

Universidad de las Ciencias Informáticas

Facultad 4



**Título: Adaptación de la metodología RUP a los nuevos
módulos del Proyecto Registros y Notarías.**

Trabajo de Diploma para optar por el título de
Ingeniero de Ciencias Informáticas

Autor(es): Elizabeth Martínez Iglesias.

Tutor(es): Ing. Karina Pérez Teruel.

Junio 2007

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Proyecto Registros y Notarías de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Elizabeth Martínez Iglesias

Ing. Karina Pérez Teruel

Nombre completo del autor

Nombre completo del tutor

DATOS DEL CONTACTO

Tutora: Ing. Karina Pérez Teruel

Correo: karinapt@uci.cu

Edificio 11-109, manzana 26 uci

Teléfono: 8372737

Autora: Elizabeth Martínez Iglesias

Correo: emartinezi@estudiantes.uci.cu

Teléfono: 8372167

DEDICATORIA

A dos hombres que han significado mucho en mi vida, a quienes amo con toda la fuerza de mi corazón, mi futuro esposo (bicho Ovany) y mi padre. Al primero por ayudarme tanto en todo momento, por estar siempre presente cuando lo necesito, por ser tan fiel compañero y amigo, por amarnos tanto. Al segundo por ser mi guía diaria, por ser la luz que me empuja a seguir adelante, por su apoyo incondicional.

A mi madre, por ser tan comprensiva y desear tanto mi bienestar.

A mis abuelas, por tantos consejos.

A mis hermanas, por ser dos pequeños tesoros de mi corazón.

A mis amigas de la universidad, por apoyarme tanto en cualquier situación.

A todos aquellos que me desean el bien y que de una forma u otra me ayudaron a ser mejor cada día.

A todos, gracias, aquí está el fruto.

Elizabeth Martínez Iglesias

RESUMEN

El éxito del proceso de desarrollo de software depende directamente de la metodología implementada para la ejecución del mismo. Las características del equipo de desarrollo así como las particularidades del proyecto a desarrollar, determinan que metodología se utilizará con el objetivo de maximizar los resultados finales. La adopción del Rational Unified Process (RUP) como metodología para regir el proceso de desarrollo de software en el proyecto Registros y Notarías (RyN) y la aparición de nuevos módulos con características particulares diferentes a los hasta el momento desarrollados trajo consigo la necesidad de un ajuste del RUP. En el presente trabajo se expone un análisis de la metodología RUP general, así como, de sus variaciones más importantes conocidas y utilizadas a nivel mundial. Además, se determinan los motivos que conllevan al desarrollo de una nueva configuración del RUP planteando las disciplinas, los roles, las actividades y los artefactos para la misma. Finalmente se analizan los resultados logrados mediante la opinión del Jefe del Proyecto, y se obtiene un artefacto de vital importancia que influye en el desarrollo organizado del equipo.

PALABRAS CLAVE

Metodología

RUP

Disciplinas

Roles

Actividades

Artefactos

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	4
1.1. Introducción.....	4
1.2. ¿Qué es una metodología de desarrollo de software?.....	4
1.3. Historia del RUP.....	6
1.4. ¿Qué es el RUP?.....	8
1.5. Utilización de la metodología RUP a nivel mundial.....	8
1.6. Características del RUP.....	9
1.7. Variaciones del RUP.....	12
1.7.1. AUP (Agile Unified Process).....	12
1.7.2. BUP (Basic Unified Process).....	13
1.7.3. EUP (Enterprise Unified Process).....	14
1.7.4. OpenUP (Open Unified Process).....	15
1.8. Variaciones del RUP en la Universidad.....	16
1.9. RUP como un proceso configurable.....	16
1.10. Selección de la variante.....	18
CAPÍTULO 2: SOLUCIÓN PROPUESTA.....	19
2.1. Introducción.....	19
2.2. Fases e Hitos.....	19
2.3. Disciplinas.....	19
2.4. Roles y Actividades.....	20
2.5. Artefactos.....	33
CAPÍTULO 3: ANÁLISIS DE RESULTADOS.....	41
3.1. Introducción.....	41
3.2. Proceso de desarrollo.....	41
3.3. Módulo Bienes.....	41
3.4. Artefacto: Caso de desarrollo, del módulo Bienes.....	42
3.5. Opinión del Jefe de Proyecto.....	42
CONCLUSIONES.....	43

RECOMENDACIONES.....	44
BIBLIOGRAFÍA.....	45
REFERENCIAS BIBLIOGRÁFICAS	46
ANEXO 1. CONFIGURACIÓN PROPUESTA DE ROLES, ACTIVIDADES Y ARTEFACTOS DE ACUERDO A LAS DISCIPLINAS DEL RUP.....	47
ANEXO 2. CASO DE DESARROLLO DEL MÓDULO BIENES.....	54

INTRODUCCIÓN

Actualmente la industria del software¹ es la más rentable a nivel mundial. El país ha implementado políticas para el desarrollo acelerado de este sector y orientadas a lograr su inserción en el mercado mundial, ejemplo de esto son los proyectos de exportación de marcada importancia para su desarrollo, tanto desde el punto de vista económico como desde el estratégico, demostrando así las capacidades y potencialidades de la Industria Cubana del Software (ICSW).

La ICSW, en pleno desarrollo, acoge a la Universidad de las Ciencias Informáticas (UCI) en uno de los roles centrales, jugando un importante papel de caras al cumplimiento de las metas planteadas, con tan solo 4 años de experiencias en este sector se han obtenidos logros capaces de plasmar los adelantos logrados de una manera rápida y comprometida.

En la UCI existen varios proyectos que se dedican a la producción del software, tanto para informatizar la sociedad cubana como para ayudar a la construcción de sistemas informáticos en países hermanos. RyN es un proyecto que desarrolla un software encargado de automatizar los Registros Mercantiles e Inmobiliarios en la República Bolivariana de Venezuela para gestionar de forma eficiente todas las actividades relativas a notarías y registros: atención al cliente, gestión documental, gestión administrativa y contable, comunicaciones con las diferentes instituciones, organismos públicos y entidades a través de servicios (Alexei, 2005).

RyN adoptó desde su surgimiento la metodología RUP como base para regir el proceso de producción de software. Esta metodología esta orientada mas bien a grandes proyectos, pero tiene como característica ser configurable, esto le permite ser adaptada según las necesidades de cada organización o proyecto de acuerdo al contexto y las necesidades en cada caso.

Los nuevos módulos del proyecto RyN presentan características propias como: tamaño del proyecto, número de miembros del equipo de desarrollo, etc., que dificultan el completo desarrollo de las actividades propuestas para cada rol y por consecuencia no se obtienen todos los artefactos correspondientes, marcando así diferencias notables al no ajustarse completamente al RUP clásico, llegando a ser esta la situación problemática del presente trabajo.

¹ El software de computadora es el producto que diseñan y construyen los ingenieros del software. Esto abarca programas que se ejecutan dentro de una computadora de cualquier tamaño y arquitectura, documentos que comprenden formularios virtuales e impresos y datos que combinan números de texto y también incluyen representaciones de información de audio, video e imágenes. (Pressman, 2001)

De acuerdo a lo antes planteado surge el siguiente problema: ¿Cómo adaptar la metodología RUP al proceso de desarrollo del software en los nuevos módulos del proyecto RyN de acuerdo a sus características?

Para dar respuesta a este problema se asume como objetivo general: configurar el proceso de desarrollo del software que se adapte a las necesidades específicas de los nuevos módulos del proyecto RyN, derivándose a su vez los siguientes objetivos específicos:

1. Adaptar las disciplinas del RUP.
2. Definir roles a desempeñar junto a las actividades a realizar por los miembros del equipo de desarrollo de acuerdo a las disciplinas del RUP.
3. Definir los artefactos de salida de acuerdo a las disciplinas de la metodología RUP.

