



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Investigación en Sistemas de Información



Modelado del Proceso de
Gestión de Cambios según ITIL
bajo el Software Libre Intalio

Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
por el Bachiller
Jorge Luis Valdez Daliz
para optar al título de
Licenciado de Computación

Tutor Dr. Pedro N. Bonillo R.

Octubre, 2009

ACTA

Quienes suscriben, miembros del Jurado designado por el Consejo de Escuela de Computación, para examinar el Trabajo Especial de Grado presentado por el Bachiller Jorge Luis Valdez Daliz CI. 16.900.730, con el título: “Modelado del Proceso de Gestión de Cambios según ITIL bajo el Software Libre Intalio”, a los fines de optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído como fue, dicho trabajo por cada uno de los miembros del jurado, se fijó el día 29 de Octubre del 2009 a las 5:30 p.m, para que el autor lo defendiera en forma pública, lo que se hizo en el Centro de Computación de la Facultad de Ciencias, mediante una presentación oral de su contenido, luego de lo cual respondió las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con la nota de ____ puntos.

En fe de lo cual se levanta la presente Acta, en Caracas a los 29 días del mes de Octubre del año 2009.

Profesor

Dr. Pedro N. Bonillo R. (Tutor)

Profesora Paola Saputelli
(Jurado)

Profesor Antonio Silva
(Jurado)

DEDICATORIAS Y AGRADECIMIENTOS

Dedicatorias

Este Trabajo Especial de Grado va dedicado a dos personitas que llenan mi corazón de luz y de alegría cada día; mis sobrinos José Jesús y Miguel Ángel. Es su existencia la que me motiva a seguir luchando. Sino fuese por ustedes no hubiese motivación para seguir.

También va dedicado a mis seres queridos que ya no se encuentran físicamente a mi lado; a mis abuelos y en especial a mi tío Luis Beltrán. Se que ustedes me protegen desde allá arriba.

Agradecimientos

En primer lugar doy gracias a mi madre celestial, protectora de Oriente y guardiana de mis pasos; la Virgen del Valle, por iluminarme toda mi vida y guiar mis pasos durante el transcurso de mi carrera estudiantil. ¡Todo honor y toda gloria a ti madre!

A mis padres Jesusita Daliz y Jesús Valdez por apoyarme toda la vida, por estar presentes en los momentos difíciles y brindarme todo su cariño incondicional. ¡Esto logro también es de ustedes!

A mi novia Naimar por ser mi confidente, mejor amiga, estar presente en los buenos y malos momentos, por compartir todas sus fuerzas conmigo. Gracias por darme todo tu cariño y amor. De nuevo: *Arigato Gosay Mashita Sam.*

Gracias a mi tutor Pedro Bonillo por apoyarme durante esta etapa de la carrera y compartir sus conocimientos conmigo.

A mi otogai Vanessa y a mi mana Lily, por ser las mejores amigas que alguien puede tener. Gracias por compartir grandes momentos y regalarme sonrisas y abrazos cuando lo necesite. ¡Ustedes saben cuanto las quiero!

A mis hermanos José, Miguel y Sixmary por apoyarme toda la vida de diferentes maneras. Se les aprecia.

A los muchachos José Ortiz, Alexis Colmenares, Enrique Talavera y José Mantilla por ser mis camaradas del alma... Ya saben la consigna: "Que viva el Cacique".

Gracias especiales a mi Sensei David Ochoa por brindarme la oportunidad de ingresar a esta prestigiosa casa de estudios. ¡Oss Sensei!

RESUMEN

La Librería de Infraestructura de Tecnología de la Información (ITIL - *Information Technology Infrastructure Library*) propone un conjunto de mejores prácticas para gestionar el proceso de cambios en las organizaciones, debido a que en muchas de ellas no existe un control eficiente de los cambios que se realizan. Para implementar este proceso de manera eficiente, se realizó la evaluación de herramientas de software libre y de sistemas de gestión de procesos de negocios, seleccionando a Intalio BPMS 5.2 para cumplir con el objetivo de este Trabajo Especial de Grado.

El uso de esta herramienta permitió modelar el proceso; definir valores a los parámetros, tales como los nombres o roles de los actores de las tareas; y poner en ejecución el proceso, sin tener que esperar a ningún desarrollo de programación.

Finalmente se obtuvo el modelo del proceso en una plataforma que permite gestionarlo y adaptarlo a cualquier organización en base a las mejores prácticas propuestas por ITIL.

Palabras Claves: BPM, BPMS, Intalio, ITIL, Proceso de Gestión de Cambios, SOA, Web Services.

ÍNDICE

ACTA	I
DEDICATORIAS Y AGRADECIMIENTOS	II
RESUMEN	IV
ÍNDICE	¡ERROR! MARCADOR NO DEFINIDO.
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	VIII
INTRODUCCIÓN	1
1.2 <i>Planteamiento del Problema</i>	7
1.3 <i>Manifestaciones y Evidencias del Problema</i>	8
1.4 <i>Objetivos</i>	9
1.4.1 <i>Objetivo General</i>	9
1.4.2 <i>Objetivos Específicos</i>	9
1.5 <i>Justificación de la Investigación</i>	10
1.5.1 <i>Justificación tecnológica</i> :.....	10
1.5.2 <i>Justificación a nivel de la organización</i>	11
1.5.3 <i>Justificación económica</i>	11
1.6 <i>Alcance</i>	12
1.7 <i>Limitaciones</i>	12
1.8 <i>Delimitación</i>	13
CAPÍTULO 2 – MARCO TEÓRICO	14
2.2 <i>Bases Teóricas</i>	15
2.2.1 <i>Gestión de Servicios de Tecnologías de Información</i>	15
2.2.2 <i>Biblioteca de Infraestructura de Tecnologías de la Información (ITIL)</i>	17
2.2.3 <i>Gestión de Cambios (Change Management)</i> :.....	20
2.2.3.1 <i>Conceptos Básicos</i>	21
2.2.3.2 <i>Proceso de Gestión de Cambios</i>	22
2.2.4 <i>Proceso de Negocio</i>	25
2.2.5 <i>Gestión de Procesos de Negocios (BPM – Business Process Management)</i>	26
2.2.6 <i>Sistemas de Gestión de Procesos de Negocio (BPMS – Business Process Management Systems)</i>	28
2.3 <i>Servicios Web (Web Services)</i>	30
2.3.1 <i>Definición de Servicio Web</i>	30
2.3.2 <i>Arquitecturas orientadas a Servicios (Service-oriented architecture – SOA)</i>	31
2.3.2.1 <i>Estándares y tecnologías subyacentes. Infraestructura básica</i>	33
2.3.3 <i>Arquitectura Tecnológica</i>	34
2.3.4 <i>SOAP (Simple Object Access Protocol)</i>	35
2.3.4.1 <i>Especificaciones de SOAP</i>	35
2.3.4.2 <i>Formato del mensaje</i>	36
2.3.5 <i>WSDL (Web Services Description Language)</i>	37
2.3.5.1 <i>Estructura de un Documento WSDL</i>	37
2.3.6 <i>UDDI (Universal Description, Discovery, and Integration)</i>	39
2.3.6.1 <i>Modelo de Información. Estructura de Datos de UDDI</i>	40
2.4 <i>Orquestación de Servicios Web</i>	41
2.4.1 <i>Protocolos de Negocios y Definición de Procesos</i>	43
2.4.2 <i>Servicio de Procesos y Socio de Servicios</i>	43
2.4.3 <i>Actividades Básicas y Actividades Estructuradas</i>	44
2.4.4 <i>Secuencias, Flujos y Acoplamientos</i>	44
2.4.5 <i>Orquestación y Actividades</i>	45
2.4.6 <i>Orquestación y Coordinación</i>	45
2.4.7 <i>Orquestación y SOA</i>	46
2.5 <i>SOAr</i>	47
2.5.1 <i>Definición y ventajas de SOAr</i>	47

2.5.2	SOAr y BPM.....	49
2.6	Intalio BPMS 5.2.1	49
2.6.1	Intalio Designer	52
2.6.2	Intalio Server.....	54
CAPÍTULO 3 – MARCO METODOLÓGICO.....		59
3.1	Metodología de desarrollo de software.....	59
3.1.2	Metodología de Gestión de los Procesos del Negocio sustentada en el uso de patrones.....	63
3.2.1	Tipo de Investigación	65
3.2.2	Método de Investigación Cualitativa.....	66
3.2.3	Tipo de Población	66
3.2.4	Técnicas e instrumentos de recolección de datos.....	67
CAPÍTULO 4 – RESULTADOS		68
4.1	Análisis de requerimientos	68
4.2	Diseño de diagramas.....	68
4.3	Modelado del Proceso de Gestión de Cambios.....	69
4.4	Configuración del motor de ejecución del BPMS	69
4.5	Ejecución del proceso de Gestión de Cambios	69
4.5.1	Cambio Estándar.....	72
4.5.2	Cambio Urgente	81
4.5.3	Clarificación de Petición de Cambio	91
CAPÍTULO 5 - CONCLUSIONES		93
REFERENCIAS BIBLIOGRÁFICAS.....		96
ANEXO A – DIAGRAMAS UML.....		99
ANEXO B – INSTALACIÓN Y CONFIGURACIÓN DE INTALIO BPMS 5.2.1		130
ANEXO C – NOTACIÓN BPMN.....		143
ANEXO D – CREACIÓN DE UNA TAREA EN LA ARQUITECTURA TEMPO		149
ANEXO E – DETALLES DEL MODELADO DEL PROCESO DE GESTIÓN DE CAMBIOS EN LAS ORGANIZACIONES DE TECNOLOGÍAS DE INFORMACIÓN CON LA HERRAMIENTA INTALIO BPMS 5.2.1.....		153
ANEXO F - COMPARACIÓN DE LAS TECNOLOGÍAS DE BPM ANALIZADAS.....		204
ANEXO G – DIAGRAMA DEL PROCESO DE GESTIÓN DE CAMBIOS MODELADO EN BPMN.		207
ANEXO H – METADATA DE LAS OPERACIONES DE WEB SERVICES		208
ANEXO I – CÓDIGO BPEL DEL PROCESO DE GESTIÓN DE CAMBIOS MODELADO EN INTALIO		214

ÍNDICE DE FIGURAS

Figura N° 1. Enfoque de la organización desde la perspectiva de ITIL.	18
Figura N° 2. Aspectos de la metodología de soporte al servicio según los estándares ITIL.	20
Figura N° 3. Pasos del Proceso de Gestión de Cambios	24
Figura N° 4. Relación entre Procesos de Negocios y los Sistemas de Información	26
Figura N° 5. Actividades que soporta BPM.....	27
Figura N° 6. Módulos que componen una solución BPMS.	29
Figura N° 7. Arquitectura Orientada a Servicios.	33
Figura N° 8. Pila de Servicios Web.	34
Figura N° 9. Formato de un mensaje SOAP	36
Figura N° 10. Mensaje SOAP	36
Figura N° 11. Estructura de un documento WSDL.....	37
Figura N° 12. Ejemplo de archivo WSDL	39
Figura N° 13. Estructura de Datos UDDI	41
Figura N° 14. Servicio de Procesos.....	43
Figura N° 15. Enlace de la capa BPM con SOAr	49
Figura N° 16. Componentes de Intalio BPM Community Edition.....	51
Figura N° 17. Diseñador Intalio Designer.....	52
Figura N° 18. Mapeador de Datos de Intalio Designer	53
Figura N° 19. Intalio Server	55
Figura N° 20. Editor WSDL.....	56
Figura N° 21. Ciclo de vida del Proceso Unificado Ágil ó AUP.....	62
Figura N° 22. Estructura de la Metodología de Gerencia de Procesos Sustentada en el Uso de Patrones.....	64
Figura N° 23. Cuadro de diálogo para exportar el proyecto al servidor de Intalio	71
Figura N° 24. Pantalla de acceso a Intalio Workflow	72
Figura N° 25. Proceso Change Request	73
Figura N° 26. Petición de Cambio	73
Figura N° 27. Asignación de desarrollador a la petición de Cambio.....	74
Figura N° 28. Tarea Proveer estimado a la petición de cambio.....	75
Figura N° 29. Asignar prioridad de estándar al cambio.....	75
Figura N° 30. Tarea Revisar Reporte de Desarrollador	76
Figura N° 31. Decisión de comenzar trabajo con el desarrollador asignado actualmente....	77
Figura N° 32. Tarea Change Request confirmed	78
Figura N° 33. Change Request confirmed	79
Figura N° 34. Notificación de Gestión de Cambio completada.....	80
Figura N° 35. Notificación de Cambio estándar aplicado.....	80
Figura N° 36. Visualizador de instancias.....	81
Figura N° 37. Proceso Change Request urgente	82
Figura N° 38. Petición de Cambio urgente	82
Figura N° 39. Asignación de desarrollador a la petición de Cambio urgente.....	83
Figura N° 40. Asignación de desarrollador a la petición de Cambio urgente.....	84
Figura N° 41. Tarea Proveer tiempo estimado para la petición de cambio urgente.....	85
Figura N° 42. Asignar prioridad de urgente al cambio	85

Figura N° 43. Tarea Revisar Reporte de cambio urgente de Desarrollador.....	86
Figura N° 44. Enviar estimado de trabajo al cliente	87
Figura N° 45. Tarea de estimado de petición de cambio	87
Figura N° 46. Tarea de aceptación o rechazo del estimado para el despliegue de la petición de cambio	88
Figura N° 47. Tarea de petición de cambio urgente confirmada	89
Figura N° 48. Confirmación de cambio urgente	89
Figura N° 49. Notificación de implementación de cambio urgente.....	90
Figura N° 50. Desarrollador indicando que necesita mayor información en la petición	91
Figura N° 51. Encargado del cambio le indica al Iniciador que el cambio necesita mayores detalles.	92

ÍNDICE DE TABLAS

Tabla N° 1. Características generales de Intalio Designer	54
Tabla N° 2. Características generales de Intalio Server	58
Tabla N° 3. Nombres de usuarios y contraseñas.....	70

INTRODUCCIÓN

Un entorno de negocios que evoluciona en un mundo que cambia constantemente debe desarrollarse e innovarse para ser competitivo. (Wells, 2008)

La madurez de un proceso de desarrollo, deriva en la correcta gestión de los continuos cambios que surgen en las organizaciones de Tecnología de Información, como resultados de nuevas necesidades del negocio, fusiones empresariales, cambios en la tecnología, entre otros. (Zayas de Glynx, 2007)

Si bien es cierto que muchas organizaciones trabajan en base a procesos, también es cierto que la mayoría no cuenta con una documentación formal de estos cambios y regularmente se llevan a cabo, basándose en lo que el personal involucrado con más experiencia establece.

Por esta razón, resulta de vital importancia tener claro al momento de analizar un proceso, cada una de sus actividades así como los actores participantes en cada una de estas.

Bajo el dominio de los Sistemas de Información es conveniente y beneficioso modelar los procesos utilizando estándares y métodos que permitan establecer una visión clara del alcance del proyecto, con la finalidad de satisfacer las necesidades de cambios organizacionales y poder gestionarlos de manera eficiente. (Smith et al., 2002)

Dentro de las organizaciones se definen modelos teóricos que permiten describir el desarrollo de actividades que generan valor al cliente final. (Felipe. 2009).

A estos modelos se les denomina *Proceso de Negocio*. Esteban Felipe define al proceso de negocio como “un conjunto de actividades y decisiones, iniciadas por la ocurrencia de un evento específico, que se ejecutan de forma coordinada para alcanzar un objetivo de negocio concreto”. (Felipe. 2009)

Frecuentemente los procesos y subprocesos de una organización son amplios, complejos y confusos lo que los convierte en una estructura difícil de comprender. El modelar estos procesos brinda la oportunidad de organizar y documentar la información sobre un sistema.

Cuando un proceso es modelado, con ayuda de una representación gráfica (diagrama de proceso), pueden apreciarse con facilidad las interrelaciones existentes entre distintas actividades, analizar cada actividad, definir los puntos de contacto con otros procesos, así como identificar los subprocesos comprendidos. Al mismo tiempo, los problemas existentes pueden ponerse de manifiesto claramente dando la oportunidad al inicio de acciones de mejora. (Mejia, et al., 2007)

En la actualidad existen diversos paquetes de software propietario que permiten modelar, diseñar, simular y monitorear los procesos de negocio. Sin embargo, estas no ofrecen las ventajas de las herramientas basadas en la filosofía de software libre.

Estas bondades son las siguientes:

- Permiten a los usuarios la libertad de estudiar cómo funcionan estas y adaptarlas a nuestras necesidades.
- Mejorarlas y hacer públicas las mejoras a los demás, de modo que toda una comunidad se beneficie.
- El uso de sus licencias sin necesidad de pagar por ellas. (Madariaga, 2009)

Al momento de seleccionar la herramienta para modelar el proceso de Gestión de Cambios se realizó una evaluación exhaustiva de los diversos sistemas de gestión de procesos de negocios.

La solución seleccionada para el modelado del proceso de Gestión de Cambios fue Intalio BPMS 5.2, ya que este satisfacía todas las características deseables en una herramienta de gestión de procesos de negocios (uso de licencia libre, diseñador BPMN, servidor BPEL, manejo de servicios web, soporte de diversos sistemas operativos y bases de datos, ambientes colaborativos *on-line*, desarrollo *zero-code*).

Para darle un enfoque sistemático al modelado del Proceso de Gestión de Cambios se utilizó el código de buenas prácticas propuestas por la Librería de Infraestructura de Tecnología de la Información (*ITIL - Information Technology Infrastructure Library*), lo que permitió desarrollar una estructura mas clara basada en un marco de referencia general que ayuda a definir objetivos y determinar el esfuerzo requerido para implementar un cambio.

En base a estos argumentos, la finalidad de este Trabajo Especial de Grado es modelar el proceso de Gestión de Cambios, siguiendo las mejores prácticas propuestas por la Biblioteca de Infraestructura de Tecnologías de la Información (*ITIL - Information Technology Infrastructure Library*) usando la herramienta de software libre Intalio.

Con el propósito de alcanzar los objetivos planteados, este documento se estructura de la siguiente manera:

Capítulo I, El Problema, se inicia la contextualización en espacio y tiempo del problema de investigación, sus manifestaciones y evidencias, luego se presentan los objetivos, la justificación y delimitación.

El Capítulo II se refiere al Marco Teórico, donde se entrega al lector los conocimientos necesarios para el desarrollo del trabajo. Aquí se especifican las definiciones, características, procesos y relaciones que dan sustento a las bases teóricas del proceso de Gestión de Cambios, incluyendo las características para la evaluación de la herramienta de software libre Intalio.

El Capítulo III comprende el Marco Metodológico, en el cual se detallan las unidades de análisis y de investigación; así como los métodos, técnicas y procedimientos de observación y recolección de datos necesarios para llevar a cabo la presente investigación. Además se indica la metodología de desarrollo de software utilizada al momento de modelar el proceso de Gestión de Cambios.

A continuación, el capítulo IV muestra los resultados de aplicar la metodología de desarrollo de software (análisis, diseño y modelado). A través de ilustraciones, se visualizará la ejecución del proceso modelado en Intalio.

Para finalizar, en el Capítulo V se realizará las conclusiones y recomendaciones de este Trabajo Especial de Grado.

A continuación se inicia con el planteamiento del objeto de estudio o contexto.

Capítulo 1 – EL PROBLEMA

En esta sección del Trabajo Especial de Grado se detalla toda la información referente a la identificación, caracterización, delimitación y justificación del problema, así como el objetivo general y los objetivos específicos que determinan el alcance del mismo.

1.1 Contexto

Los proyectos de informática se encuentran, por su naturaleza, en el centro de la innovación comercial y a través de su implementación provocan cambios en los procesos y las prácticas comerciales. (Vinogradsky, 2007)

Los procesos empresariales son complejos, dinámicos y necesitan ser mejorados constantemente. Están interrelacionados unos con otros y se desarrollan, de manera interna a la organización o, en muchos casos, interactuando con otras organizaciones (clientes, proveedores, socios, organismos). En cualquiera de los casos requieren ser gestionados de forma eficaz obteniendo una optimización del rendimiento y control sobre los procesos. (Bonillo, 2009)

En las organizaciones, es frecuente encontrar procesos de negocios que se asignan de forma manual, donde el supervisor se ocupa del reparto de tareas y monitorización del ciclo de vida del proceso, y una parte de los empleados más calificados deben abandonar sus funciones para resolver los errores y problemas que surgen. Esta situación conlleva a una falta de productividad, ineficiencia y, por ende, aumento del costo. (Ruiz, 2007)

En el actual y desafiante ambiente económico, las organizaciones buscan diferentes medios para mejorar la eficiencia de los procesos que impacten positivamente en el desempeño financiero.

Los procesos de negocio eficaces permiten que la organización entregue a los clientes productos y servicios de forma más rápida que los competidores y responda con agilidad a los cambios exigidos por el mercado. Para poder administrar sus procesos de negocios de manera eficiente, las empresas deben, en primer lugar, capturarlos y diseñarlos. Esos procesos deben documentarse mediante el uso de un lenguaje común y fácil de usar, a través de una herramienta de modelado que permita que los actores involucrados en el negocio describan y documenten fácilmente sus procesos.

A través del modelado de las actividades y procesos se logra un mejor entendimiento del negocio y muchas veces esto presenta la oportunidad de optimizarlos. La automatización de los procesos reduce errores, asegurando que los mismos se comporten siempre de la misma manera y dando elementos que permitan visualizar el estado de estos.

La administración de los procesos permite asegurar que los mismos esten ejecutándose eficientemente y obtener información que luego puede ser usada para mejorarlos.

Bajo este contexto, es necesario contar con un conjunto de herramientas y servicios que den soporte a los procesos de negocios en todo su ciclo de vida (análisis, diseño, simulación, implementación, monitoreo y optimización). A estas herramientas se les denomina *Sistemas de Gestión de Procesos de Negocios*, cuyo objetivo es asegurar la eficiencia y la eficacia de los procesos empresariales. (Owen, et al. 2003)

1.2 Planteamiento del Problema

Hoy en día, la mayoría de las empresas no llevan a cabo una gestión eficiente de sus cambios, lo que permite que no puedan adaptarse continua y rápidamente. (Pin, et al. 2006)

De acuerdo a las especificaciones de Marcelo Mejía (2007), los principales problemas que surgen en las organizaciones al momento de ejecutar el proceso de Gestión de Cambios sin el uso de una herramienta de gestión de procesos de negocios son los siguientes:

- Falta de coordinación entre personas y entre áreas funcionales, lo que provoca pérdida de tiempo y dinero.
- Las actividades del proceso se ejecutan solamente en base al conocimiento individual de las personas. Si alguna persona terminaba su relación laboral con la empresa, es necesario “reconstruir” el eslabón faltante haciendo uso de poca información.
- Ante un cambio, no se cuenta con la agilidad necesaria para responder a los requisitos del negocio.
- A falta de métricas de desempeño, las mejoras al proceso se llevan a cabo mediante elementos subjetivos.
- Las posibles inversiones en tecnologías se hacen sin contar con el sustento económico y/o funcional respecto a los beneficios esperados. (Mejía, et al. 2007)

Para solucionar los problemas mencionados anteriormente y permitir la flexibilidad y la agilidad necesaria en las empresas, se debe adoptar un conjunto de prácticas conocidas como Gestión de Procesos de negocios (*Business Process*

Management - BPM), que mediante el uso de Sistemas de Gestión de Procesos de Negocios (*Business Process Management Systems - BPMS*) permite modelar, integrar, medir y optimizar procesos de negocio.

Existen herramientas de modelado de procesos que no permiten en su totalidad analizar, diseñar y modelar de manera eficiente los procesos; bien sea por lo complejo de su manipulación, por los altos precios que se deben pagar por la licencia de uso o por ofrecer pocas funcionalidades a los usuarios. (Kotelnikov, 2008)

El proceso de Gestión de Cambios ha sido modelado en varias herramientas, tanto privadas como de código libre, pero pocas responden a la necesidad de usar conceptos como Web Services; notaciones y estándares mundiales de modelado como UML y BPMN; así como integración con otras herramientas.

Por lo tanto, en esta investigación se tomó en cuenta modelar este proceso utilizando una herramienta que cubre las necesidades anteriormente mencionadas.

1.3 Manifestaciones y Evidencias del Problema

Actualmente, la mayoría de las organizaciones no tienen un proceso formal de Gestión de Cambios que permita crear una estructura organizacional alineada a nuevas estrategias y procesos.

En algunas organizaciones se utilizan métodos para gestionar los cambios, que aunque cumple básicamente con los objetivos de negocio, no están diseñados para soportar un nivel de actividad elevado, ya que no permiten la adaptabilidad a los procesos de negocio.

En muchas empresas el tratamiento de los cambios no se da de la manera más eficiente. Esto se evidencia en aspectos como la carencia de un comité que apruebe los cambios; la falta de distinción entre la prioridad de los cambios y la ausencia de un mecanismo que permita guardar los registros de los cambios generados.

Asimismo, en las empresas que se gestionan los cambios de manera reactiva, se reduce la capacidad de solucionar problemas de forma eficiente, lo que genera deficiencias al ejecutar e implementar servicios presentes en sus librerías de hardware y software.

Los aspectos mencionados anteriormente, justifican la necesidad de utilizar una herramienta que permita optimizar el proceso de Gestión de Cambios, basándonos en un marco de referencia para la gestión de servicios.

1.4 Objetivos

Seguidamente se presenta el objetivo general y los objetivos específicos de esta investigación.

1.4.1 Objetivo General

En base a las situaciones analizadas anteriormente, el objetivo general de este Trabajo Especial de Grado es *modelar el Proceso de Gestión de Cambios según ITIL bajo el Software Libre Intalio*.

1.4.2 Objetivos Específicos

- Analizar el proceso de Gestión de Cambios desde la perspectiva sistémica centrada en los procesos y procedimientos de Tecnologías de Información establecidos por la Biblioteca de Infraestructura de las Tecnologías de Información.
- Diseñar procesos que permitan visualizar el proceso de negocio desde diferentes perspectivas.
- Modelar el proceso de Gestión de Cambios usando una notación gráfica basada en estándares internacionales.
- Configurar el motor de ejecución para ejecutar el proceso de gestión de cambios de ITIL

1.5 Justificación de la Investigación

1.5.1 Justificación tecnológica:

Normalmente los procesos de negocios son complejos; difíciles de modificar y no se encuentra al nivel del Analista de Negocios sino que están al nivel del Ingeniero del Software.

Se justifica el uso de un BPMS porque los procesos pueden ser adaptados fácilmente ya que estamos hablando a nivel de proceso de negocios y no a nivel tecnológico.

Además, el uso de un BPMS para modelar el proceso de Gestión de Cambios permite crear y manipular una lista de tareas y sobre esta lista definir roles que pueden invocar servicios web.

El manejo de BPMS nos da cierta facilidad en la interacción humano-computador.

Otro justificante tecnológico es que los BPMS nos permiten automatizar procesos de negocios sin la necesidad de desarrollar código como se hace en la programación clásica y rígida.

1.5.2 Justificación a nivel de la organización

Las características de los BPMS permitir crear agilidad en los procesos al momento de ser desplegados en la organización

Además permiten la adaptabilidad y la portabilidad de los procesos, sin importar bajo que arquitectura tecnológica se este trabajando.

1.5.3 Justificación económica

En el mercado existen herramientas basadas en software libre que son totalmente robustas y permiten modelar procesos de gestión de negocios, sin la necesidad de pagar un alto precio por el uso de su licencia.

Con el uso de BPMS *open source* no es necesario gastar los altos costos de una consultoría para programar la nueva forma de hacer el proceso.

También se justifica la investigación ya que se analiza un proceso organizacional, valiéndose de la tecnología en computación y utilizando el método científico, tal como lo establece el perfil de Licenciado en Computación en nuestra casa de estudio.

1.6 Alcance

El paradigma de BPM ofrece diferentes disciplinas al momento de gestionar procesos de negocios. En este Trabajo Especial de Grado se llegará hasta la automatización de un proceso de modelado del proceso orientada en una arquitectura orientada a servicios.

El objetivo de la investigación no contempla el uso de indicadores claves de desempeño, por lo tanto no se usará la disciplina de monitoreo de actividades de negocio (*Business Activity Monitoring - BAM*) contemplada por BPM.

El modelado del proceso se realizará bajo la herramienta de software libre Intalio, ya que el uso de estas ofrece potencialidades como desarrollo de cero código (*Zero code*), BPMN, BPEL y SOA.

El modelado del Proceso no se implementará en ningún tipo de organización.

1.7 Limitaciones

Para la realización de este trabajo se contó con un período de tiempo de dieciocho semanas (18). Las primeras 8 semanas fueron una limitante de tiempo para comenzar el modelado del Proceso de Gestión de Cambios, debido a que fue necesario la instrucción en el uso de la herramienta escogida en el seminario de este Trabajo Especial de Grado. Esto implicó el haber aprendido sobre la notación de modelado de proceso de negocios (Business Process Modeling Notation - BPMN) y el manejo de Servicios Web (Web Services) para el desarrollo de actividades durante el modelado del proceso.

Otra limitante fue la escasa información obtenida de las guías de entrenamiento de la tecnología seleccionada (Intalio BPMS), debido a que en la versión community no se ofrece el material de apoyo suficiente como en otras versiones del BPMS, donde es necesaria pagar una licencia cuyos costos son elevados.

Una de las mayores limitantes fue la falta de recursos económicos, ya que los entrenamientos para aprender a usar la herramienta Intalio y aprender sobre ITIL son excesivamente costosos.

El no poder observar la implementación del proceso de Gestión de Cambios en un ambiente real fue una limitante. Debido a esto, resulta más difícil establecer las relaciones entre los diferentes actores del proceso y las actividades donde estos se encuentran involucrados.

1.8 Delimitación

Este estudio está circunscrito al área de Gerencia de Procesos de Negocios y al Área de Sistemas de Información, dentro del dominio tecnológico existente en las ciencias computacionales para el año 2009: Software Libre, Ambientes de Soporte de Decisiones y Modelización de Procesos de Negocio.

Específicamente de los procesos de negocio existentes, se toma como caso de estudio el proceso de Gestión de Cambios de ITIL. Esto debido a que existen muchas tesis de los procesos de Gestión de Incidentes y Gestión de Problemas propuestos por ITIL, pero hay pocas que se han concentrado a resolver el problema de Gestión de Cambios.

Capítulo 2 – MARCO TEÓRICO

A continuación se presenta el marco teórico integrado por los antecedentes de la investigación y las bases teóricas.

2.1 Antecedentes de la Investigación

Para iniciar el marco teórico resulta conveniente citar los siguientes estudios previos vinculados al tema de investigación:

En el mes de Octubre del 2007, la Ingeniera de Sistemas, Dayana Marin elabora una propuesta de Tesis para desarrollar un prototipo del proceso de soporte de servicios gestión de cambios, de la librería de infraestructura de tecnologías de la información, aplicada en la empresa CANTV, destacando el problema planteado. Esta investigación sirvió de referencia para usar los conceptos básicos del Proceso de Gestión de Cambios de este Trabajo Especial de Grado.

Los Licenciados Vladimir Chávez y Raúl Matheus, en el año 2008, elaboraron como Trabajo Especial de Grado un sistema para la Administración de los Procesos de Soporte a los Servicios de ITIL usando la metodología Process Management. Esto se enfocó solo en los procesos de Gestión de Incidentes y Gestión de Problemas.

Al año siguiente, en el 2009, el Doctor Pedro Nolasco Bonillo Ramos, como requisito para su Doctorado en Ciencias de la Computación, elaboró una tesis doctoral donde propone el desarrollo de una Metodología para la Gestión de Procesos de Negocio sustentada en el uso de patrones. Esta tesis doctoral apoya y sustenta las bases teóricas de este Trabajo Especial de Grado, ya que esta metodología propuesta es la utilizada durante el desarrollo de esta investigación.

2.2 Bases Teóricas

Seguidamente se presentan al lector los conocimientos necesarios para el desarrollo del trabajo. Estos conocimientos están relacionados con conceptos sobre el Proceso de Gestión de Cambio en organizaciones, Sistemas de Gestión de Procesos de Negocios (específicamente Intalio) y arquitecturas tecnológicas necesarias para el desarrollo del sistema.

2.2.1 Gestión de Servicios de Tecnologías de Información

Las tecnologías de información han jugado un rol importante a lo largo de la historia de las organizaciones. La automatización de su gestión se ha convertido en una herramienta imprescindible y clave para empresas e instituciones. (Pin. et al, 2006).

La información es la fuente principal de negocio en las grandes organizaciones y ese negocio a su vez genera grandes cantidades de información. Su correcta gestión es de importancia estratégica y no debe considerarse como una herramienta más entre muchas otras.

La Gestión de Servicios de Tecnologías de Información (*ITSM ó IT Service Management*) es una disciplina que se utiliza ampliamente para la gestión de grandes, mediano y pequeños sistemas de tecnología de la información. ITSM está orientada hacia el cliente y se considera un consumidor enfoque de la gestión de una amplia variedad de servicios. ITSM trata de poner la relación de consumo en primer lugar, cambiando el énfasis de TI centrada en una filosofía a una filosofía de servicio al cliente. (Brito, 2008)

Van Haren (2002) indica que:

“los proveedores de los servicios de TI no pueden seguir manteniendo su enfoque en la tecnología y sus propias organizaciones, ahora tienen que considerar la calidad de los servicios que proveen y enfocarse en sus relaciones con los clientes”. (Van Harén, 2002)

Esto marca la característica principal de los ITSM.

Los servicios de TI representan generalmente una parte sustancial de los procesos de negocio.

Los objetivos de una buena gestión de servicios TI son:

- Proporcionar una adecuada gestión de la calidad.
- Aumentar la eficiencia.
- Alinear los procesos de negocio y la infraestructura TI.
- Reducir los riesgos asociados a los Servicios TI.
- Generar negocio.

Este nuevo paradigma basado en el servicio promueve la adopción de mejores prácticas bajo un enfoque de "Calidad de Servicio" y la oportunidad para el cambio del negocio con la aplicación de estándares actualizados.

La tendencia de Gestión de Servicio TI se basa en la promoción y soporte de aplicación de las mejores prácticas, marcos referenciales y estándares de aceptación internacional, tales como ISO/IEC 20000, ITIL, ITSCMM, COBIT, ISO/IEC -17799 – 2700X y otras.

Este Trabajo Especial de Grado se enfoca en la Biblioteca de Infraestructura de las Tecnologías de Información (*ITIL – Information Technology Infrastructure Library*) como marco de referencia sobre las mejores prácticas para

el diseño de los procesos de gestión de servicios de Tecnologías de Información, ya que esta presenta a la gestión del cambio como el proceso responsable de asegurar los cambios que se implementan en las empresas aplicando procedimientos, documentos y herramientas predefinidas y mundialmente probadas.

2.2.2 Biblioteca de Infraestructura de Tecnologías de la Información (ITIL)

Desarrollada a finales de 1980, la Biblioteca de Infraestructura de Tecnologías de la Información (ITIL) se ha convertido en el estándar mundial de facto en la Gestión de Servicios Informáticos. ITIL esta documentado por una serie de libros y manuales para capacitar y explicar las prácticas más beneficiosas para los servicios de TI. Las siglas de ITIL es una marca registrada, y los libros incluidos en la biblioteca de ITIL son protegidos por derechos de autor (pertenece a la Oficina de Comercio del Gobierno Británico ó OGC), pero es de libre utilización.

ITIL fue desarrollada al reconocer que las organizaciones dependen cada vez más de la Informática para alcanzar sus objetivos corporativos. Esta dependencia en aumento ha dado como resultado una necesidad creciente de servicios informáticos de calidad que se correspondan con los objetivos del negocio, y que satisfagan los requisitos y las expectativas del cliente. A través de los años, el énfasis pasó de estar sobre el desarrollo de las aplicaciones TI a la gestión de servicios TI. La aplicación TI (a veces nombrada como un sistema de información) sólo contribuye a realizar los objetivos corporativos si el sistema está a disposición de los usuarios y, en caso de fallos o modificaciones necesarias, es soportado por los procesos de mantenimiento y operaciones.

El objetivo de ITIL es que las tecnologías de las organizaciones tengan un gran valor, así como una alta calidad financiera en el día a día las operaciones de TI. ITIL indica procedimientos y operaciones sobre la infraestructura de TI.

ITIL fue producido originalmente a finales de 1980 y constaba de 10 libros centrales cubriendo las dos principales áreas de Soporte del Servicio y Prestación del Servicio. La biblioteca ITIL original incluía varios libros que abarca temas específicos en la Administración de Servicios de TI. Sin embargo, después de la publicación original, los libros en la biblioteca crecieron a más de 30 volúmenes complementarios que cubrían una numerosa variedad de temas, desde el cableado hasta la gestión de la continuidad del negocio. A partir del año 2000, se reestructuró la biblioteca para hacer acceder a la información necesaria para administrar sus servicios de manera más simple. Los libros centrales se han agrupado en dos, cubriendo las áreas de Soporte del Servicio y Prestación del Servicio, en aras de eliminar la duplicidad y mejorar la navegación. La figura N° 1 nos muestra las áreas en que se enfoca ITIL.

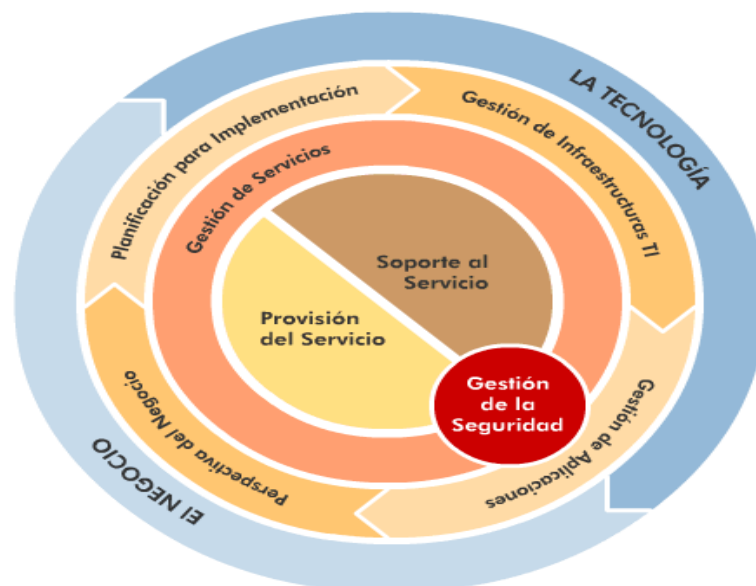


Figura N° 1. Enfoque de la organización desde la perspectiva de ITIL.

Fuente: Clinch, et al. 2004

El soporte al servicio se preocupa de de todos los aspectos que garanticen la continuidad, disponibilidad y calidad del servicio prestado al usuario. El proceso de soporte de servicio puede verse en la Figura N° 2.

A continuación se realizará un resumen de los procesos del soporte de servicio propuesto por ITIL:

- **Escritorio de Servicios:** representa el centro neurálgico de todos los procesos de soporte al servicio. Dentro de sus funciones se encuentra el de registrar y monitorizar incidentes, aplicar soluciones temporales a errores conocidos en colaboración con la Gestión de Problemas, colaborar con la Gestión de Configuraciones para asegurar la actualización de la Base de Datos de Administración de la Configuración (*CMDB – Configuration Management Database*), gestionar los cambios solicitados vía peticiones de servicio en colaboración con la Gestión de Cambios y versiones.
- **Gestión de Incidentes:** tiene como objetivo resolver cualquier incidente que cause una interrupción en el servicio de la manera más rápida y eficaz posible. No se preocupa de encontrar y analizar las causas subyacentes a un determinado incidente sino exclusivamente a restaurar el servicio. Esto es una actividad de la Gestión de Problemas.
- **Gestión de Problemas:** sus funciones principales son investigar las causas subyacentes a toda alteración, real o potencial, del servicio TI; determinar posibles soluciones; proponer las peticiones de cambio; realizar Revisiones Post Implementación.
- **Gestión de Configuraciones:** este proceso lleva el control de todos los elementos de configuración de la infraestructura de TI, realiza auditorías periódicas de configuración y proporciona información precisa sobre la configuración de TI a todos los diferentes procesos de Gestión. (Alcántara, 2007).

El otro proceso de Soporte al Servicio propuesto por ITIL es la Gestión de Cambios, el cual será objeto de estudio de este Trabajo Especial de Grado.

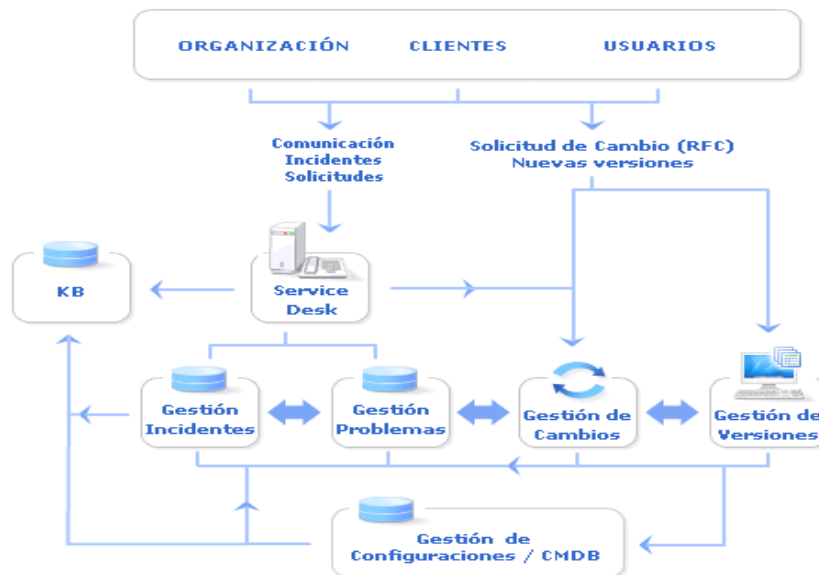


Figura Nº 2. Aspectos de la metodología de soporte al servicio según los estándares ITIL.

Fuente: Cartlidge, et al. 2007

2.2.3 Gestión de Cambios (*Change Management*):

Vivimos en una época de continuos cambios. Siempre tendemos a asociar la idea de cambio con la de progreso, y aunque esto no sea necesariamente así, es evidente que toda "evolución a mejor" requiere necesariamente de un cambio. Sin embargo, es frecuente encontrarse con gestores de servicios TI que aún se rigen por el lema: "si algo funciona, no se debe cambiar". Y aunque bien es cierto que el cambio puede ser fuente de nuevos problemas, y nunca debe hacerse sin evaluar sus consecuencias, puede resultar mucho más peligroso el estancamiento en servicios y tecnologías desactualizados.

Las principales razones para la realización de cambios en la infraestructura TI son:

- Solución de errores conocidos.
- Desarrollo de nuevos servicios.
- Mejora de los servicios existentes.
- Imperativo legal.

El principal objetivo de la Gestión de Cambios es la evaluación y planificación del proceso de cambio para asegurar que, si éste se lleva a cabo, se haga de la forma más eficiente, siguiendo los procedimientos establecidos y asegurando en todo momento la calidad y continuidad del servicio TI.

2.2.3.1 Conceptos Básicos

En el resto de este capítulo se utilizará con frecuencia varios conceptos relacionados con la Gestión de Cambios, por lo que resulta conveniente describir y diferenciar sus respectivas atribuciones:

- **Encargado de Cambios:** es el responsable del proceso del cambio y como tal debe ser el último responsable de todas las tareas asignadas a la Gestión de Cambios. En grandes organizaciones el Gestor de Cambios puede disponer de un equipo de asesores específicos para cada una de las diferentes áreas.
- **Desarrollador de versiones:** es el encargado de desplegar y llevar a cabo el cambio aprobado por el Encargado de Cambios. Aunque pertenece al Proceso de Gestión de Versiones, este guarda una relación con el proceso de Gestión de Cambios, ya que este puede recomendar el tipo de cambio a realizarse y debe indicar cuando el cambio haya sido implementado.

- **Elemento de Configuración (*Configuration Item* ó *CI*):** es un componente de TI que esta bajo el control de la Gestión de Configuración. Cada CI se puede componer de otros CI's. Los CI's pueden variar extensamente en complejidad, tamaño y tipo, desde un sistema entero (incluyendo todo el hardware, software y la documentación) hasta un sencillo modulo de software o un componente menor de hardware.
- **Iniciador del Cambio:** es el encargado de elaborar la petición de cambio, indicando que tipo de cambio desea, en que organización, los elementos de configuración implicados, etc.
- **Petición de Cambio (*Request for Change* ó *RFC*):** esto es una petición de cambio formal, incluyendo una descripción del cambio, componentes afectados, necesidades del negocio, costos estimados, niveles de riesgo, requerimientos de recursos y el estado de aprobación.

2.2.3.2 Proceso de Gestión de Cambios

A continuación se presenta los pasos del proceso de Gestión de Cambios.

- **Registro:** El primer paso del proceso de cambio es registrar adecuadamente las RFC's. El origen de una RFC puede ser de muy distinta índole, como elemento de salida del proceso Gestión de Problemas, implantación de nuevos servicios, estrategias empresariales, actualización de software a terceros, imperativos legales, etc.

No siempre un cambio implica una RFC. Para cambios de escasa importancia o que se repiten periódicamente pueden acordarse procedimientos estándar que no requiera la aprobación de la Gestión de Cambios en cada caso.

Independientemente de su origen el correcto registro inicial de una RFC requerirá, cuando menos, de los siguientes datos: fecha de recepción, identificador único de la RFC, identificador del error conocido asociado, descripción del cambio propuesto (motivación, propósito, CI's involucrados), estimación de recursos necesarios para la implementación, tiempo estimado, estatus.

Este registro deberá ser actualizado con toda la información generada durante el proceso para permitir un detallado seguimiento del mismo desde su aprobación hasta la evaluación final y cierre.

- **Aceptación:** Tras el registro del RFC se debe evaluar preliminarmente su pertinencia. Una RFC puede ser simplemente rechazada si se considera que el cambio no está justificado o se puede solicitar su modificación si se considera que algunos aspectos de la misma son susceptibles de mejora o mayor definición. En cualquiera de los casos la RFC debe ser devuelta al departamento o persona que la solicitó con el objetivo de que se puedan realizar nuevas alegaciones a favor de dicha RFC o para que pueda ser consecuentemente modificada. La aceptación del cambio no implica su posterior aprobación y es sólo indicación de que se ha encontrado justificado su ulterior procesamiento.

Clasificación: Tras su aceptación se deben asignar a la RFC una categoría dependiendo de la urgencia y el impacto de la misma. La determinación de la categoría se basa en el impacto sobre la organización y el esfuerzo requerido para su implementación.

Existen dos categorías de cambio:

- **Cambio estándar:** son cambios que no requieren de gran cantidad de tiempo ni de esfuerzo al implementarlo, por ejemplo, actualizar ciertos paquetes de software o la instalación de un nuevo dispositivo
- **Cambio urgente:** son cambios que requieren de una gran cantidad de tiempo y de esfuerzo y la implementación de estos cambios deben ser evaluados minuciosamente por el Gestor de Cambios.
- **Aprobación y Planificación:** La planificación es esencial para una buena gestión del cambio. Los sistemas de gestión de la información son muy susceptibles a los cambios de configuración por las sofisticadas interrelaciones entre todos los CI's involucrados. Un cambio estándar puede desencadenar una reacción en cadena con resultados catastróficos. Es imprescindible, como mínimo, disponer siempre de planes de "*back out*" que permitan la recuperación de la última configuración estable antes del cambio. Pero esto obviamente no es suficiente.

Una vez aprobado el cambio debe evaluarse si este ha de ser implementado aisladamente o dentro de un "paquete de cambios" que formalmente equivaldrían a un solo cambio.

- **Implementación:** Aunque la Gestión de Cambios no es la encargada de implementar el cambio, algo de lo que se encarga habitualmente la Gestión de Versiones, si lo es de supervisar y coordinar todo el proceso.

La figura N° 3 refleja los pasos del Proceso de Gestión de Cambios, propuestos por ITIL.

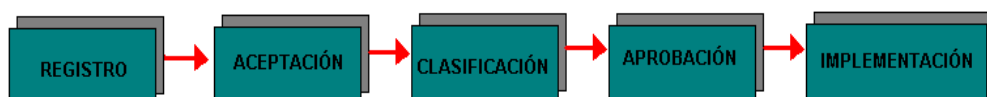


Figura N° 3. Pasos del Proceso de Gestión de Cambios

Fuente: Carlidge, et al. 2007

2.2.4 Proceso de Negocio

El concepto de proceso de negocio surge cuando una organización orienta sus actividades a satisfacer las necesidades de todos los agentes ligados a dicha organización (proveedores, empleados, clientes, terceros).

El objetivo de una organización que trabaja con un enfoque de proceso pretende optimizar las actividades ligadas al proceso en sí. Estas actividades son realizadas de forma transversal por los distintos departamentos de una organización.

Los Procesos de Negocios, según la *Workflow Management Coalition – WfMC* (Feb, 1999), son

“actividades o procedimientos que en conjunto cumplen un objetivo específico del negocio o metas de mas largo alcance, en el contexto de una estructura organizacional definiendo roles funcionales y relaciones” (wmfc.org)

En esta definición es importante recalcar lo de “objetivo de negocio”, porque permite diferenciarlos de los otros procesos. Hay procesos que no son de negocios porque no logran una meta de negocio sino una meta operativa.

Hoy en día es necesario que las empresas adapten continua y rápidamente sus procesos de negocio para mantenerse competitivas. La flexibilidad necesaria en las empresas se puede lograr mediante un conjunto de prácticas conocidas como administración de procesos de negocio (BPM). La figura N° 4 nos muestra la relación entre los Proceso de Negocio y los Sistemas de Información.

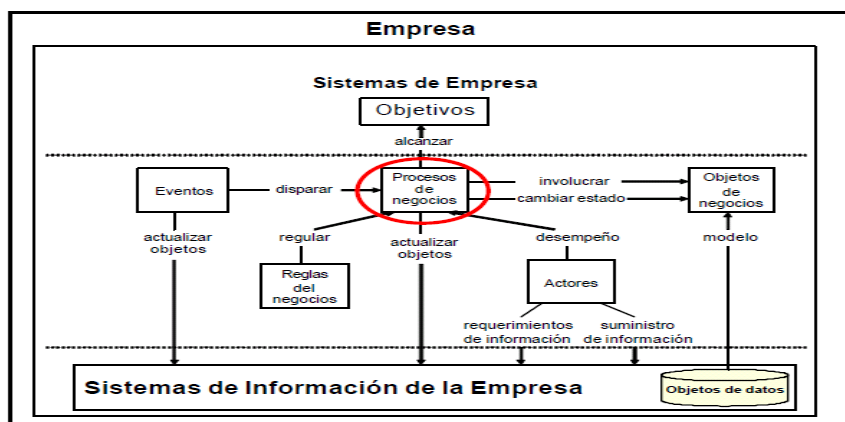


Figura Nº 4. Relación entre Procesos de Negocios y los Sistemas de Información

Fuente: Barrios, et al. 2004

2.2.5 Gestión de Procesos de Negocios (*BPM – Business Process Management*)

BPM se define como el soporte de procesos de negocios usando métodos, técnicas y software para diseñar, representar, controlar y analizar los procesos operacionales de que involucran organizaciones, aplicaciones, documentos y otras fuentes de información (Van der Aalst , 2004).

La Gestión de Proceso de Negocio es un enfoque empresarial operativo basado en la coordinación de las actividades y decisiones que todas las partes involucradas deben realizar durante un proceso de negocio con el objetivo de convertirse en una organización altamente eficiente, ágil, innovadora y adaptable.

Mediante BPM se persigue el modelado de las actividades de negocio para lograr una mejor administración, automatización y optimización.

A través del modelado de las actividades y procesos se logra un mejor entendimiento del negocio y muchas veces esto presenta la oportunidad de

mejorarlos. La automatización de los procesos reduce errores, asegurando que los mismos se comporten siempre de la misma manera y dando elementos que permitan visualizar el estado de los mismos. La administración de los procesos nos permite asegurarnos de que los mismos estén ejecutándose eficientemente y obtener información que luego puede ser usada para mejorarlos. Es a través de la información que se obtiene de la ejecución diaria de los procesos que se puede identificar posibles ineficiencias en los mismos y de esta forma optimizarlos. La figura N° 5 indica las diferentes actividades que permite el uso del paradigma BPM



Figura N° 5. Actividades que soporta BPM

Fuente: www.gartner.com, 2009

Para lograr estos objetivos y adherirse al paradigma BPM se han desarrollado sistemas o suites que automatizan la administración de procesos de negocio proporcionando herramientas para modelar, integrar, medir y optimizar

procesos de negocio. A estas herramientas se le denominan Sistemas de Gestión de Procesos de Negocio (*BPMS – Business Process Management Systems*).

2.2.6 Sistemas de Gestión de Procesos de Negocio (*BPMS – Business Process Management Systems*)

Un BPMS es una colección integrada de tecnologías de software que permiten control, manejo y mejoramiento continuo de los procesos a través de la automatización de su ciclo de vida.

Van de Putte (2001) establece que los objetivos de los BPMS son:

- Implementar cambios en las reglas y objetivos del negocio.
- Medir la efectividad de esos cambios.
- Separar el qué y cómo, independencia de administración de recursos y procesos; y
- Definir, cambiar e implementar los procesos de negocios de manera consistente.

Los principales módulos que componen una plataforma BPMS son los siguientes:

- **Modelador Gráfico de Procesos:** (*Business Modeler*) permite modelar los procesos de negocio, simular su ejecución, definir métricas para el monitoreo, y exportar a Lenguaje de Ejecución de Procesos de Negocio (*Business Process Execution Language – BPEL*).
- **Ambiente Integración y Desarrollo:** (*Integration Developer*) es la herramienta que permite implementar los procesos, y servicios. Esta herramienta permite integrar las pantallas (para interacción de un participante), y los servicios (interacción con sistemas legados).

- **Servidor de Procesos de Negocio:** (*Process Server*) es el motor que permite ejecutar los procesos de negocio, aquí se ejecutan las Aplicaciones Compuestas (flujos BPM), los *Workflows* tradicionales, y la Orquestación de Servicios (procesos compuestos solo por servicios). Este servidor también es el encargado de generar los datos de las métricas, y de monitoreo. Permite intervenir los procesos en tiempo real: balancear carga, cambiar flujo de negocio, y realizar acciones correctivas (según reglas de negocio).
- **Monitor de Actividades de Negocio:** (*BAM, Business Activity Monitor*) esta es una aplicación de administración que permite gestionar los procesos y servicios, gráficamente se pueden ver indicadores de performance, y Acuerdos de Nivel de Servicios (*SLA -Service Level Agreements*). Se puede además definir alertas y triggers de acuerdo a eventos de negocio que sucedan en el proceso. También puede proveer datos reales a los modelos (*Business Modeler*) para ajustar las simulaciones (y lograr mejoramiento continuo)

La figura N° 6 muestra los módulos que componen una herramienta BPMS:

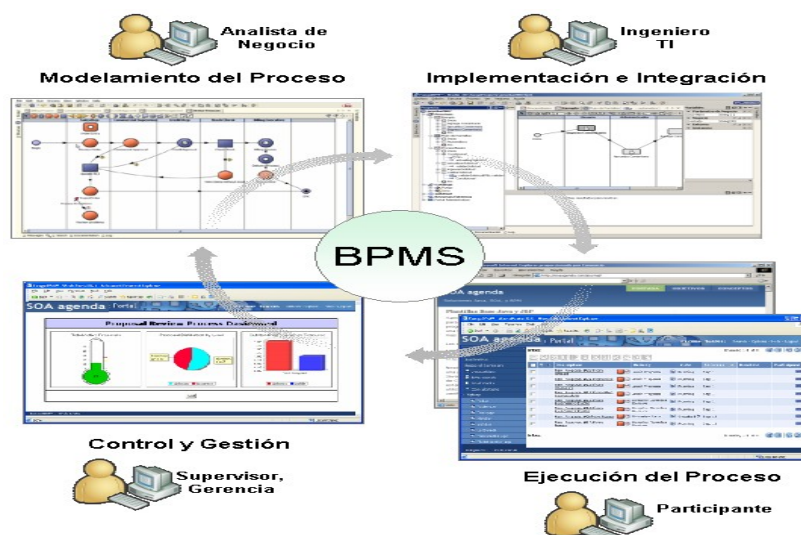


Figura N° 6. Módulos que componen una solución BPMS.

Fuente: Felipe. 2008

Actualmente existen diversas alternativas tecnológicas disponibles en el mercado en el área de GPN/BPM; basadas en Software Libre y *Open Source*; productos que implementan las disciplinas de Gestión de procesos de negocios (GPN) / Business Process Management (BPM).

En el anexo F se muestra la evaluación de herramientas GPN/BPM y donde fue seleccionada **Intalio BPMS 5.2.1** en su edición comunitaria (*Community Edition*), como herramienta para modelar el Proceso de Gestión de Cambios planteado en este Trabajo Especial de Grado.

En líneas generales, Intalio ofrece un modelo ágil para el despliegue de procesos de negocio, donde podemos integrar dos tipos de actividades: automáticas y humanas.

2.3 Servicios Web (*Web Services*)

A continuación se muestra toda la documentación sobre servicios web:

2.3.1 Definición de Servicio Web

Una definición genérica de Servicio Web es: “Una aplicación accesible a otras aplicaciones a través del Web”. (Caituro, et al. 2004)

El UDDI consortium (www.uddi.org) introduce la siguiente definición: “Aplicación de negocios modular autocontenida que está abierta, orientada a Internet y con una interfaz basada en estándares”.

La referencia mundial en el mundo Web (W3C) lo define como “un sistema de software identificado por una URI, cuyas interfaces públicas y enlaces se definen y describen usando XML. Su definición puede ser descubierta por otros

sistemas de software. Estos sistemas pueden interactuar con el servicio Web de la forma prescrita por su definición, usando mensajes basados en XML a través de protocolos estándares de Internet”.

Y por último, una definición precisa disponible la brinda Chatain:

“una manera estandarizada de integrar usos basados en el Web usando XML, SOAP, WSDL y UDDI abierta a estándares usando como espina dorsal el Protocolo de Internet, XML se utiliza para marcar los datos con etiqueta, SOAP se utiliza para transferir la data, WSDL se utiliza para describir los servicios disponibles, y UDDI se utiliza para el listado de qué servicios están disponibles”
(Chatain. et, al 2009)

2.3.2 Arquitecturas orientadas a Servicios (*Service-oriented architecture – SOA*)

Los servicios Web han impulsado las Arquitecturas Orientadas a Servicios (SOA). SOA es una forma de arquitectura para sistemas distribuidos caracterizada por las siguientes propiedades:

- **Vista lógica:** el servicio es una abstracción (vista lógica) de los programas, bases de datos, procesos de negocio, etc., definido en términos de lo que hace (llevando a cabo una operación de negocio).
- **Orientación a mensajes:** el servicio se define formalmente en términos de los mensajes intercambiados entre agentes proveedores y solicitantes, y no está basado en las propiedades de los agentes. La estructura interna del agente (lenguaje de programación, BD, proceso, etc.) se abstrae en SOA. Esto permite incorporar cualquier componente o aplicación a esta arquitectura “decorando” estos componentes con software de gestión y conversión.

- **Orientación a la Descripción:** un servicio se describe con metadatos procesables. La descripción da soporte a la naturaleza pública de SOA: sólo se incluyen en la descripción aquellos detalles que se exponen públicamente y son importantes para el uso del servicio. La semántica de un servicio debe documentarse, directa o indirectamente, por su descripción.
- **Granularidad:** los servicios tienden a usar un pequeño número de operaciones con mensajes relativamente complejos.
- **Orientación a la Red:** los servicios tienden a usarse a través de la red, aunque este no es un requisito absoluto.
- **Neutral a la Plataforma:** los mensajes se envían en un formato estándar y neutral a la plataforma, distribuido a través de las interfaces (XML).

En general, SOA y Servicios Web son apropiados para aplicaciones:

- Que deben operar a través de Internet, donde la fiabilidad y la velocidad no se puede garantizar;
- Donde no existe habilidad de gestionar la instalación de forma que todos los solicitantes (clientes) y proveedores se actualicen a la vez;
- Donde los componentes de un sistema distribuido se ejecuten en distintas plataformas y distintos productos;
- Donde una aplicación existente necesite exponerse para ser usada a través de la red y pueda “decorarse” como un servicio Web. (Acevedo, 2007)

En términos de modelos de interacción abstractos y tal como se aprecia en la Figura N° 7, SOA requiere de las siguientes interacciones

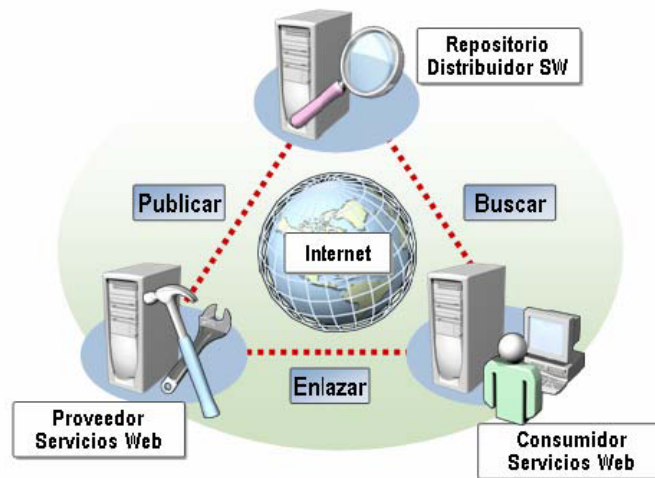


Figura Nº 7. Arquitectura Orientada a Servicios.

Fuente: Acevedo. 2008

- Un servicio Web publica su definición (WSDL) en un repositorio / directorio / registro distribuido (UDDI).
- El consumidor del servicio Web busca la definición del servicio en el repositorio.
- El repositorio usa la información de la definición (WSDL) para enlazar con el servicio y enviar peticiones (SOAP) al servicio que ofrece el proveedor de servicios.

2.3.2.1 Estándares y tecnologías subyacentes. Infraestructura básica

La infraestructura mínima que requieren los Servicios Web se puede definir en términos de:

- Lo que va en “la red”: Formatos y protocolos de comunicación.
- Lo que describe lo que va en la red: Lenguajes de Descripción de Servicios
- Lo que nos permite encontrar y almacenar dichas descripciones: *Descubrimiento de Servicios*.

2.3.3 Arquitectura Tecnológica

Los servicios web implementan una arquitectura tecnológica donde las especificaciones superiores hacen uso de las inferiores (ver figura N° 8). Estas tecnologías son:

- **HTTP:** (*Hypertext Transfer Protocol*): es un protocolo estándar de W3C para la transferencia de documentos en Internet. Los Servicios Web lo utilizan como mecanismo de comunicación. Es un protocolo genérico y sin estado.
- **XML:** (*eXtensible Markup Language*) lenguaje que deriva de SGML, diseñado para representar y transferir datos estructurados, separa datos de formateo y transformación.
- **SOAP:** (*Simple Object Access Protocol*), ofrece los mecanismos de comunicación básicos para el envío de mensajes en formato XML, permitiendo la invocación remota de servicios. Normalmente funciona sobre HTTP, pero no siempre. SOAP es el *sine qua non* de los servicios Web.
- **WSDL:** (*Web Services Description Language*), es un formato basado en XML (desarrollado por IBM y Microsoft) para describir de manera formal servicios Web.
- **UDDI:** (*Universal Description, Discovery, and Integration*), es un directorio que contiene un registro/repositorio de descripciones de servicios Web.



Figura N° 8. Pila de Servicios Web.

