las interfaces del software, la modificación de las estructuras de datos, así como los cambios en los recursos de la organización.

Infraestructura

Esta adaptabilidad surge en respuesta a la evolución de requerimientos y avances técnicos. Los sistemas necesitan adaptarse rápidamente a un ambiente de negocio modificado o a un cambio técnico, resultando una nueva configuración del sistema. Se requiere de arquitecturas flexibles que permitan a sus componentes de software ser modificados o reemplazados sin que esto afecte la ejecución de un proceso.

1.6 Sistema de administración de workflow

Para implementar un workflow, se requiere un Workflow Management System (en lo adelante WfMS): *"Sistema que define, crea y administra la ejecución de workflow's mediante el uso de software, ejecutando uno o más motores de workflow, los cuales son capaces de interpretar la definición del proceso, interactuar con los participantes del workflow, y de ser requerido, invocar el uso de herramientas de tecnologías de información y aplicaciones"*(14). El WfMS es el que le da la "vida" al workflow, definiendo y coordinando los participantes (humanos o no) y sus permisos, los motores de workflow, las excepciones y recursos, con el objetivo de cumplir el objetivo de negocio en cuestión.

Entre los beneficios que nos brinda el WfMS tenemos:

- Marco adecuado para la reingeniería.
- Modelado global de todos los procesos.
- Marco para la ejecución de procesos.
- Mejora y establecimiento de controles de calidad.
- Adaptabilidad ante cambios.
- Aumento de la velocidad de procesamiento.
- Mejora en manejo de documentos información.
- Mejora en el trabajo del personal.

1.6.1 Características

En su nivel más alto todos los WfMS proveen soporte en tres áreas funcionales:

- Las funciones en tiempo de diseño (Build-Time) relacionadas con la definición y modelado de cada proceso y actividades que le constituyen.

- Las funciones de control en tiempo de ejecución (Run-Time) relacionadas con la administración del proceso de workflow en un ambiente operacional y secuenciando las diversas actividades a ser mejoradas como parte de cada proceso.
- Las interacciones en tiempo de ejecución entre personas, aplicaciones y tecnologías de información.

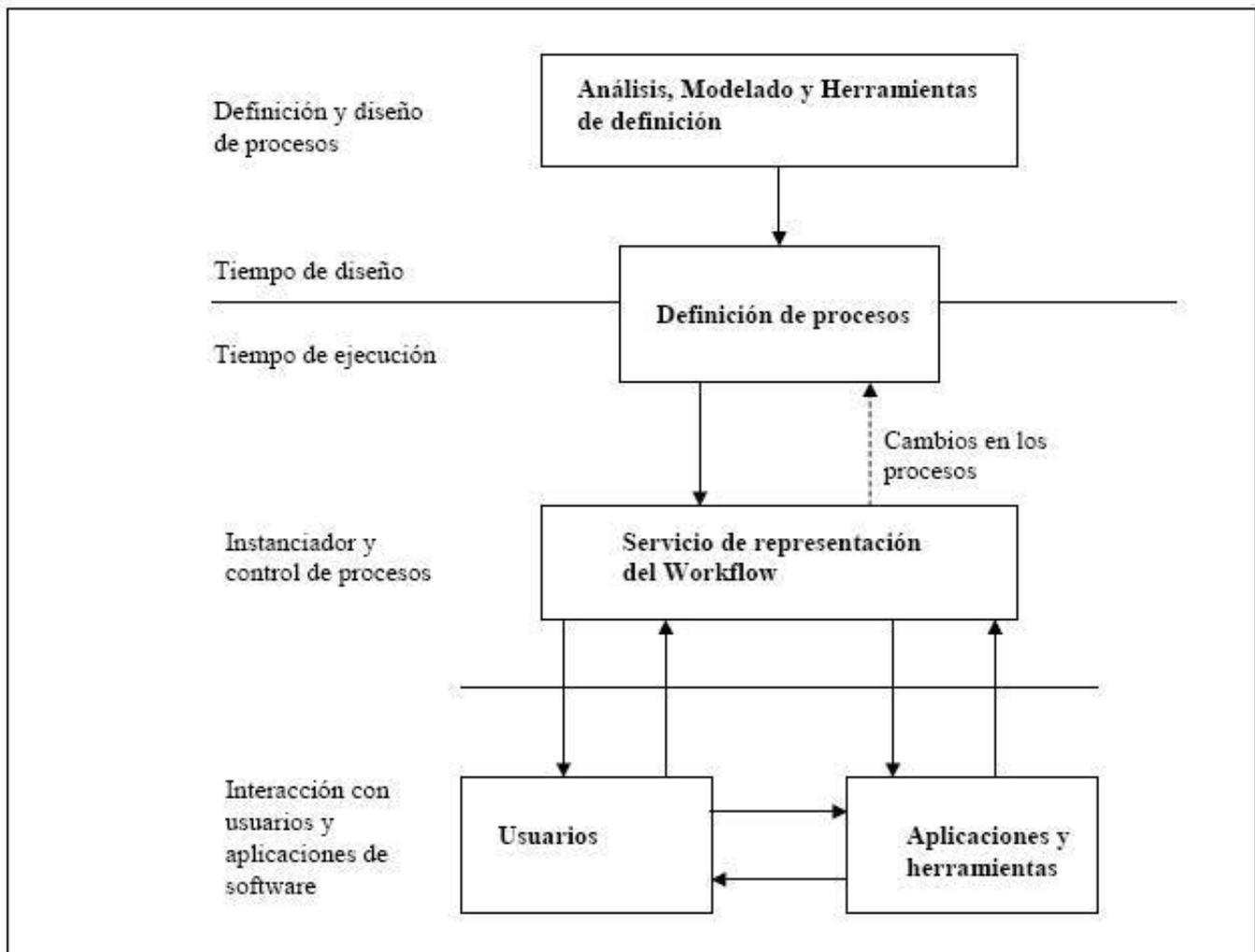


Figura 1.2 Características de un workflow.

1.6.2 Funciones y beneficios de los sistemas workflow's.

Entre las numerosas funciones más comunes se pueden citar las siguientes:

- Asignación de tareas al personal.
- Aviso al personal de tareas pendientes.

- Permitir la colaboración en las tareas comunes.
- Optimización de recursos humanos y técnicos alineándolos a la estrategia de la empresa.
- Automatización de las secuencias de los procesos de negocio y optimización de las mismas.
- Agilización de los procesos de negocio y dando por resultado, un mejor servicio al cliente.
- Control y seguimiento de dichos procesos.
- Modificación de instancias de procesos sobre la marcha.

Estas funciones repercuten en diversos beneficios para la organización, como:

- Mejora en servicio y atención al cliente.
- Incremento en la ejecución de actividades en paralelo.
- Minimización en el tiempo para acceso a documentación, aplicaciones y bases de datos.
- Disminución del tiempo de transferencia de información entre tareas.
- Se asegura la completa colaboración del personal de trabajo.
- Permite tener conocimiento inmediato del estado de alguna tarea o proceso.
- Mejora en la gestión y optimización de procesos.

1.6.3 Modelo de referencia de workflow.

La WFMC conformada por numerosas compañías de software, instituciones académicas y de investigación tuvo como sus primeros objetivos de trabajo establecer una arquitectura de referencia para los WfMS, pensando en la adaptación de los sistemas existentes así como para los nuevos desarrollos, con el fin de lograr la interoperabilidad entre estos sistemas. El modelo de referencia de workflow (17) fue presentado en su primera versión en 1994, y ha tenido muy pocas modificaciones desde su elaboración.

La figura 1.3 muestra la arquitectura para un WfMS sugerida en el modelo de referencia de workflow, y describiremos la forma en que interactúan los elementos que componen a dicho modelo (1).

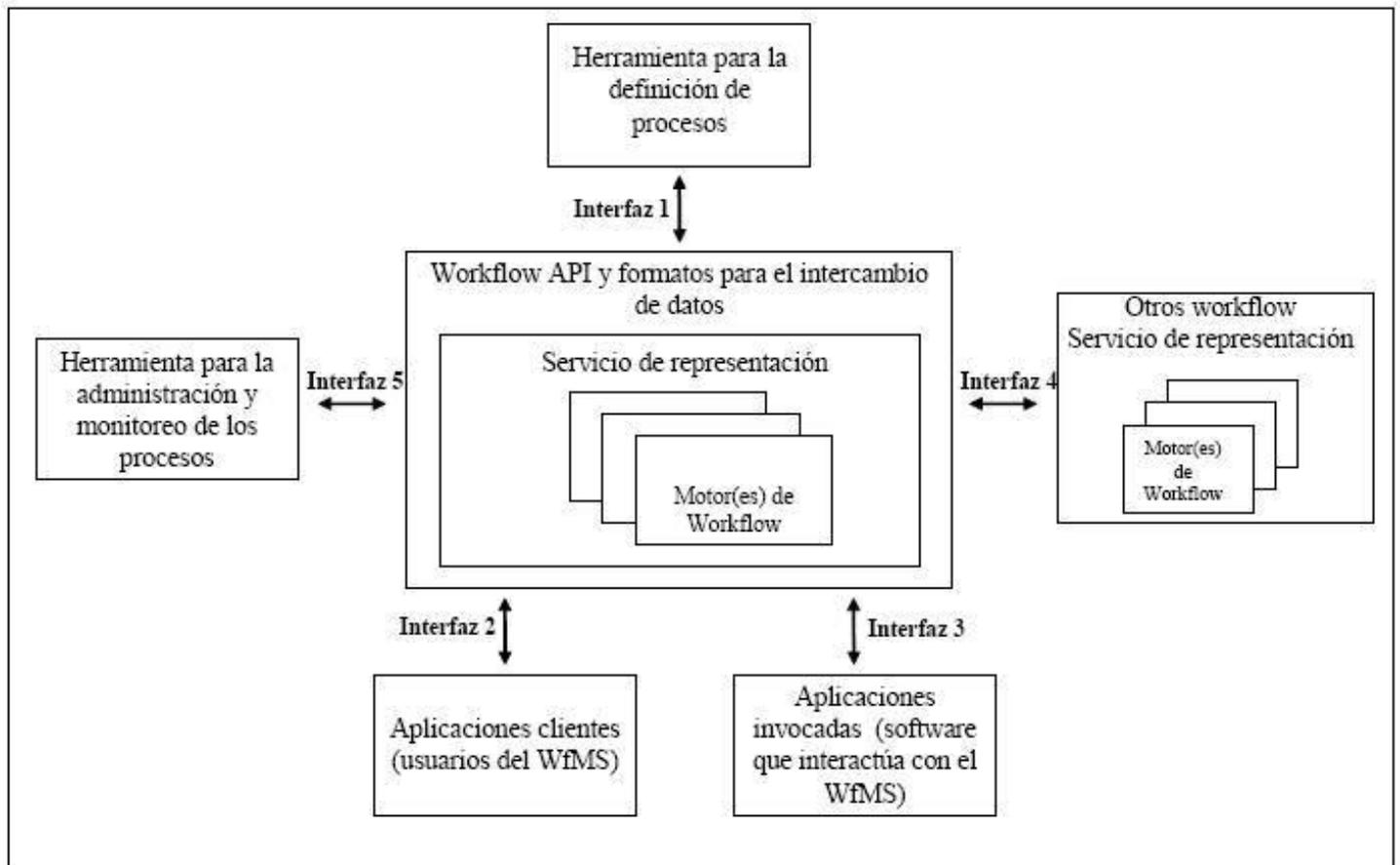


Figura 1.3 Modelo de referencia para un sistema administrador de workflow.

Herramienta para la definición de procesos de workflow.

La herramienta para la definición de procesos, permite modelar, describir y documentar un proceso de workflow. Para realizar esta actividad existen lenguajes y métodos de modelado para procesos de workflow los cuales tienen varios enfoques como pueden ser: redes de Petri, máquinas de estado infinito, diagramas de flujo, entre otros más.

La definición de los procesos se realiza modelando los conceptos de la vida real a través de los elementos que proporciona el lenguaje de modelado que sea utilizado.

La salida de este proceso de modelado y de diseño es una “definición de procesos”, la cual puede ser interpretada vía la interfaz 1 en tiempo de ejecución por el motor de workflow a través del servicio de representación. La interfaz 1 permite la comunicación entre éste y el componente de representación de workflow.

Numerosos WfMS proveen de herramientas de análisis para revisar la semántica de la “definición de procesos”, es decir determinar si una definición de procesos es correcta. Para ello utilizan avanzadas técnicas de análisis, como la simulación que permite obtener numerosos datos para la validación de una definición de procesos determinada. Este análisis se recomienda antes de poner en práctica el sistema de workflow.

Servicio de representación.

El servicio de representación de workflow's es el componente central encargado de interpretar la descripción de procesos realizada con la herramienta para definición de procesos, además de controlar las diferentes instancias de procesos, administrar las secuencias de actividades, añadir elementos a la lista de trabajo de usuarios e invocar aplicaciones necesarias.

Todas estas tareas son hechas por uno o más motores de workflow, los cuales manejan la ejecución de las distintas instancias de los procesos. Los motores de workflow son el “corazón” del sistema de workflow, sin ellos el sistema no puede operar.

En caso de estar en un entorno distribuido, pueden existir otros componentes de representación de workflow con sus respectivos motores de workflow. La comunicación con éstos se logra a través de la *interfaz 4*. El componente *Workflow WAPI y formatos para el intercambio*, permite la interacción del servicio de representación de workflow con el resto de los componentes, proporcionando datos entendibles para cada uno de los componentes.

Aplicaciones clientes

Este componente representa los programas de software utilizados por el usuario final del WfMS en las actividades que requieren participación humana. La *interfaz 2* permite definir y manejar las listas de trabajo que se encuentran en los motores de workflow. Una lista de trabajo es una lista asignada que contiene actividades por hacer, trabajos pendientes, recordatorios entre otras, las cuales deben ser ejecutadas por un usuario o grupo de usuarios.

Aplicaciones invocadas

Este componente representa todo el software existente dentro de la organización, el cual es utilizado por el WfMS con el fin de interactuar y realizar ciertas actividades. Estas aplicaciones de software pueden encontrarse en cualquier lugar dentro de la red de trabajo. La *interfaz 3* permite la comunicación entre este componente y el de representación de workflow a nivel de invocación, transformación y representación de datos, de manera que éstos sean entendibles para el motor de workflow.

Las aplicaciones invocadas se pueden clasificar en: “*interactivas*” y “*aplicaciones completamente automáticas*”.

Las “*aplicaciones interactivas*” son iniciadas como resultado de la selección de una actividad de las listas de trabajo, puede ser una herramienta como un procesador de texto, una hoja de cálculo o un programa para procesos de negocio, como por ejemplo un formulario electrónico que necesita ser completado.

Las “*aplicaciones completamente automáticas*” se caracterizan por no requerir la interacción con el usuario. Pueden ser parte de una tarea sin la intervención de un usuario, como por ejemplo un programa que realiza un cálculo complicado como, establecer la cantidad de un pago de una instalación.

Herramienta para administración y monitoreo.

El componente “Herramienta para administración y monitoreo” y la *interfaz 5* tienen como propósito permitir una vista completa del estado del workflow. Además, con éste es posible realizar auditorías sobre los datos del sistema. Esta herramienta es utilizada por el administrador del WfMS y por los altos mandos de la organización los cuales tienen la necesidad de tener información exacta del estado de sus procesos de negocio.

Este componente permite realizar otras tareas como:

- Agregar o eliminar personal.
- Gestionar la información de cada empleado, como el nombre, dirección, teléfono, rol, nivel de autorización, disponibilidad, unidad organizacional.
- Implementación de nuevas definiciones de workflow's.
- Reconfiguración del sistema de workflow.
- La inspección del estado logístico de un caso.
- Manipulación de un caso si existen problemas y circunstancias excepcionales.

Otra característica de este componente es incorporar herramientas de almacenamiento y reporte, ya que muchos aspectos pueden ser almacenados durante el funcionamiento del workflow y los mismos son útiles para su administración.

La información acerca de los casos completados del workflow es crucial, así como las advertencias puntuales sobre bloqueos, congestiones y sobrecargas pueden llevar a que el caso sea revisado. Los datos son proporcionados por “*servicio de representación*” y son administrados por las “*herramientas de almacenamiento y reporte*”. Algunos WfMS utilizan reportes predefinidos que son producidos cada intervalo regular, otros tienen un generador de reportes integrado permitiendo a los usuarios definir reportes a partir de la información que es almacenada y todavía otros no proporcionan facilidades de reporte.

1.7 Patrones de workflow

Los patrones son el artefacto ideal para ser usados en la evaluación de motores de workflow, ya que son independientes de las tecnologías y permiten abstraerse de los requerimientos específicos del dominio al que se están enfocando. El estudio de Wil van der Aalst (18) tuvo como resultado la identificación de más de veinte patrones que describen el comportamiento de los procesos de negocios (19).

Constructores básicos de enrutamiento

La WFMC define los constructores básicos para construir las posibles rutas que toma la información dentro de un proceso de workflow (17). Con estos constructores es posible representar la perspectiva de flujo de control, los mismos se muestran en la figura 1.4.

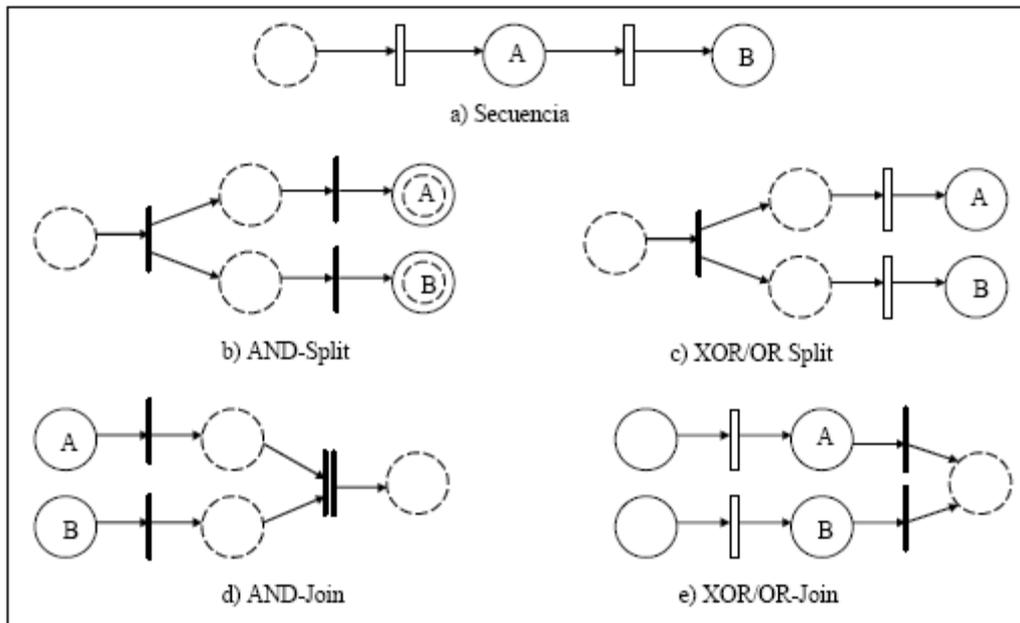


Figura 1.4 Constructores Básicos

1.7.1 Patrones de flujo de control básicos.

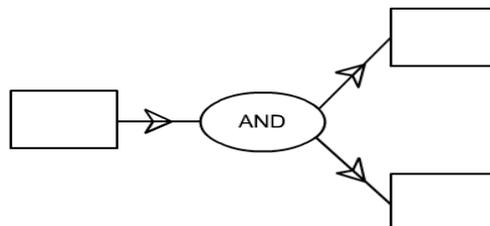
- Secuencia
- Paralelización
- Sincronización
- Elección exclusiva.
- Mezcla simple.