Para dar cumplimiento a los objetivos antes expuestos se hace necesario desarrollar las siguientes tareas:

- Estudiar las características de la metodología RUP.
- Estudiar las diferentes variaciones del RUP más aplicadas actualmente en el mundo.
- Estudiar la utilización de la metodología RUP a nivel mundial y en la universidad.
- Estudiar características y actividades desempeñadas por cada rol dentro de la metodología RUP.
- Estudiar los artefactos de salida de cada disciplina de la metodología RUP.
- Definir roles, actividades y artefactos a obtener.

El objeto de estudio sería entonces: metodología de desarrollo de software RUP y el campo de acción: adaptación de la metodología RUP al proceso de desarrollo de software en los nuevos módulos del proyecto RYN.

El presente trabajo plantea además la hipótesis de que si se implementa una configuración del proceso de desarrollo del software en los nuevos módulos del proyecto RyN entonces se logrará un proceso de desarrollo de software más organizado, mejor definido y eficiente, adaptado a sus necesidades específicas.

El trabajo se estructura en varios capítulos, los cuales se detallarán a continuación:

1. Capítulo I: Fundamentación Teórica. En este capítulo se abordan las características de la metodología RUP, las variaciones más conocidas de la misma y su utilización internacionalmente.
2. Capítulo II: Solución Propuesta. En este capítulo se aborda la configuración propuesta a seguir de acuerdo a los roles, actividades y artefactos.

3. Capítulo III: Análisis de Resultados. En este capítulo se evaluará la configuración propuesta en el módulo Bienes del proyecto RyN.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción.

En la actualidad el software, constituye un elemento de gran importancia a escala mundial formando parte de la columna vertebral de la mayoría de los procesos que se llevan a cabo diariamente en los sectores económicos, sociales y políticos de cada país.

Desde sus comienzos a mediados de los años 60, el software ha experimentado una rápida evolución. En un principio se escribían de manera personal en lenguaje de bajo nivel y orientado completamente al hardware para el que eran diseñados. A inicio de los años 80 el software se comienza a independizar del hardware en uno de los mercados más exitosos hasta la actualidad.

El avance y la evolución de las técnicas y las metodologías utilizadas en el proceso de desarrollo ha permitido lograr un perfeccionamiento gradual de los resultados obtenidos, no obstante los niveles de productos software entregados y que están siendo utilizados es aún inferior a los niveles de los proyectos que no logran terminar y entregar el producto para el cual estaban destinados inicialmente.

En su proceso de desarrollo se involucran diferentes fallas como: no responder a los intereses solicitados por el cliente; procesos de diseño, implementación y pruebas defectuosos, tiempo de duración en desacuerdo a lo estimado, cambios en el presupuesto establecido, documentación del producto innecesaria, etc., esto si no se cuenta con un método estricto que lo controle.

El logro de productos software con probada calidad en el ámbito mundial, desarrollado bajo criterios de rentabilidad, planificación, diseño y organización es una labor que merita muchos esfuerzos y una línea de producción muy bien definida por la que debe regirse el proceso en su totalidad.

Para lograr la productividad del software se necesita regir el proceso de desarrollo sobre metodologías que permitan obtener resultados de acuerdo a lo establecido con los clientes, por lo que a la hora de desarrollar un producto software uno de los aspectos más importantes a tener en cuenta es la metodología a adoptar.

1.2. ¿Qué es una metodología de desarrollo de software?

Las metodologías de desarrollo de software abarcan todo el ciclo de vida del software, y se definen como “un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software” (Patón, 2006), adoptando la misma se obtiene un producto software más predecible y permite ciertas características deseables como (Patón, 2006):

- Existencias de reglas bien definidas.
- Verificaciones intermedias.
- Planificación y control.
- Comunicación efectiva.
- Utilización sobre un abanico amplio de proyectos.
- Fácil formación.
- Herramientas CASE (Computer Aided Software Engineering).
- Actividades que mejoren el proceso de desarrollo.
- Soporte al mantenimiento.
- Soporte de la reutilización de software.

De acuerdo a su definición de metodología, los procedimientos detallan consejos para elaborar una actividad; las técnicas serían la forma de ejecutar un procedimiento para obtener un resultado determinado; las herramientas software son las que hacen posible automatizar el proceso de desarrollo del software y la documentación es la que identifica el software que se esta desarrollando.

En cualquier empresa de desarrollo de software, un rasgo fundamental es el conocimiento a plenitud que tenga su personal de desarrollo, pero esto no solo sirve para alcanzar el objetivo deseado, la comunicación en equipo que nos permite el empleo de una metodología de desarrollo de software se hace muy necesaria, pues brinda un ambiente disciplinado y organizado a la hora de obtener un producto eficaz y de calidad, también provee otros beneficios como (Methodologies in IT: Comprehension, Selection, and Implementation, 2005):

- La administración puede proteger su inversión asegurándose de que los soportes del proyecto están bien definidos y prometen tener éxitos.
- La administración y los usuarios conocen por adelantado lo que pueden esperar de cada proyecto.
- Una buena metodología asegura que el estándar de calidad sea conocido, estas expectativas exceden la realidad de los métodos de desarrollo.
- Las revisiones o las auditorias de calidad proporcionan a la administración una evaluación independiente de la calidad del trabajo hecho.
- Sorpresas como costos excedidos, cambios de alcance, tarde implementación y otros riesgos pueden ser reducidos al mínimo.
- El estado de todos los proyectos es fácilmente accesible.

- Las organizaciones obtiene incrementos significativos de la productividad en el desarrollo de la aplicación.

Como se dijo en la introducción, para el desarrollo del proyecto RyN se adoptó, desde un principio, la metodología RUP, y se mantendrá la misma para el desarrollo de los módulos nuevos.

1.3. Historia del RUP.

El Proceso Unificado surge en el año 1967 con la Metodología Ericsson elaborado por Ivar Jacobson, donde se hace un acercamiento al modelado mediante componentes, y se introduce el término de casos de uso. Ya en el año 1976 se publica el SDL (Lenguaje de Especificación y Descripción) por parte del CCITT (organismo internacional para la estandarización en el área de las telecomunicaciones), que introduce el término procesos y este estándar proponía diagramas que eran una proximidad a los utilizados actualmente por UML. En 1987 Ivar Jacobson fundó la compañía Objectory AB y hasta 1995 desarrolló un proceso de desarrollo denominado Object Factory , donde la arquitectura del software comienza a destacarse y se utiliza el término modelo para representar cada flujo de trabajo y la trazabilidad para hacer un recorrido bidireccional en estos modelos. En 1995 Rational Software Corporation adquiere la compañía Objectory AB, y se unifican los desarrollos de esas dos compañías, por su parte la Rational Software Corporation destacó su desarrollo en la arquitectura y en el desarrollo iterativo, jugando un papel muy importante Philippe Kruchten, también se diseñó el método de las 4 fases posibilitando controlar mejor las iteraciones, aquí hicieron su aporte Walter Royce, Richt Reitman, Grady Booch y Philippe Kruchten. Ya en los años 1995 – 1997 mediante esta fusión se desarrolla el ROP (Proceso Objectory de Rational) y como lenguaje se implantó el UML (Lenguaje Unificado del Modelado)², cuyos autores fueron Grady Booch y James Rumbaugh. En años posteriores la Rational Software Corporation se unió a otras empresas con disímiles experiencias que aumentaron el ROP hasta que en 1998 se publica una nueva versión conocida como RUP 5.0 (Rational Unified Process), se le cambio el nombre porque era una unificación de varias empresas. Ya en el año 1999 se publica por Booch, Rumbaugh y Jacobson el libro El Proceso Unificado de Desarrollo de Software, el mismo iba objetivamente dirigido a equipos de desarrollo cuyas actividades estuviesen en las disciplinas: requerimientos, análisis, diseño, implementación y prueba. (Ivar, et al., 2000)