Fuente: Acevedo, 2008

2.3.4 SOAP (*Simple Object Access Protocol*)

SOAP define un protocolo que da soporte a la interacción (datos + funcionalidad) entre aplicaciones en entornos distribuidos y heterogéneos, es interoperable (neutral a plataforma y lenguajes de programación, independiente del HW y protocolos). Funciona sobre la infraestructura (estándares) existente en Internet. SOAP define cómo organizar información usando XML de forma estructurada y tipada para intercambiarla entre distintos sistemas. (www.w3c.org, 2003)

2.3.4.1 Especificaciones de SOAP

SOAP especifica:

- Un formato de mensaje para una comunicación unidireccional, describiendo cómo se empaqueta la información en documentos XML.
- Un conjunto de convenciones para usar mensajes SOAP para implementar el patrón de interacción RPC, definiendo cómo los clientes pueden invocar un *Procedimiento Remoto* enviando un mensaje SOAP y cómo los servicios pueden responder enviando otro mensaje al llamador.
- Un conjunto de reglas que una entidad que procesa mensajes SOAP debe seguir, definiendo en particular los elementos XML que una entidad debe leer y entender, así como las acciones que deben tomar si no entienden el contenido. Reglas de Codificación de los Datos.
- Una descripción de cómo se debe transportar un mensaje SOAP sobre HTTP y SMTP. Se definirán “Bindings” a otros protocolos de transporte en futuras versiones de la especificación.

2.3.4.2 Formato del mensaje

SOAP intercambia información mediante mensajes. Los mensajes se utilizan como envoltorios que la aplicación utiliza para guardar la información que quiere enviar. Cada envoltorio contiene dos partes: Una cabecera (opcional) y un cuerpo (obligatorio). La cabecera y el cuerpo pueden tener múltiples subpartes en forma de bloques de la cabecera y bloques del cuerpo (ver Figuras Nº 9 y 10).

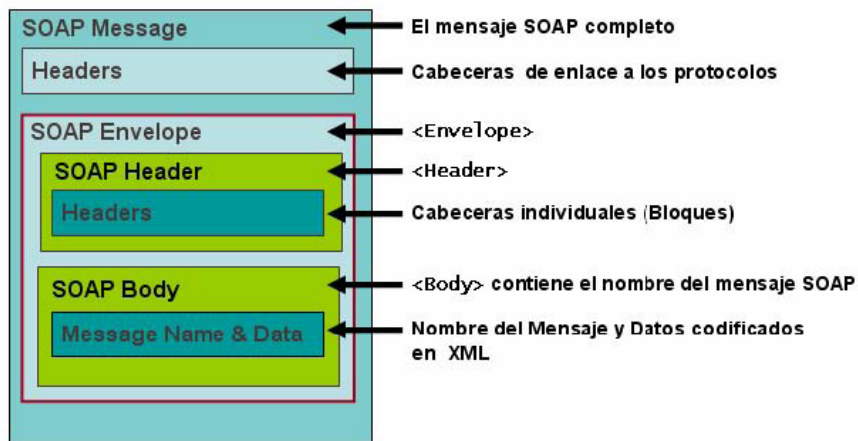


Figura Nº 9. Formato de un mensaje SOAP

Fuente: Peltz. 2003

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <WoodgroveAuthInfo xmlns="http://tempuri.org/">
      <Username>string</Username>
      <Password>string</Password>
    </WoodgroveAuthInfo>
  </soap:Header>
  <soap:Body>
    <GetAccount xmlns="http://tempuri.org/">
      <acctID>int</acctID>
    </GetAccount>
  </soap:Body>
</soap:Envelope>
```

Figura Nº 10. Mensaje SOAP

Fuente: Peltz. 2003

2.3.5 WSDL (*Web Services Description Language*)

WSDL fue creado originalmente por IBM, Microsoft y Ariba. Un archivo WSDL es un documento XML que describe los servicios Web, en particular sus interfaces. WSDL debe definir los mecanismos de acceso (protocolos) a los servicios Web. En WSDL la separación de interfaces; enlaces de protocolos y la necesidad de incluir información de localización permite la definición de especificaciones modulares. WSDL permite definir interfaces más complejas y expresivas permitiendo definiciones de interacciones asíncronas y diferentes paradigmas de interacción, y la posibilidad de combinar o agrupar operaciones.

2.3.5.1 Estructura de un Documento WSDL

El documento WSDL de un servicio proporciona dos piezas de información básicas: una parte o interfaz *abstracta* (independiente de la aplicación) y una parte *concreta* que define los enlaces a protocolos e información de los puntos finales de acceso al servicio, como lo podemos presenciar en la figura N° 11.

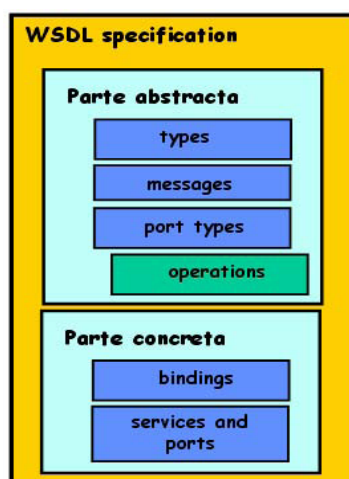


Figura N° 11. Estructura de un documento WSDL

Fuente: Acevedo. 2007

La parte abstracta está compuesta por:

- Definiciones de *Port types*: cada *port type* es una colección lógica de *operations*.
- Cada *operation* define un intercambio simple de *messages*. Un *message* es una unidad de comunicación representando un intercambio de datos en una única transmisión lógica.
- Un sistema de tipos (*types*) usados por las *operations* (por defecto XML schema).

La parte concreta está compuesta por:

- Definiciones de *Bindings*: se especifica la codificación de los mensajes, y los enlaces a protocolos de todas las operaciones y mensajes definida en un *port type*.
- Definiciones de *Ports*: se especifica en qué dirección (URI) se puede acceder la implementación del *port type*. Definen un punto final (lugar de la red) donde está el servicio.
- Definiciones de *Services*: definen una agrupación de *Ports*.

De esta forma WSDL se utiliza para describir un Servicio Web en términos de los mensajes que acepta y genera, actúa como contrato entre un consumidor (cliente) y dicho Servicio. WSDL puede describir puntos finales y sus operaciones sin especificar el formato de los mensajes o los protocolos de red (SOAP, HTTP-GET/POST y MIME) a los cuales el punto final está ligado. Ver figura N° 12.

