



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



**Sistema de Información
para la Gestión de Versiones
y Liberaciones de Software**

Caso de estudio: Gerencia de Programa de Soluciones
de Tecnologías de Información de la Empresa
Compañía Anónima Nacional de Teléfonos de Venezuela

Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
por la Bachiller

Lily Tsang Tong
para optar al título de
Licenciado en Computación

Tutor Dr. Pedro N. Bonillo R.

Junio, 2012

ACTA

Quienes suscriben, miembros del Jurado designado por el Consejo de Escuela de Computación, para examinar el Trabajo Especial de Grado presentado por la Bachiller Lily Tsang Tong CI. V-16.461.501, con el título: “**Sistema de Información para la Gestión de Versiones y Liberaciones de Software**”. Caso de estudio: Gerencia de Programa de Soluciones de Tecnologías de Información de la Empresa Compañía Anónima Nacional de Teléfonos de Venezuela, 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 7 de junio de 2012 a las 4:30 p.m, para que el autor lo defendiera en forma pública, lo que se hizo en la Sala de Postgrado de la Escuela de Computación, 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.

En fe de lo cual se levanta la presente Acta, en Caracas a los 7 días del mes de junio del año 2012.

Profesor Pedro Bonillo

Tutor

Profesora Paola Saputelli

Jurado

Profesor Carlos Acosta

Jurado

AGRADECIMIENTOS

A Dios.

A mis padres, por su constancia, dedicación y sacrificio, por apoyarme en todo momento y por inculcarme que se puede alcanzar cualquier meta si uno se lo propone.

A mis hermanos Livia y John, por su apoyo incondicional, por ser mis mejores amigos, por darme el ejemplo a seguir y por creer en mí.

A mi tutor, Prof. Pedro Bonillo, por haberme dado su mejor disposición en cada una de las consultas. Gracias Prof. Pedro, por su paciencia y dedicación.

A mi mana Jorge Valdéz, por haber compartido los buenos y malos momentos durante todo el transcurso de la carrera y por compartir su amistad.

A Keyla Ferreira, por tenderme una mano amiga para guiarme y apoyarme en este logro.

A mi amigo Julio Montaña, por apoyarme y compartir sus conocimientos de Intalio BPMS y Mule ESB.

A mis amigos Mayerling Mendoza, Deiby González y Sergio Vera por brindarme soporte con respecto a cómo la empresa CANTV realiza pruebas a las aplicaciones.

A cada uno de los profesores de la Escuela de Computación por compartir sus conocimientos y experiencias.

A todas aquellas personas no nombradas, pero que de una u otra manera prestaron su excelente colaboración.

A todos, muchísimas gracias.

RESUMEN

La Biblioteca de Infraestructura de Tecnologías de Información (ITIL) propone un conjunto de mejores prácticas para el proceso de Gestión de Versiones y Liberaciones debido a la necesidad de controlar la calidad de las versiones de hardware y software en el entorno de producción.

Es por ello que la finalidad de este proyecto es diseñar y construir un Sistema de Información para la Gestión de Versiones y Liberaciones de Software según ITIL v3 para la Gerencia de Programa de Soluciones TI de la empresa CANTV el cual asegure que se lleve a cabo el control de los desarrollos de software haciendo un seguimiento desde el momento del diseño hasta más allá de su implementación.

El proceso de Gestión de Versiones y Liberaciones consta de nueve subprocesos: política de liberación, plan de liberación, diseño y obtención de hardware y software, construcción y configuración de la liberación, prueba, aceptación de liberación, plan de implementación, comunicación y adiestramiento y distribución e instalación.

Antes de crear el sistema fue necesario realizar una evaluación de Sistemas de Gestión de Procesos de Negocio (BPMS) bajo código abierto tomando en cuenta los requerimientos del caso de estudio para la implementación del proceso.

El BPMS seleccionado fue Intalio *Community Edition* 6.0.2 ya que permite modelar, analizar, simular y mejorar el proceso en una plataforma bajo software libre.

Palabras claves: Gestión de Versiones y Liberaciones, Intalio *Community*, ITIL, Biblioteca de Infraestructura de Tecnologías de Información.

ABSTRACT

The Information Technology Infrastructure Library (ITIL) offers a set of best practices for the Release and Deployment Management process due to the need to control the quality of hardware and software versions in the production environment.

That is why the goal of this project is to design and build an Information System for Software Release and Deployment Management according to ITIL v3 for CANTV Program Management of IT Solutions Department, this will ensure that the software development control is carried out through a tracing since the moment of conception to its implementation.

The Release and Deployment Management process consists of nine stages: release policy, release plan, design and procurement of hardware and software, building and configuring the release, testing, release acceptance, roll-out plan, communication and training and distribution and installation.

Before creating the system was necessary to make an evaluation of open source Business Process Management Systems (BPMS) taking into account the requirements of the study case for the implementation process.

The BPMS Intalio Community Edition 6.0.2 was selected because it allows us to model, analyze, simulate and improve the process on a free software platform.

Keywords: Release and Deployment Management, Intalio Community, ITIL, Information Technologies Infrastructure Library.

ÍNDICE

ACTA.....	i
AGRADECIMIENTOS	ii
RESUMEN.....	iii
ABSTRACT	iv
ÍNDICE	v
ÍNDICE DE FIGURAS.....	vi
ÍNDICE DE TABLAS.....	ix
INTRODUCCIÓN.....	1
CAPÍTULO I: EL PROBLEMA.....	3
1.1 Antecedentes de la Investigación.....	3
1.2 Formulación del Problema.....	5
1.3 Justificación	6
1.4 Objetivos	7
1.4.1 Objetivo General	7
1.4.2 Objetivos Específicos	7
1.5 Alcance	8
CAPÍTULO II: MARCO TEÓRICO	9
2.1 Gestión de Versiones y Liberaciones bajo la Biblioteca de Infraestructura de Tecnologías de la Información	9
2.1.1 Gestión de Servicios de Tecnologías de la Información.....	9
2.1.2 Biblioteca de Infraestructura de Tecnologías de la Información.....	11
2.1.3 Versión	15
2.1.4 Gestión de Versiones y Liberaciones de ITIL v3	20
2.1.4.1 Proceso de Gestión de Versiones y Liberaciones de ITIL v3.....	21
2.2 Sistemas de Información para la Gestión de Procesos de Negocio	29
2.2.1 Sistemas de Información.....	29
2.2.2 Proceso de Negocio	31
2.2.3 Gestión de Procesos de Negocio.....	32
2.2.4 Sistemas de Gestión de Procesos de Negocio.....	36
2.2.4.1 Software de implementación: BPMS Intalio <i>Community Edition</i> 6.0.2	39
2.2.4.2 Servicios Web	42
2.2.4.3 Arquitectura Orientada a Servicios	43
2.2.4.4 Protocolo Simple de Acceso a Objetos	44
2.2.4.5 Lenguaje de Descripción de Servicios Web	46
2.2.4.6 Descubrimiento, Descripción e Integración Universal	48
2.2.4.7 Orquestación de Servicios Web	50

CAPÍTULO III: MARCO METODOLÓGICO.....	52
3.1 Bases metodológicas de la investigación.....	52
3.1.1 Tipo de investigación.....	52
3.1.2 Fuentes de Revisión e Investigación	53
3.2 Metodología de Desarrollo de Proyectos BPM	54
3.2.1 Metodología de Gestión de los Procesos de Negocio Sustentada en el uso de Patrones	55
3.2.1.1 Crear Proceso.....	56
3.3 Procedimiento de Evaluación.....	120
3.4 Resultados.....	125
CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES.....	126
REFERENCIAS BIBLIOGRÁFICAS.....	129
ANEXOS.....	132
Anexo 1 Aspectos de la organización de estudio CANTV	132
Anexo 2 Elementos BPMN de la paleta de Intalio <i>Designer</i>	138
Anexo 3 Características de Intalio <i>Designer</i>	140
Anexo 4 Características de Intalio <i>Server</i>	141
Anexo 5 Arquitectura Orientada a Servicios.....	143
Anexo 6 Aspectos de la Orquestación de Servicios.....	149
Anexo 7 Cartelera de Actividades	154
Anexo 8 PostgreSQL 8.3.10.....	155
Anexo 9 Mule ESB.....	158
Anexo 10 Manual Administrativo	161
Anexo 11 Manual de Usuario	176
Anexo 12 Encuesta ITIL – Gestión de Versiones y Liberaciones	198
Anexo 13 Encuesta Sistema de Información para la Gestión de Versiones y Liberaciones de software	199
Anexo 14 Glosario	200

ÍNDICE DE FIGURAS

Figura 1: Procesos de Transición del Servicio ITIL v3.....	14
Figura 2: Numeración de versiones.....	19
Figura 3: Pasos del Proceso de Gestión de Versiones y Liberaciones ITIL v3.	28
Figura 4: Modelo general de los Sistemas de Información.	29
Figura 5: Tipos de Sistemas de Información.....	30
Figura 6: Actividades del paradigma BPM.	33
Figura 7: Ciclo de vida BPM.....	35

Figura 8: Modelo general de metodologías BPM.....	36
Figura 9: Componentes de un BPMS.	39
Figura 10: Paleta de Intalio <i>Designer</i>	41
Figura 11: Formato de un mensaje SOAP.	45
Figura 12: Mensaje SOAP.....	46
Figura 13: Estructura y ejemplo de un archivo WSDL.	47
Figura 14: Estructura de Datos de UDDI.	50
Figura 15: BPD del proceso de la Metodología de Gestión de los Procesos Sustentada en el uso de Patrones.	55
Figura 16: Nivel 0 Sistema de Gestión de Versiones y Liberaciones de Software.....	60
Figura 17: Nivel 1 Sistema de Gestión de Versiones y Liberaciones de Software.....	61
Figura 18: Nivel 2 Establecer Políticas de Liberación.....	62
Figura 19: Nivel 2 Planificar Liberación.....	63
Figura 20: Nivel 2 Diseñar Versión.	63
Figura 21: Nivel 2 Construir Versión.	64
Figura 22: Nivel 2 Realizar Pruebas.	64
Figura 23: Nivel 2 Evaluar Aceptación de Liberación.	65
Figura 24: Nivel 2 Planificar Implementación.....	65
Figura 25: Nivel 2 Comunicar y Adiestrar Usuarios.	66
Figura 26: Nivel 2 Distribuir e Instalar Versión.....	66
Figura 27: Nivel 2 Archivar Cambios.....	67
Figura 28: Diagrama de Clases del Proceso de Gestión de Versiones.....	70
Figura 29: Diagrama de Secuencia del Proceso de Gestión de Versiones.	71
Figura 30: Diagrama de Estado del Proceso de Gestión de Versiones.....	72
Figura 31: Diagrama de Actividad del Proceso de Gestión de Versiones.	73
Figura 32: Diagrama de Despliegue del Proceso de Gestión de Versiones.....	74
Figura 33: Diagrama de Modelo de Objetos del Dominio del Proceso de Gestión de Versiones	75
Figura 34: Arquitectura del Sistema de Información de Gestión de Versiones y Liberaciones. 76	
Figura 35: BPD del Proceso de Gestión de Versiones y Liberaciones de Software.....	77
Figura 36: Política y Plan de Liberación.	78
Figura 37: Aceptación de Liberación.....	79
Figura 38: Planificar Implementación.	79
Figura 39: Comunicación y Adiestramiento.....	80
Figura 40: Actividad realizada por humano.	83
Figura 41: Autorización de la interacción de Gestión de Cambios.	84
Figura 42: Vista del <i>Mapper</i>	85
Figura 43: Mapeo de la decisión nueva versión.....	85
Figura 44: Base de Datos Gestión de Versiones.	87

Figura 45: BD del Proceso de Gestión de Versiones en PostgreSQL 8.3.10.....	88
Figura 46: Vista de la BD del Proceso de Gestión de Versiones desde Intalio <i>Designer</i>	89
Figura 47: Consulta insertarRFC.sql.....	90
Figura 48: SQL <i>Properties</i> de la consulta insertarRFC.sql.....	90
Figura 49: Mapeo de la consulta insertarRFC.....	92
Figura 50: Consultas del proceso de Gestión de Versiones.....	92
Figura 51: Flujo del WS setVersion.....	95
Figura 52: Mapeo de datos de entrada de setVersion.....	96
Figura 53: Mapeo del dato de salida de setVersion.....	96
Figura 54: Paleta <i>workflow form</i>	97
Figura 55: Formulario init RFC.....	99
Figura 56: Formulario <i>create and complete</i> Plan de Liberación.....	101
Figura 57: Formulario Petición de Cambio.....	105
Figura 58: Tabla RFCs.....	105
Figura 59: Formulario Plan de Liberación.....	106
Figura 60: Tabla Plan Liberación.....	107
Figura 61: Formulario Solicitar forma de desarrollo.....	107
Figura 62: Formulario Desarrollar Versión.....	108
Figura 63: Tabla Versiones.....	108
Figura 64: Formulario Plan de Pruebas Unitarias.....	109
Figura 65: Tabla Plan Pruebas Unitarias.....	109
Figura 66: Formulario Realizar Pruebas Unitarias.....	110
Figura 67: Tabla Plan Pruebas Unitarias.....	110
Figura 68: Formulario Solicitar Aprobación de Resultados de Pruebas Unitarias.....	110
Figura 69: Tabla Prueba Unitaria.....	111
Figura 70: Formulario Plan de Pruebas Integrales.....	111
Figura 71: Tabla Plan Pruebas Integrales.....	111
Figura 72: Formulario Realizar Pruebas Integrales.....	112
Figura 73: Tabla Prueba Integral.....	112
Figura 74: Formulario Pruebas de Aceptación de Usuario.....	112
Figura 75: Tabla Plan Pruebas Aceptación de Usuario.....	112
Figura 76: Formulario Realizar Pruebas de Aceptación de Usuario.....	113
Figura 77: Tabla Prueba Aceptación de Usuario.....	113
Figura 78: Formulario Solicitar Aprobación de Resultados.....	113
Figura 79: Actualizar tabla Prueba Integral.....	114
Figura 80: Actualizar tabla Prueba Aceptación de Usuario.....	114
Figura 81: Formulario Solicitar Aceptación de Liberación.....	114
Figura 82: Actualizar tabla Versiones.....	114

Figura 83: Formulario Plan de Implementación.	115
Figura 84: Tabla Plan Implementación.....	115
Figura 85: Formulario Realizar Implementación.	115
Figura 86: Tabla Implementación.	116
Figura 87: Formulario Plan de Comunicación y Adiestramiento.	116
Figura 88: Tabla Plan Comunicación y Adiestramiento.	116
Figura 89: Formulario Plan de Distribución e Instalación.....	117
Figura 90: Tabla Plan Distribución e Instalación.	117
Figura 91: Formulario Solicitar Aprobación de Cambios.....	117
Figura 92: Formulario Realizar Marcha Atrás.	118
Figura 93: Tabla Back Outs.....	118
Figura 94: Formulario Notificar Marcha Atrás realizada.....	118
Figura 95: Formulario Notificar BD actualizada.....	118
Figura 96: Resultado de los casos de pruebas.	119
Figura 97: Evaluación del conocimiento de ITIL.	121
Figura 98: Evaluación del conocimiento del Proceso de Gestión de Versiones y Liberaciones	121
Figura 99: Evaluación de la importancia de un SI de Gestión de Versiones y Liberaciones...	122
Figura 100: Evaluación de la facilidad de uso del sistema.	123
Figura 101: Evaluación de la navegación del sistema.	123
Figura 102: Evaluación de la distribución de elementos en el sistema.....	124
Figura 103: Evaluación del tiempo de respuesta del sistema.....	124

ÍNDICE DE TABLAS

Tabla 1: Diferencias en la organización de la documentación.	12
Tabla 2: Descripción de actores del Sistema de Gestión de Versiones y Liberaciones de Software.	60
Tabla 3: Especificar Caso de Uso Nivel 2 – Establecer Políticas de Liberación.....	67
Tabla 4: Especificar Caso de Uso Nivel 2 – Planificar Liberación.....	67
Tabla 5: Especificar Caso de Uso Nivel 2 – Diseñar Versión.	67
Tabla 6: Especificar Caso de Uso Nivel 2 – Construir Versión.	68
Tabla 7: Especificar Caso de Uso Nivel 2 – Realizar Pruebas.	68
Tabla 8: Especificar Caso de Uso Nivel 2 – Evaluar Aceptación de Versión.....	68
Tabla 9: Especificar Caso de Uso Nivel 2 – Planificar Implementación.....	69
Tabla 10: Especificar Caso de Uso Nivel 2 – Comunicar y Adiestrar Usuarios.	69
Tabla 11: Especificar Caso de Uso Nivel 2 – Distribuir e Instalar Versión.....	69
Tabla 12: Especificar Caso de Uso Nivel 2 – Archivar Cambios.....	69

Tabla 13: Actividades modeladas para Proceso de Gestión de Versiones y Liberaciones.	81
Tabla 14: Analogía entre los campos de la tabla RFC de PostgreSQL y los campos parameter	91
Tabla 15: Propiedades de los <i>text input</i> del formulario init RFC.	99
Tabla 16: Propiedades de los <i>combo box</i> del formulario init RFC.	100
Tabla 17: Propiedades de los <i>text area</i> del formulario init RFC	100
Tabla 18: Propiedades de los <i>labels</i> del formulario Planificar Liberación.	102
Tabla 19: Propiedades de los <i>text input</i> del formulario Planificar Liberación.	102
Tabla 20: Propiedades de los <i>text area</i> del formulario Planificar Liberación.	102
Tabla 21: Propiedades de los <i>combo box</i> del formulario Planificar Liberación.	102
Tabla 22: Propiedades de los <i>labels</i> del formulario NotificarFin2.	103

INTRODUCCIÓN

En la actualidad los Sistemas de Información (SI) y las Tecnologías de Información (TI) han cambiado la operación de las organizaciones, ya que permiten automatizar los procesos, suministrar información necesaria para la toma de decisiones y aumentar la competitividad de las empresas. Es por ello que se debe invertir en recursos para el soporte, entrega y gestión de servicios de TI.

El desarrollo de hardware y software siempre es progresivo, ambos cambian con el tiempo por diversas razones y son considerados como bienes importantes y estratégicos dentro de una organización, razón por la cual es necesario controlar la calidad de los mismos en el entorno de producción.

Tomando en cuenta las razones expuestas anteriormente, es importante contar con un SI que permita controlar la gestión de versiones de software en las empresas.

Tal es el caso de la empresa Compañía Anónima Nacional de Teléfonos de Venezuela (CANTV), que hoy en día realiza la gestión de versiones de software de forma manual, sin repositorios de hardware y software, sin seguimiento del proceso y sin responsabilidades claramente definidas para cada uno de los roles involucrados. Es por ello que surge como necesidad de la empresa contar con un sistema capaz de controlar este proceso.

Ante este requerimiento, se propone diseñar y construir un SI empleando: la Metodología de Gestión de los Procesos de Negocio Sustentada en el Uso de Patrones propuesta por Bonillo en el año 2008 y el Sistema de Gestión de Procesos de Negocio (BPMS) Intalio *Community Edition* 6.0.2 el cual es una tecnología de información libre basada en estándares internacionales.

Este Trabajo Especial de Grado tiene como finalidad construir un Sistema de Información para la Gestión de Versiones y Liberaciones de Software según el código de mejores prácticas propuestas por la Biblioteca de Infraestructura de Tecnologías de Información en su tercera versión para la empresa CANTV usando el BPMS Intalio *Community Edition* 6.0.2.

El presente documento se encuentra estructurado de la siguiente manera:

Capítulo I: en él se describe el problema de investigación, manifestaciones y evidencias necesarias para establecer el objetivo general, los objetivos específicos, la justificación y el alcance.

Capítulo II: establece las bases conceptuales que soportan la investigación. Se estudia detalladamente todo el procedimiento para llevar a cabo una correcta Gestión de Versiones y Liberaciones, los Sistemas de Información, la Gestión de Servicios de Tecnología de Información, la Biblioteca de Infraestructura de Tecnologías de la Información y los Procesos de Negocios.

Capítulo III: define el marco metodológico, describe la Metodología de Gestión de los Procesos de Negocios, indica el tipo de investigación, técnicas de revisión e investigación y se presentan los resultados obtenidos al aplicar la metodología seleccionada.

Capítulo IV: se presentan las conclusiones, limitaciones y recomendaciones de trabajos futuros.

Para finalizar, se exponen las referencias consultadas que dieron soporte al Trabajo Especial de Grado.

CAPÍTULO I: EL PROBLEMA

En este capítulo se describe la situación objeto de estudio, ubicándola en el contexto para formular el problema y de esta manera proceder a plantear, justificar y delimitar la solución y los objetivos propuestos en el Trabajo Especial de Grado (TEG).

1.1 Antecedentes de la Investigación

En las organizaciones ocurren cambios tanto en versiones de hardware como de software, los cuales deben ser gestionados correctamente. Muchas veces no se toma en cuenta el riesgo o el impacto que una nueva versión pueda causar en el entorno de producción, esto trae como consecuencia la ausencia de una clara asignación de responsabilidades durante todo el proceso de implementación del cambio, falta de entornos de pruebas adecuados para las versiones, incompatibilidades de versiones. (Chapin, 2001).

Los cambios surgen como resultados de problemas, pero muchos cambios se producen por la búsqueda anticipada de beneficios empresariales, tales como la reducción de costos o mejora de servicios. (Rodríguez et al., 2001).

Para cubrir esta deficiencia el proceso de Gestión de Versiones y Liberaciones de la Biblioteca de Infraestructura de Tecnologías de la Información (ITIL) permite hacer un seguimiento de las versiones desde su diseño hasta más allá de su implementación. Este plantea un conjunto de pasos a seguir para llevar a cabo los cambios en las organizaciones de manera satisfactoria. (Comunidad Osiatis, 2011).

Por medio del modelado de proceso se obtiene un mejor entendimiento del proceso y permite mejorar la calidad del mismo. (Club-BPM, 2009).

Actualmente existen múltiples Sistemas de Gestión de Procesos de Negocios (BPMS) para manejar todo lo relacionado con la Gestión de Procesos de Negocios, existen tanto propietarias como por ejemplo Oracle BPM así como también existen algunas de código abierto como: Intalio *Community*, Bonita, entre otras.

Cuando ocurre una incidencia, el proceso de Gestión de Incidencia intentará resolverla cuanto antes. Si durante la fase de investigación de la incidencia no se encuentra ninguna solución, el proceso de Gestión de Incidencia recurre a la Gestión de Problemas para determinar la causa de la incidencia. Entonces, la Gestión de Problemas asigna recursos para realizar un diagnóstico. Una vez que se identifica el error y tras averiguar cómo resolverlo emiten una Petición de Cambio (*Request for Change* o RFC) a la Gestión de Cambio. (Comunidad Osiatis, 2011).

Un RFC es una propuesta formal para que se realice un cambio, incluye detalles del cambio propuesto tales como las especificaciones funcionales y técnicas necesarias para diseñar y construir una nueva versión. (ITIL, 2006).

Por otra parte, la Gestión de Incidentes debe informarle a la Gestión de Versiones y Liberaciones si una nueva versión instalada provoca alguna interrupción de servicio, ya que muchas veces durante la actividad de desarrollo surgen errores conocidos pero no son resueltos. Si se detectan incidentes de alto impacto sobre la calidad de servicio, entonces se evaluará la posibilidad de activar los planes de *back-out* o marcha atrás definidos previamente por la Gestión de Versiones y Liberaciones. (Comunidad Osiatis, 2011).

Los antecedentes lo constituyen aquellos elementos teóricos que se refieren a trabajos previos relacionados con el problema planteado anteriormente, como se muestran a continuación:

En el mes de Mayo de 2006, Eduardo Sette y Ricardo Marques presentaron su Trabajo Especial de Grado titulado *Uma Proposta Ontológica para um Sistema de Gestão de Versionamento do Modelo ITIL*, el cual sirvió como referencia para los conceptos relacionados con el proceso de Gestión de Versiones. (Sette y Marques, 2006);

Posteriormente en el año 2008, Vladimir Chávez y Raúl Matheus 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*. Éste trabajo se enfocó en los procesos de Gestión de Incidentes y Gestión de Problemas. (Chávez y Matheus, 2008);

En ese mismo año 2008, el Doctor Pedro Bonillo presentó su Tesis Doctoral titulada *Metodología de Gestión de los Procesos de Negocio Sustentada en el uso de Patrones*. Ésta investigación tomó como caso de estudio el Proceso de Gestión de Versiones de ITIL. (Bonillo, 2008).

Así, con base a estos trabajos previos, nos proponemos construir un Sistema de Información para la Gestión de Versiones y Liberaciones de Software tomando como marco referencial ITIL v3 usando la Metodología de Gestión de los Procesos de Negocio Sustentada en el Uso de Patrones para la empresa de telecomunicaciones CANTV. El Anexo 1 describe la empresa CANTV.

1.2 Formulación del Problema

Con base en lo expuesto anteriormente, surgen las siguientes interrogantes:

- ¿Es factible aplicar el proceso de Gestión de Versiones y Liberaciones de ITIL v3 en las empresas del área de TI?
- ¿Mediante qué sistema de información es posible automatizar el proceso de Gestión de Versiones y Liberaciones de ITIL v3?
- ¿Cómo implementar el proceso de Gestión de Versiones y Liberaciones de ITIL v3 en las organizaciones usando sistemas de información bajo

tecnologías de información libre basándose en las mejores prácticas de TI?

1.3 Justificación

Actualmente en la empresa CANTV existe un proyecto que consta de 456 aplicaciones de software a desarrollar, de las cuales sólo se tienen el código fuente de 25 de ellas, no se tiene el código de las 431 aplicaciones restantes debido a problemas tales como el despido del personal encargado o su jubilación, por estas razones no es posible versionar. Este problema surge a raíz de la ausencia de un proceso de seguimiento o control sobre las versiones desarrolladas y a desarrollar. (Informe anual CANTV, 2009).

Por ahora CANTV sólo cumple con algunas de las actividades de supervisión del entorno de desarrollo, supervisión del entorno de pruebas, supervisión del entorno de producción y contemplar la marcha atrás, razón por la cual requiere de un Sistema de Información para la Gestión de Versiones y Liberaciones de Software según el marco referencial ITIL v3 que permita hacer seguimiento de cada una de las etapas del proceso y almacene los planes generados.

El Sistema de Información para la Gestión de Versiones y Liberaciones de Software a construir estará basado en estándares internacionales, lo cual le brinda calidad y facilita la transferencia tecnológica, además se utilizará el BPMS de código abierto Intalio *Community Edition* 6.0.2, de esta manera la empresa cumple con el Decreto Presidencial N°3390 del Ministerio de Ciencia y Tecnología de la República Bolivariana de Venezuela decretada el 23 de diciembre de 2004 el cual apoya la implantación de software libre.

1.4 Objetivos

La solución descrita anteriormente puede formularse a través de los siguientes objetivos:

1.4.1 Objetivo General

Desarrollar un Sistema de Información para la Gestión de Versiones y Liberaciones de Software basado en estándares internacionales, tecnologías de información libre y las mejores prácticas propuestas por la Biblioteca de Infraestructura de Tecnologías de Información v3 en los departamentos de la Gerencia de Tecnología de Información de CANTV.

1.4.2 Objetivos Específicos

- Diagnosticar el escenario actual del Proceso de Gestión de Versiones de Software de CANTV.
- Analizar el proceso de Gestión de Versiones y Liberaciones de ITIL v3
- Diseñar procesos que permitan visualizar el proceso de Gestión de Versiones y Liberaciones desde la perspectiva de los roles involucrados según estándares internacionales tales como la Notación para el Modelado de Procesos de Negocio (BPMN) y la Arquitectura Orientada a Servicios (SOA)
- Modelar el proceso de Gestión de Versiones y Liberaciones usando BPMN
- Implementar el proceso de Gestión de Versiones y Liberaciones de ITIL v3 utilizando el BPMS Intalio *Community* 6.0.2 privilegiando el uso de tecnologías de información libres cumpliendo con el Decreto Presidencial N°3390
- Configurar el motor de ejecución para el proceso de Gestión de Versiones y Liberaciones

- Realizar pruebas que permitan verificar el correcto funcionamiento de la solución en función de los requerimientos implementados utilizando Selenium IDE 1.7.2

1.5 Alcance

Se desarrollará un Sistema de Información para la Gestión de Versiones y Liberaciones de Software según ITIL v3 en la empresa de telecomunicaciones CANTV en el área de Gerencia de Programa de Soluciones TI.

Se automatizará el proceso con el BPMS Intalio *Community Edition* 6.0.2, ya que es de código abierto, está basado en estándares y tiene como modalidad de desarrollo el de cero código.

De la Metodología de Gestión de los Procesos de Negocio sustentada en el Uso de Patrones, se utilizará el subproceso Crear Proceso sin contemplar los patrones; el subproceso Administrar Proceso no será utilizado ya que este TEG no contempla el uso de la disciplina BAM (*Business Activity Monitoring*).

Este estudio está enmarcado al área de gerencia de procesos de negocio y los sistemas de información, dentro del dominio tecnológico existente en las ciencias computacionales para el año 2011: Software Libre, Código Abierto, Ambientes de Soporte de Decisiones y Modelado de Sistemas de Información.

CAPÍTULO II: MARCO TEÓRICO

En éste capítulo se presentan las bases teóricas que sustentan la investigación.

2.1 Gestión de Versiones y Liberaciones bajo la Biblioteca de Infraestructura de Tecnologías de la Información

2.1.1 Gestión de Servicios de Tecnologías de la Información

Los proveedores de los servicios de Tecnologías de Información (TI) no pueden seguir manteniendo su enfoque en la tecnología y sus propias organizaciones, es por ello que ahora tienen que considerar la calidad de los servicios que proveen y enfocarse en sus relaciones con los clientes. (Haren, 2002).

Las Tecnologías de la Información y las Comunicaciones (TIC) son tan antiguas como la historia y han jugado un importante papel en la misma. Sin embargo, no ha sido hasta tiempos recientes que mediante la automatización de su gestión se han convertido en una herramienta imprescindible y clave para empresas e instituciones. (Lobos et al., 2008).

La Gestión de Servicio TI (*Information Technology Service Management*, ITSM por sus siglas en inglés) de alta calidad se define como una disciplina basada en procesos, enfocada en alinear los servicios de TI proporcionados con las necesidades de las empresas, haciendo énfasis en los beneficios que percibe el cliente final. La ITSM propone cambiar el paradigma de gestión de TI, por una colección de componentes enfocados en los servicios usando distintos marcos de trabajo con las mejores prácticas. (Van Bon, 2002).

La información es considerada como la fuente principal de negocio, el negocio genera grandes cantidades de información. Es por ello que se requiere de una gestión estratégica que no debe ser vista como una herramienta más entre muchas otras.

Los procesos de datos e información que resultan de las operaciones y procesos de negocio, requieren de la aplicación de técnicas y medidas de control en el marco de un sistema de gestión que pueda garantizar la prestación de los servicios así como también la reducción de la vulnerabilidad a amenazas generadoras de riesgo que puedan poner en peligro la estabilidad del sistema operacional, organizacional y del sistema macro del negocio. Esto justifica la finalidad de la Gestión de Servicio TI. (Comunidad Osiatis, 2011).

Los objetivos de la gestión de servicios de TI son (Comunidad Osiatis, 2011):

- 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

Actualmente las inversiones en la infraestructura de TI y en los activos de información de las organizaciones son cada vez más importantes, lo cual justifica la implantación de sistemas que permiten asegurar el rendimiento de los procesos basados en servicios de TI y así lograr la reducción del costo total de propiedad y un retorno de inversión razonable.

El nuevo paradigma basado en “servicio” debe crear un acercamiento a cualquier organización independientemente del tamaño de la misma, las empresas deben a la vez adoptar y adaptar las mejores prácticas enfocándose en la calidad de servicio y la oportunidad del cambio en el negocio aplicando estándares actualizados.

La tendencia de ITSM se basa en la promoción y soporte de la aplicación de las mejores prácticas, marcos referenciales y estándares internacionales, entre los cuales se puede mencionar: *Information Technology Infrastructure Library* (ITIL), *Control Objectives for Information and related Technology* (COBIT), *IT*

Service Capability Maturity Model (ITSCMM), Application Services Library (ASL), Business Information Services Library (BISL) y otros.

Cabe destacar que el marco referencial a utilizar en este Trabajo Especial de Grado es la Biblioteca de Infraestructura de Tecnologías de la Información (ITIL) ya que proporciona un conjunto comprensivo, consistente y coherente de “mejores prácticas” óptimas para la gestión de los procesos, promocionando un enfoque de calidad para alcanzar efectividad y eficacia en el uso de los sistemas informáticos.

2.1.2 Biblioteca de Infraestructura de Tecnologías de la Información

ITIL es un conjunto de buenas prácticas destinadas a mejorar la gestión y provisión de servicios TI. Su objetivo es mejorar la calidad de los servicios TI ofrecidos, evitar los problemas asociados a los mismos y en caso de que estos ocurran ofrecer un marco de actuación para que estos sean solucionados con el menor impacto y a la mayor brevedad posible. (Comunidad Osiatis, 2011).

Aunque ITIL se desarrolló durante los años 1980, fue ampliamente adoptada a mediados de los años 1990. ITIL se construye en torno a una vista basada en proceso-modelo del control y gestión de las operaciones a menudo atribuida a W. Edwards Deming.

Las recomendaciones de ITIL fueron desarrolladas en los años 1980 por la *Central Computer and Telecommunications Agency (CCTA)* del gobierno británico como respuesta a la creciente dependencia de las tecnologías de la información y al reconocimiento de que con la ausencia de prácticas estándar, los contratos de las agencias estatales y del sector privado iban a crear sus propias prácticas de gestión de TI y a duplicar esfuerzos dentro de sus proyectos de Tecnologías de la Información y la Comunicación, lo que daba como resultado errores comunes y mayores costos.

ITIL fue publicado como un conjunto de 10 libros centrales, los cuales cubrían las dos principales áreas de Soporte del Servicio y Prestación del Servicio. Los libros centrales fueron soportados por 30 libros complementarios, cubriendo de esta manera una gran variedad de temas.

En el año 2000 se realizó una revisión a la Biblioteca y surge ITIL v2, logrando su reestructuración para facilitar el acceso a la información necesaria para administrar sus servicios. Luego los libros centrales se agruparon en dos para cubrir las áreas de Soporte del Servicio y Prestación del Servicio. (Comunidad Osiatis, 2011).

A mediados del año 2007, se realizó otra modificación a los libros y surge ITIL v3, el cual consta de 5 libros que representan una estructura articulada en base al concepto del ciclo de vida del servicio de TI.

Cabe destacar que en esta investigación se utilizará como marco de referencia ITIL v3.

En la Tabla 1 se muestra la comparación de la documentación de la segunda y tercera versión de ITIL.

Documentación de ITIL v2 Organizada según los sectores de entrega de servicios de TI	Documentación de ITIL v3 Organizada en función del ciclo de vida del servicio
<ol style="list-style-type: none"> 1. Soporte de Servicio 2. Entrega de Servicio 3. Planes para implementar la gestión del servicio 4. Gestión de la infraestructura y comunicaciones de TI (ICT) 5. Gestión de las aplicaciones 6. Perspectiva de negocio 7. Seguridad 	<ol style="list-style-type: none"> 1. Estrategia del Servicio 2. Diseño del Servicio 3. Transición del Servicio 4. Operación del Servicio 5. Mejora Continua del Servicio

Tabla 1: Diferencias en la organización de la documentación.

Fuente: Taylor y Turbitt (2008)

La Transición del Servicio se encarga de hacer que los productos y servicios definidos en la fase de Diseño del Servicio se integren en el entorno de producción y sean accesibles a los clientes y usuarios autorizados. (Comunidad Osiatis, 2011).

Sus principales objetivos se resumen en (Comunidad Osiatis, 2011):

- Supervisar y dar soporte a todo el proceso de cambio del servicio (nuevo o modificado)
- Garantizar que los nuevos servicios cumplen los requisitos y estándares de calidad estipulados en las fases de Estrategia y Diseño
- Minimizar los riesgos asociados al cambio reduciendo el posible impacto sobre los servicios existentes
- Mejorar la satisfacción del cliente respecto a los servicios prestados
- Comunicar el cambio a todos los agentes implicados

En la Figura 1 se observan los procesos que abarca la Transición del Servicio.

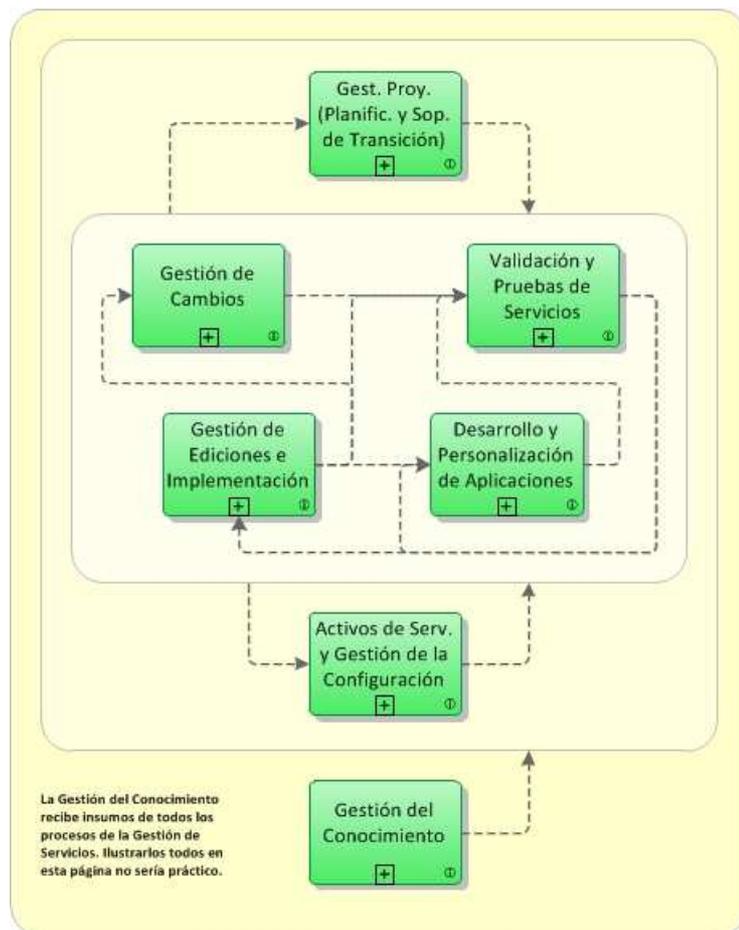


Figura 1: Procesos de Transición del Servicio ITIL v3.

Fuente: it process maps (2011)

Las principales funciones y procesos asociados directamente a la Transición del Servicio son (Comunidad Osiatis, 2011):

- Planificación y Soporte a la Transición: responsable de planificar y coordinar todo el proceso de transición asociado a la creación o modificación de los servicios TI.
- Gestión de Cambios: responsable de supervisar y aprobar la introducción o modificación de los servicios prestados garantizando que todo el proceso haya sido convenientemente planificado, evaluado, probado, implementado y documentado.

- Gestión de la Configuración y Activos del Servicio: responsable del registro y gestión de los elementos de configuración (*Configuration Items*, CIs por sus siglas en inglés) y activos del servicio.
- Validación y Pruebas: responsable de garantizar que los servicios cumplen los requisitos preestablecidos antes de su paso al entorno de producción.
- Evaluación: responsable de evaluar la calidad general de los servicios, su rentabilidad, su utilización, la percepción de sus usuarios y otros.
- Gestión del Conocimiento: gestiona toda la información relevante a la prestación de los servicios asegurando que esté disponible para los agentes implicados en su concepción, diseño, desarrollo, implementación y operación.

Otro proceso de Transición del Servicio propuesto por ITIL v3 es el proceso de Gestión de Versiones y Liberaciones, el cual será objeto de estudio en este Trabajo Especial de Grado.

2.1.3 Versión

Una versión, revisión o edición de un producto se define como el estado en el que se encuentra en un momento dado en su desarrollo o modificación. (Collins-Sussman et al., 2004).

Una versión consta de un grupo de ítems de configuración (CIs) de nueva creación o modificados que han sido validados para su instalación en el entorno de producción. Las especificaciones funcionales y técnicas de la versión se indican en la Petición de Cambio (*Request For Change*, RFC por sus siglas en inglés). (Comunidad Osiatis, 2009).

Una versión está compuesta de componentes, los cuales se definen como partes de un programa en donde se codifica una función o sub-función específica. (Moreno, 2001).

Durante la fase de desarrollo de versiones de software, éstas pasan por una serie de etapas en las cuales se les va agregando funcionalidades progresivamente:

- Pre-Alfa: contiene todas las actividades realizadas durante el proyecto de software antes de la prueba. Estas actividades pueden incluir el análisis de requerimientos, diseño de software, desarrollo de software y pruebas unitarias.
- Alfa: es la primera versión del programa, es utilizada solo por el personal de verificación para realizar pruebas. Algunas veces este término es utilizado para referirse a una fase donde el producto es inestable.
- Beta: es la segunda etapa de desarrollo y representa la primera versión completa del programa o de un producto, que tiende a ser inestable pero será utilizado para realizar demostraciones internas. Esta versión aún no se considera totalmente depurada, puede ser suministrada a usuarios interesados para que realicen pruebas con la finalidad de obtener su *feedback*. (Moreno, 2001).
- Versión Candidata a Definitiva (*Release Candidate* o RC): es un producto final considerado como versión definitiva, es relativamente estable. (Moreno, 2001).

El término RC es utilizado por muchas empresas. A las versiones prácticamente completas pero que aún se encuentran en la fase de pruebas se les denomina versiones gamma, delta y otros; para las versiones prácticamente libre de errores que están en el proceso final de pruebas se les conoce como omega.

- Versión para Manufactura o Comercialización (*Release to Manufacturing* o RTM): también conocida como versión dorada, es una versión muy estable y relativamente libre de errores, resultando ideal para el uso por parte de los usuarios finales razón por la cual se está produciendo masivamente y estará a la venta próximamente. A esta versión se le puede realizar correcciones de último momento.
- Disponibilidad General (*General Availability* o GA): en esta etapa todas las actividades de comercialización han sido completadas y el software

ha sido puesto a disposición del mercado general, ya sea a través de la web o medios físicos. El tiempo entre RTM y GA puede variar desde una semana hasta meses.

La identificación o numeración de software se define como el proceso en el que se asignan nombres o secuencias de números a la versión. La numeración se realiza de manera creciente.

En algunos sistemas, los identificadores de secuencia se usan para transmitir la importancia de los cambios entre versiones. Existen muchas estructuras para numerar las versiones, pero las más comunes son:

- *Mayor.Menor[.Build[.Revisión]]*

El número correspondiente a *Mayor*, se incrementa cuando se producen cambios importantes en la funcionalidad, el número *Menor* se incrementa cuando se añaden cambios menores o correcciones significativas, y el número de *Revisión* se incrementa cuando se incorporan correcciones de errores pequeños. (Mako, 2000).

De acuerdo a las estructuras de numeración, el tercer campo (*Build* o Mantenimiento) puede tomar los siguientes valores:

- 0 para alfa
- 1 para beta
- 2 para RC
- 3 para el público

Por ejemplo:

- 1.2.0.1 versión alfa
- 1.2.1.2 beta con algunas correcciones de errores
- 1.2.2.3 RC
- 1.2.3.0 distribución comercial
- 1.2.3.5 distribución comercial con muchas correcciones de errores

- Método X.Y.Z

Este método se basa en dos o tres cifras decimales, dependiendo de la importancia de los cambios, se modificará el valor de X, Y ó Z. La primera cifra siempre se cambia cuando se hace una modificación crítica o muy importante, siendo la segunda cifra de menor importancia.

Se debe iniciar desde **0.Y.Z**, lo cual indica que la versión aún no está lista o que no cumple con los requerimientos mínimos. Cada cambio en esta cifra denota una reescritura o la incompatibilidad con versiones anteriores.

La segunda cifra **X.0.Z** se cambia cuando hay modificaciones en el contenido o la funcionalidad, pero no lo suficientemente importantes como para decir que ya no es la misma versión. Cuando se hace un cambio mayor (en la primera cifra), el segundo número se reinicia a **0**.

La tercera cifra se cambia cuando se hacen correcciones pero no se ha añadido ni eliminado nada relevante. Si se hace un cambio en la segunda cifra se debe reiniciar el número de la tercera a **0**. (Cristalab, 2010).

De acuerdo al impacto en la infraestructura de TI, las versiones se clasifican en (Comunidad Osiatis, 2011):

- Versiones Mayores: contienen despliegues importantes de software y hardware con nuevas capacidades o funciones. Las versiones mayores acumulan todos los cambios de las versiones menores. Las versiones mayores se identifican como: 1.0, 2.0 y otros.
- Versiones Menores: incorporan correcciones de varios errores conocidos y que usualmente son modificaciones que implementan de manera documentada soluciones de emergencia. Para identificar estas versiones se incrementa el primer decimal, por ejemplo: 1.1, 1.2 y otros.
- Versiones de Emergencia: contienen soluciones rápidas para reparar un error conocido o medidas temporales para prevenir la interrupción de

servicios críticos. Las versiones de emergencia se identifican incrementando el segundo decimal, por ejemplo: 1.1.1, 1.1.2 y otros.

En la Figura 2 se puede observar un ejemplo de la numeración de versiones.

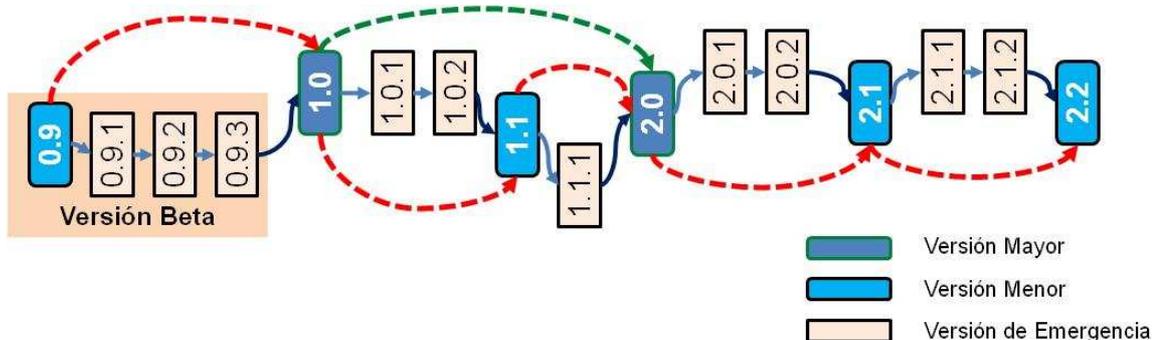


Figura 2: Numeración de versiones.

Fuente: Wikipedia (2010)

Los despliegues de nuevas versiones se realizan de distintas maneras y es responsabilidad del Gestor de Versiones decidir cuál será la forma más conveniente. El tipo de despliegue de versión puede ser (Comunidad Osatis, 2011):

- Versión Delta: solo se prueban e instalan los CIs de hardware o software que han sido modificados. Debido a su simplicidad, no es posible probar cómo los cambios afectarán al resto del entorno ya que solo se prueban los elementos modificados.
- Versión Completa: incluye todos los CIs de hardware o software, incluso aquellos que no han sido modificados. La versión completa es cuidadosamente probada por lo que es menos probable que se generen incidentes durante la ejecución.
- Paquete de Versiones: se opta por distribuir de forma sincronizada diferentes paquetes de versiones, de esta forma ofrece mayor estabilidad al entorno de TI debido a la minimización de las interrupciones en el entorno. En algunos casos la opción se ve obligada por incompatibilidades entre una nueva versión con software o hardware previamente instalado.

2.1.4 Gestión de Versiones y Liberaciones de ITIL v3

Es un proceso de planificación, construcción, pruebas y despliegue de hardware y software, control de versiones y software de almacenamiento.

Su propósito es asegurar que se lleve a cabo un método consistente de despliegue del sistema, entendiéndose por despliegue como todas aquellas actividades que deben llevarse a cabo para que el sistema esté disponible para su uso. Esto reduce la probabilidad de que ocurran incidentes como resultado de una acumulación de actividades y asegura que solo sean probadas y aceptadas versiones de hardware y software que hayan sido instaladas en algún momento. (Comunidad Osiatis, 2011).