² Lenguaje utilizado para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. (James, et al., 2000)

Ya en Junio pero de 1999 se desarrolla el RUP 5.5 basado en Web, que enfoca el desarrollo en tiempo real para una mejor ayuda a los proyectos. En Octubre de 1999, un cambio al ciclo de vida del RUP fue propuesto por Scott W. Ambler mediante una extensión del RUP con una nueva fase, la de Producción y una disciplina llamada Administración de la Infraestructura. En el 2000 se obtiene otra versión del RUP, esta incluyó una adición de técnicas de ingenierías del negocio a la disciplina Modelado del Negocio y un mayor acercamiento a los requerimientos. En el transcurso de los años 2000 – 2002 son publicados una serie de libros acerca del Proceso Unificado editados por Scott W. Ambler y Larry Constantine, describiendo temas no cubiertos aún por el RUP como prueba y agilidad mejorada, estas series mejoraron el ciclo de vida del RUP, el EUP (Enterprise Unified Process) es un ejemplo. En marzo del 2002 es publicado el libro Modelado Ágil, mostrando como es posible llevar un acercamiento ágil a un proyecto modelado por RUP, y ya en diciembre la compañía IBM (International Business Machines) compra la Rational Software Corporation. En el 2003 sale otra versión del RUP, cubriendo algunos temas de la serie de libros acerca del Proceso Unificado. En la primavera del 2003 el sitio de EUP se pone a disposición de los usuarios; ya en el 2004 sale una versión del EUP, su mejoramiento incluye la adición de la fase Retiro y la expansión de la disciplina Administración de la Empresa (anteriormente Administración de la Infraestructura) a 7 disciplinas detalladas. En febrero del 2005 es publicado el libro The Enterprise Unified Process: Extending the Rational Unified Process por Scott Ambler, John Nalbone y Michael Vizdos presentando una introducción definitiva del EUP y demostrando como se puede utilizar la misma en el ámbito que se necesite (Extending the Rational Unified Process with the Enterprise Unified Process, 2006). En septiembre es anunciado como una de las mejoras practicas de desarrollo de software en Boston el AUP (Agile Unified Process). En Octubre fue anunciado por IBM el proyecto EPF (Eclipse Project Framework)³ y el BUP (Basic Unified Process), este ultimo una versión del RUP optimizado para pequeños proyectos. Ya en noviembre es anunciado también el RMC (Rational Method Composer)⁴. En marzo del 2006 se sustituyó el nombre de BUP por OpenUP (Open Unified Process)⁵ y ya en septiembre sale la versión 1.0 del EPF y la 0.9 del OpenUp.

³ Plataforma de proceso de código abierto para ayudar a empresas a establecer la consistencia en la planificación y la ejecución de proyectos de desarrollo de software en conjunto con la fundación Eclipse. (The Eclipse Process Framework project, 2005)

⁴ Producto comercial construido sobre Eclipse y actualmente la principal evolución del RUP. (Introducing IBM Rational Method Composer, 2005)

⁵ Proceso de desarrollo de software a código abierto resultado del proyecto EPF. (Armstrong Process Group, 2005)

1.4. ¿Qué es el RUP?

RUP como metodología es un “proceso de desarrollo de software que define *quién* esta haciendo *qué*, *cuándo* y *cómo* alcanzar un determinado objetivo”, además “es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto” (Ivar, et al., 2000).

El RUP es además un proceso de ingeniería de software⁶ y una plataforma flexible de proceso de desarrollo de software que ayuda a brindar un proceso de guía personalizado y más consistente a su equipo de proyecto.

1.5. Utilización de la metodología RUP a nivel mundial.

Disímiles proyectos han sido desarrollados con éxito utilizando la metodología RUP, a continuación se brinda una breve descripción de algunas empresas que la han utilizado satisfactoriamente:

Dybox es una empresa que cuenta con más de siete años de experiencia en la construcción de diferentes proyectos utilizando con éxito esta metodología de desarrollo. Hasta el momento esta metodología les ha permitido una aproximación al problema que no sólo garantiza que los proyectos abordados serán ejecutados íntegramente sino que además evita desviaciones importantes respecto de los plazos y, tan importante como esto, permite una definición acertada del sistema en un inicio para hacer innecesarias las reconstrucciones parciales posteriores. Entre los diferentes productos que comercializan se encuentran Aplicaciones y Sitios Web de tipo e-commerce, e-business, e-marketing, Intranet, Extranet y Aplicaciones Operacionales.

Otra empresa que logra adaptar la metodología RUP de manera exitosa es la joven KadaSoftware. Dicha metodología les permite lograr niveles altos de exactitud en la planificación del tiempo de ejecución y de los recursos utilizados en los procesos de desarrollo, minimizando las fallas y logrando un cumplimiento estricto de los plazos de entrega propuestos. Sistemas de gestión y sitios Web son los principales campos que abarcan los productos de esta empresa.

⁶ Es una tecnología multicapa en la que se pueden identificar: los métodos (indican cómo construir técnicamente el software), el proceso (es el fundamento de la Ingeniería de Software, es la unión que mantiene juntas las capas de la tecnología) y las herramientas (soporte automático o semiautomático para el proceso y los métodos). (Pressman, 2001)

DATASUL* es una multinacional brasilera pionera en el desarrollo de software con 28 años de experiencia. Esta se destaca en el área como una de las mayores empresas dedicadas al desarrollo de productos software de soluciones para la gestión empresarial. Esta empresa ha atravesado diferentes etapas en el desarrollo de productos software a través de los distintos períodos por los cuales ha evolucionado la práctica de esta disciplina. Esta rica experiencia unida a las relaciones corporativas le permite acceder y utilizar la mejor base tecnológica para diseñar y construir sus productos, así como lograr las mejores prácticas en modelado, desarrollo y mantenimiento de software. En la actualidad utilizan el RUP como metodología y producto logrando ofertar soluciones software completas, integradas, manteniendo los plazos de entrega propuestos inicialmente y reduciendo nivel de fallas a niveles ínfimos. Como resultado, dicha empresa, ha obtenido una excelente aceptación por parte de los clientes en un mercado cada vez más creciente y competitivo dentro del área. Los productos software desarrollados por esta multinacional están orientados a la gestión automatizada de procesos críticos de sus clientes como son las finanzas, los recursos humanos, la logística y la manufactura entre otros.

La empresa Cingo*, también brasileña, se dedica al desarrollo de diferentes productos software entre los cuales figuran soluciones Web a la medida. Esta empresa logra mayores velocidades en el desarrollo gracias a la reutilización de componentes implementados con gran calidad. Otro factor importante que les permite lo anterior es la práctica de un gran control en los procesos implementados, utilizando un modelo de fábrica de software basado sobre un flujo de trabajo diseñado sobre la base de la metodología RUP. Esta empresa obtiene excelentes y competitivos costos de producción que lo convierten un competidor de excelencia.

* Estas dos empresas mantienen una estrecha relación corporativa.

1.6. Características del RUP.

Para la realización del producto software la metodología RUP divide el trabajo en 9 flujos de trabajo y 4 fases y tiene como características fundamentales ser iterativo e incremental, dirigido por casos de usos y centrado en la arquitectura.

El ser iterativo e incremental le permite al equipo de desarrollo en un tiempo estimado obtener una pequeña parte del producto de acuerdo a una fase que pase por todas las disciplinas, esto se conoce por iteración, permite además según cada iteración un crecimiento del producto, conocido como incremento.

Sobre esto recae gran importancia pues se puede ir observando poco a poco el adelanto que va teniendo el producto que se esta desarrollando y que se puedan corregir a tiempo las deficiencias.

La característica de ser dirigido por casos de uso ayuda también al equipo de trabajo a desarrollar el sistema según los requisitos solicitados por el cliente pues a través de los mismos estos se modelan.