Secuencia.

Una actividad en un proceso se activa después de que su predecesora finalice.

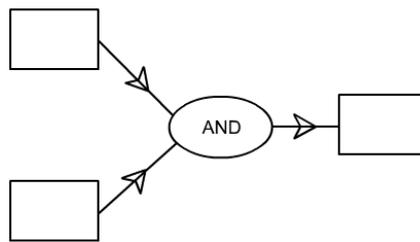
Paralelización.

Se pueden ejecutar 2 o más actividades concurrentemente.



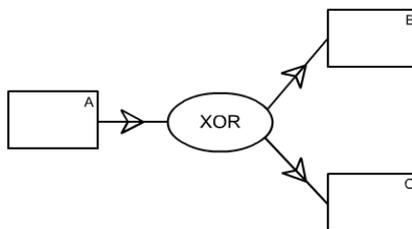
Sincronización.

Existen ramas concurrentes que deben convergir.



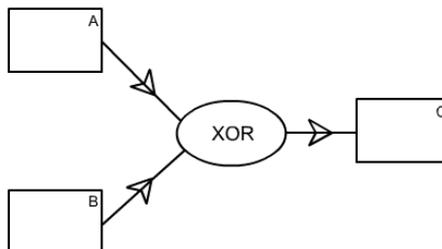
Elección exclusiva.

Una actividad puede pasar el control a una de las bifurcaciones de acuerdo a una expresión lógica.



Mezcla simple.

Una actividad se puede iniciar si alguna de las tareas predecesoras finaliza.

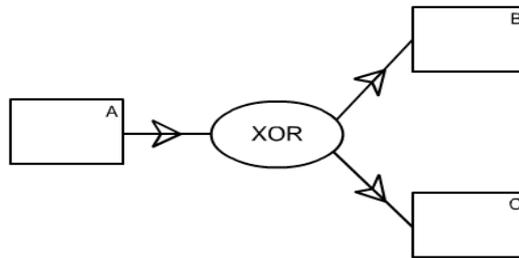


1.7.2 Bifurcación avanzada y patrones de sincronización.

- Elección múltiple
- Mezcla con sincronización
- Mezcla múltiple
- Discriminador

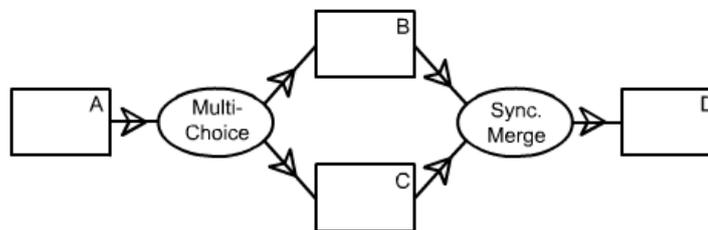
Elección múltiple.

Una actividad puede pasar el control a una o más ramas si cumplen una condición lógica.



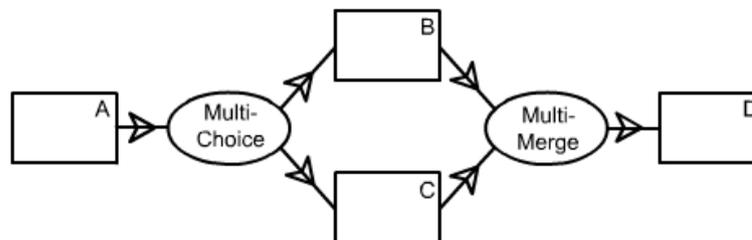
Mezcla con sincronización.

Si existen muchas ramas en ejecución las sincroniza como una mezcla simple y si solo es una, puede iniciar la tarea posterior.



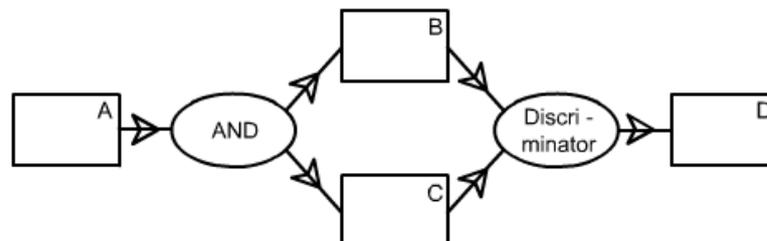
Mezcla múltiple.

Por cada una de las tareas predecesoras que finalicen, se debe ejecutar la actividad descendiente.



Discriminador.

La actividad descendiente se ejecuta una sola vez al terminar la primera actividad antecesora.

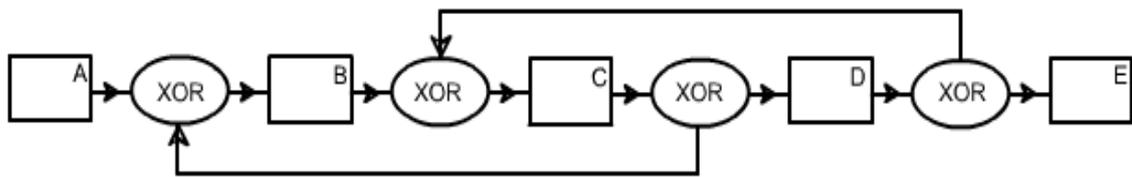


1.7.3 Patrones estructurales.

- Ciclos arbitrarios
- Terminación implícita

Ciclos arbitrarios.

Es la habilidad para representar ciclos en un modelo de proceso que posea más de una entrada o salida.



Terminación implícita.

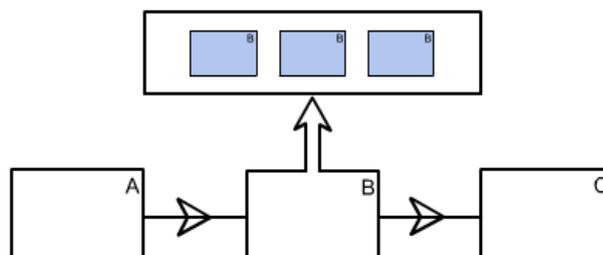
Un proceso dado debería terminarse cuando no hay elementos de trabajo suficientes para ser realizado.

1.7.4 Patrones de múltiples instancias.

- Múltiples instancias sin sincronización
- Múltiples instancias con conocimiento a priori en tiempo de diseño
- Múltiples instancias con un conocimiento a priori en tiempo de ejecución
- Múltiples instancias sin conocimiento a priori en tiempo de ejecución

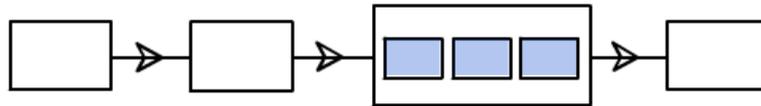
Múltiples instancias sin sincronización.

Creación de varias instancias de una actividad sin sincronizarlas al finalizar.



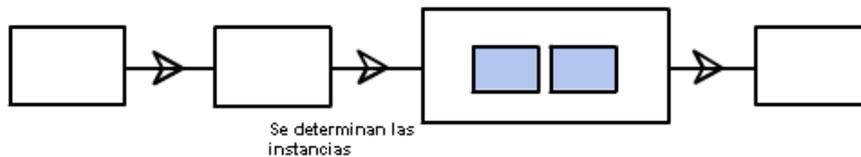
Múltiples instancias con conocimiento a priori en tiempo de diseño.

Crea varias instancias de una actividad, el número de instancias es conocido en tiempo de diseño. Es necesario sincronizar las instancias de la actividad antes de continuar con las actividades subsecuentes.



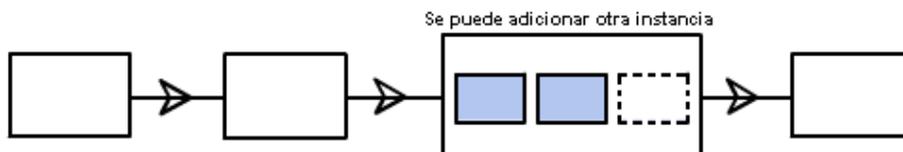
Múltiples instancias con un conocimiento a priori en tiempo de ejecución.

Crea varias instancias de una actividad, el número de instancias es determinado en cualquier punto durante la ejecución. Es necesario sincronizar las instancias de la actividad antes de continuar con las actividades subsecuentes.



Múltiples instancias sin conocimiento a priori en tiempo de ejecución.

Crea varias instancias de una actividad, se pueden adicionar instancias durante la ejecución. Es necesario sincronizar las instancias de la actividad antes de continuar con las actividades subsecuentes.

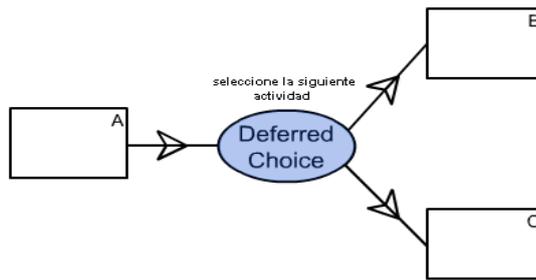


1.7.5 Patrones basados en estados.

- Elección diferida
- Enrutamiento paralelo con íter llegada
- Entregables

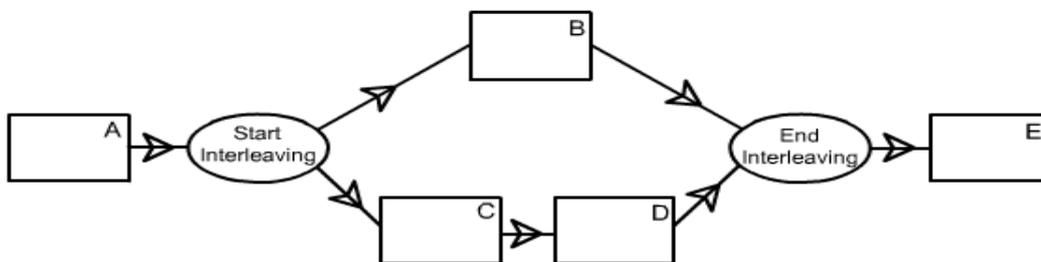
Elección diferida.

Una actividad puede seleccionar cual de las ramas va a ser la siguiente en ejecutarse. Esta decisión se puede demorar el tiempo que quiera.



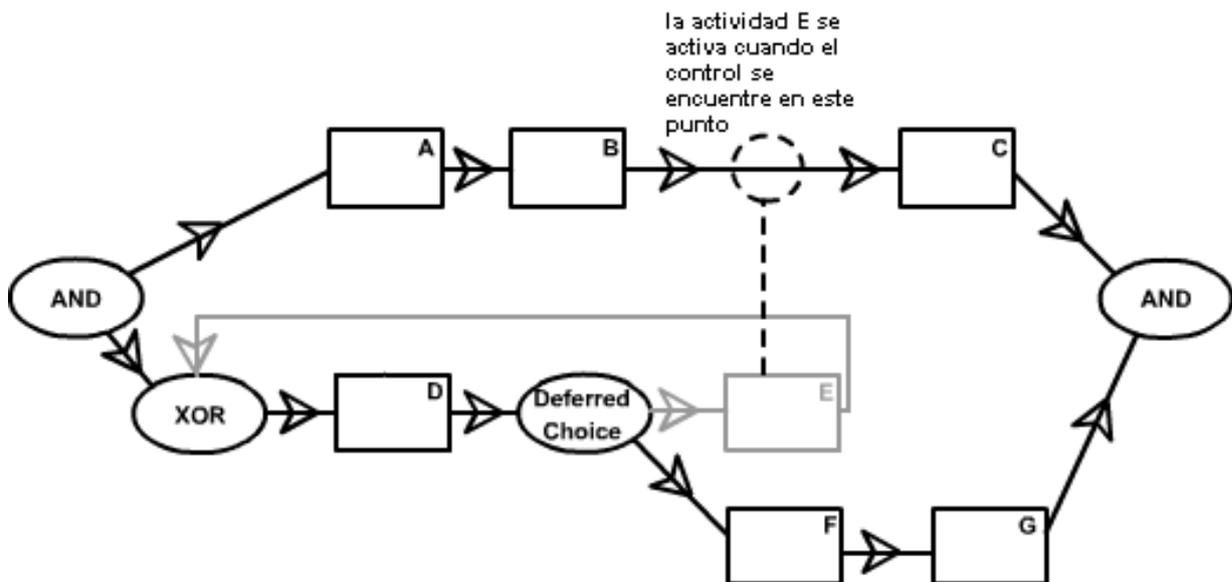
Enrutamiento paralelo con íter llegada.

Ejecuta un número de actividades en algún orden, pero no ejecuta dos actividades simultáneamente.



Entregables.

Antes de que una actividad se inicie se debe alcanzar el entregable.



1.7.6 Patrones de cancelación.

- Cancelación de actividades ó tareas
- Cancelación de casos

Cancelación de actividades ó tareas

Una tarea activa es cancelada antes de comenzar la ejecución. Si ya inició la tarea, la instancia es detenida y removida.

Cancelación de casos

La instancia del proceso es quitada junto con todas las actividades y se guarda como completada sin éxito.

1.8 Criterio de Comparación

Al analizar las diferentes vías de documentación que existen en la prensa especializada o sitios como javahispano (20), WFMC (2), WARIA (7), se pueden ver que existen muchas herramientas de workflow de licencia libre. Tratar tal cantidad de información puede ser una tarea engorrosa e improductiva, es por eso que se hace un análisis de criterios que sirven para comparar las herramientas de workflow los cuales son abordados por varias organizaciones estandarizadas.

La organización WARIA hace un estudio de los principales criterios de comparación que ayudan en la determinación y selección de herramientas workflow (21).

El análisis está enfocado en 12 criterios (Tabla 1.1) que miden el poder expresivo, la habilidad, complejidad, y costo de propiedad, para el desarrollo, despliegue y mantenimiento de aplicaciones workflow's.

Tarifas de Rendimiento	Depende de la eficiencia del motor y su capacidad de correr en cualquier servidor.
Capacidad de Proceso	La capacidad del ambiente de desarrollo para expresar la verdadera complejidad de los procesos en todos sus detalles más pequeños.

Programa de Actividad	El API ¹² es la piedra angular de la programación de actividad, con el apoyo de la dirección de plazos (de fechas límites), y mecanismos de acontecimientos.
Envío y Representación de Organización	La capacidad de enviar cada actividad individual al participante con el nivel correcto de derechos de acceso y papel apropiado en la organización.
Operación y Estadística	Todos los rasgos que facilitan la operación del servidor, y la colección e interpretación de estadística en los tratamientos de casos, incluyendo herramientas proporcionadas para reducir al mínimo el coste de la administración en la estación de trabajo y comunicaciones.
Enterprise Application Integration (EAI)	Evalúa las herramientas proporcionadas para tomar los beneficios más moderno de la EAI: formación de una cola de espera de mensaje, adaptadores de aplicación, COM/DCOM integración distribuida, CORBA, y apoyo de transacción de dos niveles.
Distribución	La cooperación dentro de la misma empresa de varios motores de flujo de trabajo independientes, manejados por departamentos.
Apoyo De Internet	La interfaz del usuario puede ser manejado desde un simple navegador web. Esto cubre las funciones siguientes: el control con las listas de trabajo, la ejecución de actividad, el estado de los procesos gráficos, y el comienzo de una instancia de un proceso.
Cambios Dinámicos	Es muy importante poder realizar caminos dinámicos en situaciones imprevistas.
Definición de Procesos	La mejor vía de acortar el tiempo en la definición de procesos es ofrecer una herramienta gráfica.

¹² Ver Glosario

<p>Definición de Actividades</p>	<p>La implementación de las actividades (sin tener que programar) puede ser lograda con el apoyo proporcionado por las librerías, y herramientas que generan formas, integrado con las herramientas para la definición de procesos. Esto proporciona la flexibilidad de poner en prácticas las actividades.</p>
<p>Agentes listos-a-empleo</p>	<p>El motor es proporcionado junto con agentes listos-a-empleo. No es necesario tener que programar para que los procesos definidos se ejecuten. Esto es obligatorio para las aplicaciones workflow's e incluyen: un manejador de lista de trabajo (worklist) y un agente de actividades estándar para manejar el diálogo con el usuario.</p>

Tabla 1.1 Criterios de Comparación.

Al analizar estos criterios se observa que las características o atributos que deben ofrecer estas herramientas para ser consideradas una solución en la automatización de procesos son: diseño gráfico de procesos, asignación de roles, definir reglas de flujo sin necesidad de programar código, manejo de excepciones, monitoreo, medición, simulación, ser proactiva, tener conectividad transparente a bases de datos y posibilidad de anexar documentos a los procesos.

A continuación se precisa una breve descripción de estas características:

- Diseño gráfico de procesos: contar con una herramienta gráfica para crear los mapas de procesos que definen el flujo del trabajo y las tareas desde el comienzo hasta el final.
- La habilidad de asignar "roles" o "funciones de trabajo" para que el diseño del flujo de trabajo no deba ser cambiado cada vez que alguien se va de vacaciones o es promovido a otro cargo, es de mucha importancia.
- Reglas: La posibilidad de crear la lógica de negocio en la definición de flujo sin necesidad de escribir código.
- Manejo de excepciones: Poder manejar las siempre presentes "excepciones a la regla" es otro atributo clave. Por ejemplo, la característica de reasignar una tarea de un usuario a otro si el usuario está ausente o por causa de un daño en su computador.

- **Monitoreo:** Cada usuario debe idealmente tener la posibilidad de ver los incidentes de workflow en los cuales han participado. Es especialmente importante para los supervisores poderlo hacer con todos sus subordinados.
- **Medición:** Crear estadísticas y métricas para que la gerencia pueda medir el costo y el tiempo de cada proceso es importante para poder detectar cuellos de botella y ayudar en los procesos de reingeniería.
- **Simulación:** Probar los procesos en una sola máquina antes de ponerlos en producción es una característica importante. Es poco práctico probar cada proceso corriendo de estación en estación para verificar que funciona.
- **Proactivo:** La característica de avanzar el proceso de manera activa. Los usuarios deben ser notificados de sus nuevas tareas, retrasos y los supervisores informados cuando un proceso se estanca.
- **Conectividad con bases de datos:** Cada proceso en un workflow utiliza información que suele residir en bases de datos y utilizada para tomar decisiones. Es también importante escribir nuevos datos o modificarlos, por tanto una conectividad transparente es un atributo clave.
- **Anexo de documentos:** Los documentos son parte esencial de los procesos de negocio, una solución de workflow debe por tanto proveer medios efectivos que permitan anexar documentos al workflow, los cuales soportarán el proceso de workflow.