La Gestión de Versiones y Liberaciones debe colaborar con la Gestión de Cambios y de Configuración y Activos del Servicio para asegurar que toda la información relativa a las nuevas versiones se refleje en la Base de Datos de Gestión de Configuración (*Configuration Management Data Base*, CMDB por sus siglas en inglés) de manera tal que ésta se encuentre correctamente actualizada para ofrecer la imagen real de la configuración de la infraestructura de TI. También debe mantener actualizada la Biblioteca de Medios Definitivos (*Definitive Media Library*, DML por sus siglas en inglés) en la cual se encuentran las copias de todo el software en producción, y el Depósito de Recambios Definitivos (*Definitive Spares*, DS por sus siglas en inglés) donde se almacenan las piezas de repuesto y documentación para la reparación de problemas de hardware en el entorno de producción. (Comunidad Osiatis, 2011).

Los objetivos de la Gestión de Versiones y Liberaciones son (Comunidad Osiatis, 2011):

- Establecer una política de implementación de nuevas versiones de hardware y software

- Implementar las versiones de hardware y software en el entorno de producción luego de su verificación
- Garantizar que el proceso de cambio cumpla con las especificaciones de la RFC
- Asegurar, en colaboración con la Gestión de Cambios y Configuraciones, que todos los cambios estén correctamente reflejados en la CMDB
- Archivar copias idénticas del software en producción, así como también toda su documentación asociada en la DML
- Mantener actualizado el DS

Entre los beneficios de la Gestión de Versiones y Liberaciones de ITIL v3 cabe destacar (Comunidad Osiatis, 2011):

- El proceso de cambio se realiza sin deteriorar la calidad de servicio
- Las nuevas versiones cumplen los objetivos propuestos
- El correcto mantenimiento de la DML impide que se pierdan (valiosas) copias de los archivos fuentes
- Se reduce el número de copias de software ilegales
- Control centralizado del software y hardware desplegado
- Protección contra virus y problemas asociados a versiones de software incontroladas

2.1.4.1 Proceso de Gestión de Versiones y Liberaciones de ITIL v3

A continuación se explicará detalladamente los pasos del Proceso de Gestión de Versiones y Liberaciones de ITIL v3 (Comunidad Osiatis, 2011):

Política de Liberación

Esta fase tiene la finalidad de aclarar funciones y responsabilidades para la Gestión de Versiones y Liberaciones.

La política de liberación realiza las siguientes actividades:

- Numeración de versiones
- Definición de versiones mayores, menores y de emergencia
- Evaluar la frecuencia de las liberaciones
- Resultados previstos para cada tipo liberación
- Grado de pruebas para los planes de marcha atrás
- Descripción de cómo se lleva a cabo el control del proceso
- La documentación de la configuración exacta de la DML y la definición de los criterios de aceptación para nuevas funcionalidades de software

Plan de Liberación

El plan de liberación debe definir:

- Los procedimientos de liberación
- Funciones y responsabilidades de todo el personal
- Herramientas para apoyar el lanzamiento de hardware y software en el entorno real
- Plantillas para la elaboración de documentos que apoyen al proceso de planificación
- Entornos de prueba
- Calendario de liberaciones
- Planes de marcha atrás

Diseño y Obtención de la Versión

La Gestión de Versiones y Liberaciones se encarga de diseñar y construir nuevas versiones en base a las pautas establecidas en las RFCs correspondientes.

A veces el desarrollo es realizado por la misma empresa y en muchos casos se requiere de la participación de proveedores externos. Dado el caso de que se recurra a los proveedores externos, la Gestión de Versiones y Liberaciones

debe asegurar que el paquete o los paquetes de software o hardware ofrecidos cumplan con las especificaciones detalladas en la RFC. La Gestión de Versiones y Liberaciones es responsable de todo el proceso de configuración.

El desarrollo debe incluir los scripts de instalación necesarios para el despliegue de la versión. Los scripts deben contener la siguiente información:

- Respaldo automático de datos
- Actualizaciones necesarias de las BD asociadas
- Instalación de las nuevas versiones en diferentes sistemas o sitios geográficos
- Creación de *logs* durante el proceso de instalación

Un aspecto importante a tomar en cuenta durante el desarrollo son los planes de marcha atrás dependiendo de lo que se haya acordado con los clientes en los Acuerdos de Nivel de Servicio (*Service Level Agreement*, SLA por sus siglas en inglés).

Pruebas

Un protocolo de pruebas bien planificado es indispensable para lanzar al entorno de producción una nueva versión con razonables garantías de éxito.

Las pruebas deben abarcar la validación de carácter técnico y pruebas funcionales con usuarios para asegurar que la versión cumple con los requisitos establecidos y es usable, debe tomarse en cuenta que existe una resistencia al cambio por parte de los usuarios.

Cabe destacar que las pruebas deben incluir planes de marcha atrás para asegurar que se podrá volver a la última versión estable de manera rápida, ordenada y sin perder información valiosa.

Entre las actividades que se deben tomar en cuenta para el proceso de prueba se tienen:

- Pruebas del correcto funcionamiento de la versión en un entorno real
- Pruebas de los procedimientos automáticos o manuales de instalación
- Lista de *bugs* o errores detectados
- Pruebas de los planes de marcha atrás
- Documentación para usuarios y personal de servicio

Una vez que se realizan las pruebas, la Gestión de Cambios dará la validación final a la versión para proceder a su instalación. Si la versión no es aceptada entonces será devuelta a la Gestión de Versiones y Liberaciones para su reevaluación.

Aceptación de Liberación

Las pruebas de la versión deben ser realizadas por personal de negocio e involucrar al personal de TI. Todos los procedimientos de marcha atrás deben ser probados como parte de esta actividad. La versión debe ser instalada y ejecutada como se tiene previsto.

Durante esta etapa se realizan pruebas tanto en los procedimientos de instalación como del funcionamiento del sistema final.

Las pruebas deben cubrir los procedimientos de instalación y la integridad funcional del sistema resultante. La aceptación final para que la versión pueda pasar al entorno de producción depende del proceso de Gestión de Cambio.

Si una versión es rechazada, entonces se realiza una reprogramación a través de Gestión de Cambio. Las versiones rechazadas deben ser rastreadas y reportadas a través de la Gestión del Cambio como cambios fallidos.

Plan de Implementación

El plan añade detalles del proceso de instalación y de implementación.

Las actividades de la planificación de la implementación son:

- Elaboración de un calendario de eventos
- Listar los CIs que deben ser instalados y desinstalados
- Documentar planes de acción por sitios
- Planificar el proceso de comunicación
- Desarrollar planes de compra de hardware y software, ya que muchas veces la liberación de una nueva versión implica la adquisición y el despliegue de activos de alto valor
- Programar reuniones para controlar el equipo de trabajo

Comunicación y Adiestramiento

La comunicación es un factor muy importante ya que los usuarios deben estar conscientes de los cambios que se tienen planeados y cómo les puede afectar. El adiestramiento debe tomarse en cuenta antes de un lanzamiento de hardware o software.

A pesar de lo importante que es la comunicación y adiestramiento, frecuentemente cuando se trata aspectos de carácter técnico, se obvia el factor humano. Es necesaria la interacción humano-aplicación; es inútil disponer de un sofisticado servicio de TI si los usuarios no se encuentran capacitados e informados para sacarle provecho al servicio.

Tomando en cuenta dicha importancia, la (in)formación debe estar estructurada en niveles:

- Los usuarios deben estar al tanto del próximo lanzamiento de una nueva versión y conocer sus funcionalidades

- Las pruebas funcionales deben ser realizadas por un selecto grupo de usuarios finales. En este proceso de prueba se documentará y analizará:
 - La experiencia del usuario
 - Los comentarios, sugerencias y dudas que hayan surgido durante la interacción del usuario con la nueva versión
 - La claridad de la documentación que estará a disposición del usuario final
- De ser necesario se impartirán cursos presenciales o remotos sobre el funcionamiento de la nueva versión
- Se desarrollará una página de Preguntas Frecuentes (*Frequently Asked Questions*, FAQs por sus siglas en inglés) para que los usuarios puedan aclarar sus dudas, solicitar ayuda o soporte técnico acerca del uso de la nueva versión

Distribución e Instalación

La distribución de las versiones de software desde el entorno de desarrollo al entorno de prueba y luego al entorno real puede estar asociada a algún cambio de hardware u otro cambio.

La distribución de software debe estar diseñada de manera que la integridad del software se mantenga durante la manipulación, el empaquetado y la entrega. La distribución automatizada de software a lugares remotos ahorra recursos y reduce el tiempo de distribución. Si el software va a ser distribuido a través de la red, es esencial comprobar que llegue la versión completa al destino.

La distribución de la nueva versión debe ser documentada para que todas las partes estén al tanto de las tareas y responsabilidades específicas. Los usuarios finales deben tener el calendario de lanzamiento y estar informados de cómo el lanzamiento puede afectar las actividades diarias.

La distribución puede realizarse de la siguiente manera:

- Completa: el servicio nuevo o modificado se implementa en todos los sitios en una sola operación.
- En fases: el servicio inicialmente se implementa en una parte de los usuarios, y luego esta operación se repite para los usuarios restantes a través de un plan de despliegue previsto.
- Empujar: se utiliza cuando el componente de servicio se implementa desde el centro y se empuja a los lugares de destino. En cuanto a la implementación del servicio, la entrega de componentes de servicios actualizados, ya sea completa o en fases, utiliza el enfoque empujar ya que el servicio nuevo o modificado debe ser suministrado a los usuarios.
- Halar: se utiliza para tomar versiones de software desde una ubicación central, pero los usuarios son libres para descargar o usar el software en el momento que prefieran o cuando se reinicia una estación de trabajo del usuario.
- Automatización: permite asegurar la consistencia. El tiempo necesario para proporcionar un mecanismo bien diseñado y automatizado de manera eficiente no siempre es viable.
- Manual: en este enfoque es importante controlar y medir el impacto de muchas actividades manuales repetidas, ya que es probable que sean ineficientes y propensas a errores. Muchas actividades manuales hacen que el equipo de liberación se atrase y surjan problemas de recursos o de capacidad que afecten a los niveles de servicio.

Durante la implementación hay que determinar claramente:

- Los CIs que deben ser borrados e instalados y el orden en que debe ser realizado este proceso
- Cuándo debe realizarse este proceso para diferentes grupos de trabajo y/o localizaciones geográficas
- Las métricas para la puesta en marcha de los planes de marcha atrás

Luego de la distribución, la Gestión de Versiones y Liberaciones debe asegurar que:

- Se haga una copia de la versión en la DML
- El DS incorpore los repuestos funcionales de los nuevos CIs
- La CMDB haya sido correctamente actualizada
- Los usuarios estén informados acerca de las funcionalidades de la nueva versión

Se debe realizar una planificación previa a la instalación tomando en cuenta los controles ambientales y eléctricos antes de realizar la conexión a la red.

Luego de la implementación el *Service Desk* le informa a la Gestión de Versiones los comentarios, incidentes, u otra situación que la nueva versión haya causado, con la finalidad de minimizar el impacto negativo para futuros cambios.

La Figura 3 muestra los pasos del Proceso de Gestión de Versiones y Liberaciones y los ambientes donde se desarrollan.

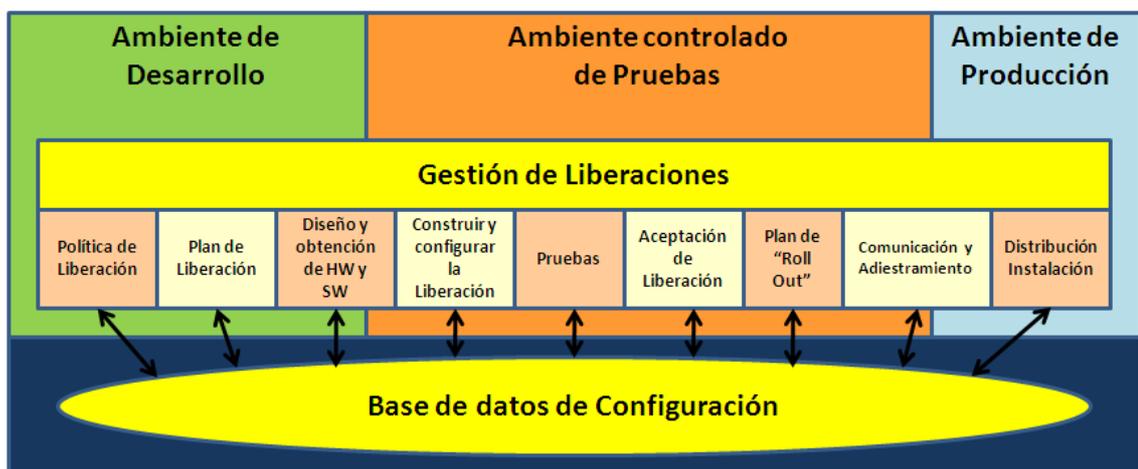


Figura 3: Pasos del Proceso de Gestión de Versiones y Liberaciones ITIL v3.

Fuente: ITIL Service Management (2007)

2.2 Sistemas de Información para la Gestión de Procesos de Negocio

2.2.1 Sistemas de Información

Se definen como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control de una organización. Además de apoyar la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores a analizar problemas, a visualizar asuntos complejos y a crear productos nuevos. (Laudon y Laudon, 2006).

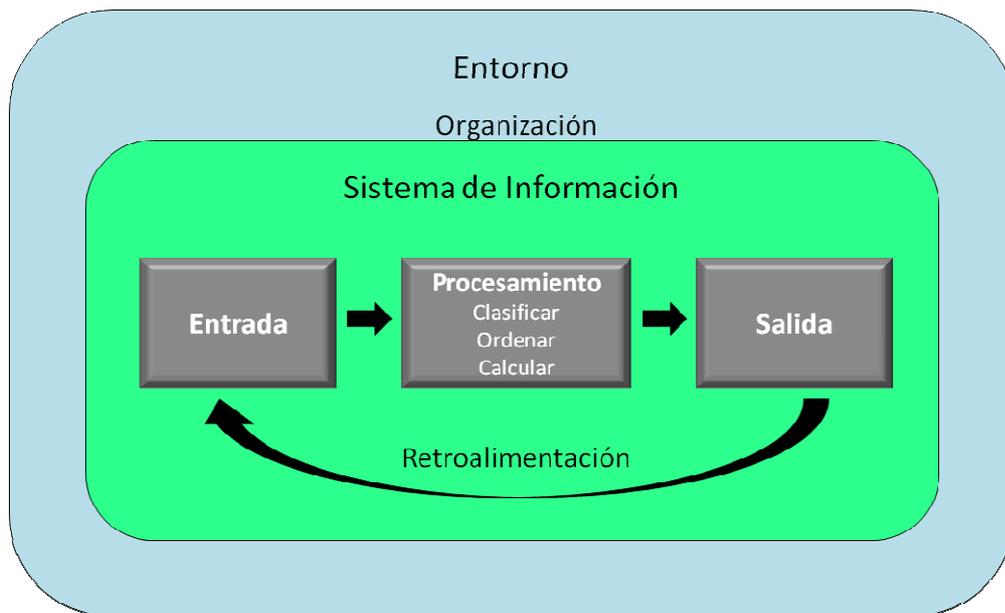


Figura 4: Modelo general de los Sistemas de Información.

Fuente: Laudon y Laudon (2006)

Tomando en cuenta la función destinada o el tipo de usuario final, los sistemas de información se clasifican de la siguiente manera (Laudon y Laudon, 2006):

- Sistemas de Procesamiento de Transacciones (*Transacción Processing System* o TPS): gestionan la información referente a las transacciones producidas en una empresa u organización.

- Sistemas de Información Gerencial (Management Information System o MIS): están orientados a solucionar problemas empresariales en general.
- Sistemas de Soporte a Decisiones (Decision Support System o DSS): son utilizados para realizar el análisis de las diferentes variables de negocio con la finalidad de apoyar el proceso de toma de decisiones.
- Sistemas de Información Ejecutiva (Executive Information System o EIS): orientados a usuarios de nivel gerencial, permiten monitorear el estado de las variables de un área o unidad de la empresa a partir de información interna y externa.

La Figura 5 muestra los tipos de sistemas de información.

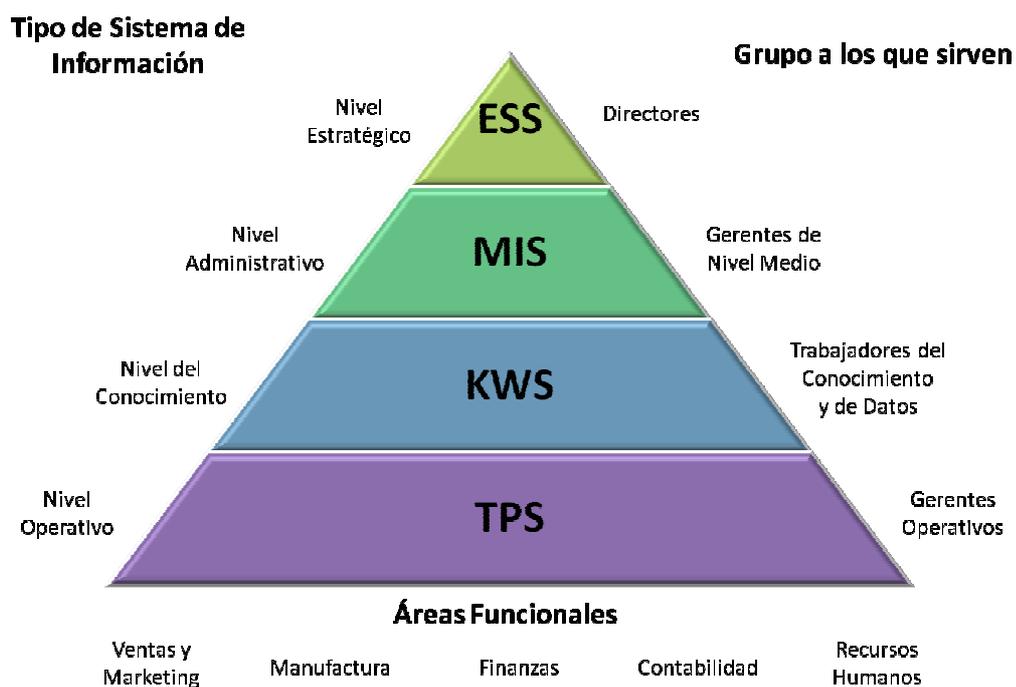


Figura 5: Tipos de Sistemas de Información.

Fuente: Laudon y Laudon (2006)

2.2.2 Proceso de Negocio

Un proceso de negocio describe las actividades de una determinada empresa u organización cuyo objetivo es satisfacer las necesidades de sus clientes.

El proceso de negocio se define como el conjunto de tareas lógicamente relacionadas que existen para conseguir un resultado bien definido dentro de un negocio; por lo tanto, toman una entrada y le agregan valor para producir una salida. Los procesos tienen entonces clientes que pueden ser internos o externos, los cuales reciben la salida, lo que puede ser un producto físico o un servicio. Estos establecen las condiciones de satisfacción o declaran que el producto o servicio es aceptable o no. (Barros, 1994).

Se han propuesto numerosas técnicas para el modelado de procesos como mecanismo para comprender la forma de trabajo en una organización, lo cual también permite encontrar los problemas que impiden a la empresa alcanzar sus objetivos.

Los procesos pueden ser medidos y están orientados al rendimiento, tienen resultados específicos, entregan resultados a clientes y responden a alguna acción.

Los procesos de negocio se clasifican en tres (Barros, 1994):

1. Procesos estratégicos: dan orientación al negocio.
2. Procesos centrales: dan el valor al cliente, ya que ellos son la parte principal del negocio.
3. Procesos de soporte: dan soporte a los procesos centrales.

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 un conjunto de prácticas conocidas como administración de procesos de negocio (BPM). (Mejía y Arzate, 2006).

2.2.3 Gestión de Procesos de Negocio

BPM (*Business Process Management*) se define como un enfoque gerencial y operativo que se basa en la coordinación de las actividades y decisiones que comprenden los procesos de negocio de una organización. (Felipe, 2009).

BPM es un conjunto de prácticas que tiene como objetivo mejorar la eficiencia por medio de la gestión sistemática de los procesos de negocio, los cuales se deben modelar, automatizar, integrar, monitorear y optimizar continuamente. El modelado de procesos permite obtener un mejor entendimiento del negocio así como también realizar una mejora en él; la automatización reduce errores; la administración de procesos permite verificar que éstos se ejecuten correctamente y la generación de información permite la mejora de los procesos.

BPM es una colaboración entre personas de negocio y tecnólogos para fomentar procesos de negocio efectivos, ágiles y transparentes. (Garimella et al., 2008).

La aplicación de BPM aporta beneficios para las empresas, aquellas empresas que adoptaron este concepto, han reportado beneficios que van desde la mejora en las capacidades de dirección, pasando por la reducción de obstáculos al momento de reaccionar ante cambios del mercado, hasta adquirir mayor capacidad de análisis sobre el desempeño de la empresa. Con BPM (Garimella et al., 2008):

- Los directores de negocio pueden, de forma más directa, medir, controlar y responder a todos los aspectos y elementos de sus procesos operacionales
- Los directores de TI pueden aplicar sus habilidades y recursos de forma más directa en las operaciones de negocio
- La dirección y los empleados de la organización pueden alinear mejor sus esfuerzos y mejorar la productividad y el rendimiento personal

- La empresa, como un todo, puede responder de forma más rápida a cambios y desafíos a la hora de cumplir sus fines y objetivos

En la Figura 6 se pueden apreciar las actividades que permite el paradigma BPM.



Figura 6: Actividades del paradigma BPM.

Fuente: Gartner (2009)

Ciclo de vida de BPM

El ciclo de vida BPM consta de cinco fases (Smart et al., 2008):

- Diseño: abarca tanto la identificación de los procesos existentes como los procesos nuevos. Se enfoca en la representación del flujo del proceso, actores, alertas, notificaciones, escalamientos, procedimientos operativos estandarizados, SLA y otros.

Un buen diseño reduce el número de problemas durante la vida útil del proceso. Este paso tiene como objetivo asegurar que se prepare un diseño teórico correcto y eficiente.

- Modelado: toma el diseño teórico e introduce combinaciones de variables.

El modelado consiste en (Silver, 2006):

- Capturar el flujo de procesos existentes mediante una notación diagramática estructurada
 - Diagramar procesos nuevos o modificados usando la misma notación. Se asocian los recursos, tiempos, costos, entradas y salidas de cada proceso
 - Simular el rendimiento de los procesos basándose en varios escenarios variando los parámetros
 - Analizar el rendimiento del procesamiento y costos de los escenarios
 - Documentar el proceso bajo algún formato exportable
- Ejecución: para automatizar los procesos se debe adquirir alguna herramienta para aplicar los pasos del proceso.
 - Monitoreo: implica el seguimiento de los procesos individuales, de modo que la información sobre su estado pueda ser visto fácilmente y aporte las estadísticas sobre el desempeño de uno o más procesos.

El grado de supervisión depende de la información que la empresa quiera evaluar y analizar, de qué modo quiere y debe ser controlado (en tiempo real, casi en tiempo real o ad-hoc).

- Optimización: incluye la recuperación de información del rendimiento de los procesos en la fase de modelización o de monitoreo, identificando los cuellos de botella y las oportunidades de ahorro de costes u otras mejoras, y luego, aplica dichas mejoras en el diseño del proceso.

La Figura 7 muestra el ciclo de vida de BPM.



Figura 7: Ciclo de vida BPM.

Fuente: oportunis consulting (2010)

Pasos de la metodología BPM

Los pasos o etapas de la metodología BPM son (Quesada, 2006):

- Organización: es quizá el paso más crítico, ya que si el proyecto BPM y la selección del equipo BPM no están bien definidos, entonces el resultado final puede ser un desastre. En esta primera etapa se debe definir quién será el líder del proyecto, los procesos críticos que se están analizando y mejorando, así como los requerimientos de los clientes de estos procesos seleccionados.
- Definición: trata específicamente la delimitación correcta del proceso o procesos críticos seleccionados. En esta etapa, el grupo del proyecto BPM debe darse a la tarea de entender y representar gráficamente el o los procesos en análisis.
- Mejoramiento: consiste en analizar detalladamente el proceso crítico seleccionado. Una de las principales herramientas a utilizar en este paso es el análisis de valor agregado o mapas de valor agregado. Este tipo de herramienta permite diferenciar aquellas actividades que no generan valor agregado ni para la organización ni para el cliente final del proceso. Una vez identificadas las actividades que generan valor agregado, entonces el proceso se debe rediseñar considerando mayormente las

actividades que sí son de valor para la organización y para el cliente final.

- Medición: consiste en el desarrollo de indicadores de medición del desempeño del proceso en análisis. Todo proceso debe tener indicadores que cuantifiquen el desempeño. Esta etapa es transcendental para poder determinar si el proceso ha mejorado con respecto a un estado anterior.
- Mejoramiento continuo: si se parte del hecho de que se quiere que el proceso sea perfecto, entonces la mejora continua nunca se debe detener en buscar la perfección. Es importante comparar el proceso en análisis con otros procesos en otras organizaciones.

Los pasos de la metodología BPM se muestran en la Figura 8.



Figura 8: Modelo general de metodologías BPM.

Fuente: Quesada (2006)

2.2.4 Sistemas de Gestión de Procesos de Negocio

Para poder llevar a cabo el conjunto de prácticas BPM se requieren de suites o sistemas BPM denominadas Sistemas de Gestión de Procesos de Negocios, el cual se define como un conjunto de servicios y herramientas que tienen como finalidad administrar los procesos de negocio es decir, analizar, definir, ejecutar, monitorear y controlar los procesos. (Gartner, 2009).

Un BPMS (*Business Process Management Systems*) debe cumplir con las siguientes funciones (Gartner, 2009):

1. Soporte al modelado y análisis de procesos de negocio, incluyendo todos los aspectos del flujo de trabajo: tareas, funciones, decisiones,

autorizaciones, opiniones, escalamiento, colaboraciones, flujos, reglas, políticas, formularios, eventos, metas y escenarios.

2. Apoyo a los cambios del proceso durante el diseño y la ejecución de transacciones.
3. Coordinar cualquier tipo de patrón de interacción entre usuarios, tareas del sistema, información de recursos.
4. Permitir a los participantes manipular y administrar información.
5. Apoyo a la definición, manipulación y gestión de reglas de negocio de las empresas y los usuarios de IT.
6. Apoyar a los usuarios y colaborar con grupos sobre elementos de trabajo del proceso, debe permitir a las empresas y a los profesionales de IT trabajar juntos en el diseño, desarrollo, ejecución y mejora de procesos.
7. Dar soporte al monitoreo, reportes, análisis y notificación de actividades y eventos que afectan el trabajo, utilizando los datos de las operaciones terminadas y en progreso.
8. Soporte al proceso de simulación del proceso y optimización de los resultados de negocio mediante datos estimados ó históricos.
9. Interoperabilidad con software externo, especialmente con los servicios web de arquitectura orientada a servicios SOA, incluyendo la orquestación de procesos.
10. Permitir la administración de los artefactos del proceso a lo largo de todo su ciclo de vida con la finalidad de facilitar la reutilización y la gestión de los cambios.
11. Incluir los procesos de muestra, conjuntos de reglas e interfaces de usuario.

Los principales componentes de un BPMS son (SOA Agenda, 2010):

- Modelador Gráfico de Procesos (*Business Modeler*): permite modelar los procesos de negocio, simular su ejecución, definir métricas para el monitoreo, y exportar al lenguaje estándar de procesos (*Business Execution Language*, BPEL por sus siglas en inglés). Esta es la etapa

donde se diseñan y construyen los procesos. Primero se dibuja el proceso mediante objetos (tareas, eventos, compuertas y otros) y luego se asignan atributos a esos objetos (ejecutores, formularios, calendarios, tiempos, condiciones). Cada día cobra más importancia que el modelado esté soportado por el estándar BPMN (*Business Process Modeling Notation*).

Este estándar se está imponiendo al resto de los formatos y estándares para diagramar existentes en el mercado ya que permite de forma fácil y sencilla una mejor interpretación de los procesos diagramados por parte del área de negocio.

- Ambiente de Integración y Desarrollo (*Integration Developer*): es la herramienta que permite implementar procesos y servicios. Esta herramienta permite integrar tanto pantallas (para interacción de un participante) como servicios (interacción con sistemas).
- Servidor de Procesos de Negocio (*Process Server*): es el motor que permite ejecutar los procesos de negocio, en ella 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 la carga, cambiar el flujo de negocio, y realizar acciones correctivas (según las reglas de negocio). La ejecución es completamente transparente para los usuarios puesto que a partir del modelado anterior (los atributos son los metadatos que se crearán en una base de datos) se generará toda la aplicación mediante los motores internos, es decir, como una compilación de lo definido anteriormente pero con la gran ventaja de que en la fase anterior se ha realizado sin escribir una sola línea de código.
- Monitor de Actividades de Negocio (*Business Activity Monitoring* o BAM): es una aplicación de administración que permite gestionar los procesos y servicios, gráficamente se pueden ver indicadores de performance, y los SLA. Adicionalmente se pueden definir alertas y triggers de acuerdo

a los eventos de negocio que sucedan en el proceso. También puede proveer datos reales a los modelos (*Business Modeler*) para ajustar las simulaciones y lograr el mejoramiento continuo.

La Figura 9 muestra los componentes de un BPMS.

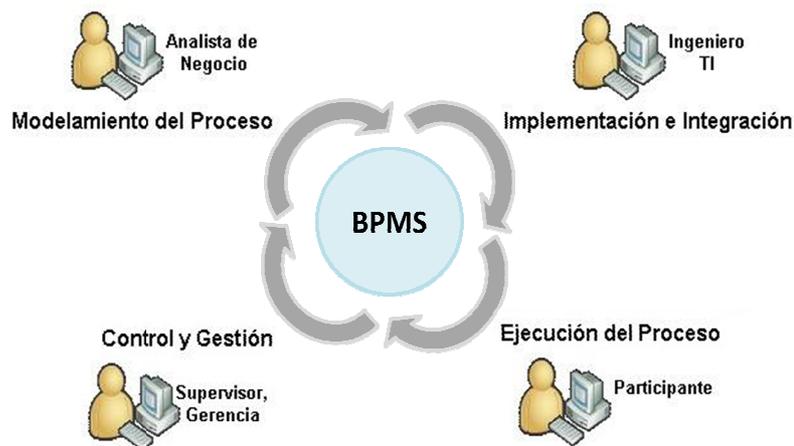


Figura 9: Componentes de un BPMS.

Fuente: Felipe (2008)

En la actualidad existen numerosos BPMS de código abierto o propietarios que implementan soluciones de BPM/flujo de trabajo, es por ello que previamente se realizó una evaluación con el formulario BPM a los BPMS Intalio, Bonita y Oracle. El BPMS seleccionado para modelar el Proceso de Gestión de Versiones y Liberaciones es Intalio *Community Edition* 6.0.2.

2.2.4.1 Software de implementación: BPMS Intalio *Community Edition* 6.0.2

Intalio proporciona una plataforma para la gestión de procesos de negocio complejos. La plataforma de Intalio está construida en Eclipse, el modelador utiliza la notación BPMN y utiliza el motor ODE (*Orchestration Director Engine*) BPEL de Apache. Proporciona todos los componentes necesarios para el diseño, despliegue y optimización de procesos de negocio más complejos. (Intalio, 2009).

Para el modelar el Proceso de Gestión de Versiones y Liberaciones en este Trabajo Especial de Grado, se utilizará el BPMS Intalio 6.0.2 en su edición comunitaria.

Intalio *Community Edition* 6.0.2 es un BPMS basado en estándares y puede ser utilizado gratuitamente, es por ello que la usan más de 50.000 organizaciones en todo el mundo. Posee las siguientes características:

- Está conformado por dos componentes: Intalio|*Designer* e Intalio|*Server*
- Es un subconjunto de Intalio *Enterprise Edition*

Componentes de Intalio

- Intalio|*Designer*

Es utilizada por analistas de negocios, ingenieros de software y administradores de sistemas para modelar procesos de negocios utilizando la notación BPMN. Esta soportado por el proyecto Eclipse Europa, y puede ser instalado en ambientes Windows, Linux y Mac OSX. El modelador BPMN de Intalio|*Designer* permite diseñar cualquier proceso usando las especificaciones de BPMN 1.1 o de BPMN 2.0, luego genera automáticamente el código BPEL. El modelador BPMN soporta BPMN 2.0, ampliada para construir flujo de trabajo humano con BPEL4People.

El diseñador de Intalio permite crear formularios para el desarrollo de *forms* usados en los pasos de *workflows* humanos. Los formularios se elaboran haciendo *drag-and-drop* sobre una extensa paleta de diseño que contiene elementos agrupados según su función. (Ver Figura 10). Los formularios están basados en el estándar XForms.

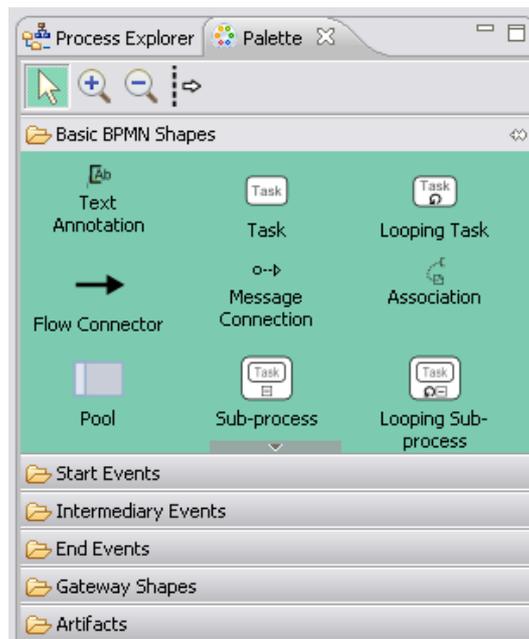


Figura 10: Paleta de Intalio|Designer.

Fuente: Intalio (2009)

En el Anexo 2 se explica cada uno de los elementos de la paleta.

Adicionalmente el diseñador incluye un mapeador (*mapper*) de datos que permite que muchas actividades diseñadas en un proceso puedan ser enlazadas a sistemas externos, o a datos generados al ejecutar procesos de un *workflow*. Este mapeador produce lenguaje XPath (*XML Path Language*) y/o código XSLT (*Extensible Stylesheet Language Transformations*) desde la asignación gráfica de mapas. Las características de Intalio|Designer se detallan en el Anexo 2.

- Intalio|Server

Es un servidor nativo de BPEL 2.0 basado en la arquitectura J2EE (*Java 2 Platform Enterprise Edition*) 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.

Este motor de ejecución realiza transformaciones XSLT durante la ejecución de procesos mediante la función `doXSLTransform` de BPEL.

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 que permite integrar formas de trabajo mediante XForms, soporta 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. Todos los sistemas externos se exponen transparentemente como servicios web WSDL y los procesos desplegados pueden colocar sus interfaces WSDL sobre un registro del catalogo UDDI. La capa de integración de servicios web de Intalio|Server es accionado por Apache Axis2. Las características de Intalio|Server se detallan en el Anexo 4.

Una vez desarrollados los procesos, es necesario que estos se comuniquen, para ello se hará uso de los servicios web.

2.2.4.2 Servicios Web

En términos sencillos, un WS (*Web Services*) se define como cualquier sistema de software diseñado para soportar interacción máquina a máquina sobre una red.

Un servicio web es un recurso programable y direccionable mediante una URL (*Uniform Resource Locator*). Desde el punto de vista del programador, se define como una clase, que se auto-define usando XML, y que es accesible mediante HTTP. (Álvarez y Espada, 2001).

El consorcio *World Wide Web Consortium (W3C)* lo define como un sistema de software diseñado para mantener interacciones interoperables de máquina-a-máquina sobre una red:

1. Tiene una interfaz descrita en un formato procesable por una máquina (específicamente WSDL).
2. Otros sistemas interactúan con el servicio web de una manera prescrita por su descripción usando mensajes SOAP, generalmente usando HTTP (*HyperText Transfer Protocol*) con una serialización de XML en conjunto con otros estándares de Web.

Los WS se han convertido en el estandarte de SOA, ya que posee un conjunto de características que permiten cubrir todos los principios de la orientación a servicios.

2.2.4.3 Arquitectura Orientada a Servicios

En líneas generales SOA (*Service Oriented Architecture*) es una arquitectura de software donde todas las tareas y procesos de software son implementados como servicios para ser consumidos sobre una red.

SOA es una arquitectura de software que comienza con una definición de interfaz y construye toda la topología de la aplicación como una topología de interfaces, implementaciones y llamados a interfaces. Sería mejor llamada "arquitectura orientada a interfaces". SOA es una relación de servicios y consumidores de servicios, ambos suficientemente amplios para representar una función de negocios completa. (Gartner, 2003).

En el Anexo 5 se describe detalladamente las propiedades, componentes y arquitectura tecnológica de SOA.

El protocolo de comunicación que van a utilizar los WS será SOAP.

2.2.4.4 Protocolo Simple de Acceso a Objetos

SOAP (*Simple Object Access Protocol*) es un protocolo simple basado en XML el cual permite que las aplicaciones intercambien información a través de HTTP. Dicho de una manera más simple SOAP es un protocolo para acceder a un Servicio Web. (W3C, 2007).

SOAP especifica (Pelechano, 2004):

- 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 (*Remote Procedure Call*), 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 procesa mensaje 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.
- Una descripción de cómo se debe transportar un mensaje SOAP sobre HTTP y SMTP (*Simple Mail Transfer Protocol*). Se definirán *bindings* a otros protocolos de transporte en futuras versiones de la especificación.

Un mensaje SOAP no es más que un documento en formato XML que consta de tres partes bien definidas que se pueden apreciar en las Figuras 11 y 12, las cuales son (W3C, 2007):

- El envoltorio (*envelope*): es el elemento más importante y de mayor jerarquía dentro del documento XML y representa al mensaje que lleva almacenado dicho documento.
- La cabecera (*header*): es un mecanismo genérico que se utiliza para añadir características adicionales al mensaje. El modo en la que se

añadan cada uno de los campos dependerá del servicio implementado entre cliente y servidor.

- El cuerpo (*body*): es un contenedor de información en el cual se almacenan los datos que se quieren transmitir de lado a lado de la comunicación. Dentro de este campo, SOAP define un elemento de uso opcional denominado *fault* utilizado en los mensajes de respuesta para indicar al cliente que se produjo algún error en el servidor.

En un mensaje la cabecera representa una parte opcional mientras que el cuerpo es una parte obligatoria. La cabecera y el cuerpo pueden tener múltiples subpartes en forma de bloques de cabecera y bloques de cuerpo.

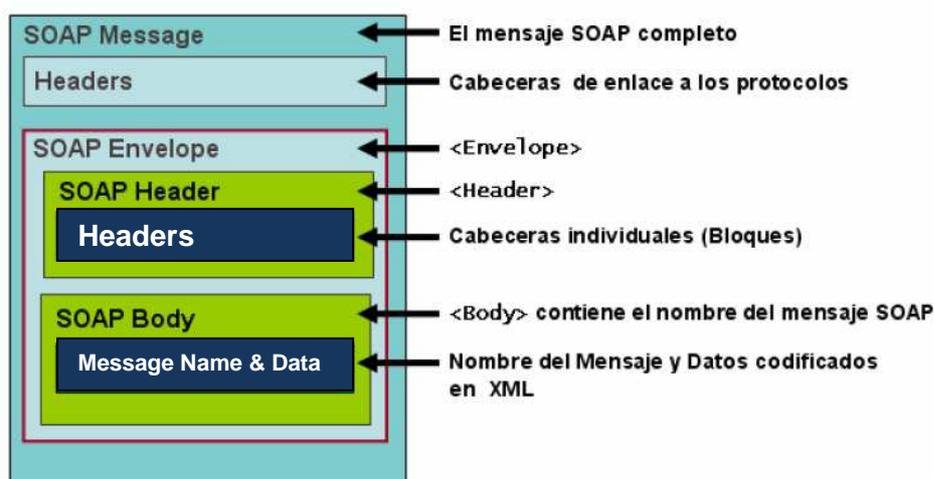


Figura 11: Formato de un mensaje SOAP.

Fuente: Pelechano (2004)

```

<?xml version="1.0" ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
              xmlns:xsd="http://www.w3.org/2001/XMLSchema"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Header>
    <sb:Unknown
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true"
      xmlns:sb="http://soapinterop.org/">
    </sb:Unknown>
  </env:Header>
  <env:Body>
    <sb:echoString xmlns:sb="http://soapinterop.org/"
      env:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
      <inputString xsi:type="xsd:string">Hello world</inputString>
    </sb:echoString>
  </env:Body>
</env:Envelope>

```

Figura 12: Mensaje SOAP.

Fuente: W3C (2007)

2.2.4.5 Lenguaje de Descripción de Servicios Web

Los protocolos de comunicación posibilitaron la estandarización de los mensajes en la web. WSDL (*Web Service Description Language*) mediante XML define una gramática específica que permite describir los servicios de la red como un conjunto de nodos de comunicación que posee la capacidad de intercambiar mensajes. Un documento en WSDL define los servicios como una colección de nodos de red o puertos. WSDL detalla los protocolos y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo; se utiliza también para la localización y ubicación de estos servicios en Internet. (Sicuma, 2007).

El documento WSDL de un servicio presenta dos partes básicas que se muestran en la Figura 13 (Pelechano, 2004):

- La parte abstracta, la cual es independiente de la aplicación y está compuesta por:
 - *portType*: para las operaciones que proporciona el servicio web. Cada *portType* es una colección lógica de *operations*.

- operation: define el intercambio simple de mensajes.
- message: es una unidad de comunicación que representa un intercambio de datos en una única transmisión lógica.
- types: define los tipos de datos que utiliza *operations* (por defecto el esquema XML).
- La parte concreta, se encarga de definir los enlaces a protocolos e información de los puntos finales de acceso al servicio, está compuesta por:
 - binding: especifica la codificación de los mensajes y los enlaces a protocolos de todas las operaciones y mensajes definida en un *portType*.
 - port: especifica una dirección URI (*Uniform Resource Identifier*) para el enlace definiendo un único punto de destino.
 - service: definen colecciones de puntos de destino.

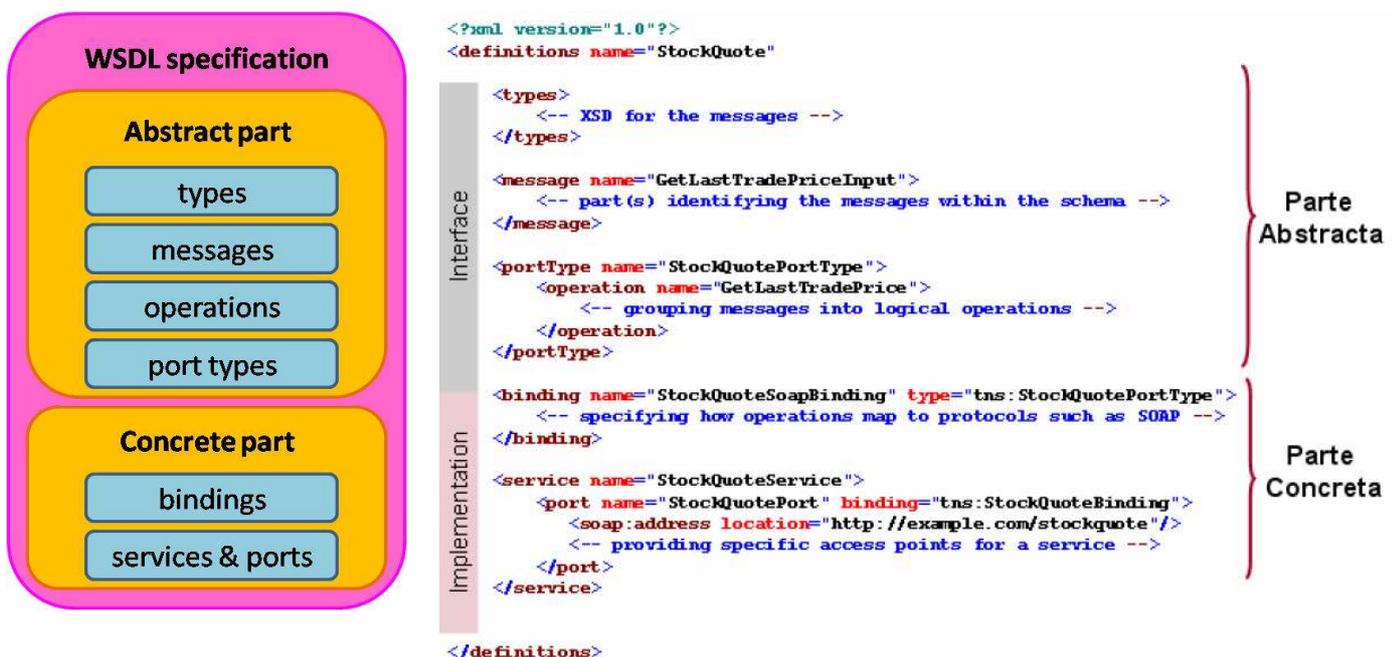


Figura 13: Estructura y ejemplo de un archivo WSDL.

Fuente: Sicuma (2007)

2.2.4.6 Descubrimiento, Descripción e Integración Universal

UDDI (*Universal Description, Discovery and Integration*) fue creado inicialmente por IBM, Microsoft y Ariba, pero desde su versión 3 (publicada en el año 2002) ha pasado a manos de OASIS que desde entonces va a determinar su futuro y extensiones. (Oasis, 2004).

UDDI especifica un marco para describir y descubrir servicios web. Adicionalmente, describe un registro de servicios web e interfaces de programación para la publicación, recuperación y gestión de la información acerca de los servicios descritos. De hecho, UDDI es en sí mismo un conjunto de servicios web. La especificación UDDI define los servicios que apoyan la descripción y el descubrimiento de (Oasis, 2004):

- Empresas, organizaciones y otros proveedores de servicios web
- Los servicios web que están disponibles
- Las interfaces técnicas que se pueden utilizar para acceder y gestionar los servicios

La especificación UDDI tiene dos objetivos esenciales (Pelechano, 2004):

- Ser un soporte a los desarrolladores para encontrar información sobre servicios web y poder construir clientes
- Facilitar el enlace dinámico de los servicios web, permitiendo consultar referencias y acceder a servicios de interés

Los proveedores cuando desarrollan un Servicio Web lo incorporan a la UDDI de manera que pueda ser localizado por quien lo requiera. El registro de un negocio en UDDI consta de tres partes (Sicuma, 2007):

- Páginas blancas: contiene la información básica para identificar y contactar a la empresa que ofrece el servicio.
- Páginas amarillas: describe utilizando una taxonomía específica al servicio web, esta categorización facilita la búsqueda.

- Páginas verdes: contiene información técnica que describe el funcionamiento y comportamiento del servicio web.

La información de un registro UDDI se almacena en ficheros XML con una estructura jerárquica (ver Figura 14). Los elementos de esta estructura son (Pelechano, 2004):

- *businessEntity*: es el elemento *top-level*, describe un negocio o una entidad que ha registrado un servicio UDDI. Por ejemplo: Departamento de Contabilidad, Servidor de Aplicaciones Corporativo. Este elemento soporta información estándar tales como nombre, descripción e información de contacto, así como información de metadatos (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 y otros).
- *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 *UriType* utilizado para especificar los siete tipos de puntos de entrada: *mailto*, *http*, *Https*, *Ftp*, *Fax*, *Phone*, *Other*.
- *tModel (Technology Model)*: es una estructura de metadatos genérica para representar cualquier concepto o construcción (definiciones de protocolos, ficheros WSDL, XML *Schemas*, espacios de nombres, esquemas de categorías y otros).