Por último, no por ello menos importante, posee la característica de estar centrado en la arquitectura, esta indica como puede construirse el sistema y en que orden. Todas estas características fueron destacadas por los autores del proceso unificado desde sus inicios.

Las disciplinas del RUP según el orden que se muestran en la figura 1.1 serían las siguientes:

- Modelado del Negocio: cuyo objetivo es obtener una breve descripción de la panorámica actual del negocio en el que se va a trabajar e identificar lo que se necesita automatizar.
- Requerimientos: cuyo objetivo es identificar las exigencias que el producto software debe cumplir de forma detallada donde se cumpla lo solicitado por el cliente y además se obtienen los prototipos que tendrá cada funcionalidad en el sistema.
- Análisis y Diseño: cuyo objetivo es indicar lo que hay que programar según lo especificado en los requerimientos, además se desarrolla la arquitectura del software.
- Implementación: cuyo objetivo es desarrollar el producto software mediante una organización del código en capas, además de integrar lo desarrollado por todo el equipo en un fichero ejecutable.
- Prueba: cuyo objetivo es encontrar las deficiencias tanto en el código del software como en la documentación, esto tiene que corresponder. Además, debe validar que se cumplan los requisitos planteados por el cliente y como resultado de esta disciplina se obtiene el producto final.
- Despliegue: cuyo objetivo es entregar el producto software para que se proceda a su instalación.
- Administración y Configuración de Cambios: cuyo objetivo es que se pueda tener control y administración de todas las actividades, artefactos y código que se este desarrollando por cada integrante del equipo.
- Administración del Proyecto: cuyo objetivo es administrar el personal del equipo, su capacitación, planificar el desarrollo del proyecto, controlar los posibles riesgos que puedan aparecer en el desarrollo del software.
- Ambiente: cuyo objetivo es configurar el proyecto de acuerdo a sus especificaciones, tanto en herramientas como en actividades.

Las Fases del RUP según el orden que se muestran en la figura 1.1 serían las siguientes:

- Inicio: esta fase tiene como objetivo obtener toda la información del negocio del proyecto con sus exigencias más importantes y de ahí preparar y planificar los casos de uso, establecer el costo del proyecto, esquematizar la arquitectura posible y preparar el ambiente del proyecto.
- Elaboración: esta fase tiene como objetivo definir la arquitectura del proyecto, desarrollar los casos de usos según las prioridades desarrollando casi totalmente el sistema.
- Construcción: esta fase tiene como objetivo optimizar el proceso para obtener el software que cumpla lo requerido por el cliente.
- Transición: esta fase tiene como objetivo entregar el material de soporte del producto y el software al cliente para su posterior utilización.

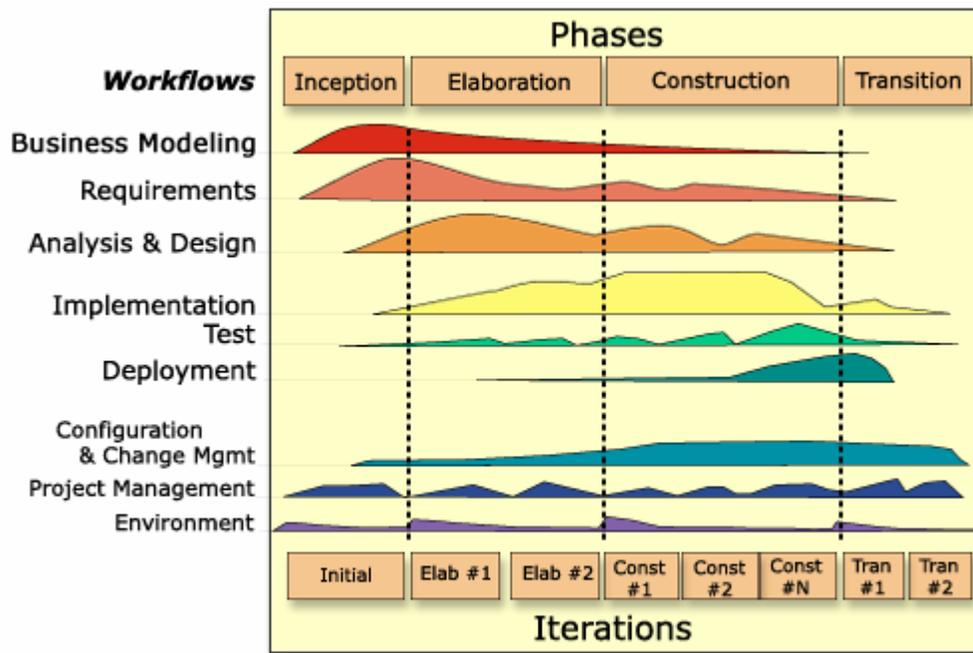


Figura 1.1. RUP en dos dimensiones.

Cada fase concluye en un hito mayor, un punto donde se realiza una evaluación para determinar si los objetivos de la fase han sido encontrados, una valoración satisfactoria permite al proyecto desplazarse a una siguiente fase. Los hitos según la figura 1.2 serían: inicio - Objetivos, elaboración - Arquitectura, construcción – Capacidad Operacional Inicial y transición - Release del producto.

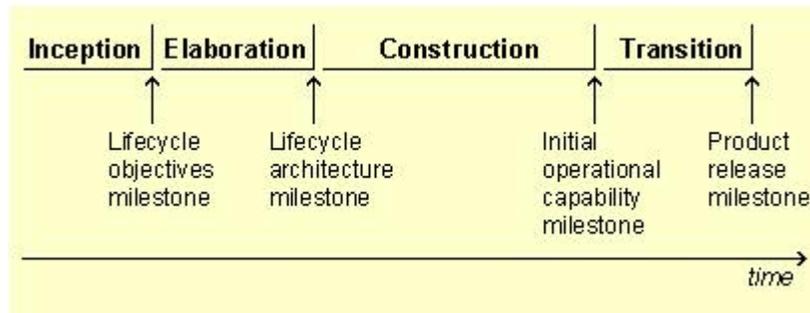


Figura 1.2. Las fases e hitos de un proyecto.

El RUP además contiene las 6 mejores prácticas de la ingeniería de software que están siendo utilizadas actualmente para el desarrollo exitoso de un proceso de desarrollo de software, desplegando las mismas ofrece al equipo de desarrollo un número significativo de ventajas pues lo provee de guías, plantillas y herramientas para una mejor forma de trabajo, estas mejores prácticas son (IBM, 2003):

- Desarrollo de software iterativo.
- Administrar requerimientos.
- Usar arquitectura de componentes.
- Modelar el software visualmente (UML).
- Verificar continuamente la calidad del software.
- Controlar cambios del software.

1.7. Variaciones del RUP.

Las variaciones del RUP vienen dadas de acuerdo a cambios en el número de fases y disciplinas, pues de acuerdo a instituciones que utilicen esta metodología incorporan sus simplificaciones o extensiones, entre las más conocidas se encuentran:

1.7.1. Agile Unified Process (AUP)

El AUP es una versión simplificada del RUP desarrollada por Scott Ambler. La misma describe un acercamiento fácil y simple de comprender al desarrollo de software de aplicaciones de negocio usando conceptos y técnicas ágiles que aun se mantienen válidos en el RUP (Ambler, 2005).