```

<?xml version="1.0"?>
<definitions name="StockQuote"
  <types>
    <!-- XSD for the messages -->
  </types>
  <message name="GetLastTradePriceInput">
    <!-- part(s) identifying the messages within the schema -->
  </message>
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <!-- grouping messages into logical operations -->
    </operation>
  </portType>
  <binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
    <!-- specifying how operations map to protocols such as SOAP -->
  </binding>
  <service name="StockQuoteService">
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
      <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>

```

Figura Nº 12. Ejemplo de archivo WSDL

Fuente: Acevedo. 2007

En el Anexo H se puede visualizar la metadata de los servicios web utilizados al modelar el proceso de Gestión de Cambios con Intalio.

2.3.6 UDDI (*Universal Description, Discovery, and Integration*)

UDDI, fue creado originalmente por IBM, Microsoft y Aribay en el 2002 paso a manos de OASIS que ha determinado su futuro y extensiones. UDDI se concibió como un Registro de Negocio = Servicio de Directorio y Nombrado sofisticado. (www.oasis-open.org)

Especifica un marco para describir y descubrir Servicios Web. UDDI define estructuras de datos y APIs para publicar descripciones de servicios en el registro y para consultar el registro en búsqueda de descripciones publicadas. Las APIs de UDDI están especificadas con WSDL y con SOAP Binding, lo que permite acceder a ellas como Servicios Web. La especificación UDDI tiene dos objetivos esenciales:

- ser un soporte a los desarrolladores para encontrar información sobre servicios Web y poder construir clientes.

- facilitar el enlace dinámico de Servicios Web, permitiendo consultar referencias y acceder a servicios de interés.

2.3.6.1 Modelo de Información. Estructura de Datos de UDDI

La información en un registro UDDI se almacena en ficheros XML con una estructura jerárquica, como lo muestra la figura N° 13. Los elementos de esta estructura son:

- **businessEntity:** es el elemento “top-level”, describe un negocio o una entidad que ha registrado un servicio en UDDI. Ejemplos: Departamento de Contabilidad, o Servidor de Aplicaciones Corporativo. Este elemento soporta información estándar tal como nombre, descripción, e información de contacto, así como también información de metadatos (por ejemplo: identificadores y categorías).
- **businessService:** describe un Servicio Web que ha sido expuesto por una entidad de negocio; soporta el nombrado de un Servicio Web y lo asocia con una entidad de negocio y con la información de *binding*. Soporta la asignación de categorías al Servicio Web (industria, productos, códigos geográficos, etc.).
- **bindingTemplate:** describe la información técnica necesaria para enlazar con un Servicio Web en particular. Este elemento soporta el nombrado de un Servicio Web y su asociación con una entidad de negocio e información de binding. La información de binding se describe como un punto de acceso que posee un atributo llamado *urlType* utilizado para especificar los siete tipos de puntos de entrada: *mailto*, *http*, *Https*, *Ftp*, *Fax*, *Phone*, *Other*.
- **tModel (Technology Model):** estructura de metadatos genérica para representar cualquier concepto o construcción (definiciones de protocolos,

ficheros WSDL, XML schemas, Espacios de Nombres, esquemas de categorías, etc.).

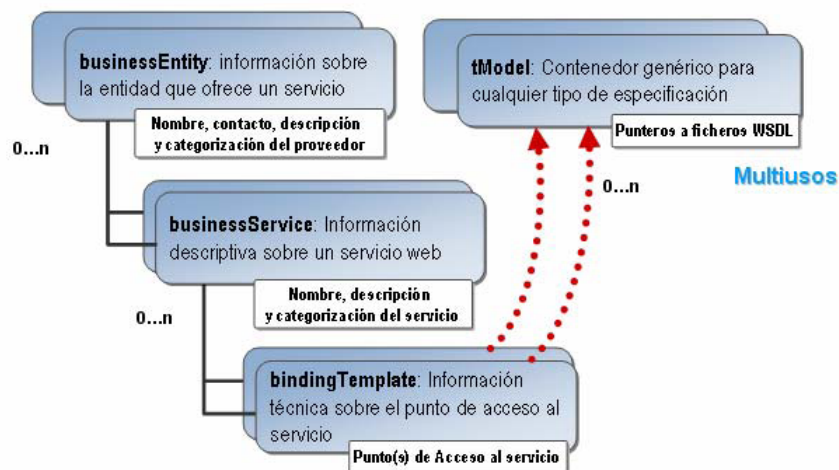


Figura Nº 13. Estructura de Datos UDDI

Medjahed. et al, 2003

2.4 Orquestación de Servicios Web

Las organizaciones que ya están ocupando integración de aplicaciones de empresas, middleware para automatizar procesos de negocios o para integrar ambientes de desarrollo, ya estarán familiarizadas con el concepto de orquestación. Poseen un controlador central que maneja un conjunto de lógicas que facilitan la interoperabilidad entre las distintas aplicaciones. Una implementación común de orquestación es el modelo hub-and-spoke que permite que múltiples participantes externos interactúen con un motor central de orquestación. (Booth, et al. 2004)

Medjahed indica que:

“Uno de los requerimientos manejados para la creación de estas soluciones fue la acomodación de la mezcla de procesos de negocios grandes. Con orquestación, diferentes procesos pueden ser conectados sin tener que volver a desarrollar la solución que originalmente fue automatizada en cada uno de los procesos, por esto, orquestación puede reducir significativamente la complejidad de los desarrollos de soluciones. Ya que orquestación lleva a introducir una nueva lógica de flujos de trabajos, que es abstraída y más fácil de mantener que cuando son soluciones embebidas con componentes individuales”. (Medjahed. et al, 2003)

El rol de la orquestación abarca el desarrollo orientado a servicios. A través del uso de las extensiones que permiten a la lógica de los procesos de negocios ser expresadas a través de servicios, la orquestación puede representar y expresar la lógica del negocio en un estándar. Cuando se construye una solución orientada a servicios, esta provee un significativo atractivo en el control y encubrimiento de la representación de la lógica del proceso a automatizar.

Orquestación más adelante tendrá la capacidad intrínseca de la interoperabilidad buscada para el diseño de servicios y para proveer la potencial integración entre procesos. Un aspecto clave de cómo orquestación está posicionada con SOA es el hecho que orquestación por sí sola existe como servicio. Por lo tanto, construido sobre lógica, la orquestación estandariza la representación del proceso a través de una organización, mientras abordar el objetivo de la supervisión de los servicios de la empresa y promueve la orientación a los servicios. (Caituro, et al. 2004)

Una primera especificación de la industria para la estandarización de la orquestación es *Web Services Business Process Execution Language (WS-BPEL)* que también es conocida como BPEL4WS o simplemente como BPEL.

2.4.1 Protocolos de Negocios y Definición de Procesos

La lógica del flujo de trabajo que comprende la orquestación puede consistir en numerosas reglas de negocios, condiciones, y eventos. Colectivamente, estas partes de una orquestación establecen un protocolo de negocio que define cómo los participantes pueden llegar a completar los procesos de negocio. El detalle de la lógica del flujo de trabajo está encapsulado y expresado por una orquestación que está contenida con una definición del proceso.

2.4.2 Servicio de Procesos y Socio de Servicios

Identificado y descrito con una definición de proceso están los participantes permisibles del proceso. Primero, el proceso en sí es representado como un servicio, resultando en un servicio de procesos, tal como lo demuestra la figura N° 14.

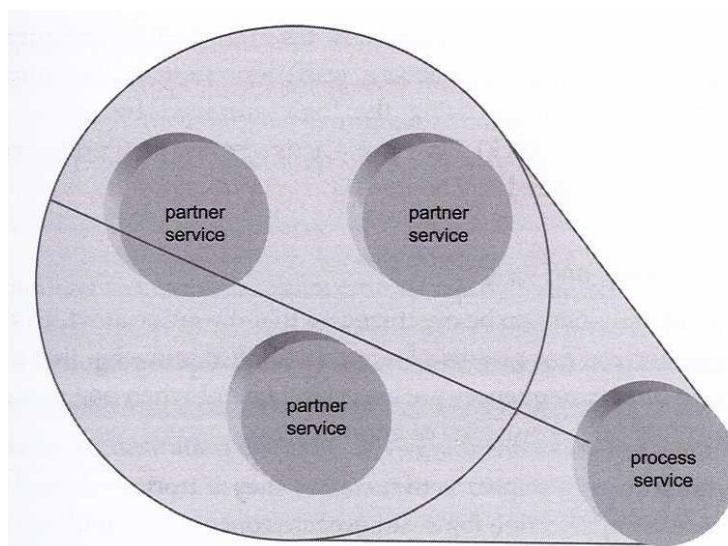


Figura N° 14. Servicio de Procesos

Fuente: Wilkes. 2006

Otro servicio permitido para interactuar con el servicio de proceso está identificado como socio de servicios o *partner links*. Dependiendo sobre la lógica el flujo de trabajo, el servicio de proceso puede ser invocado por un socio de servicio externo, o puede ser invocado por otro socio de servicio

2.4.3 Actividades Básicas y Actividades Estructuradas

BPEL4WS analiza la lógica del flujo de trabajo en una serie de actividades primitivas predefinidas. Las Actividades Básicas (*receive, invoke, reply, throw, wait*) representan fundamentalmente acciones en el flujo de trabajo las cuales puede ser ensamblada usando la lógica provista por las actividades estructuradas (*sequence, switch, while, flow, pick*).

2.4.4 Secuencias, Flujos y Acoplamientos

Actividades básicas y estructuradas pueden ser organizadas de modo que el orden en el cual ellas se ejecuten sea predefinido. Una secuencia junta grupos de actividades relacionadas en una lista que determina un orden de ejecución de la secuencia. Las secuencias son especialmente usadas cuando una pieza de la lógica de la aplicación es dependiente de las salida de otra.

Flujos también contiene grupos de actividades relacionadas, pero ellas introducen diferentes requerimientos en la ejecución. Pedazos de lógica de la aplicación puede ser ejecutada concurrentemente con un flujo, significa que éste no es necesariamente un requerimiento de un grupo de actividades que espera a que otras terminen. Sin embargo, el flujo en sí no termina hasta que todas las actividades encapsuladas en él hayan sido procesadas por completo. Esto asegura una forma de sincronización entre la lógica de la aplicación residente en los flujos individuales.

Acoplamiento es usada para establecer dependencias formales entre actividades que son partes de un flujo. Antes de que una actividad esté completamente terminada, se debe estar seguro que cualquier requerimiento establecido desde un acoplamiento externo ya fue respondido. Similarmente, antes de que cualquier actividad acoplada pueda empezar, es necesario que sea satisfecho cualquier otro requerimiento de acoplamiento hecho con anterioridad. Reglas provistas por el acoplamiento son siempre referidas como dependencias de sincronización.

En el anexo E se puede visualizar los detalles del flujo de tareas del proceso de Gestión de Cambios modelado a través de Intalio.

2.4.5 Orquestación y Actividades

Una actividad es un término genérico que puede ser aplicado a cualquier unidad de lógica de un trabajo completado por una solución orientada al servicio. Este es el alcance de una orquestación, por lo tanto, puede ser clasificado como un complejo, y muy probablemente, actividades duraderas.

2.4.6 Orquestación y Coordinación

Orquestación, es representado por *BPEL4WS*, puede utilizar completamente el manejador de contexto del cuadro de *WS-Coordination* que está incorporado en el coordinador de tipos de *WSBusinessActivity*. Esta especificación define el diseño de protocolos de coordinación para el complejo soporte, actividades duraderas.

EL proceso de Gestión de cambios modelado en Intalio, genera automáticamente código BPEL. La estructura de este lo podemos visualizar en el Anexo I.

2.4.7 Orquestación y SOA

La lógica del proceso de negocio es la raíz de la automatización de las soluciones. Orquestación provee un modelo automatizado donde la lógica de procesos es centralizada con todo aun extensible y componible. A través de uso de la orquestación, el desarrollo de la soluciones orientadas al servicio llegando a ser intrínsecamente extensibles y adaptables. Orquestación típicamente establece un punto en común de integración para otras aplicaciones, las cuales hacen una orquestación implementando una llave de integración.

Estas cualidades incrementan la agilidad organizacional porque:

- La lógica del flujo de trabajo encapsulado por una orquestación puede ser modificado o extendido en un lugar centralizado.
- Posicionando una orquestación centralizada puede facilitar significativamente la mezcla de procesos de negocios por abstracción del unificador que une la correspondiente solución automatizada.
- Por establecimiento potencialmente de interacción de arquitecturas de gran escala orientada al servicio. Orquestación, sobre un nivel fundamental, puede soportar la evolución diversificada de la empresa.

Orquestación es el ingrediente clave para la realización de una organización que contiene varias aplicaciones basadas en plataformas computacionales dispares. Avances en el middleware permiten que los motores de orquestación se

vuelvan completamente integrados al desarrollo orientado al servicio. Para muchos desarrollos, orquestación se ha convertido en el corazón de SOA. (Sleeper, 2004)

2.5 SOAr

2.5.1 Definición y ventajas de SOAr

SOAr (estrategia creada por www.gurulab.org) es una forma de integrar aplicaciones y procesos. Esta basado en una Arquitectura Orientada a Servicios (SOA), la cual tiene como objetivo principal aislar toda aplicación de la plataforma de tecnología o proceso del negocio, del resto de la arquitectura; SOAr logra esto simplificando de manera importante cualquier esfuerzo de integración empresarial, logrando:

- Desacoplar la plataforma de TI de los procesos del negocio.
- Desacoplar las aplicaciones unas de otras, escondiendo la complejidad del conjunto de aplicaciones existente a cualquier nueva aplicación que se desea integrar.
- Controlar reduciendo los riesgos derivados de los cambios de aplicaciones de la plataforma de sistemas.
- Aprovechar al máximo las capacidades de las herramientas tecnológicas existentes actualmente.

La estrategia de integración de aplicaciones SOAr provee:

- **Simplicidad:** esconde la complejidad resultante de la redundancia y solapamientos funcionales de las aplicaciones existentes.
- **Estabilidad:** para todo cliente de las aplicaciones expuestas por SOAr, estas últimas nunca cambian; esto es debido a que aún cuando en realidad las aplicaciones en realidad si cambian, SOAr lo hace transparente a los clientes de manera que estos últimos no se dan cuenta.
- **Eficiencia,** aprovecha al máximo las características de las plataformas tecnológicas actuales al definir claramente cómo utilizar cada componente de la plataforma de la empresa, independientemente de quien sea el proveedor de cada uno. (www.gurulab.org, 2009)

Según Cesar Obach-Renner, del grupo gurulab.org: “SOAr es la evolución de SOA en cuanto que al redefinir la forma de implementación, cubre los vacíos que SOA por si sólo deja sin solución. Estos son:

- **Evolución de la especificación de los contratos de los proveedores sin afectar a los clientes:** el estado ideal del SOA es contar con una especificación estándar de contratos y con un ecosistema de aplicativos que cumplan con dicha especificación estándar; de esta manera cuando uno cambie un proveedor por otro los clientes no se ven afectados.
- **Duplicidad funcional:** cuando un servicio es provisto por dos aplicaciones diferentes, diferenciados por criterios particulares tales como “tipos de cliente” (un tipo de cliente procesado por el proveedor A y para el otro tipo de cliente por el proveedor B), calidad de servicio, ubicación geográfica del cliente o del material.
- **Cohesión:** cuando la dinámica de las empresas exige hoy en día un continuo cambio de sus aplicativos para ir evolucionando en la dirección de su plataforma ideal, con cada” (Obach-Renner, 2008)

2.5.2 SOAr y BPM

SOAr desacopla los procesos del negocio de la plataforma de tecnologías de información; esto exponiendo un conjunto de servicios corporativos que están disponibles tanto a las herramientas de modelaje y análisis de procesos, como al motor de ejecución.

De la misma manera como SOAr “protege” las integraciones con aplicaciones cuando una de ellas es cambiada, protege los procesos de los cambios de aplicaciones. (Obach-Renner, 2008).

Dado que los procesos “consumen” servicios corporativos, y éstos no cambian sino las aplicaciones que SOAr encapsula, ningún cambio de aplicaciones de la plataforma de TI afecta la ejecución de procesos implementada sobre la capa de BPM, la figura N° 15 nos demuestra lo anteriormente planteado.

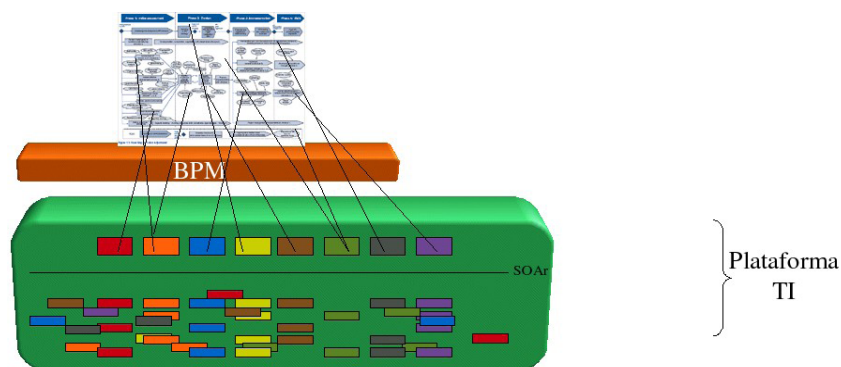


Figura N° 15. Enlace de la capa BPM con SOAr

Fuente: Obach-Renner. 2008

2.6 Intalio BPMS 5.2.1

Intalio es una solución BPMS Open Source basado en Java-J2EE y esta basado en un conjunto de estándares de la industria (BPMN, BPEL y BPEL4People) para el desarrollo de procesos de negocios.

Una de las grandes características de Intalio es la capacidad de diseñar, desplegar y optimizar procesos de negocio, con la promesa de es hacer estos sin líneas de códigos.

Desde el punto de vista tecnológico, Intalio|BPM provee la tecnología para crear una capa que provee servicios (Web Services), que modelen los procesos de negocio de la organización y todas sus reglas de negocio.

Para el modelado del Proceso de Gestión de Cambios planteado en este Trabajo Especial de Grado, se utilizará Intalio BPMS 5.2.1 en su edición comunitaria (*Community Edition*), como se nombre anteriormente. Para conocer la instalación y la configuración de esta herramienta, podemos dirigirnos al anexo B.

Intalio|BPM Community Edition ofrece los siguientes beneficios:

- Soporta estándares usados ampliamente en muchas organizaciones como la Notación de Modelamiento de Procesos de Negocio (*BPMN - Business Process Modeling Notation*) y el Lenguaje de Ejecución de Procesos de Negocio (*BPEL - Business Process Execution Language*).
- Al implementar código en BPEL un generador de código lo traduce automáticamente a un conjunto de clases de Java que se despliegan en una máquina virtual de Java o en un servidor de aplicaciones J2EE.
- BPMN y BPEL hacen una combinación poderosa de gran alcance para permitir utilizar el modelador gráfico sin tener que escribir códigos.

Intalio|BPM esta basado en un conjunto de plugins, lo que permite contar con un diseñador sobre un ambiente extensible de utilitarios y componentes desarrollados por terceros. Por ejemplo el rule engine llamado Corticon, Celequest PKI, y Orbeon para XForms.

Intalio tiene una participación activa en los siguientes proyectos: Apache Geronimo, Base de datos de MySQL, Orbeon para XForms, Corticon para Reglas de Negocio, etc.

Intalio|BPM Community Edition esta compuesto de dos módulos:

- Una herramienta para el diseño de los procesos de negocio denominada Intalio|Designer, basada en Eclipse.
- Un engine que ejecuta los artefactos de software generados por el diseñador de procesos denominado Intalio|Server.

La figura N° 16 nos muestra los componentes que utiliza Intalio Community.

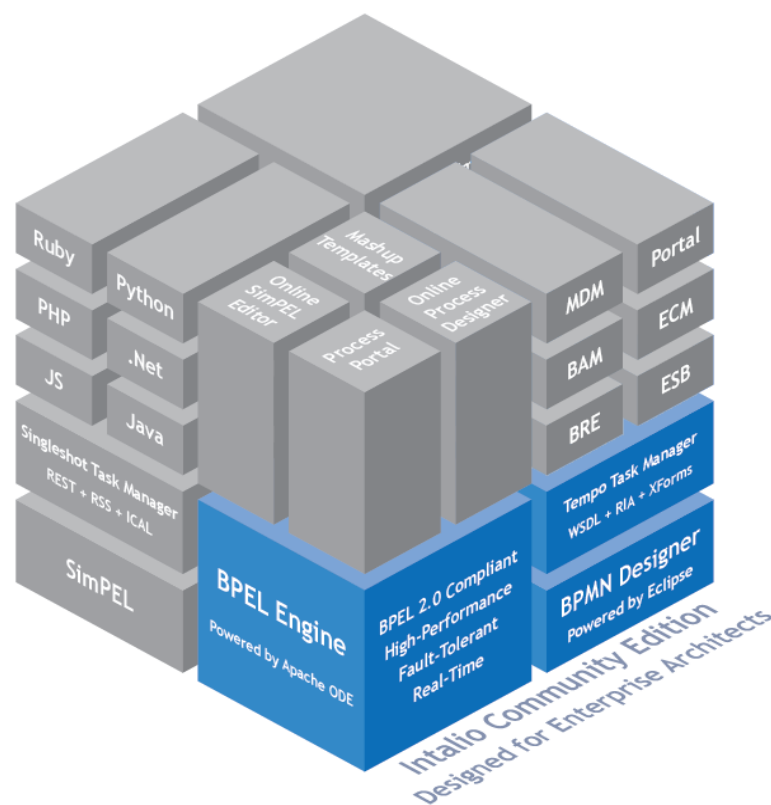


Figura N° 16. Componentes de Intalio|BPM Community Edition

Fuente: Felipe, 2008

2.6.1 Intalio|Designer

Es una herramienta para el diseño de procesos utilizando la notación BPMN. Esta soportado por el proyecto Eclipse Europa, y puede ser instalado en ambientes Windows, Linux y Mac OS X.

El modelador BPMN de Intalio|Designer permite diseñar cualquier proceso usando las especificaciones de BPMN 1.1 o de BPMN 2.0, después genera automáticamente código BPEL 2.0. El Modelador de BPMN soporta BPMN 2.0, ampliada para construir flujo de trabajo humano con BPEL4People. La figura N° 17 nos muestra una perspectiva general del diseñador de Intalio.

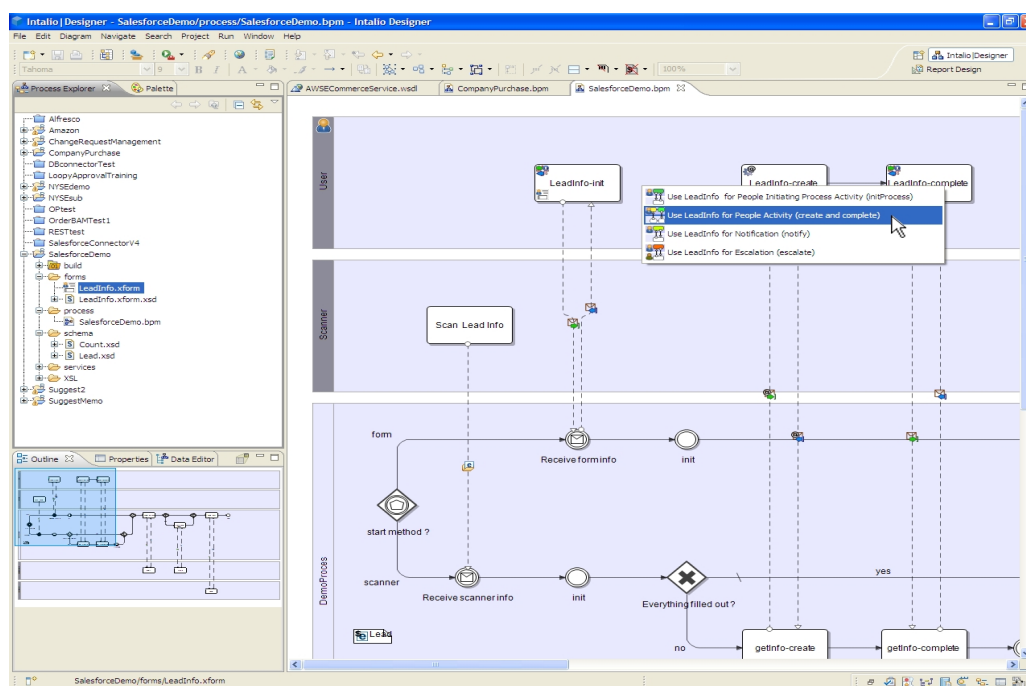


Figura N° 17. Diseñador Intalio|Designer

Fuente: www.intalio.com, 2009

Intalio|Designer incluye un diseñador de formularios para el desarrollo de formas usadas en los pasos de workflows humanos. Pueden ser diseñados en

estilo drag-and-drop usando una extensa colección de controles como campos de textos, cajas de chequeos y de combos. Los formularios están basados en el estándar XForms.

El diseñador también incluye un Mapeador de Datos que permite que muchas actividades diseñadas en un proceso requieren pueden ser enlazadas a sistemas externos, o a datos generados al ejecutarse procesos de un workflow. Este mapeador produce lenguaje XPath y/o código XSLT desde la asignación gráfica de mapas. Esto lo podemos visualizar en la figura N° 18.

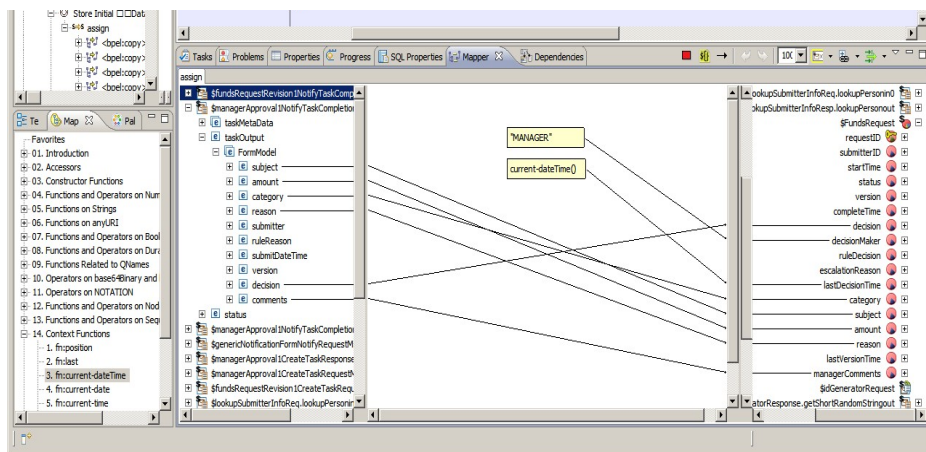


Figura N° 18. Mapeador de Datos de Intalio|Designer

Fuente: www.intalio.com, 2009

Las características en general de Intalio|Designer la podemos observar en la tabla N° 1:

<p>Importación y Generación de Código</p>	<ul style="list-style-type: none"> • Validación transparente de procesos • Importación y Generación de Código WS-BPEL 2.0 • Generación automática de código de procesos • Importación de código BPEL4WS 1.0/1.1
<p>Ambiente</p>	<ul style="list-style-type: none"> • Versión Standalone • Soporta BPMN 1.1 • Entorno de aplicaciones integradas (EAI) • Plugins de Eclipse
<p>Gestión de Ciclo de Vida</p>	<ul style="list-style-type: none"> • Versionamiento colaborativo • Búsqueda avanzada • Gestor gráfico de dependencias • Versionamiento local • Soporta esquemas XML

Editor de Mapeo	<ul style="list-style-type: none"> • Validación transparente de esquemas • Mapeo avanzado de funciones preconstruidas • Soporta esquemas complejos • Interfaz de usuario point-and-click
Despliegue de procesos	<ul style="list-style-type: none"> • Revisión dinámica de consistencia • Despliegue sencillo de procesos
Modelador de Procesos	<ul style="list-style-type: none"> • Compensación de flujos • Validación semántica transparente • Permite la reusabilidad de procesos • Exportación del mapa del proceso (PDF, PNG, SVG) • Interfaz de usuario drag-and-drop • Modelador de procesos multi-carriles
Introspección del sistema	<ul style="list-style-type: none"> • Conectores visuales para todos los sistemas que lo soportan • Sistema de internas de generación automática de WSDL

Tabla N° 1. Características generales de Intalio|Designer

Fuente: Felipe. 2008

2.6.2 Intalio|Server

Intalio|Server es un servidor nativo de BPEL 2.0 basado en la arquitectura de de J2EE y certificado para un amplio conjunto de plataformas de hardware, sistemas operativos, servidores de aplicación y servidores de base de datos. Intalio|Server pertenece al proyecto de Código Abierto de Apache ODE. La perspectiva de servidor de Intalio lo podemos visualizar en la figura N° 19.

Este engine permite ejecutar transformaciones XSLT durante la ejecución de procesos mediante la función doXSLTransform de BPEL.

Al momento de ejecutar procesos, Intalio|Server utiliza el marco de trabajo JACOB, el cual proporciona el apoyo para la persistencia y la concurrencia en tiempo de ejecución.

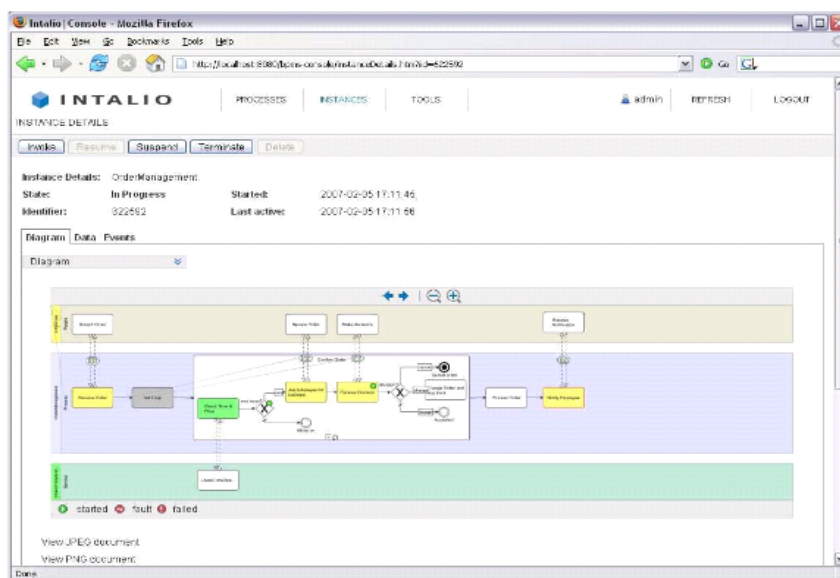


Figura Nº 19. Intalio|Server

Fuente: www.intalio.com, 2009

Intalio|Server incluye un avanzado marco de trabajo de Workflow que implementa BPEL4People para la ejecución de flujos de trabajos. Este componente está basado en el proyecto Apache Tempo (El funcionamiento de esta arquitectura es ampliado en el anexo D), el cual integra formas de trabajo mediante XForms, soportando el intercambio de mensajes con los procesos, la definición de usuarios, roles y diversos patrones de Workflow. Este framework es responsable de la gestión de las tareas del workflow, incluyendo asignación, resolución de roles para usuarios, delegación y alertas de escalación.

Intalio|Server está diseñado para ser desplegado en lo más alto de la Arquitectura Orientada a Servicio (SOA - *Service Oriented Architecture*). Todos los sistemas externos se exponen transparentemente como servicios web de Lenguajes de Definición de Servicios Web (WSDL – *Web Services Definition Language*), y los procesos desplegados pueden colocar sus interfaces WSDL sobre un registro del catálogo UDDI (*Descripción Universal, Descubrimiento e Integración - Universal Description, Discovery and Integration*). La capa de

integración a servicio web de Intalio|Server esta accionado por Apache Axis2. la estructura gráfica que define Intalio lo podemos observar a través de la figura N° 20

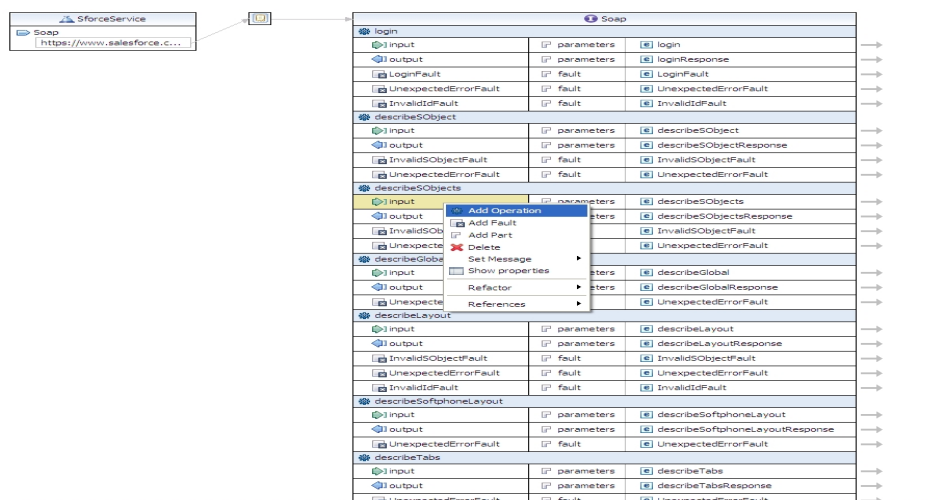


Figura N° 20. Editor WSDL

Fuente: www.intalio.com, 2009

Las características en general de Intalio|Server la podemos observar en la siguiente tabla:

Hardware que lo soporta	<ul style="list-style-type: none"> • HP-PA RISC de 32 4 64 bits • IBM Power de 32 y 64 bits • Intel x86 Intel Itanium de 32 y 64 bits • Opteron EM64 • Sun Sparc de 32 y 64 bits
Sistemas operativos que lo soporta	<ul style="list-style-type: none"> • HP-UX 11.31 para PA-Risc 64 bits y HP-UX 11.23 para Itanium 64 bits • IBM AIX 5.2, 5.3 • Sun Solaris 8, 9, 10 • SUSE Linux 9, 10 • Red Hat Linux 4, 5 • Windows 2003 Server, 2000 Server
Servidores de aplicación	<ul style="list-style-type: none"> • Apache Geronimo 2.0.1 • Apache Tomcat 5.5.26 • JBoss Application Server 4.0.5GA • IBM WebSphere Application Server 6.1.0.3
Bases de Datos	<ul style="list-style-type: none"> • Derby 10.2 • IBM DB2 8.1, 8.2, 9.1, 9.5 • MS SQL 2000, 2005 (Windows OS only) • MySQL Enterprise 5.0

	<ul style="list-style-type: none"> • Oracle 8i, 9i, 10g • PostgreSQL 8.1.10, 8.2.5 • Sybase ASE 12.5.1
Despliegue y Gestión	<ul style="list-style-type: none"> • Carga-Balanceo • Clustering • Instrumentación JMX • Depuración de API
Consola de Gestión	<ul style="list-style-type: none"> • Basada en Browser • Control de acceso basado en roles • Lista procesos desplegados • Gestión de procesos (Active y retirado) • Lista de instancias de procesos en ejecución • Interfaz de invocación de procesos basado en formularios
Ejecución de Procesos	<ul style="list-style-type: none"> • Compilador de procesos Just-in-Time • Permite BPEL 2.0 • Contenedor de servicios extensible • Soporta BPEL4WS
SOA	<ul style="list-style-type: none"> • Interfaces estilo REST • Interfaces POX (Plain Old XML) • Direccionamiento-WS • Atomicidad de transacciones-WS • Coordination-WS • Seguridad-WS
Conectores de Tecnologías	<ul style="list-style-type: none"> • Colas de datos AS400 • Correo electrónico (SMTP, POP3, IMAP4) • Sistema de archivos • FTP • HTTP,HTTP/S • JAAS • Java • JavaSpaces • JBI • JDBC • JTA • Oracle AQ • Plexus • EJB Remoto • REST • RMI • SOAP • Sistema de E/S • TCP • TIBCO • UDP • XFire
Conectores de Aplicaciones	<ul style="list-style-type: none"> • Aplicaciones de Google • Aplicaciones de Oracle • PeopleSoft • Salesforce.com • SAP • Siebel
	<ul style="list-style-type: none"> • Soporta XForms 1.1 • Diseñador XForms

Soporte dinámico para Formularios	<ul style="list-style-type: none"> • Motor XForms • Intalio AJAX
Gestor de Tareas de Workflow	<ul style="list-style-type: none"> • Tarea Planificación • Tarea Notificación • Tarea Control • Tarea Eliminación • Tarea Asignación • Gestor de Tareas Extensible • Soporte para BPEL4People
Entorno de Workflow	<ul style="list-style-type: none"> • Soporte de JSR 168 • Acceso de Control basado en roles • Portlet para creación de instancias de proceso

Tabla Nº 2. Características generales de Intalio|Server

Fuente: Felipe. 2008

Capítulo 3 – MARCO METODOLÓGICO

A continuación se muestra los aspectos metodológicos tanto para el desarrollo del software como para desarrollo de la investigación.

3.1 Metodología de desarrollo de software

Una metodología de desarrollo de software se define como “un conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a producir nuevo software” (Piattini. et al., 2001).

Maddison define a la metodología de desarrollo como:

“Un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de Sistemas de Información”. Maddison (1999)

En base a las definiciones anteriormente planteadas se puede decir que una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar software. Una metodología debe especificar los siguientes aspectos:

- División de un proyecto en etapas
- Tareas que se llevan a cabo en cada etapa
- Restricciones que deben aplicarse
- Técnicas y herramientas que se emplean
- Control y gestión de un proyecto. (Cavalho, 2005)

Con una buena metodología se pretende reducir costos y retrasos de proyectos así como mejorar la calidad del software.

Al momento de elegir una metodología general para desarrollar software debe tomarse en cuenta aquella que permita en todas sus fases la fácil adaptabilidad a cualquier tipo de requerimientos que el desarrollo exija.

Ante esta necesidad surgen las metodologías ágiles como respuesta a los problemas que presentan las metodologías tradicionales.

El desarrollo ágil persigue minimizar los tiempos de recolección y consolidación de requisitos y apostar en el desarrollo rápido e incremental que permita perfeccionar el software conforme se va construyendo. (Torres. et al, 2008).

El resultado es un software en menos tiempo y que se ajusta a las necesidades del cliente impuestas en muchos casos por un mercado que cambia de forma trepidante.

Ya que los procesos ágiles permiten una construcción rápida de software y, alternativamente, introducir cambios tan pronto como sea posible en cualquier paso, estos fueron tomados en cuenta en el desarrollo de esta investigación, específicamente el método planteado por el Proceso Unificado Ágil (*Agile UP - AUP*).

A continuación toda la descripción de esta metodología.

3.1.1 Proceso Unificado Ágil

El Proceso Unificado Ágil (*Ágile Unified Process o Ágile UP*) es un marco de desarrollo de software iterativo e incremental. Especifica muchas actividades y artefactos involucrados en el desarrollo de un proyecto software. Dado que es un marco de procesos, puede ser adaptado y la más conocida es RUP (*Rational Unified Process*) de IBM.

AUP se preocupa especialmente de la gestión de riesgos. “Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto” (Ambler - 2007).

Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos.

Al igual que en el Proceso Unificado Rational (*RUP – Rational Unified Process*), en AUP se establecen cuatro fases que transcurren de manera consecutiva:

- **Concepción:** el objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- **Elaboración:** el objetivo es profundizar en la comprensión de los requisitos del sistema y en validar la arquitectura.
- **Construcción:** durante la fase de construcción el sistema es desarrollado y probado totalmente en el ambiente de desarrollo.
- **Transición:** el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

La figura N° 21 describe el ciclo de vida de AUP

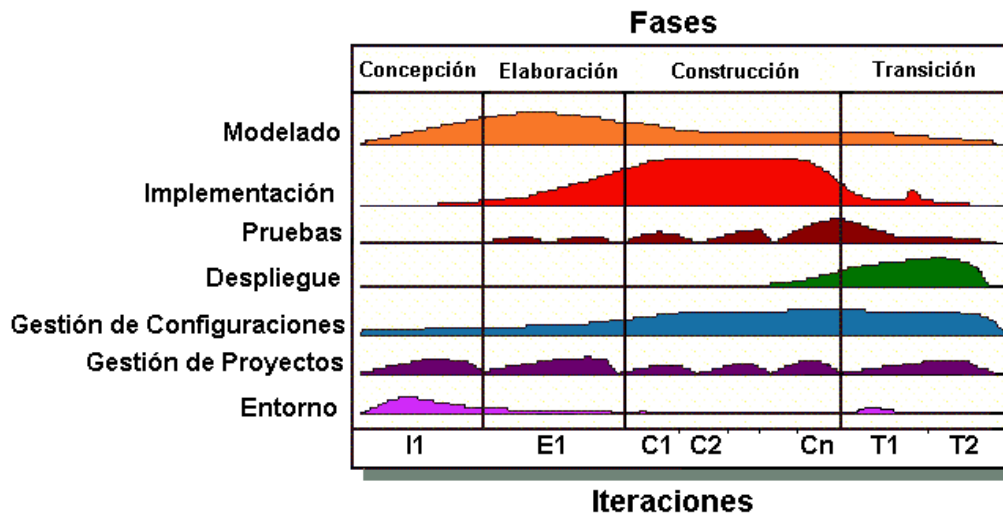


Figura Nº 21. Ciclo de vida del Proceso Unificado Ágil ó AUP

Fuente: Ambler. 2007

Las disciplinas de AUP son las siguientes:

- **Modelado:** su objetivo es entender el negocio de la organización, el dominio del problema del proyecto y determinar una solución viable para resolver el problema.
- **Implementación:** transforma el modelo en código ejecutable y realiza pruebas básicas.
- **Prueba:** realiza una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funcione como se estima y verificar el cumplimiento de requisitos.
- **Despliegue:** se encarga de planificar la forma en que el sistema estará a la disposición de los usuarios finales.
- **Gestión de configuraciones:** maneja el acceso a los artefactos del proyecto. Esto no sólo implica un seguimiento a las versiones del artefacto, sino también el control y la gestión de cambio de los mismos.
- **Gestión de proyectos:** dirige las actividades que se llevan a cabo en el proyecto. Incluye la gestión de riesgos, coordinación con el personal y

sistemas que estén fuera del alcance del proyecto para garantizar que el proyecto se entregue a tiempo y dentro del presupuesto establecido.

- **Entorno:** garantiza que el proceso adecuado, la orientación y las herramientas necesarias estén disponibles para el equipo cuando lo requieran.

Las disciplinas de AUP utilizadas en el transcurso de este Trabajo Especial de Grado fueron las de **Modelado**, que permitió definir el Proceso de Gestión de Cambios (roles, actividades, relaciones) utilizando la notación BPMN; **Implementación**, durante esta disciplina se configuró el motor de ejecución de Intalio BPMS para convertir el diagrama BPMN de la disciplina anterior a un proceso ejecutable utilizando código BPEL; y por último **Prueba**; en esta disciplina se realizó pruebas de validación ingresando diferentes tipos de datos en los XForms. Las demás disciplinas no fueron realizadas debido a que el proceso ejecutable no fue llevado a un ningún entorno de producción

3.1.2 Metodología de Gestión de los Procesos del Negocio sustentada en el uso de patrones

Además de utilizar las bondades que proporciona el uso de la metodología UP ágil para el desarrollo de software, durante el desarrollo de esta investigación se utilizó como un marco de trabajo para la elaboración de los ejecutables la **Metodología de Gestión de los Procesos del Negocio sustentada en el uso de patrones**, propuesta por el Doctor Pedro Nolasco Bonillo Ramos en el año 2009.

En esta investigación doctoral Bonillo indica lo siguiente: “la propuesta metodológica para la gerencia de los procesos de negocio sustentada en el uso de patrones está conformada por dos macro-procesos: Creación de los procesos de negocio y Administración de los procesos de negocio en ejecución”. (Bonillo, 2009)

La figura N° 22 muestra la estructura de la Metodología de Gerencia de Procesos Sustentada en el Uso de Patrones.

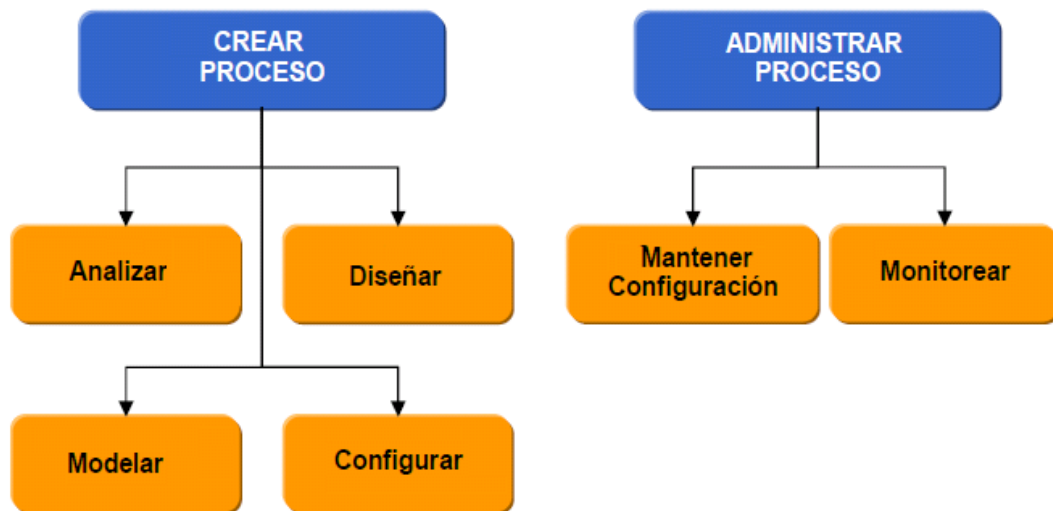


Figura N° 22. Estructura de la Metodología de Gerencia de Procesos Sustentada en el Uso de Patrones.

Fuente: Bonillo. 2009

De los macro-procesos definidos en esta metodología, se utilizó solo el macroproceso de Creación de los procesos de negocio, esto debido a que el alcance de este Trabajo Especial de Grado no incluye el monitoreo del proceso de Gestión de Cambios.

Los subprocesos de “Crear Proceso” propuestos en esta Metodología de Gerencia de Procesos sustentada en el Uso de Patrones se corresponden con este Trabajo Especial de Grado de la siguiente manera:

- **Analizar:** durante esta etapa se revisó las licitaciones de requerimientos. Esto implicó abordar temas de investigación tales como indagar todos los aspectos de marco de referencia para la gestión de procesos ITIL.

- **Diseñar:** acá se utilizó diferentes diagramas de UML para establecer una visión global del proceso de gestión de cambios para detectar las diferentes tareas y roles de este
- **Modelar:** en este paso se modeló el proceso de Gestión de Cambios usando BPMN, lo que permitió trabajar con una notación centrada en los procesos de negocio y que además es avalada a nivel mundial por organizaciones como la *Workflow Management Coalition*.
- **Configurar:** este paso corresponde a la configuración del motor de ejecución de la herramienta de software libre Intalio en BPEL. Esto permitió definir tareas que pueden ser ejecutadas con facilidad y que pueden ser exportadas a otro ambiente organizativo dado a que usa la notación estándar BPEL.

3.2 Bases metodológicas para el desarrollo de la investigación

En este punto se describen las bases metodológicas utilizadas para realizar el Trabajo Especial de Grado, tomando en cuenta que “la metodología constituye la médula del plan, se refiere a la descripción de las unidades de análisis, o de investigación, las técnicas de observación y recolección de datos, los instrumentos, los procedimientos y las técnicas de análisis”. (Tamayo, 2000).

3.2.1 Tipo de Investigación

De acuerdo al problema planteado, referido a la necesidad de realizar un modelado eficiente del proceso de Gestión de Cambios para el Soporte de Servicios de TI, este Trabajo Especial de Grado se define del tipo de investigación cualitativa; la cual consiste en: “identificar la naturaleza profunda de las realidades, su sistema de relaciones, su estructura dinámica. Busca el significado de las cosas; es exploratorio y explicativo. Los resultados arrojados son muy representativos pero no cuantitativamente proyectables”. (Rodríguez, 1996)

3.2.2 Método de Investigación Cualitativa

En atención a este paradigma de investigación, se establece el Método de Investigación Descriptiva como método cualitativo a utilizar en este Trabajo Especial de Grado. En el Método de Investigación Descriptiva “el objetivo es describir la estructura de los fenómenos y su dinámica” (Sabino, 1992).

La metodología descriptiva se adapta a la investigación del proceso de Gestión de Cambios, ya que esta permite describir completamente su estructura (reflejando el dinamismo en cada paso) y la relación con otros procesos pertenecientes al Soporte de Servicios.

En este trabajo el nivel de investigación será exploratorio, revisando la Gestión de Cambios en el dominio de los procesos de Soporte de Servicio de ITIL, estableciendo una realidad dinámica para alcanzar una correcta interpretación y evaluación del proceso.

3.2.3 Tipo de Población

Según la concepción de Tulio Ramírez, “se debe escoger una muestra cuyo tamaño garantice la representatividad del resto de la población” (Ramírez, 1999).

Dado que en la investigación se pretende exponer la gestión de un proceso, la muestra será establecida de forma tal que abarque todo el dominio de las personas involucradas con la Gestión de Cambios en la TI, bien sea directa o indirectamente.

La población de este Trabajo Especial de Grado se corresponde a una *población finita*, debido a que este depende de los *stakeholders* (personas que pueden afectar o son afectados por las actividades de una empresa) implicados en

el proceso de Gestión de Cambios, tales como el Encargado del Cambio, Iniciador de Cambio y el Encargado de Versiones.

3.2.4 Técnicas e instrumentos de recolección de datos

Las herramientas y técnicas variaban, dependiendo de la naturaleza de cierta información que era requerida. En este sentido la técnica de recolección de datos empleada fue la revisión bibliográfica y las fuentes infográficas.

- **Revisión Bibliográfica:** se recurrió a la técnica de revisión bibliográfica; tanto de libros, folletos, documentos, revistas, seminarios y entre otros, los cuales brindaron todo el soporte del marco teórico. La revisión bibliográfica, fue utilizado como base complementaria al proyecto central, con el fin de recopilar y revisar todos aquellos documentos que permitan confrontar el aspecto teórico con la situación real o práctica dentro del modelo de evaluación de las capacidades tecnológicas necesarias en el mercado de la gestión del proceso de cambio.
- **Fuentes Infográficas:** fue el instrumento de mayor uso durante el desarrollo del Trabajo Especial de Grado. Estas fuentes comprendía amplios medios en línea para recopilar información como webinars, foros, documentaciones, etc.

Capítulo 4 – RESULTADOS

En este capítulo se presenta los resultados de haber implementado la solución de software libre Intalio, para modelar el proceso de Gestión de Cambios.

Los entregables del Trabajo Especial de Grado corresponden a los que establece el macro-proceso “Crear Proceso” de la Metodología de Gerencia de Procesos sustentada en el Uso de Patrones propuesta por el doctor Pedro Bonillo.

Como se explico en el punto 3.1.2, este macroprocesos contiene 4 procesos: Analizar, Diseñar, Modelar, Configurar. A continuación se explicara los entregables elaborados en cada etapa:

4.1 Análisis de requerimientos.

Esta etapa corresponde a la respectiva investigación realizada para llevar a cabo esta investigación. Esto implicó indagar sobre temas como el proceso de Gestión de Cambios propuesto por la Biblioteca de Infraestructura de Tecnologías de Información, evaluación de la herramienta BPM Intalio y otros aspectos relacionados a la notación BPMN. Este entregable se puede visualizar a lo largo del Marco Teórico de este Trabajo especial de Grado.

4.2 Diseño de diagramas.

Durante esta etapa se elaboraron diferentes diagramas en UML para establecer la estructura necesaria del proceso de Gestión de Cambios y establecer las funcionalidades que realizara el sistema. Los resultados de esta entregable pueden visualizarse en el anexo A.

4.3 Modelado del Proceso de Gestión de Cambios

Mediante el diseñador de Intalio, se realizó el diagrama del proceso de Gestión de Cambio en notación BPMN. El diagrama puede ser visualizado en el anexo G.

4.4 Configuración del motor de ejecución del BPMS

Este paso corresponde a la configuración del motor de ejecución de la herramienta de software libre Intalio. Aquí se realizó la creación de las correlaciones, creación de los esquemas XML y las invocaciones de los web services. Los detalles de la creación del modelado del procesó de Gestión de Cambios lo podemos ver en el anexo E.

4.5 Ejecución del proceso de Gestión de Cambios

A continuación se muestra el lector todos los aspectos necesarios para ejecutar el proceso de Gestión de Cambios modelado a través de la herramienta Intalio BPMS.

Al momento de diagramar el proceso de gestión de cambios con la herramienta Intalio|Designer se definió tres roles:

- **Cliente iniciador:** es aquel que genera la petición de cambio. Proporciona toda la información referente al cambio que desea que se realice, así como todos los datos personales. También aprueba los tiempos estimados para cambios urgentes.