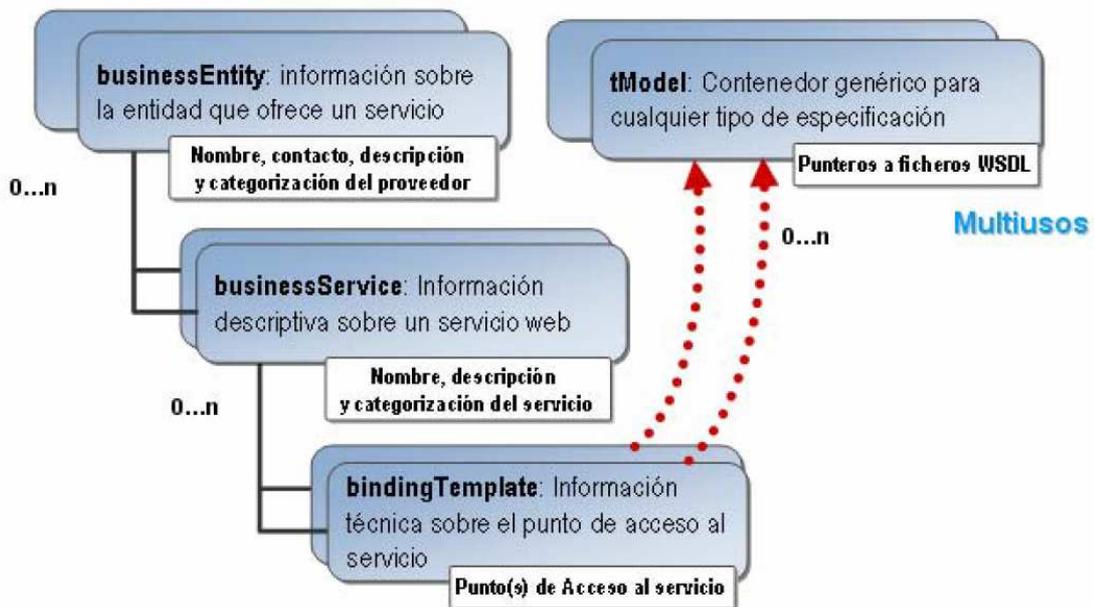


Figura 14: Estructura de Datos de UDDI.

Fuente: Pelechano (2004)

2.2.4.7 Orquestación de Servicios Web

La orquestación de servicios web se basa en un modelo centralizado en el cual las interacciones no se realizan directamente entre los servicios web sino que existe una entidad encargada de definir la lógica de interacción. (Pérez, 2007).

WSO (*Web Services Orchestration*) es el proceso de coordinar servicios web, los cuales realizan una o varias tareas o actividades. Una vez que una actividad termina, la ejecución continúa en la siguiente actividad definida. Este conjunto de servicios web se puede emplear para ofrecer una solución que se adapte a algún proceso de cualquier negocio en particular. (W3C, 2004).

Un proceso Web es de orquestación de servicios cuando es controlado totalmente por una única entidad. Ésta define completamente las interacciones con los servicios componentes, y la lógica requerida para conducir correctamente esas interacciones.

Los elementos básicos para llevar a cabo WSO son (Yang, 2003):

- Procesos: vistos como una serie de actividades cuyo fin es la ejecución de una tarea determinada.
- Actividades: representan reglas bien definidas del proceso de negocios.
- Flujo de datos: describe la información intercambiada entre actividades.
- Flujo de control: describe el orden en que serán ejecutadas las actividades del flujo.

El Anexo 6 contiene otros aspectos relacionados con la orquestación de servicios web.

CAPÍTULO III: MARCO METODOLÓGICO

Este capítulo describe la manera en que se realizará el estudio como respuesta al problema planteado previamente.

3.1 Bases metodológicas de la investigación

En esta sección se describirá el procedimiento utilizado para lograr el objetivo de la investigación.

La metodología hace referencia al conjunto de procedimientos basados en principios lógicos, utilizados para alcanzar una gama de objetivos que rigen en una investigación científica. (Eyssautier, 2006).

3.1.1 Tipo de investigación

Tomando en cuenta el problema planteado, el cual radica en realizar un Sistema de Información para la Gestión de Versiones y Liberaciones de Software según ITIL v3, este Trabajo Especial de Grado se define del tipo Proyecto Factible, que según UPEL (2003):

“Consiste en la elaboración y desarrollo de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales; puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos. El proyecto debe tener apoyo en una investigación de tipo documental, de campo o un diseño que incluya ambas modalidades” (p. 16).

El proyecto factible conforma un proceso de planificación en el cual la investigación es una etapa que le proporciona información para sustentar la propuesta.

Tomando en cuenta las diversas concepciones, el proyecto factible se desarrolla a través de las siguientes etapas (Dubs, 2002):

- El diagnóstico de las necesidades, el cual puede basarse en una investigación de campo o en una investigación documental
- Planteamiento y fundamentación teórica de la propuesta
- El procedimiento metodológico
- Las actividades y recursos necesarios para su ejecución
- El análisis de viabilidad o factibilidad del proyecto
- En caso de su desarrollo, la ejecución de la propuesta y la evaluación tanto del proceso como de sus resultados

3.1.2 Fuentes de Revisión e Investigación

Las técnicas de recolección de datos son las distintas formas o maneras de obtener la información. (Arias, 1999).

Las técnicas utilizadas para la recolección de datos fueron:

- Revisión Bibliográfica: en ella se acude especialmente a revistas científicas, informes y monografías, medios de comunicación que reflejan con más dinamismo que los libros los adelantos que se producen. (Sabino, 1992).

La técnica consiste en recopilar y revisar todos aquellos documentos que permiten 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 versiones.

La revisión de los estudios previos nos permitirá (Pedraz, 2004):

- Ahondar en la explicación de las razones por las que hemos elegido dicho tema de investigación
- Conocer el estado actual del tema: qué se sabe, qué aspectos quedan por investigar

- Identificar el marco de referencia, las definiciones conceptuales y operativas de las variables estudiadas
- Descubrir los métodos para la recolección y análisis de los datos utilizados
- Contar con elementos para la discusión, donde se compararán los resultados que obtengamos con los de los estudios previos
- Fuentes Infográficas: consiste en recopilar información a través de fuentes en línea tales como webinars, foros, páginas web.

3.2 Metodología de Desarrollo de Proyectos BPM

Diversos grupos de estudio promueven sus metodologías para la implementación de iniciativas BPM. Dentro de Intalio existe una combinación de mejores prácticas y experiencias adquiridas en la relación con los clientes denominada Marco de Trabajo de Modelado de Procesos *Process Modeling Framework*. (Felipe, 2009).

Process Modelling Framework (PMF) es un enfoque estructurado para el modelado de procesos que resulta consistente, preciso y legible ya que los diagramas emplean la notación BPMN. PMF es más que una metodología, es una práctica de gestión de procesos que involucra a ejecutivos, analistas de negocios, gerentes y personal de TI. Cada parte de una organización tiene distintas necesidades; un solo diagrama no cubre dichas necesidades, razón por la cual PMF se enfoca en la idea de que las personas y los sistemas afectan a un proceso de negocio por lo que se requiere de diagramas con diferentes niveles de detalle para los procesos. (Intalio, 2009).

El hecho de implementar procesos de negocio bajo el enfoque BPM resulta ser una tarea compleja que requiere de disciplina, un nivel de madurez metodológico y alto en estándares, es decir (SOA Agenda, 2010):

- Tener conocimiento de los procesos del negocio
- Dominar la orientación a servicios (SOA)

- Aplicar varias tecnologías nuevas (XML, *WebServices*, BAM, BPEL y otros)
- Tener un esquema de control y apoyo adecuado

De acuerdo a los requisitos antes mencionados, se puede observar que implementar BPM puede llevar mucho tiempo o una alta inversión inicial. (SOA Agenda, 2010).

3.2.1 Metodología de Gestión de los Procesos de Negocio Sustentada en el uso de Patrones

En el año 2008, Pedro Bonillo propuso en su trabajo doctoral la “Metodología de Gestión de los Procesos de Negocio Sustentada en el uso de Patrones”, en la cual define lo siguiente: “La propuesta metodológica para la gerencia de los procesos de negocio sustentada en el uso de patrones está conformada por dos sub-procesos: Crear Proceso y Administrar Proceso”. (Bonillo, 2008).

Cabe destacar que de esta metodología solo se va a tomar como referencia su estructura, sin emplear los patrones definidos en él.

La Figura 15 muestra la estructura de la metodología.

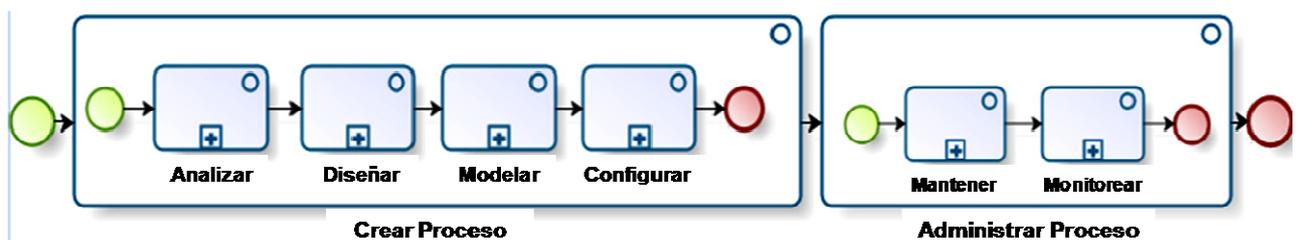


Figura 15: BPD del proceso de la Metodología de Gestión de los Procesos Sustentada en el uso de Patrones.

Fuente: Bonillo (2008)

De los subprocesos definidos en la metodología, solo se hará uso de Crear Proceso; no se requiere de Administrar Proceso ya que la versión comunitaria de Intalio no cuenta con la funcionalidad para monitorear procesos.

Crear Proceso está compuesto por cuatro sub procesos, los cuales se detallan a continuación (Bonillo, 2008):

- Analizar: la función principal de este sub-proceso es recibir los requerimientos de nuevos procesos o mejoras y levantar información detallada.
- Diseñar: tiene como objetivo crear un estándar de los procesos tomando en cuenta las mejores prácticas, identifica las brechas y riesgos presentes al diseñar el proceso.
- Modelar: se realiza un diagrama de acuerdo al modelo elaborado en la etapa de diseño, luego de modelar el proceso se simulará y finalmente será evaluado por el usuario final.
- Configurar: a través de un lenguaje estándar se configurará la lógica de integración, negocio e interfaz.

A continuación se explicarán los entregables del Trabajo Especial de Grado que se obtuvieron al aplicar la Metodología de Gestión de los Procesos de Negocio Sustentada en el uso de Patrones (GPNSUP).

3.2.1.1 Crear Proceso

Constituye la primera fase de la metodología, la cual inicia con la solicitud de creación de un nuevo proceso, dicha solicitud pasa por los sub-procesos que se indican a continuación:

Analizar

Según la Metodología GPNSUP en esta etapa una persona de la organización solicitó la creación del Sistema de Información de Gestión de Versiones y Liberaciones a través del Escritorio de Ayuda.

- Identificar requerimientos: en ella se realizan las siguientes actividades:
 - Describir requerimiento: se requiere la creación del Sistema de Información de Gestión de Versiones y Liberaciones de Software para la Gerencia de Programa de Soluciones TI de la empresa

CANTV. El sistema debe realizar seguimiento del proceso de Gestión de Versiones, almacenar los planes creados y actualizar DML y DS

Definir prioridad: el Analista de Negocio asigna prioridad Alta al requerimiento, ya que es un proceso que impacta toda la organización y se requiere con urgencia.

Evaluar cartelera de actividades: tomando en cuenta las actividades que conforman la metodología se le asigna una duración en días promedio para un proceso medianamente complejo de acuerdo a la información proporcionada por el Tutor basándose en proyectos de este tipo. (Ver Anexo 6).

- Levantar información: en esta etapa se realizan las actividades que se listan a continuación:
 - Clasificar la documentación del Proceso de Gestión de Versiones y Liberaciones
 - Conocer plataformas o sistemas utilizados para la Gestión de Procesos de Negocio
 - Definir las fases del proyecto
 - Estudiar la Metodología de Gestión de Procesos
 - Identificar las necesidades del usuario final ya que las mismas serán tomadas en cuenta durante el diseño y construcción del Sistema de Información de Gestión de Versiones y Liberaciones
 - Investigar sobre el BPMS Intalio *Community Edition* 6.0.2

Adicionalmente en esta etapa se debe:

Mapear el marco referencial del proceso: el marco referencial a utilizar es ITIL v3, específicamente el proceso de Gestión de Versiones y Liberaciones. Se realizó un análisis de brechas para comparar las actividades de gestión de versiones realizadas en CANTV con las actividades que propone el proceso de ITIL.

Levantar la situación actual: CANTV sólo cumple con las actividades de supervisión del entorno de desarrollo, supervisión

del entorno de pruebas, supervisión del entorno de producción y contemplar marcha atrás. El proceso se lleva a cabo por medio del intercambio de correos electrónicos, documentos en Excel, carece de una librería definitiva de software y de hardware y no existe una colaboración definida para cada uno de los actores.

Además fue necesario decidir la forma en que se debe llevar a cabo el proceso, para ello es necesario solicitar el RFC que contenga todos los cambios a ser implementados en la nueva versión.

Diseñar

Comprende los siguientes subprocesos:

- Estandarizar proceso: el proceso de Gestión de Versiones y Liberaciones se estandarizó tomando en cuenta el marco referencial de la Biblioteca de Infraestructura de Tecnologías de Información v3.
- Evaluar riesgos: los riesgos del diseño del proceso de Gestión de Versiones y Liberaciones fueron proporcionados por el Tutor basándose en proyectos de este tipo. Se diagnosticaron los siguientes tipos de riesgos:
 - Humano: falta de disponibilidad, cambio de ejecutivos, resistencia al cambio.
 - Operacional: generados por interrupciones o fallas en el abastecimiento de las operaciones.
 - Financiero: originados por problemas con el negocio, falta de capital.
 - Proyecto: riesgos de excederse en el presupuesto, en los plazos, o por calidad inadecuada del producto o servicio.
 - Técnico: por el uso de nuevas tecnologías, fallas técnicas, desconocimiento.
- Realizar modelo propuesto: para ello es necesario elaborar los diagramas del proceso de Gestión de Versiones y Liberaciones usando el Lenguaje Unificado de Modelado (*Unified Modeling Language* o UML por sus siglas en inglés). Las extensiones de UML para el modelado de

negocio aportan elementos muy importantes ya que muchas veces proporcionan otras vistas de la arquitectura de negocio que son más difíciles de observar usando únicamente BPMN. Estos diagramas permitieron definir el sistema, detallar los artefactos, documentar y proceder a construir el mismo.

Los objetivos de UML son muchos, pero se pueden resumir sus funciones (Hernández, 2009):

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: a partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Los diagramas UML realizados para la Gestión de Versiones y Liberaciones son:

Diagrama de Caso de Uso

Para el modelo de caso de uso se identificaron ocho actores, los cuales se describen en la Tabla 2.

Actor	Descripción
Administrador Proceso Gestión de Versiones	<ul style="list-style-type: none"> • Desarrolla estrategias de liberación • Publica: planes de pruebas, planes de implementación, planes de distribución y planes de comunicación y adiestramiento • Realiza revisiones a los planes de liberación • Solicita las actualizaciones a los CIs • Participa en reuniones con el personal de Gestión de Cambio • Asegura que los clientes, personal de apoyo y demás partes involucradas estén capacitados y provistos de la información y documentación pertinente

Administrador Gestión de Versiones	<ul style="list-style-type: none"> • Aprueba las cotizaciones en caso de que la versión tenga que ser comprada a proveedores externos • Aprueba los resultados obtenidos de las pruebas unitarias, integrales y de aceptación de usuario • Apoya las actividades de lanzamiento
Equipo de Desarrollo	<ul style="list-style-type: none"> • Desarrolla el software de acuerdo al RFC • Entrega la versión con la documentación de apoyo para la construcción de entornos de prueba • Participa en las pruebas • Participa en la implementación del sistema desarrollado • Si el sistema es comprado, se encarga de realizar la configuración del mismo • Apoya las actividades de <i>back-out</i>
Equipo de Pruebas	<ul style="list-style-type: none"> • Administra los recursos de prueba • Mantiene el entorno de pruebas en estado de alerta para probar las versiones de acuerdo a su fecha de lanzamiento • Realiza todas las pruebas, incluyendo las de <i>back-out</i> • Comunica los resultados de las pruebas • Participa en las decisiones para realizar la liberación
Gestión de Cambios	<ul style="list-style-type: none"> • Suministra el documento de petición de cambio (RFC) • Aprueba la liberación y los cambios de la versión
Gestión de Configuración y Activos del Servicio	<ul style="list-style-type: none"> • Asegura que las nuevas versiones se reflejen en la CMDB
Proveedores Externos	<ul style="list-style-type: none"> • Encargado de emitir cotizaciones de acuerdo a los requerimientos del cliente • Suministra la versión una vez que se concreta el proceso de compra
Grupo de Usuarios Finales	<ul style="list-style-type: none"> • Está formado por un grupo de usuarios finales seleccionados con la finalidad de probar la nueva versión • Aporta observaciones y resultados de la prueba

Tabla 2: Descripción de actores del Sistema de Gestión de Versiones y Liberaciones de Software.

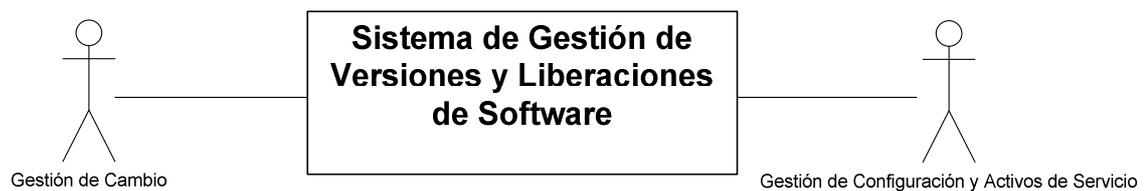


Figura 16: Nivel 0 Sistema de Gestión de Versiones y Liberaciones de Software.

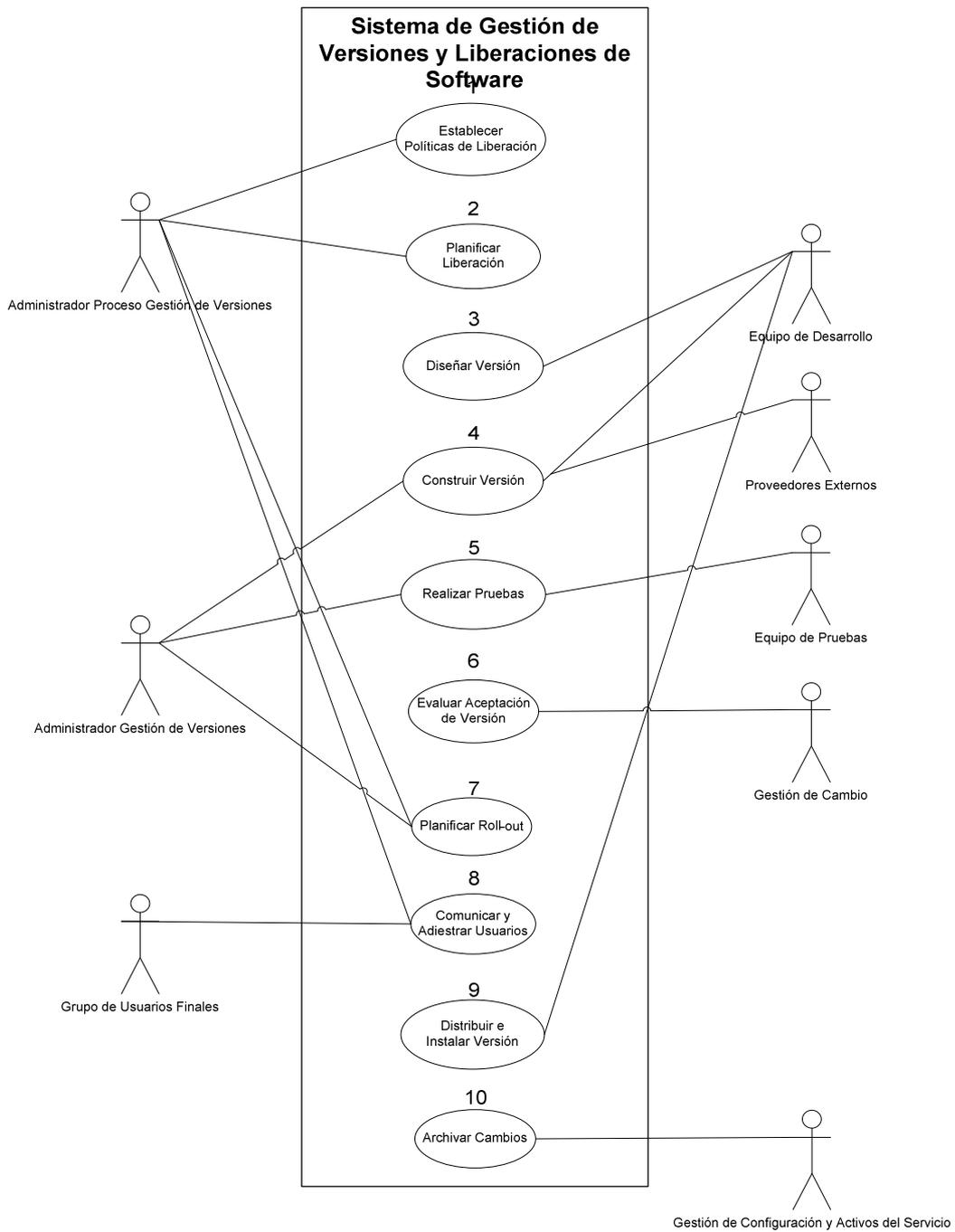


Figura 17: Nivel 1 Sistema de Gestión de Versiones y Liberaciones de Software.

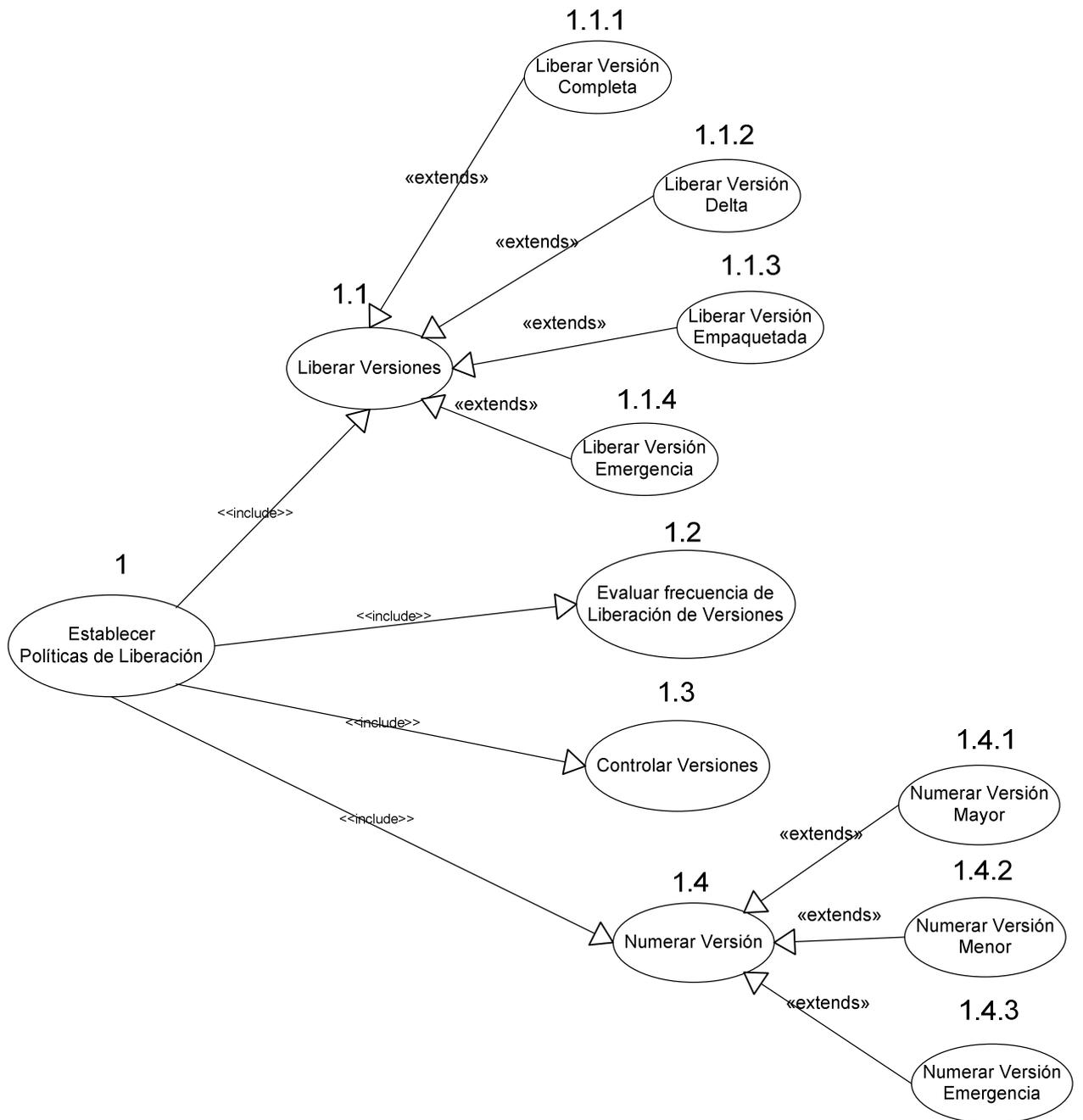


Figura 18: Nivel 2 Establecer Políticas de Liberación.

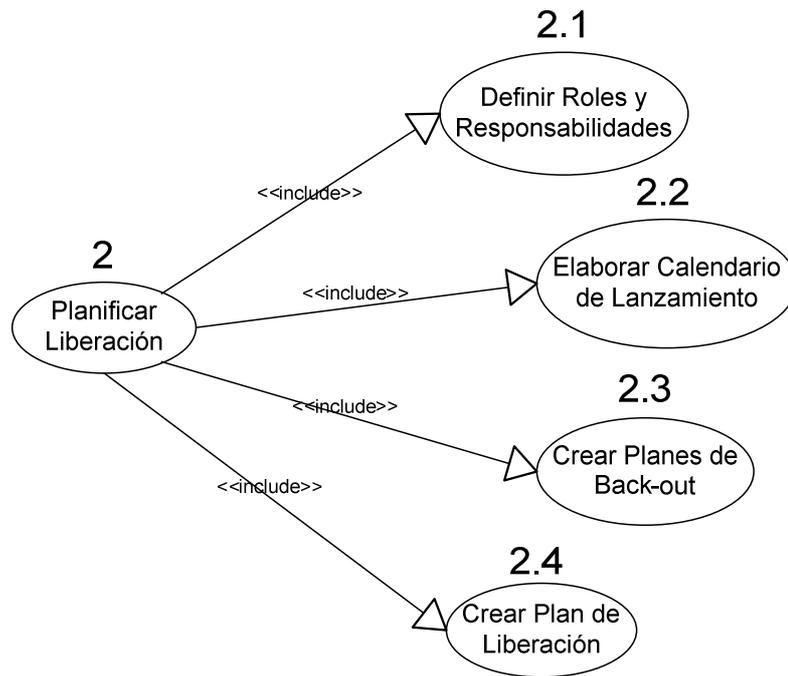


Figura 19: Nivel 2 Planificar Liberación.

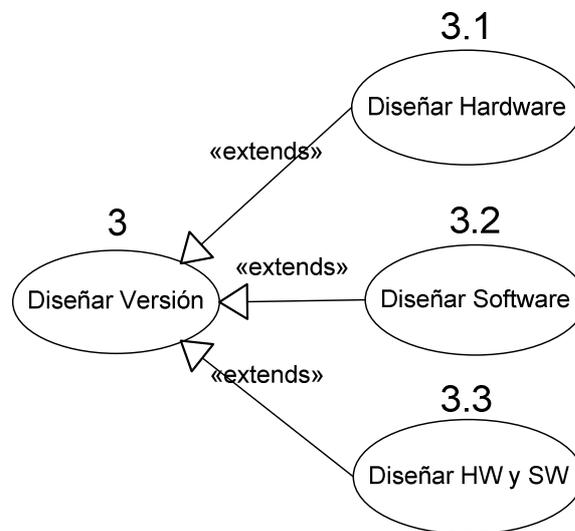


Figura 20: Nivel 2 Diseñar Versión.

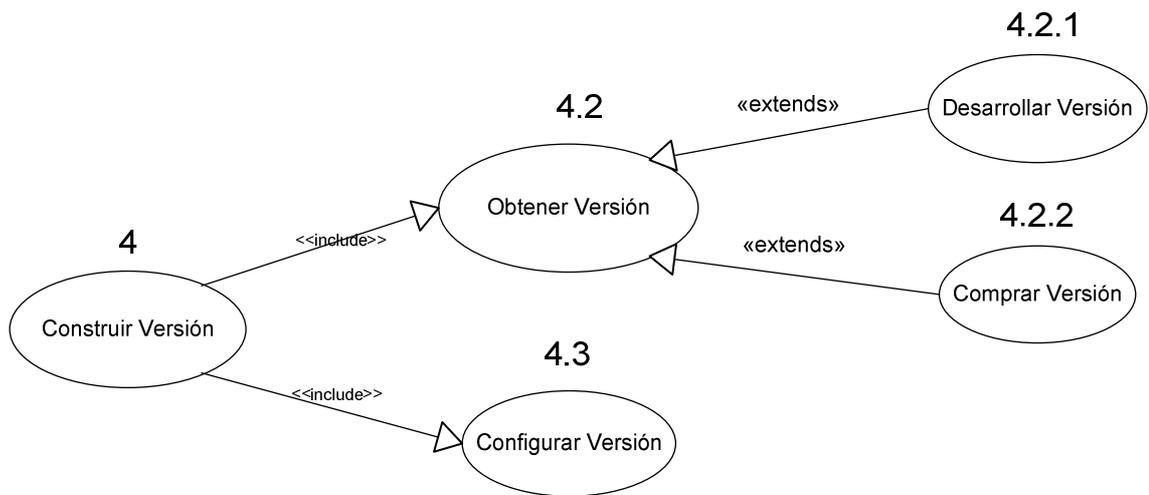


Figura 21: Nivel 2 Construir Versión.

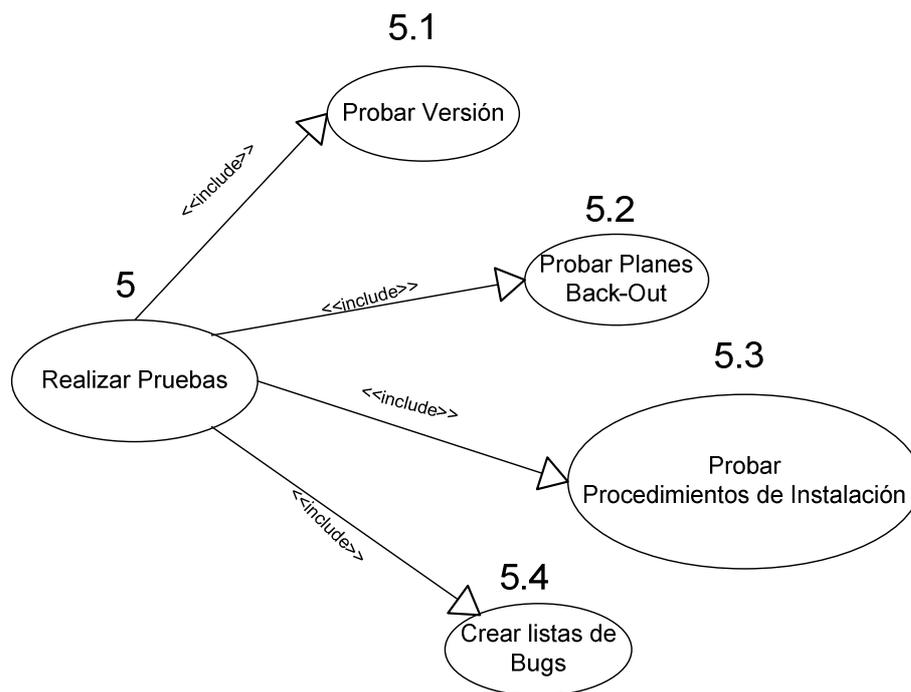


Figura 22: Nivel 2 Realizar Pruebas.

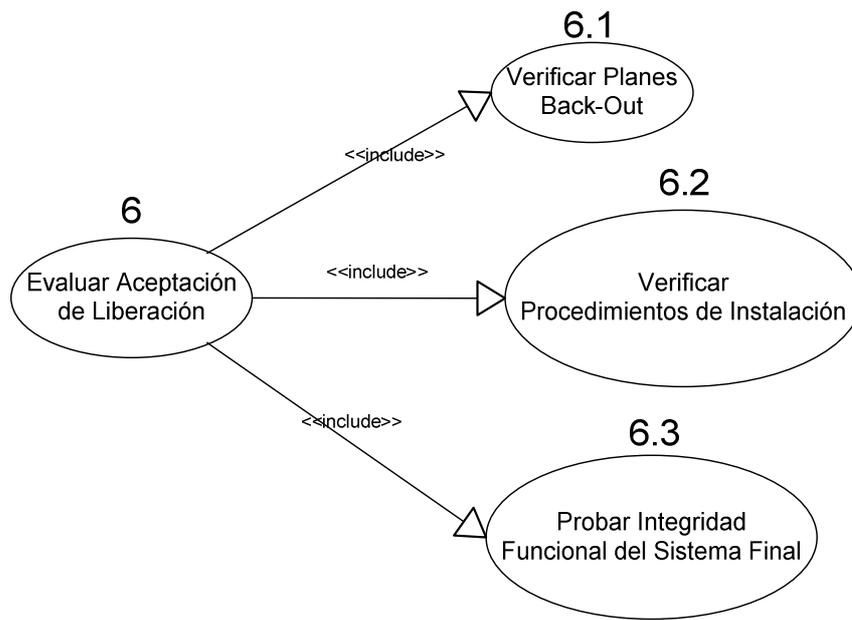


Figura 23: Nivel 2 Evaluar Aceptación de Liberación.

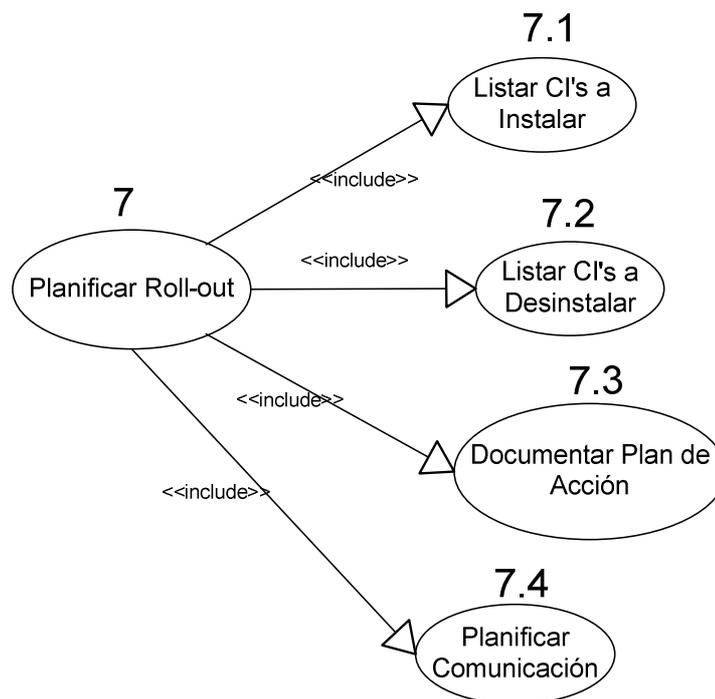


Figura 24: Nivel 2 Planificar Implementación.

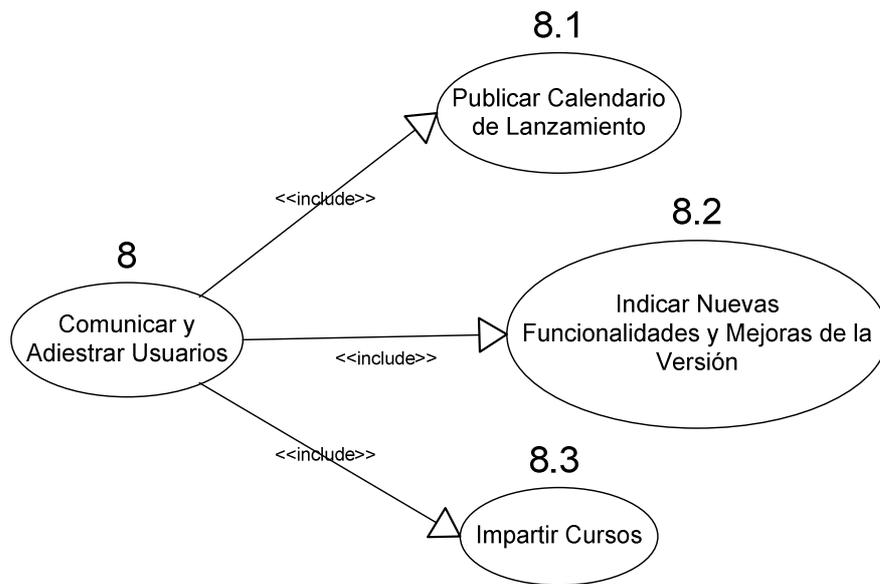


Figura 25: Nivel 2 Comunicar y Adiestrar Usuarios.

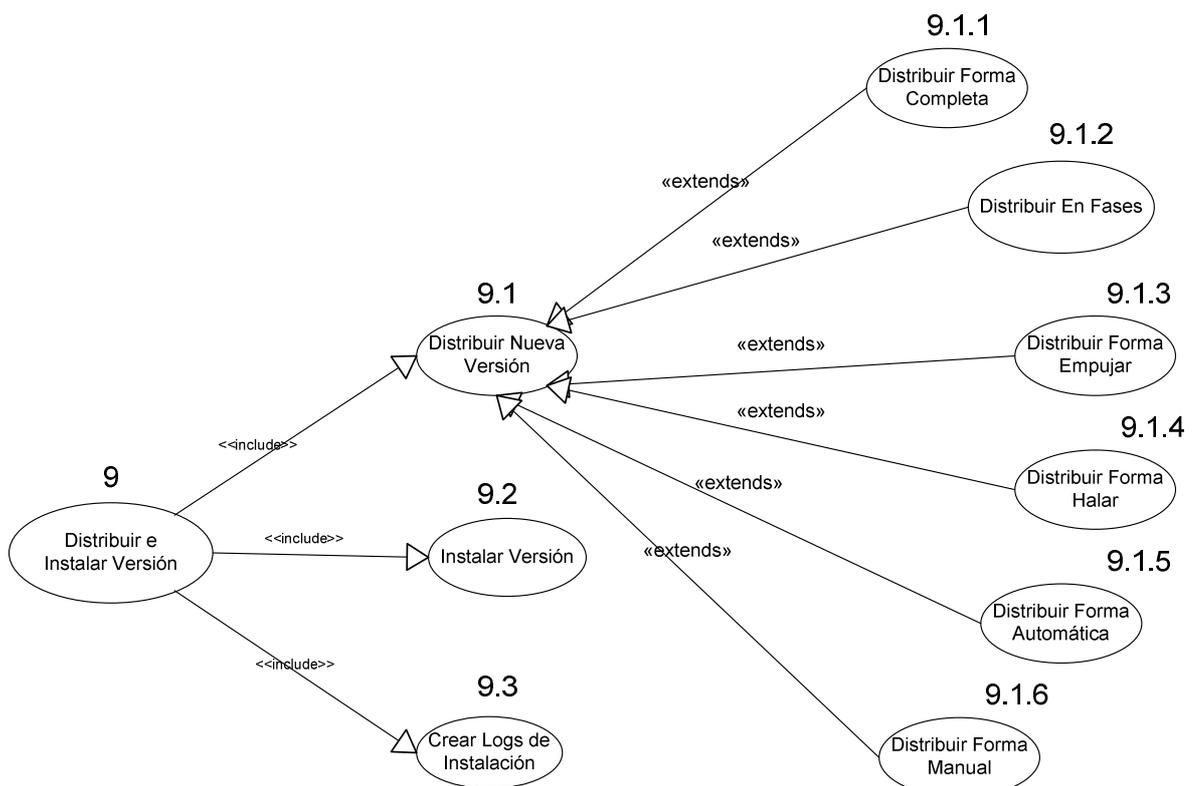


Figura 26: Nivel 2 Distribuir e Instalar Versión.

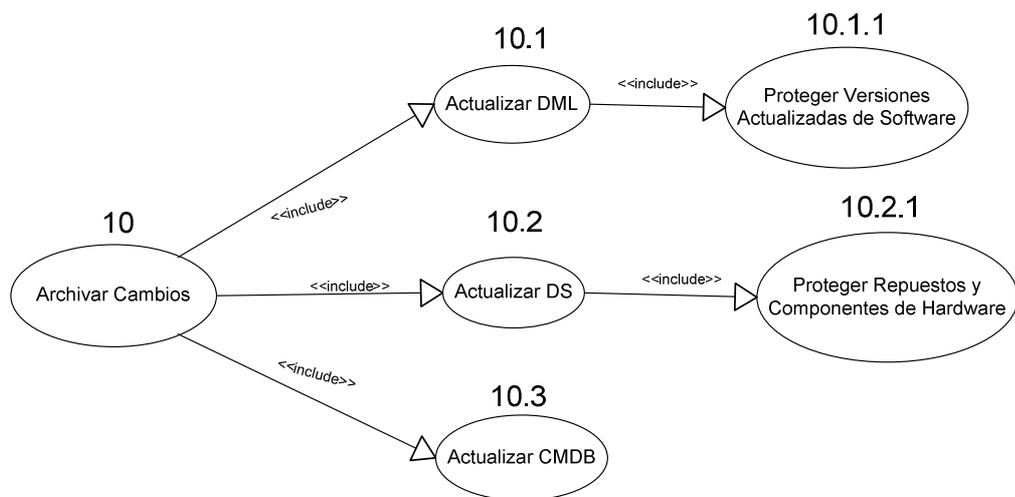


Figura 27: Nivel 2 Archivar Cambios.

Descripción de Casos de Uso

Nombre	Establecer Políticas de Liberación
Actores	Administrador Gestión de Versiones
Descripción	En esta fase es necesario definir quienes son las personas involucradas en el proceso, numerar la versión y clasificar la versión.
Pre-condición	Gestión de Cambio debe emitir un RFC.
Post-condición	Se establecen las políticas de liberación para el RFC suministrado y se procede a planificar la liberación.

Tabla 3: Especificar Caso de Uso Nivel 2 – Establecer Políticas de Liberación.

Nombre	Planificar Liberación
Actores	Administrador Gestión de Versiones
Descripción	Se definen los roles y las responsabilidades, así como también se establece la fecha estimada de la liberación.
Pre-condición	No tiene.
Post-condición	Se produce el plan de liberación.

Tabla 4: Especificar Caso de Uso Nivel 2 – Planificar Liberación.

Nombre	Diseñar Versión
Actores	Equipo de Desarrollo
Descripción	En esta fase es necesario tomar en cuenta el tipo de versión (hardware, software o documentación) para diseñar la versión de acuerdo los requerimientos del RFC.
Pre-condición	No tiene.
Post-condición	Se crea el diseño de la versión.

Tabla 5: Especificar Caso de Uso Nivel 2 – Diseñar Versión.

Nombre	Construir Versión
Actores	Equipo de Desarrollo
Descripción	En este caso de uso se evalúa si la versión va a ser comprada a un proveedor externo o construida en casa. Independientemente de la evaluación se debe realizar la configuración de la versión.
Pre-condición	Tener el diseño de la versión
Post-condición	Se guardan los siguientes archivos de acuerdo al tipo de versión construido o comprado: <ul style="list-style-type: none"> • Versión de software (aplicación): licencia, script de instalación y manuales. • Versión de software (sistema operativo): licencia. • Versión de hardware: manuales. • Documentación: enlace con la documentación. Se actualiza la CMDB, la DML y DS

Tabla 6: Especificar Caso de Uso Nivel 2 – Construir Versión.

Nombre	Realizar Pruebas
Actores	Equipo de Desarrollo, Equipo de Pruebas, Grupo de Usuarios Finales y Administrador Gestión de Versiones
Descripción	La versión pasa por una etapa de pruebas unitarias (probar módulos por separado), integrales (validar la interrelación entre los módulos del sistema) y de aceptación de usuario (un grupo de usuarios finales debe probar que la versión cumple con sus expectativas).
Pre-condición	Contar con planes de pruebas unitarias, integrales y de aceptación de usuario
Post-condición	Obtener la aprobación de la versión por parte del Equipo de Pruebas, Administrador Gestión de Versiones y Grupo de Usuarios Finales.

Tabla 7: Especificar Caso de Uso Nivel 2 – Realizar Pruebas.

Nombre	Evaluar Aceptación de Versión
Actores	Gestión de Cambio
Descripción	En esta fase la Gestión de Cambio debe corroborar que la versión obtenida cumple con los requerimientos de la RFC suministrada previamente.
Pre-condición	Los resultados de las pruebas realizadas deben ser exitosos.
Post-condición	Se obtiene el resultado de la evaluación necesario para proceder con la implementación.

Tabla 8: Especificar Caso de Uso Nivel 2 – Evaluar Aceptación de Versión.

Nombre	Planificar Implementación o Roll-Out
Actores	Administrador Gestión de Versiones
Descripción	En esta fase se indica la manera en que se llevara a cabo la implementación (completa, en fases, empujar, halar, automática o manual), se listan los CI's a instalar y a desinstalar, se describen las acciones por sitio y los recursos necesarios para la implementación.
Pre-condición	La versión es aprobada por Gestión de Cambios.
Post-condición	Se obtiene el plan de implementación o roll-out.

Tabla 9: Especificar Caso de Uso Nivel 2 – Planificar Implementación.

Nombre	Comunicar y Adiestrar Usuarios
Actores	Administrador Gestión de Versiones y Grupo de Usuarios Finales
Descripción	Este caso de uso se encarga de publicar el calendario de eventos asociados a la nueva versión, se listan las nuevas funcionalidades y/o mejoras, se busca la mejor manera de dar a conocer la versión y se establecen las pautas para los cursos de adiestramiento.
Pre-condición	La versión debe estar lista para ser liberada.
Post-condición	Se genera el plan de comunicación y adiestramiento.

Tabla 10: Especificar Caso de Uso Nivel 2 – Comunicar y Adiestrar Usuarios.

Nombre	Distribuir e Instalar Versión
Actores	Equipo de Desarrollo, Administrador Gestión de Versiones, Gestión de Cambios
Descripción	Esta fase indica la forma de distribución de la versión (versión delta, versión completa o paquete de versiones) y se procede a realizar la instalación de la nueva versión. Una vez que la versión ha sido instalada, es necesario que Gestión de Cambios apruebe los cambios implementados; si estos no son aprobados se debe realizar un back-out (aplicar medidas de contingencia o restablecer el servicio).
Pre-condición	Tener el plan de back-out
Post-condición	Se realiza la distribución e instalación de la versión. Se reflejan los cambios en la base de datos.

Tabla 11: Especificar Caso de Uso Nivel 2 – Distribuir e Instalar Versión.

Nombre	Archivar Cambios
Actores	Gestión de Configuración y Activos del Servicio, Administrador Gestión de Versiones
Descripción	En esta etapa se actualizan la DML, DS y CMDB para que las nuevas versiones se reflejen en ellas.
Pre-condición	Cambio de Versión
Post-condición	La base de datos se encuentra actualizada

Tabla 12: Especificar Caso de Uso Nivel 2 – Archivar Cambios.