El ciclo de vida del AUP es consecutivo a lo largo, esta naturaleza es capturada por sus 4 fases: inicio, elaboración, construcción y transición. Además es iterativo a lo corto pues sus disciplinas son desarrolladas de una forma iterativa, las mismas son: Modelo que abarca las disciplinas del RUP:

Modelado del Negocio, Requerimientos, y Análisis y Diseño; Implementación; Prueba; Despliegue; Administración de la Configuración, conocida en RUP como Administración de la Configuración y Cambios; Administración del Proyecto y Ambiente. Y su última característica sería entregando versiones crecientes a lo largo del tiempo, es decir, a medida que concluya cada fase se obtiene al igual que en RUP un nuevo hito o versión del producto. (Ver figura 1.3)

En cuanto a los roles a desempeñar, actividades a realizar y artefactos a obtener el AUP hace una simplificación de los mismos pues esta conformado por valores y principios de la alianza ágil. Y en cuanto a las herramientas aconsejan a utilizar las más simples posibles, pues son mas fáciles de aprender, usar y compartir con otros. Ejemplo de estas herramientas simples seria la pizarra, probablemente la herramienta de modelado mas utilizada.

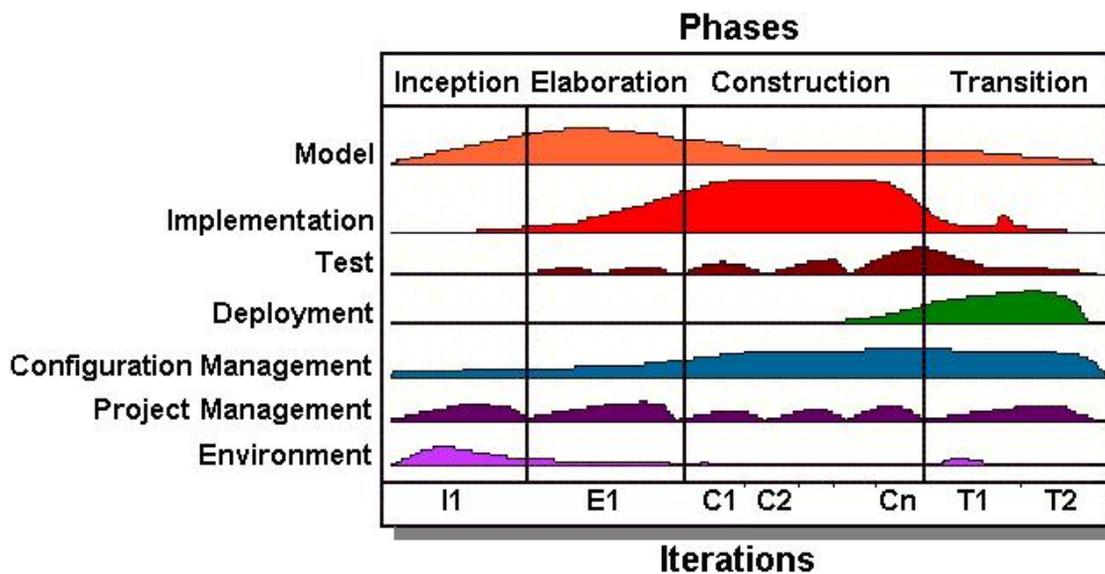


Figura 1.3. Agile Unified Process.

1.7.2. Basic Unified Process (BUP)

El BUP es una versión del RUP optimizada por IBM para proyectos pequeños, integrada por 3 a 6 miembros y una duración de 3 a 6 meses, además preserva sus características esenciales como desarrollo iterativo, desarrollo conducido por casos de usos y escenarios, administración de riesgos y un acercamiento céntrico a la arquitectura.

Con respecto al RUP el BUP contiene las disciplinas: Requerimientos, Arquitectura, Desarrollo, Prueba, Administración de Proyecto y Administración de Cambios; y excluye muchas como: Modelado del Negocio, Análisis y Diseño, Implementación, Administración de la Configuración y Ambiente, no quiere decir esto que muchas no se utilicen, por ejemplo la de Análisis y Diseño es absorbida por la de Arquitectura y Desarrollo; en cuanto a sus fases se mantienen las mismas: inicio, elaboración, construcción y transición.

En cuanto a los roles, actividades y artefactos el BUP con respecto al RUP también hace una simplificación de los mismos, los roles del BUP se agrupan en 6 en esta versión, y las actividades solo un bajo porcentaje de las mismas son consideradas y muchas fueron divididas en pasos e incluidas dentro de otras actividades principales.

1.7.3. Enterprise Unified Process (EUP).

El EUP es una extensión del RUP por Scott Ambler, John Nalbone, y Michael Vizdos para conocer las necesidades del mundo real de organizaciones medias a otras más grandes, además identifica sistemáticamente los problemas técnicos y de negocio que RUP falla al dirigir, y el cubrimiento por su parte de los mismos.

Esta extensión del RUP incluye dos nuevas fases, Producción y Retiro a las ya conocidas: inicio, elaboración, construcción y transición, aportando como consecuencia la obtención de 2 nuevos hitos al concluir las mismas y 8 nuevas disciplinas, la primera Soporte y Funcionamiento comprendida dentro de las de disciplinas de soporte, las demás serían las siguientes: Modelado del Negocio de la Empresa, Administración de Portafolio, Arquitectura de la Empresa, Reutilización Estratégica, Administración de la Persona, Administración de la Empresa y Mejoramiento del Proceso de Software dentro de la categoría Administración de la Empresa, como parte de las ya existentes. (Ver figura 1.4)

En cuanto a los roles, actividades y artefactos, al hacer el EUP una extensión del RUP trae como consecuencia no solo el aumento de los mismos sino también guías de herramientas recomendadas, esto ayuda a los usuarios a seleccionar las herramientas correctas de un rango amplio de opciones.