- **Encargado del cambio:** evalúa la petición de cambio propuesta por el iniciador, decide cuando llevar a cabo el cambio, recomienda un desarrollador para revisar la petición, revisa la recomendación de la gestión de versiones en cuanto a la estimación de tiempo de la implementación del cambio, indica el estimado del trabajo (SOW) al cliente iniciador, indica el

comienzo de implementación del cambio y verificar el reporte del desarrollador de versiones

- **Desarrollador de versiones:** aunque el desarrollador de versiones no pertenece al proceso de gestión de cambios, tiene una gran relación con este, ya que el llevará a cabo la implementación del cambio, por lo tanto este se encarga de reportar al encargado el tipo de cambio a implementar y debe confirmar cuando el cambio fue implementado.

La versión community de Intalio permite el uso de roles y actores por defecto. Estos se encuentran definidos en el archivo security.xml de la carpeta "C:\intalio-bpms-5.2.0.050\var\config".

Las instancias de los roles usados son los siguientes:

- **Cliente Iniciador:** "William Davis"
- **Encargado de Cambio:** "Michael Brown"
- **Desarrolladores:** "James Smith", "John Johnson", "Robert Williams" y "Mary Jones".

Al momento de iniciar cada tarea, dependiendo del rol correspondiente, estos deben loguearse, con su nombre de usuario y su contraseña. La tabla N° 3 nos demuestra la información para loguearse cada rol:

Nombre	William Davis	Michael Brown	James Smith	John Johnson	Robert Williams	Mary Jones
Nombre de usuario	"wdavis"	"mbrown"	"jsmith"	"jjohnson"	"rwilliams"	"mjones"
Contraseña	"password"	"password"	"password"	"password"	"password"	"password"

Tabla N° 3. Nombres de usuarios y contraseñas

El modelado de procesos contempla dos flujos principales (Cambio Estándar y Cambio Urgente) y un flujo alternativo (Clarificación de Cambio).

El primer paso para iniciar la ejecución del proceso de Gestión de Cambios es realizar un despliegue o “deploy” (es importante destacar que previamente el servidor Apache Geronimo debe estar inicializado). Para esto nos vamos a la barra de menú > Project > Deploy Project. Acá nos aparece un cuadro de dialogo, tal como lo muestra la figura N° 23.

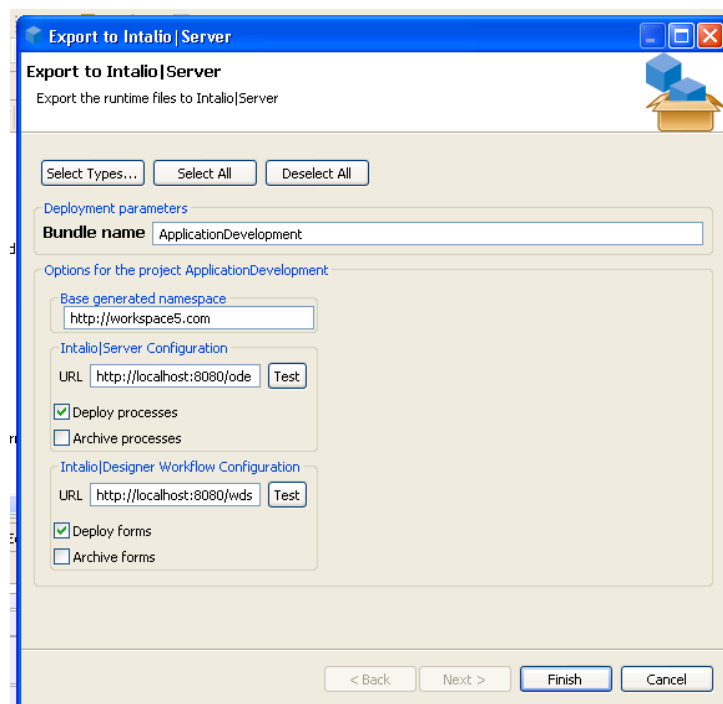


Figura N° 23. Cuadro de diálogo para exportar el proyecto al servidor de Intalio

El servidor de Intalio comienza a verificar si existen errores en el diagrama y a realizar el despliegue de las tareas en el modulo Intalio|Workflow. Una vez desplegados en el servidor las tareas, le damos click en la pestaña “Progress” y nos carga el formulario de acceso para comenzar la ejecución del diagrama modelado, como lo demuestra la figura N° 24.

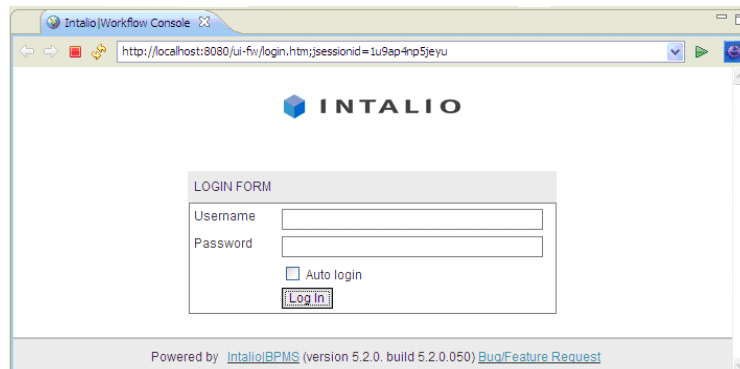


Figura N° 24. Pantalla de acceso a Intalio|Workflow

A continuación comenzamos con los flujos del molado del proceso de Gestión de Cambios modelado en Intalio|Designer.

4.5.1 Cambio Estándar

El primer paso es loguearnos como un Cliente Iniciador. Una vez que hacemos esto nos aparece la pantalla de la figura N° 25. Acá hacemos click a la actividad Change Request.

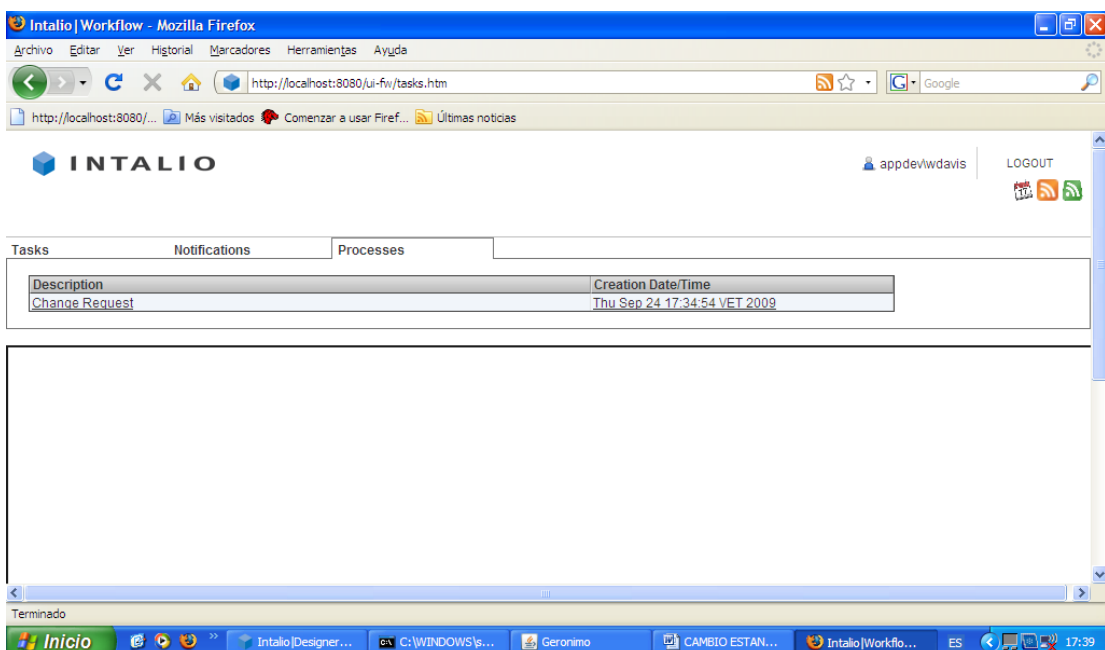


Figura N° 25. Proceso Change Request

Nos aparece la pantalla de la figura N° 26. Acá el iniciador del cambio debe llenar la petición de cambio, indicando todo la descripción del cambio que desee (nombre del iniciador, departamento, software o hardware involucrado, etc).

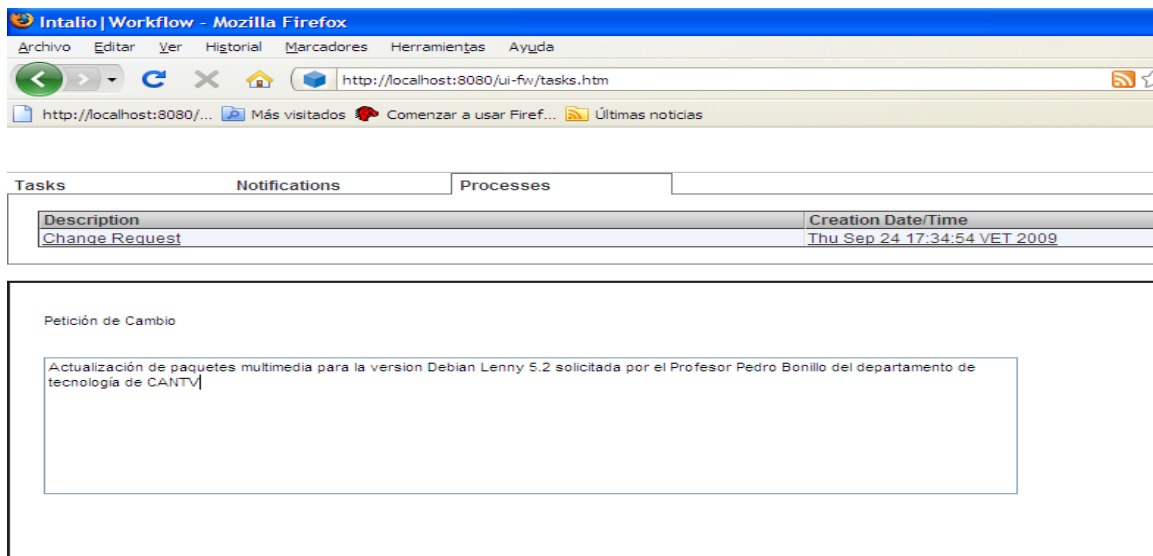


Figura N° 26. Petición de Cambio

Para iniciar este proceso le damos click en el botón "Start Process" y mientras carga nos muestra el siguiente mensaje: "Sending Request, please wait", esto nos indica que el proceso esta siendo levantado en el servidor.

Una vez finalizado el proceso de carga del proceso, nos desconectamos de la sesión de Cliente iniciador y accedemos como el Encargado del Cambio.

Al loguearnos nos aparece la tarea "Assign Developer to Change Request", tal cual como nos demuestra la figura N° 27 de la pantalla. En esta tarea el Encargado le asigna a la petición de cambio generada en el paso anterior un Desarrollador perteneciente a la Gestión de Versiones.

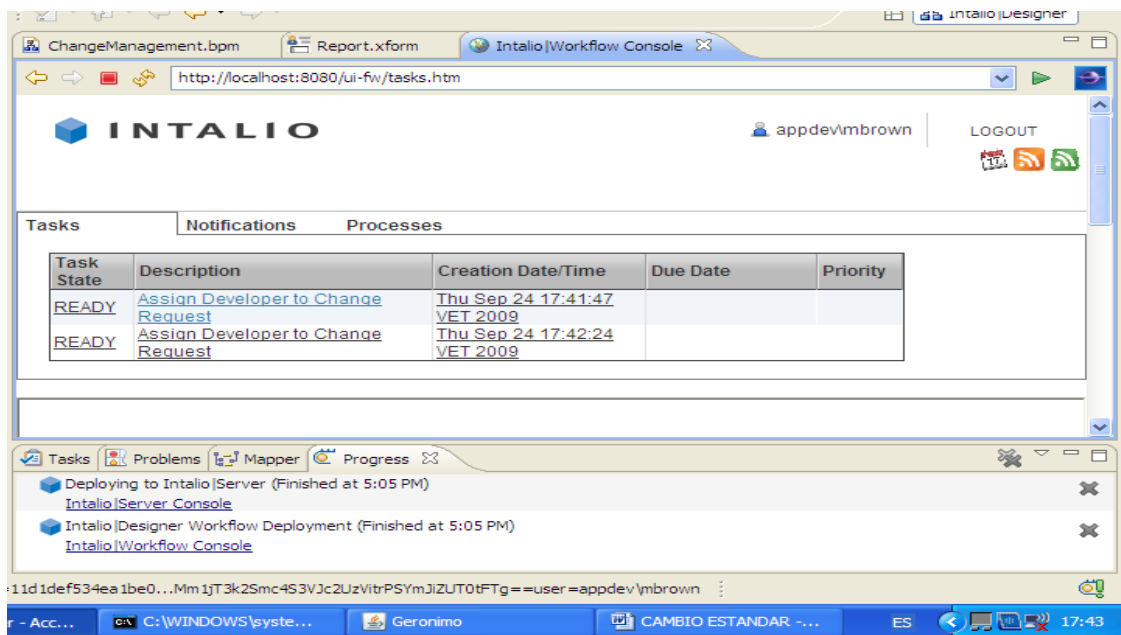


Figura N° 27. Asignación de desarrollador a la petición de Cambio

Al hacer click en la tarea creada nos aparece un combo box con los nombres de los desarrolladores que se pueden asignar para que verifiquen una petición. Para efectos del ejercicio seleccionamos al desarrollador Robert Williams. Una vez seleccionado le damos click en el botón “Complete” y mientras carga nos muestra el siguiente mensaje: “Sending Request, please wait”, esto nos indica que la tarea esta siendo levantada en el servidor.

Una vez finalizado el proceso de carga de tarea, nos desconectamos de la sesión de Encargado del Cambio y accedemos como el Desarrollador asignado, en este caso Robert Williams.

Logueados como el desarrollador Robert Williams, se no aparece la tarea “Provide Estimate for Change Request” tal como indica la figura N° 28.

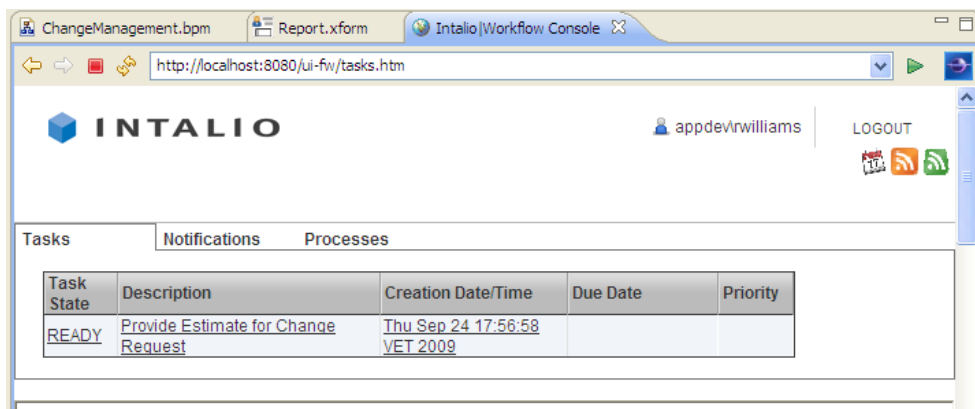


Figura N° 28. Tarea Proveer estimado a la petición de cambio.

Acá el desarrollador asignado lee la petición de cambio generada por el Iniciador y, según los parámetros establecidos en la petición, el desarrollador indica el tipo de cambio este va a desplegar. Acá hacemos click en la opción Cambio estándar, tal como indica la figura N° 29.

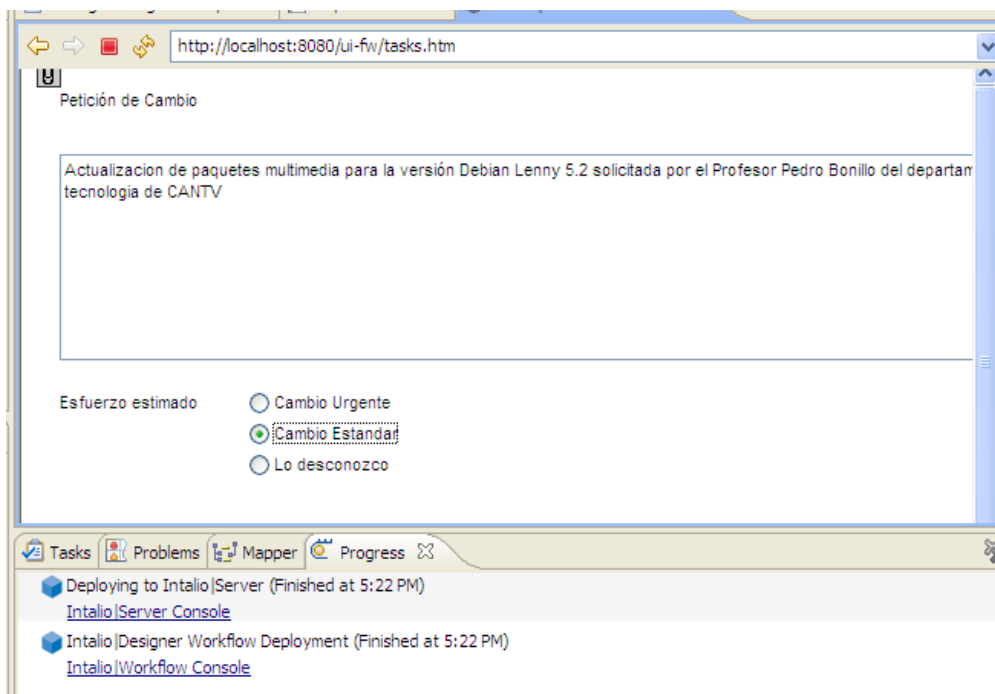


Figura N° 29. Asignar prioridad de estándar al cambio

Ya seleccionado la opción Cambio Estándar, le damos click en el botón “Complete” y mientras carga nos muestra el siguiente mensaje: “Sending Request, please wait”, esto nos indica que la tarea esta siendo levantada en el servidor.

Una vez finalizado el proceso de carga de tarea, nos desconectamos de la sesión del Desarrollador asignado, Robert Williams, y nos volvemos a loguear como el Encargado del Cambio.

Logueados como el Encargado del Cambio, se no aparece la tarea “Review Report from Developer” tal como indica la figura N° 30.

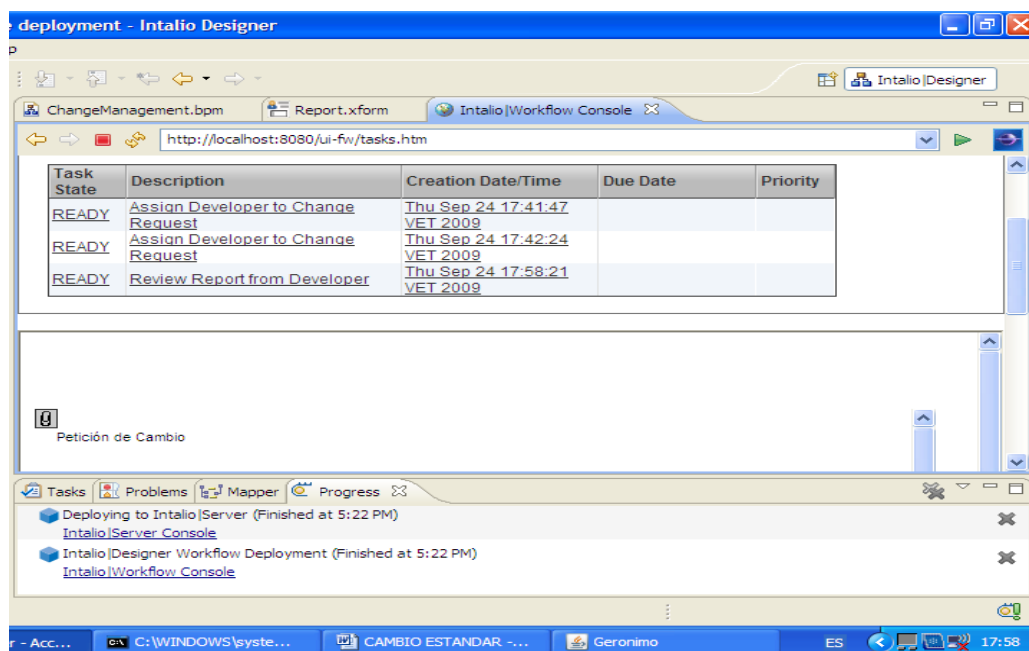


Figura N° 30. Tarea Revisar Reporte de Desarrollador

Al hacer click sobre ella nos aparece la pantalla N° 31 con varias opciones. Acá seleccionamos la opción “Comenzar trabajo con el desarrollador asignado actualmente” y luego le damos click en el botón “Complete”.

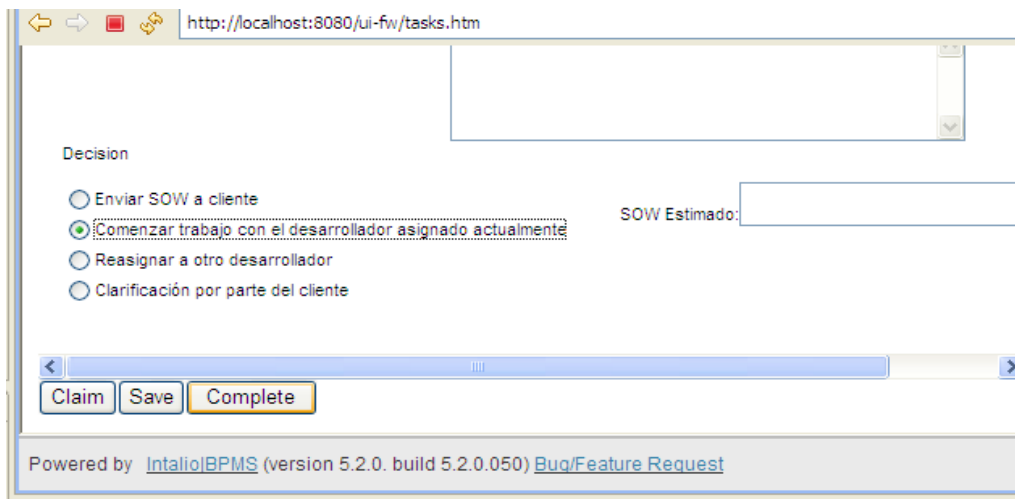


Figura Nº 31. Decisión de comenzar trabajo con el desarrollador asignado actualmente

Ya seleccionado la opción comenzar trabajo con el desarrollador asignado actualmente, le damos click en el botón “Complete” y mientras carga nos muestra el siguiente mensaje: “Sending Request, please wait”, lo cual indica que la tarea esta siendo levantada en el servidor. En este paso el Encargado indica que la petición de cambio estándar ha sido confirmada

Una vez finalizado el proceso de carga de tarea, nos desconectamos de la sesión del Encargado del Cambio y accedemos nuevamente como Desarrollador.

Logueados como el Desarrollador de Versiones, se no aparece la tarea “Change Request confirmed” tal como indica la figura Nº 32.

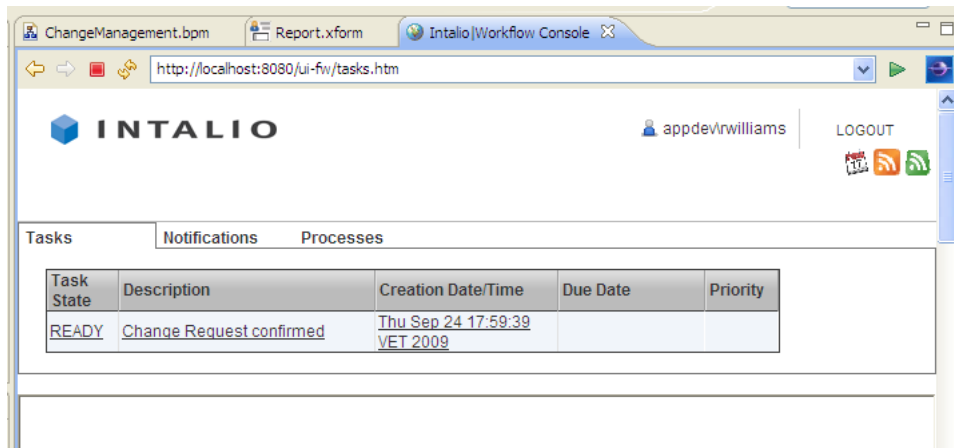


Figura N° 32. Tarea Change Request confirmed

Acá hacemos click en la tarea mostrada y el desarrollador puede visualizar la petición de cambio original creada por el iniciador de Cambio y aprobada por el Gestor de Cambios ha sido asignada para el. Una vez que el Desarrollador se ha fijado que la petición de cambio ha sido asignado para el este debe darle click en el botón “Complete”

Este paso es totalmente abstracto para el proceso de Gestión de Cambios, pero sin embargo es necesario que cuando el cambio haya sido desarrollado completamente por el Desarrollador, este le notifique al Gestor del Cambio que ha sido implementado. La figura N° 33 nos indica que la petición de cambio estándar ha sido aprobada y asignada a un desarrollador.

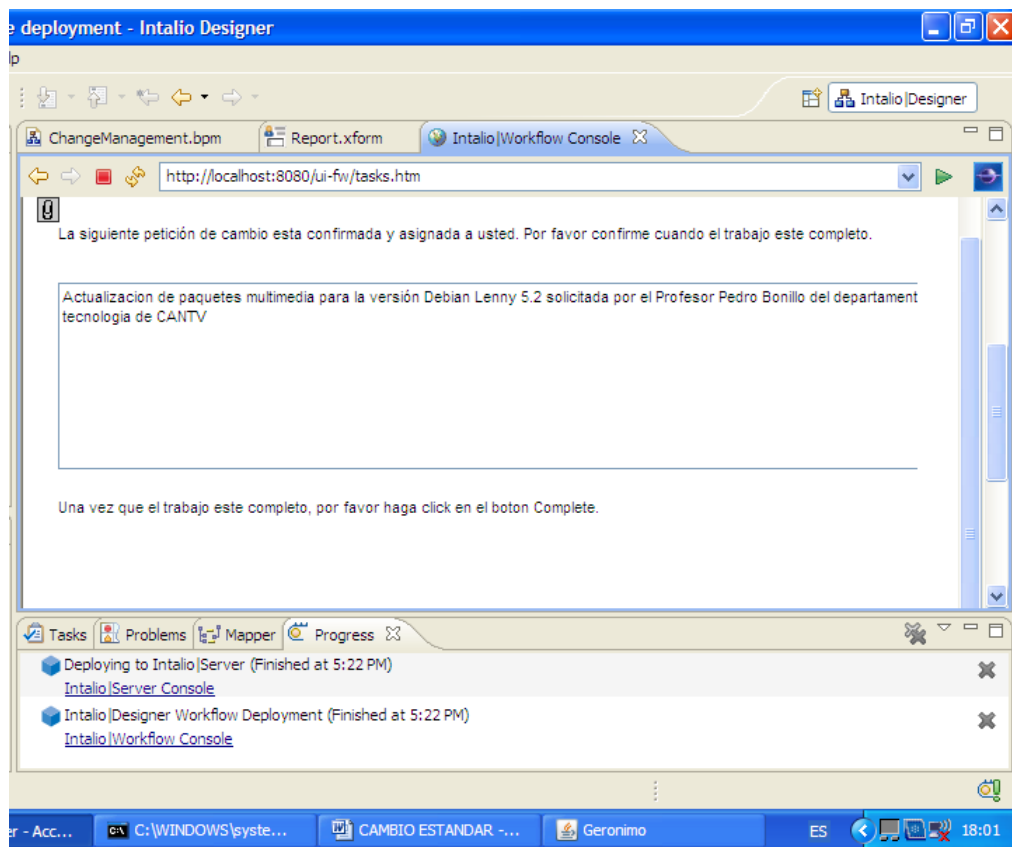


Figura N° 33. Change Request confirmed

Ahora nos salimos de la sesión de Desarrollador y volvemos a la sesión de Encargado. Acá nos vamos a la pestaña "Notifications" y aparece la notificación generada por el Desarrollador una vez que este ha implementado el cambio estándar procesado anteriormente. La figura N° 34 nos demuestra esta pantalla de la notificación.

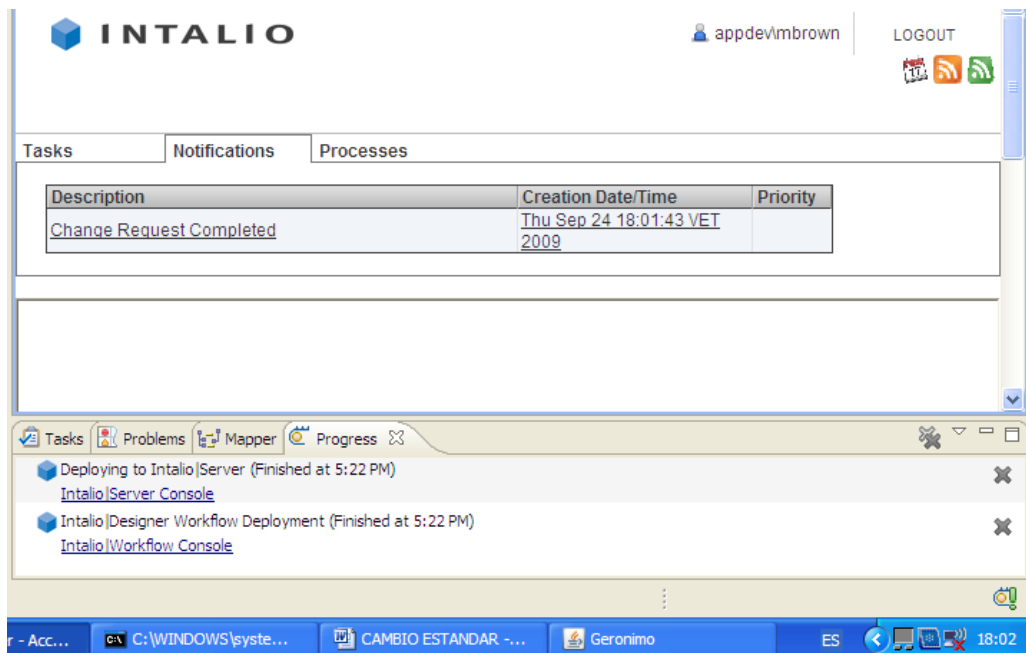


Figura Nº 34. Notificación de Gestión de Cambio completada

El encargado del cambio hace click en esta notificación y puede visualizar que la petición de cambio estándar ha sido implementada por la Gestión de Versiones, a través del desarrollador asignado para la petición. Esta notificación la podemos ver en la figura Nº 35.

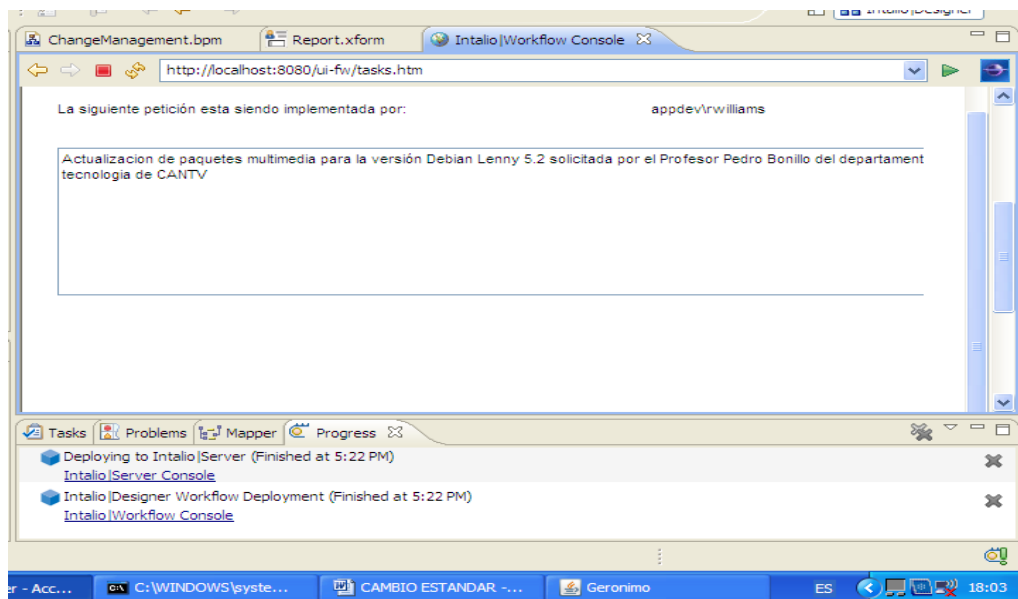


Figura Nº 35. Notificación de Cambio estándar aplicado.

Una vez que el Encargado revisa que el cambio ha sido implementado, se hace click en el botón “Dismiss” para detener la ejecución del proceso y en el centro de la pantalla nos aparece el mensaje “Dismissed” indicando que el proceso ha sido terminado.

Para verificar que el proceso ha sido terminado, nos cambiamos de ventana y le damos click en la parte inferior de la pestaña “Progress” en la parte donde dice Intalio|Server y se despliega la ventana de las Instancias de procesos de Intalio. Acá podemos visualizar la última versión del proceso de gestión de cambios que ha sido completada. La figura N° 36 nos muestra este visualizador de instancias.

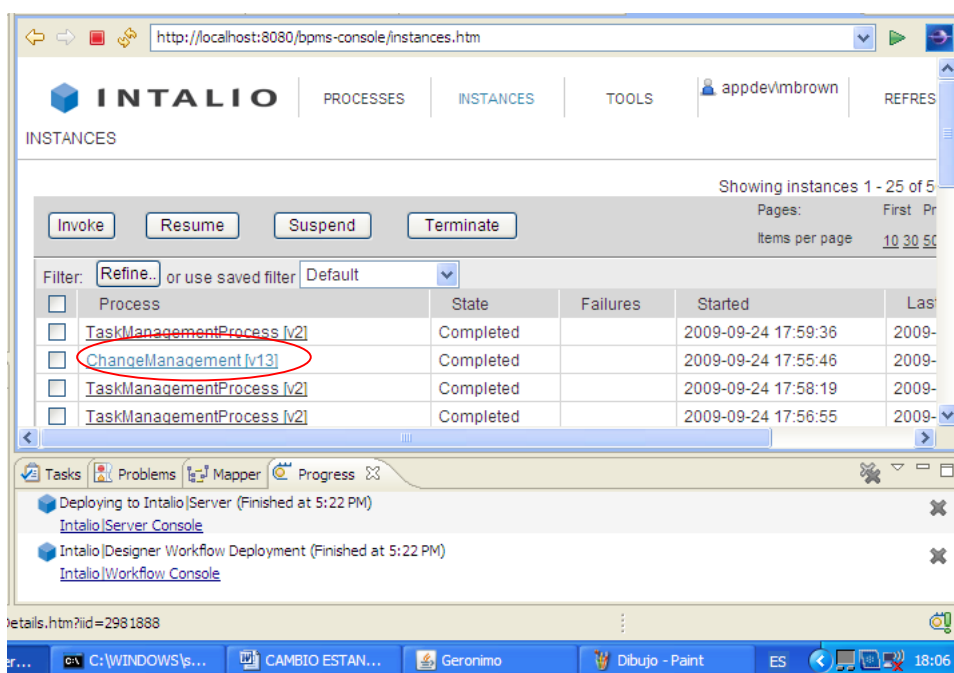


Figura N° 36. Visualizador de instancias

4.5.2 Cambio Urgente

Al igual que en el cambio estándar, lo primero que se debe generar es la petición de cambio por Parte del Cliente Iniciador. Por eso debemos loguearnos

como tal. Una vez que hacemos esto nos aparece la pantalla de la figura N° 37. Acá hacemos click en el proceso Change Request.

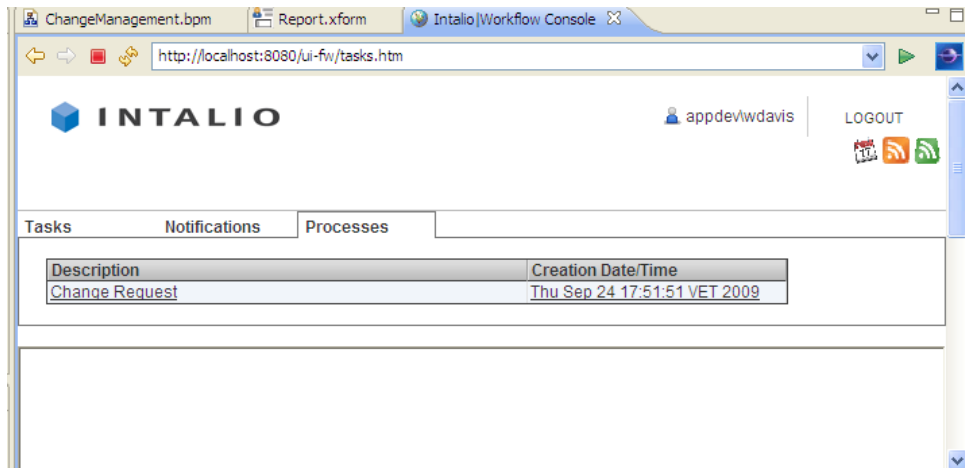


Figura N° 37. Proceso Change Request urgente

Nos aparece la pantalla de la figura N° 38. Acá el iniciador del cambio debe llenar la petición de cambio, indicando todo la descripción del cambio que desee (nombre del iniciador, departamento, software o hardware involucrado, etc).

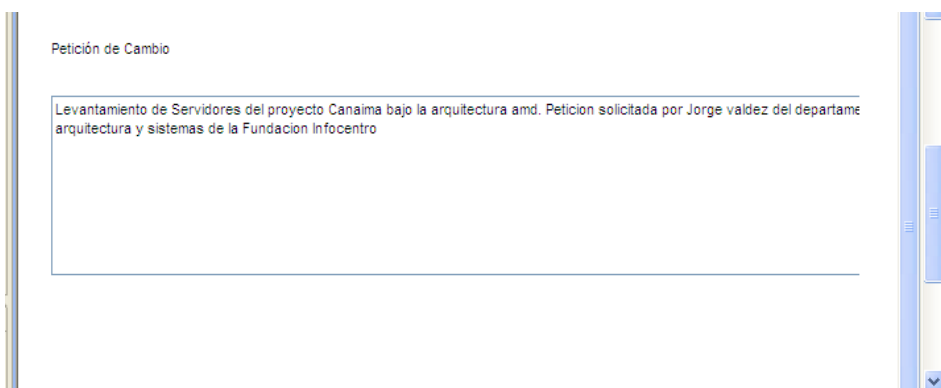


Figura N° 38. Petición de Cambio urgente

Para iniciar este proceso le damos click en el botón "Start Process" y mientras carga nos muestra el siguiente mensaje: "Sending Request, please wait", esto nos indica que el proceso esta siendo levantado en el servidor.

Una vez finalizado el proceso de carga del proceso, nos desconectamos de la sesión de Cliente iniciador y accedemos como el Encargado del Cambio.

Al loguearnos nos aparece la tarea “Assign Developer to Change Request”, tal cual como nos demuestra la figura N° 39 de la pantalla. En esta tarea el Encargado le asigna a la petición de cambio generada en el paso anterior un Desarrollador perteneciente a la Gestión de Versiones.

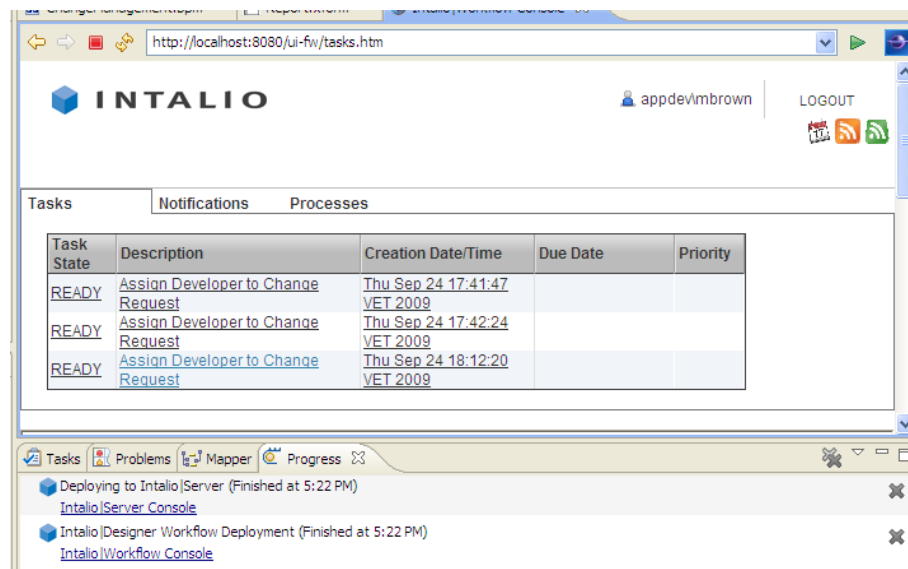


Figura N° 39. Asignación de desarrollador a la petición de Cambio urgente

Al hacer click en la tarea creada nos aparece un combo box con los nombres de los desarrolladores que se pueden asignar para que verifiquen la petición. La figura N° 40 nos demuestra esto.

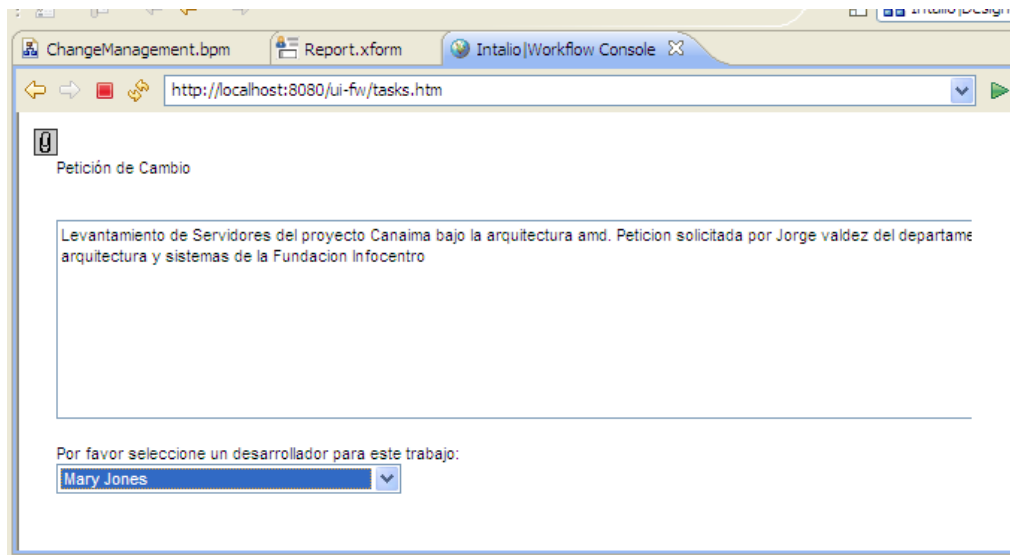


Figura N° 40. Asignación de desarrollador a la petición de Cambio urgente

Para efectos del ejercicio seleccionamos al desarrollador Mary Jones. Una vez seleccionado le damos click en el botón “Complete” y mientras carga nos muestra el siguiente mensaje: “Sending Request, please wait”, esto nos indica que la tarea esta siendo levantada en el servidor.

Una vez finalizado el proceso de carga de tarea, nos desconectamos de la sesión de Encargado del Cambio y accedemos como el Desarrollador asignado, en este caso Mary Jones.

Estando en la sesión de “Mary Jones”, aparece la tarea “Provide Estimate for Change Request” tal como indica la figura N° 41.

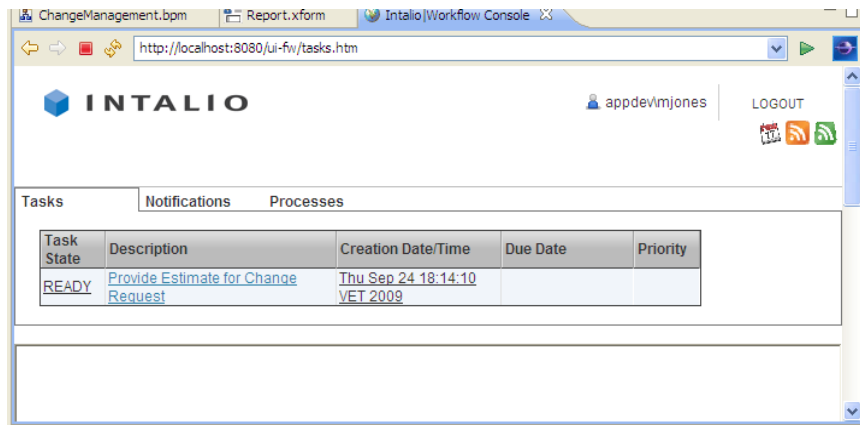


Figura N° 41. Tarea Proveer tiempo estimado para la petición de cambio urgente

Acá el desarrollador asignado lee la petición de cambio generada por el Iniciador y, según los parámetros establecidos en la petición, el desarrollador indica el tipo de cambio este va a desplegar. Acá hacemos click en la opción Cambio urgente, tal como indica la figura N° 42.

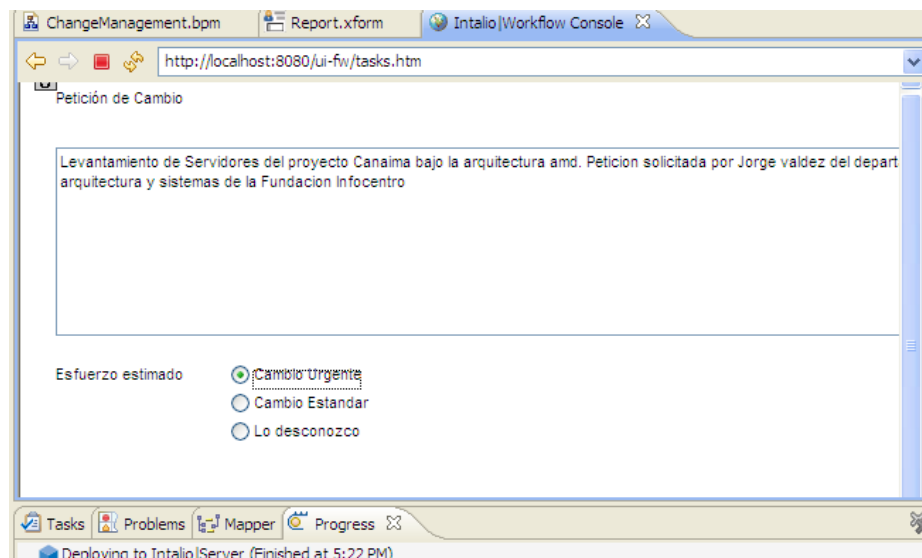


Figura N° 42. Asignar prioridad de urgente al cambio

Ya seleccionado la opción Cambio urgente, le damos click en el botón “Complete” y mientras carga nos muestra el siguiente mensaje: “Sending Request, please wait”, esto nos indica que la tarea esta siendo levantada en el servidor.

Una vez finalizado el proceso de carga de tarea, nos desconectamos de la sesión del Desarrollador asignado, Mary Jones, y nos volvemos a logear como el Encargado del Cambio.

Una vez iniciado la sesión como el Encargado del Cambio, aparece la tarea “Review Report from Developer” tal como indica la figura N° 43.

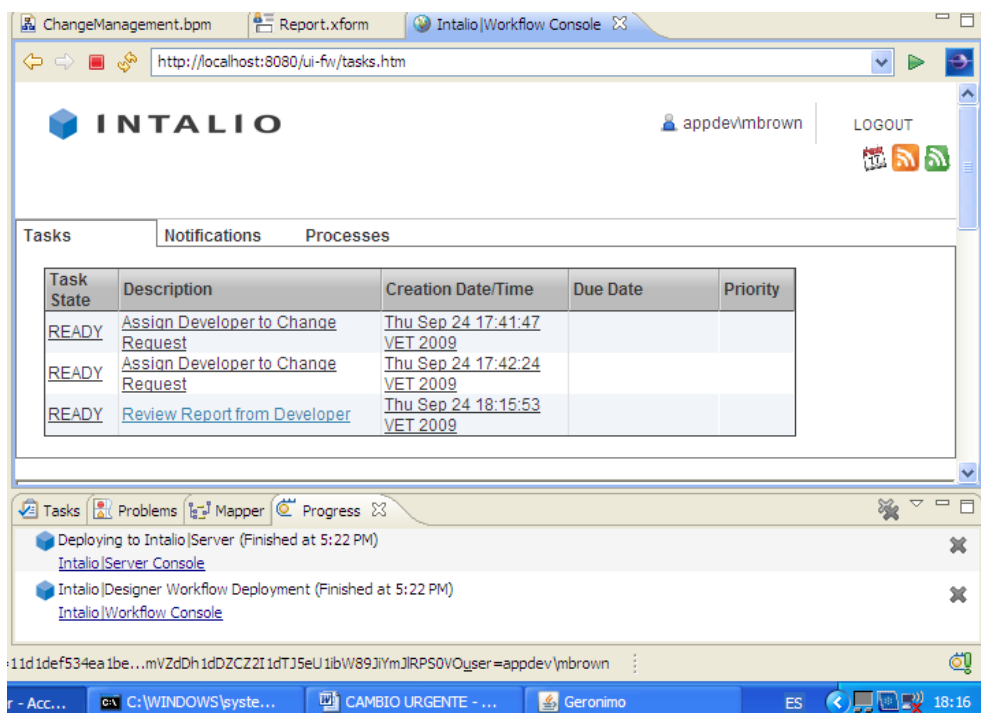


Figura N° 43. Tarea Revisar Reporte de cambio urgente de Desarrollador

Al hacer click sobre ella nos aparece la pantalla de la figura N° 44 con varias opciones. Acá seleccionamos la opción “Enviar SOW a cliente”, al lado aparece un text input donde indicaremos el tiempo estimado (en horas) en que se implementará el cambio urgente. Luego le damos click en el botón “Complete”.

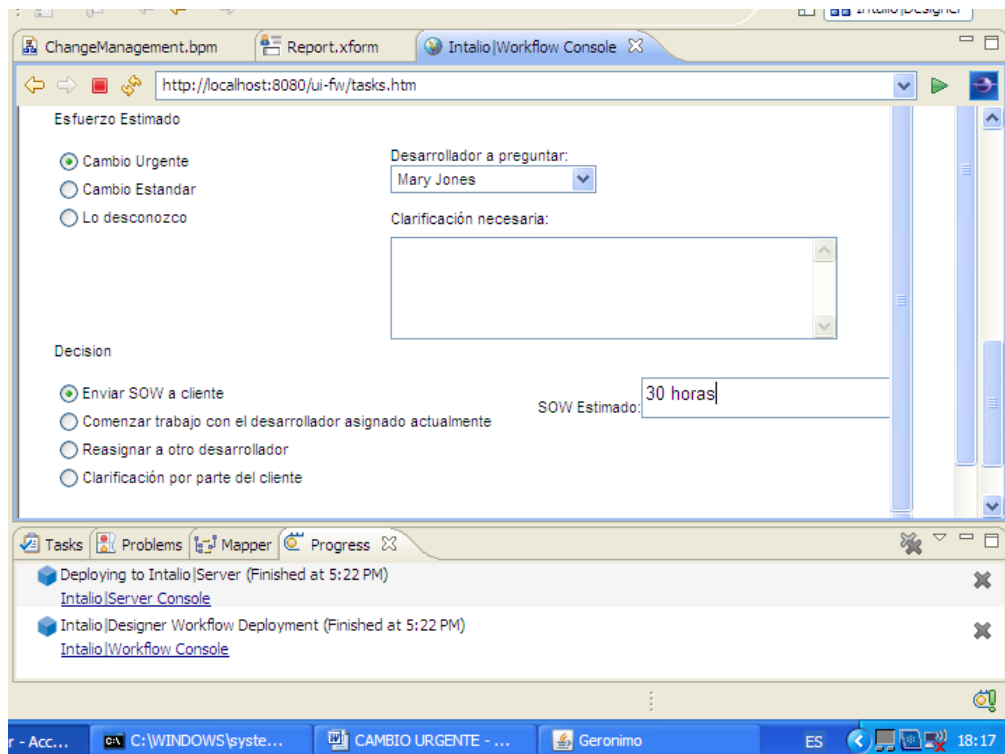


Figura Nº 44. Enviar estimado de trabajo al cliente

Nos salimos de la sesión del Encargado del cambiar e ingresamos como el Cliente Iniciador del Cambio. Una vez acá nos aparece la tarea *Change Request SOW*, que indica al iniciador el tiempo estimado en que se implementará el cambio. Esto no los muestra la figura Nº 45

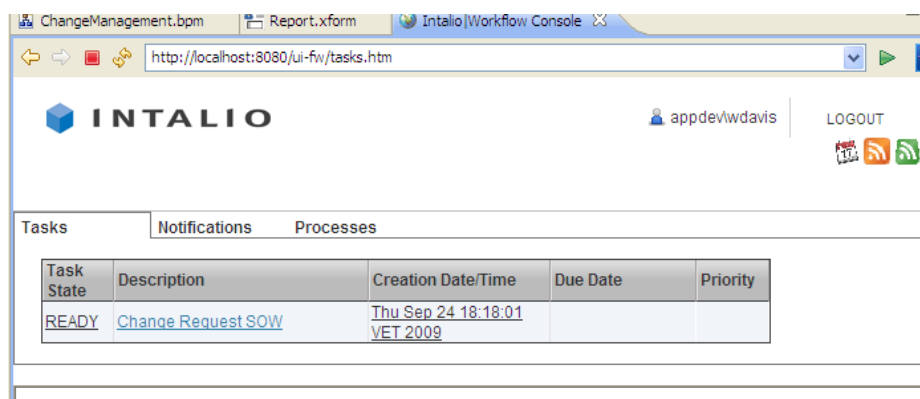


Figura Nº 45. Tarea de estimado de petición de cambio

El iniciador puede o no aceptar el tiempo estimado. En caso de no aceptarlo se cierra la petición de cambio y el cliente puede solicitar otro cambio en otro momento. Si lo acepta, el cambio comienza ser desplegado. Para efectos del modelado elegimos la opción “Si, el estimado es aceptable. Por favor comiencen la implementación”. Estas opciones la podemos ver en la figura N° 46

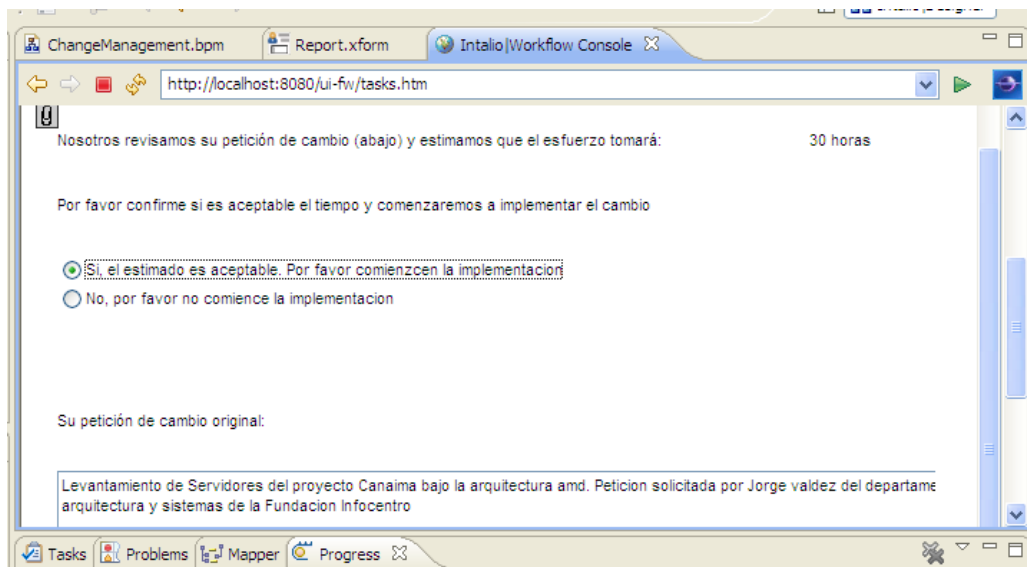


Figura N° 46. Tarea de aceptación o rechazo del estimado para el despliegue de la petición de cambio

Una vez seleccionada la opción de aceptación del estimado le damos al botón “Complete” y esperamos que la tarea se cargue. Ahora salimos de esta sesión e ingresamos como el desarrollador Mary Jones.

Acá encontramos la tarea “Change Request confirmed”, que nos indica que el cliente ha confirmado que el tiempo estimado para el desarrollo del cambio urgente es aceptable. La figura N° 47 nos indica esta tarea.

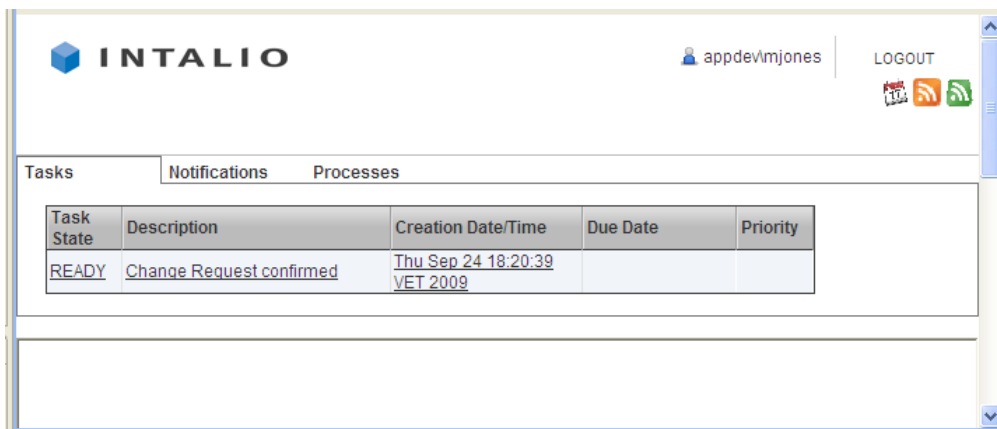


Figura N° 47. Tarea de petición de cambio urgente confirmada

Acá el desarrollador puede darse cuenta que la petición de cambio ha sido confirmada por el cliente y ha sido asignada por el Encargado a el. La figura N° 48 refleja esta situación

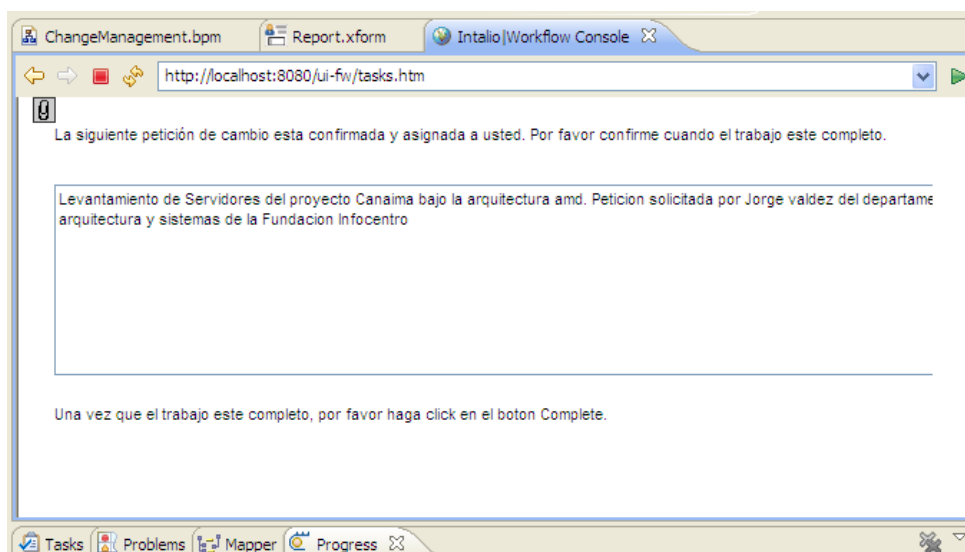


Figura N° 48. Confirmación de cambio urgente

Una vez que el Desarrollador se ha fijado que la petición de cambio ha sido asignado para el este debe darle click en el botón "Complete".

Al igual que en el flujo del cambio estándar, este paso es abstracto para el proceso de Gestión de Cambios, pero sin embargo es necesario que cuando el cambio haya sido desarrollado completamente por el Desarrollador, este le notifique al Gestor del Cambio que ha sido implementado.

Ahora salimos de la sesión del desarrollador y entramos a la del Encargado. Acá nos dirigimos a la pestaña “Notifications” y visualizamos la notificación del mensaje del Desarrollador indicando que la petición de cambio ha sido completada por el desarrollador Mary Jones, tal como lo muestra la figura N° 49

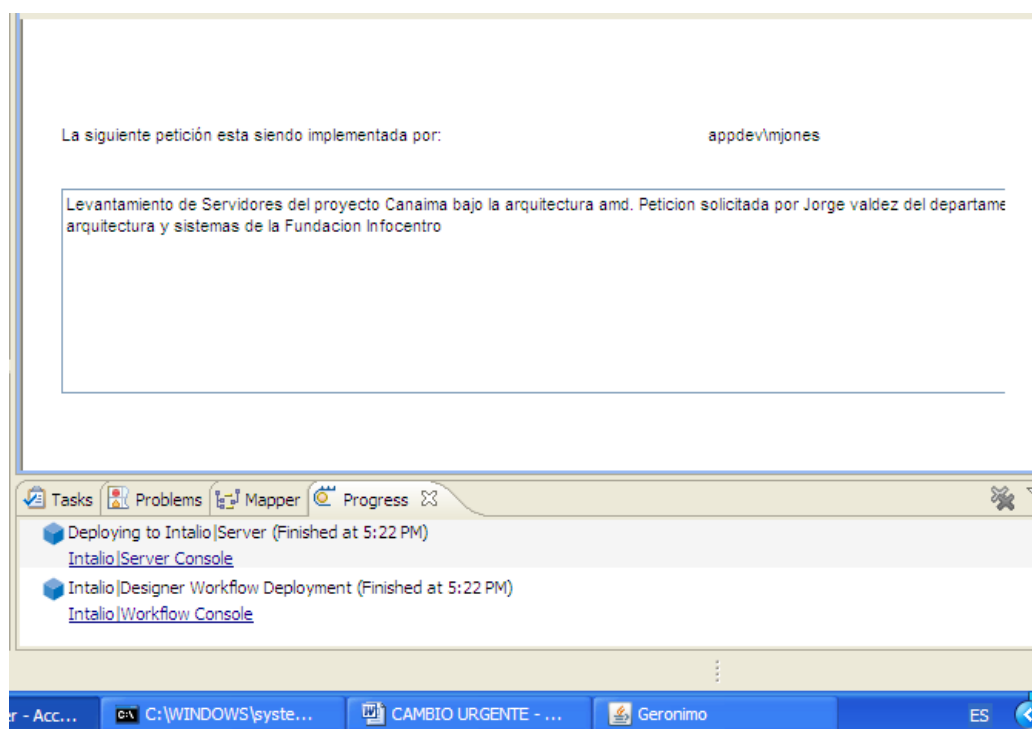


Figura N° 49. Notificación de implementación de cambio urgente

4.5.3 Clarificación de Petición de Cambio

Este es el flujo alternativo del proceso de Gestión de Cambios, que se da cuando el Desarrollador de versiones no puede calificar el cambio planteado como o estándar, debido a la poca información provista en la petición de cambio.

Este flujo comienza cuando el desarrollador, debe asignar un tiempo estimado para el cambio propuesto por el Cliente Iniciador. En este caso el Desarrollador le asigna, puede indicar a que otro desarrollador se le puede asignar la petición de cambio e indica cuales son los detalles faltantes en la petición de cambio original. La figura N° 50 indica este paso.

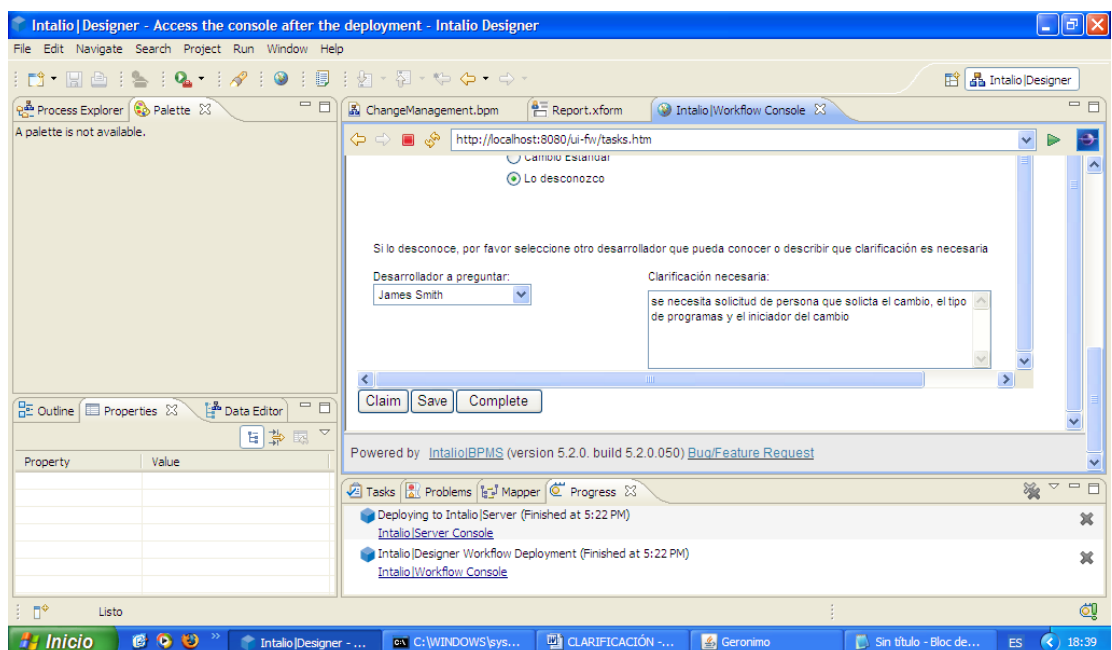


Figura N° 50. Desarrollador indicando que necesita mayor información en la petición

Luego de describir la petición necesaria le damos al botón "Complete", salimos de esta sesión y volvemos a loguearnos como el encargado del cambio, este acá revisa el reporte del desarrollador y señalar la última opción del formulario que indica "Clarificación por parte del cliente". Esto lo visualizamos en la figura N° 51.

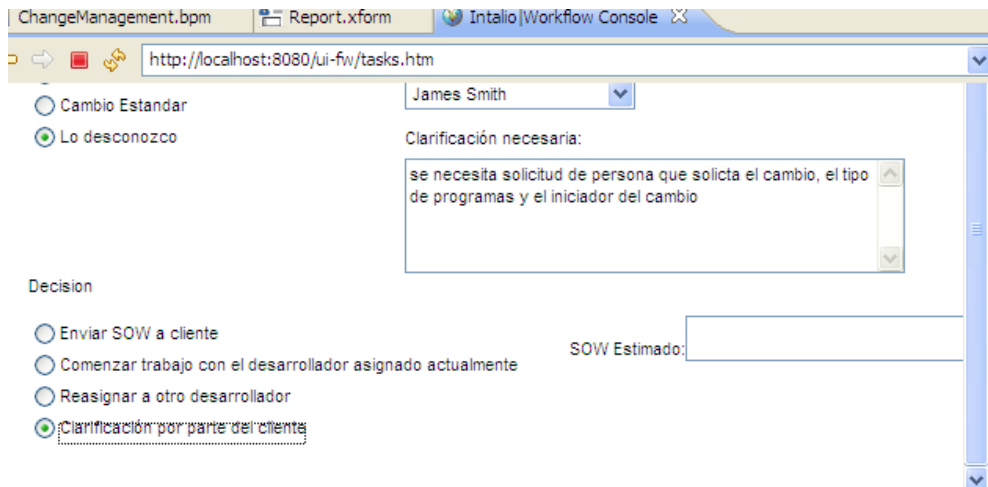


Figura Nº 51. Encargado del cambio le indica al Iniciador que el cambio necesita mayores detalles.

Salimos de esta sesión e ingresamos nuevamente como el cliente Iniciador. Ya acá aparece la petición de cambio con la misma fecha en que se envió indicando que deben agregarse mas detalles para luego ser clarificadas. En este punto el proceso puede ser retomado algunos de los dos flujos normales nombrados en los dos ítems anteriores.

Capítulo 5 - CONCLUSIONES

En muchas organizaciones los cambios son gestionados de manera reactiva y sin el control de un ente que los regule. Esto genera grandes problemas en la adaptabilidad de los procesos para cumplir sus objetivos de negocio.

El modelado del proceso de Gestión de Cambios ofrece la oportunidad de visualizar desde la perspectiva del analista de negocios el impacto de los cambios y así establecer criterios para mejorar la competitividad de la empresa.

Las empresas de hoy en día necesitan adaptar continua y rápidamente sus procesos de negocio para mantenerse competitivas. La flexibilidad necesaria en las empresas se puede lograr mediante el conjunto de prácticas conocidas como administración de procesos de negocio (BPM).

Durante el desarrollo de este Trabajo Especial de Grado la actividad de BPM en que se realizó el enfoque fue la gestión de procesos de negocio a través de flujos (*workflow*). Mediante un *workflow* se estudiaron los aspectos operacionales del proceso que es objeto de estudio. Esto es: cómo se estructuran las tareas, cómo se realizan, cual es su orden, cómo se sincronizan o la manera en que se relacionan unas con otras.

Para la adhesión del proceso de Gestión de Cambios al paradigma de BPM, fue necesario el uso de la suite BPM Intalio BPMS 5.2. La ventaja que proporcionó el uso de esta herramienta proviene de los beneficios que brinda su Arquitectura Orientada a Servicio (*SOA - Service Oriented Architecture*). SOA permitió reutilizar componentes de software, como los Web Services, para armar y modificar el proceso.

La integración de SOA con otras aplicaciones, como Tempo, vuelve a Intalio en una poderosa herramienta para el modelado flujos de trabajo, ejecución de procesos, manejo de tareas.

El manejo de tareas y flujos de procesos en Intalio, es gestionado por Tempo, a través de su Gestor de Tareas de Procesos que permite casi de manera invisible, la ejecución de actividades.

Intalio, bajo la perspectiva BPM, se basa en estándares tales como BPMN, XPD L y BPEL que permiten la portabilidad y visibilidad de los procesos a diferentes niveles en las organizaciones. Estos estándares permiten entender y diseñar la arquitectura de los procesos de manera más sencilla.

La combinación de los beneficios que ofrece el BPMS Intalio, junto al concepto de BPM, se alineó durante el desarrollo de de este Trabajo Especial de Grado con el código de mejores prácticas propuestas por ITIL.

ITIL fue utilizado como marco de referencia general para modelar el proceso de Gestión de Cambios ya que permite responder a todas las necesidades presentes y futuras de la organización, articulando diversos enfoques de una manera ágil, activa y medible.

El uso de UP Ágil como metodología para el desarrollo de software permitió extensibilidad; agregar o adaptar el sistema dependiendo de los requerimientos que iban surgiendo; proporcionó un enfoque en la arquitectura para minimizar los riesgos y organizar el desarrollo.

5.1 Recomendaciones y Trabajos Futuros

Se recomienda el modelado del proceso de Gestión de Cambios integrando Intalio BPMS 6.0 con herramientas que permitan mayor portabilidad como el

manejador de Base de Datos MySQL para manejar e implementar autenticación de usuarios; el servidor Apache DS que es mucho más rápido que el servidor Geronimo de la versión 5.2.

También se propone crear dos módulos para complementar el proceso de Gestión de Cambios modelado.

El primer debe permitir monitorear actividades de negocio (*BAM – Business Activity Monitoring*) y el manejo de inteligencia de negocios utilizando la suite *Open Source* como *Pentaho* que permite la creación de tableros de medición o dashboards.

El siguiente módulo, debe manejar reglas de negocios, por lo tanto se propone el uso de JBOSS como motor de reglas de negocio.

Para finalizar, sería conveniente modelar e integrar los diferentes procesos propuestos por la Biblioteca de Infraestructura de Tecnologías de Información, utilizando Intalio BPMS 6.0

REFERENCIAS BIBLIOGRÁFICAS

Acevedo, J. (2007). *Criterios de selección para las Herramientas de Orquestación de Servicios Web*. Santiago De Chile. Chile

Alcantara, J. (2007). *Análisis de Brecha tecnológica asociada a cada proceso ITIL*. Chile.

Ambler, S. (2007). *Agile Adoption Survey*. New York, USA.

Barrios, J; Montilva, J. y Rivero, D. (2004). *Ontología para el desarrollo de software en OWL*. Nueva Segovia – Nicaragua.

Bonillo, P. (2009). *Metodología para la Gestión de Procesos de Negocio sustentada en el uso de patrones*. Caracas – Venezuela.

Booth, D; Haas, H; Maccabe, F; Newcomer, E y Champion, M (2004). *Web Services Architecture*. USA.

Brito, J. (2008). *ITSM y su valor en las organizaciones*. 2008. Florida – USA

Caituro, M.; Rodriguez; M. (2004). *Framework for Autonomic Web Services Collaboration, Orchestration and Choreography in EGovernment Information Systems*. Brookfield, Illinois, USA.

Cartlidge, A. (2007). *Introductory overview of itil v3*. USA.

Cavalho, R. (2005). *Desarrollo de software. Un enfoque a metodologías prácticas y modernas*. Uruguay.

Chatain, T. y Jard, J. (2009). *Models for the Supervision of Web Services Orchestration with Dynamic Changes*. UK.

Chávez, V (2008). *Trabajo Especial de Grado para elaborar un sistema para la Administración de los Procesos de Soporte a los Servicios de ITIL usando la metodología Process Management*. Caracas. Venezuela.

Clinch, J. (2004). *The Future of ITIL*. United Kingdom.

Felipe, E. (2008). *Elementos de un BPMS*. Buenos Aires – Argentina.

Felipe, E (2009). *El arte del modelado de proceso de negocios ejecutables*. Buenos Aires – Argentina

Kotelnikov, V. (2008). *Synergizing Business Processes*. USA

Madariaga, B. (2009). *Aplicación de las TIC basadas en fuentes abiertas*. Colombia.

Marin, D. (2007). *Trabajo Especial de Grado sobre el Desarrollo de un prototipo del proceso de soporte de servicios gestión de cambios, de la librería de infraestructura de tecnologías de la información*. Caracas. Venezuela.

Madisson, D. (1999). *Metodología de desarrollo de software*. USA

- Medjahed, J.; Bouguettay, B; Elmagarmid, V. (2003). *Composing Web services on the Semantic Web*. USA
- Mejia, M. y Arzate, L. (2007). *Automatización de Procesos de Negocio utilizando un BPMS*. México, D.F.
- Obach-Renner, C. (2007), *Un enfoque pragmático de Arquitectura Orientada a Servicios (SOA) para Integración de Aplicaciones Empresariales (EAI)*. [Documento en línea]. Disponible en: <http://gurulab-soar.googlecode.com/files/SOAr-es-v1.1b1.pdf>. Extraído el 07 de septiembre de 2009.
- Organization for the Advancement of Structured Information Standards. UDDI (*Universal Description, Discovery, and Integration*). [Documento en línea]. Disponible en: www.oasis-open.org/committees/uddi-spec/doc/tcpspecs.htm. Extraído el 27 de septiembre de 2009.
- Owen, M. y Raj, J. (2003). *BPMN and Business Process Management. Introduction to the New Business Process Modeling Standard*. Acton-Boxborough Regional High School. Massachusetts, USA.
- Peltz, C. (2003). *Web Services Orchestration. A review of emerging technologies, tools and standards*. USA
- Pin, E; Marrero, Y. (2006). *Indicadores, una herramienta para medir la Eficiencia en el uso de las Tecnologías de la Información y las Comunicaciones en los Centros de Educación Superior*. Universidad de La Habana. La Habana - Cuba. 2006
- Piattini, M. y De Miguel, E. (2001). *Metodología para el diseño y desarrollo de software*. Editorial Ra-ma. Uruguay.
- Ruiz, A (2007). *Modelado de Procesos de Negocios basado en Ontologías*. San José. Costa Rica.
- Ramirez, T. (1999). *Como hacer un proyecto de investigación*. Editorial Panapo. Caracas. Venezuela
- Rodríguez; G. (1996). *Metodología de la Investigación Cualitativa*, México DF – México.
- Sabino, D. (1992). *Hipótesis en la investigación*. Editorial Diaspora. Valencia. España.
- Smith, H. y Fingar, P. (2007). *Business Process Management: The Third Wave*. USA.
- Tamayo, R. (2000). *Proyectos modernos de investigación*. Caracas - Venezuela
- The World Wide Web Consortium. *Web Services*. [Documento en línea]. Disponible en: <http://www.w3.org/TR/ws-arch/>. Extraído el 06 de agosto del 2009.
- Torres, C. y Alférez, G. (2008). *Establecimiento de una Metodología de Desarrollo de Software para la Universidad de Navojoa*. México.
- Van der Aalst, R. (2004). *Business Process Management System*. Amsterdam – Holanda
- Van de Putte, J. (2001). *Definición de Procesos de Negocios Seguros basados en la visión de BPM*. USA.

Van Haren, T. (2002). *BPM's. New perspective organizacional*. USA

Vinogradsky, V. (2007). *Eight Steps to Effective IT Change Management*. Saint Paul University. Ottawa – Canadá

Well, H. (2008). *Análisis y modelado de procesos de negocio*. University of Iowa, USA.

Wilkes, L. (2006). *The Web Services Protocol Stack en CBDI Web Services Roadmap - Guiding the Transition to Web Service*. Oklahoma City – USA.

Sleeper, B. (2004). *The five missing pieces of SOA*. USA

Workflow Management Coalition (1999). *Business Process*. USA. [Documento en línea]. Disponible en: www.wfmc.org/pr/WfMC_BPMI_Joint_Meeting.pdf. Extraído el 14 de agosto de 2009.

Zayas de Glynx, A. (Mayo 2007). *La Importancia del Manejo de TI en los Sistemas de Gestión*. México.

ANEXO A – Diagramas UML

Casos de Uso

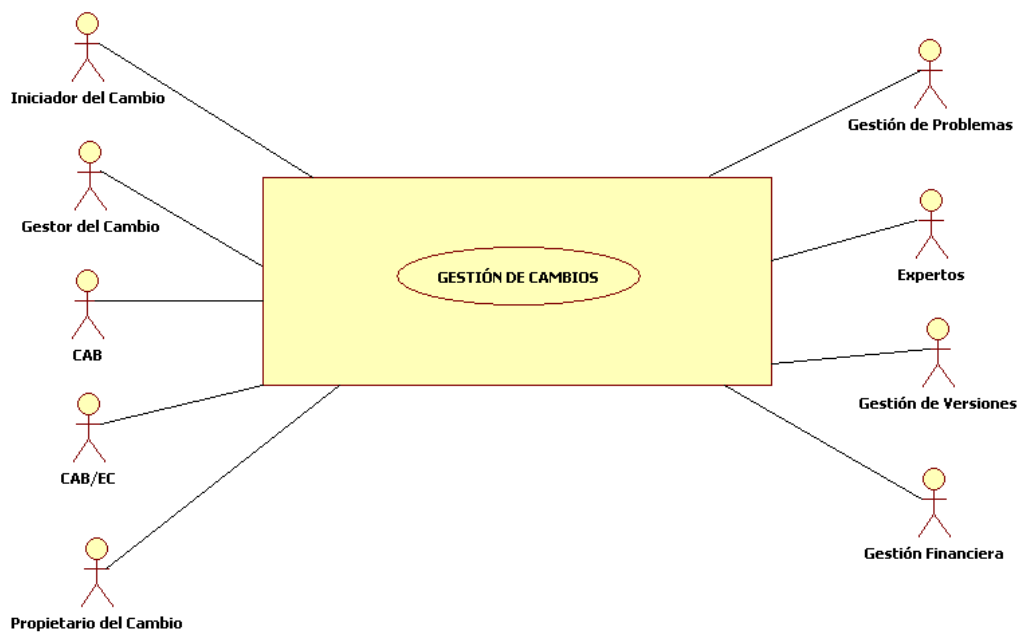


Figura A-1.Diagrama de Caso de Uso General del Proceso de Gestión de Cambios, Nivel 0

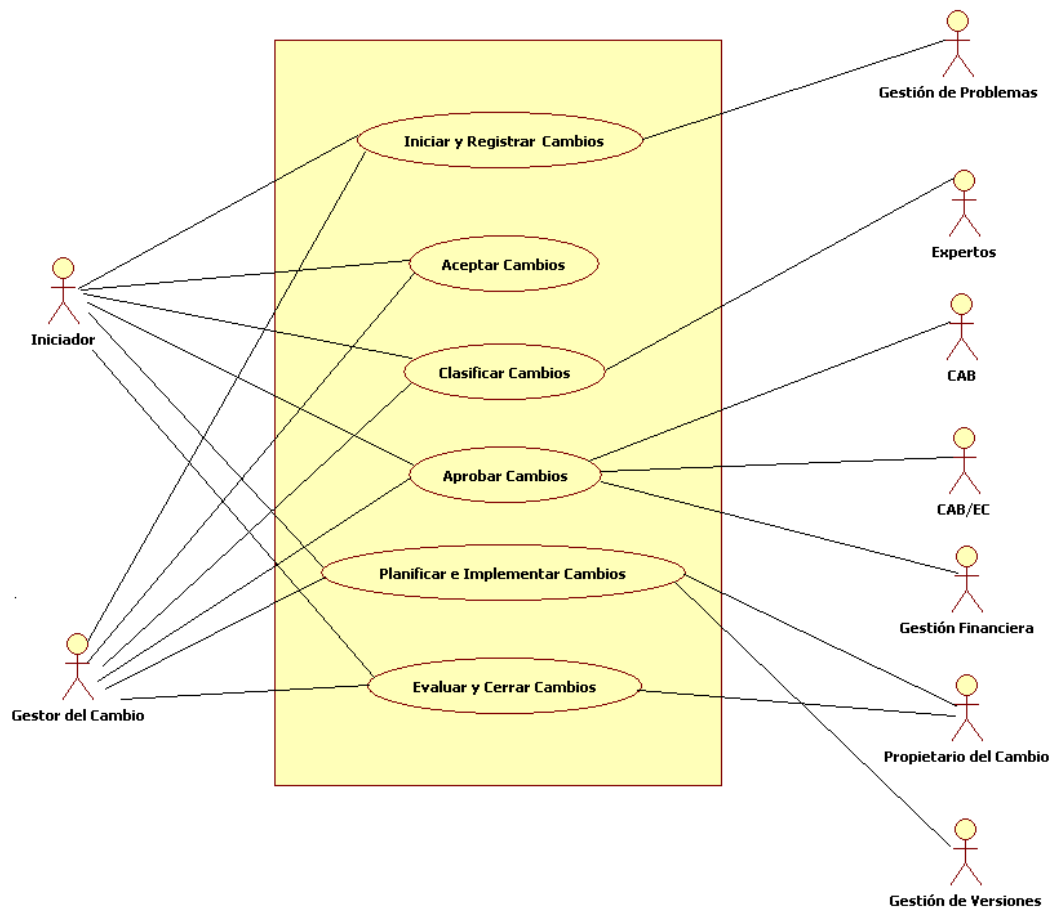


Figura A-2.Diagrama de Caso de Uso del Proceso de Gestión de Cambios, Nivel 1

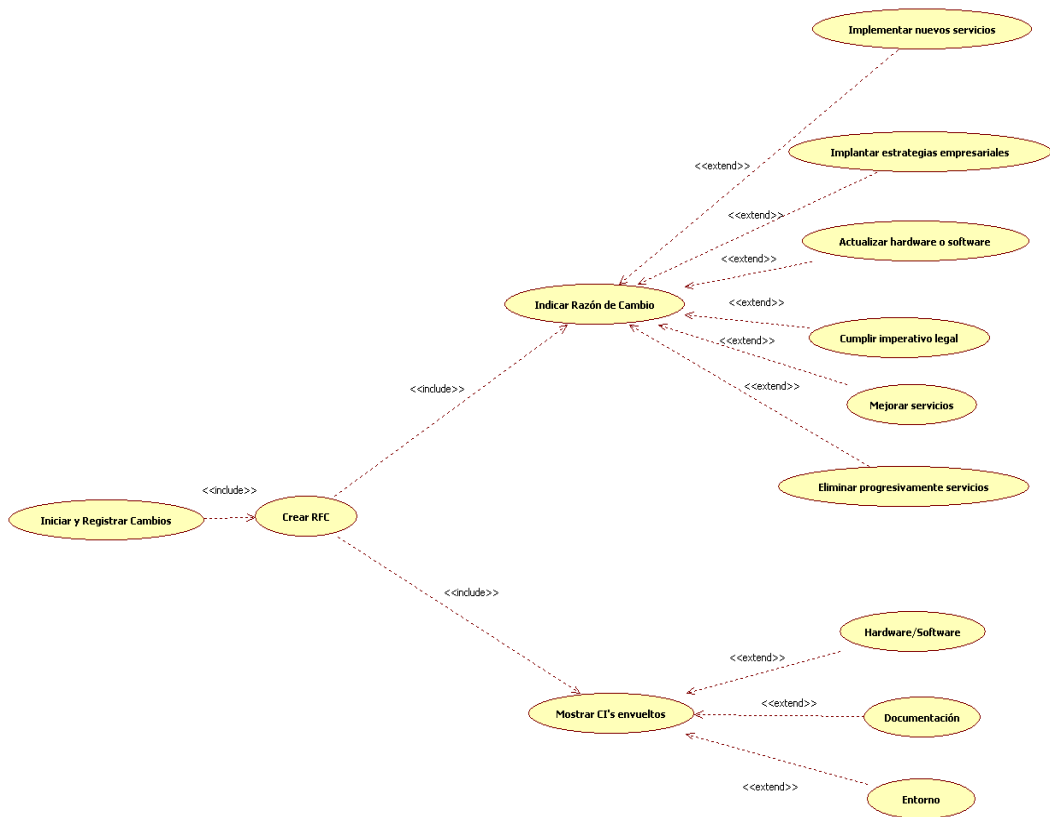


Figura A-3. Diagrama de Caso de Uso Iniciar y Registrar Cambios, Nivel 2

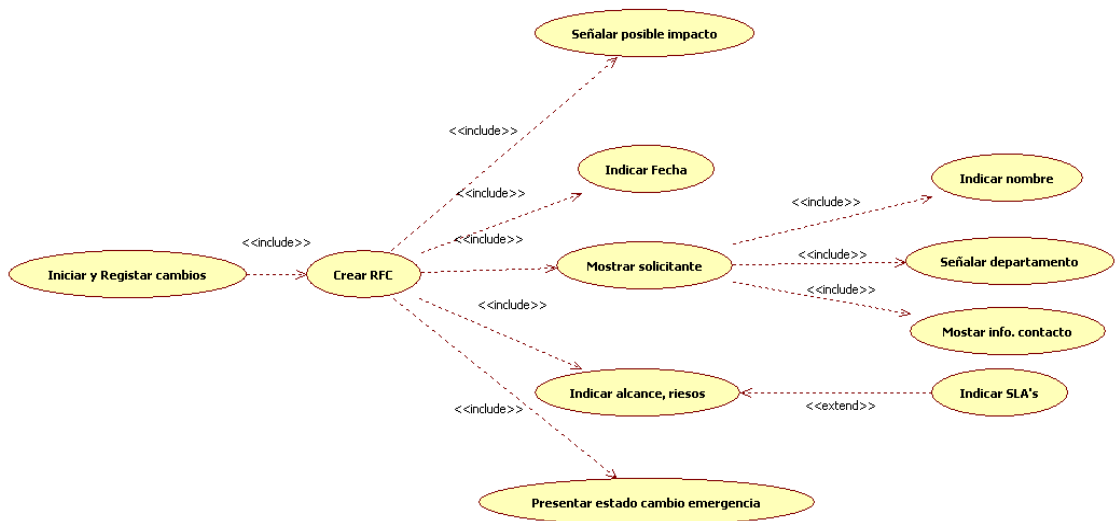


Figura A-4. Diagrama de Caso de Uso Iniciar y Registrar Cambios, Nivel 2 (Continuación)

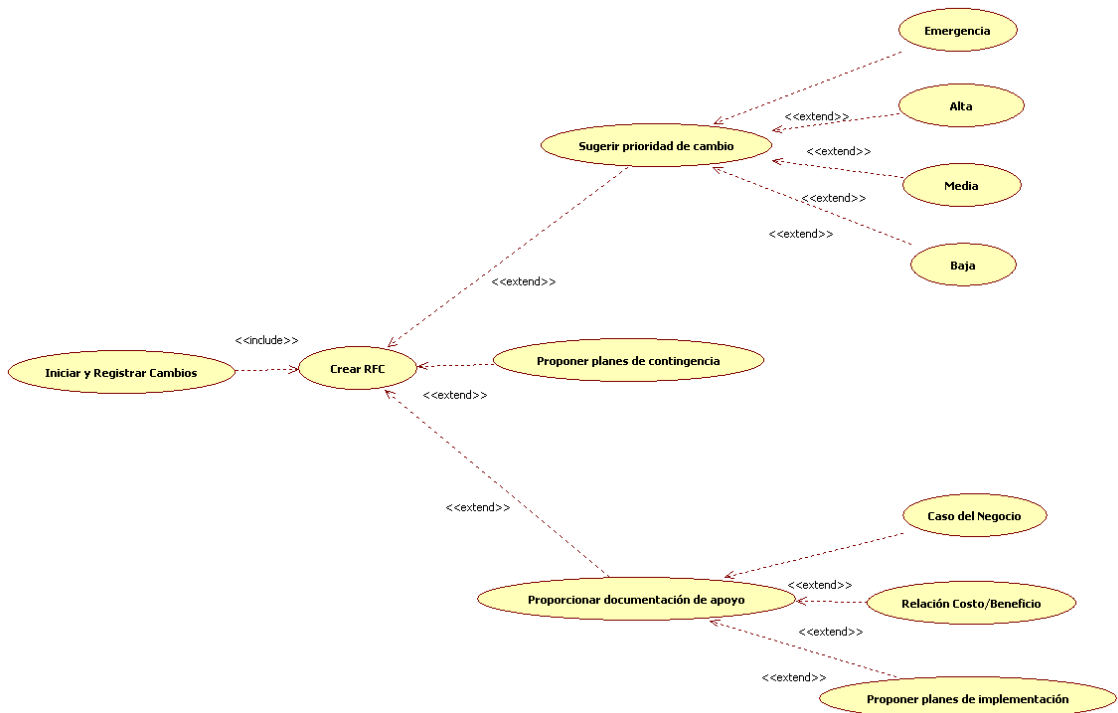


Figura A-5. Diagrama de Caso de Uso Iniciar y Registrar Cambios, Nivel 2 (Continuación)

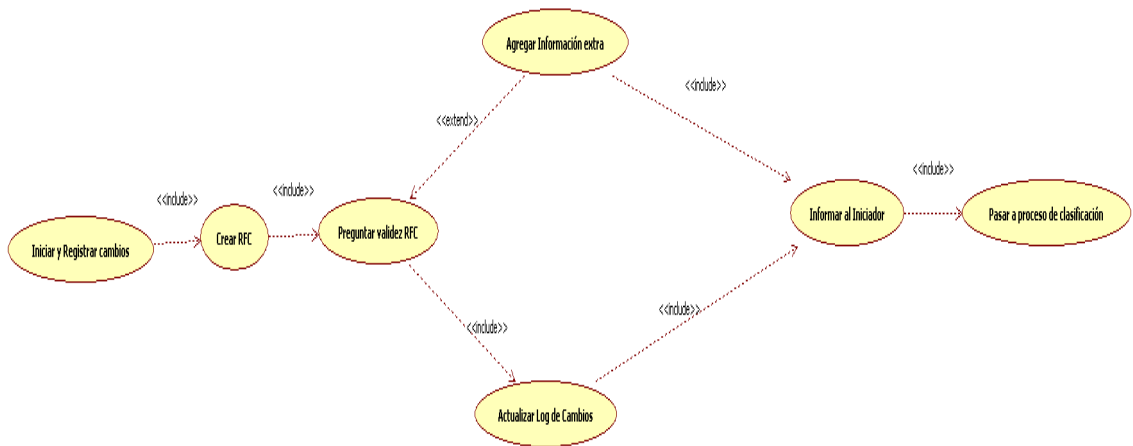


Figura A-6. Diagrama de Caso de Uso Iniciar y Registrar Cambios, Nivel 2 (Continuación)

Descripción del Caso de Uso Iniciar y Registrar Cambios

Nombre Caso de Uso:	1. Iniciar y Registrar Cambios.
Actor(es) Principal(es):	Iniciador de Cambio, Gestor de Cambios
Actor(es) Secundario(s):	Gestión de Problemas
Descripción:	Este caso de uso indica los elementos necesarios para crear una petición de cambio y la manera en que será validado
Precondiciones:	Debe existir la necesidad de un cambio de hardware; software; base de datos; actualización, mejora o eliminación de servicio en un departamento de la organización
Postcondiciones:	Existe un RFC completamente validado
Escenario de Éxito. Curso Básico	El solicitante debe registrar en un RFC el cambio deseado y el Gestor de Cambios debe validarlo de manera tal que la petición pueda ser escalado al proceso de clasificación de cambios
Extensiones (Cursos Alternos)	Sugerir prioridad del cambio, proponer planes de contingencia y proporcionar documentos de apoyo

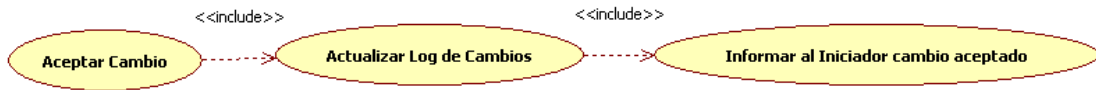


Diagrama de Caso de Uso Aceptar Cambio, Nivel 2

Figura A-7.Descripción del Caso del Uso Aceptar Cambios

Nombre Caso de Uso:	1. Aceptar Cambios.
Actor(es) Principal(es):	Gestor de Cambios
Actor(es) Secundario(s):	Iniciador de Cambios
Descripción:	Este caso de uso indica que la RFC fue aceptada para su revisión
Precondiciones:	El RFC debió haber sido correctamente validado previamente por el Gestor de Cambios
Postcondiciones:	El Iniciador debe tener seguridad que su RFC fue atendido
Escenario de Éxito. Curso Básico	El Gestor del Cambio acepta el RFC para su revisión y le indica al Iniciador del Cambio que su RFC ha sido atendido
Extensiones (Cursos Alternos)	No aplica

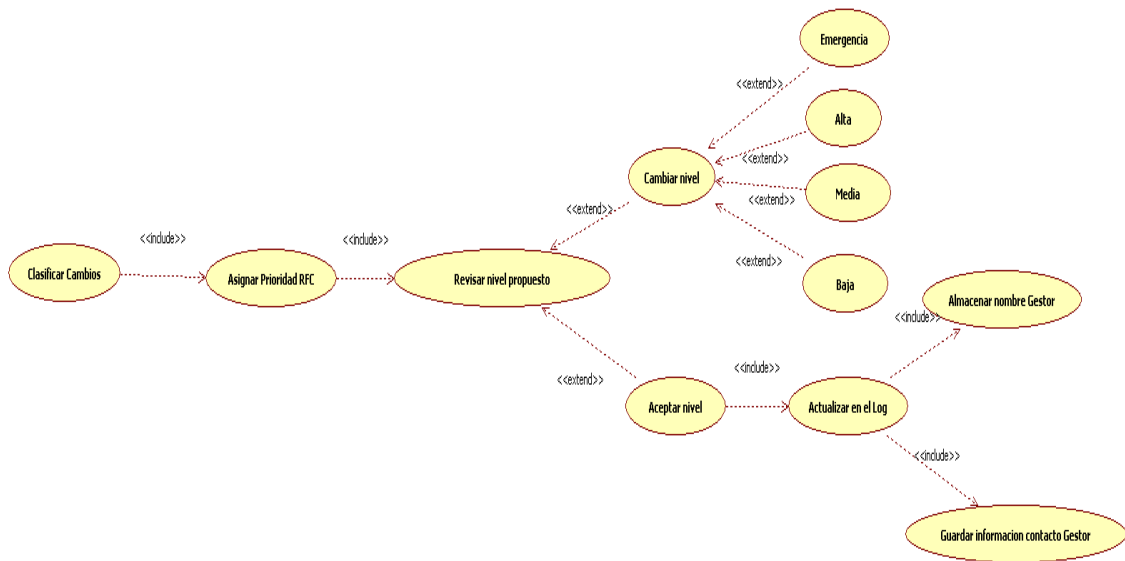


Figura A-8.Diagrama de Caso de Uso Clasificar Cambio, Nivel 2

Descripción del Caso del Uso Clasificar Cambios

Nombre Caso de Uso:	3. Clasificar Cambio.
Actor(es) Principal(es):	Gestor de Cambios
Actor(es) Secundario(s):	Iniciador de Cambios , Expertos en el tema
Descripción:	Este caso de uso señala como será priorizado y categorizado la petición de cambio, dependiendo del impacto, los riesgos y la emergencia con que este indicado.
Precondiciones:	El RFC debió haber sido correctamente validado previamente por el Gestor de Cambios
Postcondiciones:	La petición de cambio debe estar clasificada correctamente para poder ser llevada al proceso de aprobación
	El Gestor del Cambio revisa la prioridad y la categoría

Escenario de Éxito. Curso Básico	propuesta por el Iniciador del Cambio, si esta de acuerdo con ello no realiza ningún . Si no esta de acuerdo con la prioridad y categoría propuesta, puede modificarlos.
Extensiones (Cursos Alternos)	El Gestor de Cambio pudiera no estar totalmente seguro de que prioridad y que categoría asignarle a la RFC y puede contactar expertos en el tema para que lo asista en la clasificación del cambio

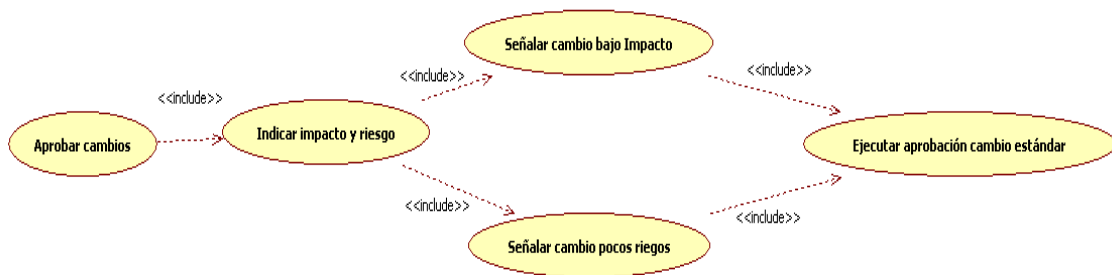


Figura A-9.Diagrama de Caso de Uso Aprobar cambio Estándar, Nivel 2

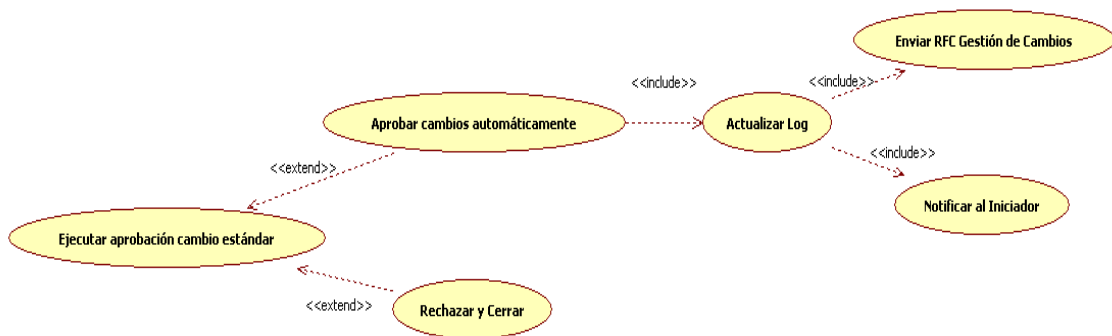


Figura A-10.Diagrama de Caso de Uso Aprobar cambio Estándar, Nivel 2 (Continuación)

Descripción del Caso de Uso Aprobar Cambio estándar

Nombre Caso de Uso:	4.1 Aprobar Cambios estándares.
Actor(es) Principal(es):	Gestor de Cambios
Actor(es) Secundario(s):	Iniciador de Cambios
Descripción:	Este caso de uso indica la manera en que se aprueba un cambio que ha sido determinado como un cambio estándar
Precondiciones:	La RFC debe haber sido indicada como un cambio de bajo impacto y pocos riesgos
Postcondiciones:	El cambio debe haber sido aprobado por el Gestor del Cambio
Escenario de Éxito. Curso Básico	La RFC debe estar indicada como un cambio estándar para poder ser aprobada por el Gestor de Cambios, el cual lo aprobará de manera inmediata.
Extensiones (Cursos Alternos)	No aplica



Figura A-11. Diagrama de Caso de Uso Aprobar cambio de emergencia, Nivel 2

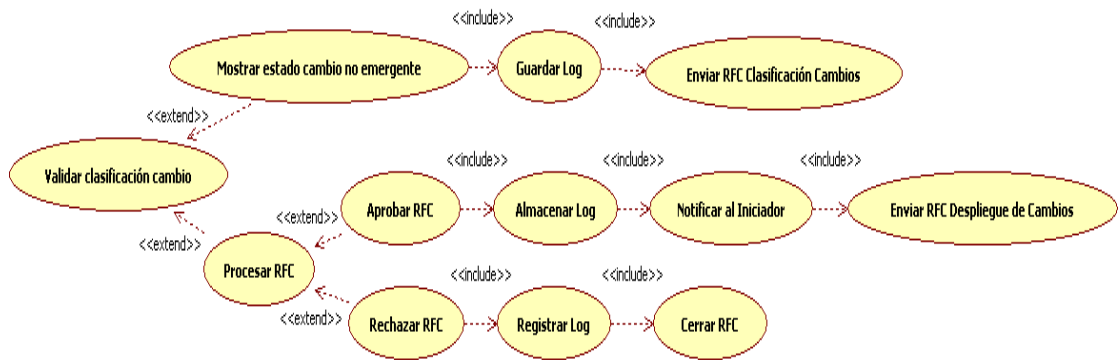


Figura A-12.Diagrama de Caso de Uso Aprobar cambio de emergencia, Nivel 2 (Continuación)

Descripción del Caso de Uso Aprobar Cambio de emergencia

Nombre Caso de Uso:	4.2 Aprobar Cambios de emergencia
Actor(es) Principal(es):	Gestor de Cambios
Actor(es) Secundario(s):	Iniciador de Cambios
Descripción:	Este caso de uso indica la manera en que se aprueba un cambio que ha sido determinado como un cambio de emergencia
Precondiciones:	El estado de la RFC debe haber sido indicado como un cambio de emergencia
Postcondiciones:	No aplica
Escenario de Éxito. Curso Básico	La RFC debe estar indicada como un cambio de emergencia para poder ser aprobada.
Extensiones (Cursos Alternos)	No aplica

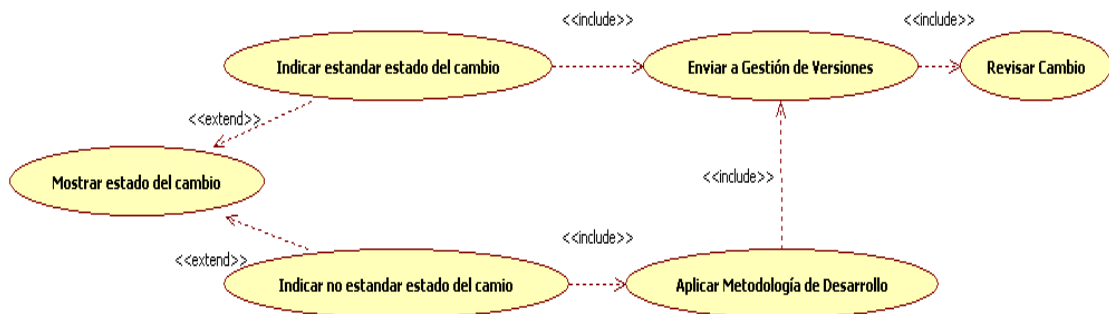


Figura A-13.Diagrama de Caso de Uso Implementar cambios, Nivel 2

Descripción del Caso de Uso Implementar Cambios

Nombre Caso de Uso:	Implementar Cambios
Actor(es) Principal(es):	Gestor de Versiones
Actor(es) Secundario(s):	Iniciador de Cambios, Gestor De Cambios
Descripción:	Este caso de uso indica la manera en que se planificar e implementar el cambio propuesto. En este caso la implementación es realizada por el proceso de Gestión de Versiones.
Precondiciones:	El cambio ha debido ser aprobado
Postcondiciones:	El cambio ha sido implementado correctamente en el departamento que lo solicitaba
Escenario de Éxito. Curso Básico	El cambio aprobado debe ser planificado por el Gestor de Cambio, Luego es enviado a la Gestión de Versiones para su despliegue según lo establecido
Extensiones (Cursos Alternos)	No aplica

Diagramas de Secuencia

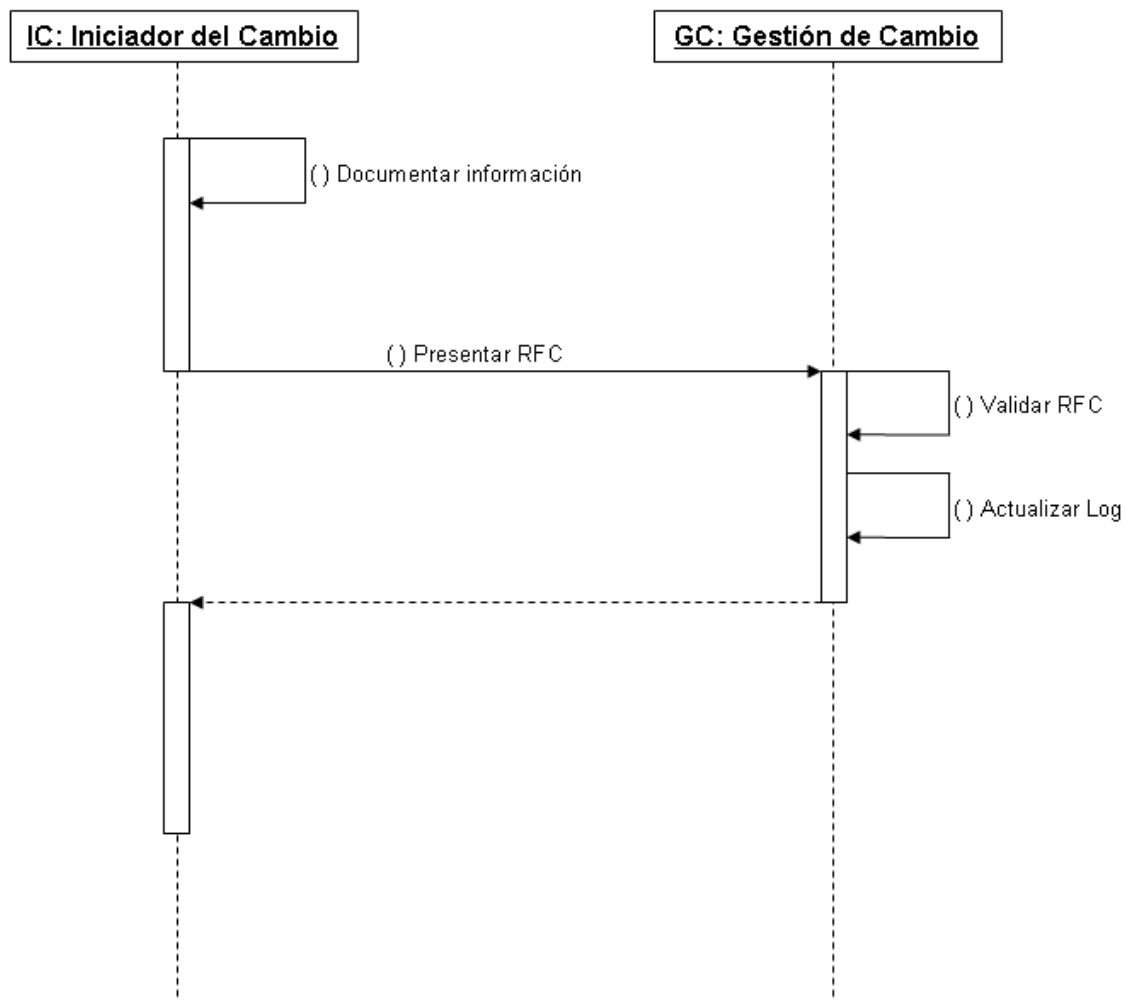


Figura A-14.Diagrama de Secuencia del proceso Iniciar y Registrar Cambios

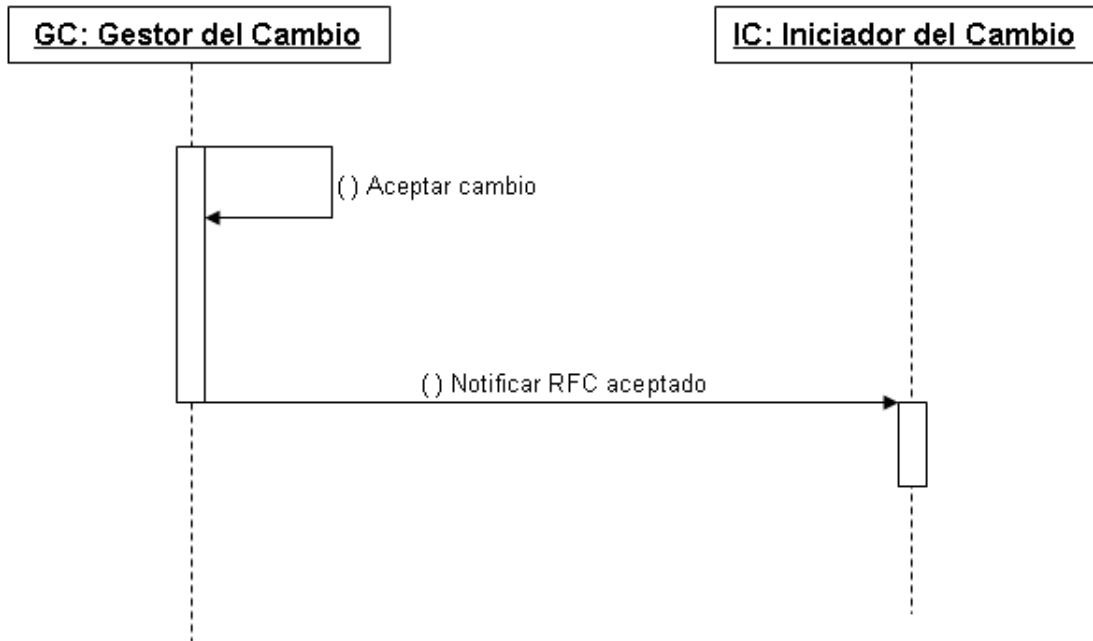


Figura A-15.Diagrama de Secuencia del proceso Aceptar Cambios

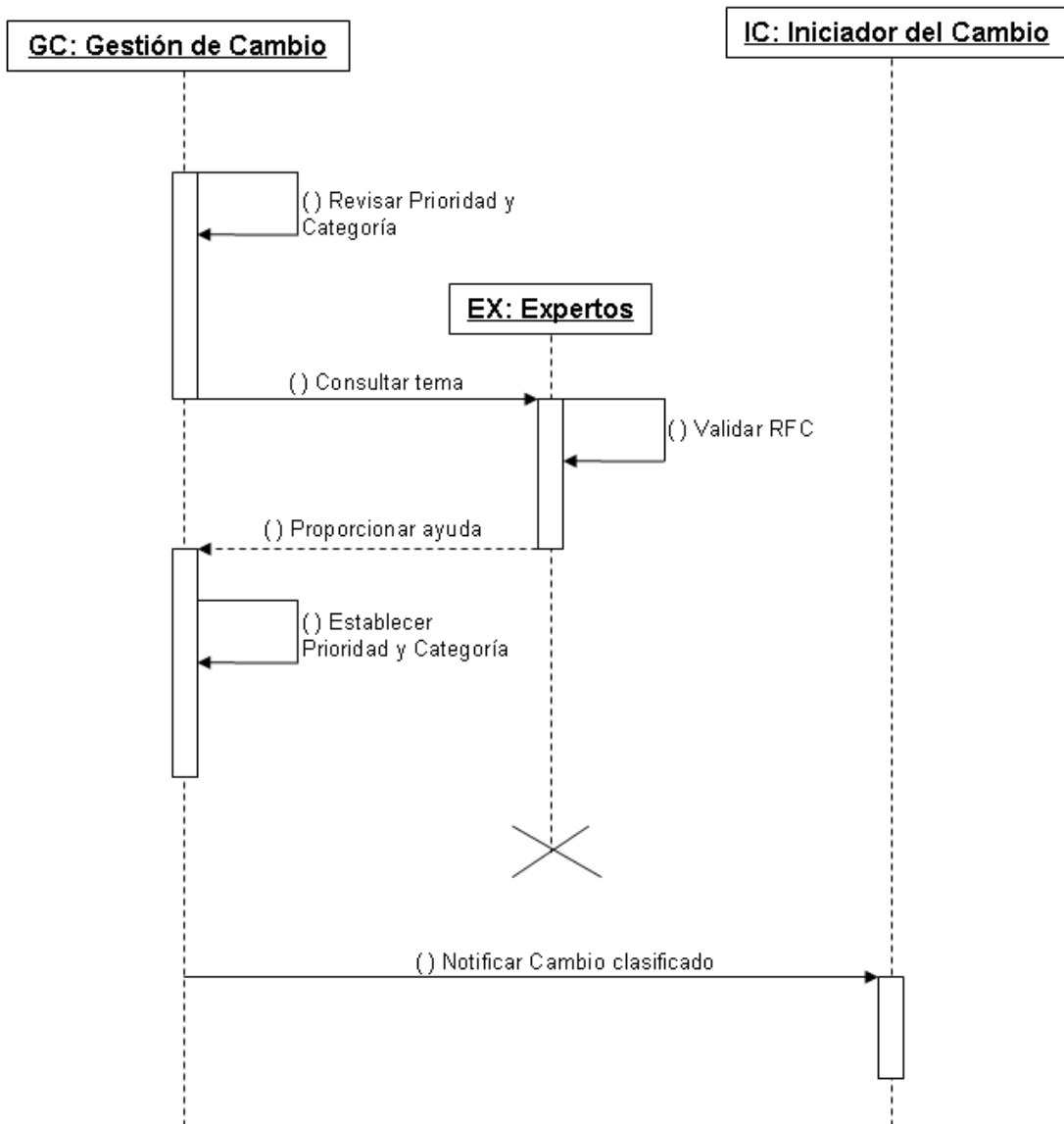


Figura A-16. Diagrama de Secuencia del proceso Clasificar Cambios

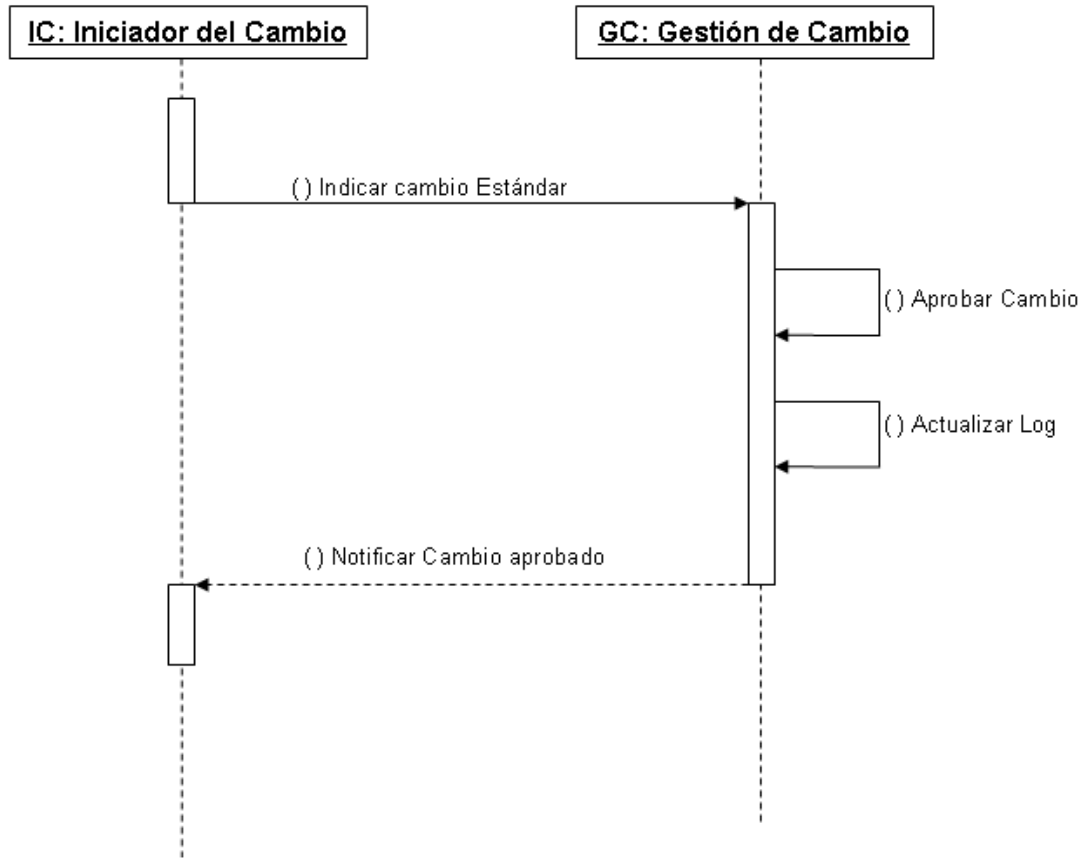


Figura A-17.Diagrama de Secuencia del proceso Aprobar Cambio Estándar

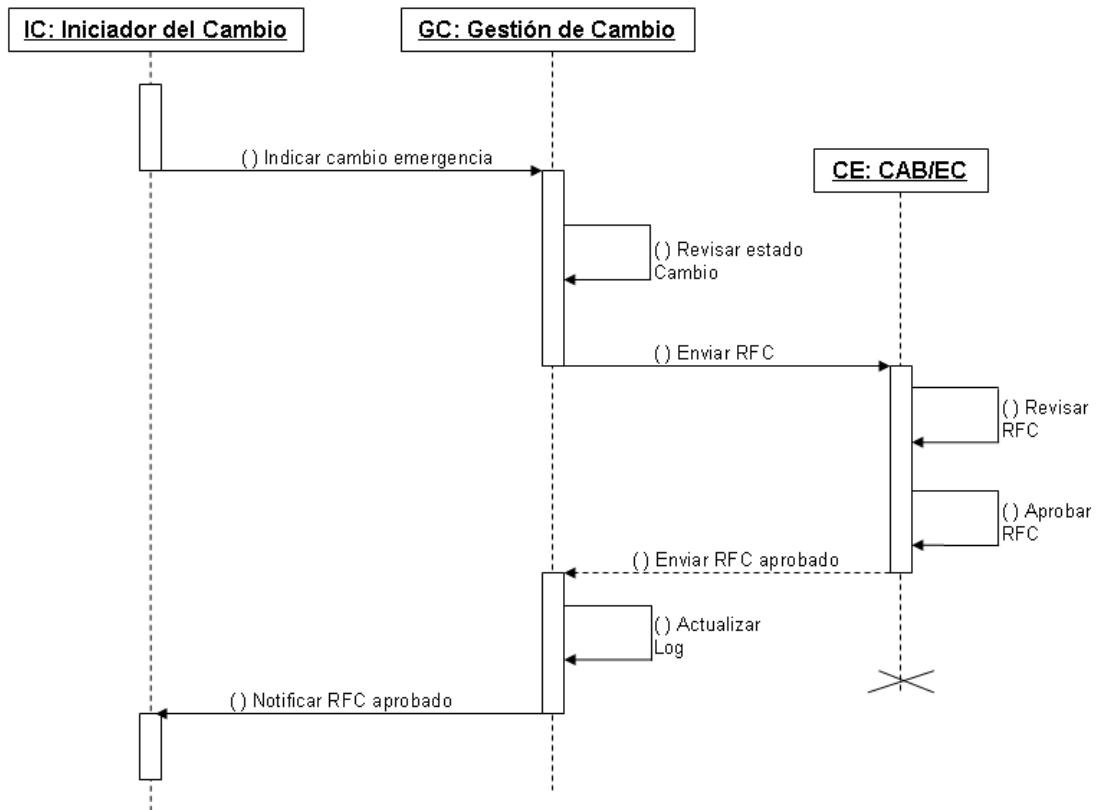


Figura A-18.Diagrama de Secuencia del proceso Aprobar Cambio de Emergencia

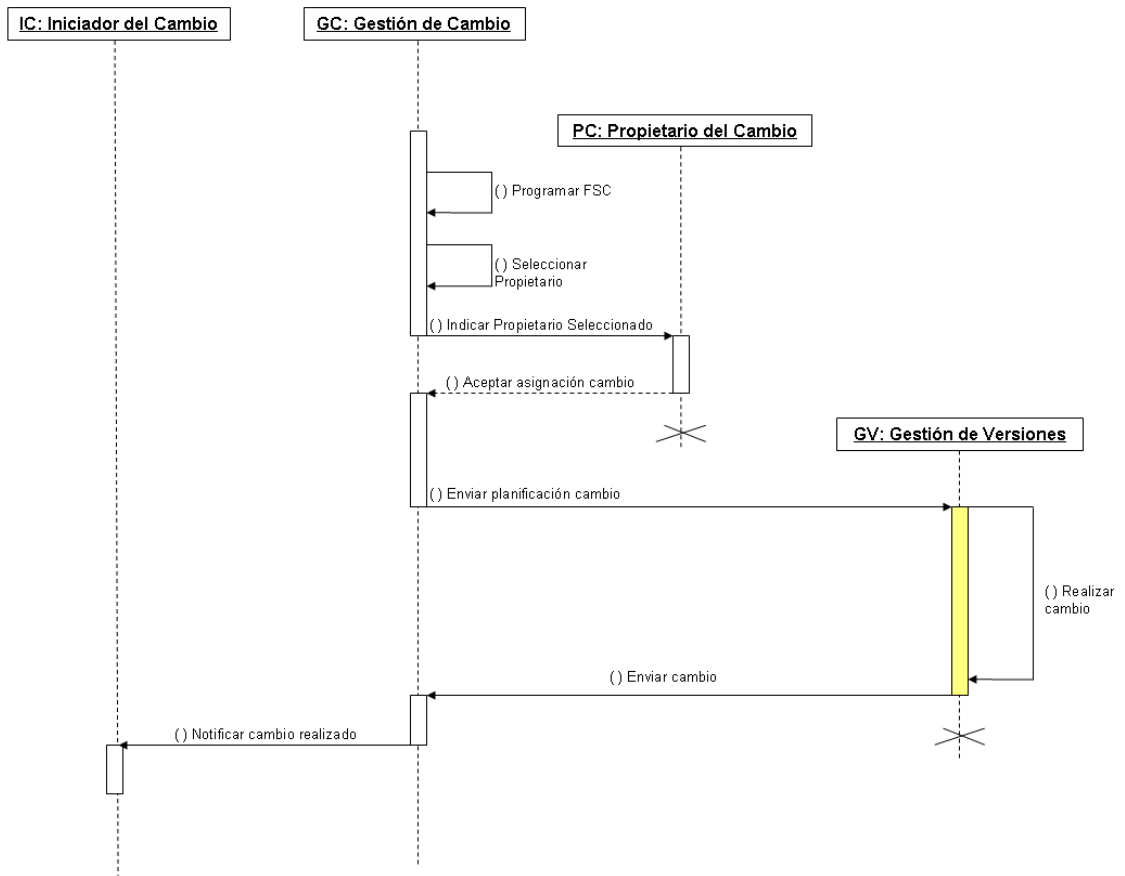


Figura A-19.Diagrama de Secuencia del proceso planificar e implementar cambios

Diagramas de Colaboración

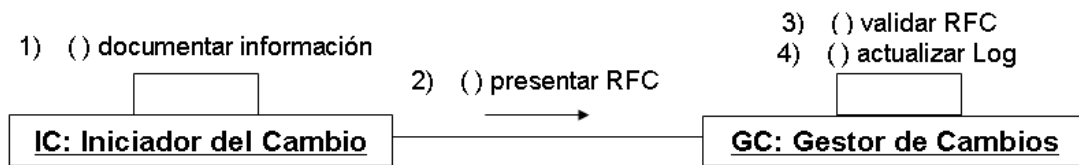


Figura A-20. Diagrama de Colaboración del proceso iniciar y registrar cambios



Figura A-21. Diagrama de Colaboración del proceso aceptar cambios

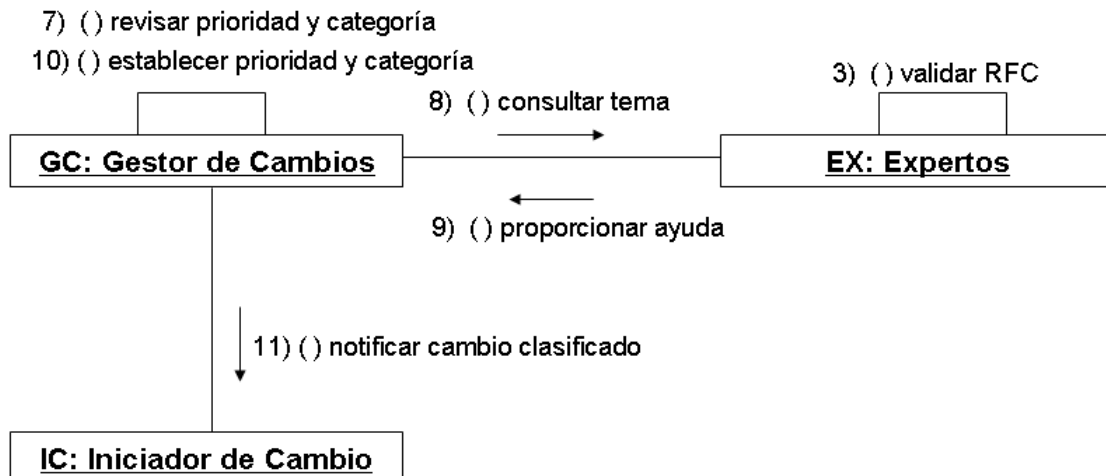


Figura A-22. Diagrama de Colaboración del proceso clasificar cambios

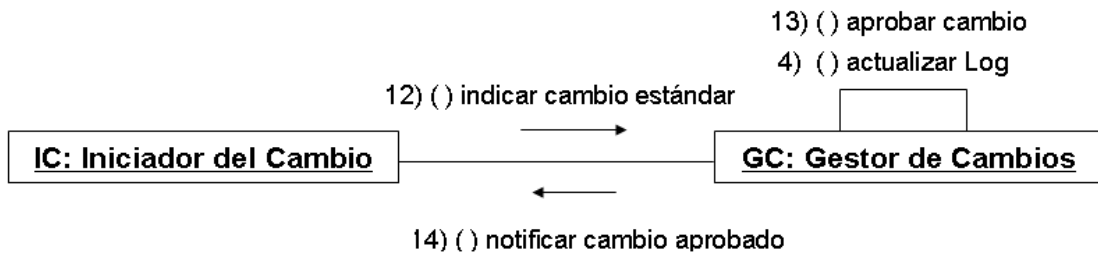


Figura A-23. Diagrama de Colaboración del proceso aprobar cambios estándares

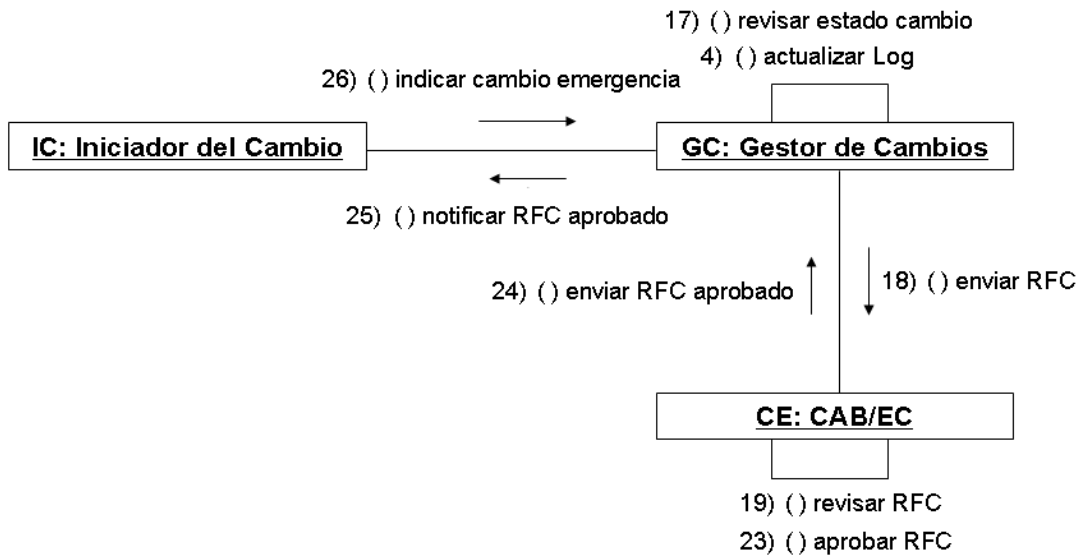


Figura A-24. Diagrama de Colaboración del proceso aprobar cambios de emergencia

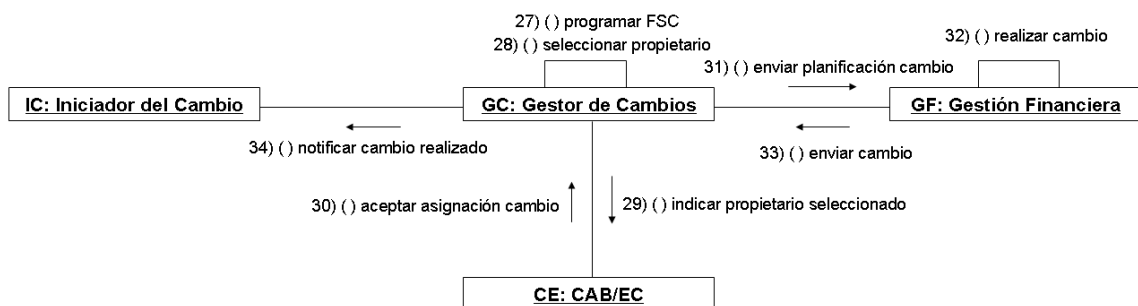


Figura A-25. Diagrama de Colaboración del proceso planificar e implementar cambios

Diagramas de Estado

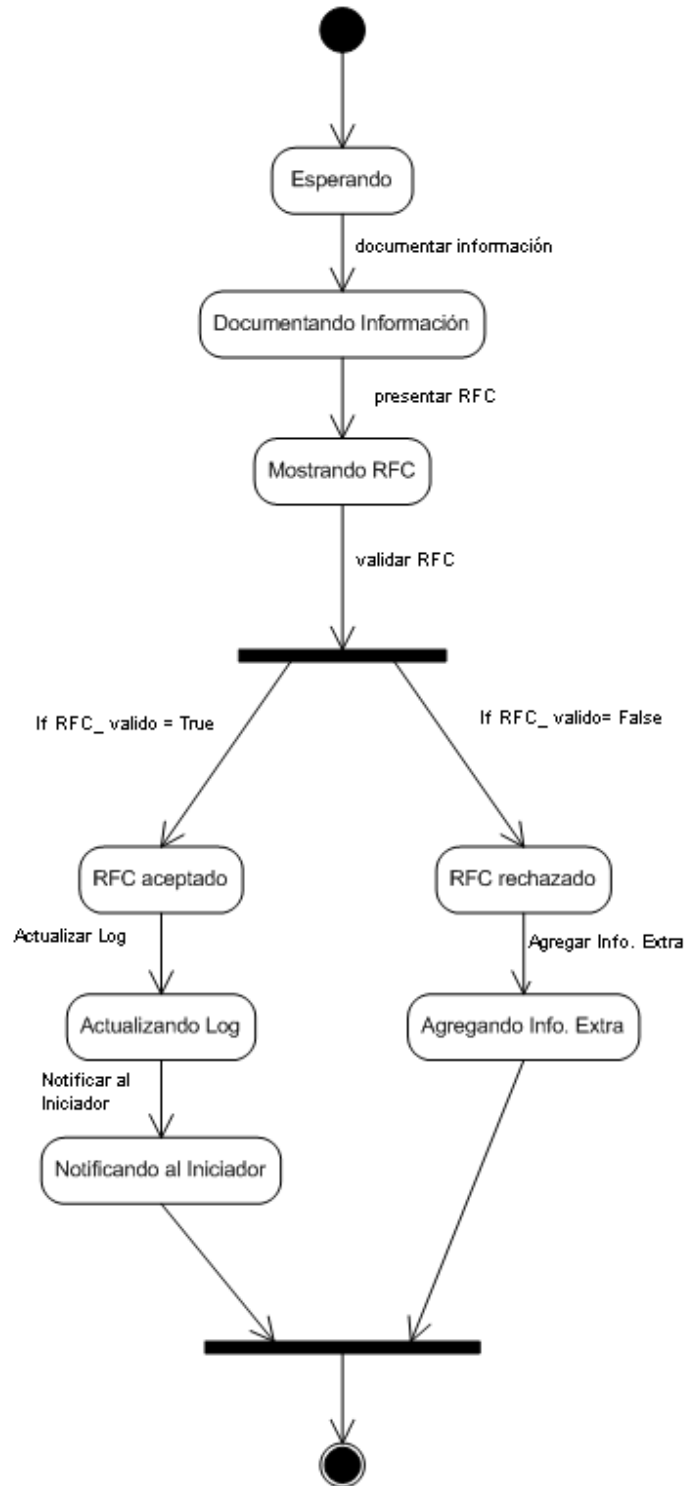


Figura A-26. Diagrama de Estado del proceso iniciar y registrar cambios

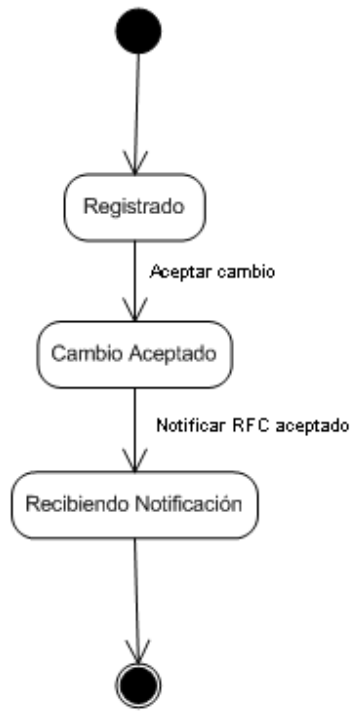


Figura A-26.Diagrama de Estado del proceso aceptar cambios



Figura A-27. Diagrama de Estado del proceso clasificar cambios según categoría

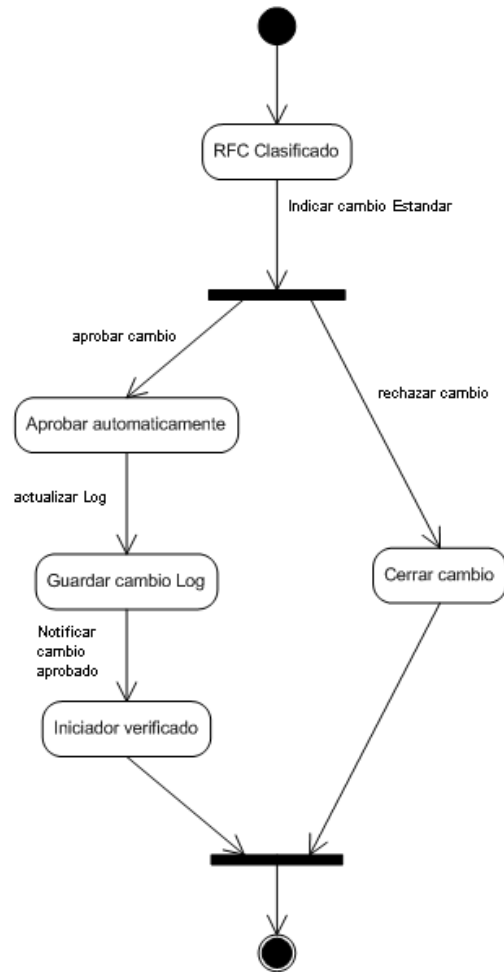


Figura A-28.Diagrama de Estado del proceso aprobar cambios estándar

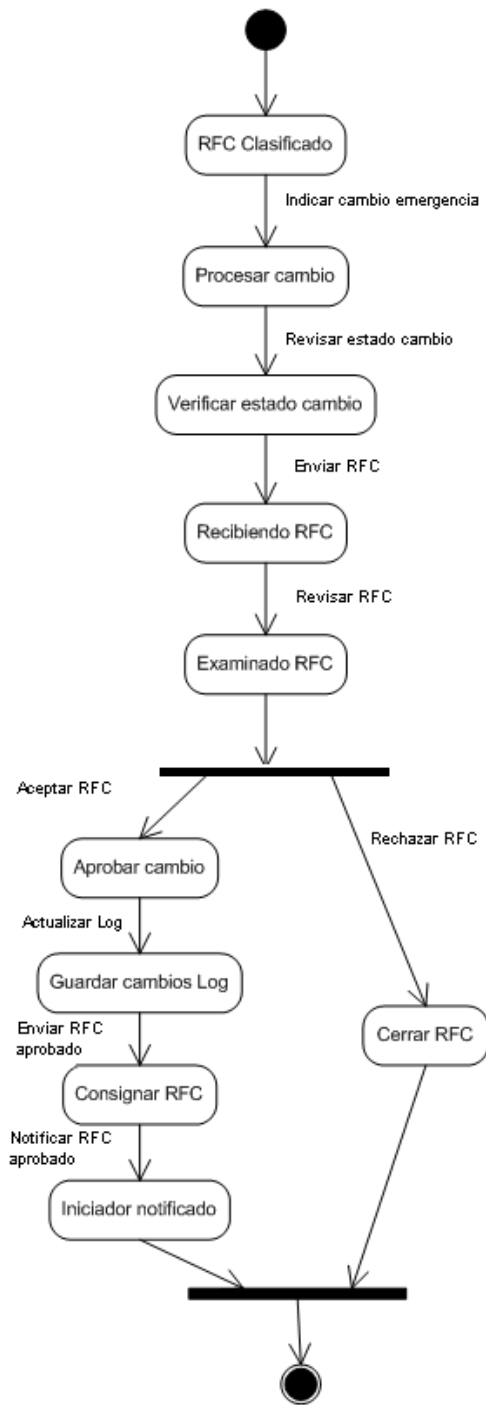


Figura A-29.Diagrama de Estado del proceso aprobar cambios de emergencia

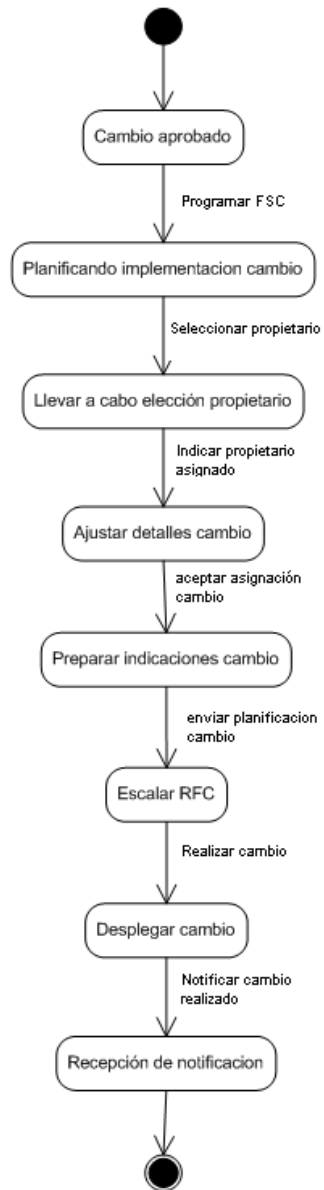


Figura A-30.Diagrama de Estado del proceso planificar e implementar cambios

Diagramas de Actividad

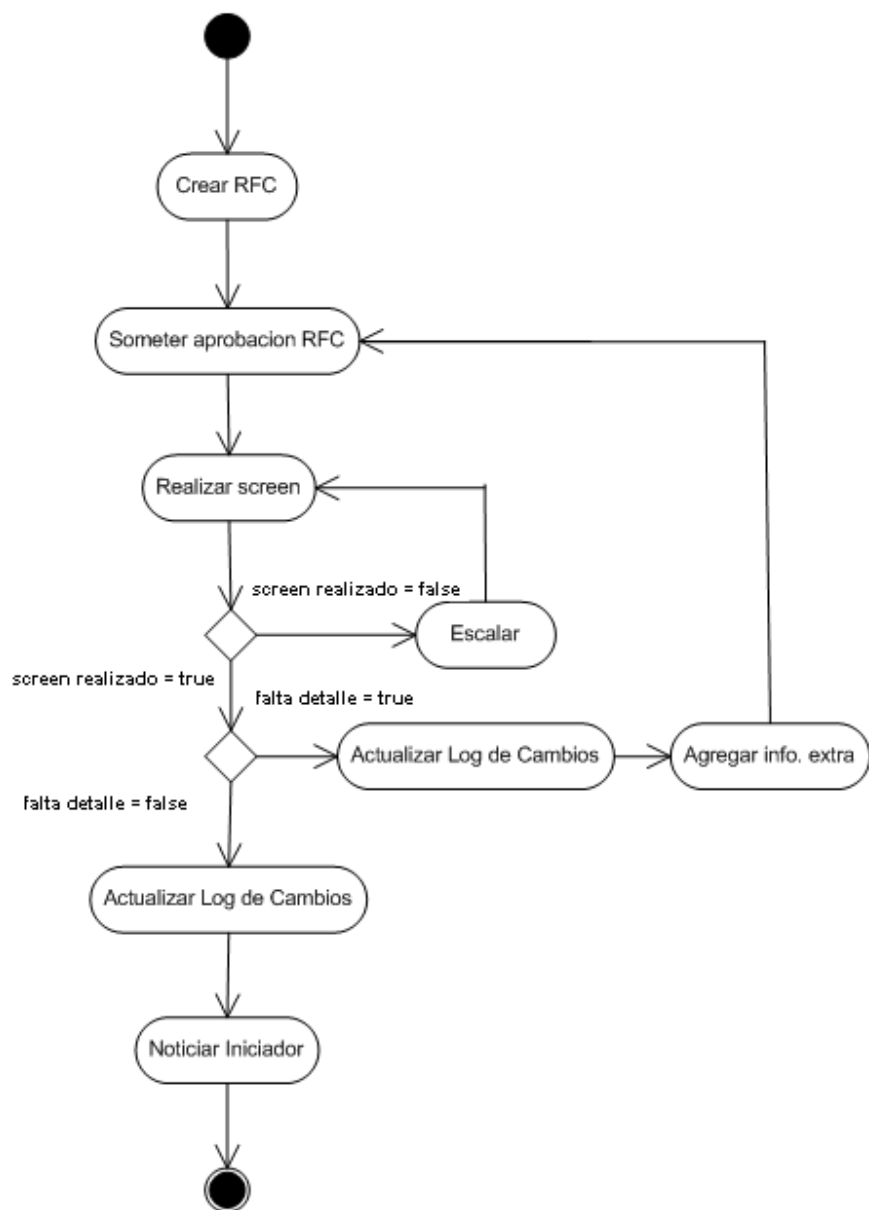


Figura A-31.Diagrama de de actividad del proceso registro o petición de cambio

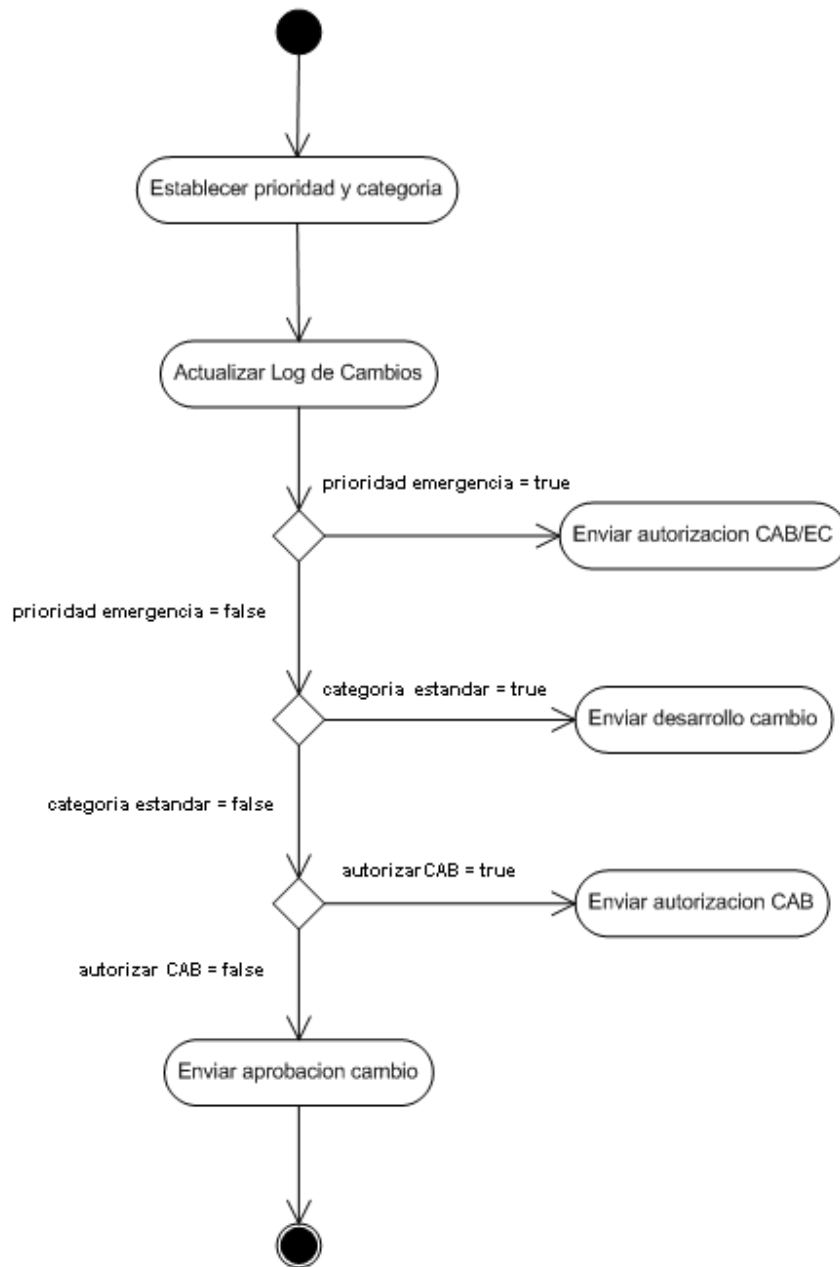


Figura A-32.Diagrama de de actividad del proceso aceptar y clasificar cambio



Figura A-33.Diagrama de de actividad del proceso autorizar cambios de emergencia

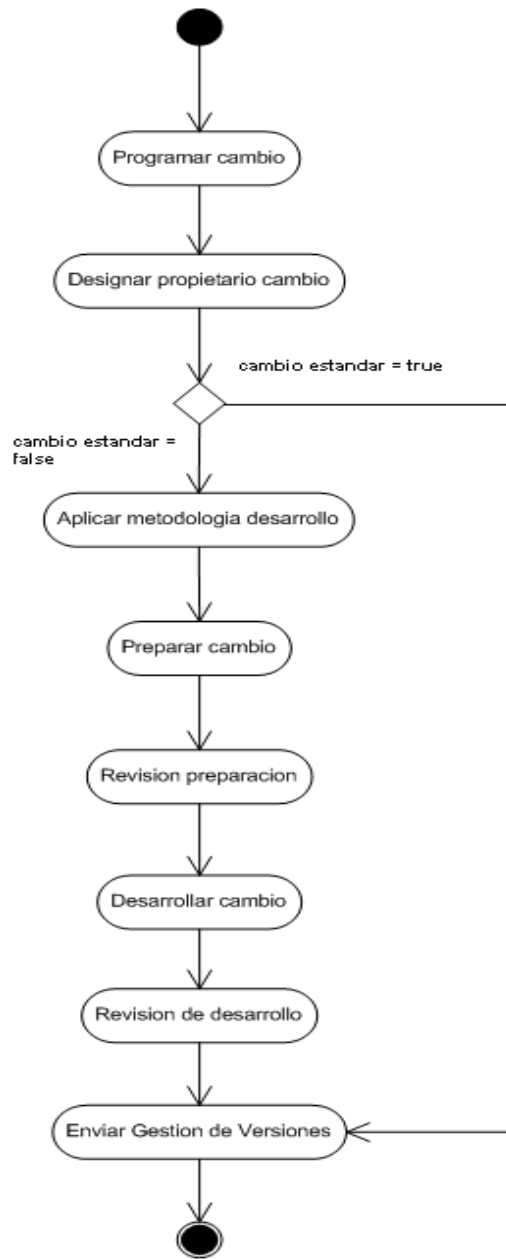


Figura A-34.Diagrama de de actividad del proceso de implementación o despliegue del cambio

Diagrama de Clases

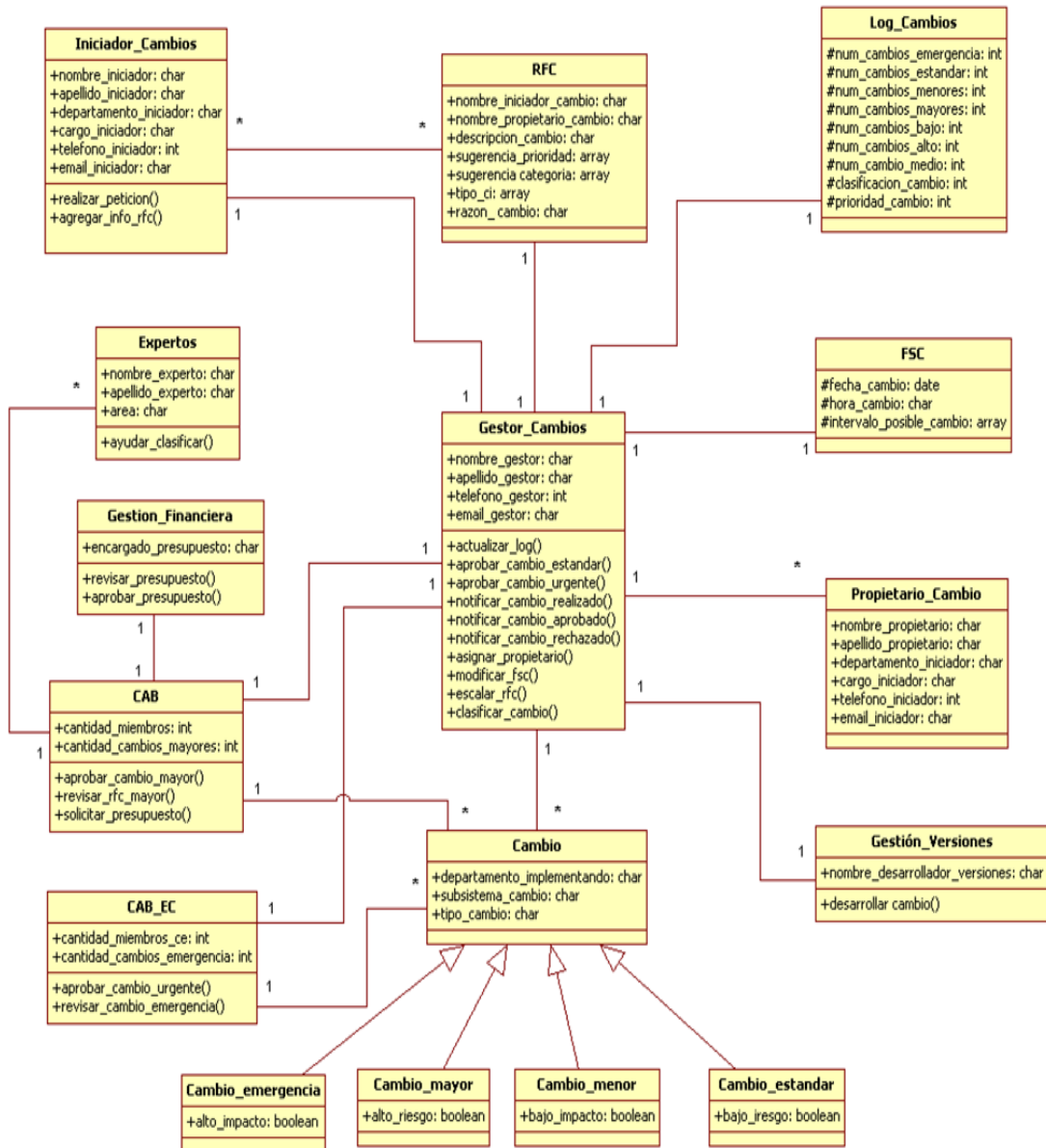


Figura A-35. Diagrama de Clases del Proceso de Gestión de Cambios

Diagrama de Modelo de Dominio

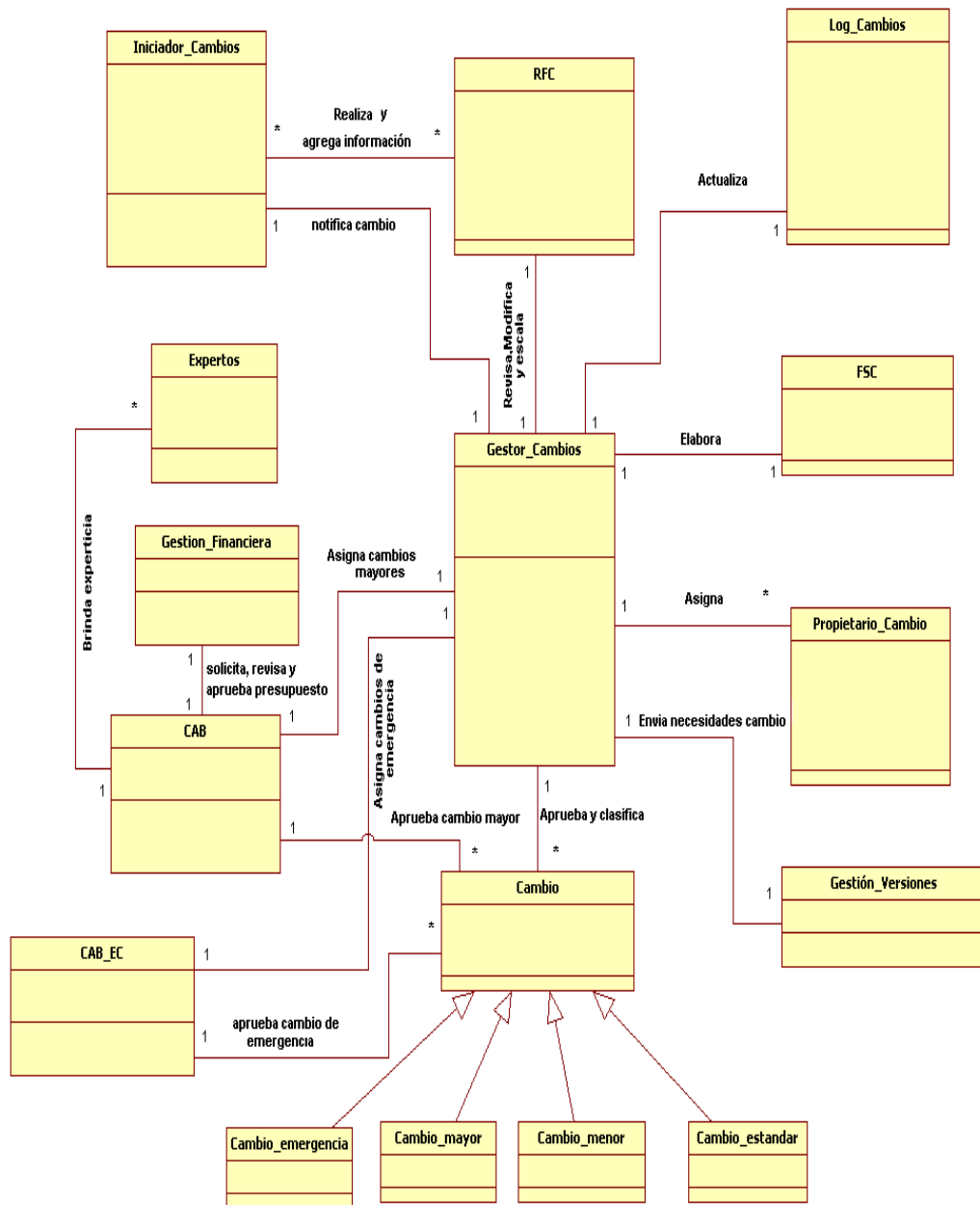


Figura A-36. Diagrama de modelo de dominio del Proceso de Gestión de Cambios

ANEXO B – Instalación y Configuración de Intalio BPMS 5.2.1

A continuación se presenta los pasos necesarios para la instalación del entorno de trabajo de Intalio|BPM Community Edition.

Instalación de Intalio|Server

Para comenzar a trabajar con Intalio Community Edition es necesario, descarga los componentes Intalio|Server e Intalio|Designer. Previamente debemos tener una cuenta en la página de Intalio. Para registrarnos nos dirigimos a la siguiente dirección: <http://community.intalio.com/register.html>

INTALIO | Community Website

Help | Forums | Downloads | Library | Services

To register, please enter your email address. We'll mail you instructions on how to complete your registration. Registration is required to download some parts of Intalio|BPMS and to participate on the [forums](#)

E-mail:

If you don't receive the instruction email, please send us an email at registration@intalio.com, or use the [Contact Us](#) page.

Copyright © Intalio, 1999-2009. [About](#) | [Feeds](#) | [Copyright Notice](#) | [Privacy Policy](#) | [User Agreement](#) | [Site Map](#)

Figura B-1. Registro de usuario en Intalio

Debemos introducir nuestro correo electrónico en la casilla de Email y luego hacemos click en el botón “Register”. Acá aparecerá un mensaje de “Registro Exitoso!”. Intalio nos enviará a nuestro correo la información para activar la cuenta.

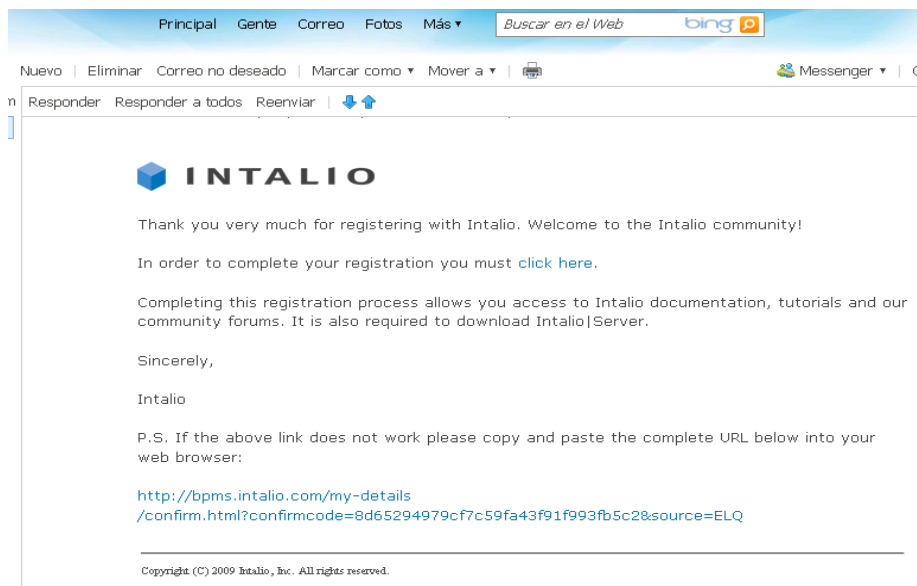


Figura B-2. Correo para activar cuenta en Intalio

Nos dirigimos a nuestro correo para activar la cuenta desde el link que nos envié el remitente de Intalio (registration@intalio.com).

El siguiente paso es dirigirnos a la siguiente dirección para descargar el servidor Intalio|Server y el diseñador Intalio|Designer <http://community.intalio.com/downloads.html>

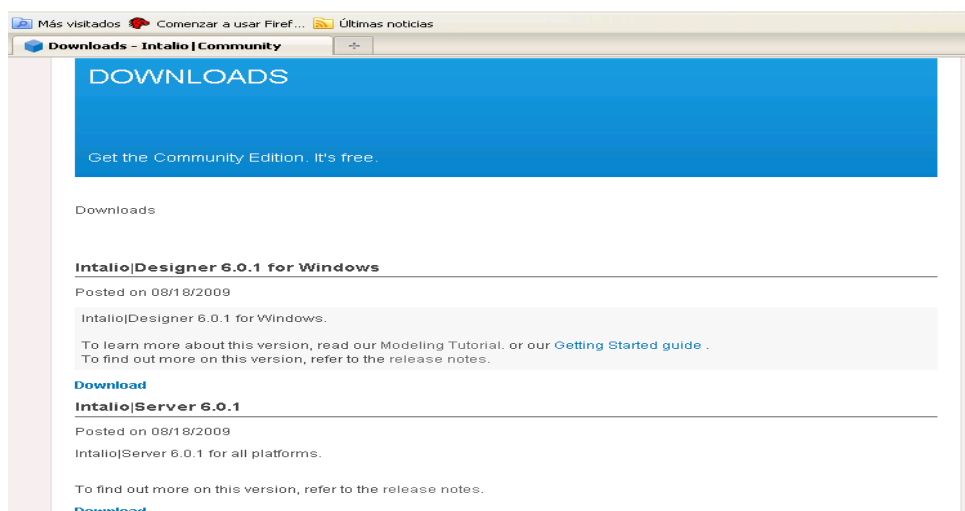


Figura B-3. Descarga de Intalio|Designer e Intalio|Server

Antes de seguir con la instalación debemos tener en cuenta los siguientes prerequisites:

- Intalio|Server corre sobre los siguientes sistemas operativos: Microsoft Windows XP (que es el sistema operativo a utilizar en el modelado planteado en este Trabajo Especial de Grado), Microsoft Windows 2000, Microsoft Windows 2003 Server, Linux y Mac OS X
- Se debe tener 512Mb mínimo de RAM
- Se necesita 200 Mb libres mínimo de espacio en el disco duro.
- Se debe tener instalado la última versión del JDK 1.5 o del 1.6
- Se debe crear una variable del sistema con el nombre JAVA_HOME que apunte a la carpeta del JDK que tengamos instalado en el equipo.

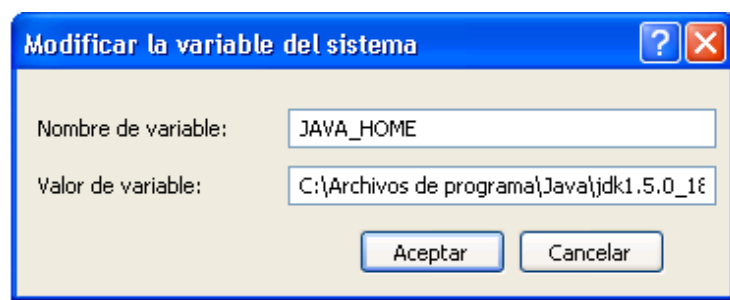


Figura B-4. Variable JAVA_HOME apuntado al JDK 1.5.0

Ahora procedemos a instalar el servidor de Intalio. Al momento de descargar Intalio|Server se generó el archivo "intalio-bpms-5.2.1.021.zip" el cual trae todos los componentes necesarios para la instalación del Servidor. Este archivo lo extraemos en la dirección raíz del equipo, que por lo general es C:\ (ejemplo. C:\intalio-bpms-5.2.1.021).

Luego de extraer en C.\ el servidor debe generarse una carpeta llamada "intalio-bpms-5.2.1.021" que contiene 8 carpetas (bin, databases, extras, gerónimo-docs, lib, repository, schema y var) y un archivo (INTALIO-LICENSE.txt).

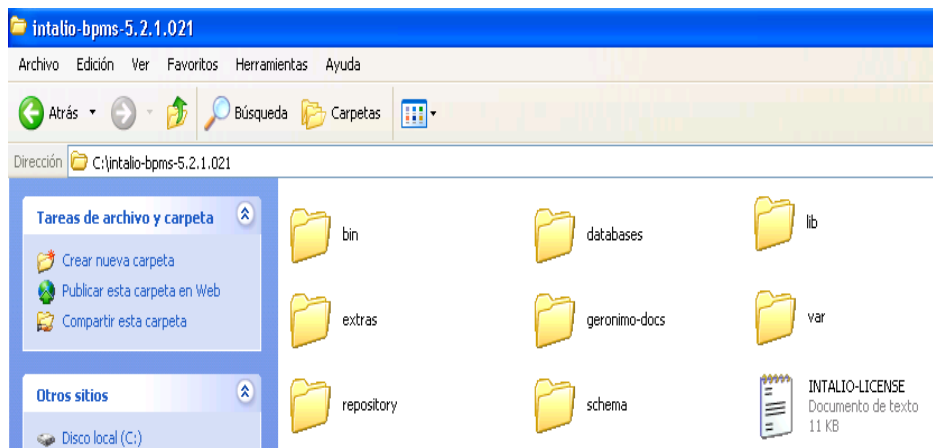


Figura B-5. Subcarpetas de intalio-bpms-5.2.1.

Una vez instalado el servidor de Intalio procedemos a ejecutarlo. Desde un terminal de consola nos dirigimos a la siguiente dirección: “C:\intalio-bpms-5.2.1.021\bin”. Acá tecleamos el siguiente comando: “startup.bat” y nos aparece el siguiente mensaje: “Starting Geronimo in a separate window ...” lo que indica que el Servidor de Intalio esta levantándose.

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

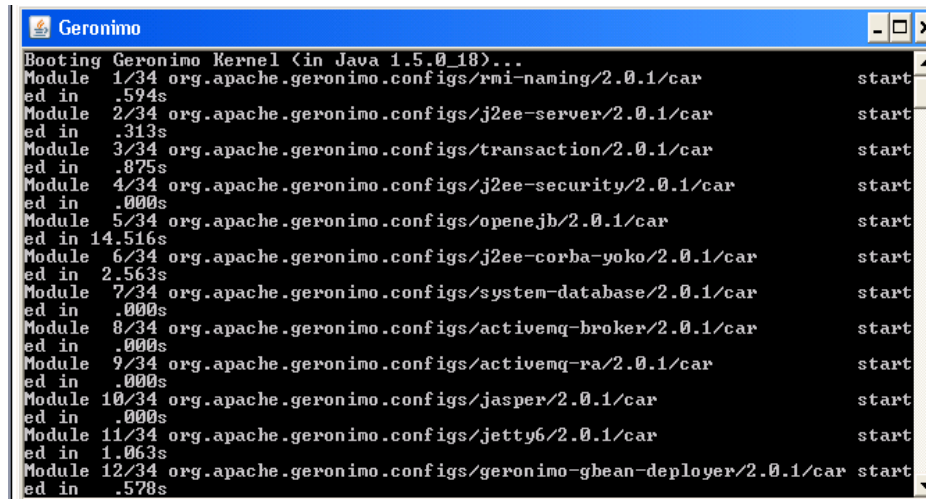
C:\Documents and Settings\Shijan Jorge>cd ..
C:\Documents and Settings>cd ..
C:\>cd intalio-bpms-5.2.1.021
C:\intalio-bpms-5.2.1.021>cd bin
C:\intalio-bpms-5.2.1.021\bin>startup.bat
Using GERONIMO_BASE: C:\intalio-bpms-5.2.1.021
Using GERONIMO_HOME: C:\intalio-bpms-5.2.1.021
Using GERONIMO_TMPDIR: var\temp
Using JRE_HOME: C:\Archivos de programa\Java\jdk1.5.0_18\jre

Starting Geronimo in a separate window...
C:\intalio-bpms-5.2.1.021\bin>