Diagrama de Clases

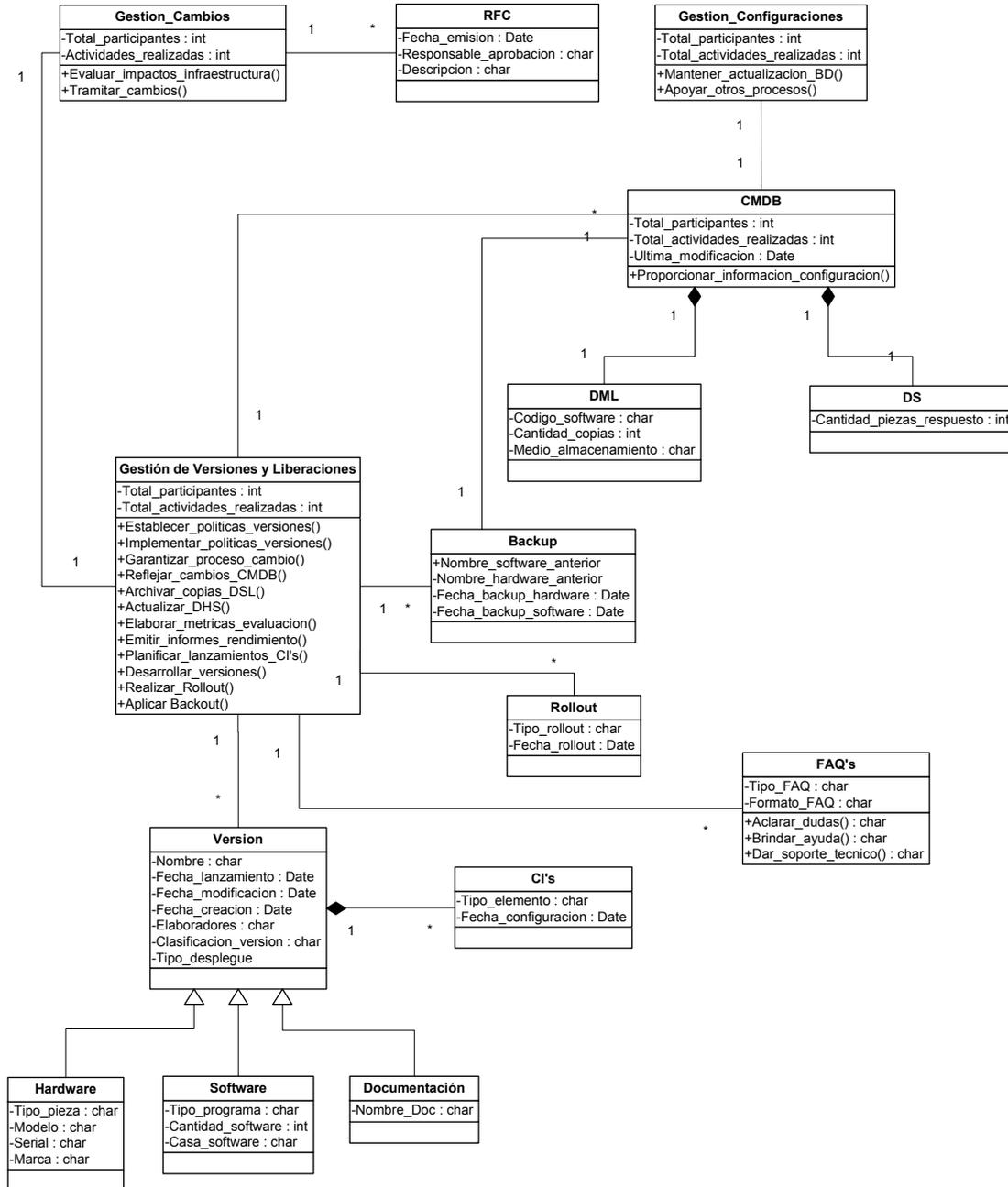


Figura 28: Diagrama de Clases del Proceso de Gestión de Versiones.

Diagrama de Secuencia

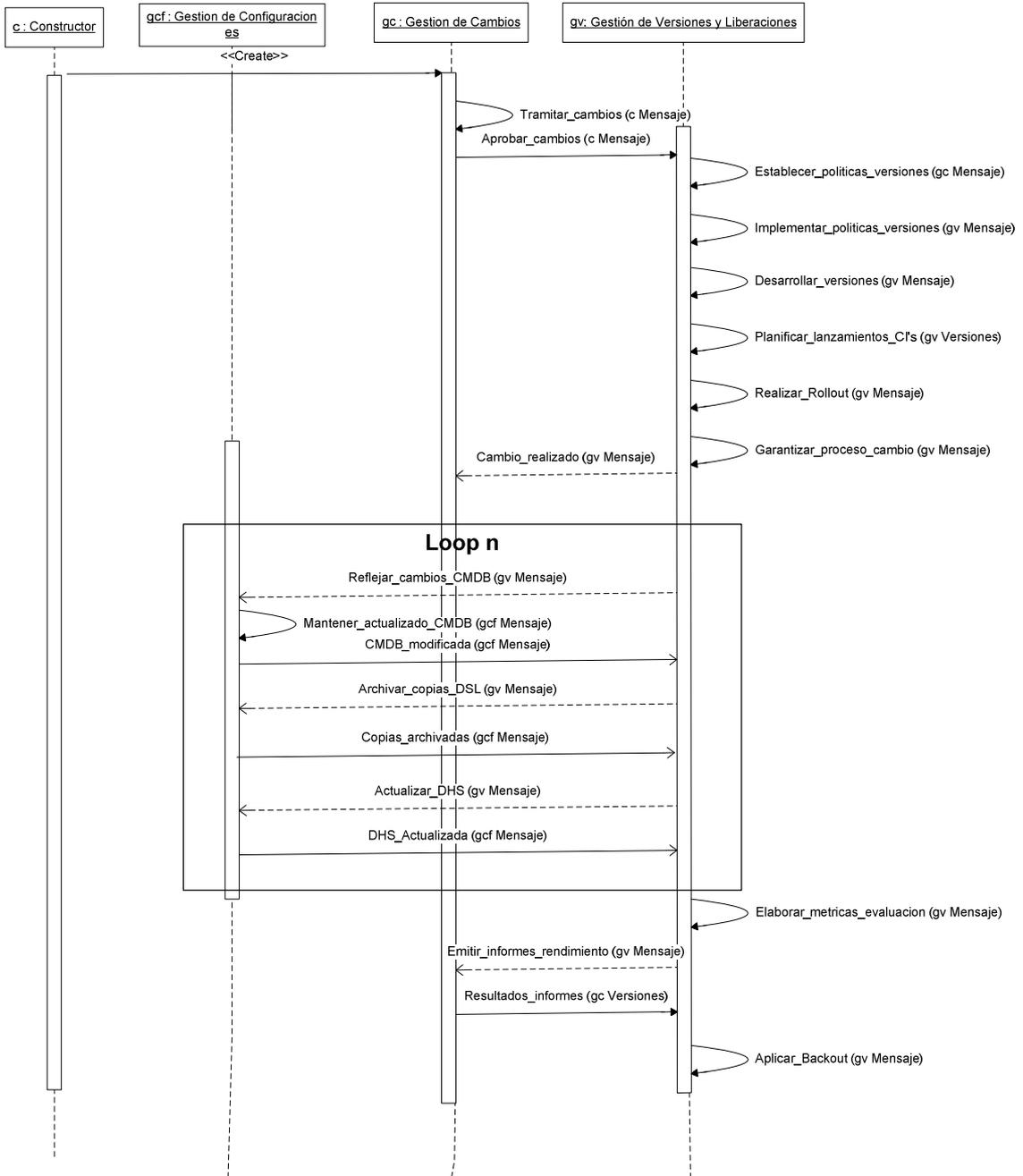


Figura 29: Diagrama de Secuencia del Proceso de Gestión de Versiones.

Diagrama de Estado

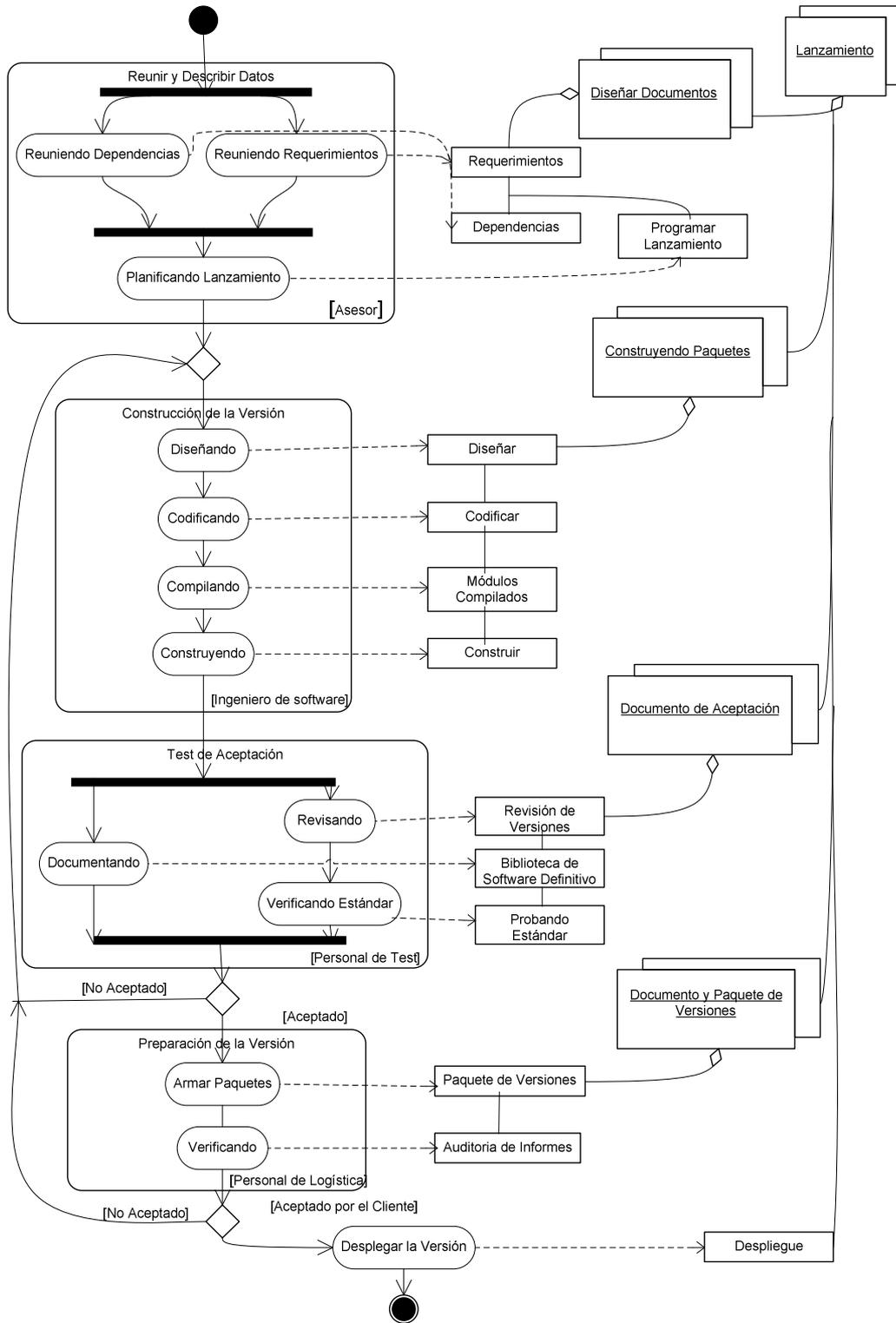


Figura 30: Diagrama de Estado del Proceso de Gestión de Versiones.

Diagrama de Actividad

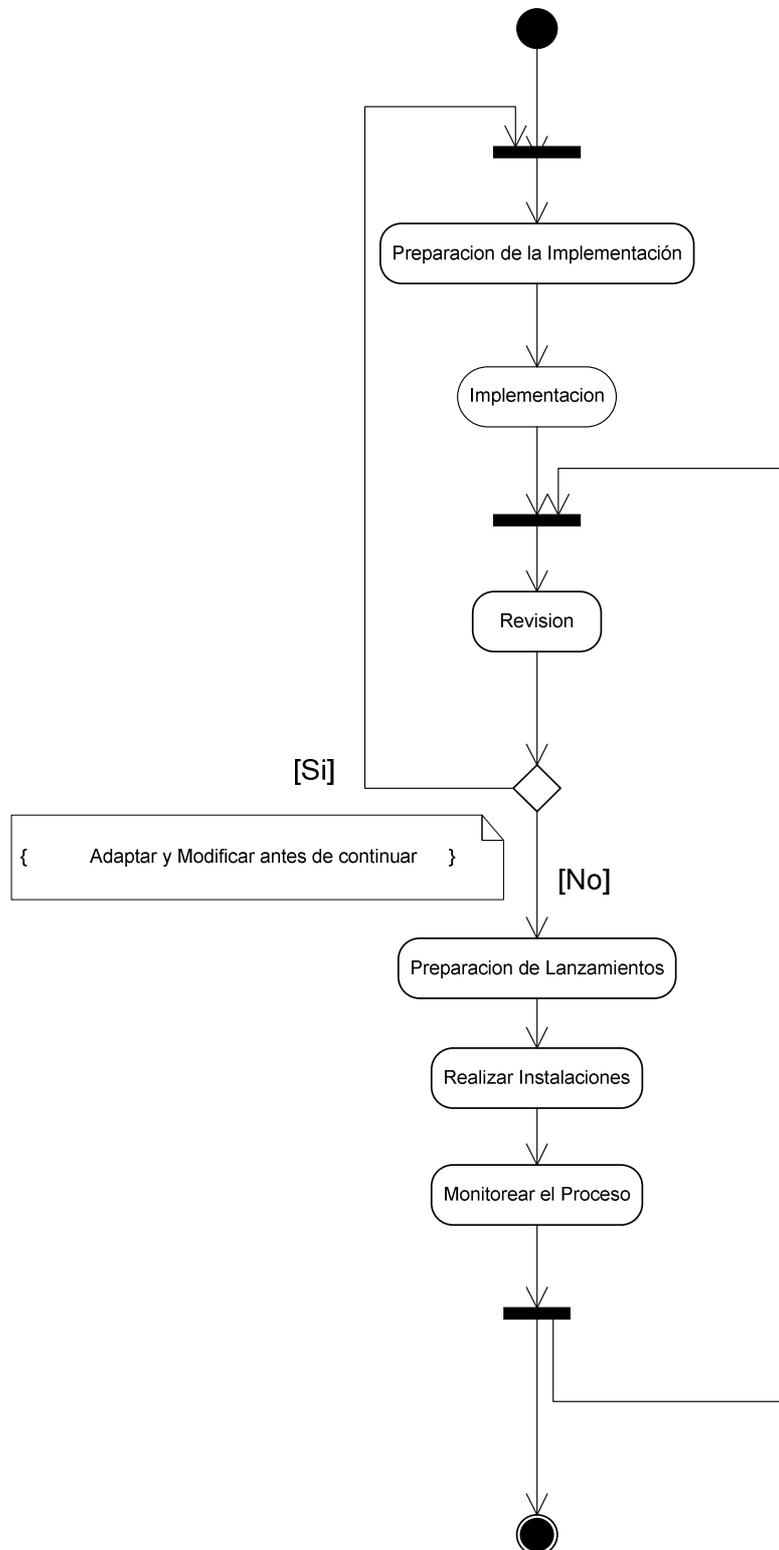


Figura 31: Diagrama de Actividad del Proceso de Gestión de Versiones.

Diagrama de Despliegue

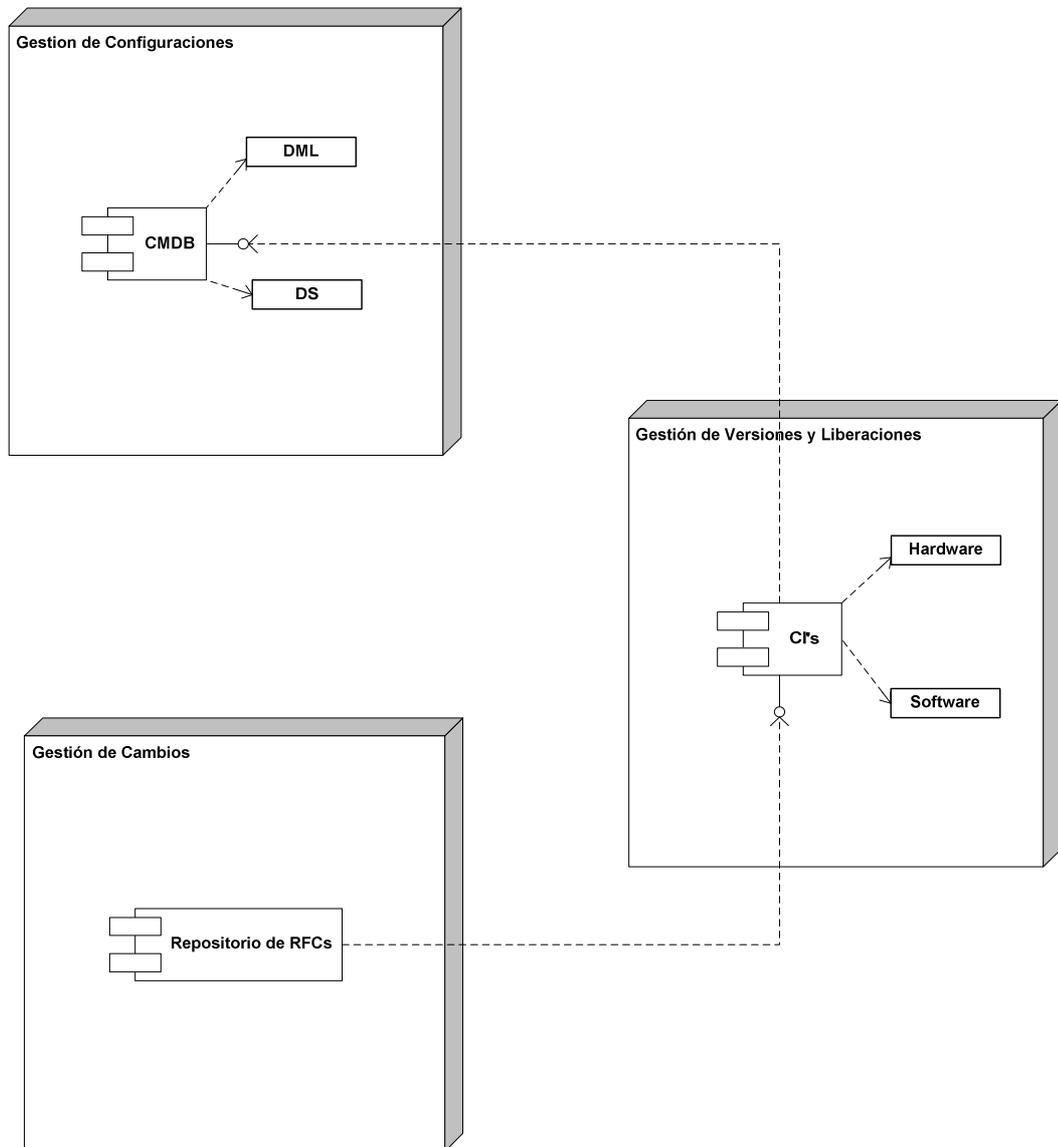


Figura 32: Diagrama de Despliegue del Proceso de Gestión de Versiones.

Modelo de Objetos del Dominio

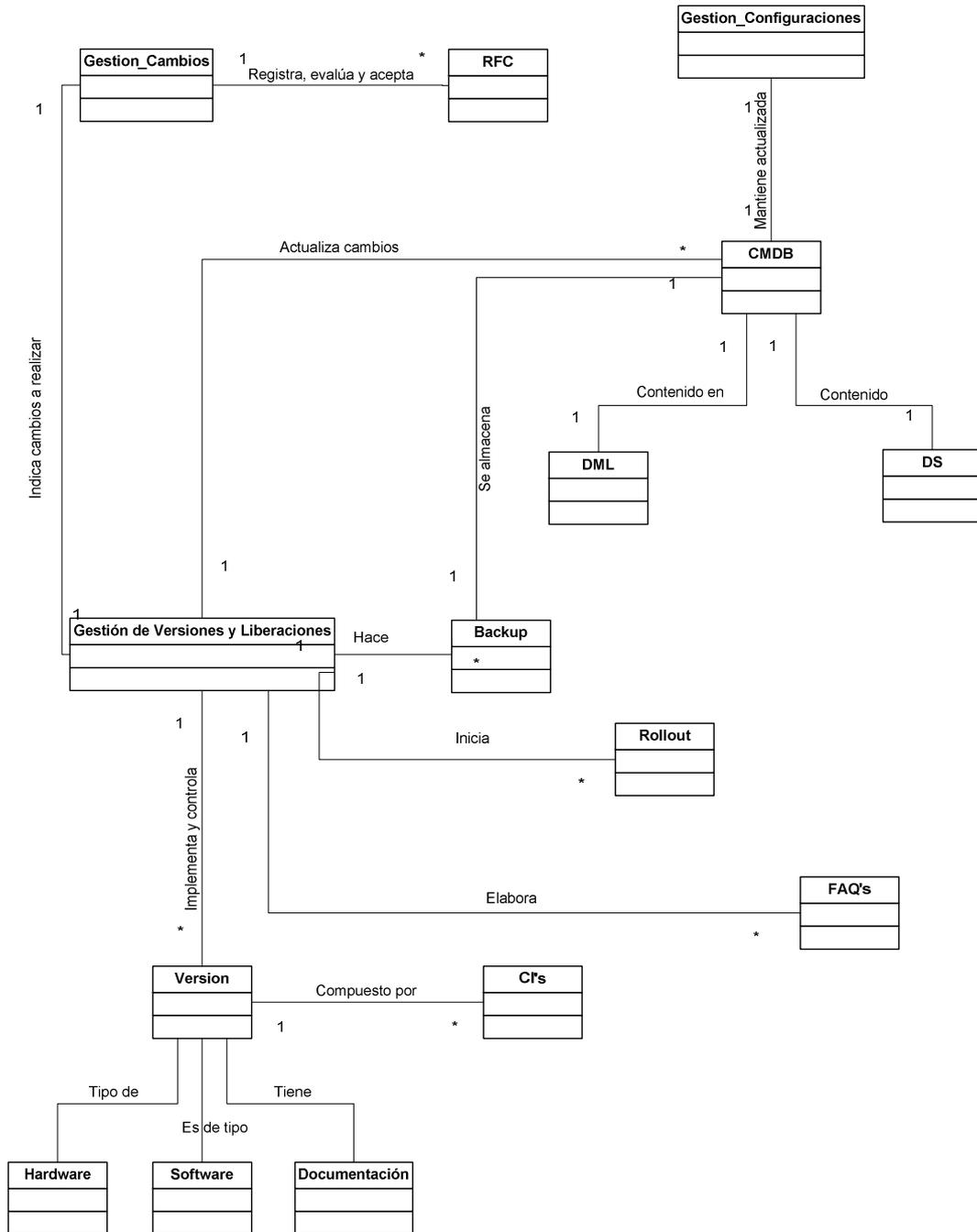


Figura 33: Diagrama de Modelo de Objetos del Dominio del Proceso de Gestión de Versiones

Adicionalmente, se realiza la arquitectura del Sistema de Información de Gestión de Versiones y Liberaciones de Software que cubre el objetivo de este Trabajo Especial de Grado, la misma resulta de la adaptación de la arquitectura del BPMS Intalio *Community Edition* 6.0.2. (Ver Figura 34).

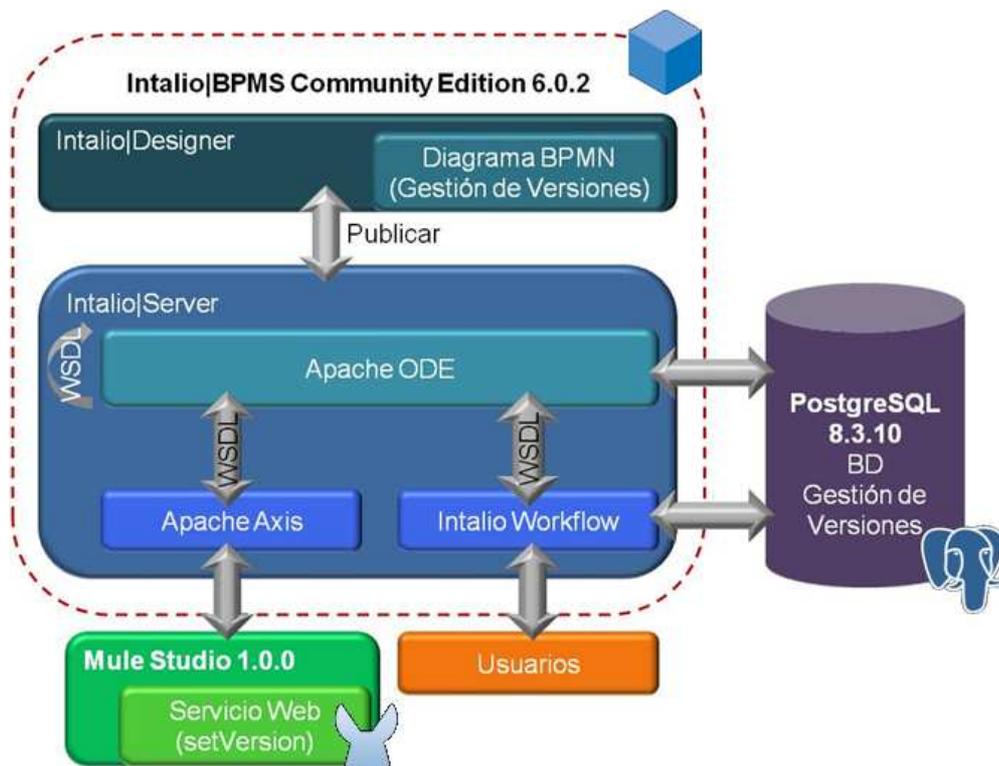


Figura 34: Arquitectura del Sistema de Información de Gestión de Versiones y Liberaciones.

Modelar

En esta fase de la metodología se realizan las siguientes actividades:

- Diagramar: se crea el *Business Process Diagram* (BPD) con los subprocesos a modelar (ver Figura 35). El primer paso implica crear el flujo de trabajo del proceso de Gestión de Versiones y Liberaciones en Intalio|*Designer* utilizando los elementos de la paleta.

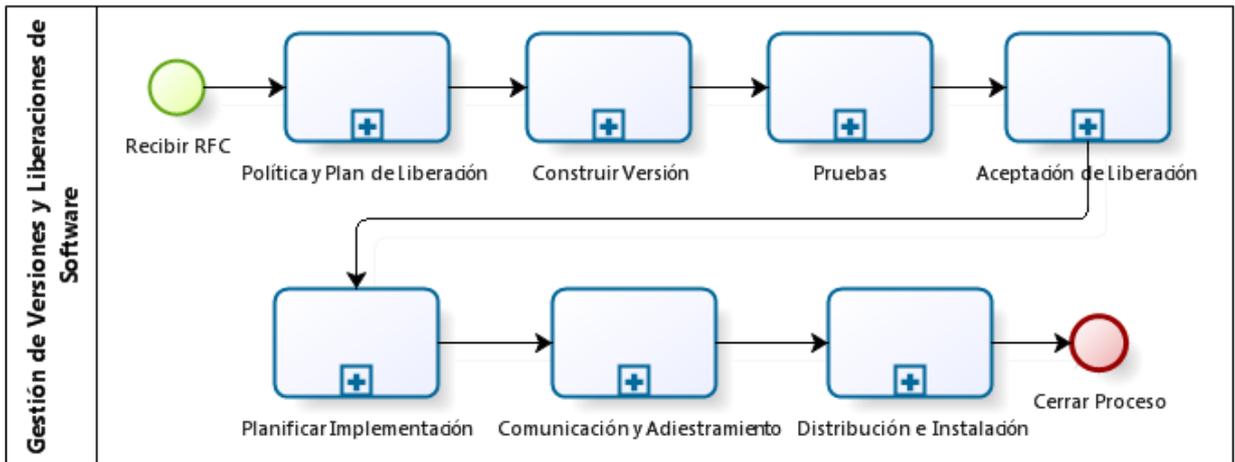


Figura 35: BPD del Proceso de Gestión de Versiones y Liberaciones de Software.

Para ello es necesario activar la perspectiva *Intalio|Designer* y crear un nuevo proyecto de tipo *Intalio|Designer Business Process Project* denominado “Gestión de Versiones_6”. Luego se crea una carpeta llamada Proceso y en ella se crea un nuevo *Business Process Diagram* llamado Gestión de Versiones_6.

Pools

Es necesario definir los roles del proceso de Gestión de Versiones y Liberaciones según ITIL v3 tomando en cuenta los actores definidos en los diagramas de caso de uso; se creará un pool con un color específico para cada rol a fin de diferenciar claramente las actividades de realiza cada rol.

Solo habrá un *pool* ejecutable llamado *Gestión de Versiones* el cual genera código BPEL.

Los *pools* no ejecutables de acuerdo a los roles son: *Equipo de Pruebas*, *Equipo de Desarrollo*, *Gestión de Cambios*, *Administrador Proceso Gestión de Versiones*, *Administrador Gestión de Versiones*, *Proveedores Externos* y *Grupo de Usuarios Finales*.

Los *pools* no ejecutables que representan sistemas que interactúan con el proceso son: *BD* y *Servicios*.

Iniciador

Al modelar en Intalio|*Designer* se requiere de un proceso cero, el cual tiene como función enviar el primer mensaje para iniciar el flujo de trabajo. En este caso Gestión de Cambios le envía un RFC a Gestión de Versiones.

Etapas

A continuación se explicará cómo se realizó el modelado de cada etapa del proceso de Gestión de Versiones y Liberaciones, para ello se modelaron las tareas que se realizan en cada una de ellas y luego se agruparon.

Política y Plan de Liberación: el Administrador del Proceso de Gestión de Versiones procede a elaborar el plan de liberación tomando en cuenta los requerimientos del RFC. (Ver Figura 36).



Figura 36: Política y Plan de Liberación.

Construir Versión: en esta etapa se crea la nueva versión o se modifica una versión existente, hay que tomar en cuenta si la versión va a ser desarrollada en casa por el Equipo de Desarrollo o si va a ser comprada a los Proveedores Externos. Una vez que se obtiene la versión, se

realizan las pruebas unitarias y el Administrador de Gestión de Versiones se encarga de evaluar los resultados.

Pruebas: el Administrador del Proceso de Gestión de Versiones se encarga de crear los planes de pruebas integrales y de aceptación de usuario para que sean ejecutadas por el Equipo de Pruebas y por el Grupo de Usuarios Finales respectivamente.

Aceptación de Liberación: se requiere que Gestión de Cambio evalúe la aprobación de la versión creada. (Ver Figura 37).

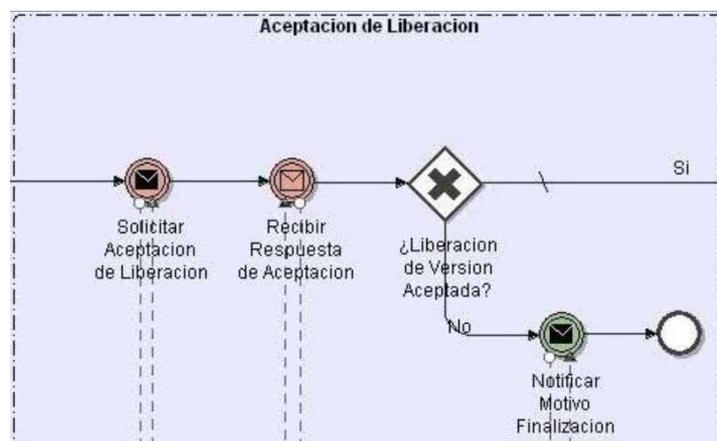


Figura 37: Aceptación de Liberación.

Planificar Implementación: el Administrador del Proceso de Gestión de Versiones realiza el plan de implementación para que el Equipo de Desarrollo ponga en marcha dicho plan. (Ver Figura 38).



Figura 38: Planificar Implementación.

Comunicación y Adiestramiento: en esta etapa hay que dar a conocer los cambios o nuevas funcionalidades de la versión creada, para ello el Administrador del Proceso de Gestión de Versiones crea un plan para cubrir esta necesidad. (Ver Figura 39).



Figura 39: Comunicación y Adiestramiento.

Distribución e Instalación: el Administrador del Proceso de Gestión de Versiones crea el plan de distribución e instalación, luego es necesario que la Gestión de Cambios apruebe los cambios realizados de acuerdo al RFC, si los cambios no son aprobados se debe realizar marcha atrás.

La Tabla 13 contiene las actividades modeladas en cada etapa del Proceso de Gestión de Versiones y Liberaciones.

Etapa	Actividad	Rol
Política y Plan de Liberación	Recibir RFC	Gestión de Cambios
	Solicitar Plan de Liberación	Administrador Proceso Gestión de Versiones
	Recibir Plan de Liberación	
Construir Versión	Solicitar Forma de Desarrollo	Equipo de Desarrollo
	Recibir Forma de Desarrollo	
	Desarrollar Versión	
	Recibir Versión	
	Solicitar Cotización	Proveedores Externos
	Recibir Cotización	
	Solicitar Aprobación de Cotización	Administrador Gestión de Versiones
	Recibir Respuesta de Aprobación	
	Comprar Versión	Proveedores Externos
	Recibir Versión	
	Notificar Motivo Finalización	Administrador Proceso Gestión de Versiones
	Solicitar Plan de Pruebas Unitarias	Equipo de Desarrollo

	Recibir Plan de Pruebas Unitarias	
	Realizar Pruebas Unitarias	
	Recibir Resultados Pruebas	
	Solicitar Aprobación Resultados Pruebas Unitarias	Administrador Gestión de Versiones
	Recibir Respuesta Aprobación	
	Notificar Motivo Finalización	Administrador Proceso Gestión de Versiones
Pruebas	Solicitar Plan de Pruebas Integrales	Administrador Proceso Gestión de Versiones
	Recibir Plan de Pruebas	
	Realizar Pruebas Integrales	Equipo de Pruebas
	Recibir Resultados Pruebas	
	Solicitar Plan de Pruebas de Aceptación de Usuario	Administrador Proceso Gestión de Versiones
	Recibir Plan de Pruebas de Aceptación	
	Realizar Pruebas de Aceptación de Usuario	Grupo de Usuarios Finales
	Recibir Resultados Pruebas AU	
	Solicitar Aprobación de Resultados de Pruebas	Administrador Gestión de Versiones
	Recibir Respuesta Aprobación	
Notificar Motivo Finalización	Administrador Proceso Gestión de Versiones	
Aceptación de Liberación	Solicitar Aceptación de Liberación	
	Recibir Respuesta de Aceptación	Gestión de Cambio
	Notificar Motivo Finalización	Administrador Proceso Gestión de Versiones
Planificar Implementación	Solicitar Plan de Implementación	Administrador Proceso Gestión de Versiones
	Recibir Plan de Implementación	
	Realizar Implementación	Equipo de Desarrollo
	Recibir Resultados Implementación	
Comunicación y Adiestramiento	Solicitar Plan de Comunicación y Adiestramiento	Administrador Proceso Gestión de Versiones
	Recibir Plan de Comunicación y Adiestramiento	
Distribución e Instalación	Solicitar Plan de Distribución e Instalación	Administrador Proceso Gestión de Versiones
	Recibir Plan de Distribución e Instalación	
	Solicitar Aprobación de Cambios	Gestión de Cambios
	Recibir Respuesta Aprobación	
	Notificar Cambios Aprobados	Administrador Gestión de Versiones
	Realizar Back Out	Equipo de Desarrollo
	Recibir Resultados Back Out	
	Notificar Back Out Realizado	Administrador Gestión de Versiones
Notificar BD Actualizada		

Tabla 13: Actividades modeladas para Proceso de Gestión de Versiones y Liberaciones.

Interacciones

A medida que se modela el proceso, se van creando interacciones entrantes a los pools que representan servicios provistos por el proceso e interacciones salientes de los pools que son servicios consumidos por el proceso. Indiferentemente de la dirección de estas interacciones se generan variables que permiten almacenar la información que se intercambian en ellas.

El proceso de Gestión de Versiones y Liberaciones posee numerosas interacciones, a continuación se explicará una de cada tipo.

Interacciones con humanos utilizadas:

Proceso iniciado por humano: el proceso de Gestión de Versiones es iniciado por Gestión de Cambio por medio del suministro del RFC.

En este tipo de interacción se crean las variables: rFCInitProcessRequestMsg y rFCInitProcessResponseMsg

Actividad realizada por un humano: se envía información al Equipo de Desarrollo y se recibe la información suministrada por ellos. (Ver Figura 40).

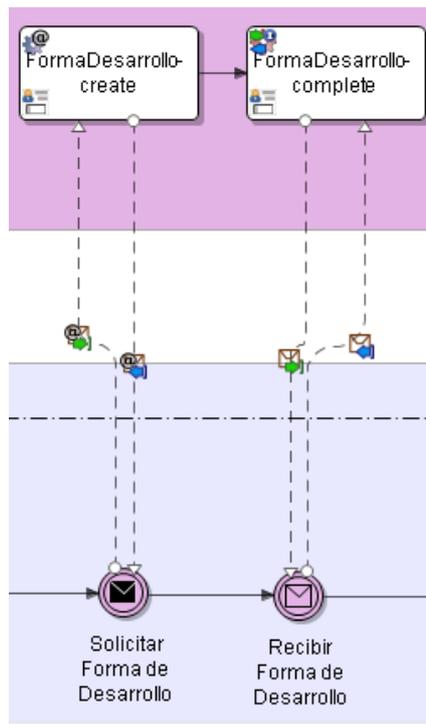


Figura 40: Actividad realizada por humano.

Esta interacción crea cuatro variables:

formaDesarrolloCreateTaskRequestMsg
 formaDesarrolloCreateTaskResponseMsg
 formaDesarrolloNotifyTaskCompletionRequestMsg
 formaDesarrolloNotifyTaskCompletionResponseMsg

Notificación: se utiliza para enviar información al Administrador Proceso Gestión de Versiones sobre el motivo de la finalización del proceso. Una notificación crea las variables: `notificarFin31NotifyRequestMsg` y `notificarFin31NotifyResponseMsg`.

La autorización de cada una de las interacciones se realizó en tiempo de diseño mediante la pestaña *Properties*. (Ver Figura 41).



Figura 41: Autorización de la interacción de Gestión de Cambios.

Los roles de las interacciones se encuentran definidas en el archivo `security.xml`.

El proceso de Gestión de Versiones posee muchas interacciones con humanos, cada interacción implica que se requiere de la creación de un formulario a través de XForms.

Mapeo de datos

Es utilizado para manipular los datos del proceso de Gestión de Versiones, el *Mapper* está ubicado en una de las pestañas inferiores de Intalio.

En la vista *Mapper* es donde realmente se realizan las asignaciones; consta de tres secciones que pueden verse como columnas: la primera columna de la izquierda define el origen de datos disponible para el mapeo y la lista de todas las variables de datos del proceso; la segunda columna ubicada en el medio, es un área de trabajo que permite el uso de operadores y realizar transformaciones a los datos de origen antes de mapearlos a los datos de destino y la tercera columna de la derecha define los datos de destino para el mapeo y también tiene una lista de todas las variables del proceso.

La siguiente Figura muestra la vista *Mapper* de Intalio|*Designer*.

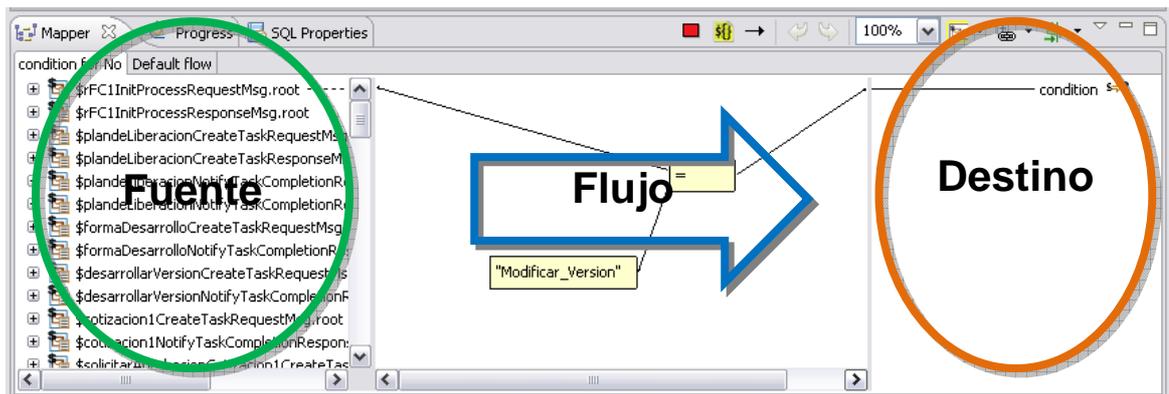


Figura 42: Vista del Mapper.

El proceso de Gestión de Versiones posee numerosos mapeos, a continuación se explicará cómo se realizó el del elemento de decisión ¿Nueva Versión? para establecer la ruta de la condición NO.

Las comparaciones requieren de dos entradas, la primera proviene del formulario de inicio y la segunda será creada utilizando el elemento operador y en él se coloca “Modificar_Versión” una vez que se tienen los elementos a comparar se crea otro operador con un “=” y finalmente se realiza la conexión, para ello se busca entre los elementos listados en el lado izquierdo del *mapper* rFC1InitProcessRequestMsg, buscamos el componente *task output* el cual contiene la información recibida por el sistema y luego se busca el combo box Accion_campo, el mapeo final se muestra en la Figura 43.

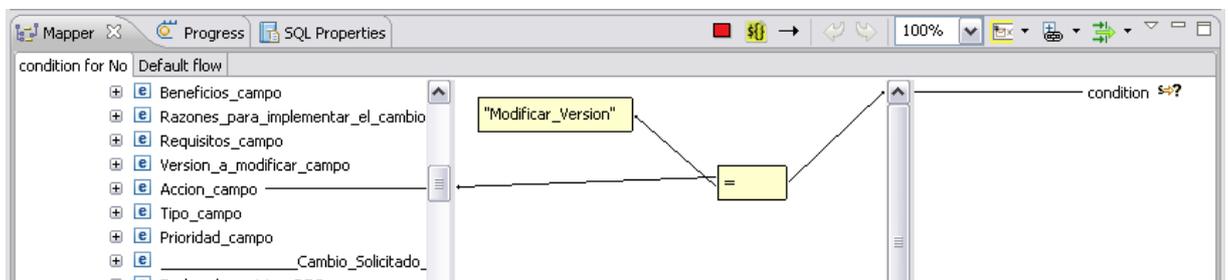


Figura 43: Mapeo de la decisión nueva versión.

- Simular prueba funcional: se realiza para validar las funcionalidades del sistema: navegación, entrada de datos, procesamiento y obtención de resultados.
- Integrar arquitectura: una vez que el cliente acepta el diagrama del proceso de Gestión de Versiones y Liberaciones, se debe proceder a integrar el proceso de negocio a la arquitectura de los procesos de negocio de CANTV.
- Simular prueba integral: en ella se valida la interrelación entre los grupos que representan cada etapa del proceso de Gestión de Versiones y Liberaciones, verificando que la información enviada entre ellas es correcta.

Configurar

En ella se realizan las siguientes actividades:

- Completar UML: se toman los diagramas UML realizados anteriormente para compararlos con el diagrama del proceso de Gestión de Versiones y Liberaciones con la finalidad comprobar que las funcionalidades reflejadas en ambas coincidan.
- Construir Lógica de Negocios: en este paso se construye la base de datos con PostgreSQL 8.3.10 y el servicio web mediante Mule *Studio* (Beta milestone 2) 1.0.0.

En los Anexos 8 y 9 se encuentra información detallada sobre PostgreSQL y Mule respectivamente.

Base de datos

El diseño de la base de datos se puede observar en la Figura 44.

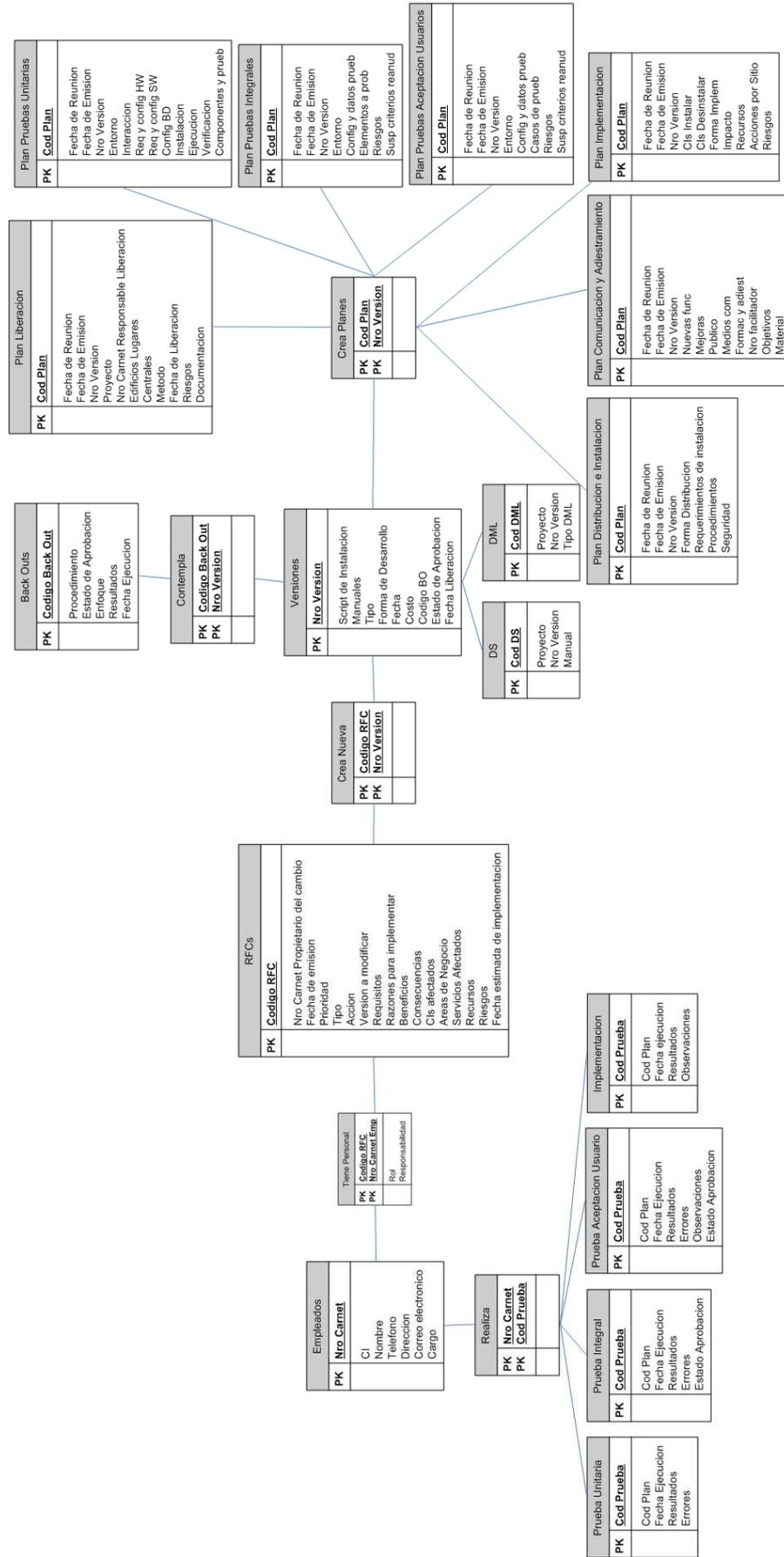


Figura 44: Base de Datos Gestión de Versiones.

Una vez que se tiene el diseño de la BD, se procede a crear las tablas de la BD en PostgreSQL 8.3.10. (Ver Figura 45).

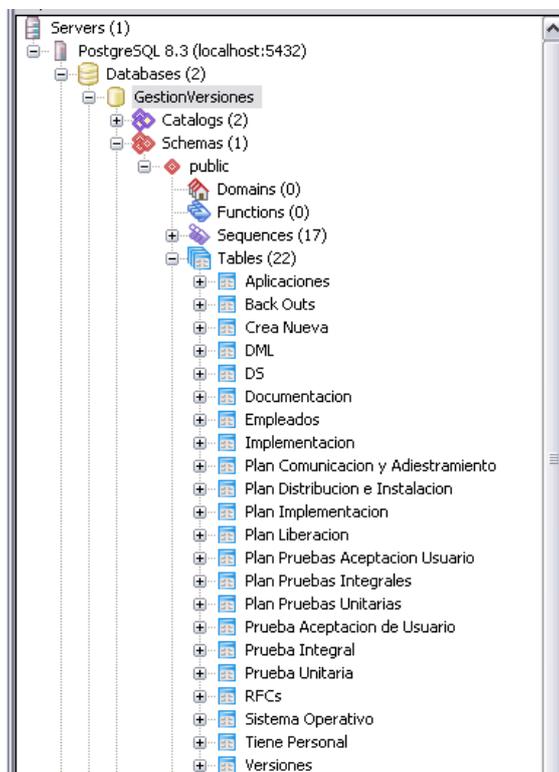


Figura 45: BD del Proceso de Gestión de Versiones en PostgreSQL 8.3.10.