1.9 Herramientas para la gestión de flujo de trabajo.

Como parte de los objetivos de la investigación se presentan varias herramientas para la gestión de flujo de trabajo, todas desarrolladas en java y de licencia libre. Debido a la gran cantidad de proyectos workflow's opensource se considera a OpenWFE, Enhydra Shark y Bonita porque son los que más están presentes en foros de debate, blogs y comunidades; cada una de ellas, se analiza con más detalle a continuación.

1.9.1 OpenWFE

OpenWFE es un WfMS, desarrollado en java, pero no está limitado a ese lenguaje, existen conectores que permiten usarlo desde Python, C# (.NET), PHP y Ruby¹³. Fue creado en el 2001 y registrado en www.sourceforge.net el 29 de Mayo del 2002, está disponible bajo una licencia BSD (22).

OpenWFE utiliza a Droflo, es una herramienta basada en la web y permite el diseño gráfico de procesos. El modelo de los procesos se almacena en formato XPDL, pero su interfaz no es muy buena sobre la web, no es muy llamativa pero es simple.

Recientemente se ha creado el plugin OpenWFE Editor, para el IDE Eclipse 3.0 que permite diseñar gráficamente los procesos y validarlos. Este plugin contiene una paleta de componentes para la creación de las actividades, participantes y reglas; además permite importar y exportar ficheros XML.

OpenWFE requiere el Java Development Kit (en lo adelante JDK) 1.4.0 o una versión superior. Usa Java Runtime Environment (JRE) 1.4.0 o una versión superior, si no se usa aplicaciones web, ya que este tipo de aplicaciones requieren del JDK para poder compilar los JSP¹⁴ (Java Server Pages) en servlets.

La arquitectura general de OpenWFE se muestra en la figura 1.5 y es una adaptación de la arquitectura propuesta por la WFMC.

¹³ Ver Glosario.

¹⁴ Ver Glosario.

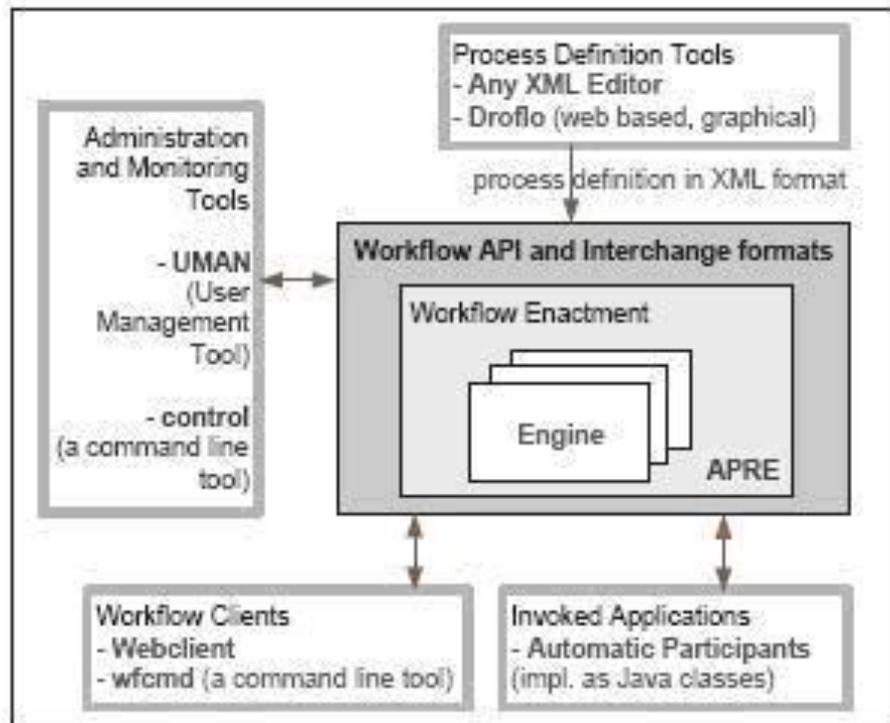


Figura 1.5 Arquitectura de OpenWFE.

El desarrollo de OpenWFE ha migrado ahora de java a Ruby con la nueva aplicación llamada OpenWFEru. Sin embargo, el presente estudio se limita a analizar solo las herramientas que hayan sido desarrolladas sobre la plataforma java (22).

1.8.2 Enhydra Shark

Enhydra Shark (25) es un motor de workflow completamente basado en las especificaciones (2) y (8). Shark utiliza el lenguaje de definición de procesos especificado por WFMC, basado en XML, como formato de definición de workflow, sin extensiones propietarias (11).

Entre sus principales características tenemos:

- Altamente configurable, de forma que el kernel de Shark completo podría ser reemplazada por cualquier otra implementación.
- Puede configurarse para usar estructuras organizativas definidas en un servidor LDAP¹⁵.

¹⁵ Ver Glosario.

- Posee interfaces que permiten trabajar con transacciones cliente.
- Permite utilizar una base de datos para almacenar la información del workflow.

Puede utilizar el Enhydra JaWE(Java Workflow Editor) (25) que no es más que un editor gráfico opensource de procesos de workflow basado totalmente en java y XML. Este editor es totalmente compatible con las especificaciones WfMC encargadas de definir un estándar para frameworks de workflow, trabaja directamente con el formato de ficheros XPDL versión 1.0, conforme a las especificaciones WfMC.

1.9.3 Bonita

Bonita es un sistema de workflow que ofrece muchos rasgos innovadores como definir actividades que pueden comenzar con anticipación, además ofrece una infraestructura de conocimientos que les permite a los usuarios ser notificados de cualquier evento que ocurra durante la ejecución en un proceso dado, o activación automática del código de usuario según un ciclo de vida de actividad definido. Incorpora también los rasgos de los workflow's tradicionales como son la definición dinámica de usuarios y roles, la ejecución secuencial de actividades también está incluida en Bonita (23).

Bonita es un workflow flexible y genérico conformado de acuerdo (17), permite el diseño gráfico para la definición de procesos y administración, además de permitir la interacción entre los usuarios y posee una tercera generación de motores para la ejecución de procesos basado en el modelo de anticipación de la actividad, posibilitando un considerable incremento de acelerar el diseño y el desarrollo de las fases de aplicaciones cooperativas (24).

Algunos de los beneficios de Bonita son:

- Mejor eficiencia en el trabajo colaborativo: un equipo comparte una vista común de las tareas, cada individuo puede tener el conocimiento del tiempo real sobre lo qué está pasando.
- Reducción de costos y riesgos automatizando procesos persona a persona y sistema a sistema.
- Manejo eficiente en ocasiones inesperadas: permite firmemente y dinámicamente modificar la definición de un proceso en ejecución, debido a eventos que no estaban planeados.

- La flexibilidad en la administración de datos y del workflow en ejecución se logra mediante el modelo de anticipación de la actividad. Todos los usuarios tienen la iniciativa de ejecutar sus actividades.
- Puede configurarse para usar estructuras organizativas definidas en un servidor LDAP.

1.9 Análisis de la situación actual acerca del conocimiento de la tecnología workflow para desarrollar eficiencia en los procesos y estandarización del trabajo.

La determinación de la situación en que actualmente se encuentra el proceso de conocimiento que se manifiestan en el uso de la tecnología workflow para desarrollar eficiencia en los procesos y estandarización del trabajo tuvo lugar a través del diagnóstico que se realizó en la UCI, en especial la facultad 3.

El cálculo del tamaño de la muestra significativa es uno de los aspectos a concretar en las fases previas de la investigación y determina el grado de credibilidad que se concede a los resultados obtenidos. Se realizó un muestreo a cada proyecto, y la suma de ellos es la muestra significativa utilizada en la encuesta.

La fórmula extendida que orienta sobre el cálculo del tamaño de la muestra que se utilizó fue la siguiente:

$$n = \frac{N}{1 + \frac{d^2(N-1)}{Z^2 \left(\frac{\alpha}{2}\right) p \cdot q}}$$

N: número total de posibles encuestados.

d: es una constante que depende del nivel de confianza = 0,1

Z: es el error muestral. = 1,96

p: es la proporción de individuos que poseen en la población la característica de estudio. = 0,5

q: es la proporción de individuos que no poseen esa característica, es decir, 1-p. = 0,5.

n: es el tamaño de la muestra (número de encuestas que se realiza).

La siguiente tabla 1.2 ilustra las muestras de cada proyecto al utilizar la fórmula anterior.

	Proyecto	Población	Muestra($\alpha=0,05$)
1	ONE	31	23
2	CCV	40	28
3	CE	9	8
4	MENPET	16	13
5	IP	15	13
6	ERP	52	23
7	SGF	12	10
8	RN	85	45
	TOTAL	260	173

Tabla 1.2 Muestras significativa por proyectos.

La muestra quedó conformada por 173 estudiantes vinculados a proyectos productivos. Estos estudiantes fueron escogidos de manera intencional a partir de su participación en los diferentes proyectos que oferta la facultad.

En el diagnóstico se utilizó como instrumentos un cuestionario aplicado a los estudiantes (**anexo No.1**) para determinar el nivel de conocimientos que tienen los mismos sobre los contenidos referidos a la tecnología workflow.

Se tiene en cuenta el criterio de los estudiantes, por lo que queda claro que hacen referencia a la automatización de los procesos en una organización. En el (**anexo No.2**) se muestra una gráfica con los resultados obtenidos del diagnóstico, esto demuestra que un 17,34% de los encuestados tienen poco conocimientos sobre esta tecnología mientras que un 82,66% no conocen sobre estos sistemas, lo que resulta insuficiente el nivel de argumentación en sus ideas, sin embargo, este resultado no es nada alentador, si se tiene en cuenta que desde el punto de vista práctico no existe una correspondencia entre el supuesto nivel cognitivo y los resultados concretos.

El análisis de los datos arrojados por la aplicación de los métodos y técnicas anteriormente mencionados, se le atribuye las siguientes regularidades.

- En la estrategia de los proyectos no se contempla una línea de trabajo que integre la aplicación de la tecnología workflow y que permita estructurar el proceso de automatización en las organizaciones.

- Se reconocen los aspectos esenciales de la automatización de los procesos en una organización, sus fines y propósitos.
- Es insuficiente el nivel de preparación para establecer una relación estratégica entre la tecnología workflow y la aplicación de sus herramientas, pues poseen escasos recursos teóricos y metodológicos necesarios.

En sentido general, se ha corroborado que no es suficiente la información sobre los motores de flujo de trabajo de licencia libre desarrollados en JEE para el desarrollo eficiente de los procesos y estandarización del trabajo.

1.10 Conclusiones del capítulo

Del análisis que se realiza y la posición teórica asumida en este capítulo, se tiene como conclusión que:

- Para implementar un workflow, se requiere un WfMS que define, crea y administra la ejecución de workflow's mediante el uso de software, ejecutando uno o más motores de workflow, los cuales son capaces de interpretar la definición del proceso, interactuar con los participantes del workflow, y de ser requerido, invocar el uso de herramientas de tecnologías de información y aplicaciones.
- Como resultado de la caracterización realizada se evidencia la necesidad de tener pleno dominio de un sistema de workflow.
- El análisis de las fuentes teóricas y la caracterización realizada permitieron determinar varios criterios de comparación para la posterior elección de un sistema de workflow teniendo en cuenta el análisis de varias herramientas que gestionan el flujo de trabajo.
- Del análisis del diagnóstico inicial se ha corroborado que no es suficiente la información sobre los motores de flujo de trabajo de licencia libre desarrollados en JEE para el desarrollo eficiente de los procesos y estandarización del trabajo.

CAPÍTULO 2: SISTEMAS WORKFLOW'S (PROPUESTA)

2.1 Introducción

Este capítulo se dirige al análisis de la concepción teórica acerca de las herramientas para la gestión de flujo de trabajo. Se expone además, la propuesta del procedimiento que estratégicamente contribuye a la materialización de la concepción teórica desde la profundización de la herramienta Bonita Workflow.

2.2 Fundamentos del procedimiento para el uso de las herramientas que implementan tecnología workflow.

El estudio que se realiza, permite ofrecer un orden lógico en la concepción de las herramientas para la gestión de flujo de trabajo. Para su utilización se determina un procedimiento, el cual deberá estimular su aplicación en diferentes proyectos de la facultad.

Con este procedimiento se concreta la concepción teórica asumida por los autores de la presente investigación.

Como procedimiento se entiende: Acción de proceder. Método de ejecutar algunas cosas (26). Desde el punto de vista teórico se comparten los criterios de que los procedimientos constituyen las vías y los modos para lograr un objetivo.

Álvarez hace alusión a que los procedimientos: “... *son los elementos en que se descompone la técnica, que interrelacionados permite alcanzar los objetivos [...] están condicionados por el medio que se utiliza*” (27).

Lo anterior presupone que un procedimiento es el conjunto de acciones, dados por etapas para lograr la realización de operaciones, donde se realizan tareas concretas, con un fin determinado que necesariamente deberá alcanzar un objetivo específico.

El procedimiento propuesto, tiene como objetivo: favorecer el uso de la tecnología workflow en los proyectos que propone la facultad.

Se determinaron como exigencias, a tener en cuenta para la aplicación del procedimiento, las siguientes:

- Realizar una descripción del objeto de estudio, se analiza las especificaciones de los sistemas workflow's.

- Tener en cuenta las características de la tecnología workflow, atendiendo a intereses y necesidades de obtener eficiencia en los procesos.

Estructuralmente, el procedimiento se divide en cuatro momentos:

Primero: Análisis de los sistemas workflow's: OpenWFE, Bonita y Enhydra Sark.

Objetivo: Evaluar las potencialidades de cada una de las características que deben estar presentes en los sistemas de workflow's.

Acciones a desarrollar para lograr este momento:

- Estudio de las herramientas workflow: OpenWFE, Bonita y Enhydra Sark.
- Análisis de las herramientas utilizando los criterios de comparación establecidos.
- Estudio del soporte que brinda cada una de estas herramientas a los patrones de workflow.

Segundo: Profundización en el estudio de la herramienta Bonita Workflow.

Objetivo: Análisis y reflexión acerca de las características específicas de la herramienta Bonita.

Acciones a desarrollar para lograr este momento:

- Caracterización de la arquitectura.
- Determinación de los requerimientos para su instalación.
- Concreción de los patrones soportados.
- Análisis del soporte y licencia.
- Valoración de la comunidad y las características o cualidades que tienen sus desarrolladores.
- Análisis de la madurez del proyecto.

Tercero: Puesta en marcha de Bonita Workflow.

Objetivo: Ofrecer un conocimiento básico para la utilización de Bonita Workflow orientado al usuario.

Acciones a desarrollar para lograr este momento:

- Determinación de los softwares previos a la instalación de Bonita.
- Análisis del conocimiento básico para la utilización de Bonita.

Cuarto: Evaluación.

Objetivo: Valoración de la efectividad del procedimiento propuesto.

Acciones a desarrollar para lograr este momento:

- Determinación del grado de satisfacción y el interés de la propuesta.
- Valoración del procedimiento propuesto por profesionales con experiencias.

2.3 Análisis de los sistemas workflow's: OpenWFE, Bonita y Enhydra Sark.

A continuación se realiza un análisis de las herramientas OpenWFE, Enhydra Sark y Bonita al tener en cuenta los principales criterios de comparación y las características mencionadas en el capítulo uno, así como los resultados obtenidos en el informe final " *Mejoramiento de la gestión y el uso de las TIC's de las MIPYMES y gobiernos locales a través del software libre* " realizado en el departamento de ingeniería industrial de la facultad de ciencias físicas y matemáticas en la Universidad de Chile(2007). El estudio permite identificar una solución óptima para su utilización en los proyectos. Se confeccionó la siguiente Tabla 2.1, donde se evalúa las potencialidades de cada una de las características que deben estar presentes en los sistemas de workflow's, para la confección de la tabla se establecen 4 escalas: Mal (M), Regular (R), Bien (B) y Excelente (E).

Características	OpenWFE	Enhydra Shark	Bonita
Versión	1.7.1	2.0	3.1
Licencia	BSD	LPGL	LPGL
Soporte	R	B	B
Funcionalidad	R	B	B
Comunidad	R	B	E
Madurez	B	R	E
Diseño gráfico de procesos	R	B	E
Definición de roles y funciones de trabajo	B	B	B
Proactivo	R	B	E
Anexo de documentos	B	B	B
Documentación	M	B	E

Tabla 2.1 Evaluación de las características de las herramientas OpenWFE 1.7.1, Enhydra Shark 2.0 y Bonita 3.1.

Para apoyar la investigación se realiza un estudio del soporte que brinda cada una de éstas herramientas a los patrones de workflow, lo que se muestra en la Tabla 2.2 a través de tres escalas, el carácter “+” indica que la herramienta brinda soporte completo a este patrón, por su parte el “+/-” muestra que la herramienta brinda soporte a este patrón pero de forma limitada y el “-” indica que la herramienta no brinda soporte a este patrón (28) (29).

Control básico	OpenWFE	Enhydra Shark	Bonita
Secuencia	+	+	+
Paralelización	+	+	+
Sincronización	+	+	+
Elección exclusiva	+	+	+

Mezcla simple	+	+	+
Patrones de sincronización			
Elección Múltiple	+/-	+	+
Mezcla con Sincronización	-	-	+/-
Mezcla múltiple	-	-	+
Discriminador	+	-	+
Patrones estructurales			
Ciclos arbitrarios	+	+	+
Terminación implícita	+	+	+
Patrones de múltiples instancias			
Múltiples instancias sin sincronización.	+	+	+
Mult. instancias con conocimiento a priori en tiempo de diseño.	+	-	+
Mult. instancias con un conocimiento a priori en tiempo de ejecución.	+	-	+
Mult. instancias sin conocimiento a priori en tiempo de ejecución.	-	-	+
Patrones basados en estados			
Elección diferida	-	-	+
Enrutamiento paralelo	+/-	-	+
Entregables	-	-	+
Patrones de cancelación			
Cancelación de actividades	-	-	+
Cancelación de casos	+/-	+	+

Tabla 2.2 Soporte de patrones en OpenWFE 1.7.1, Enhydra Shark 2.0 y Bonita 3.1.

Del análisis de las tablas se puede concluir que el sistema de workflow Bonita se destaca, por ofrecer altos niveles de funcionalidad y brindar soporte a la mayoría de los patrones analizados. Este sistema posee una excelente documentación que lo convierte en una solución factible, de gran utilidad y alcance.