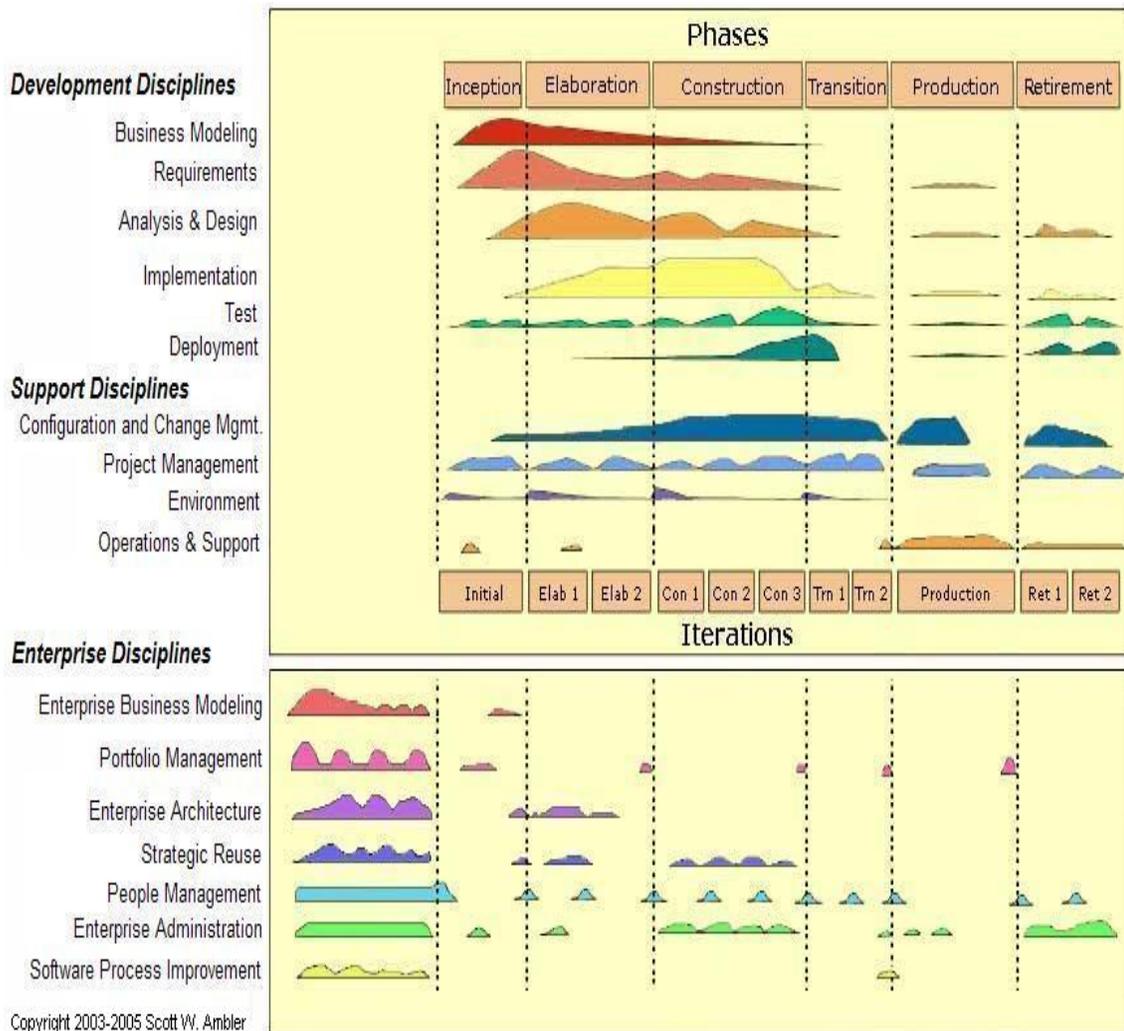


Figura 1.4. Enterprise Unified Process.

1.7.4. Open Unified Process (OpenUP).

El OpenUP es una plataforma de proceso de desarrollo de software anteriormente llamado BUP que cubre un conjunto amplio de necesidades de desarrollo y toma un acercamiento al desarrollo de software ágil, es el resultado del proyecto EPF, además mantiene características esenciales del RUP como desarrollo iterativo, desarrollo conducido por casos de usos y escenarios, administración de riesgos y un acercamiento céntrico a la arquitectura y contiene fundamentalmente un conjunto de simplificaciones de roles, actividades, artefactos y guías.

La más ligera versión del OpenUP es el OpenUP/Basic, un proceso iterativo de desarrollo de software mínimo, pues solo el contenido fundamental del proceso es incluido; completo, pues puede ser manifestado como un proceso entero para construir un sistema; y extensible, pues puede ser usado para añadir o ajustar el contenido del proceso. Además esta distribuido en 4 fases: inicio, elaboración, construcción y transición al igual que el RUP, donde se obtiene una versión del producto al culminar cada una de sus fases y 6 disciplinas como: Requisitos, Análisis y Diseño, Implementación, Pruebas, Administración de Configuración y Cambios y Administración del Proyecto. (Ver figura 1.5)

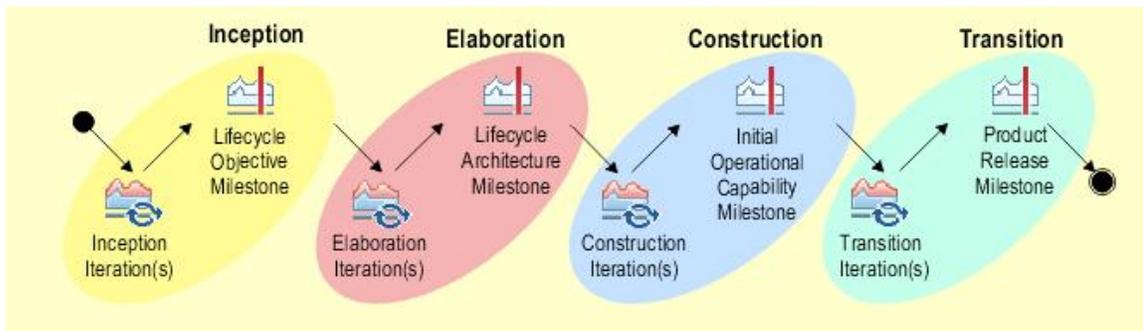


Figura 1.5. Open Unified Process.

1.8. Variaciones del RUP en la Universidad.

En la Universidad no se encontraron variaciones formalmente descritas de la metodología RUP.

1.9. RUP como un proceso configurable.

El RUP es un proceso configurable y por lo tanto puede adaptarse en una organización según las características que tenga, si es pequeña, mediana o grande, la documentación solicitada por el cliente, etc., al configurarlo se obtienen cambios en los roles, actividades, artefactos, etc.

La disciplina Ambiente del RUP se centra en las actividades necesarias para configurar el proceso para un proyecto. El propósito de las mismas es proporcionar a la organización desarrolladora de software de un ambiente de desarrollo de software de procesos y herramientas que soporten el equipo de desarrollo.

Es por lo antes expuesto que se analizan en esta disciplina los temas fundamentales para hacer esta adaptación según las necesidades y como implementarlo en conjunto con las herramientas de soporte.

Esta adaptación del proceso puede ser en dos niveles:

- A nivel organizacional: donde se modifica, mejora o configura un proceso común a ser usado por toda la organización. Tomando en consideración temas como: dominio de la aplicación, prácticas de reutilización y tecnologías principales. Para esto el RUP propone el componente Process Engineering Process (PEP) de la herramienta Rational Process Workbench™ (RPW), el mismo describe como usar el conjunto de herramientas de extensión del RUP para extender y ajustar la plataforma para una organización de desarrollo y como implementar este proceso ajustado en un ambiente de desarrollo de software.
- A nivel de proyecto: donde se toman los procesos de toda organización y se refinan para un proyecto dado, tomando en consideración el tamaño del proyecto.

En este trabajo para poder obtener una configuración para un módulo del proyecto se toma como base de la organización (Proyecto RyN) la propuesta del RUP clásico, sobre esta base se procede a refinar el mismo a nivel de proyecto (módulo).

Para poder crear la configuración el RUP propone 3 categorías, según lo que se desee hacer, las cuales son:

- Extendiendo el RUP creando plug-ins.

Esta categoría tiene como objetivo extender el RUP adicionándole el conocimiento de procesos adicionales para ajustar con las necesidades específicas del proceso del desarrollo de la organización o proyectos individuales aquellas áreas donde la cobertura del RUP es considerada insuficiente para el proyecto. El RPW habilita la creación de extensiones del RUP mediante 2 formas:

- Creando plug-ins estructurales.

Un plug-in estructural es un fragmento de un proceso de desarrollo de software que describe un dominio específico, tecnología o plataforma. Es una fracción de un proceso que extiende el modelo de proceso del RUP por la adición de elementos de procesos como roles, actividades, artefactos y flujos de trabajo. La herramienta que propone el RUP para la creación de estos plug-ins estructurales es el RUP Modeler™ un componente de el RPW.

- Creando plug-ins delgados.

Un plug-in delgado es un mecanismo para organizaciones para empaquetar sus valores organizativos como plantillas de artefactos, guías, ejemplos y otros valores reusables para

el consumo individual del proyecto. La herramienta que propone el RUP para la creación de estos plug-ins delgados es el RUP Organizer™ un componente de el RPW.

- Configurando el proceso seleccionando los componentes relevantes y añadirlos en la plataforma RUP.

Esta categoría tiene como objetivo ajustar el proceso para resolver las necesidades específicas de una organización o un proyecto individual. Esto implica la correcta selección de un conjunto de componentes de proceso y vistas apropiadas dentro de la configuración ocultando las partes de procesos irrelevantes. La herramienta que propone el RUP para hacer la configuración es el RUP Builder™. El artefacto resultante de las extensiones anteriores puede ser cargado en este producto.

- Instanciar la configuración del proceso en un proyecto.

Esta categoría se encarga de crear una instancia del proceso online para el proyecto, esto es conocido como Proceso Específico del Proyecto y se encarga de poner en marcha la configuración creada para el conocimiento de los miembros del equipo de desarrollo.