Configuración del conector de la base de datos

Es necesario contar con el driver de conexión correspondiente a PostgreSQL 8.3.10 el cual es un archivo .jar llamado: postgresql-8.3-606.jdbc4 y añadirlo a Intalio|Server ingresando a la carpeta: intalio-bpms-6.0.3.010.01\common\lib.

En Intalio|Designer se activa la perspectiva *Database Development* y para crear un nuevo conector de base de datos, se selecciona PostgreSQL y se le coloca el nombre del perfil: ConectorPostgreSQL. Se selecciona el driver de conexión que fue colocado previamente en la carpeta del servidor y se elimina el driver que tiene por defecto. Seguidamente se especifica la base de datos a utilizar: GestionVersiones, el nombre de usuario: AdministradorBD y la

contraseña: Administrador; lo cual permite a Intalio|*Designer* tener acceso y realizar modificaciones sobre la BD.

Luego se hace clic en *test connection* y si muestra el mensaje *pin succeeded* entonces el conector se ha creado correctamente y en la pantalla se puede observar la vista que muestra la Figura 46.

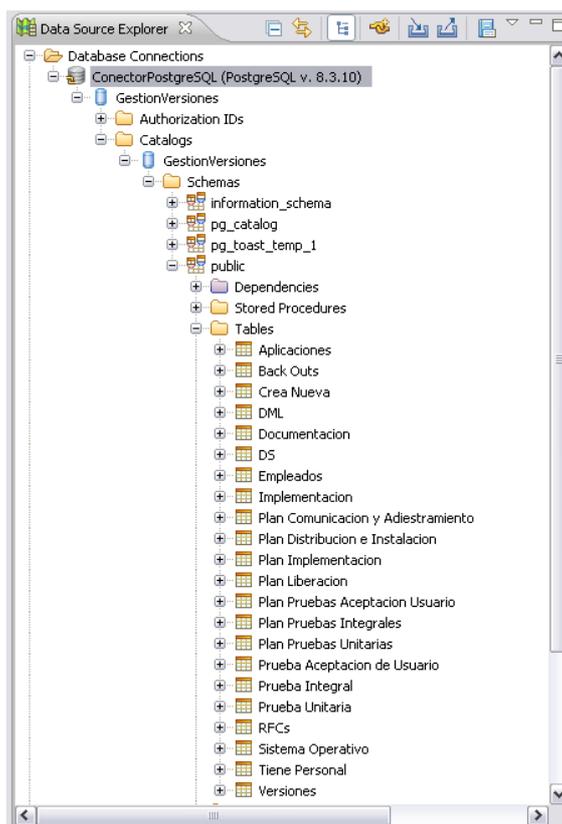


Figura 46: Vista de la BD del Proceso de Gestión de Versiones desde Intalio|*Designer*.

Crear consultas desde el proceso

Se activa la perspectiva Intalio|*Designer*, dentro de la carpeta Consultas se hace clic derecho en *New* → *Database Connector* y se coloca el nombre de la consulta y se cambia al perfil de conexión creado anteriormente.

Se procederá a explicar la creación de la consulta insertarRFC.

Para ello se hace clic en *New* → *Database Connector* y se coloca el nombre de la consulta, en este caso: insertarRFC.sql y se usa el perfil

de conexión: ConectorPostgreSQL. Se procede a escribir la consulta, la cual queda como muestra la Figura 47.

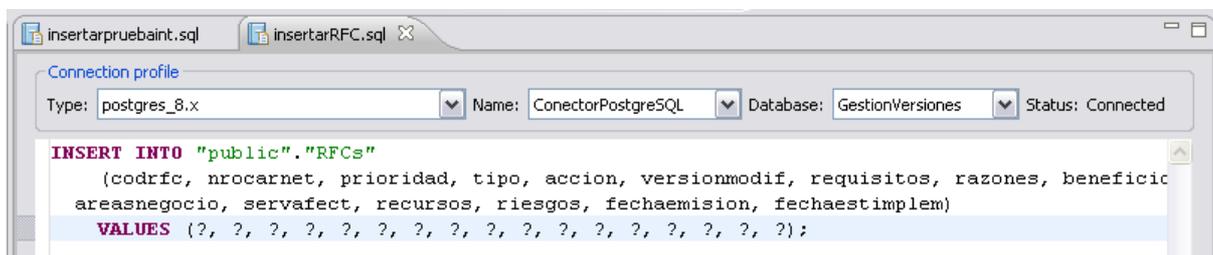


Figura 47: Consulta insertarRFC.sql.

Los signos de interrogación (?) en la expresión SQL (*Structured Query Language*) representan el valor de una variable, dicho valor dependerá del dato ingresado por el usuario.

Luego de escribir la consulta se puede probar haciendo clic derecho sobre el archivo .sql y se selecciona la opción ejecutar.

Para modificar las propiedades del SQL se utiliza la vista *SQL Properties*. (Ver Figura 48).

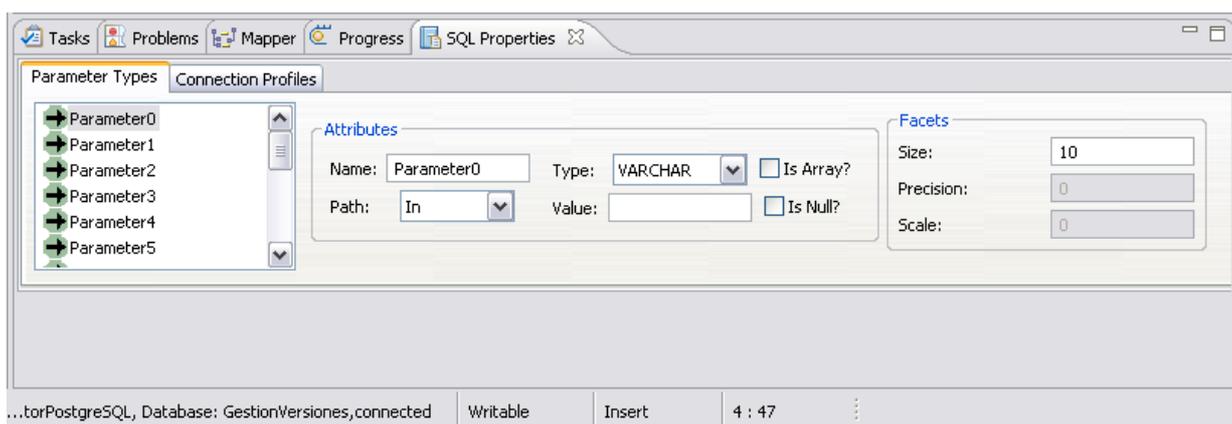


Figura 48: *SQL Properties* de la consulta insertarRFC.sql.

Cada signo de interrogación (?) contenido en una expresión SQL corresponde a una variable *Parameter<número>* creada por defecto, dicho parámetro puede ser renombrado, para la consulta insertarRFC se

listan los parámetros creados en Intalio asociados a cada campo de la tabla RFC creada en PostgreSQL y su tipo de dato en la Tabla 14.

PostgreSQL		Intalio	
Columna	Tipo de dato	Campo	Tipo de dato
Codrfc	Text	Parameter0	Varchar
Nrocarnet	Text	Parameter1	Varchar
Prioridad	Text	Parameter2	Varchar
Tipo	Text	Parameter3	Varchar
Acción	Text	Parameter4	Varchar
Versionmodif	Text	Parameter5	Varchar
Requisitos	Text	Parameter6	Varchar
Razones	Text	Parameter7	Varchar
Beneficios	Text	Parameter8	Varchar
Consecuencias	Text	Parameter9	Varchar
Cisafect	Text	Parameter10	Varchar
Areasnegocio	Text	Parameter11	Varchar
Servafect	Text	Parameter12	Varchar
Recursos	Text	Parameter13	Varchar
Riesgos	Text	Parameter14	Varchar
Fechaemision	Date	Parameter15	Date
Fechaestimplem	Date	Parameter16	Date

Tabla 14: Analogía entre los campos de la tabla RFC de PostgreSQL y los campos parameter

Al terminar de realizar el SQL se agrega como un servicio al pool BD.

Finalmente se realiza el mapeo ya que la consulta es para insertar los datos ingresados por el usuario en la base de datos, para ello se busca entre los elementos listados en el lado izquierdo del *mapper* rFC1InitProcessRequestMsg, luego se busca el componente *task output* el cual contiene la información recibida por el sistema. En el lado derecho está nsInsertarRFCRequestMsg el cual contiene información necesaria para la ejecución de la consulta. Una vez ubicados los elementos en ambos paneles se realiza el mapeo tal como se muestra en la Figura 49.

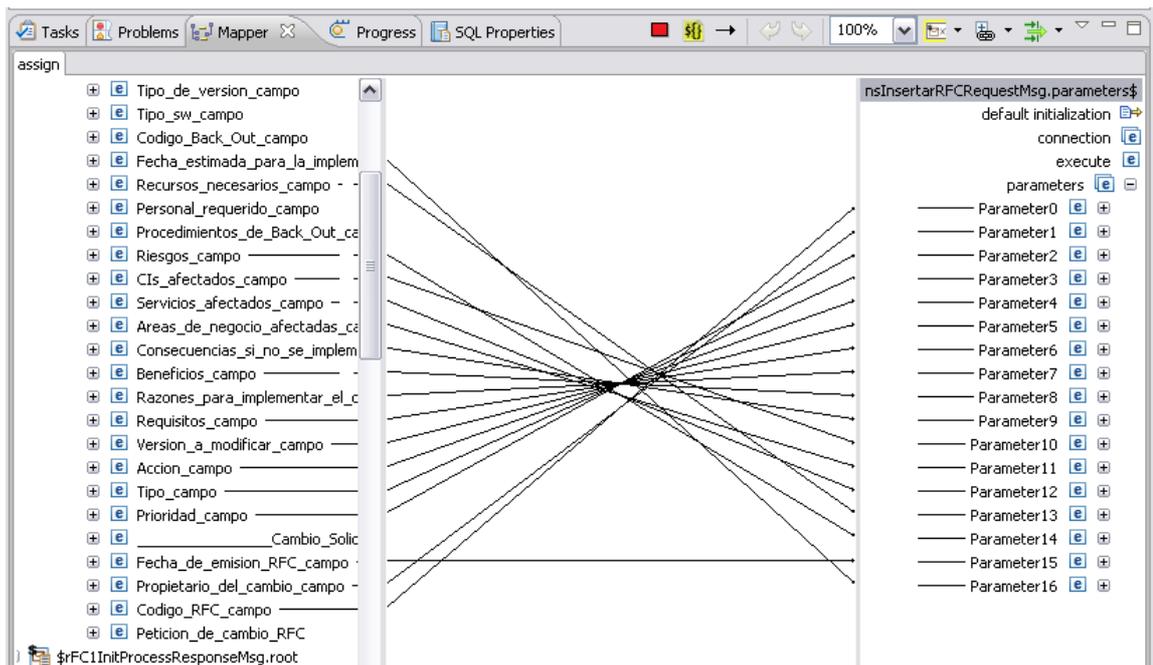


Figura 49: Mapeo de la consulta insertarRFC.

El proceso de creación de consultas descrito anteriormente se utiliza para crear el resto de las consultas del proceso de Gestión de Versiones, en la Figura 50 se pueden observar las consultas creadas.

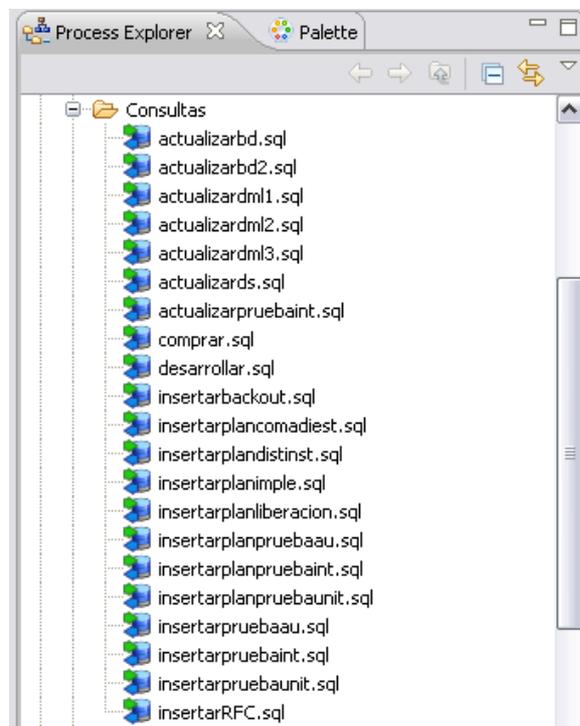


Figura 50: Consultas del proceso de Gestión de Versiones

Servicio Web

El proceso de Gestión de Versiones y Liberaciones utiliza un WS encargado de numerar las versiones tomando en cuenta el tipo de versión (mayor, menor o emergencia).

A continuación se explicaran los pasos para la creación del WS usando Mule *Studio* (Betamilestone 2) 1.0.0. el cual provee un ambiente de desarrollo sencillo y potente basado en Eclipse, permite crear fácilmente flujos, editar y configurar Mule ESB (*Enterprise Service Bus*).

Se crea un nuevo proyecto ya que en Mule los procesos son activados a partir de flujos que describen su comportamiento para ello se hace clic en *New*→*Mule Project* y se dejan los valores de configuración por defecto.

El servicio web debe numerar la nueva versión y recibe dos parámetros de entrada de tipo *string*: el tipo de versión y la versión actual.

Se crea la clase que contendrá el siguiente código:

```
@WebResult(name="version")
public String version(@WebParam(name="tipo") String
tipo,@WebParam(name="actual") String actual)
{
    String vfinal=null;
    String token1, token2, token3 =null;
    //Integer tres = 3;
    if(tipo.equals("Mayor"))
    {
        StringTokenizer token = new StringTokenizer
(actual, ".");
        token1 = token.nextToken();
        token2 = token.nextToken();
        int act = Integer.parseInt(token1);
        act++;
        vfinal =String.valueOf(act)+".0";
        return vfinal;
    }

    else if(tipo.equals("Menor"))
    {
```

```

        StringTokenizer token = new StringTokenizer
(actual, ".");
        token1 = token.nextToken();
        token2 = token.nextToken();
        int act = Integer.parseInt(token2);
        act++;
        vfinal =token1 + "." + String.valueOf(act);
        return vfinal;
    }

    else if(tipo.equals("Emergencia"))
    {
        StringTokenizer token = new StringTokenizer
(actual, ".");
        int total = token.countTokens();

        if (total==2)
        {
            token1 = token.nextToken();
            token2 = token.nextToken();
            int act = Integer.parseInt(token2);
            act++;
            vfinal =token1 + "." + token2 + ".1";
            return vfinal;
        }
        if (total==3)
        {
            token1 = token.nextToken();
            token2 = token.nextToken();
            token3 = token.nextToken();
            int act = Integer.parseInt(token3);
            act++;
            vfinal =token1 + "." + token2 + "." +
String.valueOf(act);
            return vfinal;
        }
    }

    return null;
}

```

Ahora se procede a crear el flujo en Mule que se ejecutará y enviará la respuesta: *New*→*Mule Message Flow*.

Se agrega un punto de entrada al servicio tipo *Generic* el cual se encuentra en la carpeta *Endpoints*, el elemento *Generic* permite realizar una configuración dinámica haciendo uso de expresiones de Mule.

Se agrega el WS al flujo, para ello se busca en la carpeta *Components* el elemento SOAP, se configura indicando la clase que implementa el WS. El elemento SOAP hace que un WS esté disponible a través de Apache CXF, el cual es un *framework* de servicios de código abierto. CXF ayuda a construir y desarrollar servicios utilizando JAX-WS como API de programación. Estos servicios pueden hablar una gran variedad de protocolos como SOAP, XML/HTTP, HTTP RESTful, o CORBA, y pueden trabajar sobre transportes como HTTP, JMS o JBI.

Finalmente se invoca la clase que implementa el servicio web; se busca en la carpeta *Components* y se agrega el elemento *Component*. El flujo creado se muestra en la Figura 51.

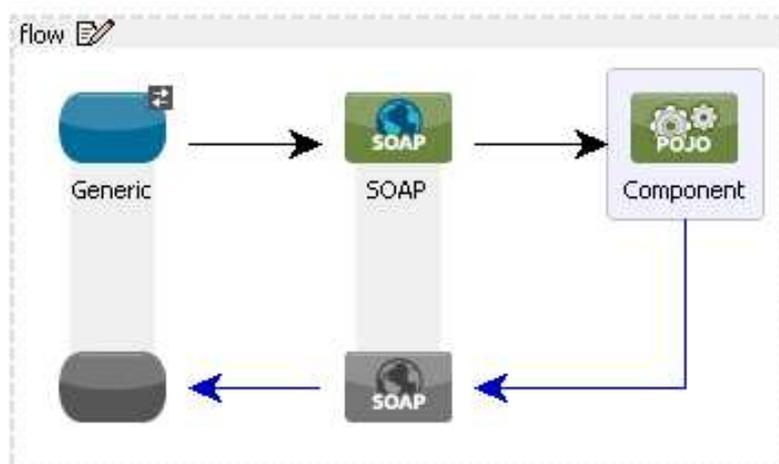


Figura 51: Flujo del WS setVersion.

Para probar el flujo se escribe en la barra de direcciones del navegador:

<http://localhost:65082/services/EchoUMO/version/tipo/Mayor/actual/2.0>

Se abre Intalio|*Designer* y en la carpeta del proceso de Gestión de Versiones se crea una nueva carpeta llamada Servicios, sobre ella se hace clic derecho: *New* → *Other* → *Web Services* → *WSDL* y se colocará el wsdl del WS a consumir.

Luego se busca el WS setVersion y se arrastra al pool Servicios agregándolo como *Single task*.

Ahora hay que buscar los datos que se enviarán como entrada a `setVersion`: tipo de versión y la versión actual; luego hay que enviar la data de respuesta generada por el WS al campo numeración de versión, esto se realiza utilizando el *mapper* como se muestra en las Figuras 52 y 53.

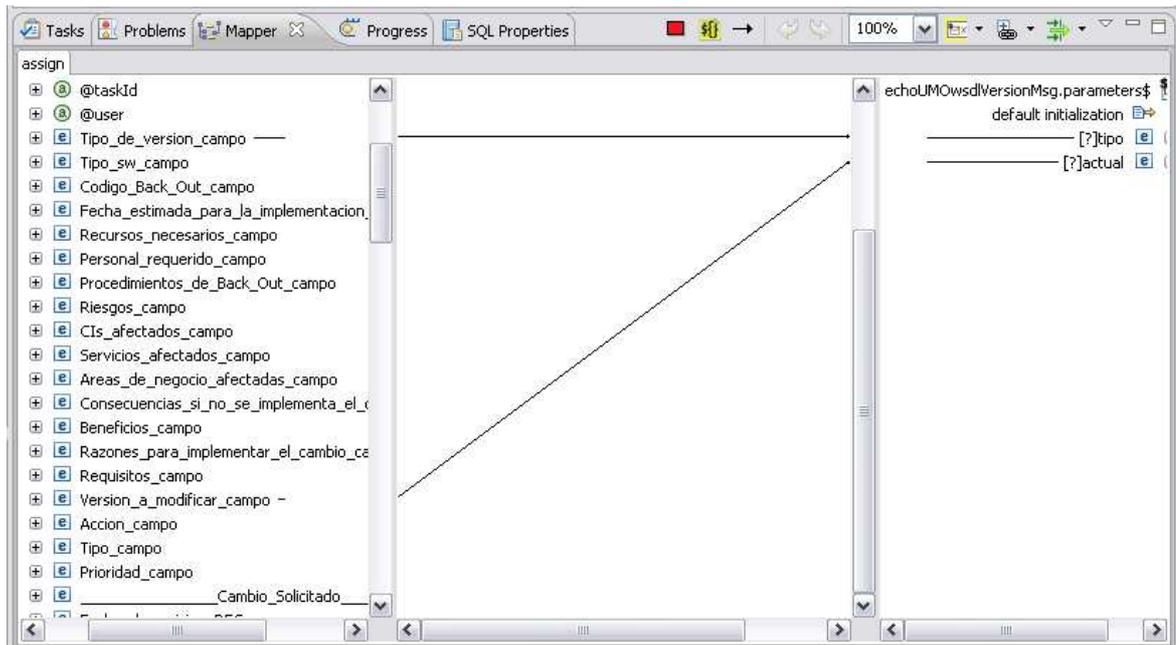


Figura 52: Mapeo de datos de entrada de `setVersion`.

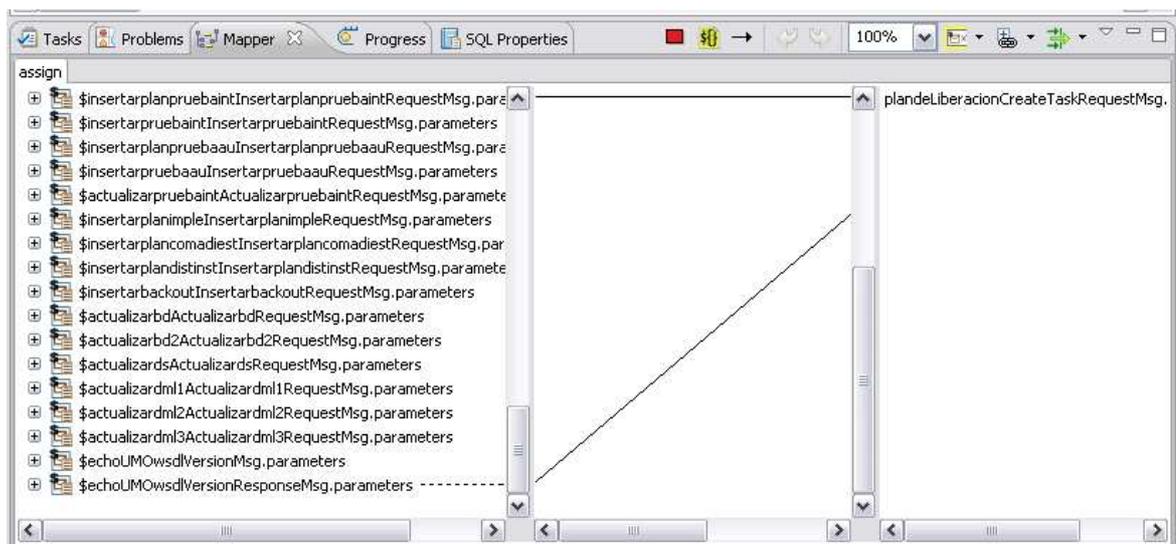


Figura 53: Mapeo del dato de salida de `setVersion`.

- Construir Interfaz:

Formularios

XForms es un formato XML que permite definir interfaces de usuario, principalmente formularios web, describir cualquier interfaz de usuario, realizar tareas simples y comunes de manipulación de datos.

Los patrones de formularios pueden ser de tipo: *init*, *create and complete* o *notify*. El proceso de Gestión de Versiones tiene un formulario del tipo *init*, veintiuno del tipo *create and complete* y siete del tipo *notify*.

Para la creación de formularios es necesario abrir la perspectiva Intalio|*Designer Form Editor* que contiene la paleta *Workflow Designer*. (Ver Figura 54).

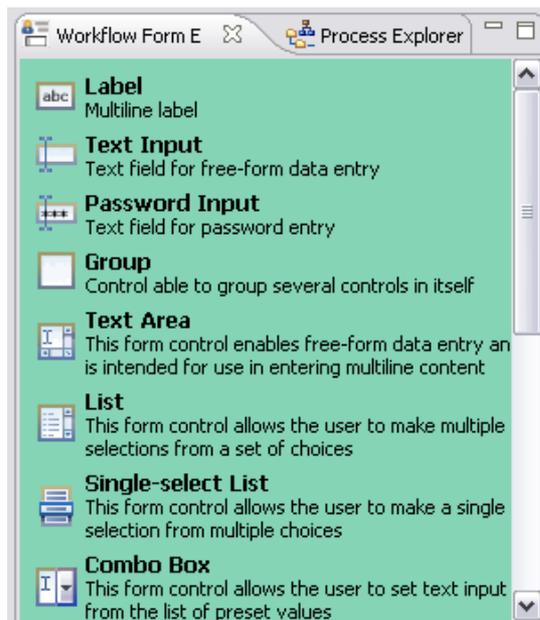


Figura 54: Paleta *workflow form*.

Para el proceso de Gestión de Versiones, se creó una carpeta llamada Formularios y en ella creamos un nuevo *Workflow Form* y se le coloca el nombre.

Durante la creación de los formularios se utilizaron 4 elementos de la paleta: *label*, *text input*, *text area* y *combo box*.

Los elementos de la paleta *workflow form* tienen propiedades comunes, tales como:

- Básicas:
 - *Control Label*: contiene la información que será mostrada al usuario.
 - *Name*: nombre de la variable que contiene la información recolectada o mostrada al usuario.
- Comportamiento:
 - *Input/Output*: indica si el elemento va a ser usado para mostrar, recolectar información o ambas.
- Ayuda:
 - *Tooltip*: permite mostrar un mensaje de ayuda al usuario cuando pase el ratón sobre el campo.
- Validación:
 - *Schema type*: se refiere al tipo de dato que soporta el elemento.

Los combo box cuentan con la propiedad *Item*, la cual es utilizada para agregar elementos que serán mostrados dentro de él.

A continuación se explicará cómo se realizó la creación de cada tipo de formulario.

Formulario init

Corresponde al RFC que Gestión de Cambio emite a Gestión de Versiones indicando los requerimientos. El formulario consta de 6 campos *text input*, 11 *text area* y 5 *combo box*. (Ver Figura 55).

Figura 55: Formulario init RFC.

Las propiedades de los campos de tipo *text input* se muestran en la Tabla 15.

Control label	Input/Output	Schema type
Codigo RFC:	In-Out	String
Propietario del cambio:	In-Out	String
Fecha de emisión:	In-Out	Date
Versión actual:	In-Out	String
Codigo Back Out:	In-Out	String
Fecha estimada para la implementación:	In-Out	Date

Tabla 15: Propiedades de los *text input* del formulario init RFC.

Las propiedades de los campos de tipo *combo box* se pueden apreciar en la Tabla 16.

Control label	Input/Output	Tooltip	Schema type	Item	
				Label	Value
Prioridad:	In-Out	-	String	Alta Media Baja	Alta Media Baja
Tipo:	In-Out	-	String	Software Hardware	Software Hardware
Tipo SW:	In-Out	Indique el tipo de software a tratar	String	Aplicación Sistema Operativo Documentación	Aplicación Sistema_Operativo Documentación
Acción:	In-Out	-	String	Modificar Versión Nueva Versión	Modificar_Versión Nueva_Versión
Tipo de versión:	In-Out	-	String	Mayor Menor Emergencia	Mayor Menor Emergencia

Tabla 16: Propiedades de los *combo box* del formulario init RFC.

Las propiedades de los campos *text area* se muestran en la Tabla 17.

Control label	Input/Output	Schema type
Requisitos:	In-Out	String
Razones para implementar el cambio:	In-Out	String
Beneficios:	In-Out	String
Consecuencias si no se implementa el cambio:	In-Out	String
Cls afectados:	In-Out	String
Areas de negocios afectados:	In-Out	String
Servicios afectados:	In-Out	String
Procedimientos de Back Out	In-Out	String
Recursos necesarios:	In-Out	String
Personal requerido:	In-Out	String
Riesgos:	In-Out	String

Tabla 17: Propiedades de los *text area* del formulario init RFC

Al terminar de crear el formulario RFC hay que arrastrarlo hasta el *pool* Gestión de Cambios y seleccionar la opción *initProcess*.

Formulario *Create and Complete*

En este caso se describirá el formulario Plan de Liberación, como su nombre lo indica, permite la creación de un plan de liberación tomando en cuenta el RFC suministrado; previo al vaciado de datos en dicho formulario, es necesario realizar una reunión con las personas involucradas para acordar la elaboración del plan. La Figura 56 muestra el formulario del plan de liberación.

Figura 56: Formulario *create and complete* Plan de Liberación.

Las propiedades de los campos del formulario se pueden apreciar en las Tablas 18, 19, 20 y 21.

Control label	Name	Input/Output	Schema type
Codigo RFC:	Codigo_RFC	In-Out	String
	Codigo_RFC_campo	In-Out	String
Prioridad:	Prioridad	In-Out	String
	Prioridad_campo	In-Out	String
Tipo:	Tipo	In-Out	String
	Tipo_campo	In-Out	String

Accion:	Accion	In-Out	String
	Accion_campo	In-Out	String
Version actual:	Version_actual	In-Out	String
	Version_actual_campo	In-Out	String

Tabla 18: Propiedades de los *labels* del formulario Planificar Liberación.

Control label	Input/Output	Schema type
Fecha de reunión:	In-Out	Date
Fecha de emisión:	In-Out	Date
Proyecto:	In-Out	String
Numeración de la versión:	In-Out	String
Propietario del cambio:	In-Out	String
Teléfono:	In-Out	Integer
Responsable de la liberación:	In-Out	String
Teléfono:	In-Out	Integer
Método de liberación:	In-Out	String
Fecha estimada para la liberación:	In-Out	Date

Tabla 19: Propiedades de los *text input* del formulario Planificar Liberación.

Control label	Input/Output	Schema type
Requisitos:	In-Out	String
Edificios/Lugares:	In-Out	String
Roles y responsabilidades:	In-Out	String
Centrales:	In-Out	String
Riesgos:	In-Out	String
Documentación necesaria:	In-Out	String

Tabla 20: Propiedades de los *text area* del formulario Planificar Liberación.

Control label	Input/Output	Schema type	Item		Read-only
			Label	Value	
Tipo de versión:	In-Out	String	Mayor Menor Emergencia	Mayor Menor Emergencia	True()

Tabla 21: Propiedades de los *combo box* del formulario Planificar Liberación.

Al terminar el formulario Planificar Liberación, se debe arrastrar al *pool* Administrador Proceso Gestión de Versiones y antes de soltarlo hay que seleccionar la opción *create and complete*.

Formulario *Notify*

El proceso de Gestión de Versiones utiliza formularios *notify* para indicar cuál fue el motivo de finalización del proceso.

Las propiedades del Formulario NotificarFin2 se pueden apreciar en la Tabla 22.

Control label	Name	Input/Output	Schema type
Los Resultados de las Pruebas Unitarias de la versión	Los_Resultados_de_las_Pruebas_Unitarias_de_la_version	In-Out	String
del proyecto	del_proyecto	In-Out	String
no fueron aprobados	no_fueron_aprobados	In-Out	String
	Numeracion_version_campo	In-Out	String
	Proyecto_campo	In-Out	String

Tabla 22: Propiedades de los *labels* del formulario NotificarFin2.

Cabe destacar que los *labels* que no contienen *control label* en las Tablas 17 y 21 van a ser usadas para mostrar datos provenientes de un mapeo.

Cuando el formulario NotificarFin2 esté listo hay que arrastrarlo al *pool* Administrador Proceso Gestión de Versiones y seleccionar la opción *notify* antes de soltarlo.

- Configurar servidor BPEL: se despliega el proceso en el servidor para poder ejecutarlo. Intalio usa como estándar BPMN para el diseñador, luego se convierte en código BPEL para correrlo desde Apache ODE.
- Realizar Pruebas Unitarias: en ella se prueba la funcionalidad de cada grupo o etapa del proceso de Gestión de Versiones y Liberaciones por separado, lo cual permite saber que cada uno de ellos se ejecuta

correctamente eliminando incertidumbres y así ayudar a la integración de los mismos.

Descripción del caso de prueba

1. El Gestor de Cambio debe crear un RFC.
2. Tomando en cuenta el RFC, Gestión de Versiones y Liberaciones construye el plan de liberación.
3. Se la versión será desarrollada en casa.
4. Se construye el plan de pruebas unitarias, se realizan las mismas y se aprueban los resultados.
5. Se construye el plan de pruebas integrales y se ejecuta el plan.
6. Se construye el plan de pruebas de aceptación de usuario, se ejecutan las pruebas y se solicita la aprobación de los resultados.
7. Se aprueba la liberación de la versión.
8. Se construye el plan de comunicación y adiestramiento; así como también el plan de distribución e instalación.
9. Los cambios no son aprobados, se debe realizar la marcha atrás.

Puntos de Chequeo

1. Llenado de los campos correctamente.
2. Validar que todos los combos estén relacionados correctamente.
3. Verificar que los planes se hayan almacenado.
4. Verificar que la versión se almacene satisfactoriamente.

A continuación se muestran los resultados de la ejecución del caso de prueba descrito anteriormente.

1. Se llena el RFC por parte del Gestor de Cambios.

Petición de cambio (RFC)

Codigo RFC: Propietario del cambio: Fecha de emision:

Cambio Solicitado

Prioridad: Tipo: Tipo SW:

Accion: Tipo de version: Version actual:

Requisitos:

Razones para implementar el cambio:

Beneficios:

Consecuencias si no se implementa el cambio:

Cls afectados:

Areas de negocio afectadas:

Codigo Back Out:

Figura 57: Formulario Petición de Cambio.



Una vez llenado el RFC hay que verificar que ha sido almacenado en la BD.

Edit Data - PostgreSQL 8.3 (localhost:5432) - GestionVersiones - RFCs

	codrfc text	nrocarnet text	prioridad text	tipo text	accion text	versionmodif text	requisitos text	razones text	beneficios text	consecuencia text	cisafect text	areasnegocio text	servafect text	recursos text	riesgos text	fe d	
1	COD1	Luis Perez	Alta	Software	Modificar_Versi	5.3	se requiere de l	la version actual	se obtendra un	una vez que el	la version actual	IVA	interrupcion del	servidores para	interrupcion del	20	
2	COD2	Alberto Mejia	Alta	Software	Nueva_Version	7.3.1	Se requiere que	Existen numero	Al descargar la	los usuarios seg	version 7.3.1		interrupcion del	servicio que pre	laboratorio	interrupcion del	20
3	cod139																

Figura 58: Tabla RFCs.

	codrfc text	nrocarnet text	prioridad text	tipo text
1	COD1	Luis Perez	Alta	Software
2	COD2	Alberto Mejia	Alta	Software

2. Se crea el Plan de Liberación.

Plan de Liberación

Fecha de reunion: 9/9/2011
Fecha de emision: 9/12/2011

Codigo RFC: COD2

Prioridad: Alta Tipo: Software Accion: Nueva_Version Version actual: 7.3.1

Requisitos: Se requiere que la nueva version contenga todas las modificaciones de las otras versiones

Informacion del Item de Liberacion

Proyecto: Actualización de SW 7.3.1 Tipo de version: Mayor Numeracion de la version: 8.0

Personas Involucradas Grupos Afectados por la Liberacion

Propietario del cambio: Alberto Mejia Telefono: Edificios/Lugares:

Responsable de la liberacion: Telefono: Centrales:

Figura 59: Formulario Plan de Liberación.

Se verifica que el plan a crear contiene el código del RFC creado anteriormente.

INTALIO

Tasks Notifications Processes

Case sensitive

Plan de Liberacion

Codigo RFC: COD2

Prioridad: Alta Tipo: Software Accion: Nueva_Vers

Requisitos: Se requiere que la nueva version contenga todas las modificaciones de las otras versiones

También hay que comprobar que el plan ha sido almacenado correctamente en la BD.

	fecha reunion date	fecha emision date	nro version text	proyecto text	nro carnet text	ediflug text	centrales text	metodo text	fecha lib date	riesgos text	documentacion text	cod plan integer
1	2011-09-07	2011-09-07	6.0	Actualización de Maria Pereira	Sede principal	Boleita			2011-10-31	interrupción de manuales y cual	73	
2	2011-09-09	2011-09-12	8.0	Actualización de Maria Alcalá	Cortijos	Boleita			2011-11-30	interrupción de manuales	74	
3	2011-09-28	2011-09-29	11.0	proy					2011-09-03		75	

Figura 60: Tabla Plan Liberación.

	fecha reunion date	fecha emision date	nro version text	proyecto text	nro carnet text	ediflug text
1	2011-09-07	2011-09-07	6.0	Actualización de Maria Pereira	Sede principal	Sede principal
2	2011-09-09	2011-09-12	8.0	Actualización de Maria Alcalá	Cortijos	Cortijos
3	2011-09-28	2011-09-29	11.0	proy		

3. La versión se desarrolla en casa.

INTALIO

Tasks Notifications Processes

Case sensitive Filter

Forma de Desarrollo

Proyecto: Actualización de SW 7.3.1 Tipo: Software Version: 8.0

Requisitos: se requiere que la nueva version contenga todas las modificaciones de los errores conocidos

Indique la forma de desarrollo: En Casa

Figura 61: Formulario Solicitar forma de desarrollo.

Indique la forma de desarrollo:

En Casa

4. Construir el plan de pruebas unitarias.

Plan de Pruebas Unitarias

Codigo RFC: COD2 Proyecto: Actualización de SW 7.3.1 Version: 8.0

Fecha de reunion: 9/13/2011
Fecha de emision: 9/14/2011

Entorno: TI Interaccion con otros sistemas: NA

Requerimientos y configuracion de HW: NA Requerimientos y configuracion de SW: hay que obtener la version y proceder a instalarla

Configuracion de la BD: realizar un back up para la version anterior Instalacion: se realiza de manera manual

Ejecucion: se realiza de manera manual Verificacion: probar que la version realice todas las

Componentes y pruebas:

Figura 64: Formulario Plan de Pruebas Unitarias.

Comprobar que el plan creado se almacenó exitosamente en la BD.

id	fecha reunion date	fecha emision date	no version text	entorno text	interaccion text	req config hw text	req config sw text	config bd text	instalacion text	ejecucion text	verificacion text	comp pruebas text	cod plano integer
1	2011-09-08	2011-09-08	6.0	TI	NA	NA	se debe descargar	NA	instalar la versio	NA	correcto funcion	Probar el funcio	892
2	2011-09-13	2011-09-14	8.0	TI	NA	NA	hay que obtene	realizar un back	se realiza de ma	se realiza de ma	probar que la ve	probar la versio	893

Figura 65: Tabla Plan Pruebas Unitarias.

	fecha reunion date	fecha emision date	no version text	entorno text
1	2011-09-08	2011-09-08	6.0	TI
2	2011-09-13	2011-09-14	8.0	TI

Ejecutar el plan de pruebas unitarias.

Realizar Pruebas Unitarias

Fecha de ejecucion: 9/15/2011

Proyecto: Actualización de SW 7.3.1 Version: 8.0

Componentes y pruebas:

Verificacion:

Resultados:

Errores:

Figura 66: Formulario Realizar Pruebas Unitarias.

Almacenar los resultados de las pruebas unitarias en la BD.

	fechaejecucion date	resultados text	errores text	codpruebu integer	codplanu integer	edoaprobacio text
1	2011-09-08	las pruebas real	NA	1039	892	Si
2	2011-09-15	las pruebas real	0	1040	893	

Figura 67: Tabla Plan Pruebas Unitarias.

Solicitar la aprobación de los resultados de las pruebas unitarias.

Solicitar Aprobacion de Resultados de Pruebas Unitarias

Proyecto: Actualización de SW 7.3.1 Version: 8.0

Resultados:

Errores:

Aprobar Resultados de Pruebas Unitarias:

Fecha: 9/15/2011

Figura 68: Formulario Solicitar Aprobación de Resultados de Pruebas Unitarias.

Actualizar en BD la tabla de resultados de las pruebas unitarias.

	fechaejecuci date	resultados text	errores text	codpruebu integer	codplanu integer	edoaprobacio text
1	2011-09-08	las pruebas real	NA	1039	892	Si
2	2011-09-15	las pruebas real	0	1040	893	Si

Figura 69: Tabla Prueba Unitaria.

5. Crear el plan de pruebas integrales.

Plan de Pruebas Integrales

Fecha de reunion: 9/16/2011
Fecha de emision: 9/19/2011

Codigo RFC: COD2 Proyecto: Actualización de SW 7.3.1 Version: 8.0

Entorno: TI

Configuracion y datos de prueba: colocar link donde se encuentren los datos de prueba

Elementos a probar: colocar link con documento indicando pruebas a realizar

Riesgos: interrupcion del servicio

Suspension de pruebas/Criterios de reanudacion: susp: fallas electricas y/o de red reanud: servicio

Figura 70: Formulario Plan de Pruebas Integrales.

Almacenar el plan en la BD.

	fechareunion date	fechaemision date	nroversion text	entorno text	configdatospi text	elemprobar text	riesgos text	suspcrioterios text	codplanin integer
1	2011-09-09	2011-09-09	6.0	TI	configurar la ver	Probar version e	interrupcion del	susp de prueba:	45
2	2011-09-16	2011-09-19	8.0	TI	colocar link donc	colocar link con	interrupcion del	susp: fallas elec	46

Figura 71: Tabla Plan Pruebas Integrales.

Ejecutar el plan de pruebas integrales.

Realizar Pruebas Integrales

Fecha de ejecucion: 9/20/2011

Proyecto: Actualización de SW 7.3.1 Version: 8.0 Codigo Back Out: CODBO2

Entorno: TI Elementos a probar: colocar link con documento indicando pruebas a realizar Procedimientos de Back Out: reestablecer el servicio si la version creada no cumple con las expectativas

Resultados: resultados satisfactorios Errores: 0

Figura 72: Formulario Realizar Pruebas Integrales.

Almacenar en la BD los resultados de las pruebas.

	fechaejecucion date	resultados text	errores text	edoaprobacion text	codpruebin integer	codplanin integer
1	2011-09-12	los resultados de	NA	Si	47	45
2	2011-09-20	resultados satisf	0		48	46

Figura 73: Tabla Prueba Integral.

6. Crear el plan de pruebas de aceptación de usuario.

Plan de Pruebas de Aceptacion de Usuario

Fecha de reunion: 9/20/2011
Fecha de emision: 9/20/2011

Codigo RFC: COD2 Proyecto: Actualización de SW 7.3.1 Version: 8.0

Entorno: TI Configuración y datos de prueba: colocar link con datos de pruebas

Casos de prueba: colocar link con casos de prueba Riesgos: interrupción del servicio

Suspension de pruebas/Criterios de reanudacion: susp: fallas electricas y/o de red reanud: servicios

Figura 74: Formulario Pruebas de Aceptación de Usuario.

Almacenar en la BD el plan creado.

	fecha reunion date	fechaemision date	nroversion text	entorno text	configdatospi text	casosprueba text	riesgos text	suspcriterios text	codplana integer
1	2011-09-12	2011-09-12	6.0	TI	el equipo de des	verificar ingreso	interrupcion del	susp de pruebas:	22
2	2011-09-20	2011-09-20	8.0	TI	colocar link con	colocar link con	interrupción del	susp: fallas elec	23

Figura 75: Tabla Plan Pruebas Aceptación de Usuario.

Realizar las pruebas de aceptación de usuario.

Realizar Pruebas de Aceptación de Usuario

Fecha de ejecución: 9/21/2011

Proyecto: Actualización de SW 7.3.1 Version: 8.0

Entorno: TI Casos de prueba: colocar link con casos de prueba

Resultados: resultados satisfactorios Errores: 0 Observaciones: mejorar la interfaz

Figura 76: Formulario Realizar Pruebas de Aceptación de Usuario.

Almacenar los resultados de las pruebas en la BD.

	fechaejecucion date	resultados text	errores text	observacione text	edoaprobacio text	codprueba integer	codplana integer
1	2011-09-13	mejorar la interf	NA	NA	Si	66	22
2	2011-09-21	resultados satisfi	0	mejorar la interf		67	23

Figura 77: Tabla Prueba Aceptación de Usuario.

Solicitar la aprobación de los resultados.

Solicitar Aprobación de Resultados de Pruebas

Proyecto: Actualización de SW 7.3.1 Version: 8.0

Resultados de Pruebas Integrales

Resultados: resultados satisfactorios Errores: 0

Resultados de Pruebas de Aceptación de Usuario

Resultados: resultados satisfactorios Errores: 0

Aprobar Resultados de Pruebas: Si Fecha: 9/22/2011

Figura 78: Formulario Solicitar Aprobación de Resultados.

Actualizar en la BD el campo estado de aprobación de la tabla prueba integral.

	fechaejecucion date	resultados text	errores text	edoaprobacion text	codpruebin integer	codplanin integer
1	2011-09-12	los resultados d	NA	Si	47	45
2	2011-09-20	resultados satisf	0	Si	48	46

Figura 79: Actualizar tabla Prueba Integral.

Actualizar en la BD el campo estado de aprobación de la tabla prueba de aceptación de usuario.

	fechaejecucion date	resultados text	errores text	observacione text	edoaprobacion text	codprueba integer	codplana integer
1	2011-09-13	mejorar la interf	NA	NA	Si	66	22
2	2011-09-21	resultados satisf	0	mejorar la inter	Si	67	23

Figura 80: Actualizar tabla Prueba Aceptación de Usuario.

7. Aprobar la liberación de la versión.

Solicitar Aceptacion de Liberacion

Proyecto: Actualización de SW 7.3.1 Version: 8.0

Fecha: 9/23/2011

Aprobar Aceptacion de Liberacion:

Figura 81: Formulario Solicitar Aceptación de Liberación.

Actualizar en la BD el campo estado de aprobación de la tabla versiones.

	nroversion text	scriptinst text	manual text	tipo text	formadesarrc text	fecha date	costo real	codbo text	edoaprobacion text	fechalib date	codvers integer
1	6.0	link script	link manuales	Software	En_Casa	2011-09-08	3000	CODBO1	Si	2011-05-17	160
2	8.0	colocar link del s	colocar link de n	Software	En_Casa	2011-09-13	5000	CODBO2	Si	2011-05-17	161

Figura 82: Actualizar tabla Versiones.

8. Crear el plan de implementación.

Plan de Implementación

Fecha de reunion: 9/23/2011
Fecha de emision: 9/23/2011

Codigo RFC: CDD2 Proyecto: Actualización de SW 7.3.1 Version: 8.0

Cls afectados: version 7.3.1

Cls a instalar: link con cls a instalar

Cls a desinstalar: link con cls a desinstalar

Forma de implementacion: Automatico

Impacto:

Recursos necesarios: laboratorios y personal encargado

Acciones por sitio: manera automatica entonces en cada sitio se van a realizar las mismas actividades de instalacion

Riesgos: interrupcion del servicio

Figura 83: Formulario Plan de Implementación.

Almacenar en la BD el plan de implementación creado.

	fecha reunion date	fecha emision date	nro version text	cls instalar text	cls desinstalar text	forma implem text	impacto text	recursos text	acciones por sitio text	riesgos text	cod planim integer
1	2011-09-14	2011-09-14	6.0	version 6.0	version 5.3	Automatico	algunos usuario	equipo de desar	realizar instalaci	interrupcion del	29
2	2011-09-23	2011-09-23	8.0	link con cls a inst	link con cls a de	Automatico		laboratorios y pi	como la impleme	interrupcion del	30

Figura 84: Tabla Plan Implementación.

Realizar la implementación.

Realizar Implementacion

Fecha de ejecucion: 9/27/2011

Proyecto: Actualización de SW 7.3.1 Version: 8.0

Forma de implementacion: Automatico

Acciones por sitio: como la implementacion va a ser de manera automatica entonces en cada sitio se van a realizar las mismas

Resultados: implementacion exitosa

Observaciones: NA

Figura 85: Formulario Realizar Implementación.

Guardar los resultados de implementación en la BD.

	fechaejecucion date	resultados text	observaciones text	codim integer	codplanim integer
1	2011-09-14	la version fue im	NA	16	29
2	2011-09-27	implementacion	NA	17	30

Figura 86: Tabla Implementación.