2.4 Profundización en el estudio de la herramienta Bonita Workflow.

Para conocer a profundidad el funcionamiento de la herramienta Bonita se hace un estudio y análisis acerca sus características específicas: arquitectura, requerimiento, patrones soportados, soporte y licencia, comunidad y la madurez que presenta en el mercado.

2.4.1 Arquitectura.

Bonita es completamente desarrollado en java, está diseñada totalmente conforme a la arquitectura de JEE, tomando las ventajas de estructuras de JEE, como son Java Beans¹⁶, mensajería asíncrona y los servicios de soporte JEE Web. Bonita utiliza el robusto servidor de licencia libre Jonas y provee el soporte al estándar XPDL (24).

Bonita está diseñada sobre la plataforma JEE que lo provee de gran rendimiento, robustez y flexibilidad. Los ejemplos de tales ventajas incluyen una alta disponibilidad de configuración, soporte transparente a cualquier tipo clase JDBC, conforme a las bases de datos y un modelo flexible de despliegue, en la figura 2.1 se muestra la arquitectura de Bonita.

¹⁶ Ver Glosario.

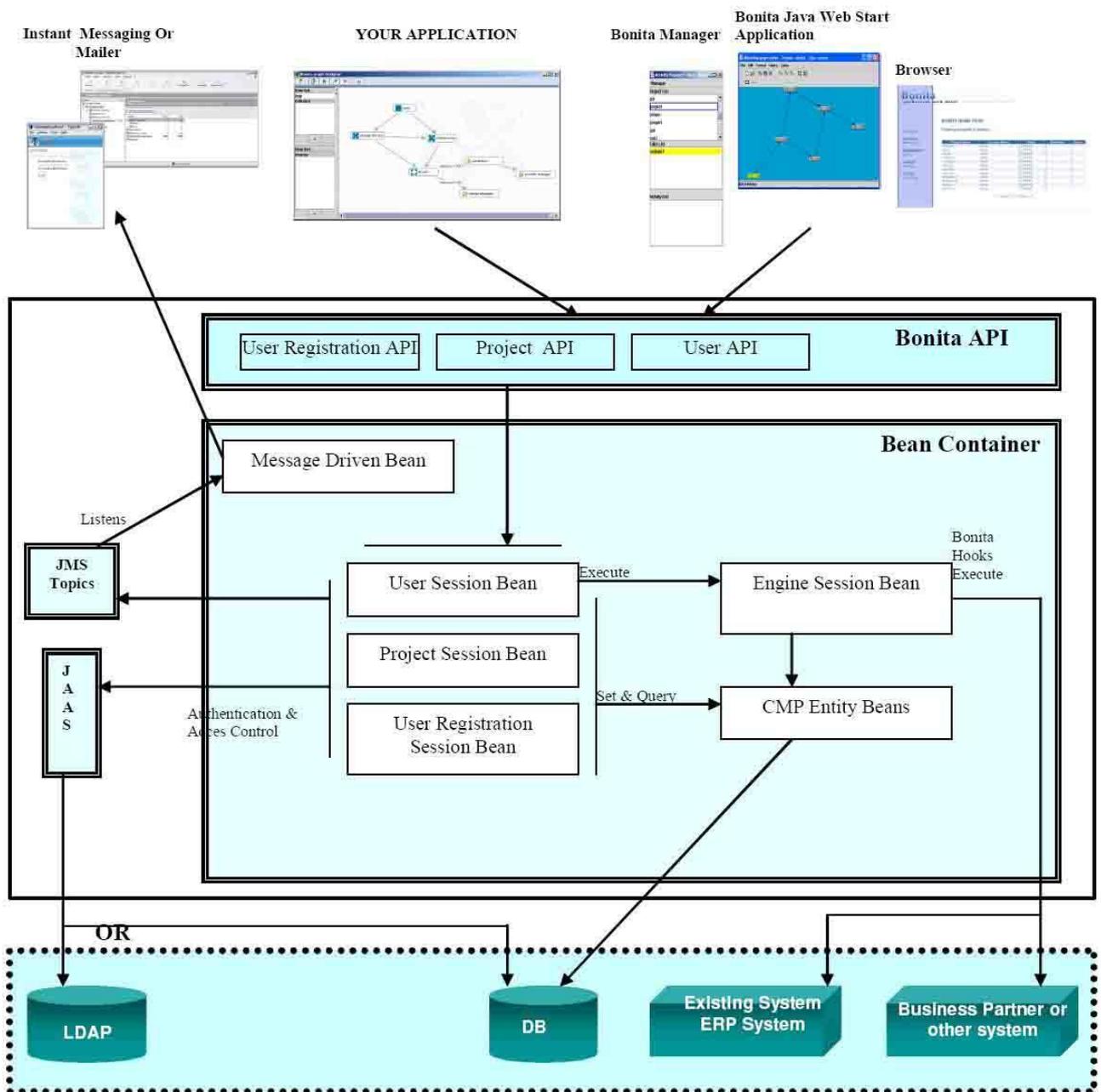


Figura 2.1 Arquitectura de Bonita.

- User RegistrationSessionBean proporciona la interfaz para:
 - Creación y administración de usuarios.
- ProjectSessionBean proporciona la interfaz para:
 - Creación de procesos.
 - Definición de actividades, sub-procesos, transiciones.

- Datos relevantes Workflow.
- UserSessionBean proporciona la interfaz para:
 - Lista de proyectos/instancias (Project list).
 - Lista de actividades preparadas e anticipadas (Todo list).
 - Lista de actividades ejecutando (Activity list).
 - Empezar /terminar/cancelar.
 - Engine bean es una sesión bean especial que controla el proceso de ejecución (no forma parte de la API).

Cada llamada a un método de la API Bonita que modifica el estado del workflow es registrada. Dependiendo de las preferencias de los usuarios estos serán notificados de los eventos o no, a través de mensajería instantánea (jabber) o sistemas tradicionales (correo electrónico).

Nota: El User API y Project API están disponibles como Session Bean o como servicios Web.

2.4.2 Requerimientos.

Bonita tiene como requerimiento un conjunto de softwares que se deben instalar previamente para su uso posterior, como por ejemplo: el Java Development Kit (JDK) 1.4 o superior (se recomienda el JDK 1.5), ANT y el Jonas Application Server.

Los desarrolladores de Bonita recomiendan instalar primero ANT y luego a Jonas Application Server. Para lograr mejores rendimientos se recomienda usar a Jonas Application Server en lugar de JBOSS. También como paso previo a la instalación de Bonita se debe declarar varias variables de ambiente a las cuales se hace referencia más adelante cuando se proponga la puesta en marcha de Bonita.

Para la instalación de Bonita se recomienda un procesador a 2.0 GHz con un mínimo de 512 Mb de RAM¹⁷. Los usuarios del sistema windows pueden evitar el intercambio de archivos y obtener un mejor rendimiento por medio de 1Gb o más de RAM.

¹⁷ Ver Glosario.

2.4.3 Patrones soportados.

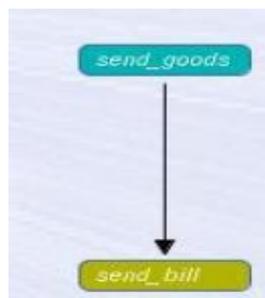
Las potencialidades de Bonita están presente en la cantidad de patrones soportados, a continuación describiremos los mismos y como se representan en Bonita.

El estado de cada actividad va ha estar representado por su color. El color gris indica que la actividad está en su estado inicial, por su parte el color amarillo muestra que la actividad está lista para ser ejecutada, el color rojo revela que la actividad está en ejecución, el verde indica que la actividad ha finalizado y el azul muestra que la actividad está en un estado muerto.

Dentro del grupo patrones de flujo de control básicos tenemos:

Secuencia

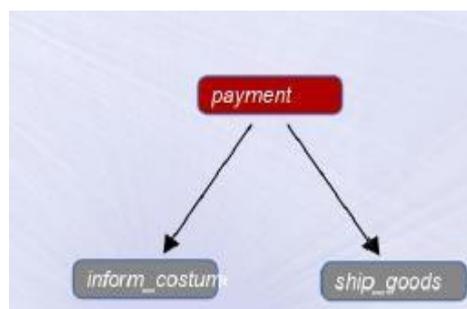
Este patrón es soportado por Bonita. La transición de una actividad a otra es vital en un sistema de workflow.



En el ejemplo se muestra como las actividades *send_goods* y *send_bill* son relacionadas por una flecha incondicional que representa una transición saliente de *send_goods* a *send_bill*. El color verde de la actividad *send_goods* representa que su estado es terminado, lo cual hace que la actividad *send_bill* cambie de su estado listo a ejecución ya que su actividad predecesora ha terminado.

Paralelización.

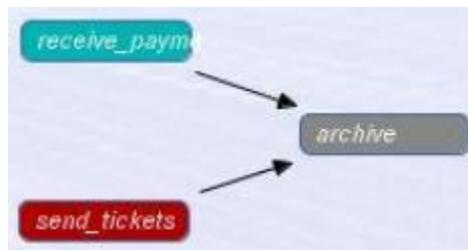
La posibilidad de ejecutar dos o más actividades concurrentemente es común en la definición de muchos flujos de trabajo. La paralelización es otro patrón soportado por Bonita.



En el ejemplo se muestra como las actividades *inform_costum* y *ship_goods* pueden ser ejecutadas de forma paralela, ambas están en estado inicial, representado por su color marrón.

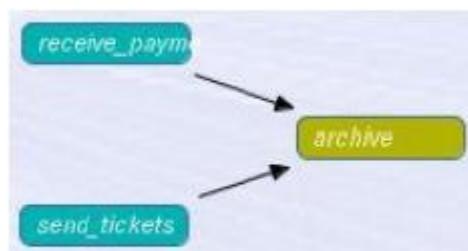
Sincronización.

La sincronización de las actividades es otro patrón soportado por Bonita, posibilitando representar actividades paralelas que puedan convergir en una sola actividad, como se muestra a continuación.



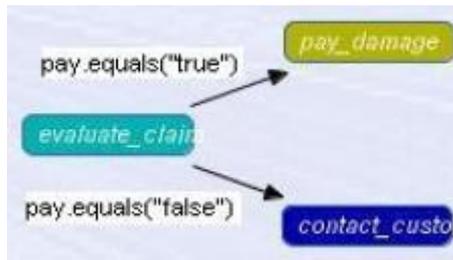
En este ejemplo se muestra como la actividad *receive_payment* tiene su estado terminado, mientras que la actividad *send_tickets* tiene su estado en ejecución, por lo tanto la actividad *archive* cuyo estado es inicial, no puede comenzar hasta que ambas actividades *receive_payment* y *send_tickets* hayan concluido.

Cuando ambas actividades hayan terminado la actividad *archive* cambia su estado a listo como se muestra continuación.



Elección exclusiva.

La eventualidad de definir el flujo a seguir en dependencia de ciertos datos es llamado comúnmente como elección exclusiva. Bonita brinda soporte a éste patrón de workflow.

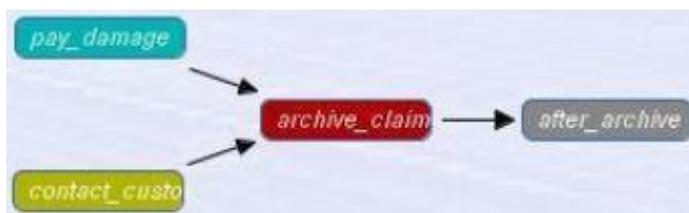


En el ejemplo se muestra como la actividad *evaluate_claim* tiene 2 transiciones salientes, que tienen 2 condiciones exclusivas, conduciendo a la exclusividad de la opción, en este caso se define la propiedad *pay* cuyo valor es verdadero, por lo que la actividad *evaluate_claim* tiene como estado terminado, la actividad *contact_customer* está en estado muerto y la actividad *pay_damage* tiene como estado listo a ejecutarse.

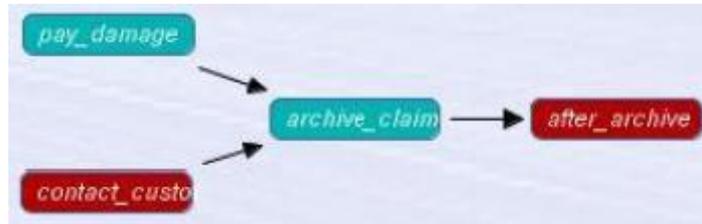
Las condiciones están relacionadas con los datos de control del proceso, los que están asociados a actividades y son llamados propiedades. Bonita define dos alcances para las propiedades: las propiedades del proyecto que son accesibles desde todas las actividades dentro del proyecto y las propiedades de un nodo, donde los datos almacenados en la característica sólo son accesibles desde un nodo concreto.

Mezcla simple.

Muchas veces es necesario representar a dos actividades que convergen en otra actividad pero de una forma no simultánea, esto es conocido como mezcla simple y es un patrón soportado por Bonita. Para seleccionar la ruta entre diferentes opciones Bonita implementa actividades de tipo OR-JOIN. En el ejemplo a continuación se muestra una actividad de este tipo: *archive_claim*, que es adicionada para realizar la mezcla. La actividad *archive_claim* puede comenzar porque la actividad *pay_damage* tiene como estado terminado, es decir ha finalizado.



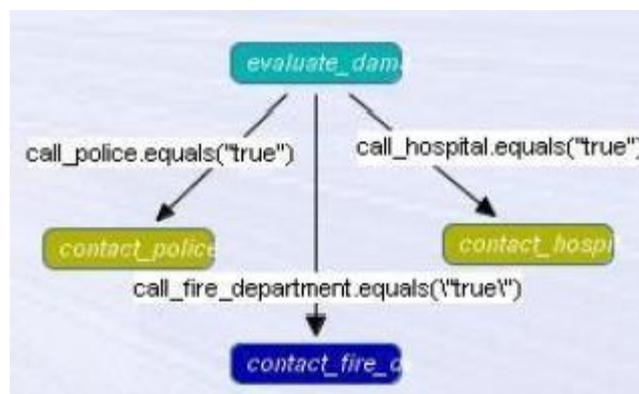
Si la actividad *contact_customer* es ejecutada y ya *archive_claim* ha concluido, como se muestra a continuación, cuando *contact_customer* finalice *archive_claim* no será ejecutada de nuevo.



Entre otros grupos de patrones que existen se encuentra: **Bifurcación Avanzada y patrones de sincronización.**

Elección múltiple.

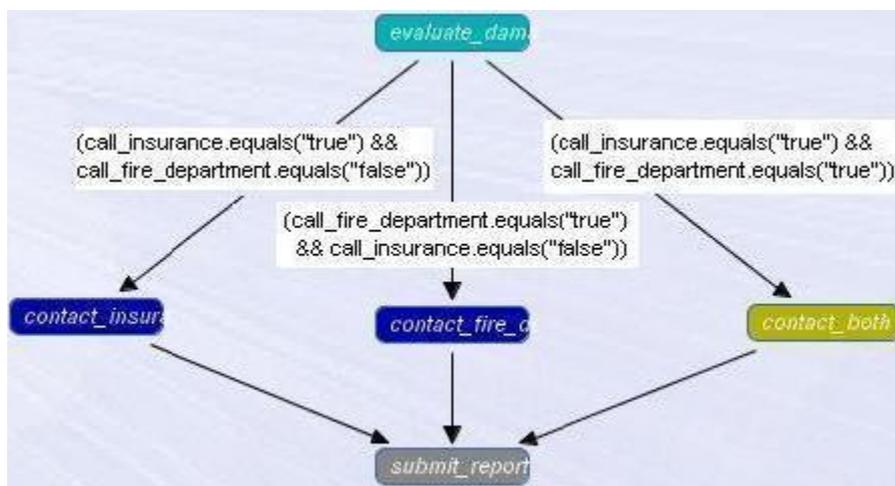
La elección múltiple es otro patrón soportado por Bonita, ya que permite ejecutar una o más actividades en dependencia de los datos de control (propiedades).



En el ejemplo se muestra que la actividad *evaluate_damage* tiene 3 propiedades: *call_police*, *call_hospital*, *call_fire_department*. Las transiciones de *evaluate_damage* a *contact_police*, *contact_hospital*, *contact_fire_department* están expresadas por propiedades. En este caso solamente las propiedades *call_police* y *call_hospital* tienen como valor "true", debido a que la actividad *evaluate_damage* ha finalizado y de acuerdo a las propiedades, las actividades *contact_police* y *contact_hospital* cambian su estado a ejecución, la condición relacionada con la actividad *contact_fire_department* es falsa, por lo que *contact_fire_department* está en estado muerto.

Mezcla con sincronización.

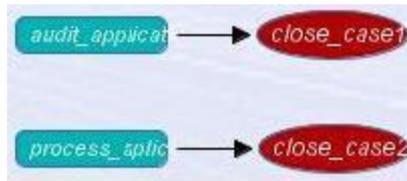
Este patrón es soportado parcialmente por Bonita, ya que el AND-JOIN no es conveniente aquí porque algunas ramas podrían no ser tomadas. La utilización del OR-JOIN no es conveniente tampoco, ya que la primera rama que termina la ejecución es considerada y no hay la sincronización con otras ramas de ejecución. Poner en práctica este patrón solo es posible al encontrarse frente a un caso sencillo, como se muestra en el ejemplo siguiente donde inicialmente existían dos caminos a tomar.



La idea es agregar una actividad extra (*contact_both*) que ejecute las actividades *contact_insurance* y *contact_fire_department*. Es importante destacar que la actividad *evaluate_damage* es implementada según el patrón elección exclusiva seleccionando una sola ruta. Este tipo de solución no es recomendada cuando tenemos muchos caminos posibles.

Mezcla múltiple.

Bonita brinda soporte al patrón mezcla múltiple. El objetivo de compartir la definición de la actividad que tiene este patrón puede ser realizado gracias al concepto de sub-procesos de Bonita.



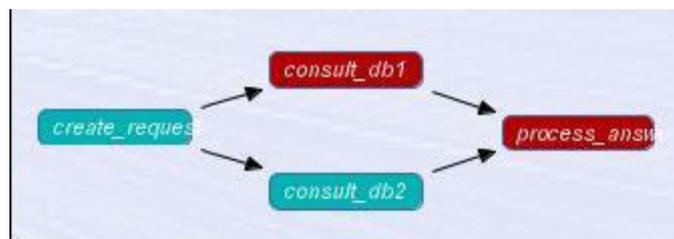
En este ejemplo se muestran las actividades *close_case1* y *close_case 2*, las cuales pertenecen al sub-proceso *close_case* (el cual no se muestra). La definición del sub-proceso *close_case* puede ser tan compleja como sea necesario. Asumiendo que las transiciones no sostienen cualquier transición en este ejemplo, una instancia de *close_case* será creada después de cada terminación de las actividades *audit_application* y *process_application*.

El uso de las propiedades de cada actividad puede ser útil para la configuración de los parámetros de cada sub-proceso, después de la ejecución del sub-proceso se puede añadir otra actividad de tipo AND-JOIN para sincronizar a ambos.