De acuerdo a las categorías antes descritas, primeramente la organización que adoptó el RUP clásico de un inicio hizo una extensión en cuanto a artefactos, este es el: documento de no conformidades, para el cual creo una plantilla. Además se prepararon 3 líneas directivas a seguir: estándar de codificación, estándar de diseño gráfico y estándar de diseño de base datos. Esta extensión no pudo ser concretada en ningún plug-ins para luego a través del RUP Builder crear la guía online pues no se cuenta con la herramienta RPW

Este trabajo se centra en realizar una configuración para los módulos nuevos a partir de la propuesta por la organización.

1.10. Selección de la variante.

Luego de un análisis de las más conocidas variaciones del RUP anteriormente descritas, se llega a la conclusión de que ninguna satisface las necesidades de este proyecto, pues no se quiere extender la plataforma en cuanto a disciplinas o fases como el EUP, ni se pretende eliminar mas de una disciplina como el BUP, AUP u OpenUP, se quiere agilizar el trabajo de acuerdo a las características del proyecto como su tamaño, personal, actividades a realizar y artefactos a obtener. Estas variaciones extienden o simplifican demasiado estas características, y no es el objetivo de este trabajo.

CAPÍTULO 2: SOLUCIÓN PROPUESTA

2.1 Introducción.

En el capítulo anterior se hizo una breve descripción de la necesidad de adoptar una metodología de desarrollo de software y las posibilidades que brinda la misma en la organización de un equipo de desarrollo para obtener una mayor producción de software a la par de productos de mejor calidad. Además se hizo una investigación acerca del surgimiento del proceso unificado hasta nuestros días, de las características propias de la metodología que se adoptó para el proyecto RyN y de su utilización mundialmente, de igual forma se analizaron las diferentes variaciones que ha tenido la misma en cuanto a fases y disciplinas.

En este capítulo se verá en detalles luego de adoptar en el proyecto el RUP clásico y su extensión en artefactos y plantillas, una variación del mismo para los nuevos módulos en una de sus disciplinas y una estructura de configuración propuesta según los roles, actividades y artefactos a utilizar a lo largo del ciclo de vida del producto.

2.2 Fases e Hitos.

En el capítulo anterior se hizo mención a las diferentes fases del RUP, con su objetivo y la obtención de sus hitos al final de cada una, en esta configuración propuesta se mantienen las mismas fases: inicio, elaboración, construcción y transición.

2.3 Disciplinas.

En cuanto a las disciplinas se seguirán las mismas del RUP con sus objetivos exceptuando la de análisis que se propone eliminarla pues no se considera necesaria en aras de agilizar el proceso, es decir que se llegaría a la representación de las clases de la disciplina de diseño a través del diagrama de casos de uso de la disciplina requerimientos, para detallar mejor esto ver figura 2.1.

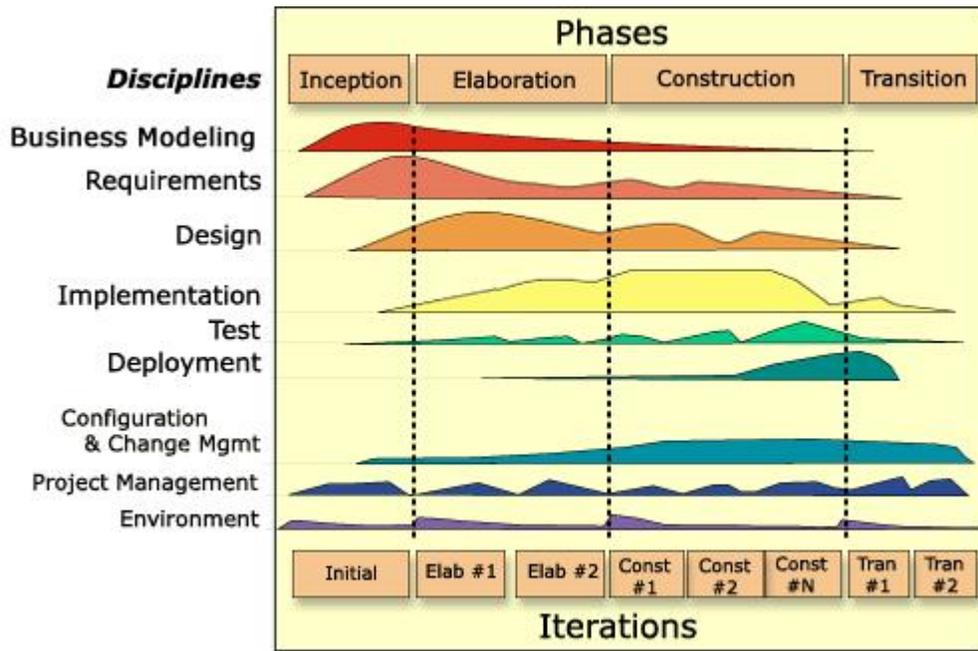


Figura 2.1. Cambio de RUP en dos dimensiones.

2.4 Roles y Actividades.

El primer término a analizar en la configuración serían los roles a desempeñar por los miembros del equipo de desarrollo y para introducir mejor el tema se brinda el siguiente concepto enunciado por el RUP: “Un rol es una definición abstracta de un conjunto de responsabilidades, para realizar actividades y producir artefactos, los roles son realizados típicamente por un individuo, o un conjunto de individuos trabajando juntos como un equipo por lo que un miembro del equipo de proyecto puede cumplir diferentes roles.” (Rational Software Corporation, 2003)

Como segundo término se detallarán las actividades que estos roles realizan, las mismas no son mas que “un conjunto de operaciones o tareas propias de una persona o entidad” (WordReference.com, 2005), estas actividades están estrechamente relacionadas y funcionalmente acopladas y son mejor realizadas por un individuo.

El RUP clásico en su totalidad propone una serie de roles que abarcan todo el ciclo de vida del proyecto, los mismos los agrupa según su funcionalidad, a continuación se da un breve resumen de ellos:

- **Analistas:** agrupa un conjunto de roles principalmente envueltos en la investigación y obtención de requerimientos, entre ellos se encuentran:

- Analista de proceso de negocio.
- Diseñador del negocio.
- Analista del sistema.
- Especificador de requerimientos.
- Desarrollador: organiza el conjunto de roles principalmente implicados en el diseño y la implementación del software, aquí se encuentran:
 - Arquitecto de software.
 - Diseñador.
 - Diseñador de interfaz de usuario.
 - Diseñador de capsula.
 - Diseñador de base datos.
 - Programador.
 - Integrador.
- Administradores: organiza el conjunto de roles principalmente implicados en la administración y configuración del proceso de ingeniería de software, dentro de ellos están:
 - Administrador del proyecto.
 - Administrador del control de cambios.
 - Administrador de la configuración.
 - Administrador de prueba.
 - Administrador de despliegue.
 - Ingeniero de procesos.
 - Revisor de la administración.
- Producción y Soporte: son aquellos no directamente relacionados con la administración, desarrollo y prueba de software, pero son necesarios para soportar el proceso de desarrollo de software o para producir materiales adicionales requeridos por el producto final, tales son los casos de:
 - Escritor técnico.
 - Administrador del sistema.
 - Especialista en herramientas.
 - Desarrollador de cursos.
 - Artista grafico.

- Probadores: organiza el conjunto de roles que tratan con las habilidades únicas específicas para prueba, aquí se encuentran:
 - Probador.
 - Analista de prueba.
 - Diseñador de prueba.
- Roles adicionales: son aquellos que no se acomodan dentro de ningún conjunto de roles anteriores.
 - Revisor.
 - Coordinador de revisión.
 - Revisor técnico.
 - Cualquier rol.
 - Stakeholder.