9. Construir el plan de comunicación y adiestramiento.

Plan de Comunicación y Adiestramiento

Fecha de reunion: 9/27/2011
Fecha de emision: 9/27/2011

Codigo RFC: COD2 Proyecto: Actualización de SW 7.3.1 Version: 8.0

Eventos

Pruebas integrales: 9/20/2011 Liberacion de version: 9/23/2011

Pruebas aceptacion de usuario: 9/21/2011 Implementacion: 9/27/2011

Comunicacion

Nuevas funcionalidades: funcionamiento de permisos que los usuarios accedan a los datos facilmente por medio del buscador

Mejoras: la interfaz es un poco mas sencilla

Publico: personas involucradas en el negocio

Medios de comunicacion y actividades: actividades

Adiestramiento

Figura 87: Formulario Plan de Comunicación y Adiestramiento.

Almacenar el plan en la tabla Plan Comunicación y Adiestramiento de la BD.

	fecha reunion date	fecha emision date	nro version text	nuevas func text	mejoras text	publico text	medios com text	forma adiest text	nro facilitador integer	objetivos text	materiales text	cod plan com integer
1	2011-09-15	2011-09-15	6.0	presenta tooltip	mejora en la int	general	correo electroni	Curso_Presenci	5	impartir un curs	manual del usua	17
2	2011-09-27	2011-09-27	8.0	la version tiene	la interfaz es un	personas involu	actividades: che	Curso_en_Linea	2	presentar el nue	manuales de usu	18

Figura 88: Tabla Plan Comunicación y Adiestramiento.

Construir el plan de distribución e implementación.

Plan de Distribucion e Instalacion

Codigo RFC: C002 Proyecto: Actualización de SW 7.3.1 Version: 8.0

Fecha de reunion: 9/27/2011
Fecha de emision: 9/28/2011

Forma de distribucion: Paquete de Versiones

Requerimientos para la instalacion: NA

Procedimientos de distribucion e instalacion: NA

Seguridad: autentificarse para saber que usuario realizo [la actual]

Figura 89: Formulario Plan de Distribución e Instalación.

Almacenar el plan en la BD.

	fecha reunion date	fecha emision date	no version text	forma distribu text	req instalacio text	procedimient text	seguridad text	codpland integer
1	2011-09-16	2011-09-16	6.0	Version_Comple	NA	se realizara de r	autenticarse pai	15
2	2011-09-27	2011-09-28	8.0	Paquete_de_Ve	NA	NA	autenticarse pai	16

Figura 90: Tabla Plan Distribución e Instalación.

10. Solicitar la aprobación de los cambios, los cuales no son aceptados.

Solicitar Aprobacion de Cambios

Proyecto: Actualización de SW 7.3.1 Version: 8.0

Nuevas funcionalidades:

Mejoras:

Aprobar Cambios: No

Fecha: 9/28/2011

Figura 91: Formulario Solicitar Aprobación de Cambios.

Se realiza la marcha atrás.

Realizar Back Out

Fecha de ejecucion:
9/29/2011

Proyecto: Actualización de SW 7.3.1 Version: 8.0 Código Back Out: CODBO2

Procedimientos de Back Out: reestablecer el servicio si la version creada no cumple con las expectativas

Enfoque del back out: Reestablecer Servicio Resultados: |

Figura 92: Formulario Realizar Marcha Atrás.

Almacenar los resultados de la marcha atrás en la BD.

	codbo text	procedimient text	enfoque text	resultados text	fechaejecucio date
1	CODBO1	si la version may	Reestablecer_Si	back out realiza	2011-09-16
2	CODBO2	reestablecer el s	Reestablecer_Si		2011-09-29

Figura 93: Tabla Back Outs.

Notificar que se realizó una marcha atrás.

Tasks **Notifications** Processes Case sensitive Filter

Las cambios que introduce la version 8.0 del proyecto Actualización de SW 7.3.1 no fueron aprobados

Por lo tanto, se realizo un Back Out

Figura 94: Formulario Notificar Marcha Atrás realizada.

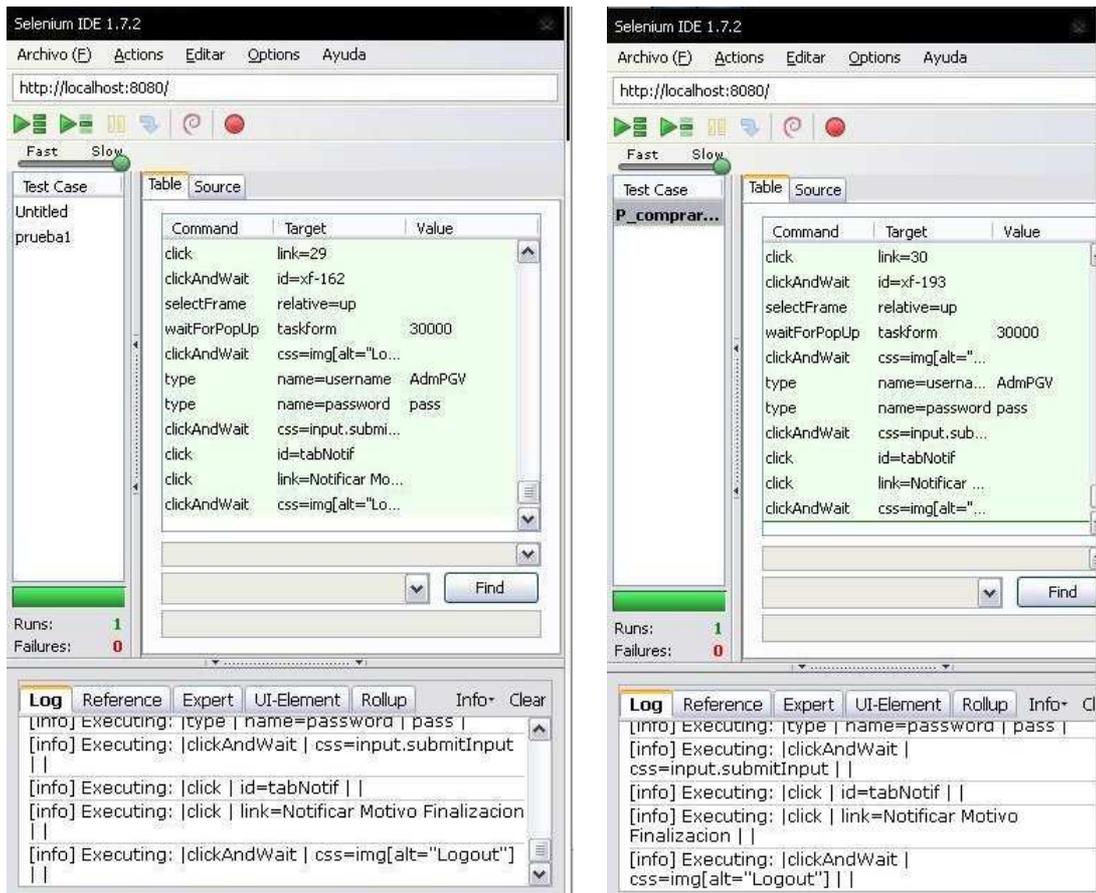
Notificar que se actualizaron las tablas de la BD.

Tasks **Notifications** Processes Case sensitive Filter

Los cambios han sido actualizadas en las BD correspondientes

Figura 95: Formulario Notificar BD actualizada.

Una vez establecidos los casos de prueba, se aplicaron utilizando Selenium IDE 1.7.2, el cual es un *plugin* de Mozilla Firefox que facilita la labor de realizar juegos de pruebas automatizadas para aplicaciones web. La Figura 96 muestra los resultados obtenidos en dos casos de pruebas.



Caso de prueba: cambios no aprobados

Caso de prueba: comprar versión

Figura 96: Resultado de los casos de pruebas.

En los Anexos 10 y 11 se encuentran el Manual Administrativo y el Manual de Usuario del Sistema de Información para la Gestión de Versiones y Liberaciones de Software.

Una vez que el sistema se encuentre en producción es necesario elaborar informes que permitan evaluar el rendimiento del proceso de Gestión de Versiones y Liberaciones.

Para que estos informes contengan información precisa es necesario elaborar métricas de referencia que cubran aspectos tales como (Comunidad Osatis, 2011):

- Número de lanzamientos de nuevas versiones
- Número de marcha atrás realizadas y razones de los mismos
- Incidencias asociadas a nuevas versiones
- Cumplimientos de los plazos previstos para cada despliegue
- Asignación de recursos en cada caso
- Corrección y alcance de la CMDB y la DS
- Existencia de versiones ilegales de software
- Adecuado registro de las nuevas versiones en la CMDB
- Incidencias provocadas por uso incorrecto (formación inadecuada) de la nueva versión por parte de los usuarios
- Disponibilidad del servicio durante y tras el proceso de lanzamiento de la nueva versión

3.3 Procedimiento de Evaluación

Para poder realizar la evaluación se hicieron dos encuestas, una antes de construir el sistema, la cual trata sobre ITIL y las funcionalidades que debe tener el mismo, y uno después de finalizar el sistema, la cual evalúa aspectos relacionados al uso del Sistema de Información para la Gestión de Versiones y Liberaciones de Software (SIGVLS).

Las encuestas fueron validadas a través de la técnica de juicio de expertos, donde intervinieron profesionales en el área de Tecnologías de Información y Comunicación, los cuales realizaron correcciones al contenido. Las encuestas fueron aplicadas a un grupo de 15 usuarios de la empresa CANTV.

El formato de las encuestas se encuentra en los Anexos 12 y 13.

Los resultados arrojados por la encuesta aplicada antes de construir el SIGVLS fueron:

En cuanto a ITIL:

El 67% de los encuestados tiene conocimiento sobre ITIL, mientras que el 33% restante no, lo cual se puede apreciar en la Figura 97.

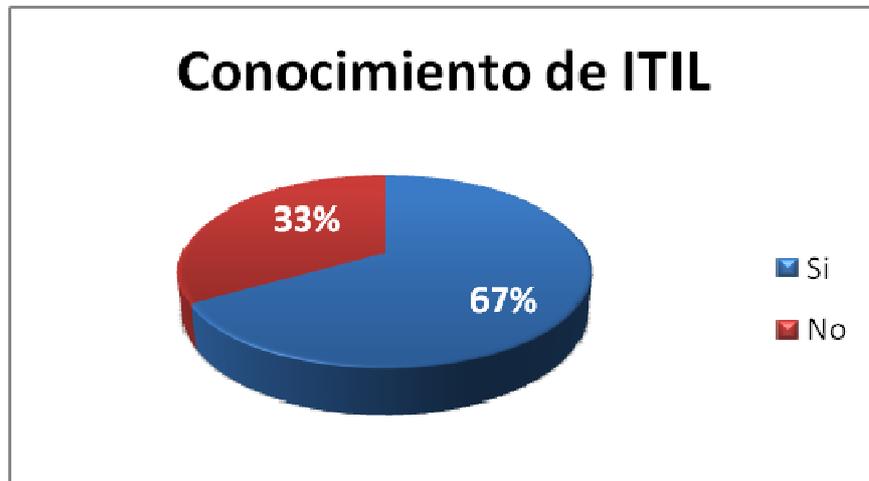


Figura 97: Evaluación del conocimiento de ITIL.

Si bien el 67% de los encuestados conoce sobre ITIL, sólo el 47% tiene conocimiento sobre el Proceso de Gestión de Versiones y Liberaciones de ITIL, mientras que el 53% restante no. (Ver Figura 98).



Figura 98: Evaluación del conocimiento del Proceso de Gestión de Versiones y Liberaciones

En cuanto a la importancia de contar con un Sistema de Información para la Gestión de Versiones y Liberaciones:

El 87% considera que es importante contar con un SI que Gestione las Versiones y Liberaciones, ya que de esta manera se puede controlar todo el proceso; mientras que 13% no lo considera importante. (Ver Figura 99).



Figura 99: Evaluación de la importancia de un SI de Gestión de Versiones y Liberaciones.

Los resultados obtenidos al aplicar la encuesta sobre los aspectos del uso del SIGVL fueron:

En cuanto a la facilidad de uso del sistema, un 87% de los encuestados consideró que es normal, mientras que el 13% restante lo consideró fácil. (Ver Figura 100).

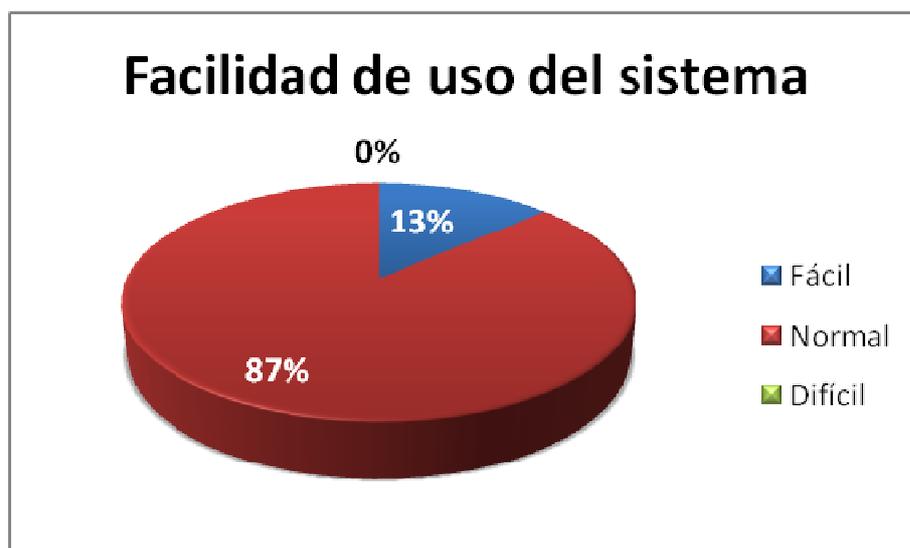


Figura 100: Evaluación de la facilidad de uso del sistema.

Al 80% de los encuestados la navegación del sistema les resultó intuitiva, al 13% les pareció muy intuitiva y a un 7% poco intuitiva. (Ver Figura 101).



Figura 101: Evaluación de la navegación del sistema.

La distribución de elementos del sistema le pareció adecuada al 93% de los encuestados mientras que al 7% restante le pareció poco adecuada. (Ver Figura 102).

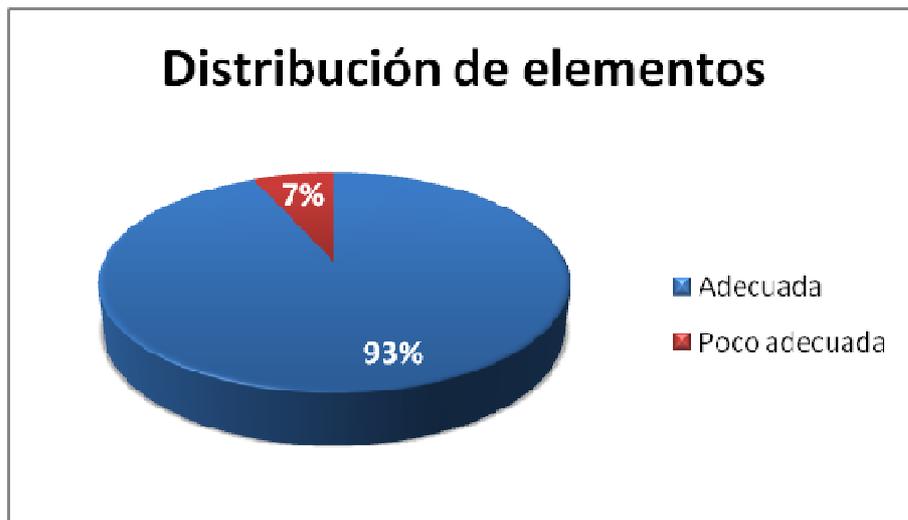


Figura 102: Evaluación de la distribución de elementos en el sistema.

El 53% de los encuestados consideró que el tiempo de respuesta del sistema es rápido mientras que el 47% lo consideró normal. (Ver Figura 103).



Figura 103: Evaluación del tiempo de respuesta del sistema.

3.4 Resultados

En la primera evaluación realizada antes de construir el sistema, se pudo observar que del grupo encuestado, un 33% no tiene conocimiento del estándar ITIL y un 53% no conoce sobre el proceso de gestión de versiones y liberaciones, para cubrir esta deficiencia, es necesario que los usuarios sean capacitados antes de empezar a usar el sistema y así aprovechar al máximo los beneficios de este proceso.

De acuerdo a los resultados de la segunda evaluación y a las pruebas realizadas, se constató que los resultados obtenidos fueron satisfactorios ya que el 87% de los participantes consideran que la facilidad de uso del sistema es normal debido a su practicidad; el 80% consideran que la navegación del sistema es intuitiva ya que al momento de usar el sistema les resultó familiar; el 93% coincidieron en que la distribución de los elementos es adecuada puesto que los formularios solicitan y muestran información de manera organizada e identificada; y el 53% consideró que el tiempo de respuesta es rápido.

En base a los resultados de la segunda evaluación, se demuestra que el SIGVLS controla las etapas del proceso, además ayuda a organizar la documentación que se produce a lo largo del proceso de gestión de versiones permitiendo que los planes se vayan almacenando en un repositorio y de esta manera se garantiza una correcta gestión de los cambios y el cumplimiento de las mejores prácticas.

Así mismo es importante destacar que al contar con un proceso formal para gestionar las versiones de hardware y software dentro de la infraestructura de TI y al tener una base de datos con la información de los componentes más importantes permitirán llevar un control de los activos de TI.

CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES

Una vez finalizado el desarrollo del Sistema de Información para la Gestión de Versiones y Liberaciones se analizaron los resultados obtenidos y se constató que los objetivos planteados en este TEG fueron cumplidos, razón por la cual se llegó a las siguientes conclusiones:

El diagnóstico de CANTV arrojó que la forma como se llevaba la Gestión de Versiones y Liberaciones era manual, lenta, desorganizada y no estaba basado en estándares. Al implementar el proceso de Gestión de Versiones y Liberaciones propuesta por el marco referencial ITIL v3 se logró llevar un control óptimo sobre el entorno de desarrollo de software.

Por consiguiente, al contar con el proceso de Gestión de Versiones y Liberaciones de ITIL v3 favorecerá significativamente los niveles de rendimiento y competitividad de la empresa, gracias a que éste proceso se adaptó a sus necesidades y permitió controlar y supervisar los cambios que se producen en el entorno de TI.

Considerando el escenario actual de CANTV y los aportes del proceso de Gestión de Versiones y Liberaciones, se diseñó el proceso basado en estándares internacionales que toma en cuenta las perspectivas de cada uno de los roles involucrados que va desde el Administrador del Proceso de Gestión de Versiones, el Administrador de Gestión de Versiones, el Equipo de Pruebas, el Equipo de Desarrollo, la Gestión de Cambios, los Proveedores Externos hasta el Grupo de Usuarios Finales. De esta manera, se logró también identificar y definir las áreas y las responsabilidades de las personas involucradas.

Al contar con un diseño definido del proceso de Gestión de Versiones y Liberaciones de Software, se logró modelar los subprocesos: política de liberación, plan de liberación, diseño de la versión, construcción de la versión,

pruebas, aceptación de liberación, planificación de implementación, comunicación y adiestramiento y distribución e instalación según las especificaciones de ITIL v3 empleando BPMN 2.0.

Una vez que el proceso fue diseñado, se logró comprender la forma de trabajo de la organización e identificar los problemas que le impedía a la empresa tener una correcta gestión.

El sistema desarrollado utilizó tecnologías de información libre tales como Intalio *Community Edition* 6.0.2 para modelar y desplegar el proceso; el manejador PostgreSQL 8.3.10 para construir la base de datos y el ESB Mule *Studio* 1.0.0 para implementar el servicio web con la arquitectura SOA permitiendo que el sistema se comuniquen con él mediante el protocolo SOAP.

Con la implementación se mejoró la organización del trabajo, logrando así una mayor eficiencia en las tareas, una reducción de costos al cumplir con el Decreto Presidencial N° 3390 sobre la implantación de software libre, un aumento en la productividad en los equipos de trabajo y la redistribución equitativa de las cargas de trabajo, una mejor respuesta a los problemas y una integración de la información del negocio.

Posterior a la implementación, se configuró y se desplegó el proceso en el motor de ejecución para poder observar su comportamiento.

Por último, para evaluar este comportamiento se realizaron pruebas de carga, volumen y estrés; pruebas unitarias e integrales al sistema y una encuesta a un grupo de usuarios finales, los cuales permiten concluir que los objetivos y requerimientos esperados se cumplieron satisfactoriamente.

Cabe destacar que el sistema se encuentra en fase de pruebas para luego ponerlo en producción.

La Metodología de Gestión de los Procesos de Negocio Sustentada en el uso de Patrones definió las actividades realizadas en las etapas de análisis, diseño, modelado y configuración para desarrollar el Sistema de Gestión de Versiones y Liberaciones de Software. La metodología también define las actividades de monitoreo y mantenimiento, las cuales no se realizaron en este TEG dejándolas a criterio de la empresa una vez que el sistema esté en fase de producción.

Limitaciones

Durante el desarrollo de la investigación se presentaron ciertas dificultades, las cuales se listan a continuación:

- Escasa información y soporte sobre *Intalio Community Edition*
- Poca información sobre ITIL v3
- La gran limitante radica en el aspecto económico, ya que tanto los entrenamientos para aprender a usar el BPMS *Intalio Community* como los cursos y libros de ITIL son costosos

Recomendaciones de Trabajos Futuros

Como recomendaciones se tienen las siguientes:

- Integrar la solución con los sistemas de Gestión de Cambio y Gestión de Problemas de la Organización
- Incorporar a la solución las actividades del subproceso de Administración propuesta en la Metodología de Gestión de los Procesos de Negocio Sustentada en el Uso de Patrones
- Se propone integrarlo con un sistema de gestión de reglas de negocio de código abierto como Drools
- Se recomienda integrar la solución al Directorio Activo de la Empresa
- Diseñar, construir e integrar el resto de los procesos del marco referencial ITIL v3 utilizando el BPMS *Intalio Community Edition*

REFERENCIAS BIBLIOGRÁFICAS

- Álvarez, J., & Espada, P. (2001). *Web Services*. Universidad de Extremadura.
- Arias, F. (1999). *El proyecto de investigación: Guía para su elaboración*. Caracas: Episteme.
- Barros, O. (1994). *Reingeniería de Procesos de Negocio*. Chile: Dolmen.
- Bonillo, P. (2008). Metodología para la Gestión de los Procesos de Negocio Sustentada en el uso de Patrones. *Tesis Doctoral*. Caracas: Universidad Central de Venezuela.
- CANTV. (2009). Informe anual 2009.
- CANTV. (2011). *La Empresa*. Recuperado el 20 de Marzo de 2011, de http://www.cantv.com.ve/seccion.asp?pid=1&sid=158&id=2&und=6&cat=item_lc&item=item_7&item_name=Historia
- Cartlidge, A., Hanna, A., Macfarlane, I., Rance, S., Rudd, C., & Windebank, J. (2007). *An Introductory Overview of ITIL V3*. The IT Service Management Forum.
- Chapin, N., Hale, J., Khan, K., Ramil, J., & Tan, W. (2001). *Types of software evolution and software maintenance*.
- Chávez, V., & Matheus, R. (2008). Sistema para la Administración de los Procesos de Soporte a los Servicios de ITIL usando la metodología Process Management. *Tesis*. Caracas: Universidad Central de Venezuela.
- Club-BPM. (3 de Noviembre de 2009). *¿Qué es BPM?* Recuperado el 10 de Diciembre de 2009, de <http://www.club-bpm.com/ApuntesBPM/ApuntesBPM01.pdf>
- Collins-Sussman, B., Fitzpatrick, B., & Pilato, C. (2004). *Version Control with Subversion*. Recuperado el 23 de Septiembre de 2010, de <http://svnbook.red-bean.com/en/1.7/svn-book.pdf>
- Comunidad Osiatis. (2009). *Gestión de Cambios*. Recuperado el 4 de Marzo de 2009, de http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/gestion_de_cambios/vision_general_gestion_de_cambios/vision_general_gestion_de_cambios.php
- Comunidad Osiatis. (2011). *Gestión de Entregas y Despliegues*. Recuperado el 15 de Marzo de 2011, de http://itilv3.osiatis.es/transicion_servicios_TI/gestion_entregas_despliegues.php
- Comunidad Osiatis. (2009). *Gestión de Servicios TI*. Recuperado el 25 de Abril de 2010, de http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/fundamentos_de_la_gestion_TI/vision_general_gestion_servicios_TI/vision_general_gestion_servicios_TI.php
- Cristalab. (15 de Diciembre de 2009). *Control de versiones de software*. Recuperado el 6 de Abril de 2010, de <http://www.cristalab.com/blog/control-de-versiones-de-software-c82353/>
- De la Cruz, J. (2006). Orquestador de servicios web para el desarrollo de aplicaciones distribuidas. *Tesis*. México: Instituto Politécnico Nacional.
- Dubs, R. (2002). *El proyecto factible: una modalidad de investigación*. Caracas: Sapiens.
- Eyssautier, M. (2006). *Metodología de la investigación: desarrollo de la inteligencia*. Cengage Learning Editores.

- Felipe, E. (31 de Agosto de 2009). *El arte del modelado de procesos de negocios ejecutables*. Recuperado el 8 de Octubre de 2010, de estebanf: <http://www.estebanf.com/bpm/2009-09-07/el-arte-del-modelado-de-procesos-de-negocio-ejecutables/>
- Felipe, E. (2008). *Elementos de un BPMS*. Buenos Aires.
- Felipe, E. (25 de Agosto de 2009). *Intalio|BPM: Mas allá de SOA*. Recuperado el 9 de Octubre de 2010, de estebanf: <http://www.estebanf.com/bpm/2009-08-26/intaliobpm-mas-alla-de-soa/>
- Garimella, K., Lees, M., & Williams, B. (2008). *Introducción a BPM para Dummies*. Indianapolis: Wiley Publishing.
- Gartner. (2009). Recuperado el 13 de Noviembre de 2009, de Technology Research | Gartner Inc: <http://www.gartner.com/technology/home.jsp>
- Gartner. (16 de Abril de 2003). *Service-Oriented Architecture Scenario*. Recuperado el 18 de Septiembre de 2010, de Technology Research | Gartner Inc: http://www.gartner.com/DisplayDocument?doc_cd=114358
- Haren, V. (2004). *Mejora Continua del Servicio basada en ITIL V3: Guía de Gestión*. Colección Best Practice.
- Hernández, E. (2009). *El Lenguaje Unificado de Modelado (UML)*.
- Hewlett Packard. (2003). *Web Services Orchestration*. Recuperado el 20 de Noviembre de 2010, de <http://xml.coverpages.org/HP-WSOrchestration.pdf>
- Intalio. (2009). Recuperado el 14 de Diciembre de 2009, de <http://www.intalio.com/products/>
- IT Process Maps. (2011). *Transición del Servicio*. Recuperado el 24 de Marzo de 2011, de http://wiki.es.it-processmaps.com/index.php/ITIL_Transici%C3%B3n_del_Servicio
- ITIL. (2006). *ITIL Glossary*.
- ITIL Service Management. (4 de Julio de 2007). Recuperado el 7 de Marzo de 2011, de <http://itservicemngmt.blogspot.com/2007/07/release-management-quick-reference.html>
- Laudon, J., & Laudon, K. (2006). *Sistemas de Información Gerencial*. Mexico: Pearson Education - Prentice Hall.
- Lobos, D., Baquinzay, M., & Bustos, M. (2008). *GESTIÓN DE SERVICIOS TIC (Tecnología de la Información y las Comunicaciones) - ITIL (Biblioteca de Infraestructura de Tecnologías de Información)*.
- Mako, B. (2000). *HOWTO Gestión de proyectos de software libre*.
- Mejía, M., & Arzate, L. (2006). *Automatización de Procesos de Negocio utilizando un BPMS*. México, D.F., México.
- Moreno, A. (2001). *Diccionario de Informática y Telecomunicaciones. Definición de Componente*. Barcelona: Ariel.
- Mulesoft. (2011). *What is Mule ESB?* Recuperado el 25 de Agosto de 2011, de <http://www.mulesoft.org/what-mule-esb>
- OASIS. (2004). *Introduction to UDDI: Important Features and Functional Concepts*. Recuperado el 18 de Agosto de 2011, de <http://www.uddi.org/pubs/uddi-tech-wp.pdf>
- Oportunis Consulting. (2010). Recuperado el 10 de Abril de 2010, de <http://www.oportunis.cl/?p=132>

- Pedraz, A. (2004). La revisión bibliográfica. *Nure investigación* .
- Pelechano, V. (2004). *Servicios Web. Estándares, Externsiones y Perspectivas de Futuro*. Recuperado el 28 de Septiembre de 2010, de ftp://jano.unicauca.edu.co/cursos/Enfasis_III/Documentacion/Referencias/ServiciosWeb.pdf
- Pérez, J. (2007). *Servicios Web: Orquestación y Coreografías*. Universidad de Oviedo.
- PostgreSQL-es. (2 de 10 de 2010). *Sobre PostgreSQL*. Recuperado el 28 de Agosto de 2011, de http://www.postgresql.org.es/sobre_postgresql
- Quesada, H. (2006). *Administración por procesos: la metodología BPM*. Costa Rica: Instituto Tecnológico de Costa Rica.
- Rodríguez, A., Márquez, A., & Toro, M. (2001). *Gestión de la evolución del software*. España: Universidad de Granada.
- Sabino, C. (1992). *El Proceso de Investigación*. Caracas: Panapo.
- Sette, E., & Marques, R. (2006). Uma Proposta Ontológica para um Sistema de Gestão de Versionamento do Modelo ITIL. *Tesis* . Brasil: Universidad de Brasilia.
- Sicuma. (2007). *SOA y BPEL*. Recuperado el 20 de Septiembre de 2010, de <http://www.sicuma.uma.es/sicuma/independientes/argentina08/Marinelli-Kuna/index.html>
- Silver, B. (2006). The 2006 BPMS Report: Understanding and Evaluating BPM Suites.
- Smart, P., Maddern, H., & Maull, R. (2008). *Understanding Business Process Management implications for theory and practice*. Recuperado el 24 de Marzo de 2010, de http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1523415
- SOA Agenda. (2010). *BPMS y ciclo de vida de los procesos*. Recuperado el 25 de Abril de 2010, de <http://soaagenda.com/journal/articulos/bpms-y-ciclo-de-vida-de-los-procesos/>
- SOA Agenda. (2010). *Xtreme BPM ó Agile BPM*. Recuperado el 25 de Abril de 2010, de <http://soaagenda.com/journal/articulos/xtreme-bpm-o-agile-bpm/>
- Taylor, S., & Turbitt, K. (2008). ITIL versión 3: un modelo adaptado a la creciente importancia de la gestión de servicios de negocio.
- Universidad Pedagógica Experimental Libertador. (2003). Manual de Trabajos de GRado de Especialización y Maestría y Tesis Doctorales. Caracas.
- Van Bon, J. (2002). *IT Service Management: An Introduction*. Van Haren Publishing.
- W3C . (2007). *SOAP Version 1.2 Part 1: Messaging Framework*. Recuperado el 28 de Septiembre de 2010, de <http://www.w3.org/TR/soap12-part1/#soapenv>
- W3C. (2003). *Web Services Architecture*. Recuperado el 25 de Septiembre de 2010, de <http://www.w3.org/TR/2003/WD-ws-arch-20030514/>
- W3C. (2004). *Web Services Glossary*. Recuperado el 24 de Septiembre de 2010, de <http://www.w3.org/TR/ws-gloss/>
- Wikipedia. (2010). *Fases del desarrollo de software*. Recuperado el 3 de Abril de 2010, de http://es.wikipedia.org/wiki/Fases_del_desarrollo_de_software
- Yang, J. (2003). *Web Service Componentization: Towards Service Reuse and Specialization*. Tilburg University.

ANEXOS

Anexo 1 Aspectos de la organización de estudio CANTV

La Compañía Anónima Nacional Teléfonos de Venezuela (CANTV), ente adscrito al Ministerio del Poder Popular para Ciencia, Tecnología e Innovación, y junto a sus filiales Movilnet y Caveguías, es la primera empresa de telecomunicaciones en Venezuela que tiene como objetivo fundamental fomentar la inclusión social y la disminución de la brecha al acceso de tecnologías digitales, facilitando así el alcance de todos a los servicios de telecomunicaciones. (CANTV, 2011).

Reseña Histórica

CANTV fue fundada en 1930, y hoy en día es el proveedor líder de servicios de telefonía fija, móvil, Internet y servicios de información del país. (CANTV, 2011).

CANTV posee una estructura de propiedad mixta, en la que participan tanto pequeños ahorristas, como trabajadores y jubilados, capitales nacionales y extranjeros y bloques de inversión institucionales y estratégicos, como por ejemplo, el Estado venezolano y experimentadas empresas de la industria mundial de las telecomunicaciones. (Ibíd.).

La Corporación dispone de las tecnologías más avanzadas, lo cual, aunado al desarrollo de mejores prácticas gerenciales, ha permitido llevar adelante una importante transformación en cobertura y calidad de servicios. (Ibíd.).

A través de los siglos XX y XXI, CANTV ha pasado por diferentes facetas que comienzan en 1930 con una concesión otorgada al venezolano Félix A. Guerrero, pasando por ser empresa pública entre 1953 y 1991, para luego volver a manos privadas por un lapso de 15 años, entre 1992 y 2007, año en que pasa, de nuevo, al control del Estado venezolano. (Ibíd.).

CANTV sirve a Venezuela con las tecnologías más avanzadas y dispone de una red de fibra óptica interurbana de 7.800 kilómetros de longitud a través de siete gigantescos anillos que proporcionan redundancia, garantizando, por tanto, confianza y seguridad en el servicio. (Ibíd.).

De igual manera, dispone de la mayor cobertura de servicio de transporte de datos y voz más usado mundialmente como es el *Frame Relay*, el cual permite un uso dinámico del ancho de banda, con velocidad de acceso escalable desde 64 hasta 2.048 Kbps con alta disponibilidad. (CANTV, 2011).

Mediante redes de transmisión que emplean sistemas de radio de microondas terrestres, CANTV satisface las necesidades de comunicación en poblaciones en donde no existe posibilidad de prestar el servicio a través de la plataforma de cableado. (Ibíd.).

Cuenta con una amplia cobertura de puertos ADSL (*Asymmetric Digital Subscriber Line*) para poder brindar el servicio de acceso a Internet de banda ancha en todo el país, siguiendo un plan de instalación de 130.000 puertos anuales en la red de IP (*Internet Protocol*) que ofrezca en promedio más velocidad, hasta 3.448 Kbps por cliente, como mínimo. (Ibíd.).

Desde el punto de vista de conexiones con el resto del mundo, CANTV forma parte del sistema internacional de cables submarinos que surca todo el planeta. De hecho, directa o indirectamente, las redes de CANTV están interconectadas a ocho cables submarinos desde sus puntos de amarre en Camurí Chico y Punto Fijo. (Ibíd.).

En cuanto a redes móviles, la Corporación se ha posicionado como la operadora de telefonía móvil con la más amplia cobertura en el país, al contar con más de mil radio bases CDMA (*Code Division Multiple Access*) 1X a lo largo y ancho de la geografía nacional; lo que le proporciona a Movilnet presencia en lugares sin competencia, cubriendo a todas las poblaciones venezolanas con más de 3.000 habitantes. (Ibíd.).

Paralelamente, Movilnet viene desarrollando una red de banda ancha inalámbrica con tecnología EvDO (*Evolution-Data Optimized* o *Evolution-Data Only*) en la Gran Caracas, doce estados del país y zonas de importancia turística como el Archipiélago de Los Roques. (CANTV, 2011).

Todas estas fortalezas tecnológicas y de mercado han sido respaldadas por un esfuerzo de inversión que supera los US\$ 6.700 millones en los últimos quince años. (Ibíd.).

Hoy, CANTV es la empresa preferida de los venezolanos porque a través de sus redes fijas, móviles y satelitales, ofrece a los venezolanos la posibilidad de estar comunicados, en cualquier momento y en cualquier lugar, con servicios de voz, datos y video de alta confiabilidad y velocidad de respuesta. (Ibíd.).

Misión

Proveer soluciones integrales de telecomunicaciones e informática, corresponsable de la soberanía y transformación de la nación, que potencia el poder popular y la integración de la región, capaz de servir con calidad, eficiencia y eficacia, y con la participación protagónica del pueblo, contribuyendo a la suprema felicidad social. (Ibíd.).

Visión

Ser una empresa socialista operadora y proveedora de soluciones integrales de telecomunicaciones e informática, reconocida por su capacidad innovadora, habilitadora del desarrollo sustentable y de la integración nacional y regional, comprometida con la democratización del conocimiento, el bienestar colectivo, la eficiencia del Estado y la soberanía nacional. (Ibíd.).

Objetivos

CANTV alineada con la visión del país, tiene como objetivos (CANTV, 2011):

- Democratizar el servicio con justicia social: ampliando la cobertura geográfica, incluyendo a todos los segmentos de la población, ofreciendo tarifas justas y solidarias para promover una competencia más equitativa, con atención particular para cada segmento de la población para facilitar la integración al uso de las telecomunicaciones.
- Potenciar la participación y el Poder Popular: las comunidades se convierten en aliadas en la prestación del servicio. En esta etapa, CANTV promueve la participación protagónica de las comunidades organizadas, al tiempo que potencia la labor de los Consejos Comunales.
- Garantizar auto sostenibilidad de la empresa: la nueva CANTV será eficiente en sus operaciones, de manera de generar los recursos requeridos para acometer proyectos con rentabilidad social, pero siempre asegurando la viabilidad económica de la empresa.
- Convertirnos en empresa socialista del Estado: la empresa se ajustará al marco legal de empresa pública e implantará el modelo laboral socialista, impulsando la participación protagónica de los trabajadores como servidores públicos, bajo un espíritu de solidaridad y abriendo espacios para los Esquemas Asociativos Solidarios con el fin de desarrollar el modelo de economía social.
- Avanzar hacia la soberanía tecnológica: la nueva CANTV apoyará la implantación del software libre cumpliendo con el Decreto Presidencial N° 3.390 del Ministerio de Ciencia y Tecnología de la República Bolivariana de Venezuela decretada el 23 de diciembre de 2004. Además, impulsará la apropiación tecnológica por parte de los ciudadanos y ciudadanas, promoverá el desarrollo endógeno, respaldará la formación de talentos nacionales y promoverá la sustitución de importaciones.
- Apalancar la transformación del Estado: CANTV jugará un papel protagónico en la transformación del Estado apalancado con el potencial

que ofrecen las tecnologías para acercarse al ciudadano y servirlo de manera más eficiente, ágil y confiable; facilitando a su vez su participación en el diseño de las políticas públicas que guían la acción del Estado.

- Apoyar la integración Nacional e Internacional: CANTV cobra una dimensión internacional, expandiendo las fronteras tecnológicas de la nación, bajo el lineamiento del acuerdo ALBA, el proyecto satelital VENESAT-1, que servirá para brindar apoyo a los programas sociales y del Estado y facilitar la transferencia tecnológica.

Asimismo, se apoyará la seguridad y la defensa integral del Estado proveyendo una red de comunicaciones segura y de alcance nacional. La nueva CANTV asume el reto de crear la concepción socialista del servicio de telecomunicaciones, abrir espacios reales para la participación de las comunidades, colocar las innovaciones tecnológicas al servicio del pueblo, convertirse en un motor de integración para los pueblos de la región, contribuir a definir el perfil del Servidor Público Socialista y coadyuvar en el desarrollo del modelo de economía social sustentable y endógeno.

Organigrama

La Figura 1.1 muestra el organigrama de la empresa CANTV, en ella se destaca al departamento de Gerencia de Programa de Soluciones TI ya que representa el caso de estudio de este Trabajo Especial de Grado.

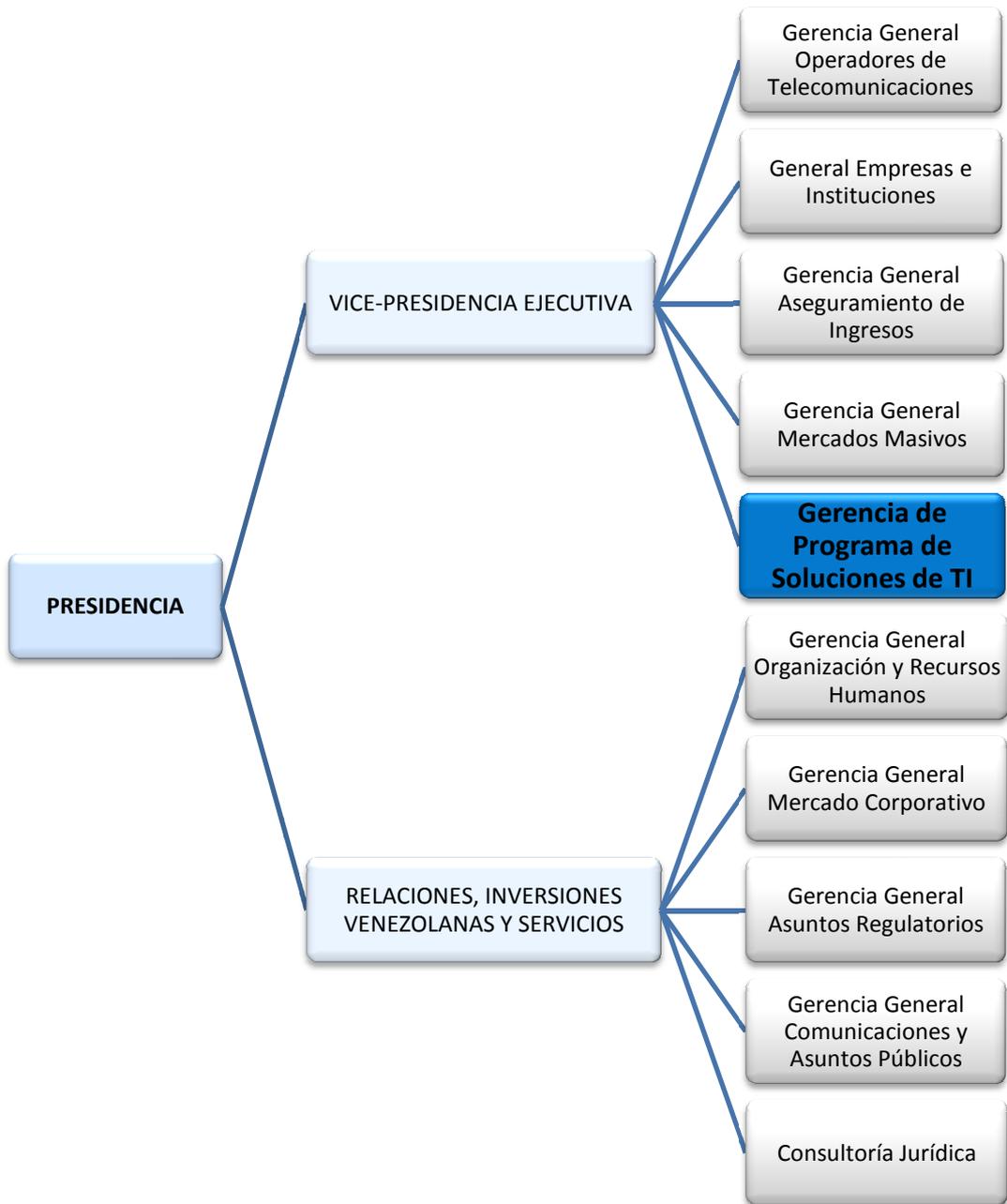
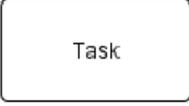
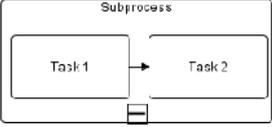
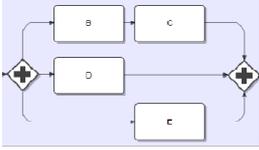
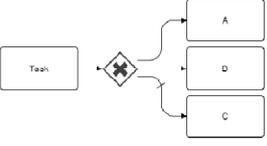
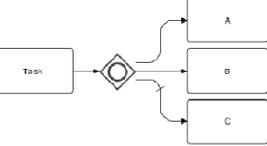
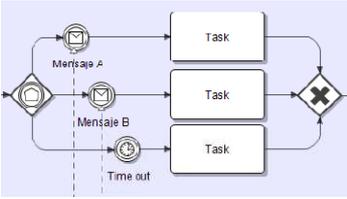


Figura 1.1: Estructura organizacional de CANTV.

Anexo 2 Elementos BPMN de la paleta de Intalio|Designer

Tipo de componente	Componente	Símbolo	Descripción
Participante	Pool ejecutable		Es el proceso principal, que será llevado a BPEL para su posterior puesta en ejecución.
	Pool no ejecutable		Son Sistemas o Personas externas que interactúan con el proceso, se convierten en Servicios Web.
Actividad	Tarea		Representa una acción realizada por un participante.
	Subproceso		Es una actividad compuesta, sirve para agrupar tareas en una sola. Se puede mostrar contraído o expandido.
Decisión	Bifurcación paralela		Sirve para indicar la ejecución paralela de tareas y/o eventos.
	Bifurcación condicional exclusiva		Solo una ruta se ejecuta, según la condición evaluada.
	Bifurcación condicional inclusiva		Al menos una ruta se ejecuta, puede tomar más de una ruta y se comporta como un paralelo.
	Bifurcación basada en eventos		Solo una ruta se ejecuta, y es la de la primera ocurrencia del evento.

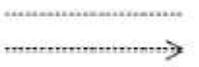
Eventos	Inicio		Ilustra que el proceso inicia en ese punto.
	Intermedio	<p>Mensaje  </p> <p>Error  </p> <p>Cancelar  </p> <p>Señal  </p> <p>Enlace  </p> <p>Compensación  </p>	Indica la presencia de un cambio de estado o un punto de captura de indicadores.
	Fin		Ilustra que el proceso culmina.
	Terminar		Detiene el proceso inmediatamente, incluyendo cualquier ruta paralela.
Conexión	Flujo		Define como un participante ejecuta una tarea (la secuencia).
	Interacción		Se refiere a un intercambio de mensajes y datos entre participantes. No existe interacción dentro de un mismo participante.
	Asociación		Asocia tareas y elementos.
Artefacto	Anotación		Es utilizado para agregar comentarios.
	Grupo		Un conjunto arbitrario de objetos pueden ser agrupados, lo cual muestra que están juntos lógicamente.

Tabla 2.1: Elementos BPMN de la paleta Intalio|Designer.

Fuente: Intalio (2009)

Anexo 3 Características de Intalio|Designer

Generación/importación de código	<ul style="list-style-type: none"> • Validación de procesos • Importación y generación de código WS-BPEL 2.0 • Generación WSDL • Generación automática de código de procesos • Importación de código en BPEL4WS 1.0/1.1 • Importación ARIS EPC
Entorno	<ul style="list-style-type: none"> • Versión <i>Standalone</i> • Soporte BPMN 1.1 • Entorno de desarrollo integrado • <i>Plugin</i> Eclipse
Gestión del ciclo de vida	<ul style="list-style-type: none"> • Versionamiento colaborativo • Registro/ comprobación • Gestor de dependencia gráfica • Búsqueda avanzada • Versionamiento local
Editor de mapeo	<ul style="list-style-type: none"> • Soporte de esquema XML • Mapeo de transformación gráfica • Mapeo de reglas gráficas • Validación de esquema transparente • Soporte a esquemas complejos • Editor de mapeo de esquema a esquema • Soporte XPath y XQuery
Despliegue de procesos	<ul style="list-style-type: none"> • Comprobación de consistencia dinámica • Seguimiento en línea
Modelador de procesos	<ul style="list-style-type: none"> • Constructor de procesos adaptable • Manejo de flujo de trabajo • Editor visual de propiedades para los procesos • Validación semántica transparente • Conjunto avanzado de procesos prefabricados • Manejo de excepción de flujos • Interfaz de usuario agarrar y soltar (<i>drag and drop</i>)
Introspección del sistema	<ul style="list-style-type: none"> • Interfaz en línea • Generación automática del sistema interfaz en WSDL

Tabla 3.1: Características de Intalio|Designer.