Un comportamiento asíncrono también puede ser alcanzado, mediante el empleo de hooks¹⁸, que brinda Bonita.

Discriminador.

Este es otro patrón soportado por Bonita. En el ejemplo se muestra que la actividad *consult_db2* a terminado, por lo tanto la actividad *process_answer* se ejecuta, independientemente que *consult_db1* este en ejecución y no haya finalizado.



Entre otro grupo de patrones que existen se tienen: **Patrones estructurales.**

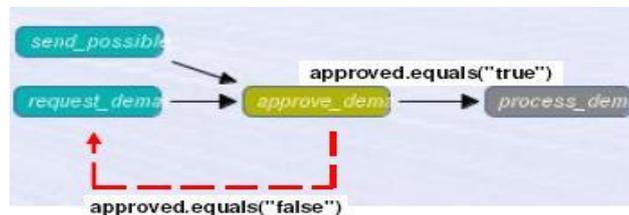
¹⁸ Ver Glosario

Ciclos arbitrarios.

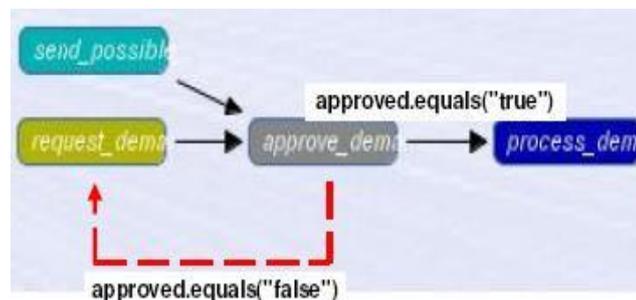
Bonita brinda soporte a este patrón asimilando ciclos estructurados y no estructurados. Además establece ciertas reglas para los procesos que tienen iteraciones, garantizando un correcto comportamiento. Partiendo de que no es posible seguir con la ejecución dentro de una iteración y dejarla al mismo tiempo, se tiene que:

- Solo una iteración es permitida entre dos nodos conectados.
- Es posible tener más de una iteración que comience en el mismo nodo.
- Todas las transiciones que salen de un nodo que comienza una iteración, debe tener una condición. En el caso que exista más de una transición saliendo del nodo, todas deben tener una condición.
- Si tenemos una iteración con un nodo de salida extra, es estrictamente necesario tener condiciones en todas las transiciones que salen de aquel nodo. Estas condiciones tienen que ser mutuamente exclusivas para tomar un camino y seguir iterando o salir de la iteración.

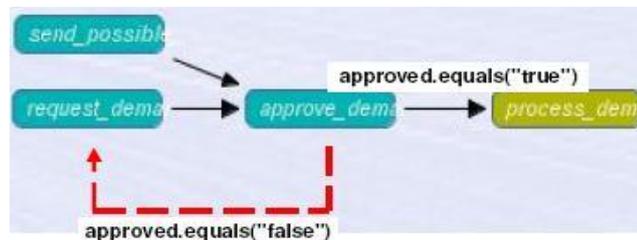
En el siguiente ejemplo se muestra una iteración con un nodo de salida extra.



Como se aprecia la actividad `send_possible_options` es el punto extra en la iteración. La actividad `approve_demand` no puede comenzar hasta que `send_possible_options` y `approve_demand` no terminen.



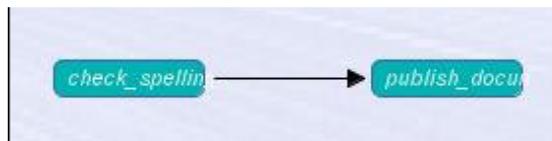
Después de la iteración, la actividad *request_demand* cambia su estado lista y es cancelada *process_demand* porque la condición era falsa (esto hace que se vuelva a iterar).



Al final cuando la propiedad o condición *approved* es verdadera *process_demand* cambia su estado a listo.

Terminación implícita.

La terminación implícita es otro patrón soportado por Bonita.

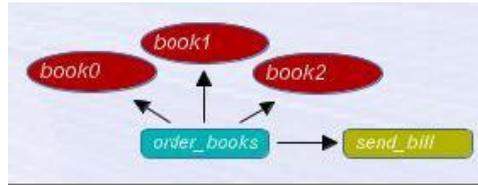


En este ejemplo ambas actividades han finalizado, así que el proceso ha culminado porque no hay nada más que hacer. Bonita permite hacer modificaciones en tiempo real de instancias de procesos, en este sentido siempre hay algo que debe ser realizado. Entonces una llamada explícita a la API debe ser hecha para terminar formalmente el proceso, cuando todas sus actividades son completadas.

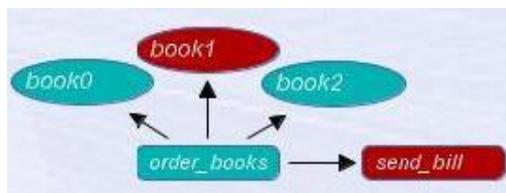
Entre otro grupo de patrones que se tiene, tenemos: **Patrones de múltiples instancias.**

Múltiples instancias sin sincronización.

Bonita brinda soporte a este patrón mediante el uso de *hooks*, otro componente de Bonita implementado en java que ha sido definido en el ciclo de vida de la actividad y que durante la ejecución de esta se activa automáticamente. El uso de Bonita Project API, hace que solo pueda ser lanzado asincrónicamente una instancia de un proceso en el contexto de una actividad.



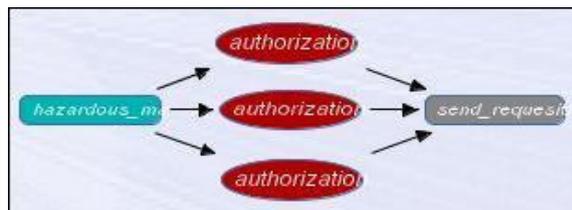
En este ejemplo se muestra como la actividad *order_books* tiene 3 instancias, durante la ejecución de este *hook*. Los sub-procesos *book* son independientes entre ellas mismas y de la actividad *send_bill*.



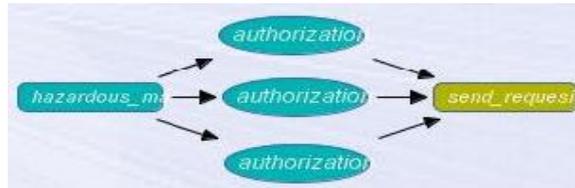
Luego se observa que *book0* y *book2* ya han sido ejecutadas. La actividad *send_bill* y *book1* están siendo ejecutadas al mismo tiempo.

Múltiples instancias con conocimiento a priori en tiempo de diseño.

Este es otro patrón soportado por Bonita. En el ejemplo se muestra que la actividad *hazardous_materials_requisition* se instanciará con varios sub-procesos (*autorización*). Después la actividad *send_requisition* se sincronizará con estos sub-procesos.

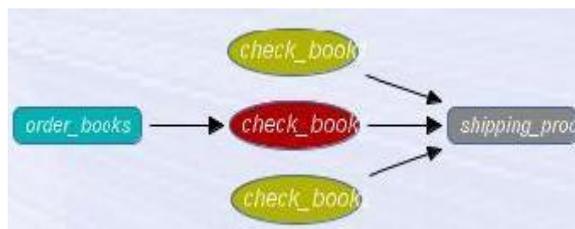


Solamente cuando todas las autorizaciones son aceptadas la actividad *send_requisition* puede ser ejecutada.

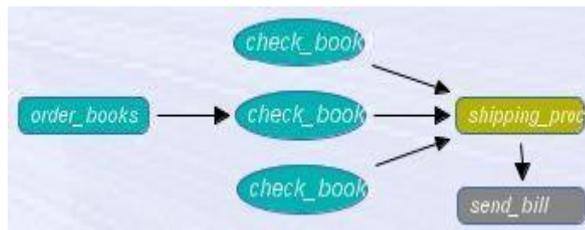


Múltiples instancias con un conocimiento a priori en tiempo de ejecución.

Bonita brinda soporte a este patrón gracias a los *hooks* y a la habilidad de modificar dinámicamente (en tiempo de ejecución) la definición de una instancia de un proceso. Para un correcto diseño una instancia del sub-proceso es definida en el modelo y las otras réplicas durante la ejecución. En el ejemplo se muestra que la actividad *check_book* se define en el modelo y es ejecutada antes que el *hook* es inicializado, creando tantos sub-procesos como sea necesario.

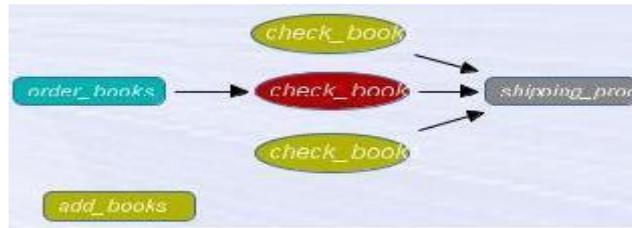


Cuando todos los sub-procesos finalizan, la actividad *shipping_process* se sincroniza con ellos y continúa la ejecución del flujo.



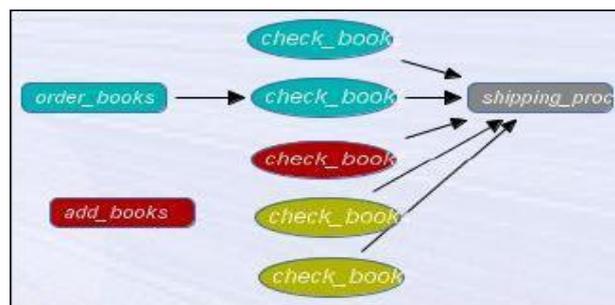
Múltiples instancias sin conocimiento a priori en tiempo de ejecución.

Este es otro patrón soportado por Bonita. En el ejemplo que se muestra a continuación agregaremos una actividad extra (*add_books*) que adicionará más sub-procesos cuando sea ejecutada.



En este caso la actividad extra es *add_books* que crea 2 nuevos sub-procesos cuando es ejecutada. La creación de nuevos sub-procesos puede ser realizada en cualquier momento después de la terminación de la actividad *order_books* y antes el comienzo de *shipping_process*.

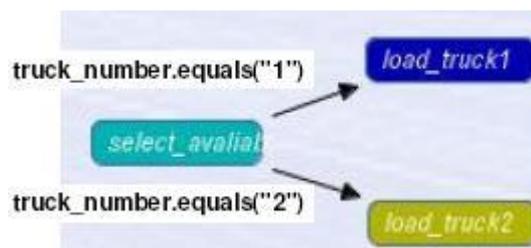
Aquí se observa como la actividad *add_books* está en ejecución y dos nuevos sub-procesos han sido agregados, la actividad *order_books* ha finalizado y *shipping_process* está en estado inicial.



Otro grupo de patrones que está definido es: **Patrones basados en estados.**

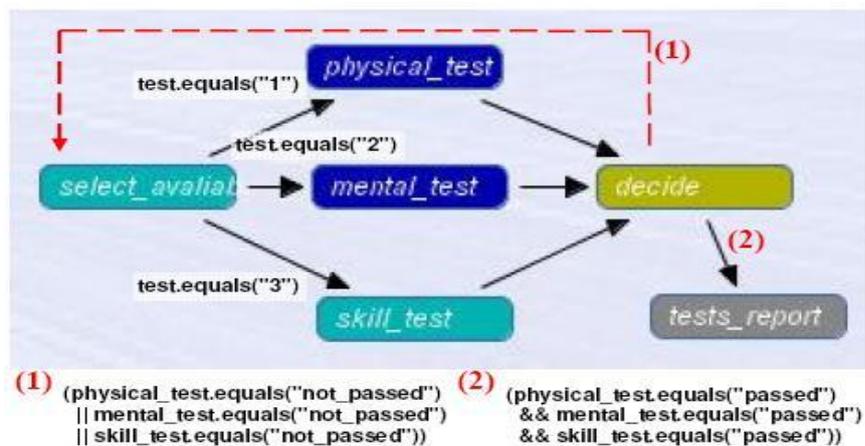
Elección diferida.

Bonita brinda soporte al patrón elección diferida. En el ejemplo a continuación se muestra la actividad *select_avaliable_truck* que decide cual camión esta disponible, esta decisión es tomada a través de un *hook* durante la ejecución de la actividad. Una vez que la decisión ha sido tomada, la propiedad *truck_number* es cambiada para tomar una transición específica. Ya que cada transición tiene relacionada una propiedad que permite tomar un solo camino.

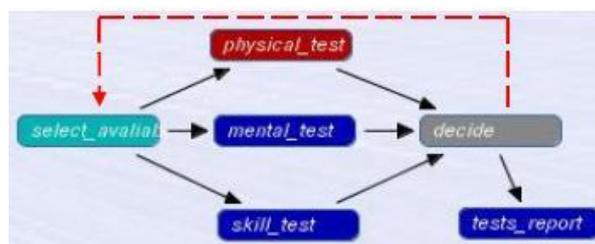


Enrutamiento paralelo.

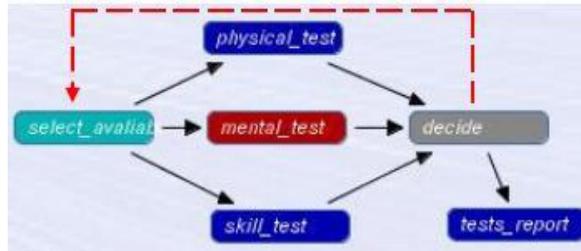
El enrutamiento paralelo es otro patrón soportado por Bonita, para ello se apoya en el patrón elección diferida. En el siguiente ejemplo se muestra que la actividad *select_avaliab_test* implementa el patrón de elección diferida para cada una de las pruebas. En este caso la actividad *skill_test* ha sido ejecutada y la actividad *decide* está lista para iniciarse; esta actividad iterará hasta que las 3 actividades predecesoras hayan finalizado. También se hace uso de propiedades para el control de las actividades que deben realizarse.



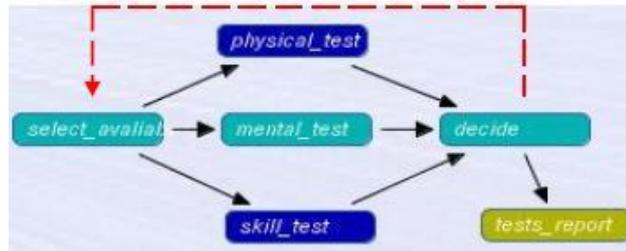
Después de la primera iteración una actividad diferente es seleccionada.



Luego de la segunda iteración la última actividad es seleccionada.



Cuando todas las actividades son ejecutadas la iteración no se realiza y la actividad *test_report* cambia su estado a lista para ejecutarse.

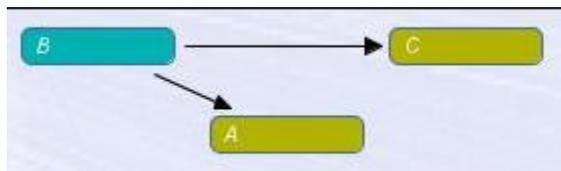


Entregables.

Bonita brinda soporte a este patrón, para ello se apoya en dos soluciones diferentes:

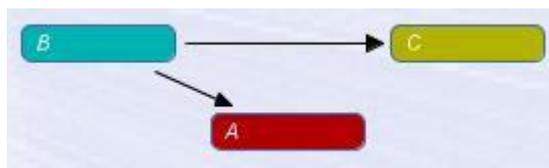
- Usando un *hook* (en la actividad C) que antes que comience C, cancele la actividad A, si esta última puede ejecutarse.
- Uso de fechas límites para cambiar la disponibilidad de A.

En el ejemplo a continuación se muestra que la actividad A sólo se activa después de la ejecución de B, esto se logra sólo con el establecimiento de una transición de B a A.

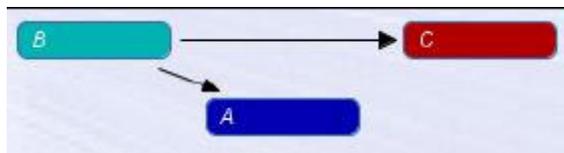


Se analiza las dos posibles situaciones que pueden suceder:

1. La ejecución de A antes que C.



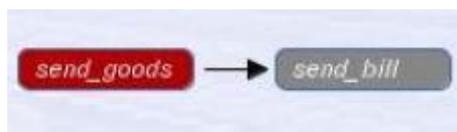
2. La actividad *C* se ejecuta antes que la *A*. Entonces el *hook* es activado y si el estado de *A* es listo, se cancelará *A*.



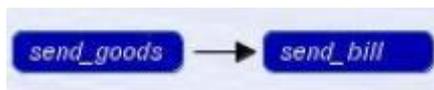
Entre otros grupos de patrones que existen se tiene: **Patrones de cancelación.**

Cancelación de actividades ó tareas.

Bonita brinda soporte al patrón de cancelación de actividades o tareas, esta cancelación se realiza a través del método *cancelActivity* de la *UserSessionBean API*. La cancelación es propagada a las actividades que dependen de la actividad cancelada y no podrán ser ejecutadas. Esto se muestra en el siguiente ejemplo. Mientras que la actividad *send_goods* es ejecutada se hace un llamado al método *cancelActivity*.



Luego la actividad *send_goods* es cancelada y *send_bill* también porque depende de la anterior.



Cancelación de casos.

La cancelación de casos es otro patrón soportado por Bonita. La misma se realiza mediante el método *removeProject* de la *UserSessionBean API*. Este método puede ser usado para eliminar cada instancia del proyecto o el modelo del proyecto. El modelo del proyecto no puede ser eliminado si el modelo tiene instancias en ejecución.

2.4.4 Soporte y licencia

Bonita cuenta con un sitio que tiene una buena estructura organizativa. Presenta un foro en el cual los desarrolladores del proyecto publican informaciones, dan respuestas a dudas presentadas por personas que se inician en el tema, es decir, es otra manera de brindar

soporte a todos los usuarios del mundo que estén interesados en el producto. El proyecto Bonita es reconocido en el mercado de las herramientas para la gestión del flujo de trabajo por su calidad, y el soporte que brindan a las organizaciones que implementan estas soluciones. Esta combinación del producto Bonita y el apoyo ofrecido por los expertos de Bull SAS ayudan a resolver problemas durante cualquier etapa del ciclo de vida de la solicitud deseada. Este servicio es una ayuda anual, el precio es en dependencia del número o tamaño de los servidores. El contrato da soporte a solicitudes ilimitadas y con un TAM (Technical Account Manager) interfaz que da soporte a los servicios específicos de los clientes.

Bonita se encuentra bajo la licencia pública general limitada (LGPL) del GNU. Los contratos de licencia de la mayor parte del software, están diseñados para jugar con su libertad de compartir y modificar dicho software. LGPL tiene como objetivo permitir el uso de bibliotecas libres, funciona como GPL cuando se redistribuye la biblioteca como tal y permite la integración de cualquier software, en este caso prácticamente no hay limitaciones.