```

Figura B-6. Comando “startup.bat” para levantar el servidor Intalio|Server

Alternativamente podemos ver en otra ventana que comienza a levantarse el servidor Gerónimo. La primera línea de la ventana debe Mostar el siguiente mensaje: “Booting Gerónimo Kernel (in Java 1.5.0_18)”.



```
Booting Geronimo Kernel (in Java 1.5.0_18)...
Module 1/34 org.apache.geronimo.configs/rmi-naming/2.0.1/car start
ed in .594s
Module 2/34 org.apache.geronimo.configs/j2ee-server/2.0.1/car start
ed in .313s
Module 3/34 org.apache.geronimo.configs/transaction/2.0.1/car start
ed in .875s
Module 4/34 org.apache.geronimo.configs/j2ee-security/2.0.1/car start
ed in .000s
Module 5/34 org.apache.geronimo.configs/openejb/2.0.1/car start
ed in 14.516s
Module 6/34 org.apache.geronimo.configs/j2ee-corba-yoko/2.0.1/car start
ed in 2.563s
Module 7/34 org.apache.geronimo.configs/system-database/2.0.1/car start
ed in .000s
Module 8/34 org.apache.geronimo.configs/activemq-broker/2.0.1/car start
ed in .000s
Module 9/34 org.apache.geronimo.configs/activemq-ra/2.0.1/car start
ed in .000s
Module 10/34 org.apache.geronimo.configs/jasper/2.0.1/car start
ed in .000s
Module 11/34 org.apache.geronimo.configs/jetty6/2.0.1/car start
ed in 1.063s
Module 12/34 org.apache.geronimo.configs/geronimo-gbean-deployer/2.0.1/car start
ed in .578s
```

Figura B-7. Servidor Gerónimo iniciándose

A continuación comienzan a iniciarse 34 módulos que comprenden el servidor Apache Gerónimo. Estos módulos son:

- org.apache.geronimo.configs/rmi-naming/2.0.1/car
- org.apache.geronimo.configs/j2ee-server/2.0.1/car
- org.apache.geronimo.configs/rmi-naming/transaction/car
- org.apache.geronimo.configs/j2ee-security/2.0.1/car
- org.apache.geronimo.configs/openjib/2.0.1/car
- org.apache.geronimo.configs/j2ee-corba-yoko/2.0.1/car
- org.apache.geronimo.configs/system-database/2.0.1/car
- org.apache.geronimo.configs/activemq-broker/2.0.1/car
- org.apache.geronimo.configs/activemq-ra/2.0.1/car
- org.apache.geronimo.configs/jasper/2.0.1/car
- org.apache.geronimo.configs/jetty6/2.0.1/car
- org.apache.geronimo.configs/geronimo-gbean-deployer/2.0.1/car

- org.apache.geronimo.configs/j2ee-deployer/2.0.1/car
- org.apache.geronimo.configs/connector-deployer/2.0.1/car
- org.apache.geronimo.configs/sharedlib/2.0.1/car
- org.apache.geronimo.configs/jetty6-deployer/2.0.1/car
- org.apache.geronimo.configs/jasper-deployer/2.0.1/car
- org.apache.geronimo.configs/welcome-jetty/2.0.1/car
- org.apache.geronimo.configs/dojo-jetty6/2.0.1/car
- org.apache.geronimo.configs/webconsole-jetty6/2.0.1/car
- org.apache.geronimo.configs/remote-deploy-jetty6/2.0.1/car
- org.apache.geronimo.configs/hot-deployer/2.0.1/car
- org.apache.geronimo.configs/jsr88-rar-configurer/2.0.1/car
- org.apache.geronimo.configs/clustering/2.0.1/car
- com.intalio.bpms/BPMSPDS/1.0/rar
- com.intalio.bpms/pxe/5.1.2/war
- com.intalio.bpms/console/5.2.1.2/war
- com.intalio.bpms/axis2-services/1.1.1/war
- com.intalio.bpms.wsi/wsi/5.2.0.5/war
- org.intalio.tempo/fds/5.2.1.5/war
- org.intalio.tempo/wi-fw/5.2.1.5/war
- org.intalio.tempo/wds/5.2.1.5/war
- org.intalio.tempo/xforms-manager/5.2.1.5/war

Luego de iniciar los 34 módulos de Gerónimo Apache aparece el siguiente mensaje al final de la ventana: “Geronimo Application Server Started” que indica que el servidor de Intalio ya se encuentra levantado y listo para ser utilizado.

Para detener la ejecución del Servidor en cualquier momento, tecleamos en la consola Ctrl+C.


```
WAR: org.apache.geronimo.configs/remote-deploy-jetty/2.0.1/car
WAR: org.apache.geronimo.configs/welcome-jetty/2.0.1/car
WAR: org.intalio.tempo/fds/5.2.1.5/war
WAR: org.intalio.tempo/ui-fw/5.2.1.5/war
WAR: org.intalio.tempo/wds/5.2.1.5/war
WAR: org.intalio.tempo/xforms-manager/5.2.1.5/war

Web Applications:
/
/axis2
/bpms-console
/console
/console-standard
/dojo
/fds
/ode
/remote-deploy
/ui-fw
/wds
/wsi
/xFormsManager

Geronimo Application Server started
```

Figura B-8. Servidor Apache Gerónimo inicializado

Para comprobar que el servidor esta inicializado correctamente, abrimos una ventana del navegador web y nos dirigimos a la dirección del Local Host de Intalio (<http://localhost:8080/bpms-console>) para tener acceso a la consola de administración. El nombre de usuario y la contraseña por defecto son: “admin” y “changeit”.

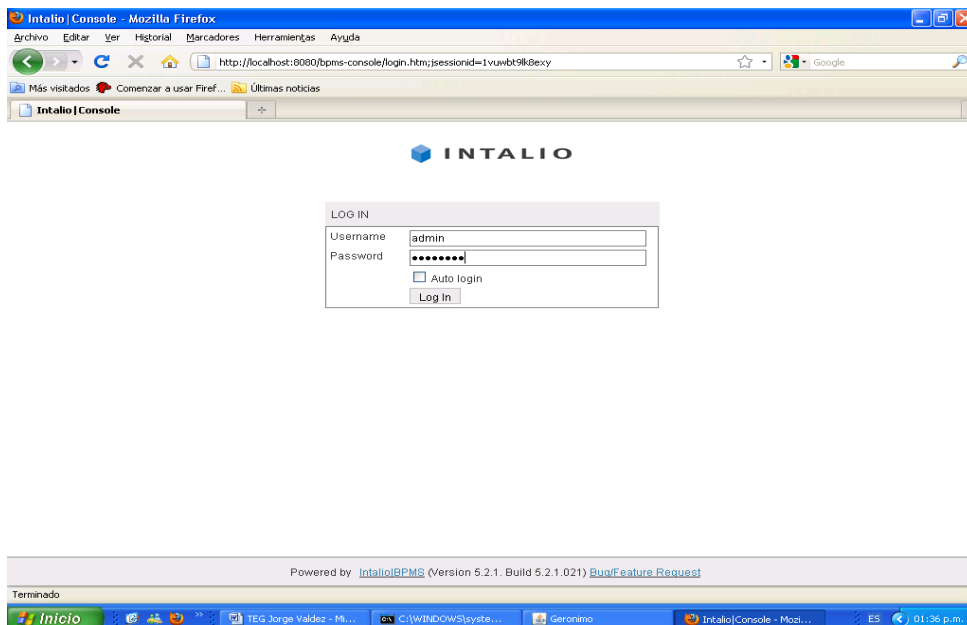


Figura B-9. Consola de Intalio|BPM

Instalación de Intalio|Designer

Una vez instalado el servidor de Intalio procedemos a instalar el Diseñador Intalio|Designer.

Al momento de descargar Intalio|Designer se generó el archivo “com.intalio.bpms.designer.distribs.ce.win32-5.2.1.158-installer” que es un archivo JAR ejecutable el cual trae todos los componentes necesarios para la instalación del diseñador. Este archivo JAR contiene un conjunto de plugins con la distribución completa de Eclipse 3.1 SDK.

Al ejecutar el instalador puede aparecer un mensaje de Advertencia de Seguridad indicando que se desconoce el fabricante del programa.

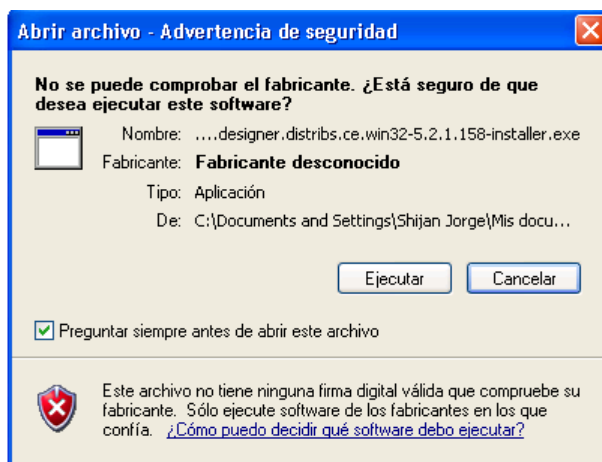


Figura B-10. Mensaje de advertencia de seguridad

Hacemos click sobre el botón “Ejecutar” y de inmediato comienza a cargarse el instalador.

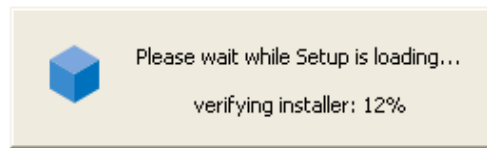


Figura B-11. Mensaje de carga de instalador de Intalio|Designer

Luego de cargarse por completo el instalador seleccionamos el idioma en que deseamos trabajar en la instalación y hacemos click en el botón “OK”. Los idiomas disponibles son ingles, portugués y francés.

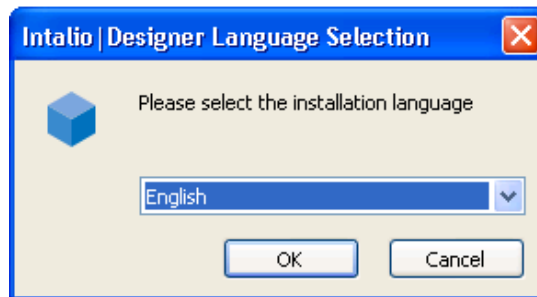


Figura B-12. Combo para seleccionar idioma del instalador de Intalio|Designer

Una vez seleccionado el idioma del instalador nos aparece una pantalla que nos da la bienvenida al programa de instalación del diseñador de Intalio para su edición comunitaria (Community Edition). Acá hacemos click en el botón “Next”.

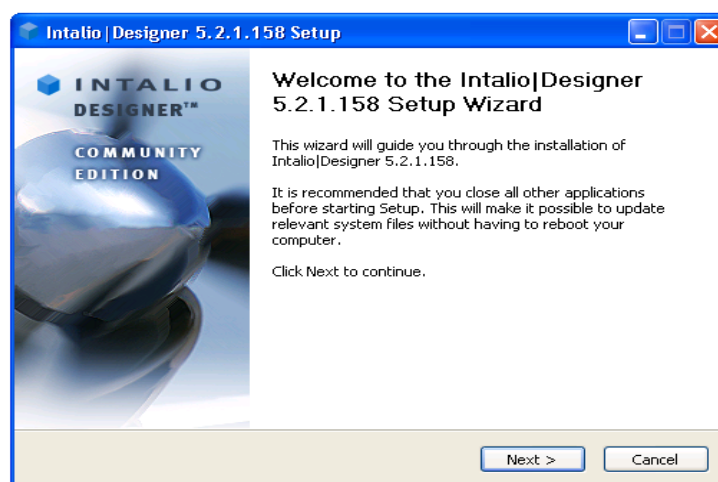


Figura B-13. Pantalla de bienvenida al Wizard de instalación de Intalio|Designer

El siguiente paso es leer los términos de la licencia de Intalio|Designer. Luego de leerla detalladamente y estar de acuerdo con lo que allí se encuentra establecido hacemos click en el botón "I Agree".

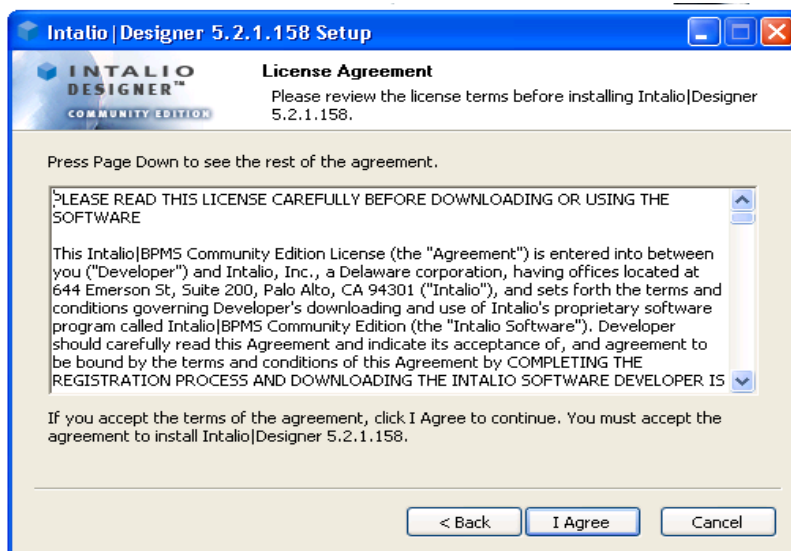


Figura B-14. Licencia de acuerdo de Intalio|Designer

Ahora se debe elegir la ubicación en que deseamos instalar el diseñador. Podemos dejar la ruta que trae por defecto (C:\Archivos de programa\Intalio-Designer-5.2.1.158) o seleccionar otra.

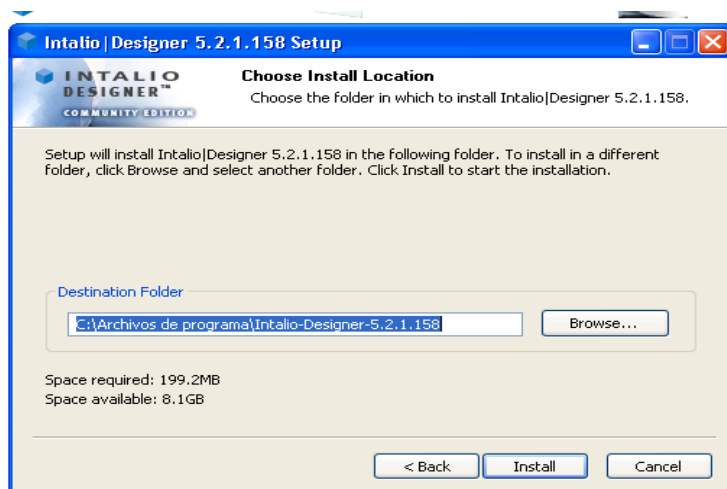


Figura B-15. Ventana para elegir ubicación de la instalación de Intalio|Designer

Ya seleccionado la ubicación, comienza el proceso de instalación del diseñador.

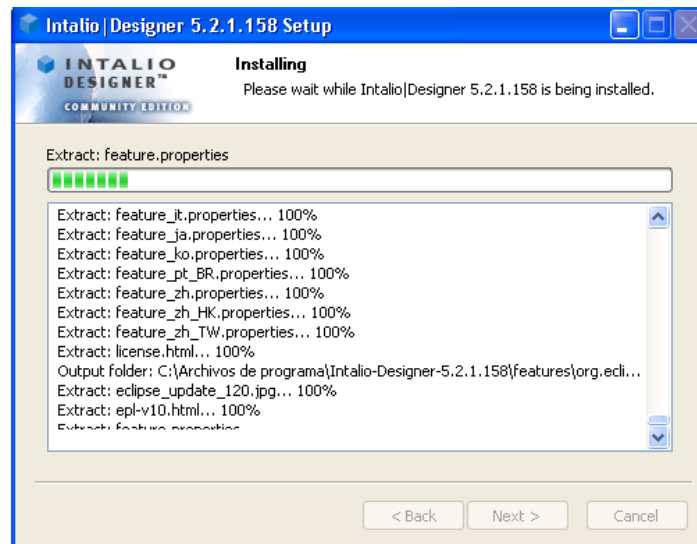


Figura B-16. Proceso de instalación de Intalio|Designer

Una vez completa la instalación nos aparece el mensaje que presenta la figura B-17 indicándonos que podemos ejecutar Intalio|Designer. Hacemos click en el botón "Finish".

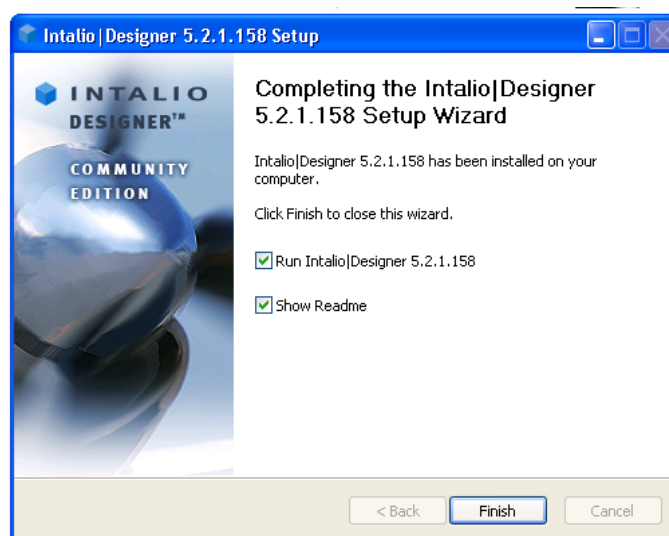


Figura B-17. Mensaje indicando que Intalio|Designer ha sido instalado

A continuación aparece la pantalla de carga de Intalio|Designer tal como lo muestra la figura B-18.



Figura B-18. Proceso de carga de Intalio|Designer

Debemos seleccionar el workspace en que vamos a trabajar. Podemos dejar el que viene por defecto (C:\Documents and Settings\\workspace) o elegir otro. Luego hacemos click en el botón “OK”.

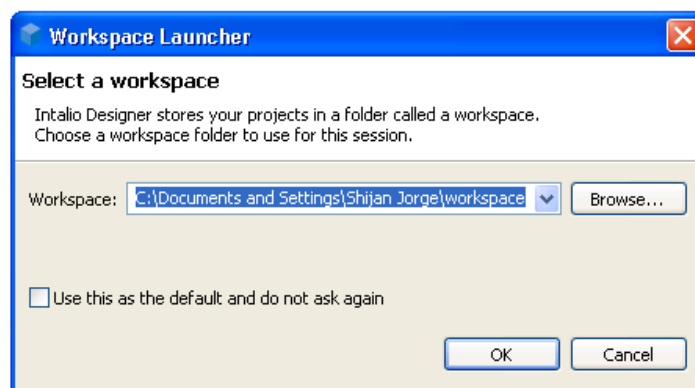


Figura B-19. Pantalla para elegir Workspace

Cuando entramos al diseñador por primera vez debemos loguearnos con el nombre de usuario y la contraseña que nos envió a nuestro correo los administradores de Intalio al momento de registrarnos.



Figura B-20. Pantalla para loguearse

Ahora si podemos presenciar la pantalla del diseñador Intalio|Designer. La figura B-21 nos demuestra el entorno de trabajo del diseñador de Intalio.

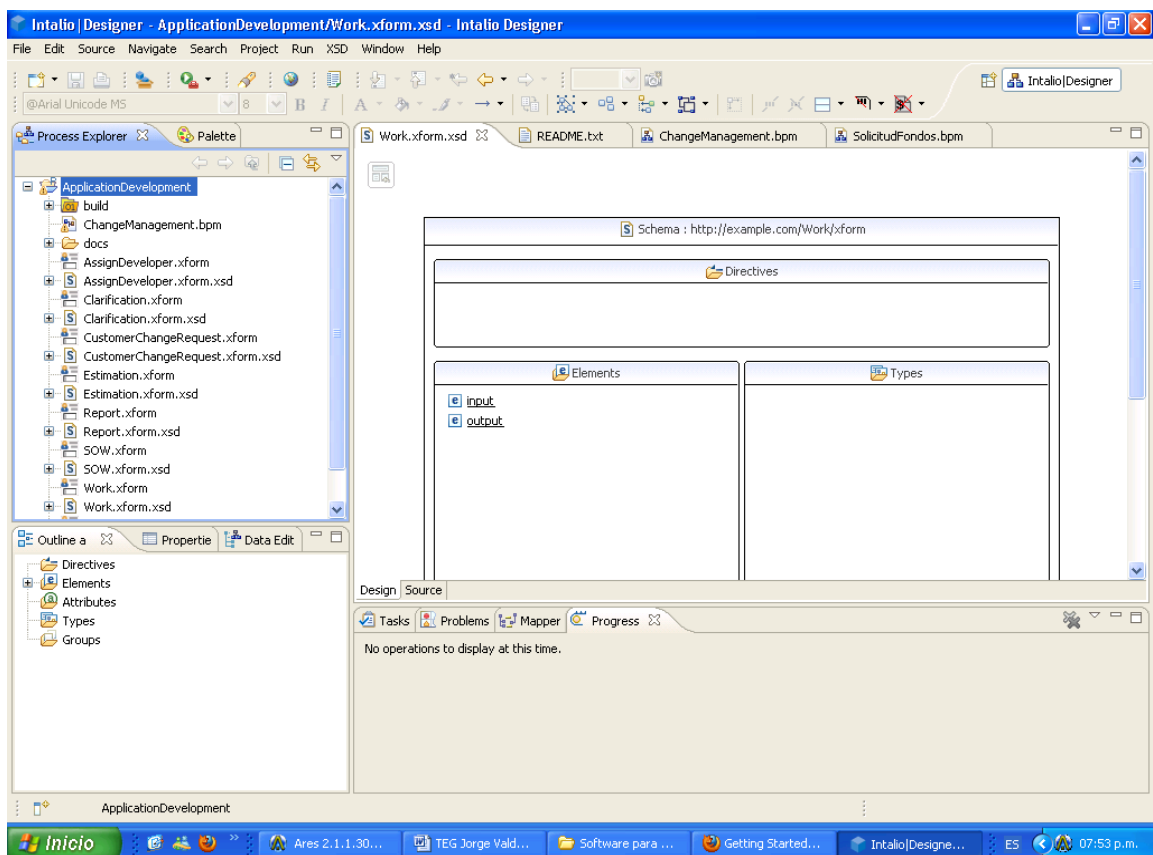
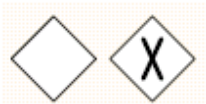


Figura B-21. Entorno de trabajo de Intalio|Designer

ANEXO C – Notación BPMN

Gateways



Gateway Exclusivo para Base de Datos: Al separarse, se encamina el flujo de la secuencia exactamente a una de las ramas según las condiciones en que se encuentre basada. Al combinarse, aguarda una rama de entrada para terminar antes de accionar el flujo de salida.



Gateway Exclusivo basado en eventos: siempre recibe eventos o tareas. El flujo de la secuencia se encamina al subsiguiente evento o tarea.



Gateway Paralelo: son utilizadas para separar el flujo de la secuencia, todas las ramas salientes se activan simultáneamente. Todas las ramas se combinan antes de terminar el flujo de salida.



Gateway Inclusivo: al separarse, una o más ramas son activadas dependiendo de las condiciones de la ramificación. Al combinarse, se aguarda por todas las ramas de entradas finalicen.



Gateway Complejo: acciona una o más ramas basadas en condiciones complejas o descripciones verbales.



Eventos

Planificación: indican donde los procesos inician o finalizan		
Inicio	Intermedio	Fin



Mensaje: recibe y envía mensajes		
Inicio	Intermedio	Fin




Temporizador: sincroniza eventos cíclicos		
Inicio	Intermedio	Fin




Error: captura o lanza errores nombrados		
Inicio	Intermedio	Fin


Cancelación: acciona la cancelación de transacciones		
Inicio	Intermedio	Fin
		


Compensación: maneja compensaciones		
Inicio	Intermedio	Fin
		

Condicionales: reacciona ante cambios de condiciones en el negocio o en la integración de reglas de negocio		
Inicio	Intermedio	Fin
		

Señales: permiten la señalización a través de diversos procesos. Una señal generada puede ser atrapada múltiples veces		
Inicio	Intermedio	Fin
		

Múltiple: emana o lanza un acontecimiento de un conjunto de ellos		
Inicio	Intermedio	Fin
		

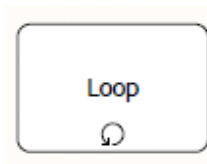
Enlace: dos eventos iguales son enlazados en un flujo de secuencias		
Inicio	Intermedio	Fin
		

Culminación: indica la culminación de un proceso		
Inicio	Intermedio	Fin
		

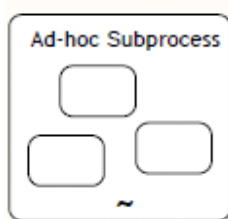
Actividades



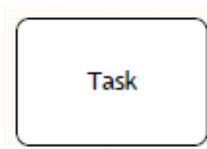
Múltiples instancias de la misma actividad son iniciadas en Paralelo o secuencialmente



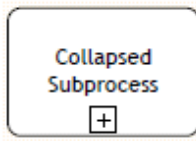
Esta condición cíclica se cumple mientras una condición del bucle sea verdadera



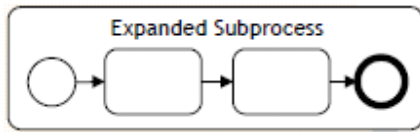
Subprocesos Ad-hoc contiene solamente tareas. Cada tarea se puede ejecutar arbitrariamente hasta que se satisfaga una condición para terminar



Una tarea es la unidad de trabajo



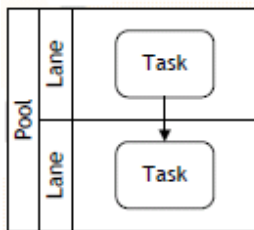
Un subprocesso es una actividad que se puede descomponer. También puede ocultarse los detalles de este



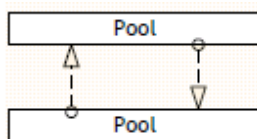
Un subprocesso expandido contiene un

diagrama BPMN valido

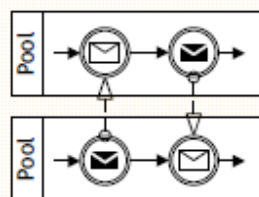
Carriles



Pool y carriles representan responsabilidades para actividades en un proceso. Un pool o un carril pueden ser una organización, un rol, o un sistema

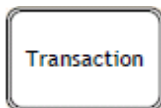


El flujo de mensajes simboliza el flujo de información entre las áreas de una organización. El flujo de mensajes puede ser adjuntado a pools, actividades, o eventos de mensajes

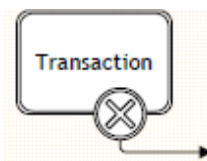


El orden del intercambio de mensajes puede ser especificada combinando el flujo del mensaje y el flujo de la secuencia.

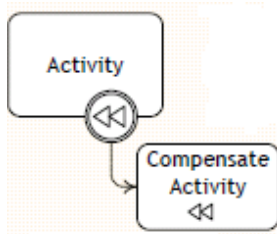
Transacciones



una transacción es un conjunto de actividades que lógicamente están juntas; y pueden seguir un protocolo especificado para la transacción



Los eventos de cancelación intermedios indican que reaccionan ante una transacción de cancelación. Las actividades de la transacción son compensadas tras la cancelación



Actividades completadas pueden ser compensadas. Una actividad y su Actividad de Compensación correspondiente son relacionadas usando un Evento de Compensación Intermedio adjuntado

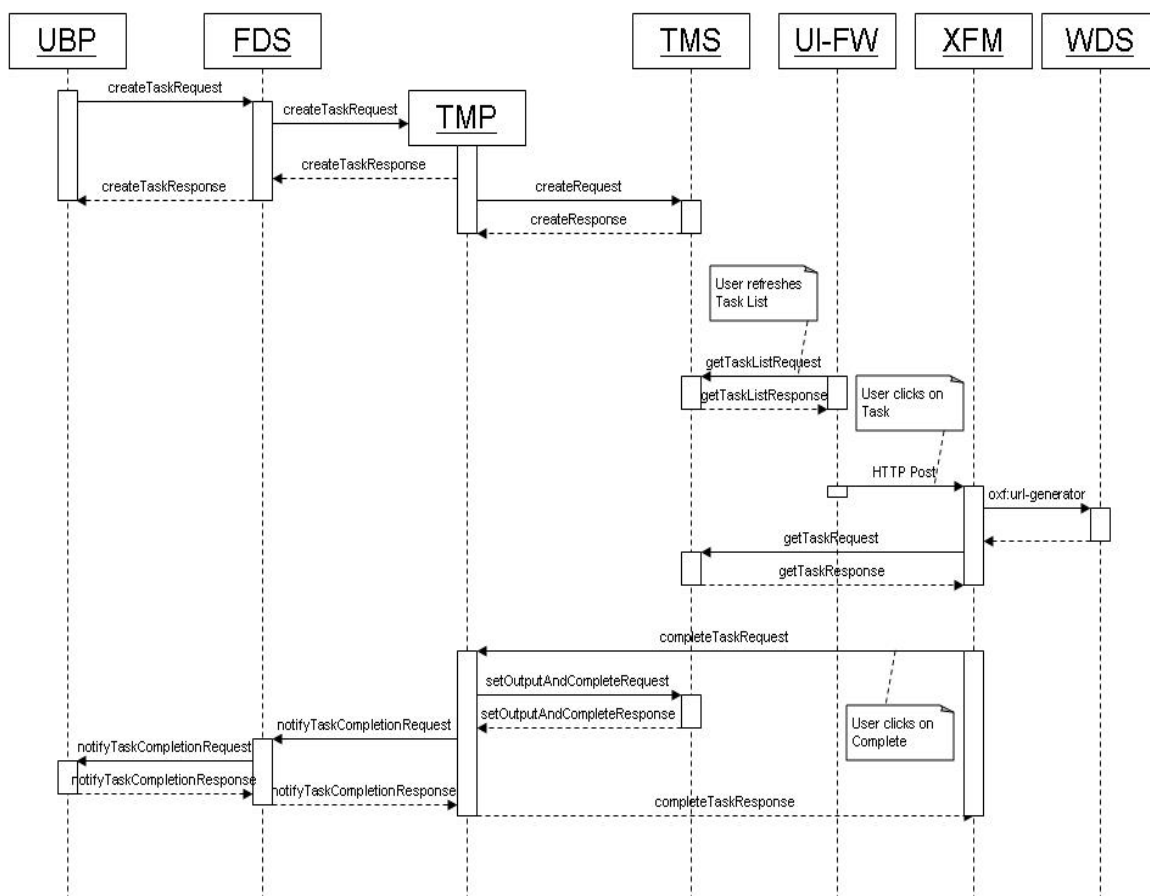
Documentación



Un conjunto arbitrario de objetos puede ser definido como un grupo que muestra que están juntos lógicamente

ANEXO D – Creación de una tarea en la arquitectura Tempo

El siguiente diagrama de secuencia explica cómo se utilizan los diversos componentes de Tempo cuando se crea y se completa una tarea.