Fuente: Intalio (2009)

Anexo 4 Características de Intalio|Server

Soporte de hardware	<ul style="list-style-type: none"> • HP-PA RISC 32 y 64 bit • IBM Power 32 y 64 bit • Intel x86 e Intel Itanium 32 y 64 bit • Opteron EM64 • Sun Sparc 32 y 64 bit
Soporte operativo	<ul style="list-style-type: none"> • HP-UX 11.21 para PA-Risc 64 bit y HP-UX 11.23 para Itanium 64 bit • IBM AIX 5.2, 5.3 – 32 y 64 bit • Sun Solaris 8, 9, 10 – 32 y 64 bit • SUSE Linux 9, 10 • <i>Red Hat</i> Linux 4, 5 • Windows 2003 Server, 2000 Server
Soporte al servidor de aplicaciones	<ul style="list-style-type: none"> • Apache Geronimo 2.0.1 • Apache Tomcat 5.5.26 • JBoss <i>Application Server</i> 4.0.5GA • IBM WebSphere <i>Application Server</i> 6.1.0.3
Soporte a base 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 (solo para Windows) • MySQL Enterprise 5.0 • Oracle 8i, 9i, 10g • PostgreSQL 8.1.10, 8.2.5 • Sybase ASE 12.5.1
Gestión y despliegue	<ul style="list-style-type: none"> • Balanceo de carga • <i>Clustering</i> • Instrumentación JMX • Depuración de API's
Consola de gestión	<ul style="list-style-type: none"> • Gestión de consola basada en navegadores • Control de acceso basado en roles • Servidor de tablero de instrumentos con indicador de estado • Lista de procesos desplegados • Gestión de proceso (activos, retirados) • Lista de instancias de procesos en ejecución • Búsqueda y filtro de instancias de procesos • Visor de instancias de procesos (estado, variables y eventos)
Ejecución de procesos	<ul style="list-style-type: none"> • Compilador de procesos <i>Just-in-Time</i> • Soporta WS-BPEL 2.0 • Soporta BPEL4WS
SOA	<ul style="list-style-type: none"> • Interfaz REST-style • Interfaces Plain Old XML (POX) • WS-Direccionamiento • WS-Transacciones atómicas • WS-Coordinación • WS-Seguridad
Conectores de tecnología	<ul style="list-style-type: none"> • AS400 Data • Email (SMTP, POP3, IMAP4) • Sistema de archivos • FTP • HTTP, HTTP/S • GigaSpaces • Glue • IIOP • JAAS • Java

	<ul style="list-style-type: none"> • JavaSpaces • JBI • JCA • JDBC • JMS • JNDI • JOTM • JTA • MQ Series • Oracle AQ • PicoContainer • Plexus • Remote EJB • REST • RMI • SOAP • Sistema de Entrada/ Salida • TCP • TIBCO Rendezvous • TLS • VFS • UDP • XFire • XMPP
Conectores de aplicación	<ul style="list-style-type: none"> • Aplicaciones Google • Aplicaciones Oracle • PeopleSoft • Salesforce.com • SAP • Siebel
Soporte dinámico para formularios	<ul style="list-style-type: none"> • Soporta XForms 1.1 • Diseñador XForms • Engine XForms • Intalio AJAX
Gestor de flujos de tareas	<ul style="list-style-type: none"> • Planificación de tareas • Notificación de tareas • Manipulación de tareas • Eliminación de tareas • Asignación de tareas • Gestor extensible de tareas • Soporta BPEL4People

Tabla 4.1: Características de Intalio|Server.

Fuente: Intalio (2009)

Anexo 5 Arquitectura Orientada a Servicios

SOA es una forma de arquitectura de sistemas distribuidos que posee las siguientes propiedades (Sicuma, 2007):

1. Orientación a la conversación: el foco de atención no está en los nodos sino en los mensajes que se intercambian entre los mismos.
2. Vista lógica: el servicio es una abstracción (vista lógica) de los programas, bases de datos, procesos de negocio y otros.
3. Orientación a mensajes: el servicio se define formalmente en términos de los mensajes que se intercambian entre agentes, proveedores y solicitantes, no debe ser definido en términos de las propiedades de los agentes. Esto permite que se incorpore un componente “decorando” estos componentes con software de gestión y conversión.
4. Abstracción del agente: la estructura interna del agente (lenguaje de programación, BD y otros.) se abstrae en SOA, un nodo es una entidad computacional que participa en conversaciones con otros nodos. No es necesario conocer las particularidades del lenguaje de implementación. Esto evita problemas arquitectónicos derivados de la necesidad de conocer determinados sistemas a nivel estructural.
5. Metadatos: SOA se asocia con metadatos, los mismos son descripciones acerca de la forma y tipo de los elementos que transportan los mensajes, el orden de los mensajes, el significado de los mensajes y otros.
6. Orientado a redes: los servicios tienden a usarse a través de la red, aunque este no es un requisito absoluto.
7. Granularidad: los servicios tienden a usar un pequeño número de operaciones con mensajes relativamente grandes y complejos.
8. Neutral a la plataforma: los mensajes son enviados en un formato estándar y neutral a la plataforma, utilizando XML.

SOA posee los siguientes componentes:

- Roles:
 1. Proveedor de servicios: se define como una entidad que permite que el servicio web esté disponible a través de la web. El proveedor se encarga de: crear una descripción de servicios, entregar un servicio en un entorno de ejecución para hacerlo disponible a otras entidades sobre una red, publicar la descripción del servicio en uno o más registros de servicios y recibir mensajes invocando servicios de los solicitantes de servicios.
 2. Solicitante de servicios: es cualquier consumidor de un servicio web. Para ello encuentra una descripción de servicio publicada en un registro de servicios y luego aplica la descripción del servicio para ligar e invocar el servicio web.
 3. Registro de servicios: permite el match entre los proveedores de servicios y los solicitantes de servicios. Una vez que se encontró el servicio buscado, las interacciones se llevan a cabo directamente entre el solicitante del servicio y el proveedor del servicio.
- Operaciones:
 - Publicar: cuando un proveedor de servicios publica su servicio web en un registro de servicios, está publicitando el servicio a toda la comunidad de potenciales solicitantes del servicio.
 - Ligar (bind): se encarga de buscar un servicio que satisface ciertas condiciones:
 1. El solicitante del servicio establece un criterio de búsqueda, tal como: el tipo de servicio, la calidad, y otros.
 2. El registro de servicios hace juego los criterios de búsqueda con las descripciones de servicio publicadas.
 3. El resultado es una lista de descripciones de servicios que encajan el criterio de selección.
 - Encontrar: crea la relación cliente-servidor entre el solicitante del servicio y el proveedor del servicio.

La Figura 5.1 muestra la arquitectura de un servicio web.

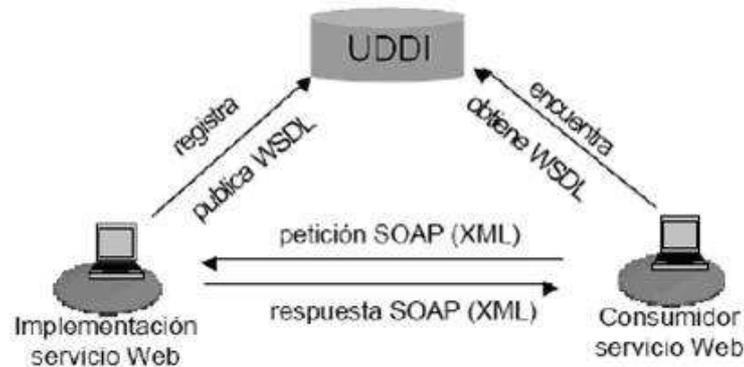


Figura 5.1: Arquitectura de un servicio web.

Fuente: W3C (2003)

SOA y WS son los enfoques más apropiados para aplicaciones que (Acevedo, 2007):

- Pueden operar sobre Internet, aceptando que la confiabilidad y el performance de comunicaciones no pueden garantizarse
- No requieren que los solicitantes de servicios y los proveedores se actualicen al mismo tiempo
- Consisten de componentes ejecutándose remotamente en diferentes plataformas de ejecución y con diferentes productos de fabricantes
- Fueron diseñadas usando tecnología llegada pero se necesita exponerlas en una red, usando una envoltura de servicio web

La infraestructura mínima que requieren los servicios web se puede definir en términos de (Pelechano, 2004):

- 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

En la Figura 5.2 se puede observar la tecnología relacionada con SOA.

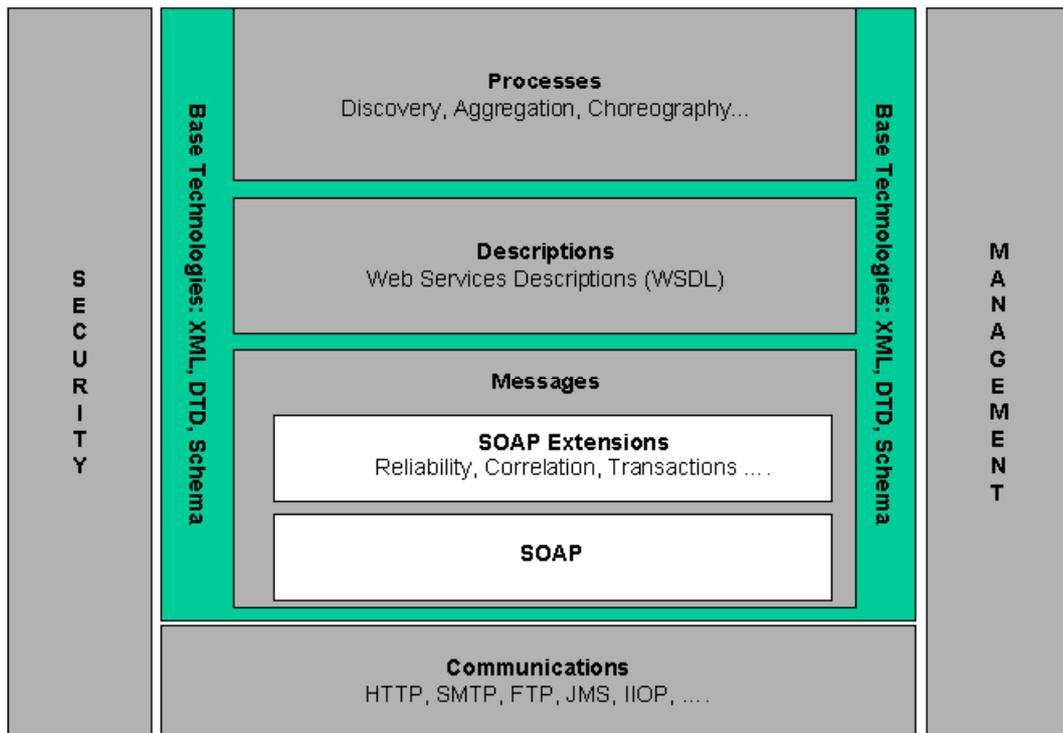


Figura 5.2: Arquitectura tecnológica relacionada con SOA.

Fuente: W3C (2003)

Capa de comunicaciones

Un mensaje de un servicio web puede ser transportado usando HTTP o HTTPS, así como otros mecanismos de transporte más específicos. HTTP (*Hypertext Transfer Protocol*) es un protocolo estándar para la transferencia de documentos en Internet. Lo utilizan los servicios web como mecanismo de comunicación.

Capa de mensajería

SOAP (*Simple Object Access Protocol*) es un protocolo para intercambiar información estructurada en un entorno distribuido, para ello 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.

Extensiones de SOAP:

- WS-ReliableMessaging: un estándar de mensajes entre servicios web para garantizar la recepción de mensajes para los solicitantes y los proveedores.
- WS-Transactions: una serie de estándares relacionados con invocaciones de servicios web en transacciones (semántica de atomicidad, consistencia y persistencia).

Capa de descripción

WSDL (*Web Services Description Language*) es un lenguaje basado en XML que permite al proveedor de servicios especificar las características funcionales de sus servicios web.

Extensiones de WSDL:

- WS-Policy: aumenta WSDL con restricciones no-funcionales de los WS.
- WS-ResourceProperties: describe cómo definir y acceder a las propiedades de los recursos a través de WS.

Capa de procesos: Descubrimiento

Consiste en ubicar una descripción de un servicio web procesable por una máquina que podría ser previamente desconocido y que satisface ciertos criterios.

UDDI (*Universal Description, Discovery and Integration*) define una forma de almacenar y buscar información sobre servicios web.

Capa de procesos: Coreografía

Define la manera en que múltiples agentes independientes cooperan intercambiando mensajes para realizar una tarea o lograr un objetivo.

WS-CDL (*WS Choreography Description Language*) describe colaboraciones *peer-to-peer* donde un intercambio de mensajes ordenado produce el cumplimiento de un objetivo de negocio común.

Tecnología Base

La comunicación de servicios web confía en la sintaxis de XML para escribir mensajes. XML (*eXtensible Markup Language*) se define como un metalenguaje extensible de etiquetas desarrollado por el consorcio W3C. Se deriva de SGML (*Standard Generalized Markup Language*), diseñado para representar y transferir datos estructurados

Con XML se obtiene:

- Una representación de datos común y bien definida
- Un conjunto bien definido de reglas de validación

Anexo 6 Aspectos de la Orquestación de Servicios

La Figura 6.1 se puede observar cómo se lleva a cabo la orquestación de servicios web, donde el rectángulo mayor representa el proceso que será orquestado, los rectángulos pequeños sombreados las actividades involucradas en el mismo, las flechas horizontales el flujo de datos intercambiado y las flechas verticales el flujo de control del proceso.

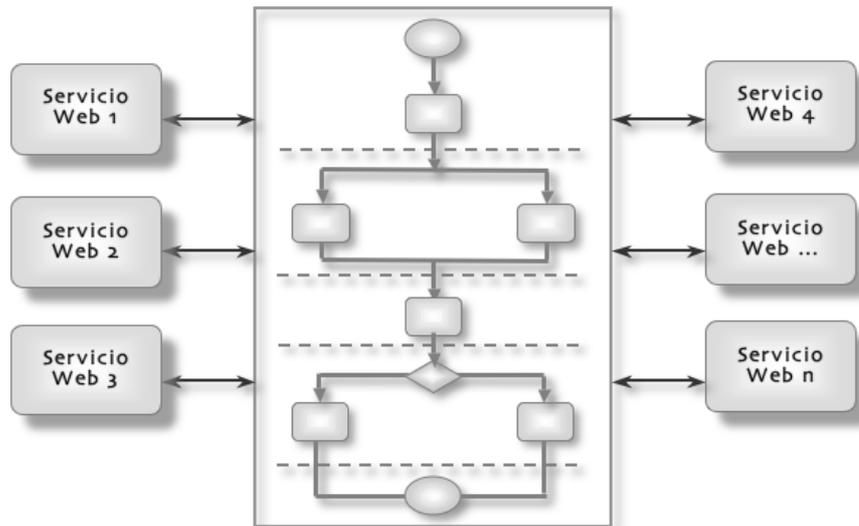


Figura 6.1: Orquestación de servicios web.

Fuente: Yang (2003)

Pero para que este proceso de orquestación se pueda llevar a cabo se emplean una serie de tecnologías y estándares actualmente existentes en el mercado, tales como BPEL (*Business Process Execution Language*), el cual permite establecer el flujo de ejecución en el proceso de orquestación de los servicios web. (Hewlett Packard, 2003).

La especificación de BPEL4WS, también llamada BPEL fue desarrollada por Microsoft, IBM, Siebel Systems, BEA y SAP. Dicha especificación modela el comportamiento de los diversos servicios web que puedan participar en la interacción de un proceso de negocio.

BPEL provee una sintaxis en XML para la descripción de la lógica de control necesaria para coordinar los servicios web que participen en un flujo de proceso. Esta gramática es ejecutada por un motor de orquestación que

coordina todas las actividades y compensa el proceso global cuando ocurre algún error. BPEL puede considerarse como una capa que se tiende sobre WSDL, ya que este último define las operaciones disponibles, mientras que BPEL define cómo secuenciarlas. (De La Cruz, 2006).

La orquestación constituye un 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. Es por ello que para muchos desarrollos, la orquestación se ha convertido en el corazón de SOA.

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.

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 un servicio de procesos.

Otro servicio permitido para interactuar con el servicio de proceso está identificado como socio de servicios o *partner links*. Dependiendo de la lógica del 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.

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 pueden ser ensambladas usando la lógica provista por las actividades estructuradas (*sequence, switch, while, flow, pick*).

Secuencias, Flujos y Acoplamientos

Las 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 la salida de otra.

Los flujos también contienen 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 no hayan sido procesadas por completo. Esto asegura una forma de sincronización entre la lógica de la aplicación residente en los flujos individuales.

Los acoplamientos se usan 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 acoplamientos hecho con

anterioridad. Reglas provistas por el acoplamiento son siempre referidas como dependencias de sincronización.

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 enlace de una orquestación, por lo tanto, puede ser clasificado como un complejo, y muy probablemente, actividades duraderas.

Orquestación y Coordinación

La orquestación es representada por BPEL4WS, puede utilizar completamente el manejador de contexto del cuadro *WS-Coordination* que se encuentra incorporado en el coordinador de tipos de *WS-BusinessActivity*. Esta especificación define el diseño de protocolos de coordinación para el completo soporte, actividades duraderas.

Orquestación y SOA

La lógica del proceso de negocio es la raíz de la automatización de las soluciones. La orquestación provee un modelo automatizado donde la lógica de procesos es centralizada con todo aún extensible y componible. A través del uso de la orquestación, el desarrollo de las soluciones orientadas al servicio llegan a ser intrínsecamente extensibles y adaptables. La 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 se 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, la orquestación sobre un nivel fundamental puede soportar la evolución diversificada de la empresa

Anexo 7 Cartelera de Actividades

Tiempo real para la creación de un proceso	
Tarea	Duración (días)
Gerencia de Procesos	105
1. Crear Proceso	105
1.1 Analizar	28
1.1.1 Identificar Requerimientos	8
1.1.1.1 Describir Requerimiento	5
1.1.1.2 Definir Prioridad	1
1.1.1.3 Evaluar Cartelera de Actividades	2
1.1.2 Levantar Información	20
1.1.2.1 Mapear al Marco Referencial del Proceso	5
1.1.2.2 Levantar Situación Actual	15
1.2 Diseñar	19
1.2.1 Estandarizar Proceso	2
1.2.2 Evaluar Riesgos	2
1.2.3 Realizar Modelo Propuesto	15
1.3 Modelar	8
1.3.1 Diagramar	5
1.3.2 Simular Prueba Funcional	1
1.3.3 Integrar Arquitectura	1
1.3.4 Simular Prueba Integral	1
1.4 Configurar	50
1.4.1 Completar UML	5
1.4.2 Construir Lógica de Negocios	10
1.4.3 Construir Interfaz	10
1.4.4 Construir Reportes	10
1.4.5 Configurar BPEL	10
1.4.6 Realizar Pruebas Unitarias	5

Tabla 7.1: Cartelera de Actividades.

Anexo 8 PostgreSQL 8.3.10

PostgreSQL 8.3.10 es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (*Berkeley Software Distribution*), su código fuente se encuentra disponible libremente. Utiliza el modelo cliente/servidor y multiprocesos para garantizar la estabilidad del sistema. (PostgreSQL-es, 2010).

Las mejoras de rendimiento más importantes de esta versión son (PostgreSQL-es, 2010):

- *Heap Organized Tuples* (HOT): eliminan hasta un 75% de la sobrecarga de mantenimiento en tablas frecuentemente actualizadas.
- *Checkpoints* extendidos y autoafinamiento del escritor en segundo plano, que reducen el impacto de los *checkpoints* en los tiempos de respuesta.
- Opción de confirmación (*commit*) asíncrono de transacciones, que permite tiempos de respuesta más breves para algunas transacciones.

Estos cambios, aceleran significativamente la tasa de procesamiento de transacciones, entre un 5 y un 30%, dependiendo de la carga de trabajo.

En la Figura 8.1 se pueden apreciar los componentes más importantes de un sistema PostgreSQL.

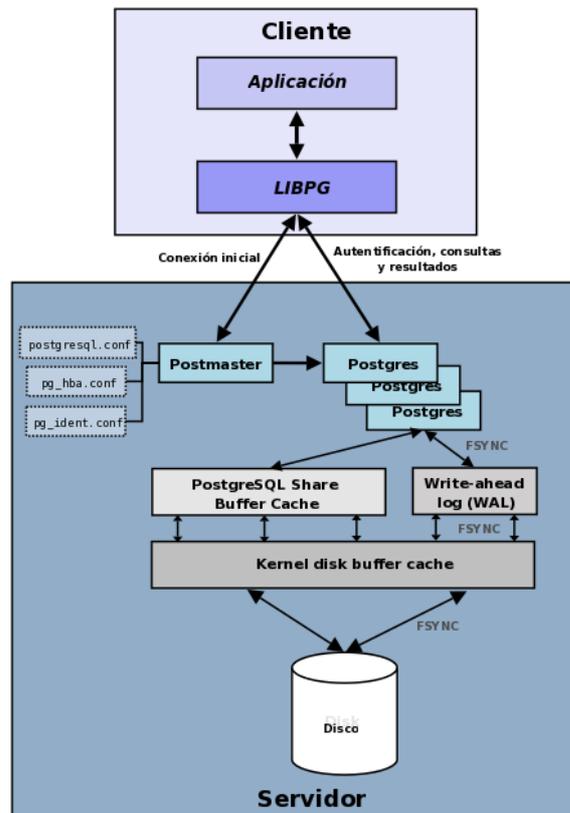


Figura 8.1: Componentes de un sistema PostgreSQL.

Fuente: PostgreSQL-es (2010)

A continuación se explican cada uno de los componentes:

- Aplicación cliente: es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP ó sockets locales.
- Demonio *postmaster*: es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargaran de autenticar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- Ficheros de configuración: los 3 ficheros principales de configuración utilizados por PostgreSQL son: `postgresql.conf`, `pg_hba.conf` y `pg_ident.conf`.

- Procesos hijos: se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- PostgreSQL *share buffer cache*: memoria compartida usada por PostgreSQL para almacenar datos en caché.
- *Write-Ahead Log (WAL)*: componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO).
- *Kernel disk buffer cache*: caché de disco del sistema operativo.
- Disco: disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione.

Anexo 9 Mule ESB

Mule ESB es una plataforma de integración que permite a los desarrolladores conectar aplicaciones de manera rápida y sencilla para intercambiar datos. Con Mule ESB se logra una fácil integración a sistemas existentes, independientemente de la tecnología que utilizan las aplicaciones, incluyendo JMS, *Web Services*, JDBC, HTTP y otras. (Mulesoft, 2011).

Entre las funciones que provee Mule ESB cabe destacar las siguientes (Mulesoft, 2011):

- Creación de servicios y *hosting*: permite exponer y alojar los servicios para ser reutilizados, Mule ESB actúa como un contenedor ligero.
- Mediación de servicios: separa la lógica de negocios de la mensajería, permite llamar a los servicios independientemente de su ubicación.
- Enrutamiento de mensajes: permite enrutar, filtrar, agregar y volver a agregar la secuencia de mensajes basándose en el contenido y las reglas.
- Transformación de datos: intercambia datos entre distintos formatos y protocolos de transporte.

La Figura 9.1 muestra la arquitectura de Mule ESB.

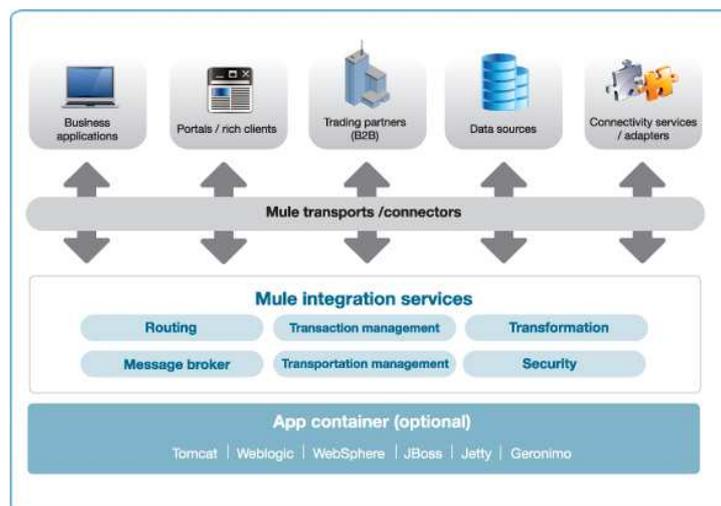


Figura 9.1: Arquitectura de Mule ESB.

Fuente: Mulesoft (2011)

Componentes de Mule ESB (Mulesoft, 2011):

- Application: la aplicación puede ser de cualquier tipo, como por ejemplo una aplicación web.
- Channel: puede ser cualquier método de comunicación de datos entre dos puntos.
- Message receiver: lee o recibe datos desde la aplicación. En Mule un receptor es sólo un elemento de un proveedor de transporte; Mule proporciona muchos transportes como JMS, SOAP, HTTP, TCP y otros.
- EndPoints: es en realidad un contenedor de configuración que se une a un conector, URI (*Uniform Resource Identifier*), transformadores, filtros e información de transacciones para proporcionar un adaptador de canal.
- Router: controla los eventos que son enviados y recibidos.
 - Inbound Routers: se utilizan para controlar cómo y qué eventos son recibidos por un componente suscrito a un canal. Se pueden utilizar para filtrar, agregar eventos dividir y volver a la secuencia antes de que sean recibidos por el componente UMO.
 - Outbound Routers: se utilizan para publicar mensajes/eventos a diferentes proveedores de acuerdo a las normas definidas en la configuración.
- Connector: sabe cómo enviar y recibir datos a través de un canal en particular.
- Transformers: convierte datos de un formato a otro.

La Figura 9.2 muestra cómo encajan los componentes de Mule.

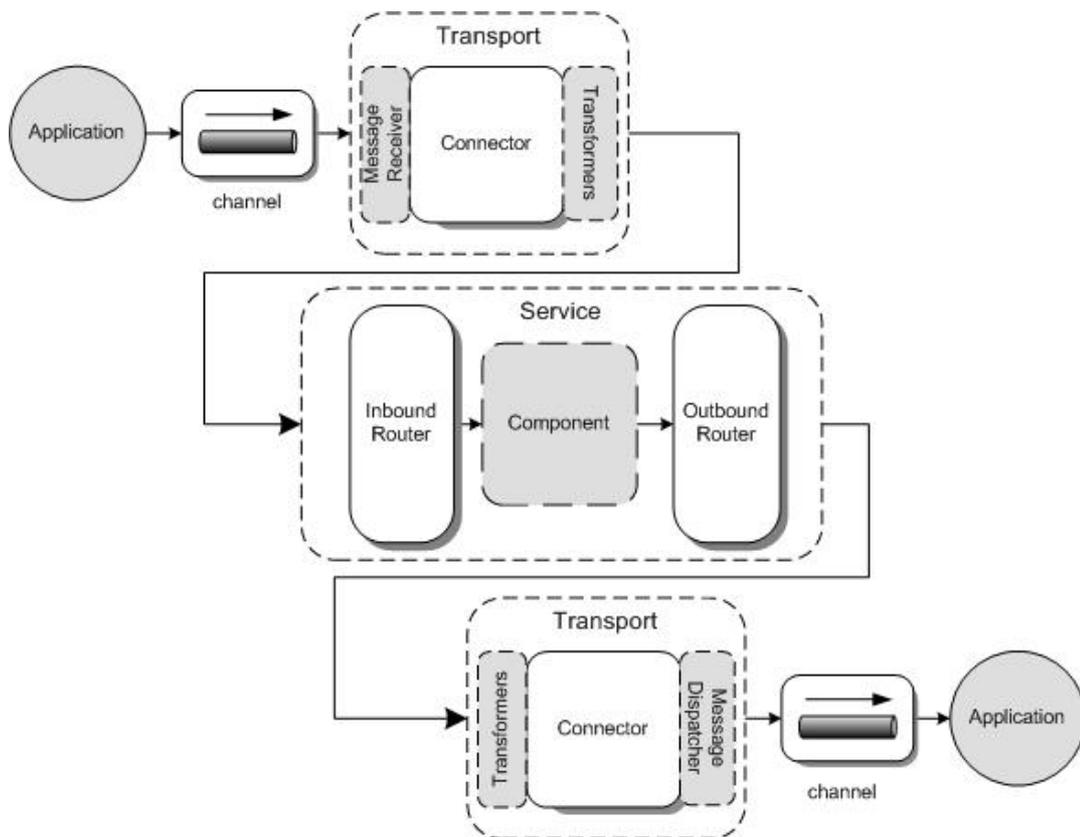


Figura 9.2: Componentes de Mule ESB.

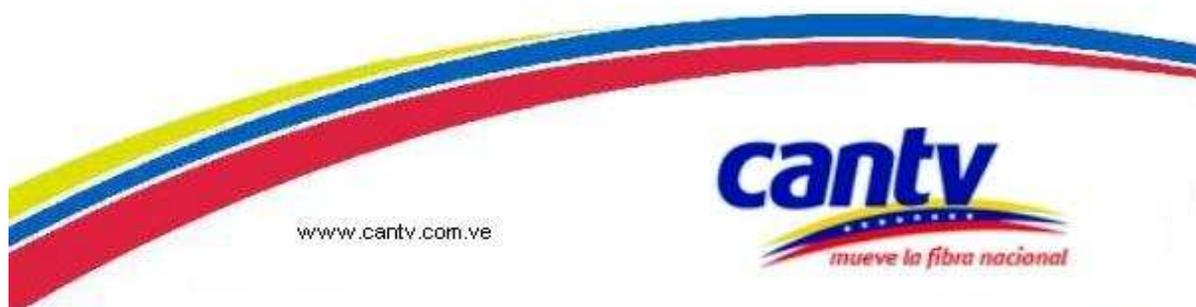
Fuente: Mulesoft (2011)

Los UMO (*Universal Message Objects*) son componentes autónomos simples que pueden enviar/recibir eventos desde/hacia otro UMO, también ejecutan lógica de negocio sobre un evento de llegada.

Sistema de Información para la Gestión de Versiones y Liberaciones de Software según ITIL v3

Manual
Administrativo

Mayo - 2012



Índice de contenidos.

Contenido	Página
Introducción	3
1. BPMS Intalio Community Edition 6.0.2	4 – 10
1.1 ¿Qué es Intalio BPM?	
1.2 Instalación de Intalio	
1.3 Intalio Designer	
1.4 Intalio Server	
2. BD PostgreSQL 8.3.10	10
2.1 ¿Qué es PostgreSQL?	
2.2 Instalación de PostgreSQL	
3. ESB Mule Studio 1.0.0	11
3.1 ¿Qué es Mule?	
3.2 Instalación de Mule Studio	
4. Ejecución del Proceso	11 – 12
5. Autenticación	12 – 14
6 Manejo de la Interfaz	14 – 15

Introducción

Este manual está dirigido al personal del Centro de Modelado de Procesos, encargada de diseñar los procesos operativos de la empresa.

El portal de procesos de CANTV para la administración de procesos de negocio BPM está construido bajo código abierto. A continuación se explicará el manejo del software utilizado en el Sistema de Información para la Gestión de Versiones y Liberaciones de Software y las actividades que se pueden realizar en ella.

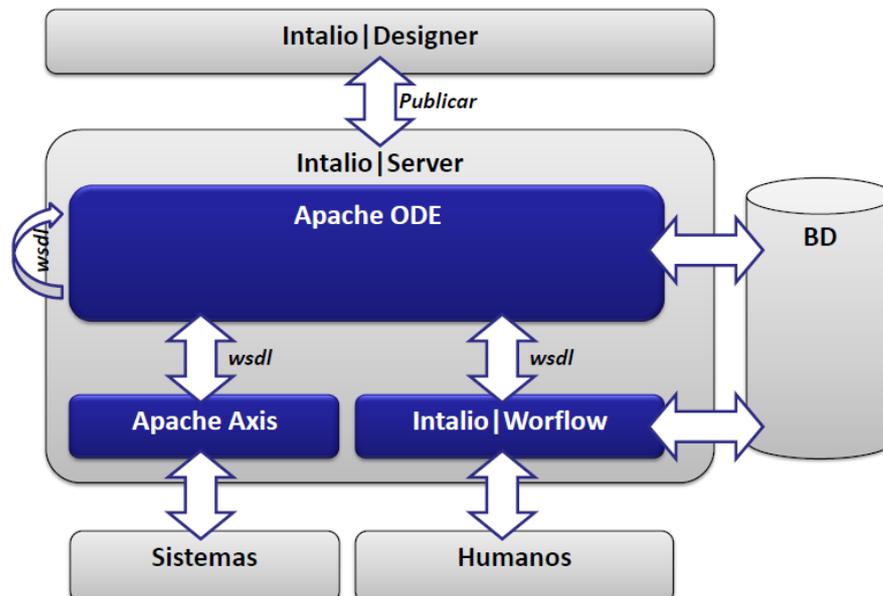
1. BPMS Intalio Community Edition 6.0.2

1.1 ¿Qué es Intalio|BPM?

Es una solución integral de BPMS construida bajo filosofía de código abierto, basada en estándares y optimizada para atender los distintos tipos de requerimientos del ámbito empresarial.

Intalio busca modelar y automatizar una secuencia de actividades que coordine las tareas de sistemas, humanos y procesos para lograr la representación de un proceso de negocio. Para ello se utiliza BPMN (Business Process Modeling Notation) que es una notación gráfica estandarizada para el modelado de procesos de negocio. Esta notación es simple, pero rica semánticamente, que permite identificar roles y responsabilidades, secuencia de actividades en el tiempo y manejar distintas perspectivas.

Componentes bases



BPMS Intalio Community Edition 6.0.2 Cont...

1.2 Instalación del software

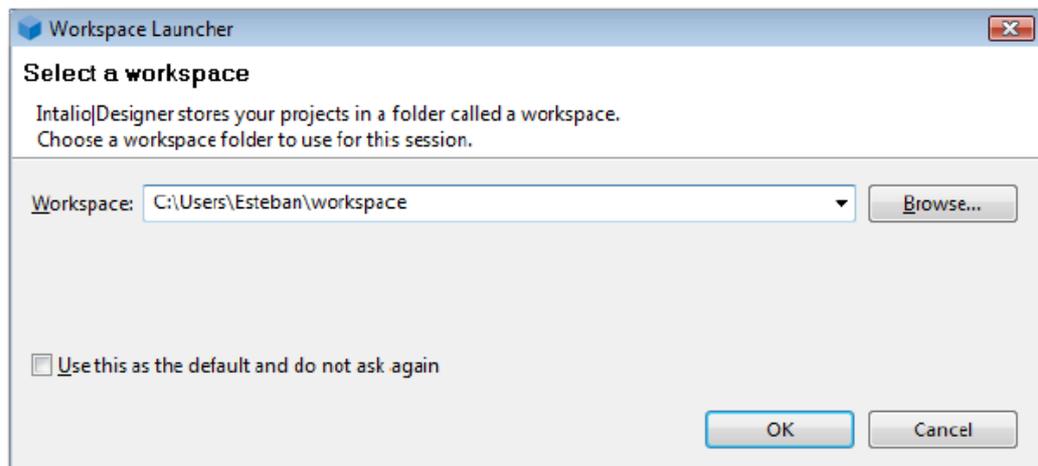
Para ello es necesario seguir estos pasos:

1. Instalar JDK o JRE
2. Configurar variable de entorno JAVA_HOME o JRE_HOME
3. Descomprimir Intalio|Designer
4. Iniciar con designer.exe o designer
5. Descomprimir Intalio|Server
6. Iniciar con `/bin/startup.bat` o `/bin/startup.sh`

1.3 Intalio|Designer

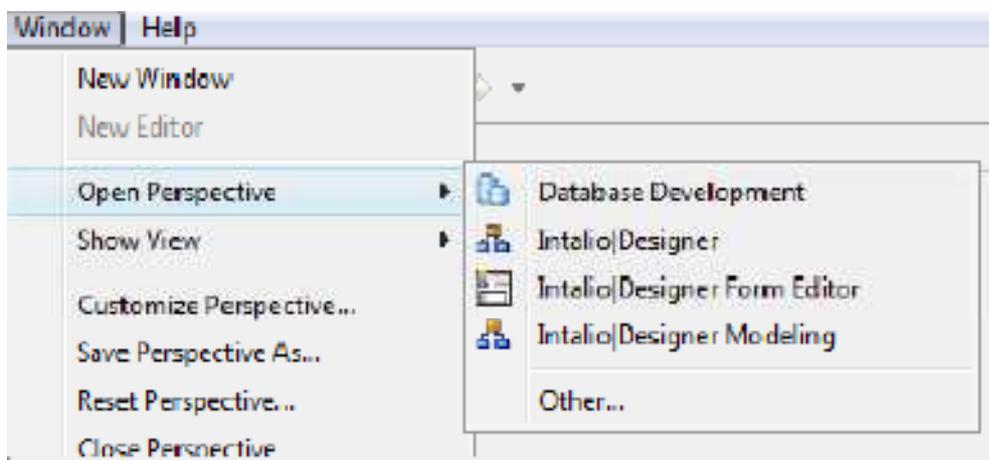
Es una herramienta de modelado, basada en Eclipse. Maneja 3 conceptos básicos:

- Espacios de trabajo

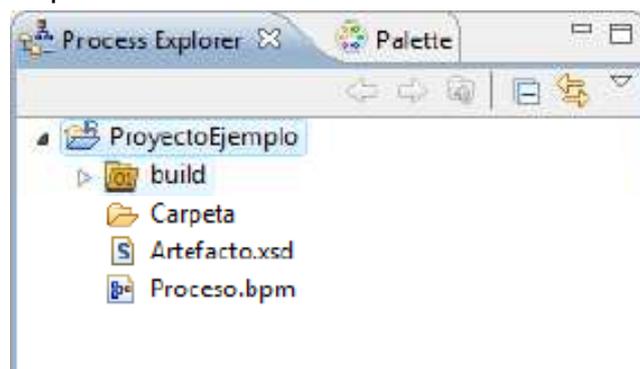


BPMS Intalio Community Edition 6.0.2 Cont...

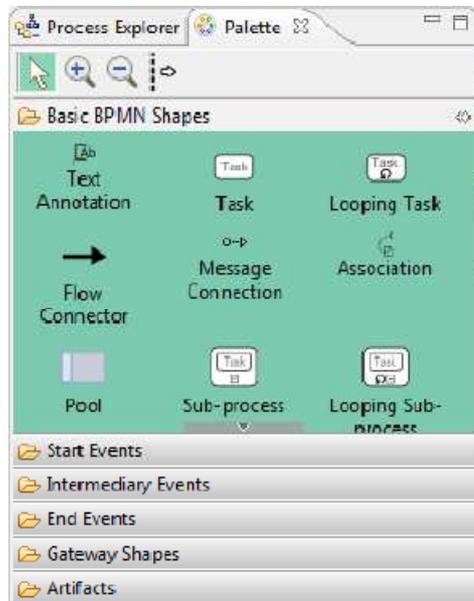
- Perspectivas:
 - Intalio|Designer
 - Intalio|Designer Form Editor
 - Intalio|Designer Modeling



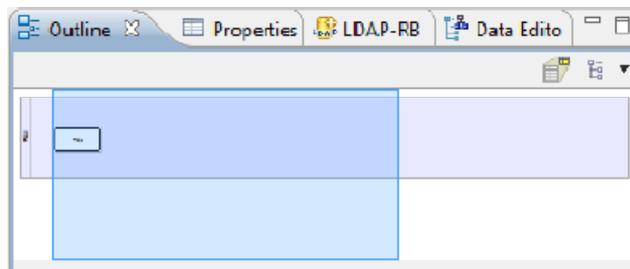
- Vistas
 - Process Explorer



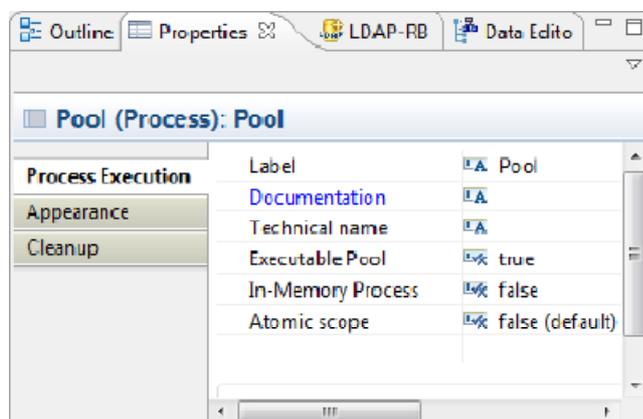
◦ Palette



◦ Outline

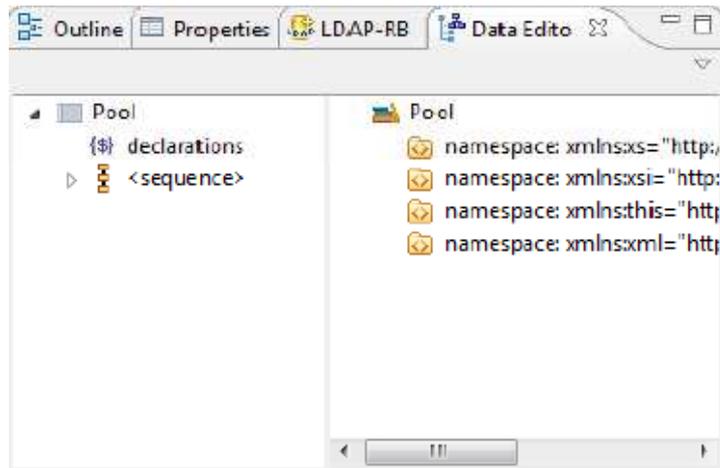


◦ Properties

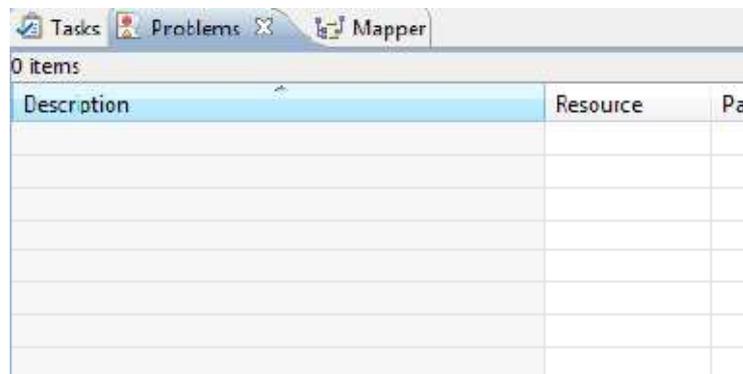


BPMS Intalio Community Edition 6.0.2 Cont...

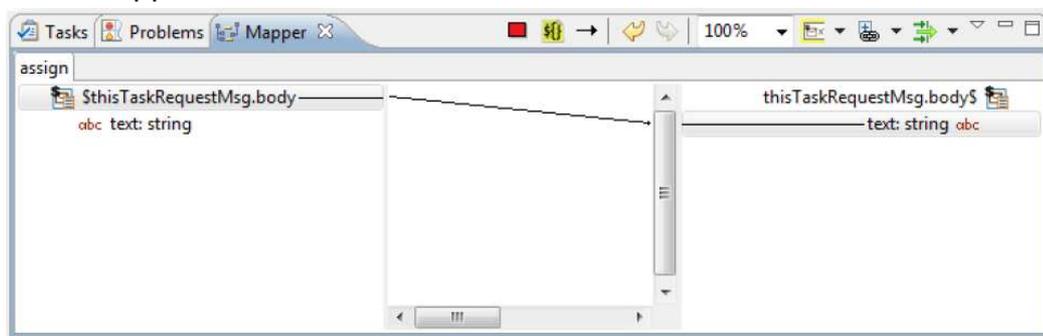
- Data Editor



- Problems

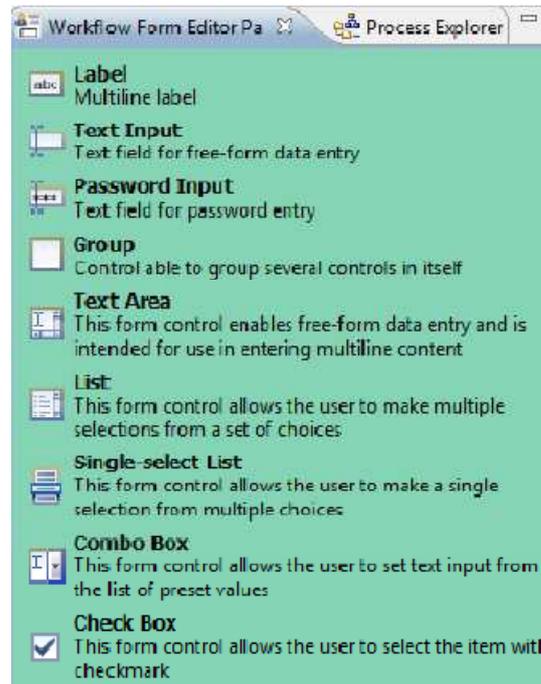


- Mapper

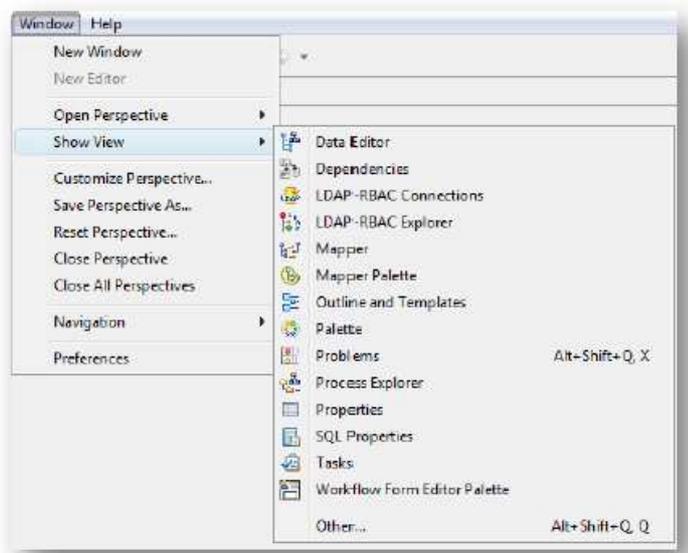


BPMS Intalio Community Edition 6.0.2 Cont...

- Workflow form editor palette

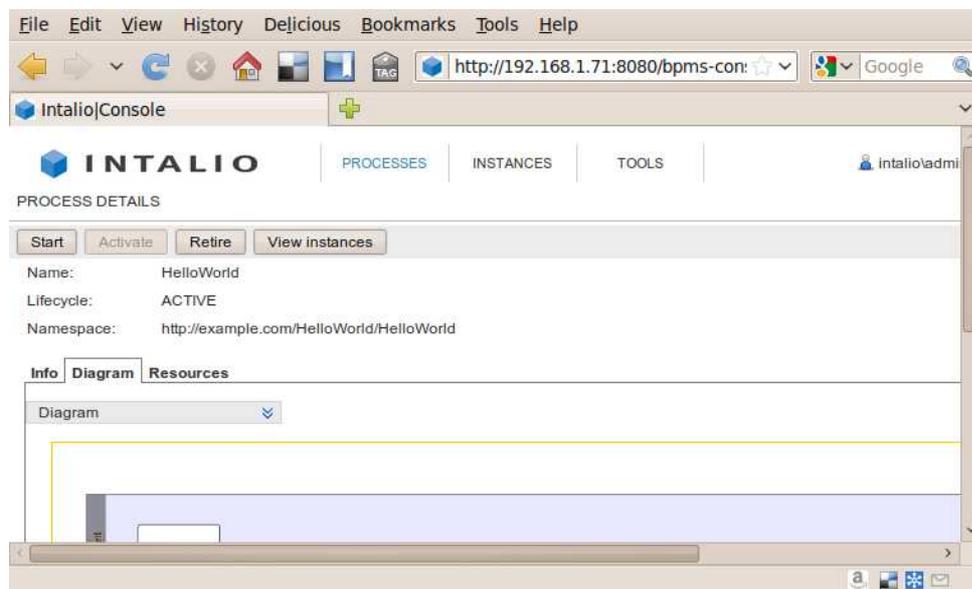


- Otras vistas



1.4 Intalio|Server

Es un servidor nativo de BPEL 2.0 basado en la arquitectura J2EE (*Java 2 Platform Enterprise Edition*) 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.