2.4.5 Comunidad

Es muy importante destacar que para poder tener una buena referencia de cualquier sistema para la gestión de los flujos de trabajo, es necesario conocer las principales características o cualidades que tienen sus desarrolladores. Es una realidad, y está demostrado en la vida cotidiana, que teniendo conocimiento del personal que realiza el desarrollo de determinado producto, se podrá tener una mayor confianza en la calidad del mismo.

Bonita cuenta con una amplia calidad intelectual de desarrolladores (**Anexo 3**), así como la participación de organizaciones que tienen experiencia sobre el tema.

ObjectWeb

ObjectWeb es un consorcio de empresas líderes y organizaciones de investigación de todo el mundo que han unido sus fuerzas para producir la próxima generación de Open Source Middleware. ObjectWeb tiene como objetivos las soluciones alternativas a productos patentados para el e-business, EAI, la conectividad de datos y aplicaciones de mensajería empresarial. Basado en estándares abiertos, ObjectWeb del middleware incluye servidores de aplicaciones, componentes, marcos y herramientas. Fundada en 2002 por Bull, France

Telecom y el INRIA. ObjectWeb es organizada por INRIA, y está patrocinado por Together Teamlösungen GMBH.

INRIA

INRIA, un co-fundador de Objectweb, ha sido el contribuidor clave de Bonita. El objetivo principal es la cooperación a través de la participación en aplicaciones interactivas (co-diseño, ingeniería de software, las ediciones cooperativas, flujo de trabajo) distribuidos en una red de área extendida (normalmente Internet). La ambición de INRIA es ser una potencia mundial, un instituto de investigación en el corazón de la sociedad de la información. La red de INRIA tiene las capacidades y talentos de los ámbitos de la información y ciencias de la computación y la tecnología de todo el sistema de investigación francés. Esta red permite la excelencia científica que debe utilizarse para el progreso tecnológico, creación de empleo, riquezas y nuevos usos en respuesta a las necesidades socioeconómicas. INRIA tiene una organización descentralizada (6 unidades de investigación) y los pequeños equipos autónomos. También participa en el desarrollo de los resultados de las investigaciones y la transferencia de tecnología: 400 contratos de investigación y desarrollo con la industria y 50 empresas de tecnología han nacido del Instituto.

BULL

Bull, uno de los miembros fundadores de ObjectWeb, es contribuidor de Bonita. Es un grupo internacional de TIC¹⁹ que se especializa en el diseño, desarrollo y ejecución de infraestructuras de TIC seguros. Bull se dedica a ayudar a los clientes a transformar sus procesos e infraestructuras con el fin de aprovechar al máximo la tecnología de la información para impulsar su crecimiento en el mercado. Para alcanzar este objetivo, proporciona una oferta completa que incluye no sólo servicios de TIC, desde consultoría a integración de sistemas y subcontratación, sino también el diseño y desarrollo de soluciones de infraestructura, con especial énfasis en servidores, almacenamiento y gestión segura de software. Bull ha adquirido gran reconocimiento en los principales sectores económicos, como el sector público, las finanzas, la industria y las telecomunicaciones.

¹⁹ Ver Glosario.

2.4.6 Madurez

El proyecto Bonita es utilizado por empresas de todos los tamaños y en todas las industrias alrededor del mundo. Estas han integrado con éxito Bonita Workflow para mejorar sus procesos de negocio, ofrecer servicios en línea y aprovechar las soluciones que brinda. Aprender, y conocer las opiniones de estas empresas ayudan a tener una visión de la capacidad y robustez de la herramienta para la gestión de flujo de trabajo (30).

A continuación se reflejan varias opiniones de empresas y organizaciones que han implementado a Bonita, demostrando la madurez del proyecto en cuanto a la calidad en el mercado.



Esta solución integra la Plataforma Exo y el proyecto Bonita, parte del ecosistema de ObjectWeb. Con Exo-Bonita a los usuarios se les permiten interactuar con el motor de flujo de trabajo a través de un JSR 168 portlet interfaz de usuario.



Bonita es el motor de flujo de trabajo de esta aplicación de gestión de la información para los programas europeo Sócrates / Erasmus. El sistema maneja dos tipos de procesos de flujo de trabajo de Sócrates / Erasmus de becas y estudios extranjeros de procesos de convalidación. Se trata de un proyecto de servicio dirigido por Bull España.



Laurence Davi, líder del proyecto en el Ministerio Francés de Finanzas:

“Ya en el verano de 2004, se nos pidió desarrollar una nueva aplicación para gestionar el proceso de firma en el departamento. La elección de utilizar una herramienta de flujo de trabajo era necesario para nosotros, puesto que ya tenía una importante experiencia en el desarrollo de aplicaciones basadas en este tipo de productos, pero en el aspecto comercial”.

“Hemos decidido para esto utilizar proyecto open source, y elegir Bonita Workflow del consorcio ObjectWeb, con la ayuda del servicio de operaciones Bull. El proyecto ha tenido bastante éxito, y tenemos ahora un prototipo completamente funcional.”

“¿Cuál es nuestra opinión sobre Bonita? Bueno, en el punto de vista técnico, se utilizó ampliamente la flexibilidad de Bonita, y la posibilidad de definir sobre la marcha las características de un proceso de instancia. Y que realmente ha impresionado por la reactividad del equipo de desarrollo, que en algunas ocasiones ha resuelto los problemas para nosotros de la noche a la mañana”.



Bonita Workflow está incluido en la suite CollabCAD. CollabCAD es un software 3D CAD / CAM, sistema de colaboración en el diseño y desarrollo de modelos industriales desde el diseño asistido por computador (CAD) grupo del Centro Nacional de Informática (NIC), los profesionales en soporte de TI de la India.



El proyecto de Valorización del Conocimiento Matriz ha sido desarrollado por INRIA y KAPPA con el fin de simplificar y ayudar a las organizaciones de la Gestión del Conocimiento. Bonita se integró en el portal de usuario para gestionar la matriz de proceso de búsqueda. Como este portal no se desarrolló en java, la integración se realiza a través de los servicios web Bonita API's.



OpenCities es un proyecto para integrar y mejorar varios middleware de código abierto (como Bonita y JASS) para ofrecer una plataforma que se puede utilizar en la administración pública de España. Este es un proyecto conjunto entre Andago y el Laboratorio de Sistemas Distribuidos (LSD) de la Universidad Politécnica de Madrid.



La plataforma LibreSource es un entorno de cooperación dedicada principalmente a proyectos de cooperación asíncrono. Bonita se ha integrado en esta plataforma como el sistema de flujo de trabajo dedicado a los procesos de desarrollo de software.



Philippe Isorce director técnico de MT System: *“Proporcionamos los servicios públicos para mejorar la organización y gestión de los transportes. Estas utilidades son apoyadas por los múltiples modos de Sistema de Información, una plataforma de software que permite definir, gestionar y controlar los datos cambiados. Nosotros utilizamos la API del motor Bonita y en la parte superior Jonas 4.1.2 dentro de los múltiples modos de los Sistemas de Información. Bonita es utilizado para la recolección de viaje calendario y la información sobre el transporte público. Organizamos el proceso de recogida con los diferentes actores y funciones”.*

“La función administrativa puede asignar tareas específicas acerca de un autobús, tren o tranvía, y toda la información que se recopila en formato CVS. A continuación, esta información se carga en el servidor y en una tarea específica de flujo de trabajo, los datos son controlados e insertados en la base de datos, mientras que la última tarea administrativa es validar los datos antes de que se publiquen en los servicios web”.

2.5 Gestión y configuración de Bonita Workflow.

En los epígrafes anteriores se definieron los requerimientos para la instalación y puesta en marcha del sistema de workflow Bonita. A continuación se describirá paso a paso este proceso.

Como primer paso se debe chequear la instalación del Java Development Kit (JDK) 1.6 e instalar el software ANT (apache-ant-1.7.0) y declarar la variable de ambiente “ANT_HOME = C:\apache-ant-1.7.0” como se muestra en la figura 2.2.



Figura 2.2 Definición de la variable de ambiente ANT_HOME.

Luego se debe ejecutar el instalador de Jonas 4.8.4 y aparecerá la figura 2.3, donde se debe dar clic en “Next” para comenzar.



Figura 2.3 Interfaz de instalación de Jonas 4.8.4.

Después aparecerá la figura 2.4, aquí se realiza el seteo de ANT y BCEL (Byte Code Engineering Library).

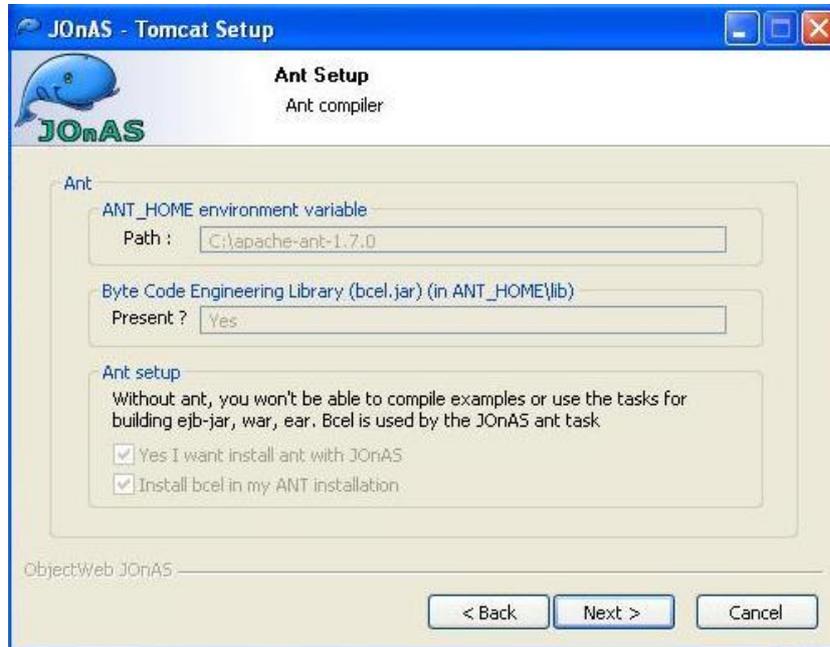


Figura 2.4 Configuración del ANT y BCEL (Byte Code Engineering Library).

Luego se selecciona el tipo de instalación (se recomienda seleccionarla completa) y el directorio de ubicación, como se muestra en la figura 2.5 y figura 2.6 respectivamente.

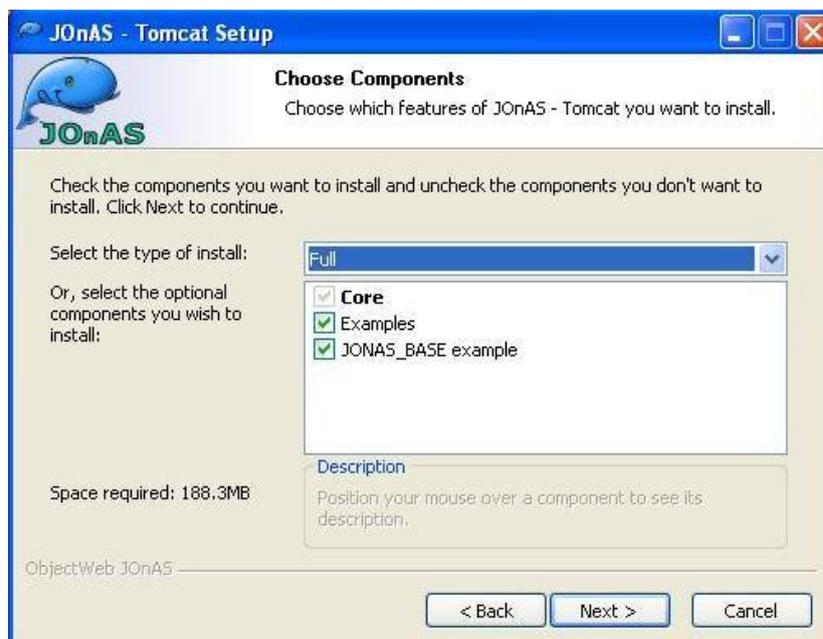


Figura 2.5 Selección del tipo de instalación.

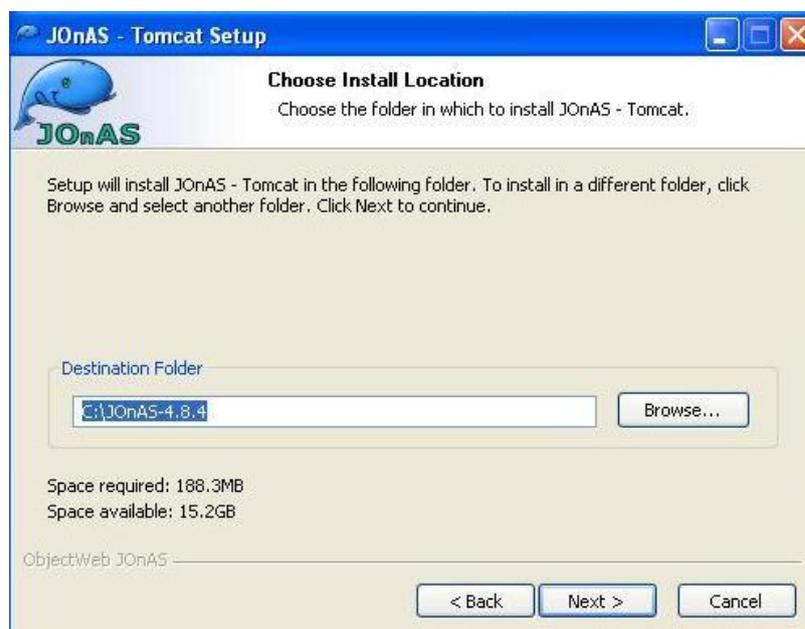


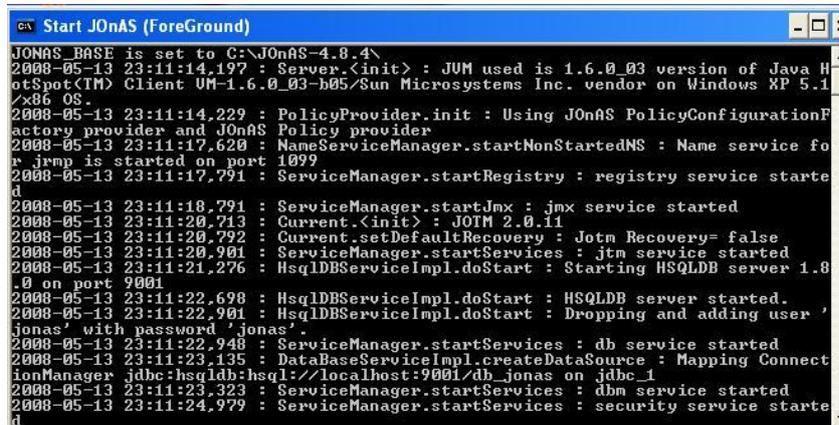
Figura 2.6 Selección del directorio en el cual se va a instalar Jonas 4.8.4.

Después se procede con la instalación a través de la opción "Install" tal como se muestra en la figura 2.7.



Figura 2.7 Interfaz de ejecución de la instalación de Jonas 4.8.4.

Una vez que Jonas este instalado, debemos iniciar el servidor. El puerto que usa Jonas por defecto es el 9000 (figura 2.8).



```
Start JOnAS (ForeGround)
JONAS_BASE is set to C:\JOnAS-4.8.4\
2008-05-13 23:11:14.197 : Server.<init> : JUM used is 1.6.0_03 version of Java H
otSpot(TM) Client VM-1.6.0_03-b05/Sun Microsystems Inc. vendor on Windows XP 5.1
/x86 OS
2008-05-13 23:11:14.229 : PolicyProvider.init : Using JOnAS PolicyConfigurationF
actory provider and JOnAS Policy provider
2008-05-13 23:11:17.620 : NameServiceManager.startNonStartedNS : Name service fo
r jtmp is started on port 1099
2008-05-13 23:11:17.791 : ServiceManager.startRegistry : registry service starte
d
2008-05-13 23:11:18.791 : ServiceManager.startJmx : jmx service started
2008-05-13 23:11:20.713 : Current.<init> : JOTM 2.0.11
2008-05-13 23:11:20.792 : Current.setDefaultRecovery : Jotm Recovery= false
2008-05-13 23:11:20.901 : ServiceManager.startServices : jtm service started
2008-05-13 23:11:21.276 : HsqlDBServiceImpl.doStart : Starting HSQLDB server 1.8
.0 on port 9001
2008-05-13 23:11:22.698 : HsqlDBServiceImpl.doStart : HSQLDB server started.
2008-05-13 23:11:22.901 : HsqlDBServiceImpl.doStart : Dropping and adding user
'jonas' with password 'jonas'
2008-05-13 23:11:22.948 : ServiceManager.startServices : db service started
2008-05-13 23:11:23.135 : DataBaseServiceImpl.createDataSource : Mapping Connect
ionManager jdbc:hsqldb:hsq://localhost:9001/db_jonas on jdbc_1
2008-05-13 23:11:23.323 : ServiceManager.startServices : dbm service started
2008-05-13 23:11:24.979 : ServiceManager.startServices : security service starte
d
```

Figura 2.8 Levantamiento del servidor Jonas 4.8.4.

En la siguiente figura 2.9 se muestra la pantalla de bienvenida.



Figura 2.9 Interfaz de bienvenida de Jonas 4.8.4.

Podemos ir a la pantalla de administración de Jonas a través de <http://localhost:9000/jonasAdmin> con usuario “jonas” y password “jonas”.

En las aplicaciones se muestran las que están publicadas (deploradas) más un conjunto de aplicaciones que podemos deplorar, las cuales muestran en la siguiente figura 2.10 y vienen como parte de la instalación.