- Proceso de Negocio del Usuario (User Business Process - UBP).** Este es el proceso que crea la tarea. Es típicamente un proceso BPEL, pero de hecho podría ser cualquier aplicación. Hace una llamada a un servicio web para crear la tarea, y proporciona una operación de servicio web para completar la tarea.

- **Servicio Despachador de Formularios** (*Form Dispatcher Service - FDS*). Actúa como un Proxy el UBP y el TMP.
- **Gestor de procesos de Tareas** (*Task Manager Process - TMP*). Es el proceso BPEL responsable de manejar el ciclo de vida de la tarea. Es instanciado por la recepción del proceso de petición de creación de tarea (`createTaskRequest`) desde el FDS
- **Servicios de Gestión de Tareas** (*Task Management Services - TMS*). Esto es un servicio web que proporciona las acciones para el flujo de trabajo, se encarga de crear persistencia en la base de datos y de gestionar la seguridad.
- **Marco de Interfaz de Usuario** (*User Interface Framework - UI-FW*): Muestra la lista de tareas y envía los formularios al encargado de los formularios (XFM).
- **Encargado de XForms** (*XForms Manager – XFM*) Ésta es una implementación de un Encargado de de Formularios, para formularios XForms.
- *Servicio del despliegue del flujo de trabajo* (*Workflow Deployment Service – WDS*):. Maneja el almacenamiento y el acceso a los formularios.

La creación y completado de tareas son totalmente separadas. El identificador de tareas (Task ID) se utiliza para correlacionar el mensaje del `createTaskResponse` con el mensaje `notifyTaskCompletionRequest`. El Task ID es generado por el TMP.

Crear una tarea

Esto comienza por una llamada de un servicio web al FDS. EL FDS enruta la petición al TMP. TMP entonces crea la tarea llamando al TMS.

El FDS se requiere para manejar cualquier proceso. Desde la perspectiva de TMP, el mensaje `createTaskRequest` se puede enviar solamente por el único `partnerLink` definido. Todos los diferentes UBP se corresponde a diferentes `partnerLink`. Para solucionar este problema, el FDS es un servlet que puede dirigir la relación cualquiera-a-uno a la relación aceptando todos los mensajes de `createTaskRequest` van a `/fds/workflow` y cambian dinámicamente el namespace donde el TMP.

Cuando el TMP vuelve el `createTaskResponse`, también se envía a `/fds/workflow` y por lo tanto es capturado por el FDS, que cambia el namespace de nuevo al anterior. El namespace se encuentra en el mensaje de `createTaskResponse` y fue trazado des el `createTaskRequest`. Esta es la razón por la cual el namespace requiere ser trazado en el UBP al crear una tarea. Permite ser devuelto en el `createTaskResponse` al FDS de modo que el FDS puede construir el mensaje de `createTaskResponse` que una instancia específica de UBP está esperando.

Realización de una tarea

Cuando un usuario abre una sesión o restaura la lista de tarea, el UI-FW hace una llamada a TMS para recuperar la lista de tarea actualizada. Al hacer click en una tarea, la petición es encaminada al Encargado de formularios correspondiente de la forma. En el momento, UI-FW postea solamente a XFM, pero podrían ampliar para seleccionar dinámicamente a un Encargado de Formularios basado en las propiedades de la tarea. XFM entonces llama a TMS para conseguir todos los detalles de la tarea. También llama WDS para recuperar

el formulario actual. Con toda esta información, XFM rinde el formulario con sus datos de entrada, así es como el botón "Complete" permite al usuario termina la tarea o cualquier otra acción XFM que el flujo de trabajo apoya.

Cuando se hace click en el botón "Complete", XFM llama a TMP para terminar la tarea. Entonces TMP cambia la salida de la tarea y su estado llamando a TMS.

Entonces remite la tarea de vuelta al UBP a través del FDS. Aquí nuevamente, el FDS necesita estar implicado. El Identificador de la Tarea es usado para la correlación.

Finalmente, TMP envía las respuestas de nuevo al XFM después de recibir la respuesta desde UBP.

ANEXO E – Detalles del modelado del Proceso de Gestión de Cambios en las organizaciones de Tecnologías de Información con la herramienta Intalio BPMS 5.2.1

El primer paso en el modelado del Proceso de Gestión de Cambios es crear el Workflow del mismo.

Intalio|Designer utiliza la Notación para el Modelado de Procesos de Negocio (Business Process Modeling Notation o BPMN), que es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (Workflow).

Antes de iniciar el proceso de modelado es necesario indicar cuantos roles van a participar en el Workflow.

El proceso de Gestión de Cambios, planteado por la Biblioteca de Infraestructura de Tecnologías de la Información, indica varios actores. Para efectos prácticos del modelado utilizaremos los tres actores más destacados: el iniciador de la petición de cambio, el gestor del cambio y el desarrollador de versiones. Por cada rol se creará un **“pool no ejecutable”** y adicionalmente se creará otro pool no ejecutable para la interfaz de inicio.

Un **“pool no ejecutable”** indica que un determinado pool no representara un proceso de negocio y por lo tanto no será ejecutado en el servidor.

Para el proceso en si se creó otro **“pool ejecutable”**. Un pool ejecutable indica que el proceso representa un proceso de negocio y este será el que se ejecute en el servidor.

Por lo tanto la distribución de los pools queda de la siguiente manera:

- **Agency:** pool que representa la interfaz de inicio. Es un pool no ejecutable
- **Customer:** representa al iniciador del cambio. Es un pool no ejecutable
- **Change Management:** es el proceso en si. Es el único pool ejecutable del diagrama.
- **Manager:** este pool representa al Gestor de Cambios. Es un pool no ejecutable
- **Developer:** representa al Gestor de Versiones. Es un pool no ejecutable.

Para comenzar a modelar es necesario utilizar los controles que nos proporciona la paleta de controles del diseñador:

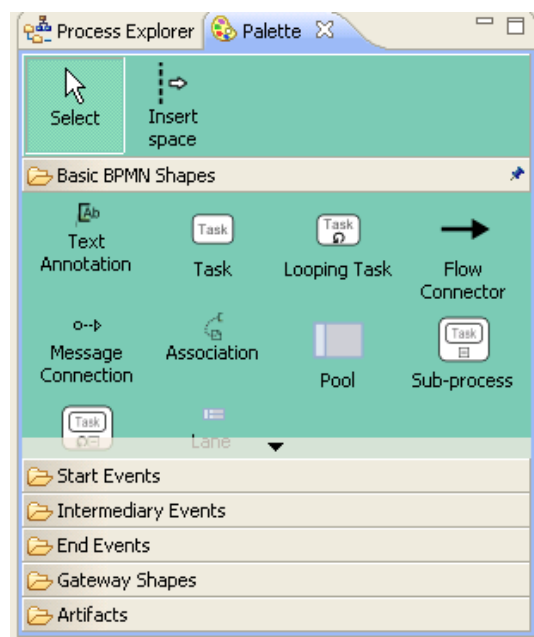


Figura E-1. Paleta de Controles de Intalio|Designer

En Intalio|Designer todo proceso de negocio a modelar debe iniciar con un “proceso cero” también llamado agencia, interfaz o proceso inicial que es el encargado de enviar el mensaje de inicio al primer rol del Workflow.

La Agencia envía un mensaje de inicio al iniciador del cambio llamado Cliente. El Cliente recibe y califica (según su criterio) la petición de cambio para luego enviarla al proceso de Gestión de Cambio.

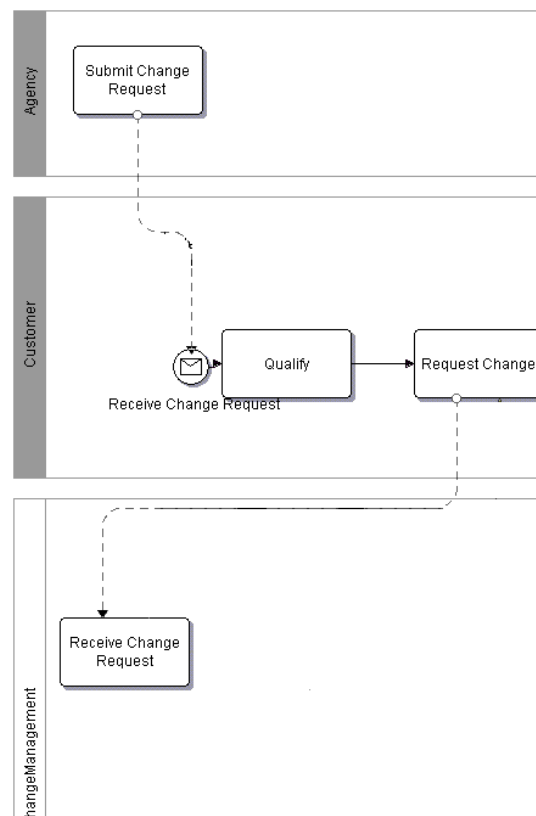


Figura E-2.El Iniciador envía la RFC al Encargado del cambio.

Cuando al Proceso de Cambios le llega una petición, este le notifica al Encargado del Cambio que se ha recibido una RFC. Este revisa dicha RFC y le asigna un desarrollador basado en el conocimiento inicial establecido en la descripción de la petición y en el conjunto de habilidades que tiene el desarrollador. Este desarrollador asignado es procesado en el Proceso de Gestión de Cambios.

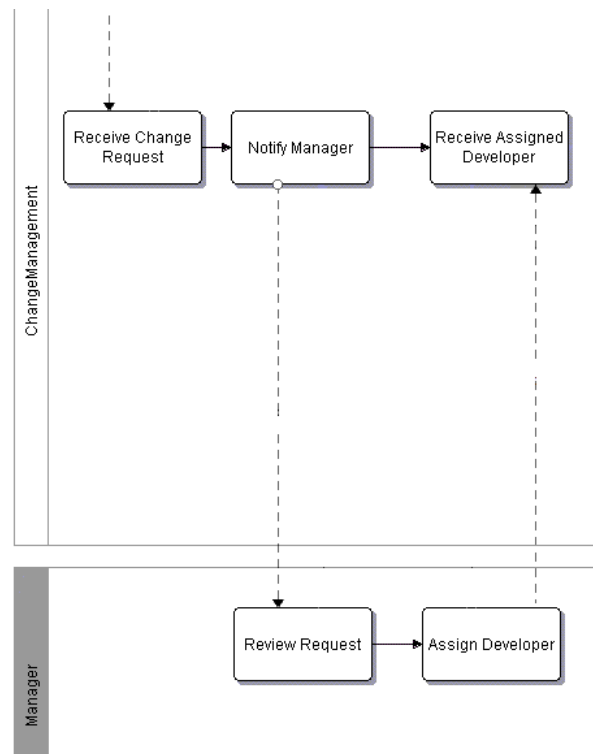


Figura E-3. El Encargado asigna un desarrollador para el cambio.

Luego de que el Encargado asigna un Desarrollador al RFC, el proceso de Cambio entra en un subproceso iterativo donde se evalúa la petición y se asigna una prioridad de cambio.

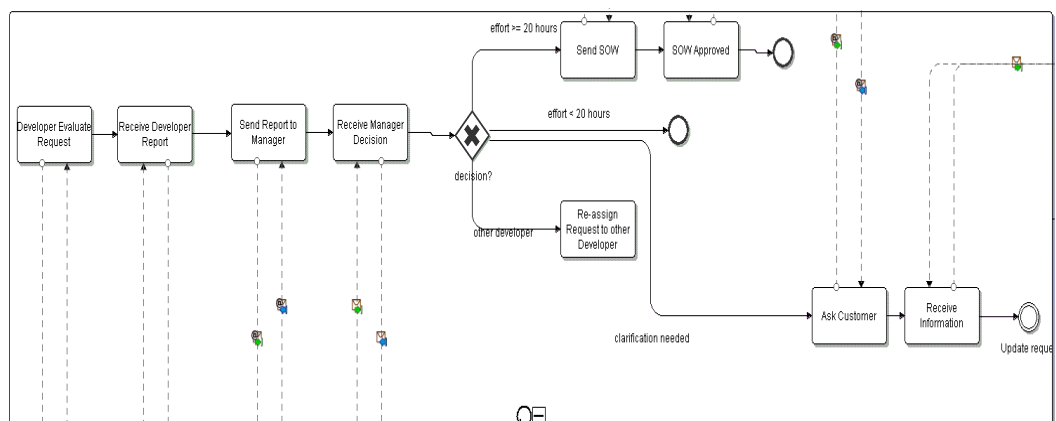


Figura E-4. Subproceso de evaluación y asignación de prioridad de RFC

Una vez dentro del subproceso iterativo, el primer paso es del Desarrollador. Este evalúa la petición, estima el esfuerzo en que se realizará el cambio y le responde al Encargado del Cambio con recomendaciones para asignar el cambio al desarrollador más apropiado. El Encargado recibe el reporte y decide la prioridad del cambio.

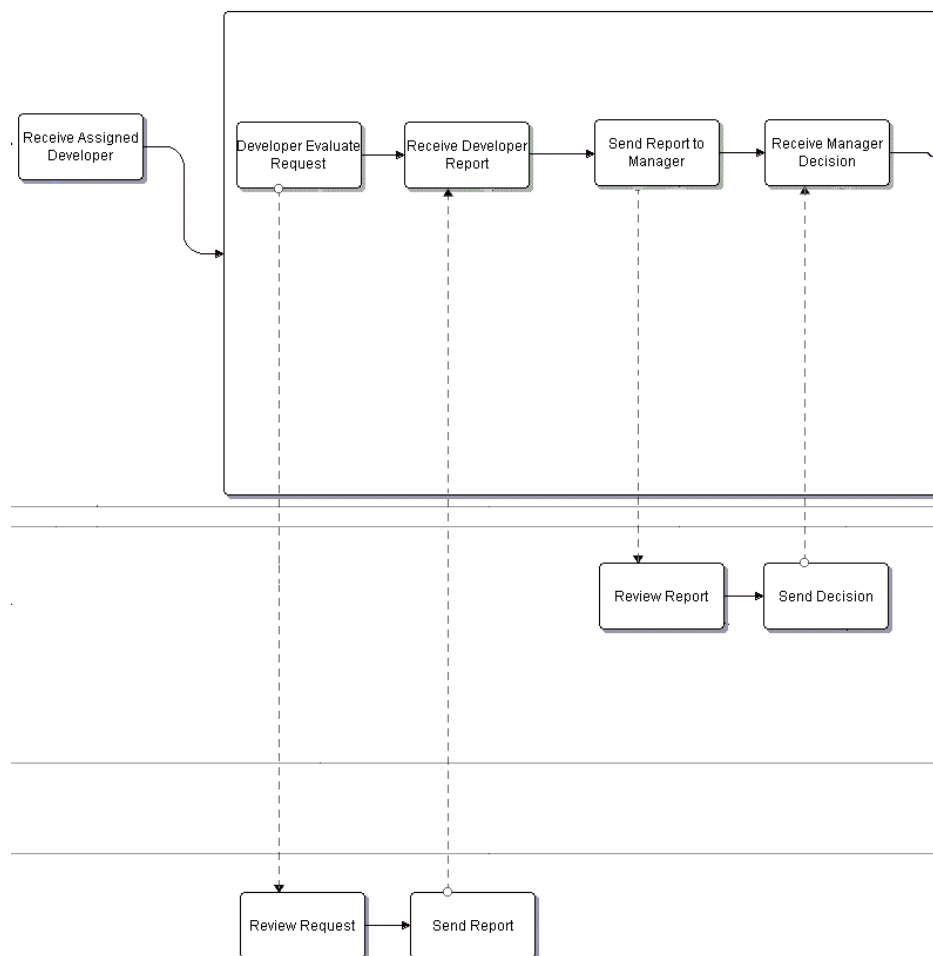


Figura E-5.Revisión de RFC por parte del Desarrollador y el Encargado del Cambio

El encargado del Cambio decide si el cambio es estándar o urgente. El criterio para establecer que el cambio es estándar es indicando que el cambio tiene un tiempo estimado de 20 horas o menos. Un cambio de urgencia es aquel

que necesita ser llevado en un tiempo mayor de 20 horas y en este caso es necesario indicarle al Iniciador del cambio del tiempo que se tiene estimado para implementar el cambio.

El Encargado del Cambio puede necesitar más detalles en la RFC, por lo que se puede escalar la petición al Iniciador del cambio para que agregue la información necesaria y vuelva a reenviarlo al Encargado.

Otro flujo que se puede tomar en la decisión es reasignar la RFC a otro desarrollador.

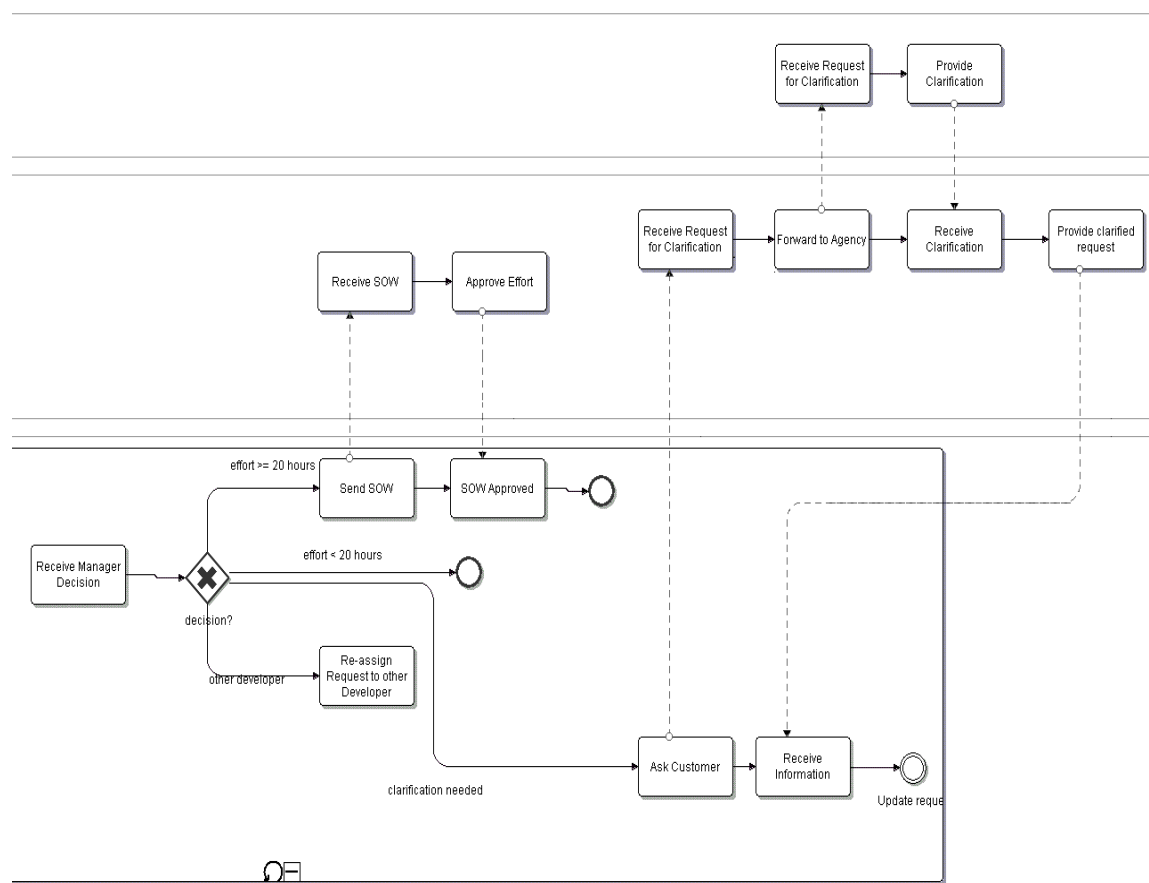


Figura E-6.Proceso de decisión del RFC

Para finalizar el Workflow luego de haber decidido que se va a realizar con la RFC, se asigna a este un desarrollador que llevara a cabo el cambio, cuando este termine el trabajo se le notifica al Encargado de que el proceso fue realizado.

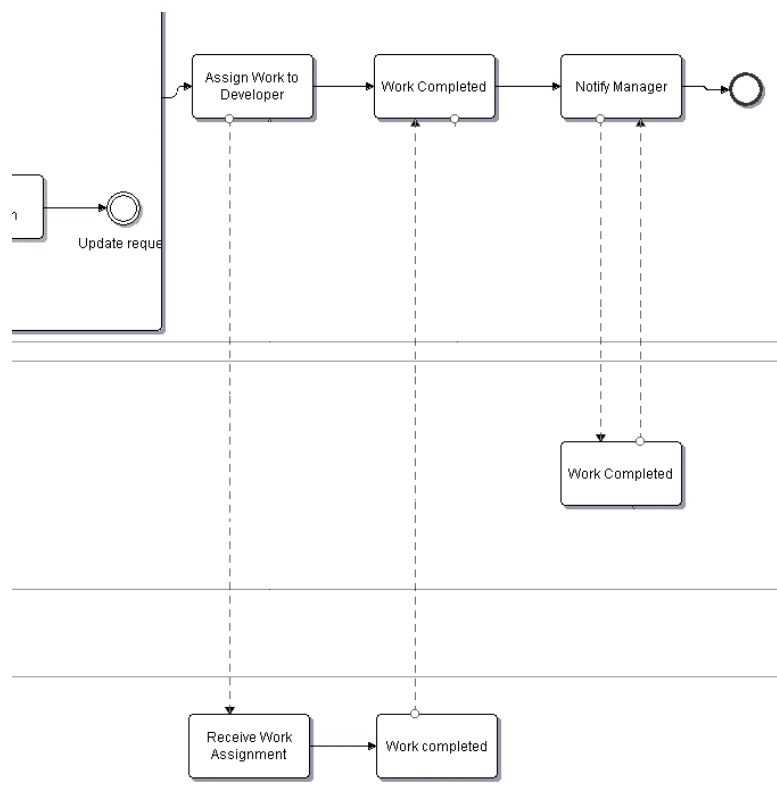


Figura E-7. Proceso de aceptación del RFC

Al finalizar el Workflow, utilizando la notación BPMN que provee Intalio|Designer, cada pool queda con las siguientes tareas:

Agency	Customer	Change Management	Subproceso de Change Management	Manager	Developer
Emitir petición de cambio	Calificar	Recibir petición de cambio	Evaluación de petición por el Desarrollador	Revisar petición	Revisar petición
Recibir petición para clarificación	Petición de cambio	Notificar al encargado	Recibir reporte del Desarrollador	Asignar Desarrollador	Enviar reporte
Proveer	Recibir SOW	Recibir	Enviar reporte	Revisar	Recibir

clarificación		desarrollador asignado	al Encargado	reporte	trabajo asignado
	Aprobar esfuerzo	Asignar trabajo a Desarrollador	Recibir decisión del Encargado	Enviar decisión	Trabajo completado
	Recibir petición para clarificación	Trabajo completado	Enviar SOW	Trabajo completado	
	Enviar a agencia	Notificar al encargado	SOW aprobado		
	Recibir clarificación		Reasignar petición a otro desarrollador		
	Proveer petición clarificada		Preguntar al Cliente		
			Recibir Información		

Tareas de cada pool del diagrama de Gestión de Cambios

Cada tarea del diagrama es representado por un control “task” del diseñador de Intalio.

Adicionalmente de los tasks, en el Workflow existen otros controles BPMN necesarios. La figura N° 38 nos indica los controles utilizados en el modelado del proceso de Gestión de Cambios:

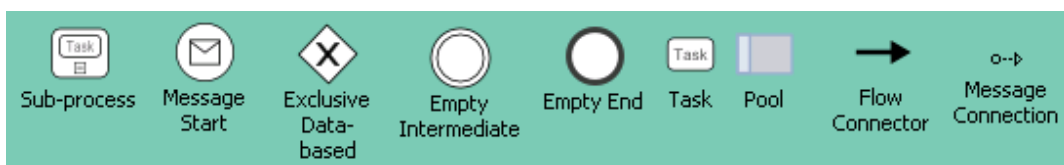


Figura E-8. Controles BPMN usados en el diagrama

Una vez finalizado el modelado del Workflow es necesario crear correlaciones entre instancias de procesos. A continuación, la explicación de las correlaciones en el modelado del Proceso de Gestión de Cambios.

Correlaciones

Un aspecto importante en Intalio BPMS, son las correlaciones. Las correlaciones se usan para identificar explícitamente una instancia de un proceso de negocio. Es un mecanismo a nivel de la aplicación que permite relacionar los mensajes y conversaciones con las instancias de los procesos de negocio a los cuales han sido enviados.

Cuando un proceso de negocio es iniciado, una instancia del mismo es creada, y esta tiene un tiempo de vida. Dentro del engine BPEL, pueden existir múltiples instancias de un proceso de negocio activas al mismo tiempo. Todos los mensajes que son enviados al proceso tienen que ser entregados a la correcta instancia del proceso.

Una correlación es usada para asegurar que un mensaje va a una instancia apropiada basada en el contenido del mensaje. Típicamente, un elemento del mensaje coincide con un valor en la instancia del proceso, para asegurar que el mensaje es enrutado de forma correcta.

Hay que tomar en cuenta que las correlaciones se generaran entre los roles del Iniciador, el Encargado y el Desarrollador todos contra el pool del Proceso de Gestión de Cambios, esto debido a que el proceso en si debe asegurar que los mensajes generados durante su despliegue sean correctamente enlazados a la instancia correspondiente.

El resultado del Workflow muestra 6 correlaciones. La primera es para asegurar que el mensaje de inicio generado por el Iniciador sea procesado por el pool del Proceso de Gestión de Cambios. La figura E-9 nos indica la primera correlación del Workflow.

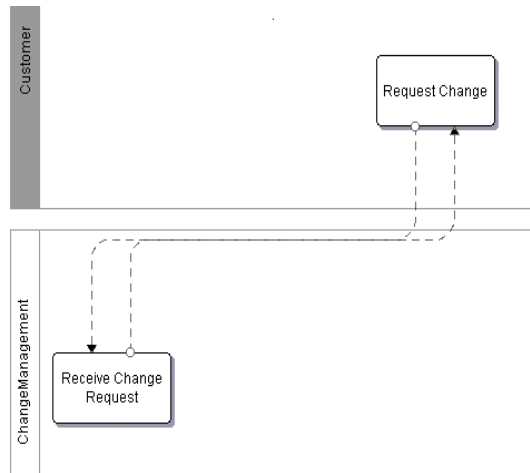


Figura E-9. Correlación entre el Iniciador del Cambio y el proceso Gestión de Cambios

Entre el pool el Proceso de Gestión de Cambios y el Encargado del cambio existe una correlación que indica la creación de la petición y la notificación de que se asignará un desarrollador a dicha petición. La figura E- 10 nos indica la primera correlación entre el pool del Proceso de Gestión de Cambios y el Encargado.

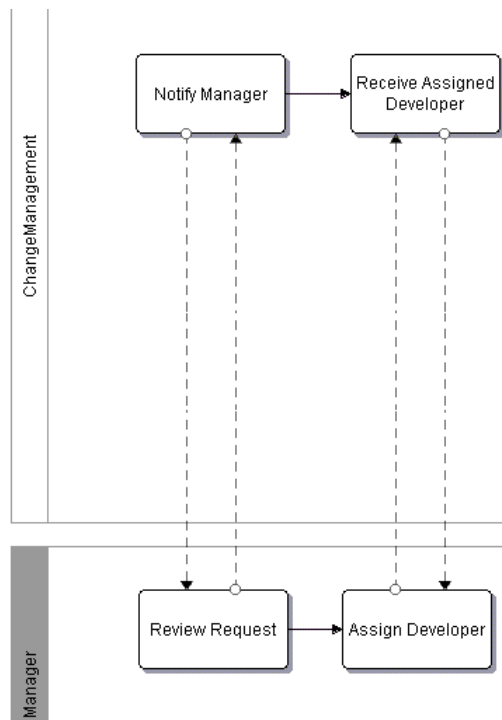


Figura E-10. Correlación entre el proceso Gestión de Cambios y el Encargado

La figura E-11 nos indica la correlación pertenece a la interacción entre el pool del proceso de Gestión de Cambios y el Desarrollador. Acá el desarrollador evalúa la petición y envía un reporte al pool del proceso.

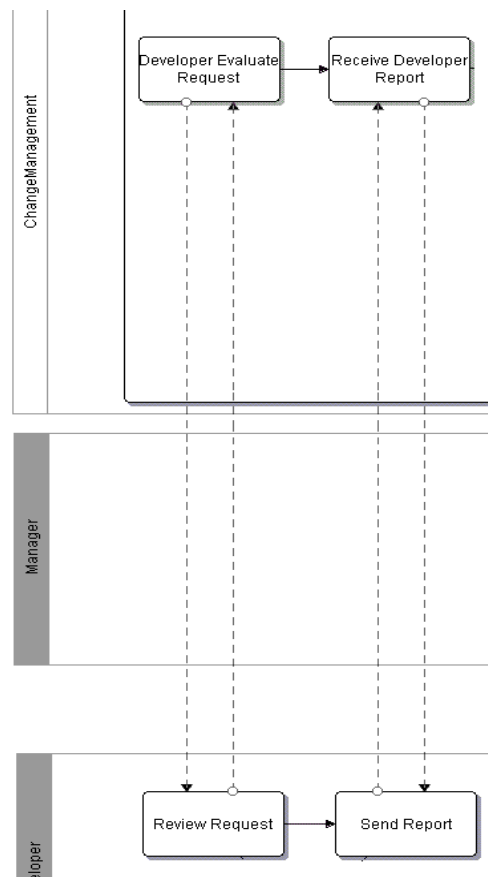


Figura E-11. Correlación entre el proceso Gestión de Cambios y el Desarrollador

Entre el pool del proceso y el Encargado existe una segunda correlación, acá el Encargado revisa el reporte del Desarrollador y comienza el proceso de decisión sobre la petición. La figura E-12 nos indica la segunda correlación entre el pool del Proceso de Gestión de Cambios y el Encargado

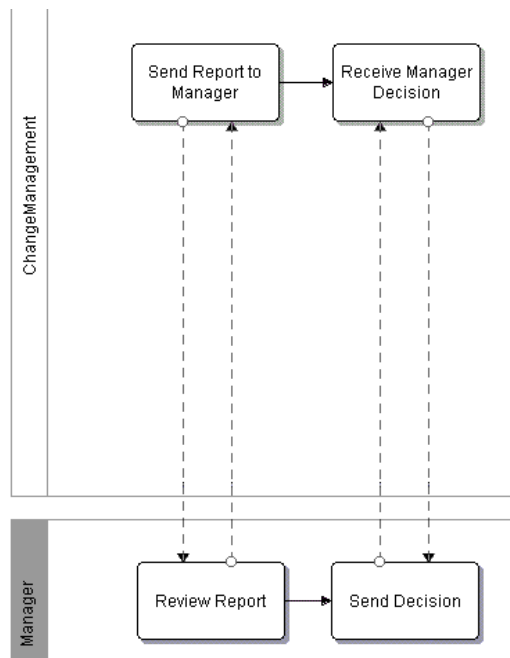


Figura E-12. Segunda correlación entre el proceso Gestión de Cambios y el Encargado

Al momento de entrar en el flujo de la decisión que indica que la prioridad de la petición es urgente se crea la segunda correlación entre el Iniciador y el pool del Proceso de Gestión de Cambios. El Iniciador recibe el informe de tiempo estimado en que se implementará el cambio tal como indica la figura E-13.

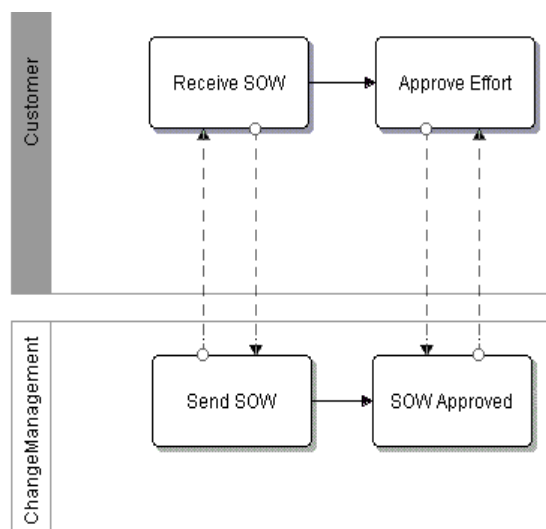


Figura E-13. Segunda correlación entre el proceso Gestión de Cambios y el Iniciador

La figura E-14 nos muestra otro flujo del proceso de decisión al momento de darle prioridad a la petición de cambio. En caso de que se requiera mayor información en la petición el Iniciador debe proveerla.

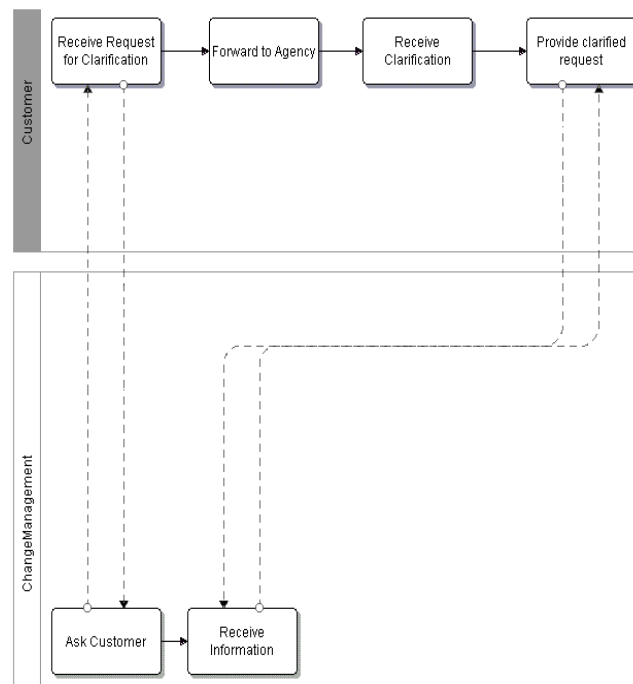


Figura E-14. Tercera correlación entre el proceso Gestión de Cambios y el Iniciador

Una vez aprobada la petición de cambio se le asigna un desarrollador que llevará a cabo el proceso de implementación del cambio, al finalizar este indicará al pool del proceso que el trabajo ha sido completado. La figura E-15 indica como se lleva a cabo esta correlación.

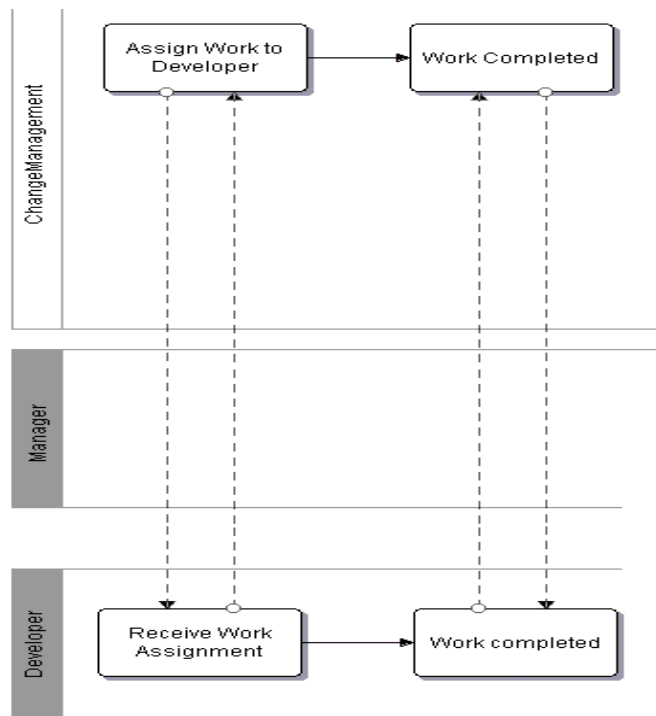


Figura E-15. Segunda correlación entre el proceso Gestión de Cambios y el Desarrollador

La figura E-16 muestra la última correlación entre el pool del Proceso y el Encargado. Acá el Encargado es notificado de que el proceso ha sido finalizado.

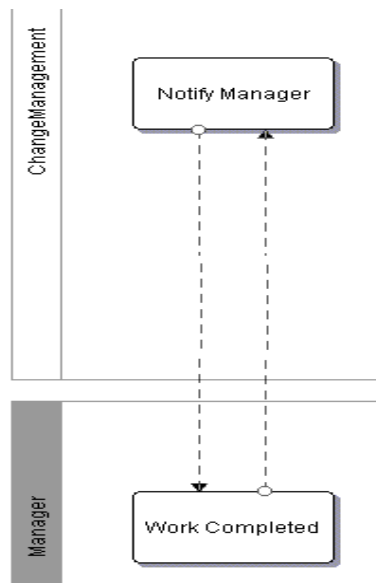


Figura E-16. Tercera correlación entre el proceso Gestión de Cambios y el Encargado

Diseño de formularios

En Intalio BPMS 5.2 se puede crear formularios para representar los procesos modelados a través de XForms.

XForms es un lenguaje de etiquetado para formularios Web, diseñado para ser el sustituto de los formularios tradicionales HTML, y que va a permitir a los desarrolladores la integración con Servicios Web.

Los XForms en Intalio|Designer son elaborados a través del Editor de Formularios (Form Editor). El Editor de Formularios es un componente de Intalio|BPMS que esta embebido en el diseñador. Provee la habilidad de crear gráficamente formularios y generar código en la interfaz de usuario.

Para comenzar a crear los XForms en el diseñador, hacemos click con el botón derecho en el Explorador de Proceso > New > Workflow Form. Este paso lo vemos reflejado en la figura E-17.

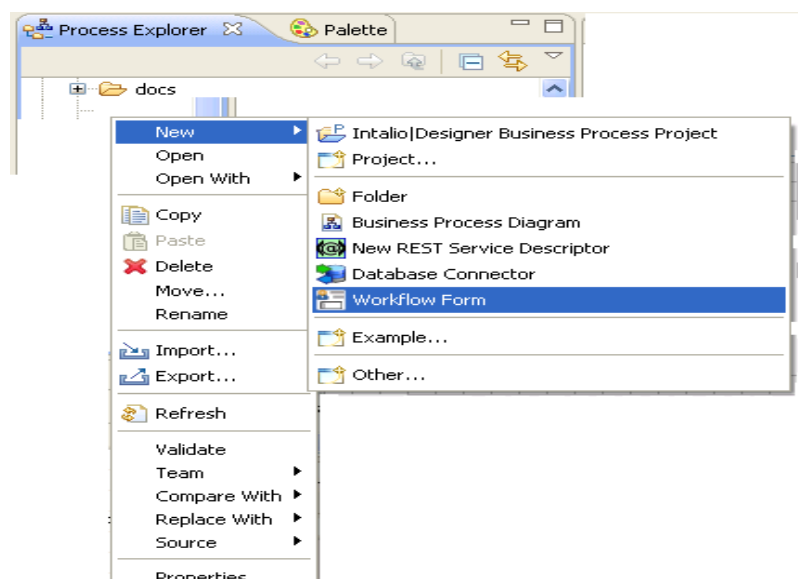


Figura E-17. Ejecución del Workflow Form

A continuación sale una pantalla como la que se refleja en la figura E-18. En “Form file name” proveemos de nombre a los formularios que vamos a crear e inmediatamente aparece el formulario creado y comenzamos a diseñar los formularios según las necesidades.

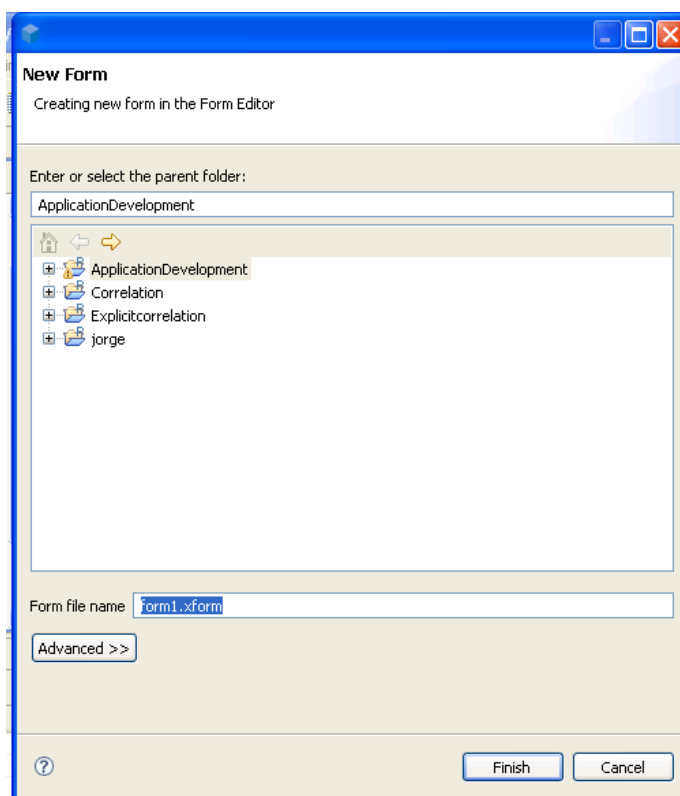


Figura E-18. Creación de nuevo formulario

Para diseñar los formularios es necesario abrir el editor de formularios de Intalio. Para esto vamos a la barra de menú y hacemos click en Window > Open Perspective > Intalio|designer Form Editor. Al lado izquierdo aparece una ventana con los controles para diseñar XForms.

El primer formulario a crear es la Petición de Cambio en si. Este formulario solo tiene un control Text Area que es donde se describirá la petición de cambio. Las propiedades de este control son las siguientes:

- **Input/Output:** in-out
- **Rendered:** true()
- **Trim Input Value:** false
- **Schema type:** acá hacemos click en los puntos suspensivos y se despliega el cuadro de dialogo que muestra la figura E- 19. Seleccionamos tipo de base string.

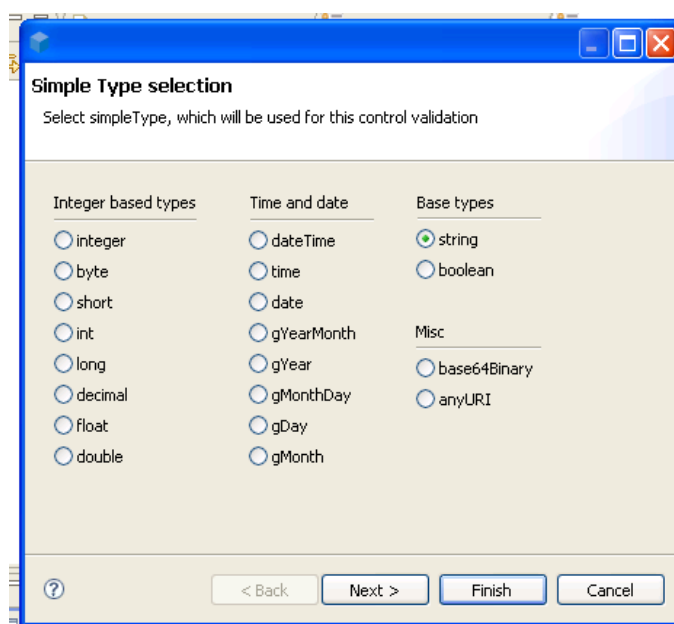


Figura E-19 Selección de Tipo Simple

El formulario CustomerChangeRequest.xform queda como muestra la siguiente figura:

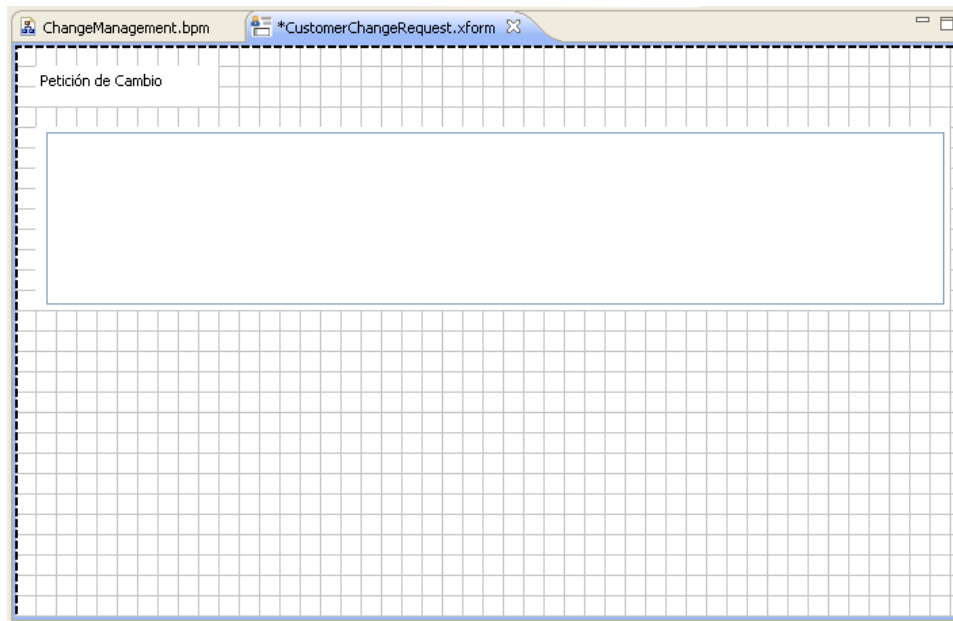


Figura E-20. Formulario CustomerChangeRequest

El siguiente formulario es para asignar el desarrollador a una petición. Este contiene un control Text Area y un Single-select List, en el primero fue donde describió la petición de cambio y en el segundo es donde se seleccionara el desarrollador que se desea asignar para el tratamiento de la RFC.

Las propiedades del control Text Area que debemos modificar son las siguientes:

- **Input/Output:** in
- **Rendered:** true()
- **Trim Input Value:** false
- **Schema type:** tipo de base string.

Las propiedades del control Single-select List que debemos modificar son las siguientes:

- **Input/ Output:** out
- **Rendered:** true

- **Schema type:** tipo de base string
- **Items:** acá hacemos click en los puntos suspensivos y se despliega el cuadro de dialogo que muestra la figura E-21. Agregamos cada una de las instancias para el rol de desarrollador y como valor agregamos la siguiente sentencia: `appdev\<nicknamedel desarrollador>`

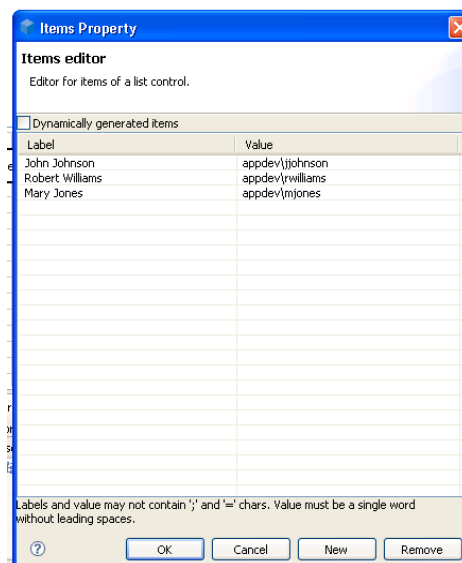


Figura E-21. Ventana Editor de Items

El formulario AssignDeveloper.xform queda como muestra la figura E-22:

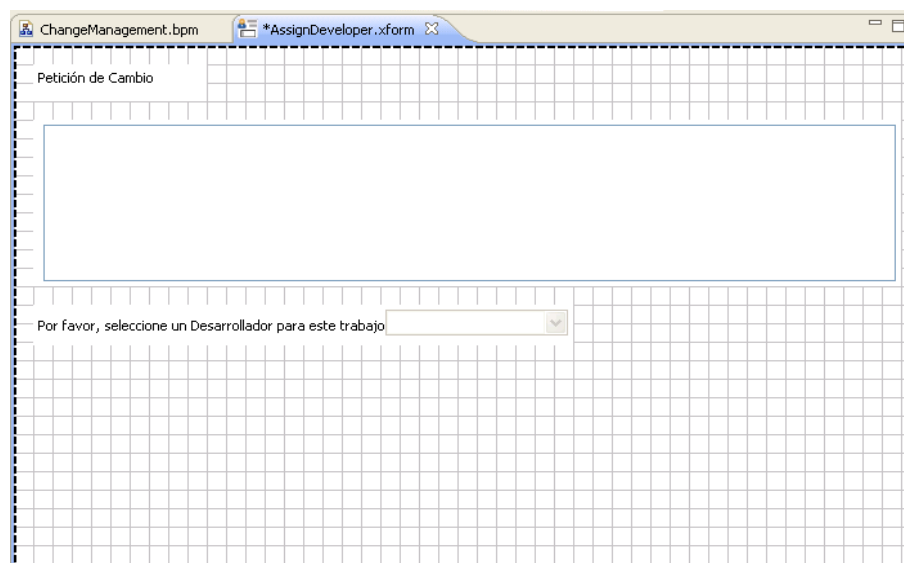


Figura E-22. Formulario AssignDeveloper

El tercer formulario es para indicar el esfuerzo estimado para implementar el cambio. Los controles utilizados en este formulario son 2 Text Area, un Radio Button y un Single-select List.

Un Text Area contiene la petición de cambio previamente almacenada. El Radio Button permite seleccionar la prioridad del cambio. El Single-select List contiene la lista de los desarrolladores que puede clarificar una petición de cambio y el otro Text Area es para describir la clarificación que se necesite en caso de ser necesario.

Las propiedades del primer Text Area que debemos modificar son las siguientes:

- **Input/Output:** in
- **Rendered:** true()
- **Trim Input Value:** false
- **Schema type:** tipo de base string.

Las propiedades del control Single-select List que debemos modificar son las siguientes:

- **Input/Output:** in-out
- **Rendered:** true
- **Schema type:** tipo de base string
- **Items:** acá hacemos click en los puntos suspensivos y se despliega un cuadro de dialogo..

Las propiedades del control Radio Button que vamos a editar son:

- **Input/Output:** out
- **Rendered:** true
- **Schema type:** tipo de base string

- **Items:** desplegamos el cuadro de dialogo que muestra la figura E-23. Agregamos cada una de las instancias para el la prioridad que se le dará a la petición, Mayor o igual a 20 horas si es un cambio urgente; Menor a 20 horas si es un cambio estándar o Desconocido sino sabemos que cambio es.

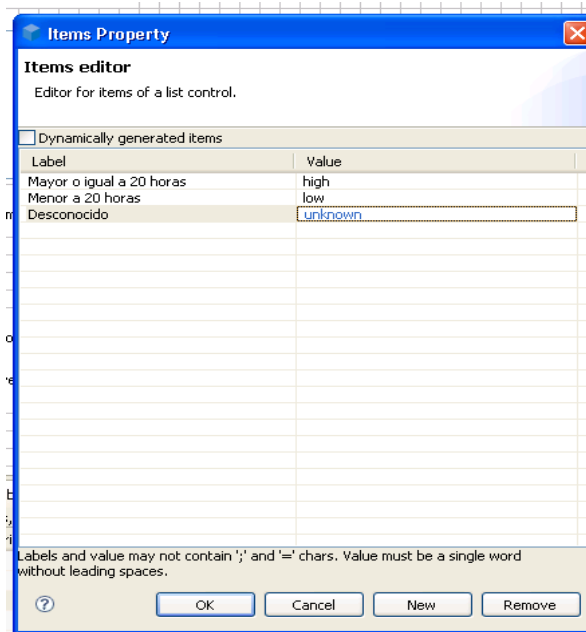


Figura E-23. Ventana de Editor de ítems para el formulario Estimation.xform

Para el siguiente Tex Area editamos las siguientes características:

- **Input/Output:** in
- **Rendered:** true()
- **Trim Input Value:** false
- **Schema type:** tipo de base string.

El formulario Estimation.xform queda como muestra la figura E-24:

Figura E-24. Formulario Estimation

El siguiente formulario es para indicar la acción a realizar una vez que se ha indicado que prioridad tiene el cambio.

2 Text Area, 2 Radio Button, 1 Single-select List y un Text Input son los controles de este XForm.

Un Text Area contiene la petición de cambio previamente almacenada. El Radio Button contiene la prioridad del cambio almacenada. El Single-select List contiene la lista de los desarrolladores que puede clarificar una petición de cambio y el otro Text Area es para describir la clarificación que se necesite en caso de ser necesario. Para la decisión tomada se utiliza otro Radio Button y para el Estimado de trabajo (SOW) se usa un Text Input.

Del nuevo control Radio Button, se modifica las siguientes propiedades:

Las propiedades del control Radio Button que vamos a editar son:

- **Input/Output:** out
- **Rendered:** true
- **Schema type:** tipo de base string

- **Items:** desplegamos el cuadro de dialogo para agregar cada uno de los valores de la decisión que se vaya a tomar.

La figura E-25 nos muestra el diseño final del formulario Report.xform.

Figura E-25. Formulario Report

El quinto formulario es para que el Iniciador indique si el estimado para la implementación del cambio es aceptable. Un Text Area y un Radio Button son los controles con que se diseñó este formulario y sus propiedades que se editaron fueron las siguientes.

TextArea:

- **Input/Output:** in
- **Rendered:** true()
- **Trim Input Value:** false
- **Schema type:** tipo de base string.

Radio Button:

- **Input/Output:** out
- **Rendered:** true
- **Schema type:** tipo de base string
- **Items:** desplegamos el cuadro de dialogo para agregar las opciones sobre si se esta de acuerdo o no con el estimado.

El diseño del formulario estimado de trabajo (SOW) queda como nos muestra la figura E-26

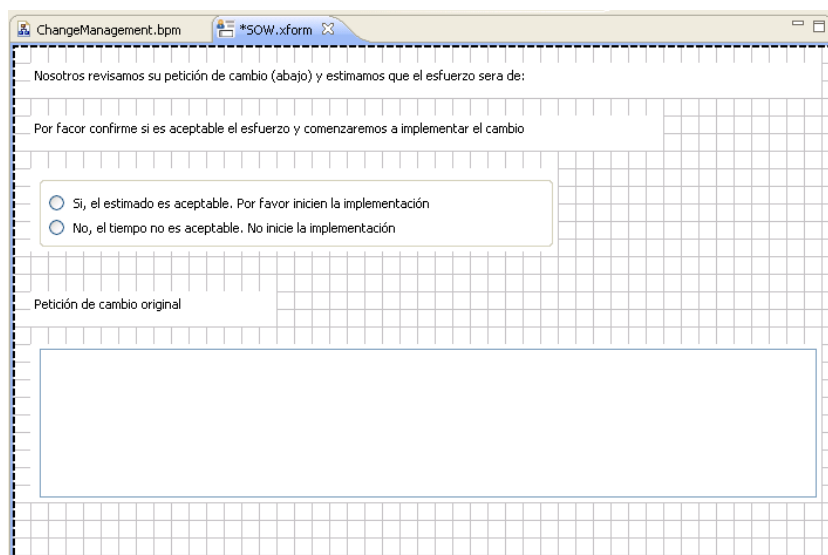
The image shows a screenshot of a web browser window. The browser has two tabs: 'ChangeManagement.bpm' and '*SOW.xform'. The form content is as follows:
- A text box containing: "Nosotros revisamos su petición de cambio (abajo) y estimamos que el esfuerzo sera de:"
- A text box containing: "Por favor confirme si es aceptable el esfuerzo y comenzaremos a implementar el cambio"
- A group box containing two radio buttons:
 - Si, el estimado es aceptable. Por favor inicien la implementación
 - No, el tiempo no es aceptable. No inicie la implementación
- A text box containing: "Petición de cambio original"
- A large, empty text area for input.

Figura E-26 Formulario SOW

Se crea otro formulario que indica que la petición de cambio ya esta completa y se le puede asignar a un desarrollador para que la lleve a cambio.

Este formulario tiene un control Text Area que sirve para reflejar la descripción del cambio que se va a llevar a cabo. Las propiedades del Text Area del formulario Work.xform que son modificadas son las siguientes:

- **Input/Output:** in
- **Rendered:** true()
- **Trim Input Value:** false
- **Schema type:** tipo de base string

El formulario Work.xform queda finalmente como muestra la figura E- 27:

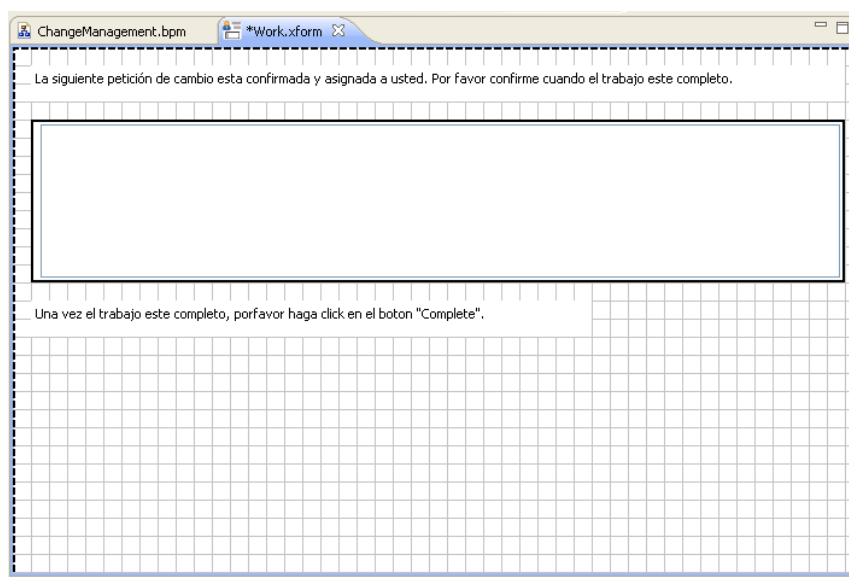


Figura E-27. Formulario Work

Para finalizar, el último XForm se diseñó para indicar que el trabajo ha sido implementado. El formulario WorkCompleted queda como lo indica la siguiente figura.

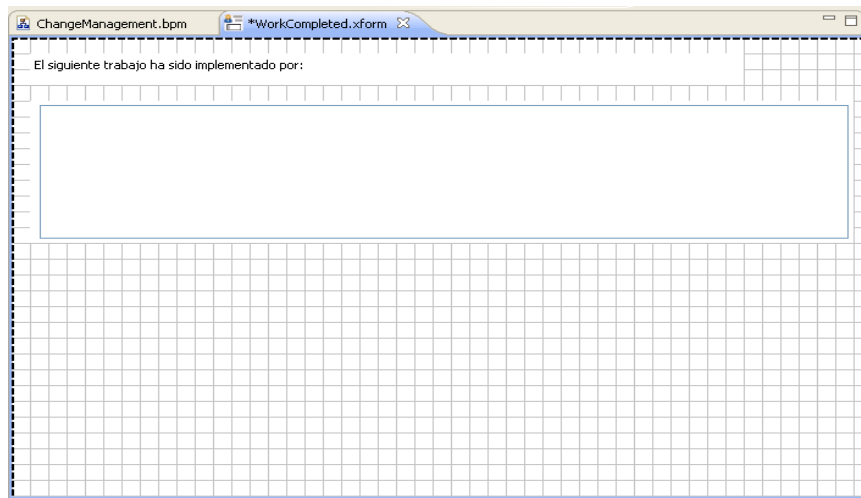


Figura E-28. Formulario WorkCompleted

Diseño de Esquemas XML

Un esquema XML representa la manera de documentar la estructura de un documento XML. Intalio|Designer permite generar un esquema XML que corresponde a una estructura de datos.

Para crear un esquema XML se trabajó con las Herramienta de Plataformas Web de Eclipse (Web Tools Platform - WTP), que incluye un editor gráfico de esquemas XML (XML schemas) y de Lenguaje de Definición de Servicios Web (Web Services Definition Language).

Sobre el primer formulario creado, hacemos click con el botón derecho y le damos en New > Other. A continuación aparece la pantalla reflejada en la figura E-29.

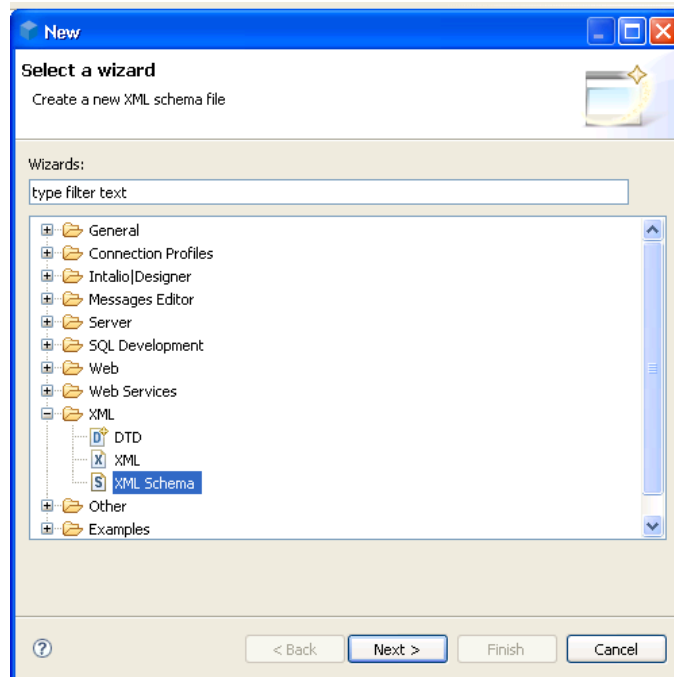


Figura E-29. Wizard para crear un esquema XML

Luego hacemos click en el botón Next y le damos el nombre al esquema creado. Para verificar que el esquema XML fue creado, nos dirigimos al explorador de procesos y exactamente debajo del formulario debe aparecer asociado el nuevo esquema XML creado con el siguiente formato:

< nombre_del_archivo.xform.xsd >

Al hacerle click a este archivo .xsd creado se despliega la ventana representada por la figura E-30

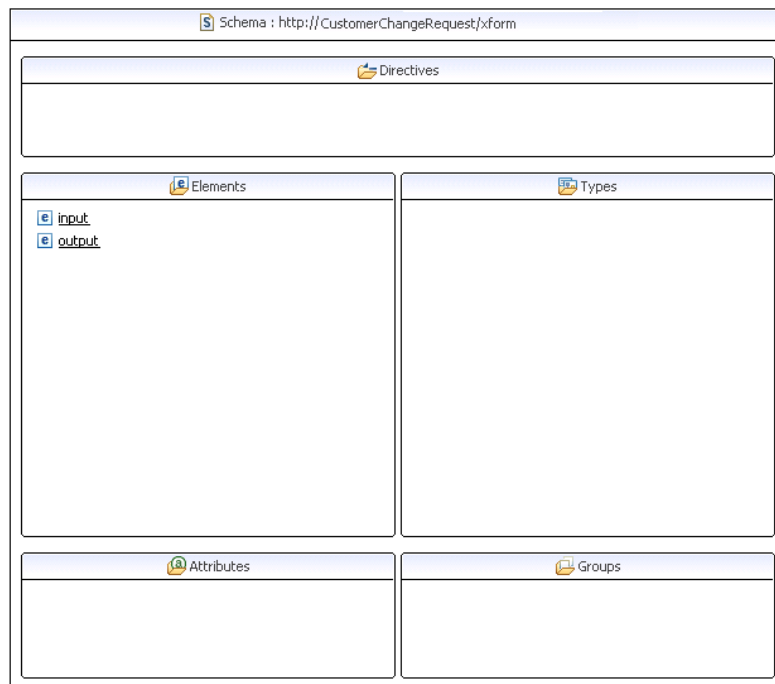


Figura E-30. Esquema XML del formulario CustomerChangeRequest

Independientemente del formulario que se le asocie a un esquema XML, este genera dos tipos de elementos simples: input y output.