2. BD PostgreSQL 8.3.10

2.1 ¿Qué es PostgreSQL?

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (*Berkeley Software Distribution*), su código fuente se encuentra disponible libremente.

2.2 Instalación

1. Descomprimir el archivo
2. Ejecutar el archivo .msi postgresql-8.3.10
3. Descargar el driver de conexión postgresql-8.3-606.jdbc4 y añadirlo en la carpeta de common/lib de Intalio|Server
4. Hacer clic en Start Server
5. Iniciar pgAdmin III

3. ESB Mule Studio 1.0.0

3.1 ¿Qué es Mule?

Mule ESB es una plataforma de integración que permite a los desarrolladores conectar aplicaciones de manera rápida y sencilla para intercambiar datos. Con Mule ESB se logra una fácil integración a sistemas existentes, independientemente de la tecnología que utilizan las aplicaciones, incluyendo JMS, *Web Services*, JDBC, HTTP y otras.

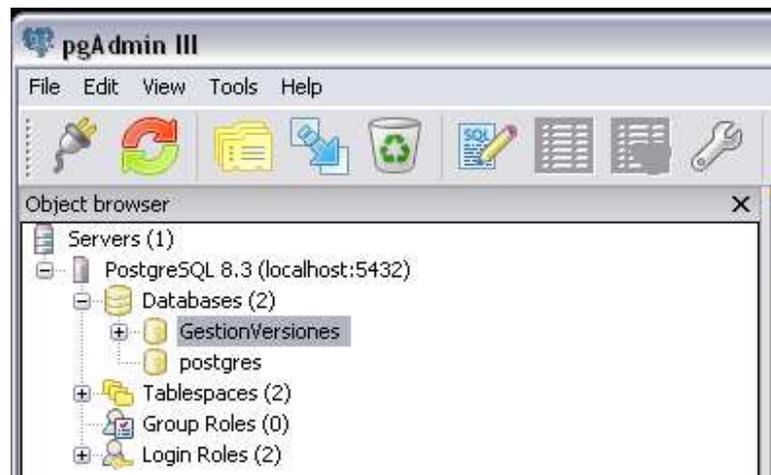
3.2 Instalación

1. Descomprimir el archivo
2. Ejecutar Mule Studio 1.0.0

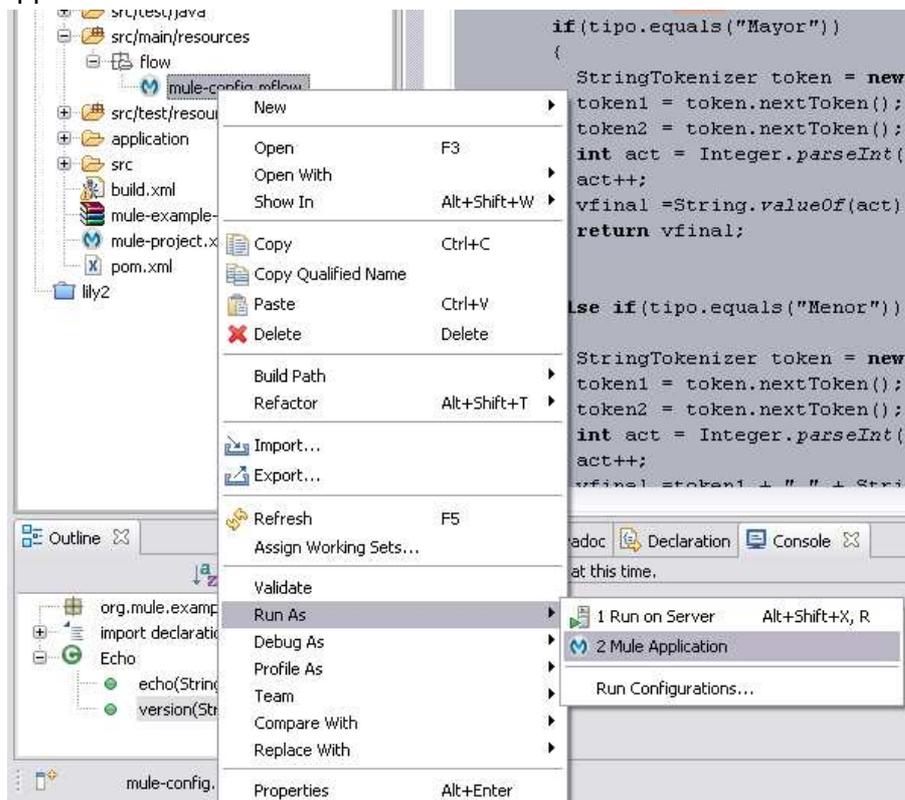
4. Ejecución

Una vez que el proceso esté desplegado en el servidor de Intalio, ejecutamos PostgreSQL y Mule:

1. En PostgreSQL hay que buscar la BD de Gestión de Versiones



2. En Mule hacer clic derecho sobre el archivo .mflow → Run As → Mule Application



Luego es necesario abrir el navegador y copiar la siguiente dirección: localhost:8080/ui-fw/login.htm

5. Autenticación

Para poder ingresar al Sistema de Información para la Gestión de Versiones y Liberaciones es necesario que el usuario se encuentre definido en el archivo security.xml tomando en cuenta la permisología permitida de acuerdo a su rol.

El archivo security.xml se encuentra en la carpeta var\config de Intalio|Server. Los siete roles y sus respectivos usuarios creados son:

Autenticación Cont...

Rol	Usuario	Contraseña
AdministradorProcesoGV	Ltsang	Pass
EquipodePruebas	EqPrueba	Pass
EquipodeDesarrollo	EqDesarrollo	Pass
GestioneCambios	GestCambio	Pass
AdministradorGV	AdmGV	Pass
ProveedoresExternos	ProvExt	Pass
GrupoUsuariosFinales	GrupUS	Pass

Pasos para definir más usuarios

1. En el archivo security.xml ubicar la sección: <realm identifier="gv">
2. Por cada usuario que se desea crear se debe definir lo siguiente:
<user identifier="login_del_usuario">
 <name>nombre y apellido del usuario</name>
 <password>contraseña</password>
 <assignRole>rol</assignRole>
</user>

Pasos para ingresar al sistema

1. Abrir el navegador
2. Copiar en la barra de direcciones: localhost:8080/ui-fw/login.htm
3. Ingresar usuario y contraseña
4. Hacer clic en Log In



LOGIN FORM

Username	<input type="text"/>
Password	<input type="password"/>
	<input type="checkbox"/> Auto login
	<input type="button" value="Log In"/>

Una vez ingresado al sistema se observa la siguiente pantalla mostrando los procesos que se encuentran en el servidor:

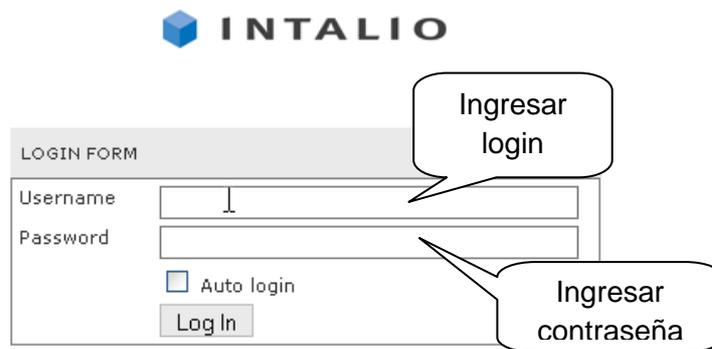


En la esquina superior derecha se puede observar bajo qué usuario se ingresó al sistema



6. Manejo de la Interfaz

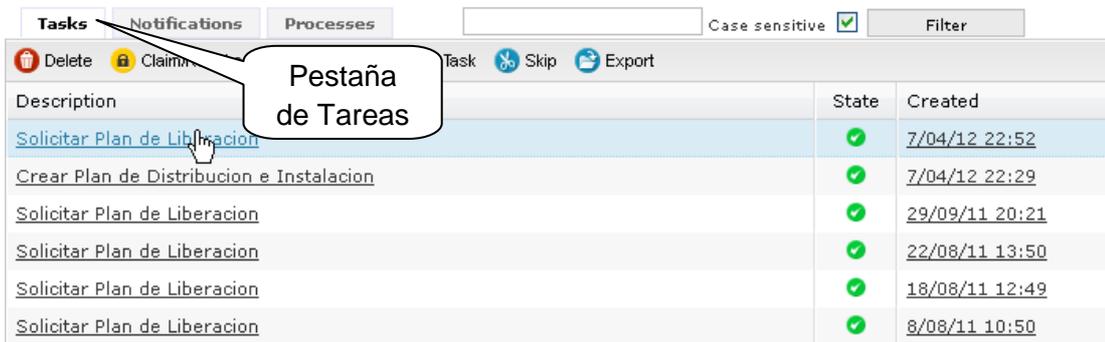
Pantalla de Autenticación



El Administrador del Proceso de Gestión de Versiones posee permisología para completar cualquier tarea, ejecutar cualquier proceso y ver todas las notificaciones que se encuentre activas.

Vista de Tareas

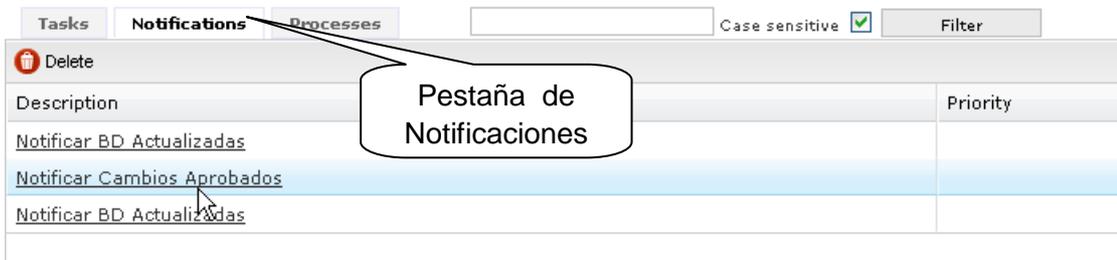
Muestra las tareas pendientes.



Description	State	Created
Solicitar Plan de Liberacion	✓	7/04/12 22:52
Crear Plan de Distribucion e Instalacion	✓	7/04/12 22:29
Solicitar Plan de Liberacion	✓	29/09/11 20:21
Solicitar Plan de Liberacion	✓	22/08/11 13:50
Solicitar Plan de Liberacion	✓	18/08/11 12:49
Solicitar Plan de Liberacion	✓	8/08/11 10:50

Vista de Notificaciones

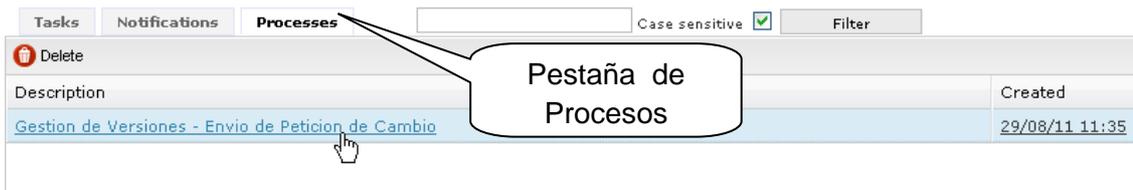
Muestra las notificaciones generadas por el sistema.



Description	Priority
Notificar BD Actualizadas	
Notificar Cambios Aprobados	
Notificar BD Actualizadas	

Vista de Procesos

Contiene los procesos desplegados en el Servidor de Intalio



Description	Created
Gestion de Versiones - Envio de Peticion de Cambio	29/08/11 11:35

Sistema de Información para la Gestión de Versiones y Liberaciones de Software según ITIL v3

Manual de Usuario

Mayo - 2012



Índice de contenidos.

Contenido	Página
Introducción	3
1. Autenticación	4 – 5
2. Manejo de la Interfaz	5 – 6
3. Equipo de Pruebas	7
4. Equipo de Desarrollo	8 – 11
5. Gestión de Cambios	11 – 14
6. Administrador Proceso Gestión de Versiones	14 – 19
7. Administrador Gestión de Versiones	19 – 21
8. Proveedores Externos	21 – 22
9. Grupo de Usuarios Finales	22

Introducción

Este manual está dirigido a los usuarios del Sistema de Información para la Gestión de Versiones y Liberaciones de Software.

El portal de procesos de CANTV está construido bajo código abierto. A continuación se explicará el manejo del Sistema de Información para la Gestión de Versiones tomando en cuenta los distintos roles involucrados y sus actividades.

1. Autenticación

Para autenticarse es necesario que la persona cuente con un usuario y su respectiva contraseña. A cada usuario se le asigna la permisología de acuerdo a su rol.

La siguiente tabla muestra los usuarios creados actualmente.

Rol	Usuario	Contraseña
AdministradorProcesoGV	Ltsang	Pass
EquipodePruebas	EqPrueba	Pass
EquipodeDesarrollo	EqDesarrollo	Pass
GestioneCambios	GestCambio	Pass
AdministradorGV	AdmGV	Pass
ProveedoresExternos	ProvExt	Pass
GrupoUsuariosFinales	GrupUS	Pass

Si se requieren mas cuentas de usuario contacte al administrador del sistema.

Pasos para ingresar al sistema

5. Abrir el navegador
6. Copiar en la barra de direcciones: localhost:8080/ui-fw/login.htm
7. Ingresar usuario y contraseña
8. Hacer clic en Log In



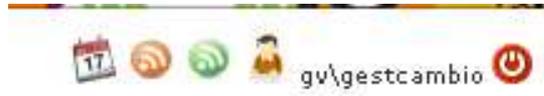
LOGIN FORM

Username

Password

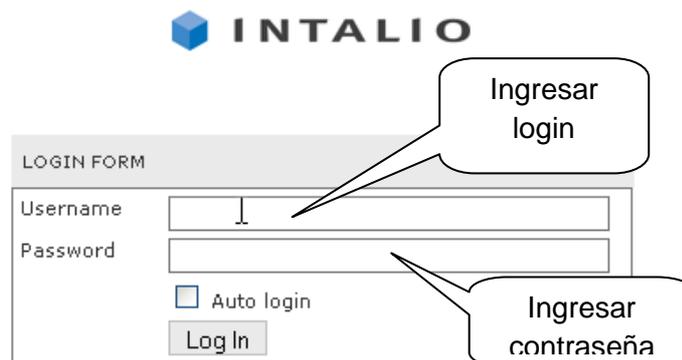
Auto login

En la esquina superior derecha se puede observar bajo qué usuario se ingresó al sistema



2. Manejo de la Interfaz

A continuación se puede observar la pantalla de autenticación del sistema, para ingresar es necesario ingresar el nombre de usuario o login y la contraseña



INTALIO

LOGIN FORM

Username

Password

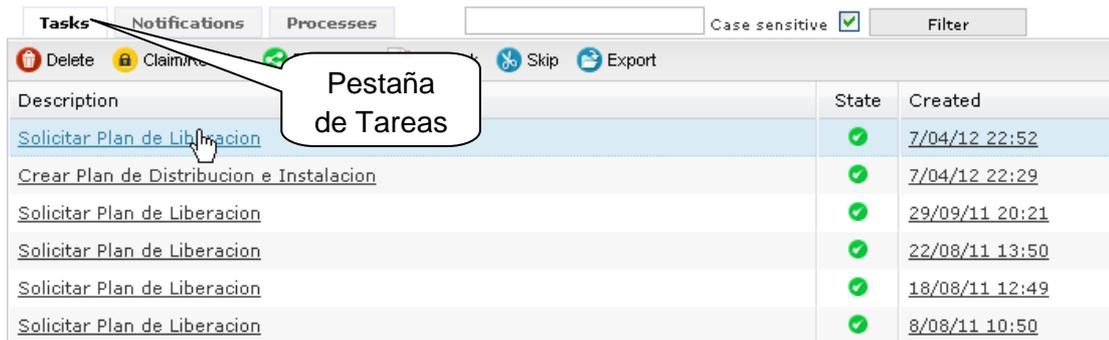
Auto login

Callout 1: Ingresar login (points to the Username field)

Callout 2: Ingresar contraseña (points to the Password field)

Vista de Tareas

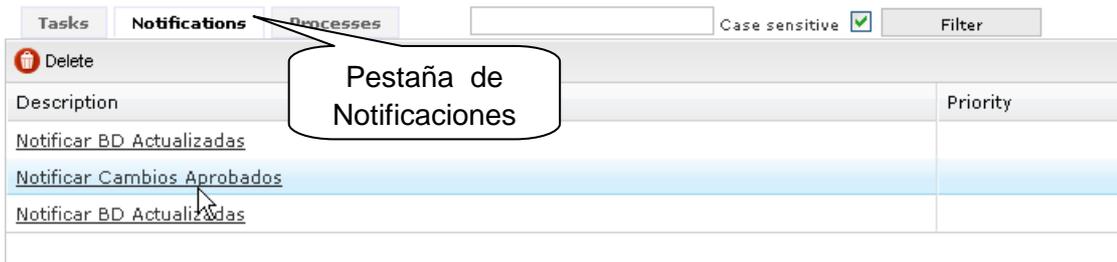
Muestra las tareas a ser completadas por el usuario.



Description	State	Created
Solicitar Plan de Liberacion	✓	7/04/12 22:52
Crear Plan de Distribucion e Instalacion	✓	7/04/12 22:29
Solicitar Plan de Liberacion	✓	29/09/11 20:21
Solicitar Plan de Liberacion	✓	22/08/11 13:50
Solicitar Plan de Liberacion	✓	18/08/11 12:49
Solicitar Plan de Liberacion	✓	8/08/11 10:50

Vista de Notificaciones

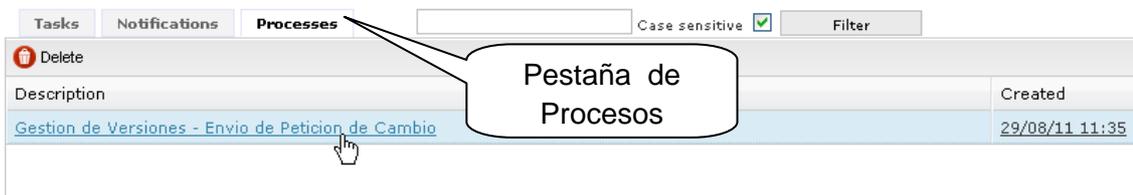
Muestra las notificaciones dirigidas al usuario.



Description	Priority
Notificar BD Actualizadas	
Notificar Cambios Aprobados	
Notificar BD Actualizadas	

Vista de Procesos

Contiene los procesos desplegados en el Servidor de Intalio.



Description	Created
Gestion de Versiones - Envio de Peticion de Cambio	29/08/11 11:35

3. Equipo de Pruebas

El Equipo de Pruebas tiene las siguientes responsabilidades:

- Administrar los recursos de prueba
- Mantener el entorno de prueba en un estado de alerta y preparar el entorno de prueba para las pruebas de los próximos lanzamientos de acuerdo con el calendario de lanzamientos
- Completar todas las pruebas necesarias, incluidas las pruebas del plan de back-out
- Reunir todos los resultados de las pruebas
- Comunicar los resultados de las pruebas, incluidos los errores conocidos
- Contribuir en las decisiones para la liberación

Vista Pruebas Integrales

Realizar Pruebas Integrales

Fecha de ejecución: 9/20/2011 

Proyecto: Actualización de SW/ 7.3.1 Version: 8.0 Código Back Out: CODBO2

Entorno: TI Elementos a probar: colocar link con documento indicando pruebas a realizar Procedimientos de Back Out: restablecer el servicio si la version creada no cumple con las expectativas

Resultados: resultados satisfactorios Errores: 0

Está compuesto por los campos:

Fecha de ejecución: indicar la fecha en que se realizarán las pruebas integrales.

Resultados: contiene los resultados arrojados por las pruebas realizadas.

Errores: indicar el número de errores encontrados durante las pruebas y una breve explicación de las mismas.

3. Equipo de Desarrollo

El Equipo de Desarrollo se encarga de:

- Desarrollar una aplicación de software integrada de acuerdo al RFC
- Entregar la versión con su respectiva documentación de apoyo para la construcción de entornos de pruebas

Vista Forma de Desarrollo

Forma de Desarrollo

Proyecto: Actualización de SW 7.3.1 Tipo: Software Version: 8.0

Requisitos:

Indique la forma de desarrollo:

Campos:

Indique la forma de desarrollo: puede ser realizada en casa, es decir desarrollada en la misma empresa ó adquirida a proveedores externos.

Vista Desarrollar Versión

Desarrollar Version

Fecha de desarrollo:

Costo:

Código RFC: Proyecto: Actualización de SW 7.3.1 Tipo: Software Version: 8.0

Software

Aplicacion Sistema Operativo

Licencia:

Enlace script de instalacion:

Enlace manuales:

Hardware

Enlace manuales:

Campos:

Fecha de desarrollo: indicar la fecha en que la versión fue desarrollada.

Costo: precio estimado del desarrollo de la versión.

Licencia: indicar el código de licencia o el enlace donde se encuentre la licencia.

Enlace script de instalación: indicar la ruta del script.

Enlace manuales: indicar la ruta donde se encuentran los manuales.

Vista Plan de Pruebas Unitarias

Plan de Pruebas Unitarias

Código RFC: COD2 Proyecto: Actualización de SW 7.3.1 Versión: 8.0

Fecha de reunión: 9/13/2011
Fecha de emisión: 9/14/2011

Entorno:	TI	Interacción con otros sistemas:	NA
Requerimientos y configuración de HW:	NA	Requerimientos y configuración de SW:	hay que obtener la version y proceder a instalarla
Configuración de la BD:	realizar un back up para la version anterior	Instalación:	se realiza de manera manual
Ejecución:	se realiza de manera manual	Verificación:	probar que la version realice todas las
Componentes y pruebas:			

Campos:

Fecha de reunión: indicar la fecha en que se reunieron las partes involucradas para realizar el Plan de Pruebas Unitarias.

Fecha de emisión: una vez realizada la reunión para acordar el plan, es necesario emitir el plan.

Entorno: indicar en donde se van a realizar las pruebas unitarias.

Interacción con otros sistemas: especificar si la versión a probar se comunica con otro sistema.

Requerimientos y configuración de HW: indicar si es necesario realizar alguna configuración o requerimiento de HW adicional.

Requerimientos y configuración de SW: indicar si es necesario algún requerimiento o configuración de SW adicional.

Configuración de la BD: indicar si se requiere de una nueva configuración de BD.

Instalación: describir cómo instalar la versión.

Ejecución: indicar los pasos para ejecutar la versión.

Verificación: describir la forma en que se debe realizar el proceso de verificación.

Componentes y pruebas: listar los componentes y las pruebas a las cuales deben ser sometidas.

Vista Realizar Pruebas Unitarias

Realizar Pruebas Unitarias Fecha de ejecución: 9/15/2011

Proyecto: Actualización de SW 7.3.1 Version: 8.0

Componentes y pruebas:

Verificación:

Resultados:

Errores:

Campos:

Fecha de ejecución: indicar la fecha en que se realizarán las pruebas unitarias.

Resultados: contiene los resultados arrojados por las pruebas realizadas.

Errores: indicar el número de errores encontrados durante las pruebas y una breve explicación de las mismas.

Vista Realizar Implementación

Realizar Implementacion Fecha de ejecucion: 9/27/2011

Proyecto: Actualización de SW 7.3.1 Version: 8.0

Forma de implementacion:

Acciones por sitio:

Resultados:

Observaciones:

Campos:

Fecha de ejecución: indica la fecha en que se realizó la implementación.

Resultados: resultados obtenidos al hacer la implementación.

Observaciones: comentarios adicionales.

Vista Realizar Marcha atrás

Realizar Back Out

Fecha de ejecución: 9/29/2011 

Proyecto: Actualización de SW 7.3.1 Version: 8.0 Código Back Out: CODB02

Procedimientos de Back Out: reestablecer el servicio si la version creada no cumple con las expectativas

Enfoque del back out: Reestablecer Servicio 

Resultados: | 

Campos:

Fecha de ejecución: fecha en que se realizó la marcha atrás.

Enfoque del back out: existen dos enfoques: restablecer el servicio o aplicar medidas de contingencia.

Resultados: resultados obtenidos al realizar la marcha atrás.

5. Gestión de Cambios

La Gestión de Versiones y Liberaciones se comunica con el proceso de Gestión de Cambios. Para generar una nueva versión es necesario que Gestión de Cambio emita un documento indicando los cambios solicitados, es por ello que representa el iniciador del proceso.

Gestión de Cambios se encarga de:

- Suministrar el documento de petición de cambio (RFC)
- Aprobar la liberación y los cambios de la versión

Vista Petición de Cambio (RFC)

Peticion de cambio (RFC)

Código RFC: Propietario del cambio: Fecha de emisión:

Cambio Solicitado

Prioridad: Tipo: Tipo SW:

Accion: Tipo de version: Version actual:

Requisitos: Razones para implementar el cambio:

Beneficios: Consecuencias si no se implementa el cambio:

Cls afectados: Areas de negocio afectadas:

Campos:

Código RFC: identificador único para las peticiones de cambio.

Propietario del cambio: nombre de la persona que solicita el cambio.

Fecha de emisión: fecha en que se emite el RFC.

Prioridad: prioridad del RFC, puede ser: alta, media o baja.

Tipo: tipo cambio, puede ser de tipo hardware o software.

Tipo SW: si el cambio es de tipo software indicar si es de tipo: aplicación, sistema operativo o documentación.

Acción: indicar si se va a modificar una versión existente o si hay que construir una nueva versión.

Tipo de versión: indicar si la versión es mayor, menor o de emergencia.

Versión actual: en él se debe ingresar el número de la versión actual que está implementado en el entorno de TI.

Requisitos: son los requerimientos del cambio solicitado.

Razones para implementar el cambio: explicar por qué es importante realizar el cambio solicitado.

Beneficios: indicar los beneficios que se obtiene al realizar el cambio.

Consecuencias si no se implementa el cambio: indicar los problemas que surgen si no se realiza el cambio especificado.

Cls afectados: listar los elementos de configuración afectados por el cambio.

Áreas de negocio afectadas: indicar las áreas afectadas por el cambio.
Servicios afectados: servicios afectados por el cambio.
Código de Back out: identificador único para el back out.
Procedimientos de back out: explicar cómo se llevaría a cabo el back out si la implementación del cambio falla.
Recursos necesarios: recursos materiales necesarios.
Personal requerido: personal necesario para realizar el cambio.
Riesgos: riesgos al introducir el cambio.
Fecha estimada para la implementación: fecha estimada en que se tendrá lista la versión para realizar su implementación.

Vista Solicitar aprobación de liberación

Solicitar Aceptacion de Liberacion

Proyecto: Actualización de SW 7.3.1 Version: 8.0

Aprobar Aceptacion de Liberacion: Fecha:

Campos:

Aprobar aceptación de liberación: decidir la aprobación: si o no.
Fecha: fecha de solicitud de aprobación.

Vista Solicitar aprobación de cambios

Solicitar Aprobacion de Cambios

Proyecto: Actualización de SW 7.3.1 Version: 8.0

Nuevas funcionalidades:

Mejoras:

Aprobar Cambios: Fecha:

Campos:

Aprobar cambios: Gestión de Cambios indicar si aprueba o no los cambios que introduce la versión.

Fecha: fecha en que se realiza la solicitud de aprobación de cambios.

6. Administrador Proceso Gestión de Versiones

Se encarga de:

- Desarrollar una estrategia de liberación
- Publicar un plan de pruebas
- Publicar el plan de implementación
- Realizar revisiones del plan de liberación
- Recibir, registrar, calificar, asignar todas las solicitudes de liberación
- Desarrollar programas de instalación
- Publicar el plan de comunicación y adiestramiento
- Participar en las reuniones del CAB
- Solicitud de actualización de CI
- Asegura que todos los clientes, personal de apoyo y personal de Service Desk estén capacitados y provistos de la información y documentación pertinente

Vista Crear Plan de Liberación

Plan de Liberación		Fecha de reunion:	
Codigo RFC: COD2		9/9/2011	
		Fecha de emision:	
		9/12/2011	
Prioridad: Alta	Tipo: Software	Accion: Nueva_Version	Version actual: 7.3.1
Requisitos:	Se requiere que la nueva version contenga todas las modificaciones de las versiones asociadas.		
Informacion del Item de Liberacion			
Proyecto:	Tipo de version:	Numeracion de la version:	
Actualización de SW 7.3.1	Mayor	8.0	
Personas Involucradas		Grupos Afectados por la Liberacion	
Propietario del cambio:	Telefono:	Edificios/Lugares:	
Alberto Mejia			
Responsable de la liberacion:	Telefono:	Centrales:	

Prohibida la reproducción total o parcial de este documento

Campos:

Fecha de reunión: fecha en que se reunieron las partes involucradas para fijar los puntos del plan.

Fecha de emisión: fecha en que se publica el plan.

Proyecto: nombre del proyecto que introduce el cambio.

Numeración de la versión: número de la versión a construir.

Propietario del cambio: persona que solicita el cambio.

Responsable de la liberación: persona encargada de la liberación de la versión.

Teléfono: teléfono del propietario del cambio y del responsable de la liberación.

Roles y responsabilidades: asignar roles y responsabilidades del personal.

Edificios/Lugares: listar edificios y/o lugares afectados por la liberación.

Centrales: listar centrales afectados.

Método de liberación: explicar el procedimiento mediante el cual los usuarios podrán acceder a la liberación.

Fecha estimada para la liberación: fecha en que se tiene previsto liberar la nueva versión.

Riesgos: resumir los riesgos de los planes y riesgos de las estrategias de mitigación.

Documentación necesaria: guías, panfletos.

Vista Plan de Pruebas Integrales

Plan de Pruebas Integrales			Fecha de reunion:
Codigo RFC: CDD2	Proyecto: Actualización de SW 7.3.1	Version: 8.0	9/16/2011
			Fecha de emisión: 9/19/2011

Entorno: TI	Configuración y datos de prueba: colocar link donde se encuentren los datos de prueba
Elementos a probar: colocar link con documento indicando pruebas a realizar	Riesgos: interrupcion del servicio
Condicion de pruebas/Criterios de reanudacion: susp: fallas electricas y/o de red reanud: servicio	

Campos:

Fecha de reunión: fecha en que se reunieron las partes involucradas para fijar los puntos del plan.

Fecha de emisión: fecha en que se publica el plan.

Entorno: indicar en dónde se van a realizar las pruebas integrales.

Configuración y datos de pruebas: indicar si es necesario realizar alguna configuración así como también los datos de pruebas.

Elementos a probar: listar los elementos a probar.

Riesgos: resumir los riesgos del plan.

Suspensión de pruebas/Criterios de reanudación: prever situaciones en las cuales es necesario suspender las pruebas y el ambiente para reanudar las mismas.

Vista Plan de Pruebas de Aceptación de Usuario

Plan de Pruebas de Aceptación de Usuario

Codigo RFC: COD2	Proyecto: Actualización de SW 7.3.1	Version: 8.0	Fecha de reunion: 9/20/2011 Fecha de emision: 9/20/2011
------------------	-------------------------------------	--------------	--

Entorno: TI	Configuración y datos de prueba: colocar link con datos de pruebas
Casos de prueba: colocar link con casos de prueba	Riesgos: interrupción del servicio
Suspensión de pruebas/Criterios de reanudación: susp: fallas electricas y/o de red reanud: servicios	

Campos:

Fecha de reunión: fecha en que se reunieron las partes involucradas para fijar los puntos del plan.

Fecha de emisión: fecha en que se publica el plan.

Entorno: indicar en dónde se van a realizar las pruebas de aceptación.

Configuración y datos de pruebas: indicar si es necesario realizar alguna configuración así como también los datos de pruebas.

Elementos a probar: listar los elementos a probar.

Riesgos: resumir los riesgos del plan.

Suspensión de pruebas/Criterios de reanudación: prever situaciones en las cuales es necesario suspender las pruebas y el ambiente para reanudar las mismas.

Vista Plan de Implementación

Plan de Implementacion

Codigo RFC: COD2 Proyecto: Actualización de SW 7.3.1 Version: 8.0

Fecha de reunion: 9/23/2011
Fecha de emision: 9/23/2011

Cls afectados: version 7.3.1

Cls a instalar: link con cls a instalar

Cls a desinstalar: link con cls a desinstalar

Forma de implementacion: Automatico

Impacto:

Recursos necesarios: laboratorios y personal encargado

Acciones por sitio: manera automatica entonces en cada sitio se van a realizar las mismas actividades de instalacion

Riesgos: interrupcion del servicio

Campos:

Fecha de reunión: fecha en que se reunieron las partes involucradas para fijar los puntos del plan.

Fecha de emisión: fecha en que se publica el plan.

Cls a instalar: elementos de configuración a instalar.

Cls a desinstalar: elementos de configuración a eliminar.

Forma de implementación: seleccionar entre: completa, en fases, empujar, halar, automático, manual.

Impacto: cómo la versión puede afectar al entorno y a los usuarios.

Recursos necesarios: recursos materiales.

Acciones por sitio: describir las acciones de implementación por sitio.

Riesgos: resume los riesgos del plan.

Vista Plan de Comunicación y Adiestramiento

Plan de Comunicación y Adiestramiento			Fecha de reunion: <input type="text" value="9/27/2011"/>
			Fecha de emision: <input type="text" value="9/27/2011"/>
Codigo RFC: CDD2	Proyecto: Actualización de SW 7.3.1	Version: 8.0	
----- Eventos -----			
Pruebas integrales: <input type="text" value="9/20/2011"/>	Liberacion de version: <input type="text" value="9/23/2011"/>		
Pruebas aceptacion de usuario: <input type="text" value="9/21/2011"/>	Implementacion: <input type="text" value="9/27/2011"/>		
----- Comunicacion -----			
Nuevas funcionalidades: <input type="text" value="funcionalidad que permite que los usuarios accedan a los datos facilmente por medio del buscador"/>	Mejoras: <input type="text" value="la interfaz es un poco mas sencilla"/>		
Publico: <input type="text" value="personas involucradas en el negocio"/>	Medios de comunicacion y actividades: <input type="text" value="actividad"/>		

Campos:

Fecha de reunión: fecha en que se reunieron las partes involucradas para fijar los puntos del plan.

Fecha de emisión: fecha en que se publica el plan.

Pruebas integrales, pruebas aceptación de usuario, liberación de versión e implementación: fechas en las cuales se realizaron las actividades que indica su nombre.

Nuevas funcionalidades: listar nuevas funcionalidades que presenta la versión.

Mejoras: listar mejoras con respecto a la versión anterior.

Público: identificar el público tomando en cuenta las partes interesadas y los usuarios finales.

Medios de comunicación y actividades: definir canales de comunicación y acciones previstas.

Forma de adiestramiento: indicar si se requiere de un curso presencial o curso en línea.

Nro de facilitadores: número de personas calificadas que se necesitan para impartir el curso.

Objetivos: listar objetivos del curso a impartir.

Material necesario: incluir link con el material del curso tales como: guías, manuales, diapositivas.

Vista Plan de Distribución e Instalación

Plan de Distribucion e Instalacion

Codigo RFC: COD2 Proyecto: Actualización de SW 7.3.1 Version: 8.0

Fecha de reunion: 9/27/2011
Fecha de emision: 9/28/2011

Forma de distribucion: Paquete de Versiones

Requerimientos para la instalacion: NA

Procedimientos de distribucion e instalacion: NA

Seguridad: autentificarse para saber que usuario realizo la actual

Campos:

Fecha de reunión: fecha en que se reunieron las partes involucradas para fijar los puntos del plan.

Fecha de emisión: fecha en que se publica el plan.

Forma de distribución: seleccionar la forma en que se va a realizar la distribución de la versión: versión delta, versión completa, paquete de versiones.

Requerimientos para la instalación: listar requerimientos que debe tener la máquina para soportar la instalación.

Procedimientos de distribución e instalación: indicar cómo debe realizarse la distribución e instalación de la versión.

Seguridad: seguridad para tener acceso a la versión.

7. Administrador Gestión de Versiones

Se encarga de:

- Aprobar cotizaciones en caso de que la versión tenga que ser comprada a proveedores externos
- Aprobar los resultados de las pruebas unitarias, pruebas integrales y de aceptación de usuario
- Apoyar en las actividades de lanzamiento

Vista Solicitar aprobación de cotización

Solicitar Aprobacion de Cotizacion

Cotizacion Nro: 1

Fecha de emision:
2011-08-29

Empresa: empre

Telefono: 1

Direccion: dir

Cantidad	Producto solicitado	Precio Unid	Total
1	prod	1	1

Aprobar Cotizacion: SELECT

Fecha:

Campos:

Aprobar cotización: indicar si la cotización va a ser aprobada: si o no.

Fecha: fecha de la solicitud de aprobación.

Vista Solicitar aprobación de resultados de pruebas unitarias

Solicitar Aprobacion de Resultados de Pruebas Unitarias

Proyecto: Actualización de SW 7.3.1

Version: 8.0

Resultados: las pruebas realizadas a la version fueron satisfactorias

Errores: 0

Aprobar Resultados de Pruebas Unitarias: Si

Fecha: 9/15/2011

Campos:

Aprobar resultados de pruebas unitarias: indicar si los resultados van a ser aprobados.

Fecha: fecha de la solicitud de aprobación.

Vista Solicitar aprobación de resultados de pruebas integrales y de aceptación de usuario

Solicitar Aprobacion de Resultados de Pruebas

Proyecto: Actualización de SW 7.3.1

Version: 8.0

Resultados de Pruebas Integrales

Resultados: resultados satisfactorios

Errores: 0

Resultados de Pruebas de Aceptacion de Usuario

Resultados: resultados satisfactorios

Errores: 0

Aprobar Resultados de Pruebas: Si

Fecha: 9/22/2011

Campos:

Aprobar resultados de pruebas: indicar si los resultados van a ser aprobados.

Fecha: fecha de la solicitud de aprobación.

8. Proveedores Externos

Lo conforma el grupo de empresas a los cuales se les compra alguna versión; se encargan de emitir cotizaciones con la versión que cumpla con las requerimientos del RFC



Campos:

Nro: número de cotización.

Fecha de emisión: fecha en que se realizó la cotización.

Empresa, teléfono y dirección: datos de la empresa que emite la cotización.

Para, teléfono y dirección: datos de la empresa que solicita la cotización.

Cantidad: indicar la cantidad de productos.

Producto solicitado: indicar el o los productos solicitados.

Precio unitario: precio de cada producto.

Total: total a pagar por los productos a adquirir.

8. Grupo de Usuarios Finales

Lo conforman un grupo de personas que son usuarios finales, los cuales fueron seleccionados para probar la nueva versión.

Vista Realizar pruebas de aceptación de usuario

Realizar Pruebas de Aceptacion de Usuario		Fecha de ejecucion:
		9/21/2011
Proyecto: Actualización de SW 7.3.1	Version: 8.0	
Entorno: <input type="text" value="TI"/>	Casos de prueba: <input type="text" value="colocar link con casos de prueba"/>	
Resultados: <input type="text" value="resultados satisfactorios"/>	Errores: <input type="text" value="0"/>	Observaciones: <input type="text" value="mejorar la interfaz"/>

Campos:

Resultados: contiene los resultados de las pruebas realizadas.

Errores: indicar el número de errores encontrados durante las pruebas y una breve explicación de las mismas.

Observaciones: comentarios adicionales.

Anexo 12 Encuesta ITIL – Gestión de Versiones y Liberaciones

Fecha: ___ - ___ - ___

Empresa: _____

Encuesta

ITIL – Gestión de Versiones y Liberaciones

1. ¿Conoce el estándar ITIL (Information Technologies Infrastructure Library)?

Si No

2. ¿Tiene conocimiento sobre el Proceso de Gestión de Versiones y Liberaciones propuesto por ITIL?

Si No

3. ¿Considera importante contar con un Sistema de Información que gestione las Versiones de software en la empresa?

Si No

Justifique su respuesta:

4. Indique aspectos que deben ser tomados en cuenta para la construcción del Sistema de Información para gestionar Versiones

Gracias por brindarnos su opinión

Anexo 13 Encuesta Sistema de Información para la Gestión de Versiones y Liberaciones de software

Fecha: ____ - ____ - ____

Empresa: _____

Encuesta

Sistema de Información para la Gestión de Versiones y Liberaciones

1. El uso del sistema le pareció:

Fácil

Normal

Difícil

2. La navegación del sistema le resultó:

Muy intuitiva

Intuitiva

Poco intuitiva

3. La distribución de los elementos le pareció:

Adecuada

Poco adecuada

4. El tiempo de respuesta del sistema le pareció:

Rápido

Normal

Lento

Gracias por brindarnos su opinión

Anexo 14 Glosario

A

Aceptación Acuerdo formal que indica que un Servicio de TI, Proceso, Plan, u otro Entregable se han completado, es preciso, Confiable y cumple con los Requisitos especificados. Normalmente la Aceptación es precedida por una Evaluación o Prueba y es requerida antes de proceder con la siguiente fase de un Proyecto o Proceso.

B

Back-out Marcha atrás. Sinónimo de refuerzo

BPM – Business Process Management Gestión de Procesos de Negocio. Es un conjunto de disciplinas empresariales, basadas en enfoques metodológicos, aplicadas con el fin de mejorar la eficiencia a través de la gestión holística de los procesos, reglas y servicios del negocio, que se deben modelar, automatizar, integrar, monitorizar y mejorar de forma continua.

BPMN – Business Process Modeling Notation Estándar mundialmente reconocido, aceptado y aplicado en el mercado, para la diagramación y especificación de procesos de negocio, desde el modelado de procesos conceptuales y lógicos, hasta diseños de procesos orientados a tecnología Workflow. BPMN fue desarrollado por la Business Process Management Initiative (BPMI), y actualmente el Object Management Group (OMG) lo mantiene desde 2005, año en que se produjo la fusión entre las dos organizaciones. La versión actual de BPMN es la 1.2, pero desde enero del 2009 se está progresando en la revisión de la versión BPMN 2.0.

BPMS – Business Process Management System Conjunto de software especializado para el modelado, integración, orquestación, y monitorización de procesos, reglas de negocio, servicios, sistemas y recursos empresariales.

C

CI – Configuration Item Elemento de Configuración. Cualquier componente que necesita ser gestionado con el fin de ofrecer un servicio de TI. La información sobre cada entidad se registra en el Sistema de Gestión de Configuración.

CMDB – Configuration Management Data Base Base de Datos de Gestión de la Configuración, es usada para almacenar Registros de Configuración durante todo su Ciclo de Vida. El Sistema de Gestión de la Configuración mantiene una o más CMDBs, y cada CMDB contiene Atributos de CIs, y Relaciones con otros CIs.

D

Desarrollo	Proceso responsable de crear o modificar un Servicio TI o Aplicación. También usado para referirse al Rol o grupo a cargo del trabajo de Desarrollo.
Despliegue	La Actividad responsable del movimiento de hardware, software, documentación, procesos, etc. nuevos o cambiados, en un Ambiente en Producción. Despliegue es parte del Proceso de Gestión de la Entrega y Desarrollo.
DS – Definitive Spares	Almacenamiento de Hardware Definitivo. Lugar reservado para el almacenamiento seguro de repuestos de hardware definitivo que deben ser iguales a los elementos de configuración de hardware en el entorno real. Sólo HW autorizado debe ser aceptado en el DS, estrictamente controlada por Gestión de Cambio y Gestión de Versiones.
Diseño	Actividad o Proceso que identifica Requerimientos y entonces define una solución que es capaz de alcanzar dichos Requerimientos.
DML – Definitive Media Library	Biblioteca de Software Definitivo. Lugar donde se almacenan con seguridad las versiones definitivas aprobadas de Elementos de Configuración de Software. La DML también puede contener CIs asociado tales como licencias y documentación. El DML es un área de almacenamiento lógico única cuando haya múltiples localizaciones. Todo el software en la DML está bajo el control de Cambios y Gestión de la Entrega y es registrada en el Sistema de Gestión de Configuración. Solamente el software que está en la DML es aceptable para utilizar en una nueva Entrega.

G

Gestión de Cambio	Proceso responsable del control del Ciclo de Vida de los Cambios. El objetivo primario de Gestión del Cambio es permitir la ejecución de los Cambios a realizar, con la mínima afectación a los Servicios de TI.
Gestión de Configuración y Activos del Servicio	Proceso responsable de conservar la información acerca de CIs requeridos en la prestación de un servicio de TI, incluyendo las relaciones entre los mismos.

I

ITIL – Information Technology Infrastructure Library	Biblioteca de Infraestructura de Tecnologías de la Información. Documenta las Mejores Prácticas para la Administración de Servicios de TI. ITIL está basada en la administración de servicios desde el punto de vista del negocio,
--	--

M

Modelado Técnica que se emplea para predecir el comportamiento futuro de un Sistema, Proceso, Servicio de TI, Elemento de Configuración etc. El Modelado suele emplearse en Gestión Financiera, Gestión de Capacidad y Gestión de la Disponibilidad.

P

Política Documento formal que contiene las intenciones y expectativas de gestión. Las Políticas se utilizan para dirigir las decisiones, y asegurar un desarrollo e implementación coherente y apropiado de los Procesos, Estándares, Roles, Actividades, Infraestructura de TI y otros

Proceso Conjunto estructurado de Actividades diseñado para la consecución de un Objetivo determinado. Los Procesos requieren de una o más entradas y producen una serie de salidas, ambas previamente definidas. Un Proceso suele incorporar la definición de los Roles que intervienen, las responsabilidades, herramientas y Controles de gestión necesarios para obtener las salidas de forma eficaz. El Proceso podrá definir las Políticas, Estándares, Guías de Actuación, Actividades, y las Instrucciones de Trabajo que fueran necesarias.

R

RFC – Request for Change Petición de cambio. Propuesta formal para que se realice un Cambio. Una RFC incluye detalles del Cambio propuesto, y puede registrarse en papel o electrónicamente. El término RFC se suele confundir con Registro de Cambio, o con el Cambio en sí.

S

Servicio Web – Web Service Es una colección de protocolos y estándares informáticos que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes y ejecutadas sobre cualquier plataforma pueden utilizar los servicios web para intercambiar datos en redes de ordenadores, tales como Internet o Intranet.

Simulación de Proceso Son un conjunto de técnicas y algoritmos que permiten, a través de software especializado, realizar predicciones del funcionamiento y comportamiento de los procesos de negocio, y los recursos necesarios para ejecutar estos. Las simulaciones más comunes que se realizan sobre los procesos modelados, son los de tiempos de ejecución y utilización de recursos, para simular las cargas de trabajo y tiempos de respuesta de los procesos, intentando descubrir posibles cuellos de botellas u otros posibles problemas de cada proceso. Con las simulaciones de procesos podemos crear escenarios de diseño, rediseño e implantación de nuevos procesos o procesos mejorados.

SOA – Services Oriented Architecture Arquitectura Orientada a Servicios, es un concepto de arquitectura de software en cuyo diseño y funcionamiento utiliza servicios (web services), para dar soporte a los requerimientos de software de los usuarios.

SOA, como parte del BPM, permite modificar o crear nuevos procesos de forma dinámica, reduciendo enormemente los costes y el tiempo, e imprimiéndole agilidad y flexibilidad a la organización.

Subproceso Determinados procesos, por su complejidad, necesitan de subprocesos para su ejecución.

Por tanto un subproceso es “un proceso” subordinado y necesario para ejecutar el proceso principal del que depende.

Por ejemplo: en el proceso desarrollar un nuevo producto, hay que ejecutar el subproceso de pruebas del producto, para verificar el perfecto funcionamiento en otros procesos.

U

UML – Unified Modeling Language Es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

V

Versión Colección de hardware, software, documentación, Procesos, u otros Componentes requeridos para implementar uno o más Cambios aprobados a los Servicios de TI. Los contenidos de cada Versión son Administrados, Probados, e Implementados como una única entidad.

W

WSDL - Web Service Description Language WSDL es un formato XML para describir servicios de red como un conjunto de puntos operativos en mensajes que contienen información orientada a documento u orientada a procedimiento.

X

XML – eXtensive Markup Language Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Permite definir la gramática de lenguajes específicos (de la misma manera que HTML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.