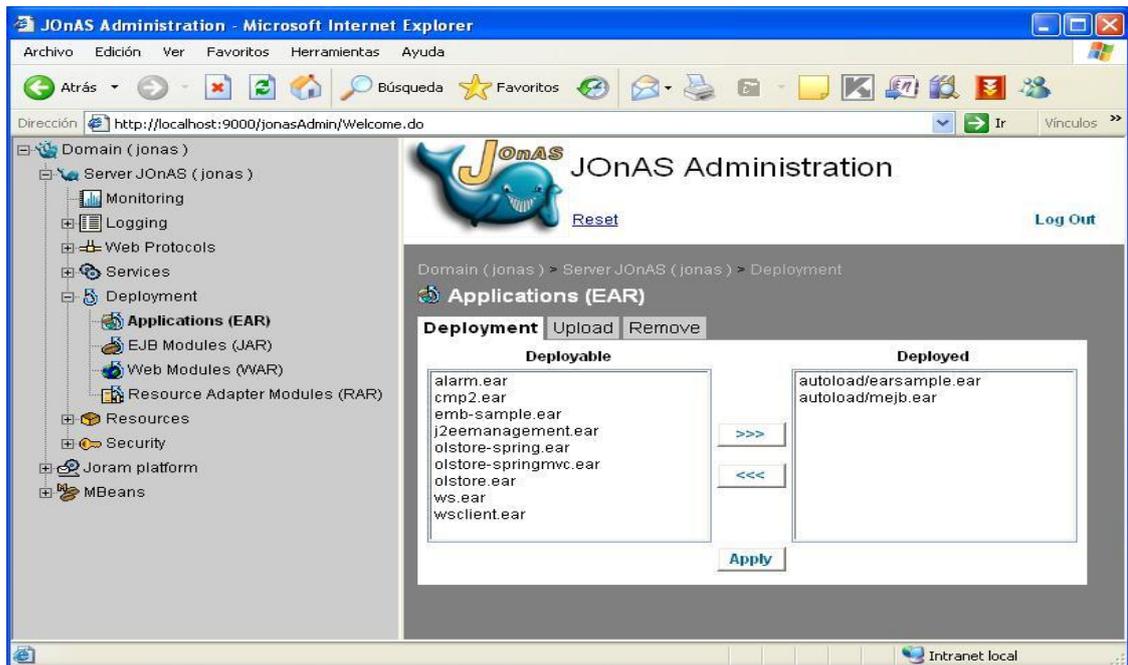


Figura 2.10 Aplicaciones publicadas en Jonas 4.8.4.

Después de haber cumplido todos los pasos anteriores se puede proceder a la instalación de Bonita. Antes de comenzar debemos asegurarnos que Jonas no se encuentre activo y tener declaradas las siguientes variables de ambiente:

JAVA_HOME = C:\Archivos de programa\Java\jdk1.6.0_03.

JONAS_ROOT = C:\JOnAS-4.8.4.

JONAS_BASE = C:\JOnAS-4.8.4\BSOA_base. Crear el directorio BSOA_base en C:\JOnAS-4.8.4.

ANT_HOME = C:\apache-ant-1.7.0.

Como se muestra en la figura 2.11.

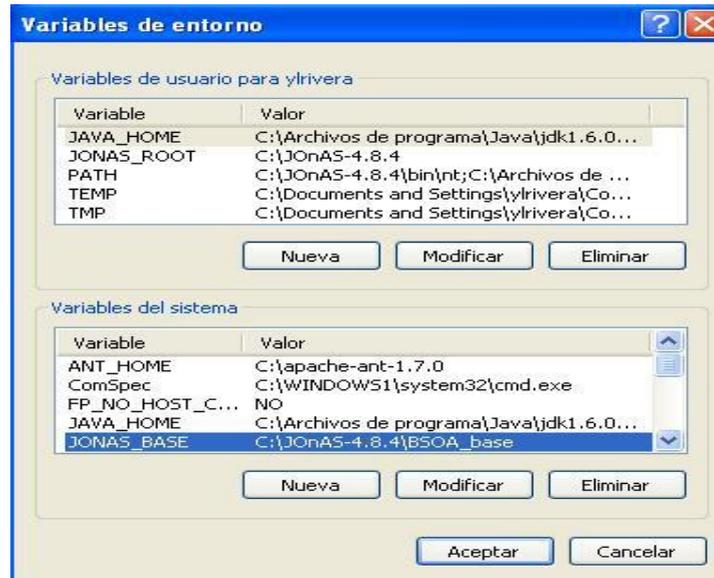


Figura 2.11 Declaración de las variables de ambiente necesarias para la instalación de Bonita 3.1.

Luego se ejecuta el instalador para comenzar, tal como se muestra en la figura 2.12, más adelante se selecciona la ubicación de la instalación y los puertos que por defecto se van a usar (figura 2.13).



Figura 2.12 Interfaz de instalación del sistema Bonita Workflow 3.1.

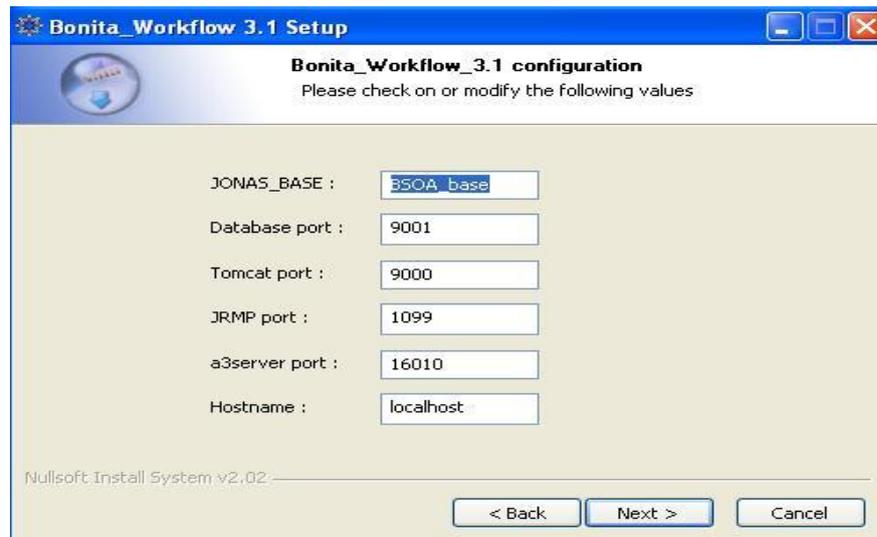


Figura 2.13 Configuración del sistema Bonita Workflow 3.1.

También es necesario especificar una ubicación para los respaldos (backups) como se muestra en la figura 2.14.

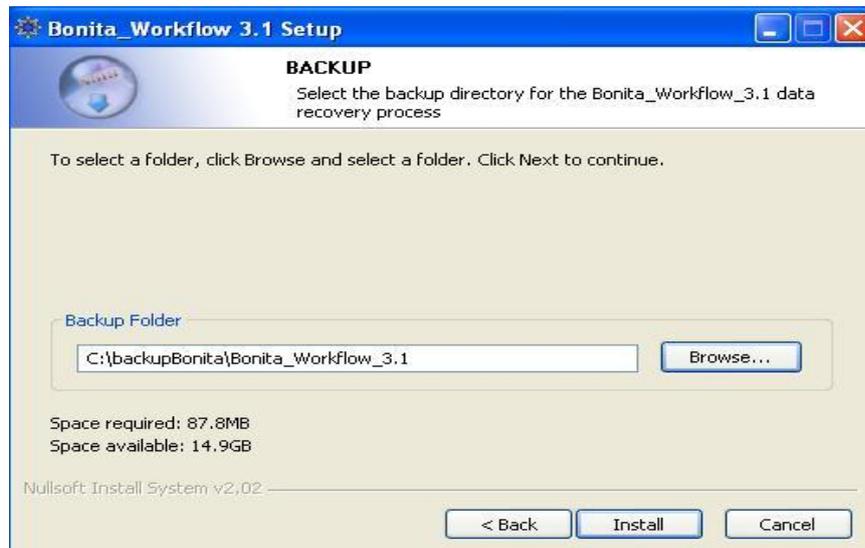


Figura 2.14 Definición del directorio de salvallas del sistema Bonita Workflow 3.1.

2.5.1 Ejecutar Bonita.

Para comenzar a trabajar con Bonita el servidor Jonas debe estar detenido, desde que se instala Bonita el servidor debe ser inicializado desde el nuevo acceso directo. A través de la ruta Inicio/ Programas/ Bonita_Workflow/ BonitaWorkflow Start.

Ya que en el caso que el servidor Jonas estuviera levantado dará un error. Después de haber chequeado todas las indicaciones anteriores, se ejecuta el sistema Bonita como se muestra en la figura 2.15.



```
S-4.8.4\BSOA_base\work\apps\jonas\jiapAdmin_2008.05.13-23.40.27\jiapAdmin.war available at the context /jiapAdmin.
2008-05-13 23:48:49.080 : AbsJWebContainerServiceImpl.registerWar : War /C:/JOnAS-4.8.4\BSOA_base\work\apps\jonas\jiapAdmin_2008.05.13-23.40.27\proed.war available at the context /proed.
2008-05-13 23:48:49.095 : EarServiceImpl.deployEar : Ear file:/C:/JOnAS-4.8.4\BSOA_base\apps\autoload\jiapAdmin.ear available.
2008-05-13 23:48:50.142 : JContainer.addBean : MEJB available
2008-05-13 23:48:50.205 : JContainer.addBean : DomainMEJB available
2008-05-13 23:48:51.283 : AbsJWebContainerServiceImpl.registerWar : War /C:/JOnAS-4.8.4\BSOA_base\work\apps\jonas\mejb_2008.05.13-23.39.14\mejb.war available at the context /mejb.
2008-05-13 23:48:51.314 : EarServiceImpl.deployEar : Ear file:/C:/JOnAS-4.8.4\BSOA_base\apps\autoload\mejb.ear available.
2008-05-13 23:48:51.314 : ServiceManager.startServices : ear service started
2008-05-13 23:48:51.314 : MailServiceImpl.doStart : JOnAS: Cannot create mail factory MailSession1 : java.lang.RuntimeException: Cannot find properties for C:\JOnAS-4.8.4\BSOA_base\conf\MailSession1.properties
2008-05-13 23:48:51.314 : MailServiceImpl.doStart : Please check the MailSession1.properties file
2008-05-13 23:48:51.314 : ServiceManager.startServices : mail service started
El servidor de aplicaciones JOnAS 'jonas' version 4.8.4 esta disponible
2008-05-13 23:48:56.596 : Server$1.run : El servidor de aplicaciones JOnAS 'jonas' esta lanzado en rmi/jrmp
```

Figura 2.15 Levantamiento del sistema Bonita Workflow 3.1.

Luego se debe ir a la administración de Jonas y verificar que las aplicaciones de Bonita estén publicadas, como se muestra en la figura 2.16.

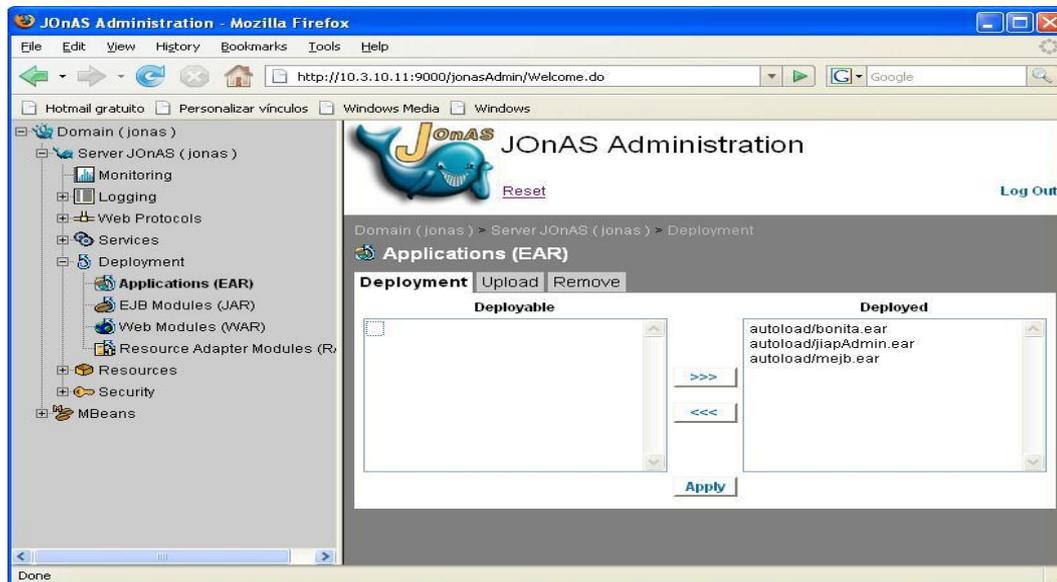


Figura 2.16 Verificación de la publicación del sistema de Bonita Workflow 3.1.

Se puede acceder a la consola de administración a través de la url <http://localhost:9000/jiapAdmin>, con el usuario “bsoa” y password “bsoa”. Una vez que el usuario se haya autenticado se mostrará la pantalla de bienvenida (figura 2.17).

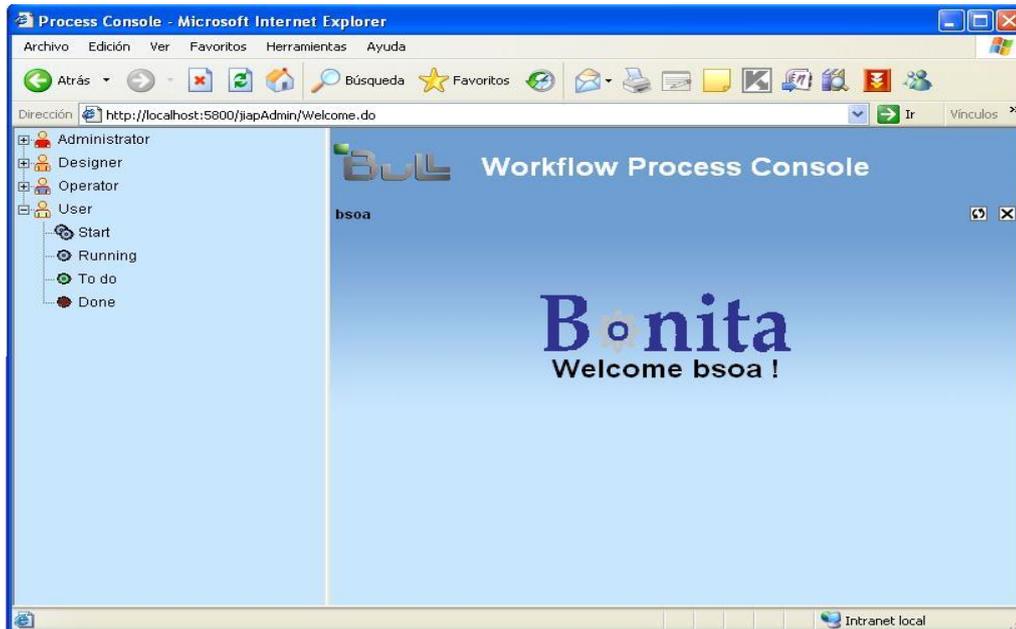


Figura 2.17 Interfaz de inicio del sistema Bonita Workflow 3.1.

Una vez que Bonita está instalada, el datasource realm “*dsrlm_jiap_1*” contiene toda la información para la administración de los usuarios. Este datasource trabaja con la base de datos de los usuarios, que por defecto es hsql.

Durante la instalación de Bonita ocurren varios procesos:

- Se crea y se inicializa el datasource realm (mediante la creación y configuración de los ficheros *jiapadmin.properties* y *bonita.properties*).
- Se agregan algunos datasource realm asociados a la seguridad (actualizando los ficheros *server.xml* y *jonas-realm.xml*).

Bonita define cuatro grupos de usuarios para la realización de las principales funciones del sistema de workflow: usuario, operador, diseñador y administrador. Cada grupo de usuario tiene un rol específico que define las acciones que debe hacer:

Usuario:

- Crea instancias de un modelo de procesos definido.
- Monitorea y activa las instancias que están en ejecución, así como las actividades a realizar.

Operador:

- Crea instancias de un modelo de procesos definido.
- Configura las trazas, logs y la historia de una instancia específica.
- Inicializa y detiene actividades de una instancia determinada.

Diseñador:

- Es el encargado de crear, modificar e importar los modelos de procesos del workflow.

Administrador:

- Administra la seguridad del datasource realm para el acceso de los usuarios al sistema de workflow Bonita, agrega o elimina usuarios y define sus roles.

Configuración del acceso al sistema Bonita Workflow.

Bonita permite cambiar el datasource realm e incluso cambiar el modo de autenticación mediante el uso del directorio activo (LDAP), posibilita modificar el cambio del tipo de base de datos asociado al datasource realm.

Para gestionar el datasource realm o LDAP realm se debe acceder a <http://localhost:9000/jonasAdmin/> (Domain (jonas) / Server Jonas (jonas)/ Security)

(**anexo No.4**).

Para el acceso a la herramienta Jonas hay que utilizar un usuario que sea miembro del perfil *jonas_admin* (usuario “bsoa” y password “bsoa”). Una vez finalizado el proceso es necesario reiniciar sistema de workflow.

Una vez configurado el datasource realm o LDAP realm se debe acceder a <http://localhost:9000/jiapAdmin/> (Administrador/ User Management) (**anexo No.5**) y seleccionar el datasource realm que vamos a utilizar.

La configuración de la base de datos del sistema de workflow está en <http://localhost:9000/jiapAdmin/> (Administrador/ Engine Databases) (**anexo No.6**).

Envío de notificaciones utilizando correo tradicional y sistema de notificación instantánea.

Bonita permite el envío de correos tradicionales o notificaciones instantáneas (jabber). Para realizar la configuración del correo se debe acceder a <http://localhost:9000/jonasAdmin> (Domain (jonas)/ Server Jonas (jonas)/ Resources/Mail) (**anexo No.7**), además es necesario definir los datos de un nuevo *Mail Factory* y publicarlo. Se necesita además un servidor de correo y un cliente de correo.

Para configurar el jabber se debe ir a <http://localhost:9000/jonasAdmin> (Domain (jonas)/ Server Jonas (jonas)/ Resources/JMS) (**anexo No.8**) y definir un nuevo *Jms destination*. Se necesita además un servidor de jabber y un cliente jabber.

Es necesario reiniciar el servidor de Bonita que para que cada modificación sea actualizada.

Diseño gráfico de procesos.

Cada sistema para la gestión del flujo de trabajo debe contar con una herramienta gráfica que permita modelar fácilmente los procesos de negocio, así como definir el flujo de trabajo y las tareas desde el comienzo hasta el final, sin la necesidad de programar. Bonita utiliza la herramienta ProEd para la definición del flujo de procesos, que es de fácil configuración y trabaja con el estándar XPDL. Permite diseñar toda la estructura básica que caracterizan los modelos de los sistemas de workflow, definir las actividades, transiciones, sub-procesos, participantes, iteraciones, hooks, ruteo de actividades y bloqueo de actividades. La definición del flujo de procesos es guardada en formato XPDL y luego es importada a la consola de administración de Bonita.

Para poder inicializar el ProEd se debe ser miembro del perfil diseñador, a través de la dirección <http://localhost:9000/jiapAdmin/> (Designer/ Workflow Editor (ProEd)) (**anexo No.9**). El ProEd se ejecuta vía Java Web Start²⁰ y puede iniciarse en modo “online” y “offline”. Se recomienda iniciar en modo “online” para que el ProEd pueda acceder a algunos recursos que se encuentran en la base de datos del sistema de workflow, por ejemplo el XForm Editor, hooks, ficheros XPDL que están en el repositorio, entre otros. Para visualizar el flujo de actividades ProEd utiliza dos vistas: “vista de participantes”, en la cual se muestran las

²⁰ Ver Glosario.

actividades y los participantes que las ejecutan (**anexo No.10**) y la “vista de actividades”, en la que se exponen las actividades y sus transiciones (**anexo No.11**).