Un elemento simple es un elemento XML que puede contener solo texto. No puede contener cualquier otro tipo de elementos o atributos.

Sin embargo, la restricción de "solo texto" es absolutamente engañosa. El texto puede contener diferentes tipos incluidos en la definición de los esquemas XML (booleano, string, fecha, etc).

Al hacer click en cada elemento de la estructura aparece los controles (Text Area, Radio Button, Text Input y Label) del formulario que hemos definidos como de entrada (in) y salida (out). Para los controles definidos como in-out aparecen en ambos tipos de elementos.

En el caso del elemento tipo in del formulario CustomerChangeRequest.xform nos demuestra los controles ChangeRequestLabel que es un label y ChangeRequest que es Text Input. Esto lo podemos observar en la figura E-31.

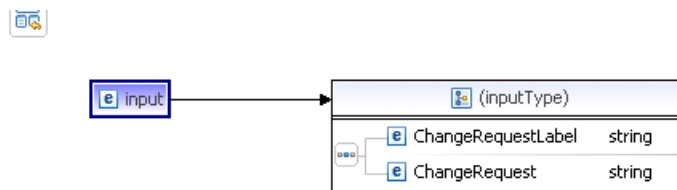


Figura E-31. Elemento input del esquema XML del formulario CustomerChangeRequest

Para el elemento tipo out del formulario CustomerChangeRequest.xform nos demuestra el control ChangeRequest que es Text Input que también es output.

Adicionalmente, a los elementos definidos en el formulario como tipo output, al generar el esquema .xsd de cada formulario Intalio|Designer genera automáticamente los siguientes atributos:

- **taskId:** esta propiedad define un identificador único para cada tarea y se fija automáticamente al momento de ejecución.
- **participantToken:** es un campo usado por tempo para propósitos de sincronización. Un participantToken se puede obtener del servicio Security.xml.
- **user:** define el usuario y/o rol que se esta ejecutando en determinado momento.
- **formUrl:** esta propiedad define la ruta donde, al momento de ejecutarse, se localizara el formulario que apoya a la tarea.

La figura E- 32 refleja las propiedades del elemento output del formulario CustomerChangeRequest

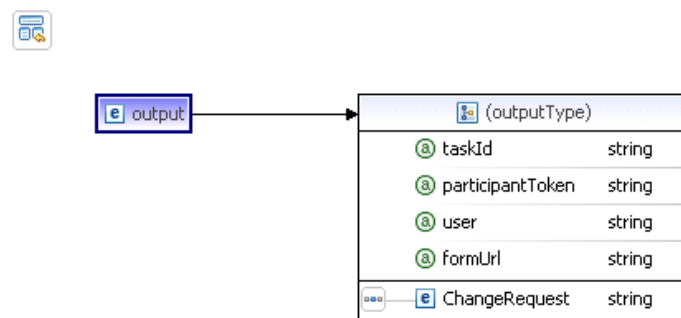


Figura E-32. Elemento output del esquema XML del formulario CustomerChangeRequest

Estas propiedades pertenecen a la metadata de las diferentes tareas que son generadas por actividades humanas y son sincronizadas por Tempo.

Al crear el esquema XML del formulario CustomerChangeRequest, Intalio genera código XML con las especificaciones del formulario en cuestión. Las siguientes líneas representan al esquema XML del formulario CustomerChangeRequest:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:fe="http://workspace5/CustomerChangeRequest/xform"
  xmlns:xform=" http://workspace5/CustomerChangeRequest/xform "
  targetNamespace=" http://workspace5/CustomerChangeRequest/xform "
  elementFormDefault="qualified">
  <xs:element name="output">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ChangeRequest" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="taskId" type="xs:string"/>
      <xs:attribute name="participantToken" type="xs:string"/>
      <xs:attribute name="user" type="xs:string"/>
      <xs:attribute name="formUrl" type="xs:string"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="input">
    <xs:complexType>
```

```

        <xs:sequence>
            <xs:element name="ChangeRequestLabel" type="xs:string"/>
            <xs:element name="ChangeRequest" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>

```

Donde las siguientes líneas de código son la raíz del esquema XML

```

<?xml version="1.0" encoding="UTF-8"?>
.
.
.
</xs:schema>

```

La segunda da línea de código indica que los elementos usados en el esquema viene del namespace

```
"<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

El fragmento:

```
targetNamespace=" http://workspace5/CustomerChangeRequest/xform "
```

Indica que los elementos definidos por el esquema (ChangeRequest, ChangeRequestLabel) vienen del namespace

```
http://workspace5/CustomerChangeRequest/xform"
```

Las siguientes líneas indican los tipos de elementos presentes en el esquema XML:

```

<xs:element name="output">
.
.
.
<xs:element name="input">

```


Los atributos de metadata pertenecientes al elemento output son representados en XML por las siguientes líneas

```
<xs:attribute name="taskId" type="xs:string"/>
<xs:attribute name="participantToken" type="xs:string"/>
<xs:attribute name="user" type="xs:string"/>
<xs:attribute name="formUrl" type="xs:string"/>
```

El fragmento:

```
elementFormDefault="qualified">
```

Indica que cualquier elemento usado por la instancia del documento XML que fuese declarado en este esquema debe ser calificado para el namespace.

Cada uno de los 6 formularios restante tiene asociado su esquema XML con elementos Input y Output, al igual que le formulario CustomerChangeRequest.

Los atributos de los diferentes esquemas XML asociado a cada formulario se reflejan en la siguiente tabla:

Formulario	Elementos Input	Elementos Output	Elementos Input/Output
AssignDeveloper	ChangeRequest ChangeRequestLabel	Developer	X
CustomerChangeRequest	ChangeRequest ChangeRequestLabel	ChangeRequest	ChangeRequest
Estimation	DeveloperToAsk ChangeRequest ChangeRequestLabel Label1	Estimado DeveloperToAsk	DeveloperToAsk
Report	DecisionLabel ChangeRequest ChangeRequestLabel Estimado DeveloperToAsk	SOWestimado Decision	X
SOW	ChangeRequest ChangeRequestLabel Text Text2 SOWestimado	Decision	X
	Text2		

Work	ChangeRequest Text	X	X
WorkCompleted	ChangeRequest Developer Text	Text	Text
Todos los esquemas XML de cada formulario tienen siguientes atributos: taskId, participantToken, user y formUrl, que fueron generados automáticamente			

Elementos de los esquemas .xsd de los formularios.

Enlace entre tareas y formularios

Intalio|Designer permite enlazar tareas con sus respectivos formularios con la característica de “Drag and Drop”. De esta manera se le asigna un rol a cada formulario al momento de ejecución.

Cada formulario creado debe ser arrastrado a cada tarea creado para los roles participantes en el proceso (Iniciador, Encargado, Desarrollador). Al arrastrar un formulario sobre una tarea nos aparece el siguiente mensaje: “Attach this file (nombredel_formulario.xform as documentation”

La siguiente tabla nos indica las tareas y los formularios adjuntados a cada formulario:

Formulario	Tarea(s)	Actor encargado
PeticionCambioIniciador	Request Change	Iniciador
AsignarDesarrollador	Review request Assign developer	Encargado
Estimacion	Review Request Send Report	Desarrollador
Reporte	Review report Send decision	Encargado
SOW	Receive SOW Approve effort	Iniciador
Clarificacion	Receive request for clarification Provide clarified request	Iniciador
Work	Receive Work Assignment	Desarrollador

	Work Completed	
WorkCompletado	Work Completed	Encargado

Enlace entre tareas y formularios.

Al adjuntar un formulario a una tarea se hereda la estructura XML creada en el paso anterior. La figura E-33 refleja como queda el formulario CustomerChangeRequest adjuntado a la tarea de Petición de Cambio.

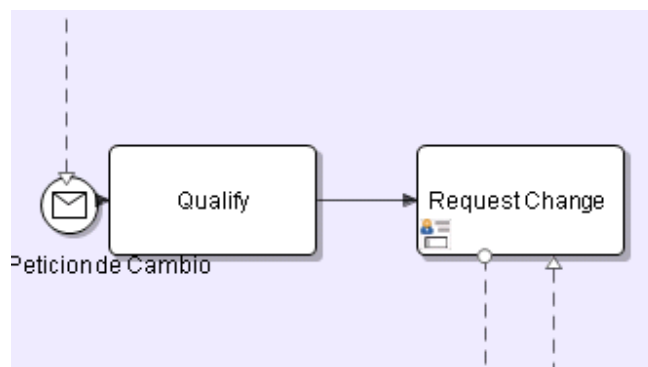


Figura E-33. Formulario adjuntado a la tarea de Petición de Cambio

Consumo de Web Services

Un aspecto importante en la utilización de BPM mediante Intalio, es su capacidad para unir los conceptos de Web Services y Workflow en un proceso de negocio.

El proceso modelado en este Trabajo Especial de Grado tiene la responsabilidad de notificar, enviar, escalar y crear peticiones de cambio entre diversos actores (Iniciador de Cambio, Gestor de Cambio, Desarrollador de versiones).

Los Web Services que se utilizan en Intalio Community se encuentran en el contenedor Axis2. Al momento de levantar el servidor de Intalio este modulo es

invocado por el servidor Gerónimo. Este modulo es el siguiente:
com.intalio.bpms/axis2-services/1.1.1/war.

Para crear la estructura de Web Services nos dirigimos al explorador de procesos y hacemos click con el botón derecho sobre el formulario que deseamos crear todos los archivos de Servicios Web; en el menú contextual hacemos click en: New > Other.

Al hacerle click acá nos aparece el cuadro de dialogo mostrado en la figura E-34.

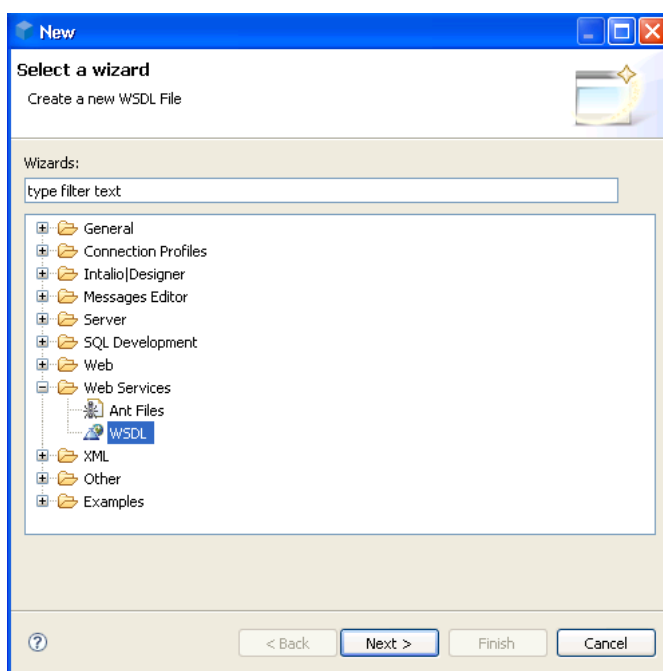


Figura E-34. Wizard para crear estructura de Servicios Web

Luego le colocamos el nombre a la estructura de wsdl que deseamos para ese formulario y nos aparece en la carpeta Build del proyecto los siguientes componentes representados en la figura E-35.

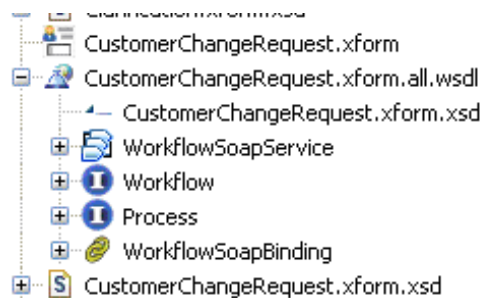


Figura E-35. Estructura wsdL para el formulario CustomerChangeRequest

Mediante el uso del módulo de Axis2, Intalio BPMS 5.2 provee una estructura de Servicios Web para ser manipulados en el Workflow de los procesos y durante la ejecución del modelado del proceso.

Al crear la estructura de Web Services para un formulario, Intalio|Designer genera 4 tipos de módulos WSDL (Lenguaje de Descripción de Servicios Web - Web Services Description Language): WorkflowSoapService, Workflow, Process y el WorkflowSoapBinding.

A continuación se describe cada uno de los módulos .wsdl generados para el manejo de los servicios web.

WorkflowSoapService

Este módulo .wsdl contiene los servicios web que permiten mostrar los formularios creados en el workflow a través de la interfaz xformPort. Esta interfaz permite la comunicación entre los formularios y la dirección del localhost de la consola de Intalio.

La comunicación se realiza utilizando el protocolo de acceso de objetos simples (SOAP - Simple Object Access Protocol) proyectando los formularios con

toda su información a la siguiente dirección:
<http://localhost:8080/fds/workflow/xform>

El WorkflowSoapService proporciona tres tipos de servicios que se pueden ver desplegados a través de la figura E-36:

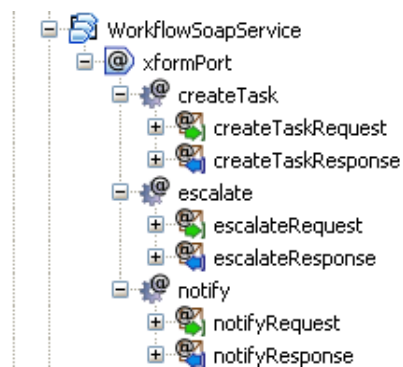











Figura E-36. Servicios web del WorkflowSoapService

-  **createTask**: web service destinado a la creación de tareas que van dirigidas desde las tareas del pool ejecutable del proceso a las pools de los actores (Iniciador, Desarrollador, Encargado).
Este servicio provee dos operaciones:
 -  **createTaskRequest**: esta operación genera un mensaje de entrada para solicitar la creación de una tarea usando la información del formulario asignado.
 -  **createTaskResponse** : esta operación genera un mensaje de salida indicando que la creación la tarea solicitada fue realizada.
-  **escalate** : web service dedicado a la escalación de tareas entre pools no ejecutables, es decir entre diferentes actores. Proporciona dos tipos de operaciones:

- ∅  `escalateRequest`: esta operación genera un mensaje de entrada para solicitar la escalación de una tarea entre pools.
 - ∅  `escalateResponse` : esta operación genera un mensaje de salida indicando que la escalación de la tarea solicitada fue ejecutada.
-  `notify` : este servicio web permite crear notificaciones de ejecuciones de tareas. Sus operaciones son:
 - ∅  `notifyRequest` : esta operación genera un mensaje de entrada para solicitar la una petición de notificación de tarea ejecutada.
 - ∅  `notifyResponse` : operación que genera un mensaje de salida para indicar que la solicitud de notificación de la ejecución de una tarea fue realizada.

Workflow

El workflow es el segundo modulo .wsdl cuyos servicios web contiene las operaciones que permite la creación, la escalación y la notificación de tareas en el momento de ejecución del modelado.

La figura E-37 nos brinda la vista de los tres tipos de servicios del Workflow:

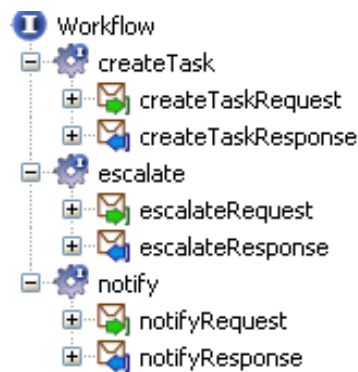











Figura E-37. Servicios web del Workflow

-  **createTask** : se encarga de crear tareas al momento de la ejecución del proceso modelado. Sus operaciones son las siguientes:
 - ∅  **createTaskRequest** : esta operación genera un mensaje de entrada para solicitar la creación de una tarea en momento de ejecución desplegado por la consola de Intalio.
 - ∅  **createTaskResponse** : se crea un mensaje de salida indicando que la solicitud de creación de tarea fue ejecutada correctamente en la consola.

-  **escalate**: operación que escala tareas en el proceso modelado cuando se encuentra en ejecución. Para realizar esto necesita las siguientes operaciones:
 - ∅  **escalateRequest**: esta operación genera un mensaje de entrada para solicitar la escalación de una tarea en ejecución.
 - ∅  **escalateResponse** : se crea un mensaje de salida para indicar que la escalación fue ejecutada.

-  notify : servicio web que permite notificar cuando una tarea fue creada al momento del despliegue del proceso

∅  notifyRequest : crea un mensaje de entrada solicitando una notificación de que una tarea se ha ejecutado.

∅  notifyResponse : genera un mensaje de salida notificando que la tarea se ha ejecutado.

Process

Este modulo del .wsdl proporciona la interfaz entre el proceso modelado y los servicios web. Permite el mapeo de mensajes directamente sobre el diagrama modelado desde los diferentes pools de los diferentes roles hacia el proceso ejecutable. Las operaciones del modulo Process se visualizan mediante la figura E-38

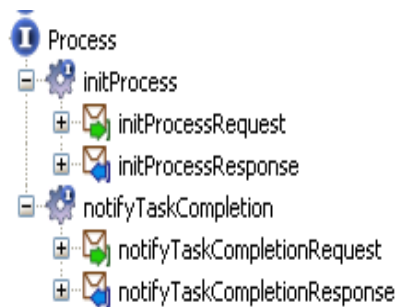








Figura E-38. Servicios web del Process

-  initProcess : el servicio es invocado por las tareas que forman parte de las correlaciones modeladas para asegurar que el envío de mensajes entre los pools no ejecutables y el pool ejecutable del proceso se realice de la manera esperada.

- ∅  `initProcessRequest`: esta operación es el primer web service que se utiliza cuando se modela, ya que este solicita el inicio de un proceso, por lo tanto va adjuntado en la primera tarea junto con el primer formulario.
 - ∅  `initProcessResponse` : la operación de salida indica que la solicitud de inicio de proceso fue confirmada.
-  `notifyTaskCompletion` : este servicio asegura que el mensaje de notificación en una correlación llegue al proceso ejecutable.
 - ∅  `notifyTaskCompletionRequest`: esta operación permite solicitar una notificación de que la tarea fue completada.
 - ∅  `notifyTaskCompletionResponse` : es la operación de confirmación de la operación anterior.

WorkflowSoapBinding

Se especifica la codificación de los mensajes, y los enlaces a protocolos de todas las operaciones y mensajes definida en un port type. Esto quiere, que define los enlaces entre el WorkflowSoapService y el Workflow. Los enlaces o bindings a servicios de este modulo se muestran en la figura E-39:

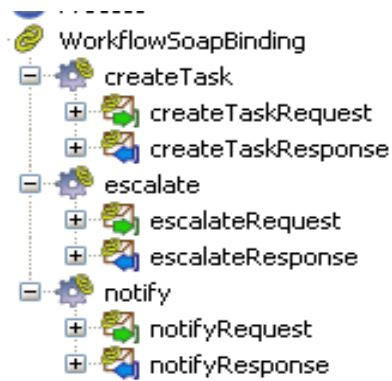











Figura E-39. Operaciones del WorkflowSoapBinding

-  **createTask** : es la operación que se encarga de hacer los bindings de la creación de tarea en el WorkflowSoapService
-  **createTaskRequest** : realiza el enlace entre todos los mensajes de petición de creación de tareas y el xformPort.
-  **createTaskResponse** : ejecuta el binding entre las repuestas de las peticiones de creación de tareas y el xformPort.
-  **escalate** : la función de esta operación es realizar el enlace de las tareas de escalación al xformPort del WorkflowSoapService.
-  **escalateRequest** : realiza el enlace entre todos los mensajes de entrada para la petición de escalación en el WorkflowSoapService.
-  **escalateResponse** : ejecuta el binding entre los mensajes de salida de respuesta de la escalación al xformPort.
-  **notify** : esta operación se encarga de hacer los enlaces de las notificaciones con el WorkflowSoapService.

- ∅  notifyRequest : realiza el enlace las peticiones de notificaciones en el xformPort.
- ∅  notifyResponse : ejecuta el binding entre las respuestas de las notificaciones y el WorkflowSoapService.

Toda la estructura wsdl es generada para cada uno de los formularios diseñados y puede ser visualizada mediante el diseñador gráfico, como lo muestra la figura E-40.

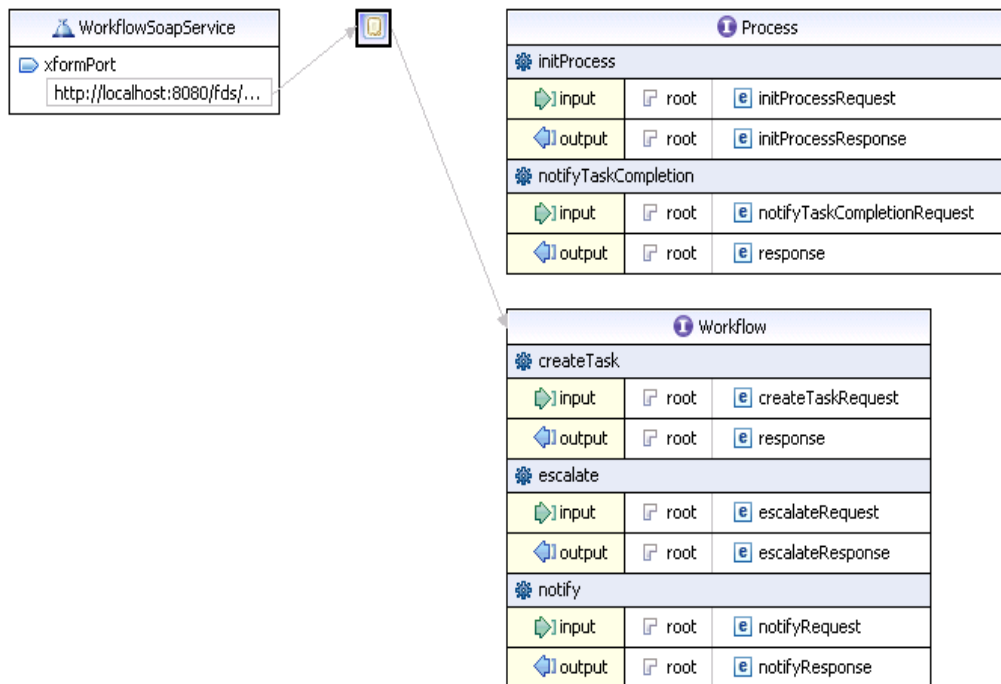


Figura E-40. Estructura gráfica de los módulos wsdl

La figura anterior nos permite visualizar cada modulo con sus respectivas operaciones, descritas anteriormente.

El enlace o “binding” entre el WorkflowSoapService y el Workflow especifica en qué dirección (URI) se accede a la implementación del xformPort y a la vez define el lugar donde se encuentra el servicio.

El funcionamiento y la información de cada uno de los servicios web provistos por Intalio (web services internos) están definidos en una estructura de metadata escrita en WSDL.

Esta metadata describe los formatos de los mensajes necesarios para interactuar con los servicios nombrados anteriormente. Las operaciones y mensajes que soporta se describen en abstracto en formato WSDL y se ligan después al protocolo SOAP y al formato del mensaje en las tareas y correlaciones diseñadas en el diagrama BPMN del proceso.

A continuación se mostrarán las figuras de la metadata involucrada en cada operación de servicio web.

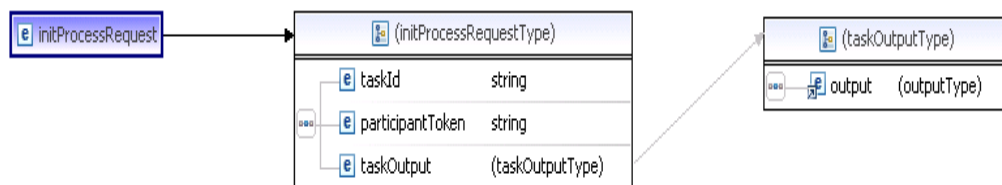


Figura E-41. Metadata del servicio web `initProcessRequest`

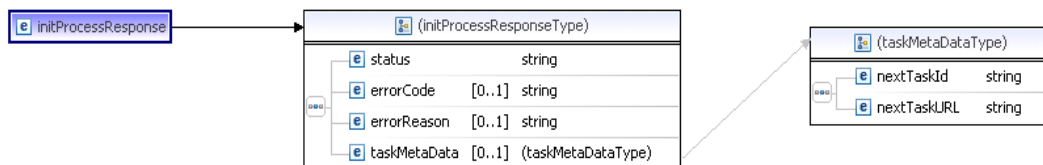


Figura E-42. Metadata del servicio web `initProcessRequest`

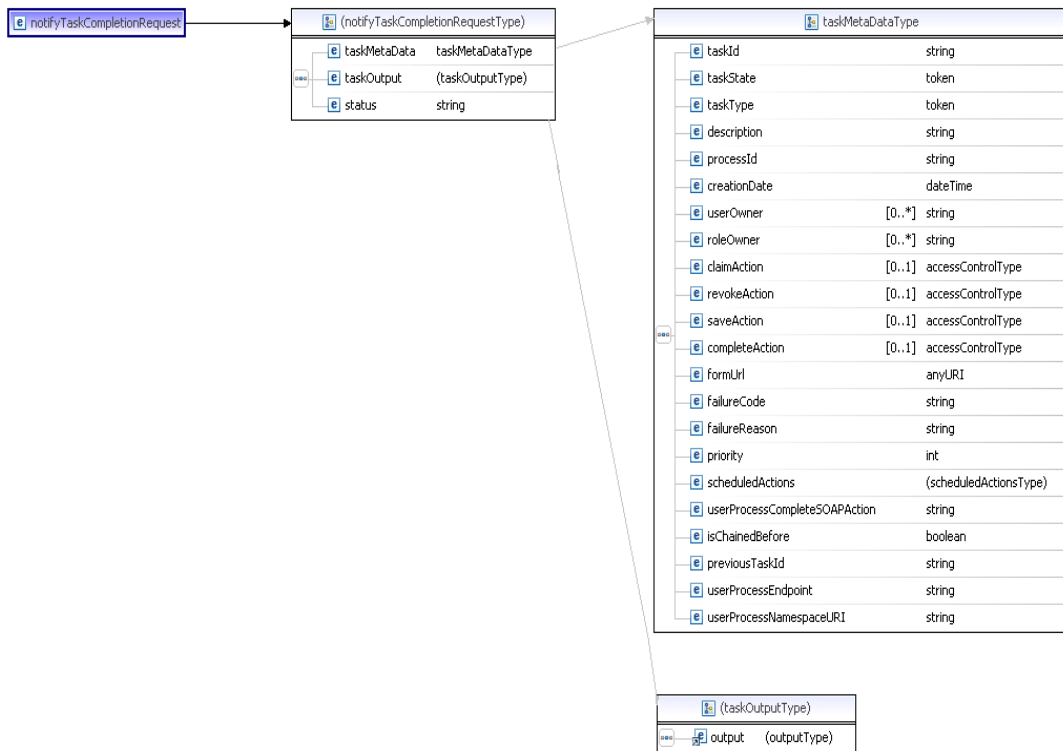


Figura E-43. Metadata del servicio web notifyTaskCompletionRequest

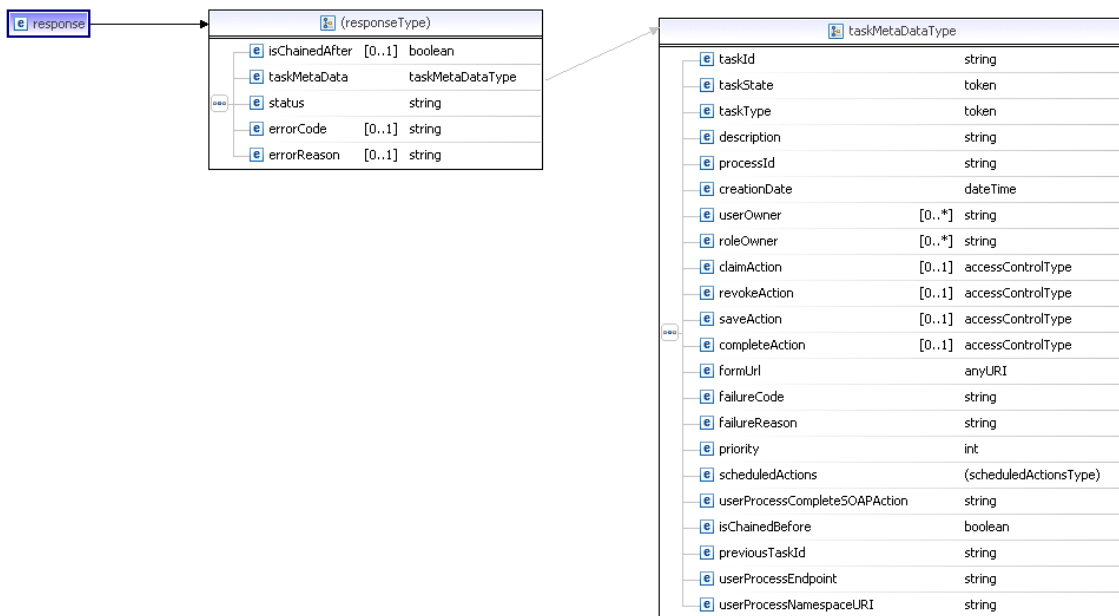


Figura E-44. Metadata del servicio web response

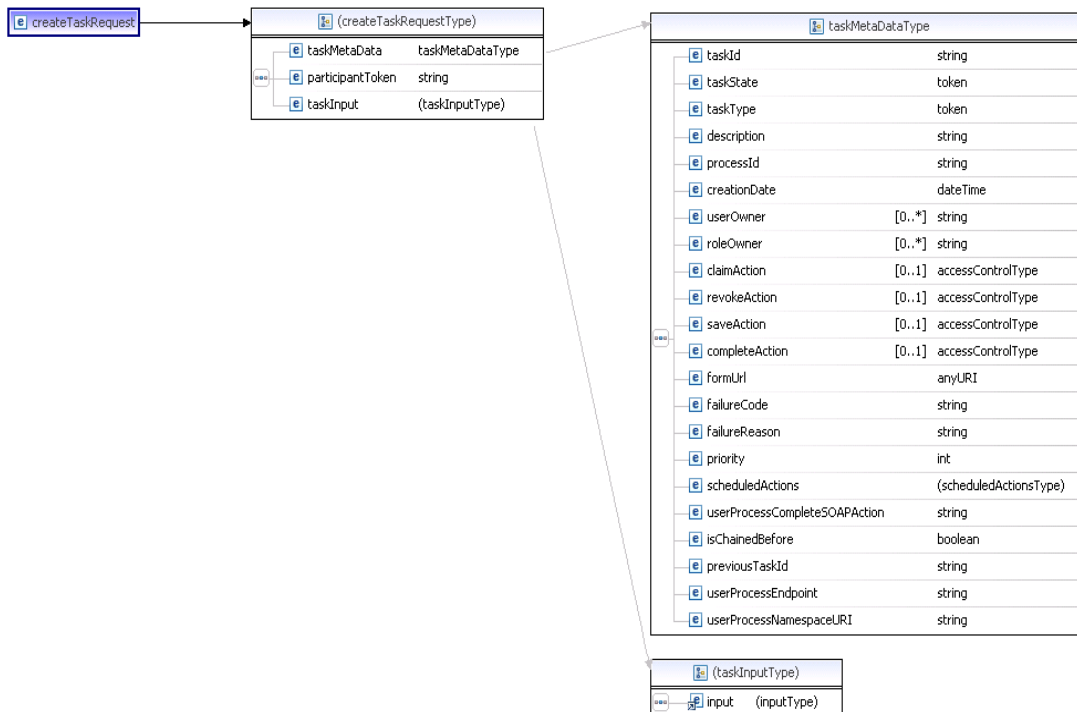


Figura E-45. Metadata del servicio web createTaskRequest

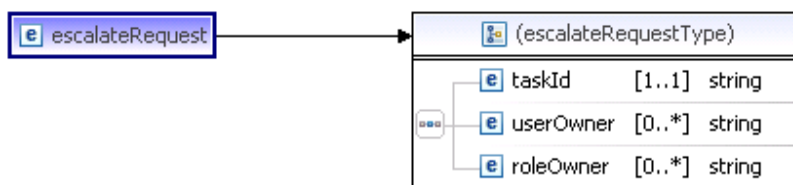


Figura E-46. Metadata del servicio web escalateRequest

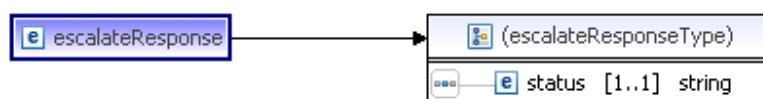


Figura E-47. Metadata del servicio web escalateResponse

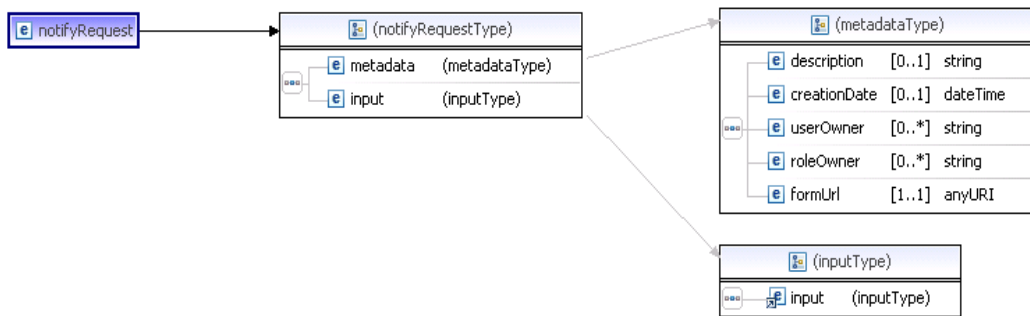


Figura E-48. Metadata del servicio web notifyRequest

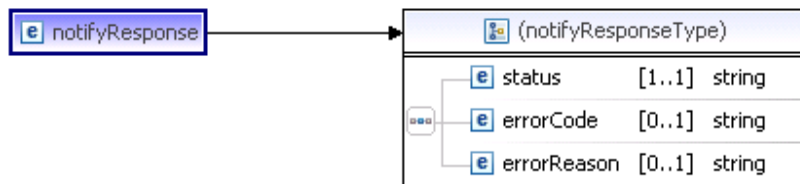


Figura E-49. Metadata del servicio web notifyResponse

Invocación a los de Web Services desde las tareas

Para el modelado del proceso de Gestión de Cambios de este Trabajo Especial de Grado, no se crearon servicios web, solo fueron invocados en cada tarea que fuese necesario.

Al momento de diagramar el proceso de Gestión de Cambios en BPMN, se creó un mecanismo que permite asegurar que un mensaje va a una instancia apropiada basada en el contenido del mensaje. Este mecanismo, denominado correlaciones, va a ser el encargado de transmitir los mensajes generados entre tareas.

Las invocaciones de los servicios web se realizan desde las tareas de los procesos no ejecutables (pools de actores) hacia tareas del pool del proceso de Gestión de Cambio (pool ejecutable).

Cada correlación se encarga de transmitir los mensajes de entrada y salida para realizar la operación invocada desde una tarea en específico.

La primera invocación que se realiza es desde la tarea Request Change (Petición de Cambio) del Iniciador hacia y desde la tarea Receive Change Request (Recibir Petición de Cambio) del proceso Gestión de Cambio.

En la línea de correlaciones entre estas tareas arrastramos desde el explorador de procesos la operación deseada. Para esto desplegamos el módulo wsdl "Process" y seleccionamos la operación initProcess. Acá llevamos el mensaje de entrada "initProcessRequest" hasta la línea de la correlación que va desde la tarea Request Change a Receive Change Request. Para la otra línea de la correlación que va en sentido contrario se selecciona el mensaje de salida "initProcessResponse"

Una vez que hemos establecidos los mensajes de inicio, tenemos que invocar estos web services. Para esto vamos a la tarea Request Change hacemos click con el botón derecho y en el menú contextual que aparece seleccionamos la opción "Switch to operation". Acá nos demuestra opciones para invocar y proveer operaciones de servicios web, tal como lo podemos ver en la figura E-50.

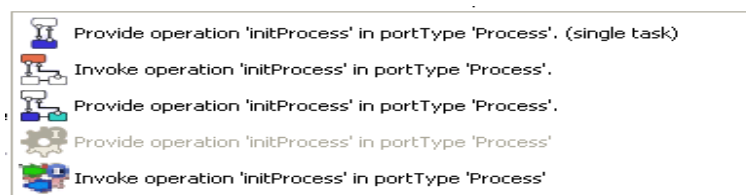


Figura E-50. Menú para invocar/proveer operaciones del 'initProcess'

Seleccionamos la opción “Invoke operation ‘initProcess’ in portType ‘Process’ para invocar las operaciones de inicio de proceso definidas en la estructura WSDL del formulario CustomerChangeRequest.

La correlación entre las tareas queda como lo demuestra la figura E-51

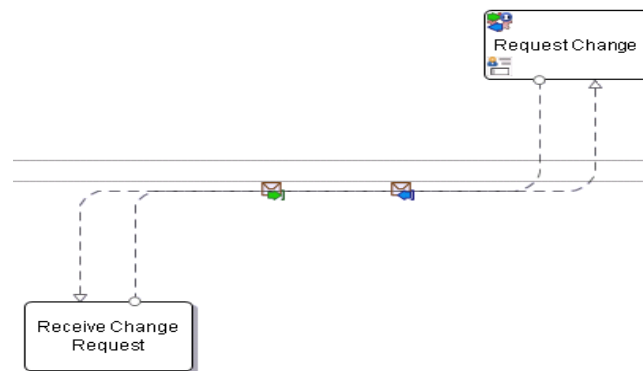


Figura E-51. Invocación de las operaciones request- response del Web Service “initProcess”

Esta es la única vez que se invocará a este servicio durante el diagrama del proceso, ya que solo es necesario iniciar el proceso una vez durante la ejecución. El resto de los servicios web a utilizar son para crear tareas, notificaciones de que estas se han creado y una notificación de proceso completado. A continuación la invocación del resto de los Web Services.

La siguiente invocación es para crear la tarea de asignación de desarrollador. Para esto se utiliza los siguientes web services del formulario AssignDeveloper.xform: createTaskRequest, createTaskResponse, notifyTaskCompletionRequest, notifyTaskCompletionResponse. Los dos primeros servicios se utilizan en las dos primeras líneas de la correlación entre la tarea Notify Manager (Notificar al Encargado) del Proceso de Gestión de Cambios y la tarea del Review Request (Revisar petición) en el pool del Encargado.

Para invocar estas operaciones le damos click con el botón derecho en la tarea Review Request, vamos a la opción “Switch to invoke operation” e invocamos a la operación “createTask” que esta dentro del WorkflowSoapService, tal como lo demuestra la figura E-52



Figura E-52. Menú para invocar operaciones del WorkflowSoapService

Para las otras líneas de la correlación entre esas tareas se utilizan los dos Servicios Web faltantes: notifyTask Completion Request y Notify Task Completion Response, estas operaciones son para asegurar que los mensajes de notificaciones de tareas han llegado correctamente al proceso Gestión de Cambios. Los invocamos desde la tarea Assign Developer del Encargado como lo muestra la figura E-53.

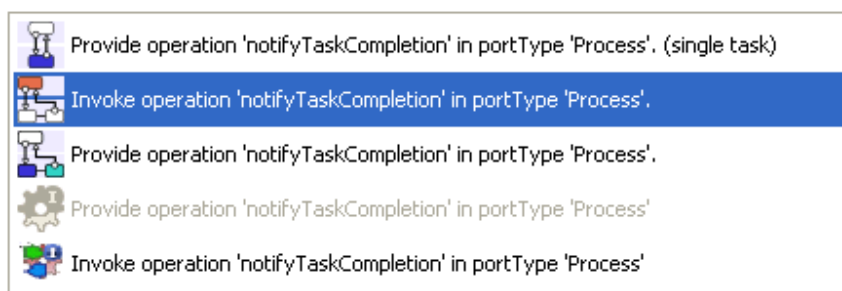


Figura E-53. Menú para invocar operaciones response-request de ‘notifyTaskCompletion’

Las siguientes 5 correlaciones creadas utilizan los mismos web services usados en la segunda correlación, esto debido a que en cada correlación se crea una tarea, que son las que se ejecutan al momento de la ejecución. Los web services son createTaskRequest, createTaskResponse, notifyTaskCompletionRequest, notifyTask CompletionResponse, pero cada web service invocado corresponde a la estructura wsdl del formulario correspondiente.

En la última correlación se invoca a la tarea para notificar al servidor de Intalio que la ejecución del modelado ha terminado completamente.

Los Web Services del formulario workcompleted.xform utilizados son el notifyRequest y el notifyResponse. Estos servicios se utilizan en las líneas de la correlación entre la tarea Notify Manager (Notificar al Encargado) del Proceso de Gestión de Cambios y la tarea Work Completed (Trabajo completado) del pool del Encargado.

Para invocar estas operaciones le damos click con el botón derecho en la tarea Work Completed, vamos a la opción “Switch to invoke operation” e invocamos a la operación “notify” dentro del WorkflowSoapService del formulario. La figura E-54

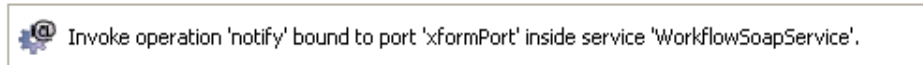


Figura E-54 Menú para invocar operaciones de 'notify' del WorkflowSoapService

Cabe destacar que al momento de invocar los Web Services internos de Intalio, no fueron utilizados todos los que fueron definidos.

ANEXO F - Comparación de las tecnologías de BPM analizadas

En base a las características, componentes y beneficios generados del análisis realizado previamente de las herramientas BPM, se puede fijar una comparación reflejada en la siguiente tabla comparativa, para determinar la solución tecnológica mas apropiada para modelar el proceso de Gestión de Cambio en las organizaciones:

Característica	INTALIO	UENGINE	SINGULARITY
Tipo de Licencia	Libre	Libre	Privada
Diseñador BPMN	SI	SI	SI
Mantenimiento	Actualizaciones automáticas y manuales	Manual	Manual
Soporte	Comunidad Online	Comunidad Online	---
Servidor BPEL	Si	SI	
Gestor de tareas Bpel4people	Si		
SOA	Si	Si	Si
Clustering	Si		
Transacciones distribuidas	Si	Si	
BAM	Si		
BRE	Si	Si	
ECM	Si		
ESB	Si		
Portal de Gestión	Si	Si	
Red Hat Linux	Si		
Suse Linux	Si		
Windows 2003 Server	Si	Si	Si
HP-UX	Si		
IBM AIX	Si		
IBM Power	Si		

Debian Linux	Si	Si	
Sun Solaris	Si		
Liferay	Si	Si	
Apache Axis	Si	Si	
Apache Geronimo	Si		
Apache Tomcat	Si		
IBM WebSphere CE	Si		
Servidor de aplicaciones JBoss	Si	Si	
SAP Netweaver	Si		
Derby Database	Si		
MySQL Enterprise	Si	Si	Si
EnterpriseDB/PostgreSQL	Si		
IBM DB2	Si		
Microsoft SQL Server	Si	Si	Si
MySQL Cluster	Si		
Oracle Database	Si		Si
Sybase ASE	Si		
Abundante documentación	Si		Si
XPDL	Si		
Soporte a Servicios Web	Si	Si	Si
Soporte de herramientas B2B	Si		Si
Dashboard	Si		
Integración con OLAP	Si		
Ambiente Colaborativo on- line	Si	Si	Si

Comparación entre elementos y características entre Intalio, uEngine y Singularity

Tomando en cuenta las características y componentes de cada herramienta comparada fue necesario establecer los siguientes criterios para seleccionar el BPM que permitió modelar el Proceso de Gestión de Cambios.

- Es conveniente utilizar una herramienta de software libre, para evitar un alto gasto de capital.
- Utilizar una herramienta que maneje los estándares de BPM como BPMN, WSDL, BPEL, etc.
- Simplicidad al momento de uso.
- Soporte de diversas herramientas open source.
- Manejo de múltiples bases de datos.
- Trabajar con servicios web e integrarlos a diferentes aplicaciones.
- Manejo e integración con aplicaciones multidimensionales.
- Portabilidad en diversos sistemas operativos.
- Motor de ejecución de reglas de negocio.
- Potente diseñador para de procesos de negocio.
- Capacidad de monitorear actividades de negocio en tiempo real.
- Facilidad en la integración con aplicaciones empresariales.

Ante tales aspectos, la herramienta seleccionada en Trabajo Especial de Grado fue Intalio BPMS, ya que esta provee una cantidad razonable de herramientas y en su entorno que permiten el uso de estándares, integración de aplicaciones, manejo de flujo de trabajos, despliegue “cero código”, optimización de procesos dinámicamente, motor de reglas de negocio, simulación nativa de procesos, generación servicios web, escalabilidad, manejo de transacciones multidimensionales, soporte de módulos adicionales, generación de tableros de rendimiento.

ANEXO H – Metadata de las operaciones de web services

```
<?xml version="1.0" encoding="utf-8" ?>
<wsdl:definitions targetNamespace="http://example.com/A ChangeManagement
/xform"
  xmlns:xform="http://example.com/ChangeManagement/xform"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">

  <wsdl:types>
    <!-- declare the namespace prefix again just to make sure
    this particular fragment is valid stand-alone schema. -->
    <xsd:schema
targetNamespace="http://example.com/AssignDeveloper/xform"
      xmlns:xform="http://example.com/ ChangeManagement /xform"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified">
      <xsd:include schemaLocation=" ChangeManagement.xform.xsd"/>
      <!-- the wsdl message elements -->
      <xsd:element name="initProcessRequest">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="taskId" type="xsd:string"/>
            <xsd:element name="participantToken" type="xsd:string"/>
            <xsd:element name="taskOutput">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element ref="xform:output"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="initProcessResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="status" type="xsd:string"/>
            <xsd:element name="errorCode" type="xsd:string" minOccurs="0"/>
            <xsd:element name="errorReason" type="xsd:string" minOccurs="0"/>
            <xsd:element name="taskMetaData" minOccurs="0">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="nextTaskId" type="xsd:string"/>
                  <xsd:element name="nextTaskURL" type="xsd:string"/>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </wsdl:types>
</wsdl:definitions>
```

```

<xsd:element name="createTaskRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="taskMetaData" type="xform:taskMetaDataType"/>
      <xsd:element name="participantToken" type="xsd:string"/>
      <xsd:element name="taskInput">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="xform:input"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="response">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="isChainedAfter" type="xsd:boolean"
minOccurs="0"/>
      <xsd:element name="taskMetaData" type="xform:taskMetaDataType"/>
      <xsd:element name="status" type="xsd:string"/>
      <xsd:element name="errorCode" type="xsd:string" minOccurs="0"/>
      <xsd:element name="errorReason" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

  <xsd:complexType name="accessControlType">
    <xsd:sequence>
      <xsd:element name="user" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:element name="role" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
<xsd:complexType name="scheduledActionsDetails">
  <xsd:choice minOccurs="0" maxOccurs="1">
    <xsd:element name="until" type="xsd:dateTime"
maxOccurs="1" minOccurs="0" />
    <xsd:element name="for" type="xsd:duration"
maxOccurs="1" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="taskMetaDataType">
  <xsd:sequence>
    <xsd:element name="taskId" type="xsd:string"/>
    <xsd:element name="taskState" type="xsd:token"/>
    <xsd:element name="taskType" type="xsd:token"/>
    <xsd:element name="description" type="xsd:string"/>
    <xsd:element name="processId" type="xsd:string"/>
    <xsd:element name="creationDate" type="xsd:dateTime"/>
    <xsd:element name="userOwner" type="xsd:string"
maxOccurs="unbounded" minOccurs="0"/>
  </xsd:sequence>

```

```

    <xsd:element name="roleOwner" type="xsd:string"
maxOccurs="unbounded" minOccurs="0"/>

    <xsd:element name="claimAction" type="xform:accessControlType"
maxOccurs="1" minOccurs="0"/>
    <xsd:element name="revokeAction" type="xform:accessControlType"
maxOccurs="1" minOccurs="0"/>
    <xsd:element name="saveAction" type="xform:accessControlType"
maxOccurs="1" minOccurs="0"/>
    <xsd:element name="completeAction" type="xform:accessControlType"
maxOccurs="1" minOccurs="0"/>

    <xsd:element name="formUrl" type="xsd:anyURI"/>
    <xsd:element name="failureCode" type="xsd:string"/>
    <xsd:element name="failureReason" type="xsd:string"/>
    <xsd:element name="priority" type="xsd:int"/>
    <xsd:element name="scheduledActions">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="expiration"
type="xform:scheduledActionsDetails"/>
                <xsd:element name="deferActivation"
type="xform:scheduledActionsDetails"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="userProcessCompleteSOAPAction"
type="xsd:string"/>
    <xsd:element name="isChainedBefore" type="xsd:boolean"/>
    <xsd:element name="previousTaskId" type="xsd:string"/>
    <xsd:element name="userProcessEndpoint" type="xsd:string"/>
    <xsd:element name="userProcessNamespaceURI" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>

<xsd:element name="notifyTaskCompletionRequest">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="taskMetaData" type="xform:taskMetaDataType"/>
            <xsd:element name="taskOutput">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="xform:output"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="status" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="notifyRequest">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="metadata">

```

```

        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="description" type="xsd:string"
minOccurs="0" maxOccurs="1" />
            <xsd:element name="creationDate" type="xsd:dateTime"
minOccurs="0" maxOccurs="1" />
            <xsd:element name="userOwner" type="xsd:string"
minOccurs="0" maxOccurs="unbounded" />
            <xsd:element name="roleOwner" type="xsd:string"
minOccurs="0" maxOccurs="unbounded" />
            <xsd:element name="formUrl" type="xsd:anyURI" minOccurs="1"
maxOccurs="1" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="input">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="xform:input" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="notifyResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="status" type="xsd:string" minOccurs="1"
maxOccurs="1" />
      <xsd:element name="errorCode" type="xsd:string" minOccurs="0"
maxOccurs="1" />
      <xsd:element name="errorReason" type="xsd:string" minOccurs="0"
maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="escalateRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="taskId" type="xsd:string"
minOccurs="1" maxOccurs="1" />
      <xsd:element name="userOwner" type="xsd:string"
minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="roleOwner" type="xsd:string"
minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="escalateResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="status" type="xsd:string"
minOccurs="1" maxOccurs="1" />

```

```

        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

    <!-- end of the wsdl message elements -->
</xsd:schema>
</wsdl:types>

<!-- Messages Create -->
<wsdl:message name="createTaskRequest">
    <wsdl:part name="root" element="xform:createTaskRequest"/>
</wsdl:message>
<wsdl:message name="createTaskResponse">
    <wsdl:part name="root" element="xform:response"/>
</wsdl:message>
<!-- Messages Complete -->
<wsdl:message name="notifyTaskCompletionRequest">
    <wsdl:part name="root" element="xform:notifyTaskCompletionRequest"/>
</wsdl:message>
<wsdl:message name="notifyTaskCompletionResponse">
    <wsdl:part name="root" element="xform:response"/>
</wsdl:message>
<!-- Messages Escalate-->
<wsdl:message name="escalateRequest">
    <wsdl:part name="root" element="xform:escalateRequest"/>
</wsdl:message>
<wsdl:message name="escalateResponse">
    <wsdl:part name="root" element="xform:escalateResponse"/>
</wsdl:message>
<!-- Messages Init -->
<wsdl:message name="initProcessRequest">
    <wsdl:part name="root" element="xform:initProcessRequest"/>
</wsdl:message>
<wsdl:message name="initProcessResponse">
    <wsdl:part name="root" element="xform:initProcessResponse"/>
</wsdl:message>
<!-- Messages -->
<wsdl:message name="notifyRequest">
    <wsdl:part name="root" element="xform:notifyRequest"/>
</wsdl:message>
<wsdl:message name="notifyResponse">
    <wsdl:part name="root" element="xform:notifyResponse"/>
</wsdl:message>

<!-- Operations provided by the processes interacting with workflow.
no port and binding for those as they are provided by the process. -->
<wsdl:portType name="Process">
    <wsdl:operation name="initProcess">
        <wsdl:input message="xform:initProcessRequest"
name="initProcessRequest"/>
        <wsdl:output message="xform:initProcessResponse"
name="initProcessResponse"/>
    </wsdl:operation>
    <wsdl:operation name="notifyTaskCompletion">

```

```

        <wsdl:input message="xform:notifyTaskCompletionRequest"
name="notifyTaskCompletionRequest" />
        <wsdl:output message="xform:notifyTaskCompletionResponse"
name="notifyTaskCompletionResponse" />
    </wsdl:operation>
</wsdl:portType>

<!-- operation provided by the workflow -->
<wsdl:portType name="Workflow">
    <wsdl:operation name="createTask">
        <wsdl:input message="xform:createTaskRequest"
name="createTaskRequest" />
        <wsdl:output message="xform:createTaskResponse"
name="createTaskResponse" />
    </wsdl:operation>
    <wsdl:operation name="escalate">
        <wsdl:input message="xform:escalateRequest"
name="escalateRequest" />
        <wsdl:output message="xform:escalateResponse"
name="escalateResponse" />
    </wsdl:operation>
    <wsdl:operation name="notify">
        <wsdl:input message="xform:notifyRequest" name="notifyRequest" />
        <wsdl:output message="xform:notifyResponse" name="notifyResponse" />
    </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="WorkflowSoapBinding" type="xform:Workflow">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="createTask">
        <soap:operation soapAction="createTask" style="document" />
        <wsdl:input><soap:body use="literal" /></wsdl:input>
        <wsdl:output><soap:body use="literal" /></wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="escalate">
        <soap:operation soapAction="escalate" style="document" />
        <wsdl:input><soap:body use="literal" /></wsdl:input>
        <wsdl:output><soap:body use="literal" /></wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="notify">
        <soap:operation soapAction="notify" style="document" />
        <wsdl:input><soap:body use="literal" /></wsdl:input>
        <wsdl:output><soap:body use="literal" /></wsdl:output>
    </wsdl:operation>
</wsdl:binding>

<wsdl:service name="WorkflowSoapService">
    <wsdl:port name="xformPort" binding="xform:WorkflowSoapBinding">
        <soap:address location="http://localhost:8080/fds/workflow/xform" />
    </wsdl:port>
</wsdl:service>

</wsdl:definitions>

```

ANEXO I – Código BPEL del proceso de Gestión de Cambios modelado en Intalio

```
<?xml version='1.0' encoding='UTF-8'?>
<bpel:process xmlns:bpel="http://docs.oasis-
open.org/wsbpel/2.0/process/executable"
xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:vprop="http://docs.oasis-open.org/wsbpel/2.0/varprop"
xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOW="http://example.com/SOW/xform"
xmlns:Encargado="http://example.com/ChangeManagement/Encargado"
xmlns:Iniciador="http://example.com/ChangeManagement/Iniciador"
xmlns:Work="http://example.com/Work/xform"
xmlns:WorkCompleted="http://example.com/WorkCompleted/xform"
xmlns:Desarrollador="http://example.com/ChangeManagement/Desarrollador"
xmlns:diag="http://example.com/ChangeManagement"
xmlns:AssignDeveloper="http://example.com/AssignDeveloper/xform"
xmlns:Report="http://example.com/Report/xform"
xmlns:Clarification="http://example.com/Clarification/xform"
xmlns:xform="http://example.com/ChangeManagement/ChangeManagement"
xmlns:this="http://example.com/ChangeManagement/ChangeManagement"
xmlns:Estimation="http://example.com/Estimation/xform"
xmlns:bpmn="http://www.intalio.com/bpms"
xmlns:atomic="http://ode.apache.org/atomicScope"
queryLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath2.0"
expressionLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath2.0"
bpmn:label="ChangeManagement" bpmn:id="_K02h4HK0EddyJdv25eCjJg"
name="ChangeManagement"
targetNamespace="http://example.com/ChangeManagement/ChangeManagement">
  <bpel:import namespace="http://example.com/AssignDeveloper/xform"
location="AssignDeveloper.xform.all.wsdl"
importType="http://schemas.xmlsoap.org/wSDL/"></bpel:import>
  <bpel:import namespace="http://example.com/Clarification/xform"
location="Clarification.xform.all.wsdl"
importType="http://schemas.xmlsoap.org/wSDL/"></bpel:import>
  <bpel:import namespace="http://example.com/ChangeManagement/ChangeManagement"
location="CustomerChangeRequest.xform.all.wsdl"
importType="http://schemas.xmlsoap.org/wSDL/"></bpel:import>
  <bpel:import namespace="http://example.com/Estimation/xform"
location="Estimation.xform.all.wsdl"
importType="http://schemas.xmlsoap.org/wSDL/"></bpel:import>
  <bpel:import namespace="http://example.com/Report/xform"
location="Report.xform.all.wsdl"
importType="http://schemas.xmlsoap.org/wSDL/"></bpel:import>
  <bpel:import namespace="http://example.com/SOW/xform"
location="SOW.xform.all.wsdl"
importType="http://schemas.xmlsoap.org/wSDL/"></bpel:import>
```

```

    <bpel:import namespace="http://example.com/Work/xform"
location="Work.xform.all.wsdl"
importType="http://schemas.xmlsoap.org/wsdl/"></bpel:import>
    <bpel:import namespace="http://example.com/WorkCompleted/xform"
location="WorkCompleted.xform.all.wsdl"
importType="http://schemas.xmlsoap.org/wsdl/"></bpel:import>
    <bpel:import namespace="http://example.com/ChangeManagement"
location="ChangeManagement.wsdl"
importType="http://schemas.xmlsoap.org/wsdl/"></bpel:import>
    <bpel:partnerLinks>
        <bpel:partnerLink name="wFmagic_iOfdqHK7EdyyJdv25eCjJgPlkVar"
partnerLinkType="diag:WFmagic_iOfdqHK7EdyyJdv25eCjJg"
initializePartnerRole="false"
myRole="ChangeManagement_for_EncargadoForThePortTypeProcess"
partnerRole="Encargado_for_ChangeManagementForXformPort"></bpel:partnerLi
nk>
        <bpel:partnerLink name="wFmagic_Sp58IHLAEdyL6qoRBK0TuAPlkVar"
partnerLinkType="diag:WFmagic_Sp58IHLAEdyL6qoRBK0TuA"
initializePartnerRole="false"
myRole="ChangeManagement_for_DesarrolladorForThePortTypeProcess"
partnerRole="Desarrollador_for_ChangeManagementForXformPort"></bpel:partn
erLink>
        <bpel:partnerLink name="wFmagic_a3iaCHLCEdyL6qoRBK0TuAPlkVar"
partnerLinkType="diag:WFmagic_a3iaCHLCEdyL6qoRBK0TuA"
initializePartnerRole="false"
myRole="ChangeManagement_for_EncargadoForThePortTypeProcess"
partnerRole="Encargado_for_ChangeManagementForXformPort"></bpel:partnerLi
nk>
        <bpel:partnerLink name="wFmagic_BAb8VnLGEdydN6900C3LYAPlkVar"
partnerLinkType="diag:WFmagic_BAb8VnLGEdydN6900C3LYA"
initializePartnerRole="false"
myRole="ChangeManagement_for_IniciadorForThePortTypeProcess"
partnerRole="Iniciador_for_ChangeManagementForXformPort"></bpel:partnerLi
nk>
        <bpel:partnerLink name="wFmagic_-qtlpnLGEdyFGs4L8GL2DAPlkVar"
partnerLinkType="diag:WFmagic_-qtlpnLGEdyFGs4L8GL2DA"
initializePartnerRole="false"
myRole="ChangeManagement_for_IniciadorForThePortTypeProcess"
partnerRole="Iniciador_for_ChangeManagementForXformPort"></bpel:partnerLi
nk>
        <bpel:partnerLink name="wFmagic_NQVgUHLJEdyqb04eiX9ZdAPlkVar"
partnerLinkType="diag:WFmagic_NQVgUHLJEdyqb04eiX9ZdA"
initializePartnerRole="false"
myRole="ChangeManagement_for_DesarrolladorForThePortTypeProcess"
partnerRole="Desarrollador_for_ChangeManagementForXformPort"></bpel:partn
erLink>
        <bpel:partnerLink
name="iniciadorAndChangeManagementForPortTypeProcessPlkVar"
partnerLinkType="diag:IniciadorAndChangeManagementForPortTypeProcessPlk"
myRole="ChangeManagement_for_Iniciador"></bpel:partnerLink>
        <bpel:partnerLink
name="encargadoAndChangeManagementForXformPortPlkVar"
partnerLinkType="diag:EncargadoAndChangeManagementForXformPortPlk"

```



```

initializePartnerRole="true"
partnerRole="Encargado_for_ChangeManagement"></bpel:partnerLink>
</bpel:partnerLinks>
<bpel:variables>
  <bpel:variable name="xformInitProcessRequestMsg"
messageType="xform:initProcessRequest"></bpel:variable>
  <bpel:variable name="xformInitProcessResponseMsg"
messageType="xform:initProcessResponse"></bpel:variable>
  <bpel:variable name="assignDeveloperCreateTaskRequestMsg"
messageType="AssignDeveloper:createTaskRequest"></bpel:variable>
  <bpel:variable name="assignDeveloperCreateTaskResponseMsg"
messageType="AssignDeveloper:createTaskResponse"></bpel:variable>
  <bpel:variable name="assignDeveloperNotifyTaskCompletionRequestMsg"
messageType="AssignDeveloper:notifyTaskCompletionRequest"></bpel:variable
>
  <bpel:variable name="assignDeveloperNotifyTaskCompletionResponseMsg"
messageType="AssignDeveloper:notifyTaskCompletionResponse"></bpel:variabl
e>
  <bpel:variable name="workCreateTaskRequestMsg"
messageType="Work:createTaskRequest"></bpel:variable>
  <bpel:variable name="workCreateTaskResponseMsg"
messageType="Work:createTaskResponse"></bpel:variable>
  <bpel:variable name="workNotifyTaskCompletionRequestMsg"
messageType="Work:notifyTaskCompletionRequest"></bpel:variable>
  <bpel:variable name="workNotifyTaskCompletionResponseMsg"
messageType="Work:notifyTaskCompletionResponse"></bpel:variable>
  <bpel:variable name="workCompletedNotifyRequestMsg"
messageType="WorkCompleted:notifyRequest"></bpel:variable>
  <bpel:variable name="workCompletedNotifyResponseMsg"
messageType="WorkCompleted:notifyResponse"></bpel:variable>
  <bpel:variable name="reEvaluate" type="xs:boolean"></bpel:variable>
</bpel:variables>
<bpel:sequence>
  <bpel:receive
partnerLink="iniciadorAndChangeManagementForPortTypeProcessPlkVar"
portType="xform:Process" operation="initProcess"
variable="xformInitProcessRequestMsg" createInstance="yes"
bpmn:label="Recibir Petición de Cambio"
bpmn:id="_lsvxYHK0EdyyJdv25eCjJg"></bpel:receive>
  <bpel:assign name="init-variables-ChangeManagement">
  <bpel:copy bpmn:label="$xformInitProcessResponseMsg">
  <bpel:from>
  <bpel:literal>
<xform:initProcessResponse>
  <xform:status>OK</xform:status>
  <xform:errorCode></xform:errorCode>
  <xform:errorReason></xform:errorReason>
  <xform:taskMetaData>
  <xform:nextTaskId></xform:nextTaskId>
  <xform:nextTaskURL></xform:nextTaskURL>
  </xform:taskMetaData>
</xform:initProcessResponse></bpel:literal>
  </bpel:from>
  <bpel:to>$xformInitProcessResponseMsg.root</bpel:to>

```

```

        </bpel:copy>
        <bpel:copy bpmn:label="$assignDeveloperCreateTaskRequestMsg">
            <bpel:from>
                <bpel:literal>
<AssignDeveloper:createTaskRequest>
    <AssignDeveloper:taskMetaData>

<AssignDeveloper:taskId>_iOfdqHK7EdyyJdv25eCjJg</AssignDeveloper:taskId>
    <AssignDeveloper:taskState></AssignDeveloper:taskState>
    <AssignDeveloper:taskType></AssignDeveloper:taskType>
    <AssignDeveloper:description>Assign Developer to Change
Request</AssignDeveloper:description>
    <AssignDeveloper:processId></AssignDeveloper:processId>
    <AssignDeveloper:creationDate></AssignDeveloper:creationDate>
    <AssignDeveloper:userOwner />
    <AssignDeveloper:roleOwner>appdev\manager</AssignDeveloper:roleOwner>
    <AssignDeveloper:claimAction>
        <AssignDeveloper:user></AssignDeveloper:user>
        <AssignDeveloper:role></AssignDeveloper:role>
    </AssignDeveloper:claimAction>
    <AssignDeveloper:revokeAction>
        <AssignDeveloper:user></AssignDeveloper:user>
        <AssignDeveloper:role></AssignDeveloper:role>
    </AssignDeveloper:revokeAction>
    <AssignDeveloper:saveAction>
        <AssignDeveloper:user></AssignDeveloper:user>
        <AssignDeveloper:role></AssignDeveloper:role>
    </AssignDeveloper:saveAction>
    <AssignDeveloper:completeAction>
        <AssignDeveloper:user></AssignDeveloper:user>
        <AssignDeveloper:role></AssignDeveloper:role>
    </AssignDeveloper:completeAction>

<AssignDeveloper:formUrl>oxf://ApplicationDevelopment/AssignDeveloper.xfo
rm</AssignDeveloper:formUrl>
    <AssignDeveloper:failureCode></AssignDeveloper:failureCode>
    <AssignDeveloper:failureReason></AssignDeveloper:failureReason>
    <AssignDeveloper:priority></AssignDeveloper:priority>
    <AssignDeveloper:scheduledActions>
        <AssignDeveloper:expiration>
            <AssignDeveloper:until></AssignDeveloper:until>
            <AssignDeveloper:for></AssignDeveloper:for>
        </AssignDeveloper:expiration>
        <AssignDeveloper:deferActivation>
            <AssignDeveloper:until></AssignDeveloper:until>
            <AssignDeveloper:for></AssignDeveloper:for>
        </AssignDeveloper:deferActivation>
    </AssignDeveloper:scheduledActions>

<AssignDeveloper:userProcessCompleteSOAPAction>http://example.com/AssignD
eveloper/xform/Process/notifyTaskCompletion</AssignDeveloper:userProcessC
ompleteSOAPAction>
    <AssignDeveloper:isChainedBefore></AssignDeveloper:isChainedBefore>
    <AssignDeveloper:previousTaskId></AssignDeveloper:previousTaskId>

```

```

<AssignDeveloper:userProcessEndpoint>http://localhost:8080/ode/processes/
ApplicationDevelopment/ChangeManagement/ChangeManagement/Encargado/WFmagi
c_iOfdqHK7EdyyJdv25eCjJg</AssignDeveloper:userProcessEndpoint>

<AssignDeveloper:userProcessNamespaceURI>http://example.com/AssignDevelop
er/xforn</AssignDeveloper:userProcessNamespaceURI>
</AssignDeveloper:taskMetaData>
<AssignDeveloper:participantToken></AssignDeveloper:participantToken>
<AssignDeveloper:taskInput>
  <AssignDeveloper:input>
    <AssignDeveloper:ChangeRequest></AssignDeveloper:ChangeRequest>

<AssignDeveloper:ChangeRequestLabel></AssignDeveloper:ChangeRequestLabel>
  </AssignDeveloper:input>
  </AssignDeveloper:taskInput>
</AssignDeveloper:createTaskRequest></bpel:literal>
  </bpel:from>
  <bpel:to>$assignDeveloperCreateTaskRequestMsg.root</bpel:to>
  </bpel:copy>
  <bpel:copy

bpmn:label="$assignDeveloperNotifyTaskCompletionResponseMsg">
  <bpel:from>
    <bpel:literal>
<AssignDeveloper:response>
  <AssignDeveloper:isChainedAfter></AssignDeveloper:isChainedAfter>
  <AssignDeveloper:taskMetaData>
    <AssignDeveloper:taskId></AssignDeveloper:taskId>
    <AssignDeveloper:taskState></AssignDeveloper:taskState>
    <AssignDeveloper:taskType></AssignDeveloper:taskType>
    <AssignDeveloper:description></AssignDeveloper:description>
    <AssignDeveloper:processId></AssignDeveloper:processId>
    <AssignDeveloper:creationDate></AssignDeveloper:creationDate>
    <AssignDeveloper:userOwner></AssignDeveloper:userOwner>
    <AssignDeveloper:roleOwner></AssignDeveloper:roleOwner>
    <AssignDeveloper:claimAction>
      <AssignDeveloper:user></AssignDeveloper:user>
      <AssignDeveloper:role></AssignDeveloper:role>
    </AssignDeveloper:claimAction>
    <AssignDeveloper:revokeAction>
      <AssignDeveloper:user></AssignDeveloper:user>
      <AssignDeveloper:role></AssignDeveloper:role>
    </AssignDeveloper:revokeAction>
    <AssignDeveloper:saveAction>
      <AssignDeveloper:user></AssignDeveloper:user>
      <AssignDeveloper:role></AssignDeveloper:role>
    </AssignDeveloper:saveAction>
    <AssignDeveloper:completeAction>
      <AssignDeveloper:user></AssignDeveloper:user>
      <AssignDeveloper:role></AssignDeveloper:role>
    </AssignDeveloper:completeAction>
    <AssignDeveloper:formUrl></AssignDeveloper:formUrl>
    <AssignDeveloper:failureCode></AssignDeveloper:failureCode>
    <AssignDeveloper:failureReason></AssignDeveloper:failureReason>

```

```

<AssignDeveloper:priority></AssignDeveloper:priority>
<AssignDeveloper:scheduledActions>
  <AssignDeveloper:expiration>
    <AssignDeveloper:until></AssignDeveloper:until>
    <AssignDeveloper:for></AssignDeveloper:for>
  </AssignDeveloper:expiration>
  <AssignDeveloper:deferActivation>
    <AssignDeveloper:until></AssignDeveloper:until>
    <AssignDeveloper:for></AssignDeveloper:for>
  </AssignDeveloper:deferActivation>
</AssignDeveloper:scheduledActions>

<AssignDeveloper:userProcessCompleteSOAPAction></AssignDeveloper:userProcessCompleteSOAPAction>
  <AssignDeveloper:isChainedBefore></AssignDeveloper:isChainedBefore>
  <AssignDeveloper:previousTaskId></AssignDeveloper:previousTaskId>

<AssignDeveloper:userProcessEndpoint></AssignDeveloper:userProcessEndpoint>

<AssignDeveloper:userProcessNamespaceURI></AssignDeveloper:userProcessNamespaceURI>
  </AssignDeveloper:taskMetaData>
  <AssignDeveloper:status>OK</AssignDeveloper:status>
  <AssignDeveloper:errorCode></AssignDeveloper:errorCode>
  <AssignDeveloper:errorReason></AssignDeveloper:errorReason>
</AssignDeveloper:response></bpel:literal>
  </bpel:from>

<bpel:to>$assignDeveloperNotifyTaskCompletionResponseMsg.root</bpel:to>
  </bpel:copy>
  <bpel:copy bpmn:label="$workCreateTaskRequestMsg">
    <bpel:from>
      <bpel:literal>
<Work:createTaskRequest>
  <Work:taskMetaData>
    <Work:taskId>_NQVgUHLJEdyqb04eiX9ZdA</Work:taskId>
    <Work:taskState></Work:taskState>
    <Work:taskType></Work:taskType>
    <Work:description>Change Request confirmed</Work:description>
    <Work:processId></Work:processId>
    <Work:creationDate></Work:creationDate>
    <Work:userOwner />
    <Work:roleOwner />
    <Work:claimAction>
      <Work:user></Work:user>
      <Work:role></Work:role>
    </Work:claimAction>
    <Work:revokeAction>
      <Work:user></Work:user>
      <Work:role></Work:role>
    </Work:revokeAction>
    <Work:saveAction>
      <Work:user></Work:user>

```

```

    <Work:role></Work:role>
  </Work:saveAction>
  <Work:completeAction>
    <Work:user></Work:user>
    <Work:role></Work:role>
  </Work:completeAction>
  <Work:formUrl>oxf://ApplicationDevelopment/Work.xform</Work:formUrl>
  <Work:failureCode></Work:failureCode>
  <Work:failureReason></Work:failureReason>
  <Work:priority></Work:priority>
  <Work:scheduledActions>
    <Work:expiration>
      <Work:until></Work:until>
      <Work:for></Work:for>
    </Work:expiration>
    <Work:deferActivation>
      <Work:until></Work:until>
      <Work:for></Work:for>
    </Work:deferActivation>
  </Work:scheduledActions>

  <Work:userProcessCompleteSOAPAction>http://example.com/Work/xform/Process
  /notifyTaskCompletion</Work:userProcessCompleteSOAPAction>
    <Work:isChainedBefore></Work:isChainedBefore>
    <Work:previousTaskId></Work:previousTaskId>

  <Work:userProcessEndpoint>http://localhost:8080/ode/processes/Application
  Development/ChangeManagement/ChangeManagement/Desarrollador/WFmagic_NQVgU
  HLJEdyqb04eiX9ZdA</Work:userProcessEndpoint>

  <Work:userProcessNamespaceURI>http://example.com/Work/xform</Work:userPro
  cessNamespaceURI>
    </Work:taskMetaData>
    <Work:participantToken></Work:participantToken>
    <Work:taskInput>
      <Work:input>
        <Work:Text2></Work:Text2>
        <Work:ChangeRequest></Work:ChangeRequest>
        <Work:Text></Work:Text>
      </Work:input>
    </Work:taskInput>
  </Work:createTaskRequest></bpel:literal>
    </bpel:from>
    <bpel:to>$workCreateTaskRequestMsg.root</bpel:to>
  </bpel:copy>
  <bpel:copy bpmn:label="$workNotifyTaskCompletionResponseMsg">
    <bpel:from>
      <bpel:literal>
<Work:response>
  <Work:isChainedAfter></Work:isChainedAfter>
  <Work:taskMetaData>
    <Work:taskId></Work:taskId>
    <Work:taskState></Work:taskState>
    <Work:taskType></Work:taskType>

```

```

<Work:description></Work:description>
<Work:processId></Work:processId>
<Work:creationDate></Work:creationDate>
<Work:userOwner></Work:userOwner>
<Work:roleOwner></Work:roleOwner>
<Work:claimAction>
  <Work:user></Work:user>
  <Work:role></Work:role>
</Work:claimAction>
<Work:revokeAction>
  <Work:user></Work:user>
  <Work:role></Work:role>
</Work:revokeAction>
<Work:saveAction>
  <Work:user></Work:user>
  <Work:role></Work:role>
</Work:saveAction>
<Work:completeAction>
  <Work:user></Work:user>
  <Work:role></Work:role>
</Work:completeAction>
<Work:formUrl></Work:formUrl>
<Work:failureCode></Work:failureCode>
<Work:failureReason></Work:failureReason>
<Work:priority></Work:priority>
<Work:scheduledActions>
  <Work:expiration>
    <Work:until></Work:until>
    <Work:for></Work:for>
  </Work:expiration>
  <Work:deferActivation>
    <Work:until></Work:until>
    <Work:for></Work:for>
  </Work:deferActivation>
</Work:scheduledActions>

<Work:userProcessCompleteSOAPAction></Work:userProcessCompleteSOAPAction>
  <Work:isChainedBefore></Work:isChainedBefore>
  <Work:previousTaskId></Work:previousTaskId>
  <Work:userProcessEndpoint></Work:userProcessEndpoint>
  <Work:userProcessNamespaceURI></Work:userProcessNamespaceURI>
</Work:taskMetadata>
<Work:status>OK</Work:status>
<Work:errorCode></Work:errorCode>
<Work:errorReason></Work:errorReason>
</Work:response></bpel:literal>
  </bpel:from>
  <bpel:to>$workNotifyTaskCompletionResponseMsg.root</bpel:to>
</bpel:copy>
<bpel:copy bpmn:label="$workCompletedNotifyRequestMsg">
  <bpel:from>
    <bpel:literal>
<WorkCompleted:notifyRequest>
  <WorkCompleted:metadata>

```

```

    <WorkCompleted:description>Change Request
Completed</WorkCompleted:description>
    <WorkCompleted:creationDate></WorkCompleted:creationDate>
    <WorkCompleted:userOwner />
    <WorkCompleted:roleOwner>appdev\manager</WorkCompleted:roleOwner>

<WorkCompleted:formUrl>oxf://ApplicationDevelopment/WorkCompleted.xform</
WorkCompleted:formUrl>
</WorkCompleted:metadata>
<WorkCompleted:input>
  <WorkCompleted:input>
    <WorkCompleted:ChangeRequest></WorkCompleted:ChangeRequest>
    <WorkCompleted:Developer></WorkCompleted:Developer>
    <WorkCompleted:Text></WorkCompleted:Text>
  </WorkCompleted:input>
</WorkCompleted:input>
</WorkCompleted:notifyRequest></bpel:literal>
  </bpel:from>
  <bpel:to>$workCompletedNotifyRequestMsg.root</bpel:to>
</bpel:copy>
<bpel:copy bpmn:label="$reEvaluate">
  <bpel:from>
    <bpel:literal></bpel:literal>
  </bpel:from>
  <bpel:to>$reEvaluate</bpel:to>
</bpel:copy>
</bpel:assign>
<bpel:reply
partnerLink="iniciadorAndChangeManagementForPortTypeProcessPlkVar"
portType="xform:Process" operation="initProcess"
variable="xformInitProcessResponseMsg" bpmn:label="Recibir Petición de
Cambio" bpmn:id="_lsvxYHK0EdyyJdv25eCjJg"></bpel:reply>
  <bpel:assign bpmn:label="Notificar Encargado"
bpmn:id="_K02h4nK0EdyyJdv25eCjJg">
    <bpel:copy>

<bpel:from>$xformInitProcessRequestMsg.root/xform:taskOutput/xform:output
/xform:ChangeRequest</bpel:from>

<bpel:to>$assignDesarrolladorCreateTaskRequestMsg.root/AssignDeveloper:ta
skInput/AssignDeveloper:input/AssignDeveloper:ChangeRequest</bpel:to>
  </bpel:copy>
</bpel:assign>
<bpel:assign>
  <bpel:copy>

<bpel:from>"http://localhost:8080/ode/processes/ApplicationDevelopment/Ch
angeManagement/ChangeManagement/Encargado/WFmagic_iOfdqHK7EdyyJdv25eCjJg"
</bpel:from>

<bpel:to>$assignDeveloperCreateTaskRequestMsg.root/AssignDeveloper:taskMe
taData/AssignDeveloper:userProcessEndpoint</bpel:to>
  </bpel:copy>
  <bpel:copy>

```

```

<bpel:from>"http://example.com/AssignDeveloper/xform/Process/notifyTaskCo
mpletion"</bpel:from>

<bpel:to>$assignDeveloperCreateTaskRequestMsg.root/AssignDeveloper:taskMe
taData/AssignDeveloper:userProcessCompleteSOAPAction</bpel:to>
  </bpel:copy>
  <bpel:copy>
    <bpel:from>"_iOfdr3K7EdyyJdv25eCjJg"</bpel:from>

<bpel:to>$assignDeveloperCreateTaskRequestMsg.root/AssignDeveloper:taskMe
taData/AssignDeveloper:taskId</bpel:to>
  </bpel:copy>
  <bpel:copy>
    <bpel:from>"Assign Developer to Change Request"</bpel:from>

<bpel:to>$assignDeveloperCreateTaskRequestMsg.root/AssignDeveloper:taskMe
taData/AssignDeveloper:description</bpel:to>
  </bpel:copy>
  </bpel:assign>
  <bpel:invoke partnerLink="wFmagic_iOfdqHK7EdyyJdv25eCjJgPlkVar"
portType="AssignDeveloper:Workflow" operation="createTask"
inputVariable="assignDeveloperCreateTaskRequestMsg"
outputVariable="assignDeveloperCreateTaskResponseMsg"
bpmn:label="Notificar Encargado"
bpmn:id="_K02h4nK0EdyyJdv25eCjJg"></bpel:invoke>
  <bpel:receive partnerLink="wFmagic_iOfdqHK7EdyyJdv25eCjJgPlkVar"
portType="AssignDeveloper:Process" operation="notifyTaskCompletion"
variable="assignDeveloperNotifyTaskCompletionRequestMsg"
bpmn:label="Recibir Desarrollador Asignado"
bpmn:id="_09q9wHK0EdyyJdv25eCjJg"></bpel:receive>
  <bpel:assign bpmn:label="Recibir Desarrollador Asignado"
bpmn:id="_09q9wHK0EdyyJdv25eCjJg">
    <bpel:copy>
      <bpel:from>true()</bpel:from>
      <bpel:to>$reEvaluate</bpel:to>
    </bpel:copy>
  </bpel:assign>
  <bpel:reply partnerLink="wFmagic_iOfdqHK7EdyyJdv25eCjJgPlkVar"
portType="AssignDeveloper:Process" operation="notifyTaskCompletion"
variable="assignDeveloperNotifyTaskCompletionResponseMsg"
bpmn:label="Recibir Desarrollador Asignado"
bpmn:id="_09q9wHK0EdyyJdv25eCjJg"></bpel:reply>
  <bpel:scope name="SubProcess_tl_eMHK3EdyyJdv25eCjJg"
bpmn:label="SubProcess" bpmn:id="_tl_eMHK3EdyyJdv25eCjJg">
    <bpel:variables>
      <bpel:variable name="estimationCreateTaskRequestMsg"
messageType="Estimation:createTaskRequest"></bpel:variable>
      <bpel:variable name="estimationCreateTaskResponseMsg"
messageType="Estimation:createTaskResponse"></bpel:variable>
      <bpel:variable name="estimationNotifyTaskCompletionRequestMsg"
messageType="Estimation:notifyTaskCompletionRequest"></bpel:variable>
      <bpel:variable name="estimationNotifyTaskCompletionResponseMsg"
messageType="Estimation:notifyTaskCompletionResponse"></bpel:variable>

```



```

        <bpel:variable name="reportCreateTaskRequestMsg"
messageType="Report:createTaskRequest"></bpel:variable>
        <bpel:variable name="reportCreateTaskResponseMsg"
messageType="Report:createTaskResponse"></bpel:variable>
        <bpel:variable name="reportNotifyTaskCompletionRequestMsg"
messageType="Report:notifyTaskCompletionRequest"></bpel:variable>
        <bpel:variable name="reportNotifyTaskCompletionResponseMsg"
messageType="Report:notifyTaskCompletionResponse"></bpel:variable>
        <bpel:variable name="sOWCreateTaskRequestMsg"
messageType="SOW:createTaskRequest"></bpel:variable>
        <bpel:variable name="sOWCreateTaskResponseMsg"
messageType="SOW:createTaskResponse"></bpel:variable>
        <bpel:variable name="sOWNotifyTaskCompletionRequestMsg"
messageType="SOW:notifyTaskCompletionRequest"></bpel:variable>
        <bpel:variable name="sOWNotifyTaskCompletionResponseMsg"
messageType="SOW:notifyTaskCompletionResponse"></bpel:variable>
        <bpel:variable name="clarificationCreateTaskRequestMsg"
messageType="Clarification:createTaskRequest"></bpel:variable>
        <bpel:variable name="clarificationCreateTaskResponseMsg"
messageType="Clarification:createTaskResponse"></bpel:variable>
        <bpel:variable name="clarificationNotifyTaskCompletionRequestMsg"
messageType="Clarification:notifyTaskCompletionRequest"></bpel:variable>
        <bpel:variable
name="clarificationNotifyTaskCompletionResponseMsg"
messageType="Clarification:notifyTaskCompletionResponse"></bpel:variable>
    </bpel:variables>
    <bpel:while>
        <bpel:condition>$reEvaluate = true()</bpel:condition>
        <bpel:sequence>
            <bpel:assign name="init-variables-SubProcess">
                <bpel:copy bpmn:label="$sestimationCreateTaskRequestMsg">
                    <bpel:from>
                        <bpel:literal><Estimation:createTaskRequest>
<Estimation:taskMetaData>
                <Estimation:taskId>_Sp58IHLAEdyL6qoRBK0TuA</Estimation:taskId>
                <Estimation:taskState>
</Estimation:taskState>
                <Estimation:taskType>
</Estimation:taskType>
                <Estimation:description>Provide Estimate for Change
Request</Estimation:description>
                <Estimation:processId>
</Estimation:processId>
                <Estimation:creationDate>
</Estimation:creationDate>
                <Estimation:userOwner />
                <Estimation:roleOwner />
                <Estimation:claimAction>
                    <Estimation:user>
</Estimation:user>
                    <Estimation:role>
</Estimation:role>
                </Estimation:claimAction>
                <Estimation:revokeAction>

```

```

        <Estimation:user>
        </Estimation:user>
        <Estimation:role>
        </Estimation:role>
    </Estimation:revokeAction>
    <Estimation:saveAction>
        <Estimation:user>
        </Estimation:user>
        <Estimation:role>
        </Estimation:role>
    </Estimation:saveAction>
    <Estimation:completeAction>
        <Estimation:user>
        </Estimation:user>
        <Estimation:role>
        </Estimation:role>
    </Estimation:completeAction>

<Estimation:formUrl>oxf://ApplicationDevelopment/Estimation.xform</Estima
tion:formUrl>
    <Estimation:failureCode>
    </Estimation:failureCode>
    <Estimation:failureReason>
    </Estimation:failureReason>

<Estimation:userProcessCompleteSOAPAction>http://example.com/Estimation/x
form/Process/notifyTaskCompletion</Estimation:userProcessCompleteSOAPActi
on>
    <Estimation:isChainedBefore>
    </Estimation:isChainedBefore>
    <Estimation:previousTaskId>
    </Estimation:previousTaskId>

<Estimation:userProcessEndpoint>http://localhost:8080/ode/processes/Appli
cationDevelopment/ChangeManagement/ChangeManagement/Desarrollador/WFmagic
_Sp58IHLAEdyL6qoRBK0TuA</Estimation:userProcessEndpoint>

<Estimation:userProcessNamespaceURI>http://example.com/Estimation/xform</
Estimation:userProcessNamespaceURI>
    </Estimation:taskMetaData>
    <Estimation:participantToken>
    </Estimation:participantToken>
    <Estimation:taskInput>
        <Estimation:input>
            <Estimation:DeveloperToAsk>
            </Estimation:DeveloperToAsk>
            <Estimation:TextWhenUnknown>
            </Estimation:TextWhenUnknown>
            <Estimation:EstimatedEffortLabel>
            </Estimation:EstimatedEffortLabel>
            <Estimation:ChangeRequest>
            </Estimation:ChangeRequest>
            <Estimation:ChangeRequestLabel>
            </Estimation:ChangeRequestLabel>
        </Estimation:input>
    </Estimation:taskInput>

```

```

        </Estimation:input>
    </Estimation:taskInput>
</Estimation:createTaskRequest></bpel:literal>
    </bpel:from>
    <bpel:to>$estimationCreateTaskRequestMsg.root</bpel:to>
</bpel:copy>
<bpel:copy
bpmn:label="$estimationNotifyTaskCompletionResponseMsg">
    <bpel:from>
        <bpel:literal>
<Estimation:response>
    <Estimation:isChainedAfter></Estimation:isChainedAfter>
<Estimation:taskMetaData>
    <Estimation:taskId></Estimation:taskId>
    <Estimation:taskState></Estimation:taskState>
    <Estimation:taskType></Estimation:taskType>
    <Estimation:description></Estimation:description>
    <Estimation:processId></Estimation:processId>
    <Estimation:creationDate></Estimation:creationDate>
    <Estimation:userOwner></Estimation:userOwner>
    <Estimation:roleOwner></Estimation:roleOwner>
    <Estimation:claimAction>
        <Estimation:user></Estimation:user>
        <Estimation:role></Estimation:role>
    </Estimation:claimAction>
    <Estimation:revokeAction>
        <Estimation:user></Estimation:user>
        <Estimation:role></Estimation:role>
    </Estimation:revokeAction>
    <Estimation:saveAction>
        <Estimation:user></Estimation:user>
        <Estimation:role></Estimation:role>
    </Estimation:saveAction>
    <Estimation:completeAction>
        <Estimation:user></Estimation:user>
        <Estimation:role></Estimation:role>
    </Estimation:completeAction>
    <Estimation:formUrl></Estimation:formUrl>
    <Estimation:failureCode></Estimation:failureCode>
    <Estimation:failureReason></Estimation:failureReason>
    <Estimation:priority></Estimation:priority>
    <Estimation:scheduledActions>
        <Estimation:expiration>
            <Estimation:until></Estimation:until>
            <Estimation:for></Estimation:for>
        </Estimation:expiration>
        <Estimation:deferActivation>
            <Estimation:until></Estimation:until>
            <Estimation:for></Estimation:for>
        </Estimation:deferActivation>
    </Estimation:scheduledActions>

<Estimation:userProcessCompleteSOAPAction></Estimation:userProcessComple
eSOAPAction>

```

```

    <Estimation:isChainedBefore></Estimation:isChainedBefore>
    <Estimation:previousTaskId></Estimation:previousTaskId>
    <Estimation:userProcessEndpoint></Estimation:userProcessEndpoint>

<Estimation:userProcessNamespaceURI></Estimation:userProcessNamespaceURI>
  </Estimation:taskMetaData>
  <Estimation:status>OK</Estimation:status>
  <Estimation:errorCode></Estimation:errorCode>
  <Estimation:errorReason></Estimation:errorReason>
</Estimation:response></bpel:literal>
  </bpel:from>

<bpel:to>${estimationNotifyTaskCompletionResponseMsg.root}</bpel:to>
  </bpel:copy>
  <bpel:copy bpmn:label="$reportCreateTaskRequestMsg">
    <bpel:from>
      <bpel:literal>
<Report:createTaskRequest>
  <Report:taskMetaData>
    <Report:taskId>_a3iaCHLCEdyL6qoRBK0TuA</Report:taskId>
    <Report:taskState></Report:taskState>
    <Report:taskType></Report:taskType>
    <Report:description>Review Report from Developer</Report:description>
    <Report:processId></Report:processId>
    <Report:creationDate></Report:creationDate>
    <Report:userOwner />
    <Report:roleOwner>appdev\manager</Report:roleOwner>
    <Report:claimAction>
      <Report:user></Report:user>
      <Report:role></Report:role>
    </Report:claimAction>
    <Report:revokeAction>
      <Report:user></Report:user>
      <Report:role></Report:role>
    </Report:revokeAction>
    <Report:saveAction>
      <Report:user></Report:user>
      <Report:role></Report:role>
    </Report:saveAction>
    <Report:completeAction>
      <Report:user></Report:user>
      <Report:role></Report:role>
    </Report:completeAction>

<Report:formUrl>oxf://ApplicationDevelopment/Report.xform</Report:formUrl
>
  <Report:failureCode></Report:failureCode>
  <Report:failureReason></Report:failureReason>
  <Report:priority></Report:priority>
  <Report:scheduledActions>
    <Report:expiration>
      <Report:until></Report:until>
      <Report:for></Report:for>
    </Report:expiration>

```

```

    <Report:deferActivation>
      <Report:until></Report:until>
      <Report:for></Report:for>
    </Report:deferActivation>
  </Report:scheduledActions>

  <Report:userProcessCompleteSOAPAction>http://example.com/Report/xform/Process/notifyTaskCompletion</Report:userProcessCompleteSOAPAction>
    <Report:isChainedBefore></Report:isChainedBefore>
    <Report:previousTaskId></Report:previousTaskId>

  <Report:userProcessEndpoint>http://localhost:8080/ode/processes/ApplicationDevelopment/ChangeManagement/ChangeManagement/Encargado/WFmagic_a3iaCHLCEdyL6qoRBK0TuA</Report:userProcessEndpoint>

  <Report:userProcessNamespaceURI>http://example.com/Report/xform</Report:userProcessNamespaceURI>
  </Report:taskMetaData>
  <Report:participantToken></Report:participantToken>
  <Report:taskInput>
    <Report:input>
      <Report:DecisionLabel></Report:DecisionLabel>
      <Report:ChangeRequestLabel></Report:ChangeRequestLabel>
      <Report:ChangeRequest></Report:ChangeRequest>
      <Report:EstimatedEffortLabel></Report:EstimatedEffortLabel>
      <Report:Estimate></Report:Estimate>
      <Report:DeveloperToAsk></Report:DeveloperToAsk>
      <Report:ClarificationNeeded></Report:ClarificationNeeded>
    </Report:input>
  </Report:taskInput>
</Report:createTaskRequest></bpel:literal>
  <bpel:from>
    <bpel:to>$reportCreateTaskRequestMsg.root</bpel:to>
  </bpel:copy>
  <bpel:copy
bpmn:label="$reportNotifyTaskCompletionResponseMsg">
  <bpel:from>
    <bpel:literal>
<Report:response>
  <Report:isChainedAfter></Report:isChainedAfter>
  <Report:taskMetaData>
    <Report:taskId></Report:taskId>
    <Report:taskState></Report:taskState>
    <Report:taskType></Report:taskType>
    <Report:description></Report:description>
    <Report:processId></Report:processId>
    <Report:creationDate></Report:creationDate>
    <Report:userOwner></Report:userOwner>
    <Report:roleOwner></Report:roleOwner>
  <Report:claimAction>
    <Report:user></Report:user>
    <Report:role></Report:role>
  </Report:claimAction>
  <Report:revokeAction>

```

```

        <Report:user></Report:user>
        <Report:role></Report:role>
    </Report:revokeAction>
    <Report:saveAction>
        <Report:user></Report:user>
        <Report:role></Report:role>
    </Report:saveAction>
    <Report:completeAction>
        <Report:user></Report:user>
        <Report:role></Report:role>
    </Report:completeAction>
    <Report:formUrl></Report:formUrl>
    <Report:failureCode></Report:failureCode>
    <Report:failureReason></Report:failureReason>
    <Report:priority></Report:priority>
    <Report:scheduledActions>
        <Report:expiration>
            <Report:until></Report:until>
            <Report:for></Report:for>
        </Report:expiration>
        <Report:deferActivation>
            <Report:until></Report:until>
            <Report:for></Report:for>
        </Report:deferActivation>
    </Report:scheduledActions>

<Report:userProcessCompleteSOAPAction></Report:userProcessCompleteSOAPAction>
    <Report:isChainedBefore></Report:isChainedBefore>
    <Report:previousTaskId></Report:previousTaskId>
    <Report:userProcessEndpoint></Report:userProcessEndpoint>
    <Report:userProcessNamespaceURI></Report:userProcessNamespaceURI>
</Report:taskMetaData>
<Report:status>OK</Report:status>
<Report:errorCode></Report:errorCode>
<Report:errorReason></Report:errorReason>
</Report:response></bpel:literal>
    </bpel:from>

<bpel:to>$reportNotifyTaskCompletionResponseMsg.root</bpel:to>
    </bpel:copy>
    <bpel:copy bpmn:label="$sOWCreateTaskRequestMsg">
        <bpel:from>
            <bpel:literal>
<SOW:createTaskRequest>
    <SOW:taskMetaData>
        <SOW:taskId>_BAb8VnLGEdydN6900C3LYA</SOW:taskId>
        <SOW:taskState></SOW:taskState>
        <SOW:taskType></SOW:taskType>
        <SOW:description>Change Request SOW</SOW:description>
        <SOW:processId></SOW:processId>
        <SOW:creationDate></SOW:creationDate>
        <SOW:userOwner />
        <SOW:roleOwner>appdev\customer</SOW:roleOwner>

```

```

<SOW:claimAction>
  <SOW:user></SOW:user>
  <SOW:role></SOW:role>
</SOW:claimAction>
<SOW:revokeAction>
  <SOW:user></SOW:user>
  <SOW:role></SOW:role>
</SOW:revokeAction>
<SOW:saveAction>
  <SOW:user></SOW:user>
  <SOW:role></SOW:role>
</SOW:saveAction>
<SOW:completeAction>
  <SOW:user></SOW:user>
  <SOW:role></SOW:role>
</SOW:completeAction>
<SOW:formUrl>oxf://ApplicationDevelopment/SOW.xform</SOW:formUrl>
<SOW:failureCode></SOW:failureCode>
<SOW:failureReason></SOW:failureReason>
<SOW:priority></SOW:priority>
<SOW:scheduledActions>
  <SOW:expiration>
    <SOW:until></SOW:until>
    <SOW:for></SOW:for>
  </SOW:expiration>
  <SOW:deferActivation>
    <SOW:until></SOW:until>
    <SOW:for></SOW:for>
  </SOW:deferActivation>
</SOW:scheduledActions>

<SOW:userProcessCompleteSOAPAction>http://example.com/SOW/xform/Process/notifyTaskCompletion</SOW:userProcessCompleteSOAPAction>
  <SOW:isChainedBefore></SOW:isChainedBefore>
  <SOW:previousTaskId></SOW:previousTaskId>

<SOW:userProcessEndpoint>http://localhost:8080/ode/processes/ApplicationDevelopment/ChangeManagement/ChangeManagement/Iniciador/WFmagic_BAb8VnLGEdydN6900C3LYA</SOW:userProcessEndpoint>

<SOW:userProcessNamespaceURI>http://example.com/SOW/xform</SOW:userProcessNamespaceURI>
</SOW:taskMetadata>
<SOW:participantToken></SOW:participantToken>
<SOW:taskInput>
  <SOW:input>
    <SOW:ChangeRequest></SOW:ChangeRequest>
    <SOW:ChangeRequestLabel></SOW:ChangeRequestLabel>
    <SOW:Text2></SOW:Text2>
    <SOW:SOWEstimate></SOW:SOWEstimate>
    <SOW:Text></SOW:Text>
  </SOW:input>
</SOW:taskInput>
</SOW:createTaskRequest></bpel:literal>

```

```

        </bpel:from>
        <bpel:to>$sOWCreateTaskRequestMsg.root</bpel:to>
    </bpel:copy>
    <bpel:copy bpmn:label="$sOWNotifyTaskCompletionResponseMsg">
        <bpel:from>
            <bpel:literal>
<SOW:response>
    <SOW:isChainedAfter></SOW:isChainedAfter>
    <SOW:taskMetaData>
        <SOW:taskId></SOW:taskId>
        <SOW:taskState></SOW:taskState>
        <SOW:taskType></SOW:taskType>
        <SOW:description></SOW:description>
        <SOW:processId></SOW:processId>
        <SOW:creationDate></SOW:creationDate>
        <SOW:userOwner></SOW:userOwner>
        <SOW:roleOwner></SOW:roleOwner>
        <SOW:claimAction>
            <SOW:user></SOW:user>
            <SOW:role></SOW:role>
        </SOW:claimAction>
        <SOW:revokeAction>
            <SOW:user></SOW:user>
            <SOW:role></SOW:role>
        </SOW:revokeAction>
        <SOW:saveAction>
            <SOW:user></SOW:user>
            <SOW:role></SOW:role>
        </SOW:saveAction>
        <SOW:completeAction>
            <SOW:user></SOW:user>
            <SOW:role></SOW:role>
        </SOW:completeAction>
        <SOW:formUrl></SOW:formUrl>
        <SOW:failureCode></SOW:failureCode>
        <SOW:failureReason></SOW:failureReason>
        <SOW:priority></SOW:priority>
        <SOW:scheduledActions>
            <SOW:expiration>
                <SOW:until></SOW:until>
                <SOW:for></SOW:for>
            </SOW:expiration>
            <SOW:deferActivation>
                <SOW:until></SOW:until>
                <SOW:for></SOW:for>
            </SOW:deferActivation>
        </SOW:scheduledActions>

<SOW:userProcessCompleteSOAPAction></SOW:userProcessCompleteSOAPAction>
    <SOW:isChainedBefore></SOW:isChainedBefore>
    <SOW:previousTaskId></SOW:previousTaskId>
    <SOW:userProcessEndpoint></SOW:userProcessEndpoint>
    <SOW:userProcessNamespaceURI></SOW:userProcessNamespaceURI>
</SOW:taskMetaData>

```



```

<SOW:status>OK</SOW:status>
<SOW:errorCode></SOW:errorCode>
<SOW:errorReason></SOW:errorReason>
</SOW:response></bpel:literal>
  </bpel:from>
  <bpel:to>$sOWNotifyTaskCompletionResponseMsg.root</bpel:to>
</bpel:copy>
  <bpel:copy bpmn:label="$clarificationCreateTaskRequestMsg">
    <bpel:from>
      <bpel:literal>
<Clarification:createTaskRequest>
  <Clarification:taskMetaData>
    <Clarification:taskId>_-qtlpnlGEdyFGs4L8GL2DA</Clarification:taskId>
    <Clarification:taskState></Clarification:taskState>
    <Clarification:taskType></Clarification:taskType>
    <Clarification:description>Change Request Needs
Clarification</Clarification:description>
    <Clarification:processId></Clarification:processId>
    <Clarification:creationDate></Clarification:creationDate>
    <Clarification:userOwner />
    <Clarification:roleOwner>appdev\customer</Clarification:roleOwner>
    <Clarification:claimAction>
      <Clarification:user></Clarification:user>
      <Clarification:role></Clarification:role>
    </Clarification:claimAction>
    <Clarification:revokeAction>
      <Clarification:user></Clarification:user>
      <Clarification:role></Clarification:role>
    </Clarification:revokeAction>
    <Clarification:saveAction>
      <Clarification:user></Clarification:user>
      <Clarification:role></Clarification:role>
    </Clarification:saveAction>
    <Clarification:completeAction>
      <Clarification:user></Clarification:user>
      <Clarification:role></Clarification:role>
    </Clarification:completeAction>

<Clarification:formUrl>oxf://ApplicationDevelopment/Clarification.xform</
Clarification:formUrl>
  <Clarification:failureCode></Clarification:failureCode>
  <Clarification:failureReason></Clarification:failureReason>
  <Clarification:priority></Clarification:priority>
  <Clarification:scheduledActions>
    <Clarification:expiration>
      <Clarification:until></Clarification:until>
      <Clarification:for></Clarification:for>
    </Clarification:expiration>
    <Clarification:deferActivation>
      <Clarification:until></Clarification:until>
      <Clarification:for></Clarification:for>
    </Clarification:deferActivation>
  </Clarification:scheduledActions>

```

```

<Clarification:userProcessCompleteSOAPAction>http://example.com/Clarification/xform/Process/notifyTaskCompletion</Clarification:userProcessCompleteSOAPAction>
  <Clarification:isChainedBefore></Clarification:isChainedBefore>
  <Clarification:previousTaskId></Clarification:previousTaskId>

<Clarification:userProcessEndpoint>http://localhost:8080/ode/processes/ApplicationDevelopment/ChangeManagement/ChangeManagement/Iniciador/WFmagic_-qtlpnLGEdyFGs4L8GL2DA</Clarification:userProcessEndpoint>

<Clarification:userProcessNamespaceURI>http://example.com/Clarification/xform</Clarification:userProcessNamespaceURI>
  </Clarification:taskMetaData>
  <Clarification:participantToken></Clarification:participantToken>
  <Clarification:taskInput>
    <Clarification:input>

<Clarification:ClarificationNeeded></Clarification:ClarificationNeeded>
  <Clarification:Text></Clarification:Text>
  <Clarification:Text2></Clarification:Text2>
  <Clarification:ChangeRequest></Clarification:ChangeRequest>
  </Clarification:input>
</Clarification:taskInput>
</Clarification:createTaskRequest></bpel:literal>
  </bpel:from>
  <bpel:to>$clarificationCreateTaskRequestMsg.root</bpel:to>
</bpel:copy>
<bpel:copy
bpmn:label="$clarificationNotifyTaskCompletionResponseMsg">
  <bpel:from>
  <bpel:literal>
<Clarification:response>
  <Clarification:isChainedAfter></Clarification:isChainedAfter>
  <Clarification:taskMetaData>
    <Clarification:taskId></Clarification:taskId>
    <Clarification:taskState></Clarification:taskState>
    <Clarification:taskType></Clarification:taskType>
    <Clarification:description></Clarification:description>
    <Clarification:processId></Clarification:processId>
    <Clarification:creationDate></Clarification:creationDate>
    <Clarification:userOwner></Clarification:userOwner>
    <Clarification:roleOwner></Clarification:roleOwner>
  <Clarification:claimAction>
    <Clarification:user></Clarification:user>
    <Clarification:role></Clarification:role>
  </Clarification:claimAction>
  <Clarification:revokeAction>
    <Clarification:user></Clarification:user>
    <Clarification:role></Clarification:role>
  </Clarification:revokeAction>
  <Clarification:saveAction>
    <Clarification:user></Clarification:user>
    <Clarification:role></Clarification:role>

```

```

</Clarification:saveAction>
<Clarification:completeAction>
  <Clarification:user></Clarification:user>
  <Clarification:role></Clarification:role>
</Clarification:completeAction>
<Clarification:formUrl></Clarification:formUrl>
<Clarification:failureCode></Clarification:failureCode>
<Clarification:failureReason></Clarification:failureReason>
<Clarification:priority></Clarification:priority>
<Clarification:scheduledActions>
  <Clarification:expiration>
    <Clarification:until></Clarification:until>
    <Clarification:for></Clarification:for>
  </Clarification:expiration>
  <Clarification:deferActivation>
    <Clarification:until></Clarification:until>
    <Clarification:for></Clarification:for>
  </Clarification:deferActivation>
</Clarification:scheduledActions>

<Clarification:userProcessCompleteSOAPAction></Clarification:userProcessC
ompleteSOAPAction>
  <Clarification:isChainedBefore></Clarification:isChainedBefore>
  <Clarification:previousTaskId></Clarification:previousTaskId>

<Clarification:userProcessEndpoint></Clarification:userProcessEndpoint>

<Clarification:userProcessNamespaceURI></Clarification:userProcessNamespa
ceURI>
  </Clarification:taskMetaData>
  <Clarification:status>OK</Clarification:status>
  <Clarification:errorCode></Clarification:errorCode>
  <Clarification:errorReason></Clarification:errorReason>
</Clarification:response></bpel:literal>
  </bpel:from>

<bpel:to>$clarificationNotifyTaskCompletionResponseMsg.root</bpel:to>
  </bpel:copy>
  </bpel:assign>
  <bpel:assign bpmn:label="Peticion evaluación a Desarrollador"
bpmn:id="_6NGQAHK0EdyyJdv25eCjJg">
  <bpel:copy>

<bpel:from>$assignDesarrolladorNotifyTaskCompletionRequestMsg.root/Assign
Developer:taskOutput/AssignDeveloper:output/AssignDeveloper:Developer</bp
el:from>

<bpel:to>$estimationCreateTaskRequestMsg.root/Estimation:taskInput/Estima
tion:input/Estimation:DeveloperToAsk</bpel:to>
  </bpel:copy>
  <bpel:copy>

<bpel:from>$xformInitProcessRequestMsg.root/xform:taskOutput/xform:output
/xform:ChangeRequest</bpel:from>

```

```

<bpel:to>$estimationCreateTaskRequestMsg.root/Estimation:taskInput/Estima
tion:input/Estimation:ChangeRequest</bpel:to>
    </bpel:copy>
    <bpel:copy>

<bpel:from>$assignDesarrolladorNotifyTaskCompletionRequestMsg.root/Assign
Developer:taskOutput/AssignDeveloper:output/AssignDeveloper:Developer</bp
el:from>

<bpel:to>$estimationCreateTaskRequestMsg.root/Estimation:taskMetaData/Est
imation:userOwner</bpel:to>
    </bpel:copy>
    </bpel:assign>
    <bpel:assign>
    <bpel:copy>

<bpel:from>"http://localhost:8080/ode/processes/ApplicationDevelopment/Ch
angeManagement/ChangeManagement/Desarrollador/WFmagic_Sp58IHLAEdyL6qoRBK0
TuA"</bpel:from>

<bpel:to>$estimationCreateTaskRequestMsg.root/Estimation:taskMetaData/Est
imation:userProcessEndpoint</bpel:to>
    </bpel:copy>
    <bpel:copy>

<bpel:from>"http://example.com/Estimation/xform/Process/notifyTaskComple
tion"</bpel:from>

<bpel:to>$estimationCreateTaskRequestMsg.root/Estimation:taskMetaData/Est
imation:userProcessCompleteSOAPAction</bpel:to>
    </bpel:copy>
    <bpel:copy>
        <bpel:from>"_SqDtJXLAEdyL6qoRBK0TuA"</bpel:from>

<bpel:to>$estimationCreateTaskRequestMsg.root/Estimation:taskMetaData/Est
imation:taskId</bpel:to>
    </bpel:copy>
    <bpel:copy>
        <bpel:from>"Provide Estimate for Change Request"</bpel:from>

<bpel:to>$estimationCreateTaskRequestMsg.root/Estimation:taskMetaData/Est
imation:description</bpel:to>
    </bpel:copy>
    </bpel:assign>
    <bpel:invoke partnerLink="wFmagic_Sp58IHLAEdyL6qoRBK0TuAPlkVar "
portType="Estimation:Workflow" operation="createTask"
inputVariable="estimationCreateTaskRequestMsg"
outputVariable="estimationCreateTaskResponseMsg" bpmn:label="Petición
evaluación a Desarrollador"
bpmn:id="_6NGQAHK0EddyJdv25eCjJg"></bpel:invoke>
        <bpel:receive
partnerLink="wFmagic_Sp58IHLAEdyL6qoRBK0TuAPlkVar "
portType="Estimation:Process" operation="notifyTaskCompletion"

```

```

variable="estimationNotifyTaskCompletionRequestMsg" bpmn:label="Recibir
Reporte de Desarrollador"
bpmn:id="_PfDxwHK1EddyJdv25eCjJg"></bpel:receive>
  <bpel:reply partnerLink="wFmagic_Sp58IHLAEdyL6qoRBK0TuAPlkVar"
portType="Estimation:Process" operation="notifyTaskCompletion"
variable="estimationNotifyTaskCompletionResponseMsg" bpmn:label="Recibir
Reporte de Desarrollador" bpmn:id="_PfDxwHK1EddyJdv25eCjJg"></bpel:reply>
  <bpel:assign bpmn:label="Enviar Reporte a Encargado"
bpmn:id="_f4s5sHK1EddyJdv25eCjJg">
    <bpel:copy>

<bpel:from>$xformInitProcessRequestMsg.root/xform:taskOutput/xform:output
/xform:ChangeRequest</bpel:from>

<bpel:to>$reportCreateTaskRequestMsg.root/Report:taskInput/Report:input/R
eport:ChangeRequest</bpel:to>
  </bpel:copy>
  <bpel:copy>

<bpel:from>$estimationNotifyTaskCompletionRequestMsg.root/Estimation:task
Output/Estimation:output/Estimation:Estimate</bpel:from>

<bpel:to>$reportCreateTaskRequestMsg.root/Report:taskInput/Report:input/R
eport:Estimate</bpel:to>
  </bpel:copy>
  <bpel:copy>

<bpel:from>$estimationNotifyTaskCompletionRequestMsg.root/Estimation:task
Output/Estimation:output/Estimation:DeveloperToAsk</bpel:from>

<bpel:to>$reportCreateTaskRequestMsg.root/Report:taskInput/Report:input/R
eport:DeveloperToAsk</bpel:to>
  </bpel:copy>
  <bpel:copy>

<bpel:from>$estimationNotifyTaskCompletionRequestMsg.root/Estimation:task
Output/Estimation:output/Estimation:ClarificationNeeded</bpel:from>

<bpel:to>$reportCreateTaskRequestMsg.root/Report:taskInput/Report:input/R
eport:ClarificationNeeded</bpel:to>
  </bpel:copy>
  </bpel:assign>
  <bpel:assign>
  <bpel:copy>

<bpel:from>"http://localhost:8080/ode/processes/ApplicationDevelopment/Ch
angeManagement/ChangeManagement/Encargado/WFmagic_a3iaCHLCEdyL6qoRBK0TuA"
</bpel:from>

<bpel:to>$reportCreateTaskRequestMsg.root/Report:taskMetaData/Report:user
ProcessEndpoint</bpel:to>
  </bpel:copy>
  <bpel:copy>

```

```

<bpel:from>"http://example.com/Report/xform/Process/notifyTaskCompletion"
</bpel:from>

<bpel:to>$reportCreateTaskRequestMsg.root/Report:taskMetaData/Report:user
ProcessCompleteSOAPAction</bpel:to>
  </bpel:copy>
  <bpel:copy>
    <bpel:from>"_a3rj83LCEdyL6qoRBK0TuA"</bpel:from>

<bpel:to>$reportCreateTaskRequestMsg.root/Report:taskMetaData/Report:task
Id</bpel:to>
  </bpel:copy>
  <bpel:copy>
    <bpel:from>"Review Report from Developer"</bpel:from>

<bpel:to>$reportCreateTaskRequestMsg.root/Report:taskMetaData/Report:desc
ription</bpel:to>
  </bpel:copy>
  </bpel:assign>
  <bpel:invoke partnerLink="wFmagic_a3iaCHLCEdyL6qoRBK0TuAPlkVar"
portType="Report:Workflow" operation="createTask"
inputVariable="reportCreateTaskRequestMsg"
outputVariable="reportCreateTaskResponseMsg" bpmn:label="Enviar Reporte a
Encargado" bpmn:id="_f4s5SHK1EdyyJdv25eCjJg"></bpel:invoke>
    <bpel:receive
partnerLink="wFmagic_a3iaCHLCEdyL6qoRBK0TuAPlkVar"
portType="Report:Process" operation="notifyTaskCompletion"
variable="reportNotifyTaskCompletionRequestMsg" bpmn:label="Receive
Manager Decision" bpmn:id="_acfTMHK2EdyyJdv25eCjJg"></bpel:receive>
      <bpel:reply partnerLink="wFmagic_a3iaCHLCEdyL6qoRBK0TuAPlkVar"
portType="Report:Process" operation="notifyTaskCompletion"
variable="reportNotifyTaskCompletionResponseMsg" bpmn:label="Receive
Manager Decision" bpmn:id="_acfTMHK2EdyyJdv25eCjJg"></bpel:reply>
    <bpel:if>

<bpel:condition>$reportNotifyTaskCompletionRequestMsg.root/Report:taskOut
put/Report:output/Report:Decision = "sow"</bpel:condition>
  <bpel:sequence>
    <bpel:assign bpmn:label="Enviar SOW"
bpmn:id="_iU5Z4HK2EdyyJdv25eCjJg">
      <bpel:copy>

<bpel:from>$xformInitProcessRequestMsg.root/xform:taskOutput/xform:output
/xform:ChangeRequest</bpel:from>

<bpel:to>$sOWCreateTaskRequestMsg.root/SOW:taskInput/SOW:input/SOW:Change
Request</bpel:to>
  </bpel:copy>
  <bpel:copy>

<bpel:from>$reportNotifyTaskCompletionRequestMsg.root/Report:taskOutput/R
eport:output/Report:SOWEstimate</bpel:from>

```

```

<bpel:to>$sOWCreateTaskRequestMsg.root/SOW:taskInput/SOW:input/SOW:SOWEstimate</bpel:to>
    </bpel:copy>
</bpel:assign>
<bpel:assign>
    <bpel:copy>

<bpel:from>"http://localhost:8080/ode/processes/ApplicationDevelopment/ChangeManagement/ChangeManagement/Iniciador/WFmagic_BAb8VnLGEdydN6900C3LYA"
</bpel:from>

<bpel:to>$sOWCreateTaskRequestMsg.root/SOW:taskMetaData/SOW:userProcessEndpoint</bpel:to>
    </bpel:copy>
    <bpel:copy>

<bpel:from>"http://example.com/SOW/xform/Process/notifyTaskCompletion"</bpel:from>

<bpel:to>$sOWCreateTaskRequestMsg.root/SOW:taskMetaData/SOW:userProcessCompleteSOAPAction</bpel:to>
    </bpel:copy>
    <bpel:copy>
        <bpel:from>"_BAb8XXLGEdydN6900C3LYA"</bpel:from>

<bpel:to>$sOWCreateTaskRequestMsg.root/SOW:taskMetaData/SOW:taskId</bpel:to>
    </bpel:copy>
    <bpel:copy>
        <bpel:from>"Change Request SOW"</bpel:from>

<bpel:to>$sOWCreateTaskRequestMsg.root/SOW:taskMetaData/SOW:description</bpel:to>
    </bpel:copy>
    </bpel:assign>
    <bpel:invoke partnerLink="wFmagic_BAb8VnLGEdydN6900C3LYAPlkVar" portType="SOW:Workflow" operation="createTask"
inputVariable="sOWCreateTaskRequestMsg"
outputVariable="sOWCreateTaskResponseMsg" bpmn:label="Enviar SOW"
bpmn:id="_iU5Z4HK2EddyJdv25eCjJg"></bpel:invoke>
        <bpel:receive
partnerLink="wFmagic_BAb8VnLGEdydN6900C3LYAPlkVar" portType="SOW:Process"
operation="notifyTaskCompletion"
variable="sOWNotifyTaskCompletionRequestMsg" bpmn:label="SOW Aprobado"
bpmn:id="_8_OHcHK2EddyJdv25eCjJg"></bpel:receive>
            <bpel:reply
partnerLink="wFmagic_BAb8VnLGEdydN6900C3LYAPlkVar" portType="SOW:Process"
operation="notifyTaskCompletion"
variable="sOWNotifyTaskCompletionResponseMsg" bpmn:label="SOW Aprobado"
bpmn:id="_8_OHcHK2EddyJdv25eCjJg"></bpel:reply>
                <bpel:assign bpmn:label="EventEndEmpty"
bpmn:id="_00E1gHK3EddyJdv25eCjJg">
                    <bpel:copy>

```

```

        <bpel:from>>false()</bpel:from>
        <bpel:to>$reEvaluate</bpel:to>
    </bpel:copy>
</bpel:assign>
</bpel:sequence>
<bpel:elseif>

<bpel:condition>$reportNotifyTaskCompletionRequestMsg.root/Report:taskOutput/Report:output/Report:Decision = "implement"</bpel:condition>
    <bpel:sequence>
        <bpel:assign bpmn:label="EventEndEmpty"
bpmn:id="_20q0QHK3EdyyJdv25eCjJg">
            <bpel:copy>
                <bpel:from>>false()</bpel:from>
                <bpel:to>$reEvaluate</bpel:to>
            </bpel:copy>
        </bpel:assign>
    </bpel:sequence>
</bpel:elseif>
<bpel:elseif>

<bpel:condition>$reportNotifyTaskCompletionRequestMsg.root/Report:taskOutput/Report:output/Report:Decision = "clarify"</bpel:condition>
    <bpel:sequence>
        <bpel:assign bpmn:label="Preguntar Iniciador"
bpmn:id="_SykfAHK3EdyyJdv25eCjJg">
            <bpel:copy>

<bpel:from>$reportNotifyTaskCompletionRequestMsg.root/Report:taskOutput/Report:output/Report:ClarificationNeeded</bpel:from>

<bpel:to>$clarificationCreateTaskRequestMsg.root/Clarification:taskInput/Clarification:input/Clarification:ClarificationNeeded</bpel:to>
            </bpel:copy>
            <bpel:copy>

<bpel:from>$xformInitProcessRequestMsg.root/xform:taskOutput/xform:output/xform:ChangeRequest</bpel:from>

<bpel:to>$clarificationCreateTaskRequestMsg.root/Clarification:taskInput/Clarification:input/Clarification:ChangeRequest</bpel:to>
            </bpel:copy>
        </bpel:assign>
    </bpel:assign>
    <bpel:copy>

<bpel:from>"http://localhost:8080/ode/processes/ApplicationDevelopment/ChangeManagement/ChangeManagement/Iniciador/WFmagic_qt1pnLGEdyFGs4L8GL2DA"</bpel:from>

<bpel:to>$clarificationCreateTaskRequestMsg.root/Clarification:taskMetadata/Clarification:userProcessEndpoint</bpel:to>
            </bpel:copy>
            <bpel:copy>

```



```

<bpel:from>"http://example.com/Clarification/xform/Process/notifyTaskCompletion"</bpel:from>

<bpel:to>$clarificationCreateTaskRequestMsg.root/Clarification:taskMetadata/Clarification:userProcessCompleteSOAPAction</bpel:to>
  </bpel:copy>
  <bpel:copy>
    <bpel:from>"_qtlrXLGEdyFGs4L8GL2DA"</bpel:from>

<bpel:to>$clarificationCreateTaskRequestMsg.root/Clarification:taskMetadata/Clarification:taskId</bpel:to>
  </bpel:copy>
  <bpel:copy>
    <bpel:from>"Change Request Needs Clarification"</bpel:from>

<bpel:to>$clarificationCreateTaskRequestMsg.root/Clarification:taskMetadata/Clarification:description</bpel:to>
  </bpel:copy>
  </bpel:assign>
  <bpel:invoke partnerLink="wFmagic_-qtlpnLGEdyFGs4L8GL2DAPlkVar"
portType="Clarification:Workflow" operation="createTask"
inputVariable="clarificationCreateTaskRequestMsg"
outputVariable="clarificationCreateTaskResponseMsg" bpmn:label="Preguntar Iniciador" bpmn:id="_SykfAHK3EdyyJdv25eCjJg"></bpel:invoke>
    <bpel:receive partnerLink="wFmagic_-qtlpnLGEdyFGs4L8GL2DAPlkVar" portType="Clarification:Process"
operation="notifyTaskCompletion"
variable="clarificationNotifyTaskCompletionRequestMsg"
bpmn:label="Recibir Informacion"
bpmn:id="_dTMMHK3EdyyJdv25eCjJg"></bpel:receive>
      <bpel:reply partnerLink="wFmagic_-qtlpnLGEdyFGs4L8GL2DAPlkVar" portType="Clarification:Process"
operation="notifyTaskCompletion"
variable="clarificationNotifyTaskCompletionResponseMsg"
bpmn:label="Recibir Informacion"
bpmn:id="_dTMMHK3EdyyJdv25eCjJg"></bpel:reply>
        <bpel:assign bpmn:label="Peticion actualizada"
bpmn:id="_2HlecHL8EdyDAusiggh1HA">
          <bpel:copy>

<bpel:from>$clarificationNotifyTaskCompletionRequestMsg.root/Clarification:taskOutput/Clarification:output/Clarification:ChangeRequest</bpel:from>

<bpel:to>$estimationCreateTaskRequestMsg.root/Estimation:taskInput/Estimation:input/Estimation:ChangeRequest</bpel:to>
  </bpel:copy>
  </bpel:assign>
  </bpel:sequence>
  </bpel:elseif>
  <bpel:elseif>

<bpel:condition>$reportNotifyTaskCompletionRequestMsg.root/Report:taskOutput/Report:output/Report:Decision = "reassign"</bpel:condition>

```

```

        <bpel:sequence>
            <bpel:assign bpmn:label="Reasignar Petición a otro
desarrollador" bpmn:id="_yj9CCHK2EdyyJdv25eCjJg">
                <bpel:copy>

<bpel:from>$estimationNotifyTaskCompletionRequestMsg.root/Estimation:task
Output/Estimation:output/Estimation:DeveloperToAsk</bpel:from>

<bpel:to>$assignDesarrolladorNotifyTaskCompletionRequestMsg.root/AssignDe
veloper:taskOutput/AssignDeveloper:output/AssignDeveloper:Developer</bpel
:to>
                    </bpel:copy>
                </bpel:assign>
            </bpel:sequence>
        </bpel:elseif>
    </bpel:if>
</bpel:sequence>
</bpel:while>
</bpel:scope>
    <bpel:assign bpmn:label="Asignar trabajo a desarrollador"
bpmn:id="_Psaw8HK4EdyyJdv25eCjJg">
        <bpel:copy>

<bpel:from>$xformInitProcessRequestMsg.root/xform:taskOutput/xform:output
/xform:ChangeRequest</bpel:from>

<bpel:to>$workCreateTaskRequestMsg.root/Work:taskInput/Work:input/Work:Ch
angeRequest</bpel:to>
        </bpel:copy>
    <bpel:copy>

<bpel:from>$assignDesarrolladorNotifyTaskCompletionRequestMsg.root/Assign
Developer:taskOutput/AssignDeveloper:output/AssignDeveloper:Developer</bpel
:from>

<bpel:to>$workCreateTaskRequestMsg.root/Work:taskMetaData/Work:userOwner<
/bpel:to>
        </bpel:copy>
    </bpel:assign>
    <bpel:assign>
    <bpel:copy>

<bpel:from>"http://localhost:8080/ode/processes/ApplicationDevelopment/Ch
angeManagement/ChangeManagement/Desarrollador/WFmagic_NQVgUHLJEJdyqb04eiX9
ZdA"</bpel:from>

<bpel:to>$workCreateTaskRequestMsg.root/Work:taskMetaData/Work:userProces
sEndpoint</bpel:to>
        </bpel:copy>
    <bpel:copy>

<bpel:from>"http://example.com/Work/xform/Process/notifyTaskCompletion"</
bpel:from>

```

```

<bpel:to>$workCreateTaskRequestMsg.root/Work:taskMetaData/Work:userProcessesCompleteSOAPAction</bpel:to>
  </bpel:copy>
  <bpel:copy>
    <bpel:from>"_NQVgV3LJEJdyqb04eiX9ZdA"</bpel:from>

<bpel:to>$workCreateTaskRequestMsg.root/Work:taskMetaData/Work:taskId</bpel:to>
  </bpel:copy>
  <bpel:copy>
    <bpel:from>"Change Request confirmed"</bpel:from>

<bpel:to>$workCreateTaskRequestMsg.root/Work:taskMetaData/Work:description</bpel:to>
  </bpel:copy>
  </bpel:assign>
  <bpel:invoke partnerLink="wFmagic_NQVgUHLJEJdyqb04eiX9ZdAPlkVar"
portType="Work:Workflow" operation="createTask"
inputVariable="workCreateTaskRequestMsg"
outputVariable="workCreateTaskResponseMsg" bpmn:label="Asignar trabajo a desarrollador" bpmn:id="_Psaw8HK4EdyyJdv25eCjJg"></bpel:invoke>
  <bpel:receive partnerLink="wFmagic_NQVgUHLJEJdyqb04eiX9ZdAPlkVar"
portType="Work:Process" operation="notifyTaskCompletion"
variable="workNotifyTaskCompletionRequestMsg" bpmn:label="Trabajo completado" bpmn:id="_qFwcMHK4EdyyJdv25eCjJg"></bpel:receive>
  <bpel:reply partnerLink="wFmagic_NQVgUHLJEJdyqb04eiX9ZdAPlkVar"
portType="Work:Process" operation="notifyTaskCompletion"
variable="workNotifyTaskCompletionResponseMsg" bpmn:label="Trabajo completado" bpmn:id="_qFwcMHK4EdyyJdv25eCjJg"></bpel:reply>
  <bpel:assign bpmn:label="Notificar encargado"
bpmn:id="_z4lYcHLJEJdyqb04eiX9ZdA">
    <bpel:copy>

<bpel:from>$xformInitProcessRequestMsg.root/xform:taskOutput/xform:output/xform:ChangeRequest</bpel:from>

<bpel:to>$workCompletedNotifyRequestMsg.root/WorkCompleted:input/WorkCompleted:input/WorkCompleted:ChangeRequest</bpel:to>
  </bpel:copy>
  <bpel:copy>

<bpel:from>$assignDesarrolladorNotifyTaskCompletionRequestMsg.root/AssignDeveloper:taskOutput/AssignDeveloper:output/AssignDeveloper:Developer</bpel:from>

<bpel:to>$workCompletedNotifyRequestMsg.root/WorkCompleted:input/WorkCompleted:input/WorkCompleted:Developer</bpel:to>
  </bpel:copy>
  </bpel:assign>
  <bpel:assign>
  <bpel:copy>
    <bpel:from>"Change Request Completed"</bpel:from>

```

```
<bpel:to>$workCompletedNotifyRequestMsg.root/WorkCompleted:metadata/WorkC
ompleted:description</bpel:to>
  </bpel:copy>
</bpel:assign>
<bpel:invoke
partnerLink="encargadoAndChangeManagementForXformPortPlkVar"
portType="WorkCompleted:Workflow" operation="notify"
inputVariable="workCompletedNotifyRequestMsg"
outputVariable="workCompletedNotifyResponseMsg" bpmn:label="Notificar
encargado" bpmn:id="_z4lYcHLJEdyqb04eiX9ZdA"></bpel:invoke>
  <bpel:empty bpmn:label="EventEndEmpty"
bpmn:id="_DwGqwHLKEdyqb04eiX9ZdA"></bpel:empty>
  </bpel:sequence>
</bpel:process>
```