La creación de un proyecto nuevo en el ProEd, se hace a través de la dirección (File / New), definiendo todos los parámetros que están presentes en un modelo de flujo de procesos. ProEd define una serie de símbolos para cada uno de sus componentes (**anexo No.12**).

El XForm Editor se activa cuando se define al menos un atributo del proyecto, es usado para diseñar formularios y enviar datos en formato XML. La inicialización del XForm Editor se hace a través de la dirección (Process/Properties/Attributes/Edit Xform) y es importante que cada acción que se realice esté sincronizada con el servidor, para ello se deben guardar todas las modificaciones que se hagan. La interface de diseño principal del XForm Editor se muestra en la figura 2.18, permite definir los datos que son relevantes en el formulario y validarlos.

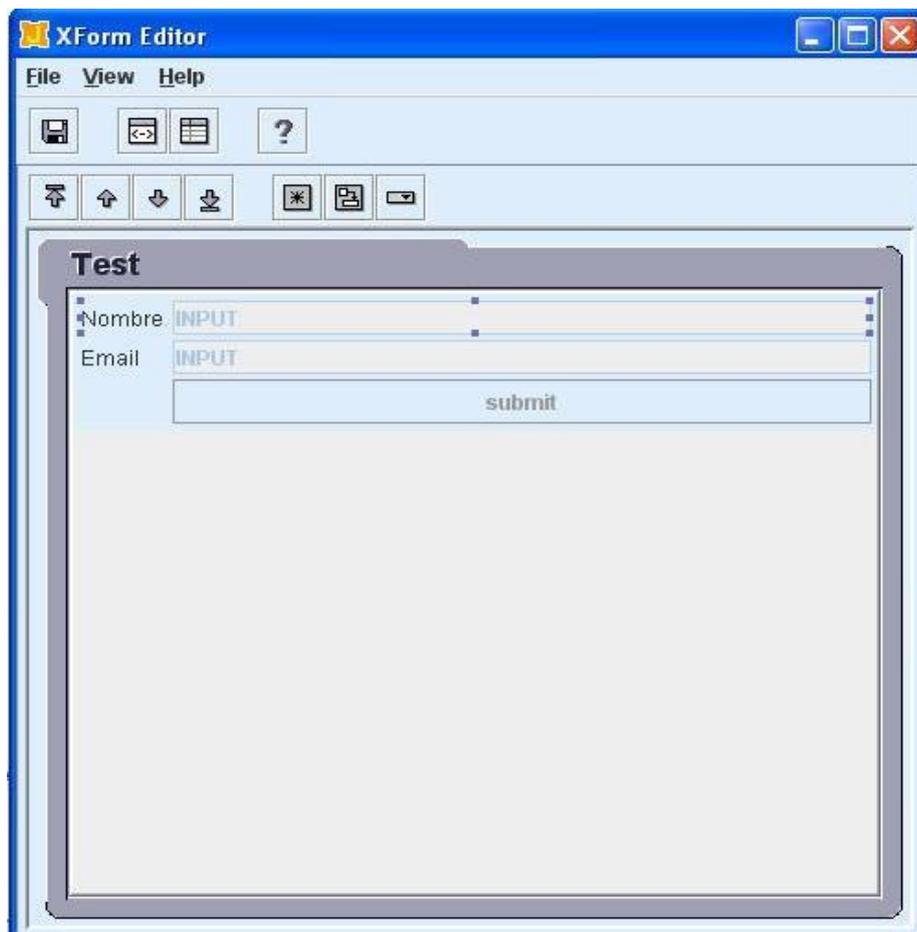


Figura 2.18 Interfaz de la herramienta XFormEditor

Cada actividad tiene asociado un grupo de elementos que la definen, como por ejemplo: nombre, tipo, modo de inicialización, participantes, atributos, hooks, entre otros (**anexo No.13**).

Las actividades pueden hacer uso del XForm Editor y definir sus propios atributos que estarán presentes en el formulario cuando se ejecute la actividad.

Trabajo con modelos de procesos definidos.

Una vez que se ha realizado el modelo de procesos es necesario importarlo. Esta acción la debe realizar el diseñador a través de la url <http://localhost:9000/jiapAdmin/> (Designer/ Import XPD) (**anexo No.14**).

Existen dos opciones para importar un XPD: desde un fichero local del sistema o desde un repositorio de workflow. Cuando se carga del fichero automáticamente el sistema de Bonita Workflow valida el modelo de los procesos que contiene, detectando posibles errores (**anexo No.15**).

La lista de los modelos de procesos disponibles va a estar en <http://localhost:9000/jiapAdmin/> (Designer/ Process models). Para comenzar a ejecutar el proceso se debe ser miembro del perfil usuario y acceder a <http://localhost:9000/jiapAdmin/> (User/ Start / Nombre del proceso) (**anexo No.16**).

Al ejecutarse el proceso se muestra el formulario previamente definido con el XForm Editor, permitiendo definir los valores de cada atributo y enviarlos (**anexo No.17**), acto seguido el sistema de workflow muestra si fue satisfactorio la creación de una instancia del proceso (**anexo No.18**).

El listado de todas las instancias se muestra en <http://localhost:9000/jiapAdmin/> (Operator/ Process Instances) (**anexo No.19**), aquí se pueden realizar varias operaciones como: eliminar una instancia, modificar sus datos, filtrar instancias de acuerdo a varios parámetros y terminar una instancia. En <http://localhost:9000/jiapAdmin/> (User/ Running/ Nombre Proceso) (**anexo No.20**) se muestran las instancias en ejecución, su estado y la próxima actividad a realizar. Se puede visualizar, chequear el estado y ejecutar las actividades asociadas a cada instancia a través de <http://localhost:9000/jiapAdmin/> (Operator/ Activities) (**anexo No. 21**).

A través de <http://localhost:9000/jiapAdmin/> (User/ To do) y <http://localhost:9000/jiapAdmin/> (User/ Done) se muestran las actividades que faltan por cumplirse y las que ya han culminado respectivamente.

Monitoreo

La funcionalidad “Setup” (**anexo No.22**) posibilita la administración de cuatro características de Bonita:

- Servicio de mensajes en java (“JMS”), usado por el sistema de workflow.
- Trazas técnicas de Bonita (“Trace”).
- Bonita log (“Log”).
- Creación del historial de cada instancia que hay finalizado (“History”).

Todas las características anteriores permiten chequear y obtener información detallada de los eventos ocurridos en el sistema.

2.6 Evaluación de la propuesta por especialistas.

Para la valoración de los resultados, se tomó en cuenta la autoevaluación de los especialistas acerca de su competencia y de las fuentes que le permiten argumentar sus criterios. Se aplicó el cuestionario (**anexo No.23**) para obtener las opiniones de especialistas sobre el procedimiento propuesto, los que recibieron una caracterización sintética de los aspectos relacionados con el procedimiento para el uso de la tecnología workflow.

Para la evaluación de los aspectos puestos a la consideración de los profesionales, se determinaron las siguientes categorías: muy adecuado (A), bastante adecuado (B), adecuado (C), poco adecuado (D), inadecuado (E). Una vez realizada la valoración del procedimiento por parte de los profesionales seleccionados, se hicieron los cálculos correspondientes.

La valoración final del grupo de especialistas, refleja que estos otorgan la categoría bastante adecuado (B) a la estructura del procedimiento y a la correspondencia entre el procedimiento y la concepción teórica. La evaluación del procedimiento fue un paso importante en la investigación, todos consideran que teóricamente el procedimiento está en condiciones de utilizarse.

2.7 Conclusiones del capítulo

Del análisis realizado se puede concluir que:

- El fundamento de la concepción teórica para la utilización de las herramientas para la gestión de flujo de trabajo se sustenta en el estudio de sus características y requerimientos.
- La concepción teórica expuesta se concreta a través de un procedimiento para la utilización de un sistema de gestión del flujo de trabajo.
- El procedimiento propuesto se concibe en cuatro momentos con sus objetivos y acciones a desarrollar.
- Se obtuvo una herramienta propuesta para la gestión del flujo de trabajo en los proyectos productivos.
- Los criterios aportados por los profesionales con experiencia, unidos a los resultados de la valoración del procedimiento demostraron que la propuesta es favorecedora de potenciar la utilización de las herramientas para la gestión de flujo de trabajo.

CONCLUSIONES GENERALES

Dándole solución y brindar una base esencial para el conocimiento e integración de la tecnología workflow en los proyectos productivos de la Facultad 3, mediante un procedimiento que estratégicamente contribuye a la materialización de la concepción teórica desde la profundización de un sistema workflow para potenciar la eficiencia en los procesos y en la estandarización del trabajo, se concluye la investigación cumpliéndose los objetivos generales.

Una vez concluida toda la investigación conjuntamente con la realización del procedimiento y su evaluación se tienen las siguientes conclusiones:

- Todos los objetivos que inicialmente se trazaron fueron cumplidos cabalmente.
- Se logró el estudio detallado de los conceptos y funcionalidades principales de los sistemas workflow's para un mejor entendimiento de ésta tecnología.
- Se realizó la propuesta de una herramienta workflow que se destaca por ofrecer altos niveles de funcionalidad, una comunidad de desarrollo brillante, madurez en el mercado de los sistemas workflow's y brindar soporte a la mayoría de los patrones workflow que se analizan.
- Se obtuvo un procedimiento para la gestión y configuración de la herramienta propuesta profundizando en el conocimiento básico que deben tener los usuarios para su utilización.
- Se obtuvo una evaluación positiva sobre el procedimiento realizado.

RECOMENDACIONES

Luego de concluir con la investigación de este trabajo se recomienda:

- Mantener un control sistemático de las tendencias actuales de los sistemas de workflow's, de forma tal que si surge alguno nuevo, o se hagan modificaciones a alguno de los existentes, se evalúe y se aplique para lograr una mejor calidad y control de los proyectos productivos.
- Seguir profundizando en elementos que enriquezcan el conocimiento acerca de la herramienta propuesta.
- Que el procedimiento para el uso de la herramienta propuesta sea respaldada por especialistas y que se use en los proyectos.
- Publicar los resultados de la investigación para ampliar el conocimiento acerca de las herramientas que gestionan el flujo de trabajo.
- Presentar el siguiente trabajo en eventos científicos.

BIBLIOGRAFÍA

1. **Wil van der Aalst, Kees van Hee.** *Workflow Management Models, Methods, and Systems.* Cambridge : Asco Typesetters, 2002. 0-262-01189-1.
2. **WFMC.** Workflow Management Coalition. [Online] <http://www.wfmc.org/>.
3. **GNU.** Portal de GNU. [Online] <http://www.gnu.org/home.es.html>.
4. **Iniciative, Open Source.** Portal Open Source Initiative. [Online] <http://www.opensource.org/>.
5. **Foundation, Free Software.** Portal de Free Software Foundation. [Online] <http://www.fsf.org/>.
6. **Microsystems, Sun.** Acerca de la tecnología Java. [Online] <http://java.com/es/about/>.
7. **WARIA.** Workflow And Reengineering International Association. [Online] <http://www.waria.com/index-waria.html>.
8. **Group, The Object Management.** Portal Object Management Group. [Online] <http://www.omg.org/>.
9. **OMG.** Recursos BPMN . [Online] <http://www.bpmn.org/>.
10. **Oasis.** Recursos BPEL. [Online] <http://www.oasis-open.org/committees/wsbpel/>.
11. **WFMC.** Recursos XPD. [Online] <http://www.wfmc.org/standards/xpdl.htm>.
12. **WfMC, WARIA.** Portal workflow. [Online] <http://www.e-workflow.org>.
13. **WFMC.** *Terminology & Glossary. The Workflow Magnagement Coalition Specification.* 1999. pp. 8-12.
14. **G.J. Houben, W.M.P. van der Aalst, K.M. van Hee.** *Modelling and analysing workflow using a petri-net based approach.* 1998. pp. 98—132. Vol. 2.
15. **Caro, José.** *Curso Doctorado Workflow.* 2004.
16. **S Jones, M Wilikens, P Morris , M Masera.** *Trust requirementes in e-business.* 2000. p. 42(12).
17. **Hollingsworth, David.** *The workflow reference model version 1.1.* 1995.
18. **Wil van der Aalst, Arthur ter Hofstede , Bartek Kiepuszewski, Alistair Barros.** *Workflows Patterns.* 2002.

19. **Patterns, Workflow.** Structured Workflow Patterns. [Online]
http://www.workflowpatterns.com/patterns/control/advanced_branching/wcp7.php,
<http://is.tm.tue.nl/research/patterns/>.
20. **Hispano, Java.** Documentación . [Online] <http://www.javahispano.org/>.
21. **WARIA.** Workflow Comparative Study. [Online] <http://www.waria.com/books/study-volume1-2.htm>.
22. **OpenWFE.** Manual de Openwfe. [Online] <http://www.openwfe.org/manual>.
23. **Web, Object.** Bonita Documentation. [Online]
<http://wiki.bonita.objectweb.org/xwiki/bin/view/Main/Documentation>.
24. **Valdes, Miguel.** Bonita v2 Series: Parte One. [Online]
<http://www.theserverside.com/tt/articles/article.tss?l=BonitaPart1>.
25. **Shark, Enhydra.** Acerca de Enhydra Shark. [Online]
<http://www.enhydra.org/workflow/shark/index.html>.
26. Diccionario. [Online] Concepto de procedimiento. <http://www.diclib.com/cgi-bin/d1.cgi?base=alkonageneral&page=showid&id=80887>.
27. **Viamontes, MsC. Vilfredo Avalo.** *Procedimiento metodológico de educación patriótica en las Organizaciones Sociales Especiales en la escuela primaria.* 2005-2006. p. 14. Evento pedagogia 2007.
28. **Petia Wohed, Birger Andersson, Arthur H.M. ter Hofstede, Nick Russell, and Wil M.P. van der Aalst.** *Patterns-based Evaluation of Open Source BPM Systems: The Cases of jBPM, OpenWFE, and Enhydra Shark.* 2007.
29. **Inisiative, Workflow Patterns.** Workflow Patterns. [Online]
<http://www.workflowpatterns.com/evaluations/index.php>.
30. **ObjectWeb.** Bonita. [Online]
<http://wiki.bonita.objectweb.org/xwiki/bin/view/Main/References>.

GLOSARIO

Apache License: Es una licencia de software libre creada por la *Apache Software Foundation*. No es una licencia copyleft, ya que permite el uso y distribución del código fuente para software libre y software propietario.

API: *Application Programming Interface - Interfaz de Programación de Aplicaciones* es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

BPMI: Es una disciplina que se enfoca al diseño de los procesos y las posibilidades de ejecutarlos, de forma tal que respondan a las necesidades de los clientes.

BSD: Licencia de software libre resultado del proyecto llamado *Berkeley Software Distribution* en la universidad del mismo nombre en California en la década de los 70s. La *Free Software Foundation* la considera compatible con la GPL.

FLOSS: Su siglas en inglés "*Free/Libre Open Source Software*" hace mención a la idea de la filosofía del Software Libre, donde "*Free*" menciona el concepto de "*Libre*", y no la idea del software gratis, de ahí lo de "*F/L*", ya que se tiende a confundir.

FOSS: Es un concepto que designa al "*Free and Open Source Software*", es decir, al "*Software Libre y de Código Abierto*", sin hacer distinciones entre sus diferencias.

GPL: Es una licencia creada por la *Free Software Foundation* a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

Hooks: En el contexto de Bonita Workflow los hooks son clases de javas externas que controlan las operaciones tanto al nivel de proceso, como de actividades.

IBM: *International Business Machines* es conocida como el *Gigante Azul* una empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática.

Java: Tecnología, lenguaje de programación creado por Sun Microsystems.

Java Beans: Es un componente que se puede reutilizar y que puede ser manipulado visualmente por una herramienta de programación en lenguaje java.

Java Enterprise Edition: Siglas en inglés de Java 2 Platform Enterprise Edition (edición empresarial de la plataforma Java 2). Comprende un conjunto de especificaciones y funcionalidades orientadas al desarrollo de aplicaciones empresariales.

Java Web Start: Es un software que permite descargar y ejecutar aplicaciones java desde la web, garantiza que se está ejecutando la última versión de la aplicación y elimina complejos procedimientos de instalación o actualización.

JBOSS: Es un servidor de aplicaciones J2EE de código abierto implementado en java puro.

JDBC: *Java Database Connectivity*, un API que permite la ejecución de operaciones sobre base de datos desde el lenguaje de programación java independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.

JDK: *Java Development Kit* es un software que provee herramientas de desarrollo para la creación de programas en java.

Jonas: El servidor de aplicaciones Jonas, desarrollado por el consorcio *ObjectWeb*, fue el primer servidor de aplicaciones libre en lograr certificación oficial de compatibilidad con J2EE.

JRE: *Java Runtime Environment* (entorno en tiempo de ejecución java) es un conjunto de utilidades que permite la ejecución de programas java sobre todas las plataformas soportadas.

JSP: *JavaServer Pages* es una tecnología java que permite generar contenido dinámico para web, en forma de documentos html, xml o de otro tipo.

JSR: *Java Specification Request*, son documentos formales que describen las especificaciones y tecnologías propuestas para que sean añadidas a la plataforma java.

LDAP: Es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

LGPL: Es una licencia de software creada por la *Free Software Foundation*, se aplica a cualquier programa o trabajo que contenga una nota puesta por el propietario de los derechos del trabajo estableciendo que su trabajo puede ser distribuido bajo los términos de esta.

MIT License: Permite reutilizar el software tanto para ser software libre como para ser software no libre, permitiendo no liberar los cambios realizados al programa original. También permite licenciar dichos cambios con licencia BSD y GPL.

Mozilla Public License: Es una licencia de código abierto y software libre. Fue desarrollada originalmente por *Netscape Communications Corporation* y más tarde su control fue traspasado a la *Fundación Mozilla*. MPL deja abierto el camino a una posible reutilización no libre del software, si el usuario así lo desea, sin restringir la reutilización del código ni el relicenciamiento bajo la misma licencia.

Oracle Corporation: Es una de las mayores compañías de software del mundo. Sus productos van desde bases de datos (Oracle) hasta sistemas de gestión.

RAM: Memoria de acceso aleatorio donde el computador guarda los datos que está utilizando en el momento presente. El almacenamiento es considerado temporal por que los datos y programas permanecen en ella mientras que la computadora este encendida o no sea reiniciada.

Ruby: Es un lenguaje de programación reflexivo y orientado a objetos.

Sun: *Sun Microsystems* es una empresa informática de Silicon Vallery, fabricante de semiconductores y software.

TIC: Se refieren al conocimiento necesarios que hacen referencia a la utilización de múltiples medios informáticos para almacenar, procesar y difundir todo tipo de información.

XML: Desarrollado por el *World Wide Web Consortium*, es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.