A partir de estos roles se hará entonces una agrupación de los mismos con actividades que se mantendrán, agregarán o eliminarán de acuerdo a las necesidades del proyecto. La configuración se detallará a continuación según las disciplinas del RUP:

- Modelación del Negocio
 - Analista: este rol abarca el rol de analista del negocio, además realiza actividades del diseñador del negocio, es el responsable de definir los casos de uso y actores del negocio y como ellos interactúan.
Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según este rol y en esta disciplina:
 - Capturar el vocabulario común del negocio: tiene como objetivo definir un vocabulario común que pueda ser usado en todas las descripciones textuales del negocio, especialmente en descripciones de casos de uso del negocio.
 - Identificar reglas del negocio: tiene como objetivo encontrar que metas pueden ser planeadas y administradas.
 - Encontrar actores y casos de usos del negocio: tiene como objetivo definir quién y qué interactuará con el negocio.
 - Encontrar trabajadores y entidades del negocio: tiene como objetivo describir como las realizaciones de los casos de uso del negocio son ejecutadas por trabajadores y entidades del negocio.

- Definir los requerimientos a automatizar: tiene como objetivo entender como las nuevas tecnologías pueden ser usadas para hacer el objetivo de la organización más efectivo y derivar los requerimientos del sistema de los artefactos del modelo del negocio.
 - Estructurar el modelo de casos de uso del negocio: tiene como objetivo extraer el comportamiento de los casos de usos del negocio que necesitan ser considerados como casos de uso abstractos.
- Especificador: este rol abarca el rol de diseñador del negocio, es el responsable de la realización de los casos de uso del negocio.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según este rol y en esta disciplina:

- Detallar un caso de uso del negocio: tiene como objetivo describir en detalle el mismo.
 - Detallar trabajadores y entidades del negocio: tiene como objetivo describir las responsabilidades del trabajador y garantizar que la entidad del negocio sea capaz de proporcionar el comportamiento requerido.
- Revisor: este rol abarca el rol de revisor técnico, es realizado por los roles administrador del módulo, ingeniero de procesos y administrador de pruebas, es el responsable de contribuir a la retroalimentación del proyecto mediante la supervisión de los artefactos.

Solo realiza la siguiente actividad de acuerdo a todas las propuestas por el RUP según este rol y en esta disciplina:

- Revisar el modelo de casos de uso y análisis del negocio: tiene como objetivo verificar que los resultados de estos modelos se adapten a las opiniones de los involucrados en el negocio.

- **Requerimientos**

- Analista: este rol abarca los roles de analista del sistema y diseñador de interfaz de usuario, además realiza actividades del especificador de requerimientos, es el responsable de definir los casos de uso y actores del sistema y como ellos interactúan, además es el responsable del recogimiento de los requerimientos de usabilidad y diseñar la interfaz de usuario candidata para conocer estos requerimientos.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según estos roles y en esta disciplina:

- Desarrollar la visión y capturar el vocabulario común: tiene como objetivo definir los objetivos futuros de la organización y pulir o ampliar el glosario anterior.
 - Encontrar actores y casos de usos del sistema: tiene como objetivo definir quién y qué interactuará con el sistema.
 - Desarrollar el plan de administración de requerimientos: tiene como objetivo crear un plan para documentar los mismos.
 - Detallar los requerimientos de software: tiene como objetivo recolectar, detallar y organizar el conjunto de artefactos que describen completamente los requerimientos de software del sistema.
 - Producir solicitudes de los Stakeholder: tiene como objetivo recoger las solicitudes sobre las necesidades que el sistema debe cumplir.
 - Estructurar el modelo de casos de uso del sistema: tiene como objetivo extraer el comportamiento de los casos de usos del sistema que necesitan ser considerados como casos de uso abstractos.
 - Diseñar la interfaz de usuario: tiene como objetivo producir un diseño de la interfaz de usuario para guía de los programadores.
 - Administrar dependencias: tiene como objetivo usar atributos y rastreabilidad de los requerimientos para asistir en la administración del alcance del proyecto y cambios en los requerimientos.
- Especificador: este rol abarca el rol de especificador de requerimientos, es el responsable de la realización de los casos de uso del sistema.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según este rol y en esta disciplina:

- Detallar un caso de uso del sistema: tiene como objetivo describir en detalle el mismo.
- Arquitecto de Software: este rol abarca el rol de arquitecto de software, es el responsable de priorizar los casos de uso.

Solo realiza la siguiente actividad de acuerdo a todas las propuestas por el RUP según este rol y en esta disciplina:

- Priorizar casos de uso: tiene como objetivo definir la selección del conjunto de escenarios y casos de uso que serán analizados en la iteración de acuerdo a su prioridad.
- Revisor: este rol abarca el rol de revisor técnico, es realizado por los roles administrador del módulo, ingeniero de procesos y administrador de pruebas, es el responsable de contribuir a la retroalimentación del proyecto mediante la supervisión de los artefactos.

Solo realiza la siguiente actividad de acuerdo a todas las propuestas por el RUP según este rol y en esta disciplina:

- Revisar los requerimientos: tiene como objetivo verificar que los resultados de los requerimientos se adapten a las opiniones de los clientes en el sistema.

- Diseño

- Arquitecto de Software: este rol abarca el rol de arquitecto de software, es el responsable de la arquitectura del software, la cual incluye las decisiones técnicas claves que conllevan a la mayor parte del diseño.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según este rol y en esta disciplina:

- Análisis de la arquitectura: tiene como objetivo definir una arquitectura candidata para el sistema basada en la experiencia obtenida de sistemas similares, además de definir patrones de arquitectura y mecanismos claves.
- Describir distribución: tiene como objetivo describir como las funcionalidades del sistema son distribuidas a través de los nodos físicos.
- Identificar e incorporar los elementos del diseño: tiene como objetivo analizar interacciones entre los casos de uso del sistema para identificar los elementos del diseño (clases, interfaces, etc.)
- Diseñador: este rol abarca el rol de diseñador y diseñador de la interfaz de usuario, es el responsable de diseñar una parte del sistema, teniendo en cuenta las restricciones del sistema, arquitectura y procesos de desarrollo para el proyecto, además es el responsable crear el prototipo del diseño de la interfaz de usuario candidata.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según este rol y en esta disciplina:

- Diseñar clase: tiene como objetivo diseñar las clases con sus atributos, métodos, asociaciones, etc. garantizando que las mismas proporcionen el comportamiento requerido de realización de casos de uso.
 - Diseñar subsistema: detallar el comportamiento interno del subsistema y documentarlo.
 - Diseñar caso de uso: tiene como objetivo refinar la realización de un caso de uso en términos de interacciones y las exigencias sobre las operaciones de las clases del diseño.
 - Prototipar la interfaz de usuario: tiene como objetivo crear un prototipo de la interfaz de usuario del sistema que valide su diseño contra los requerimientos funcionales y de usabilidad.
- Diseñador de Base Datos: este rol se mantiene según el RUP y es el responsable de diseñar el almacenaje de los datos persistentes a ser usados por el sistema.

Realiza la actividad propuesta por el RUP según este rol y en esta disciplina:

- Diseñar la base de datos: tiene como objetivo asegurarse de que los datos persistentes son almacenados consistente y eficientemente y definir el comportamiento que debe ser implementado en la base de datos.
- Revisor: este rol abarca el rol de revisor técnico, es realizado por los roles administrador del módulo, ingeniero de procesos y administrador de pruebas, es el responsable de contribuir a la retroalimentación del proyecto mediante la supervisión de los artefactos.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según este rol y en esta disciplina:

- Revisar la arquitectura: tiene como objetivo descubrir u observar cualquier riesgo en el cronograma, además de detectar algún defecto en el diseño arquitectónico o desajuste potencial entre los requerimientos y la arquitectura y evaluar una o varias cualidades específicas de la arquitectura como: funcionamiento, confiabilidad, seguridad, etc.

- Revisar el diseño: tiene como objetivo verificar que el modelo del diseño cumple a cabalidad los requerimientos del sistema y que sirva como una buena base para implementarlos.

- Implementación

- Arquitecto de Software: este rol abarca los roles de arquitecto de software e integrador del RUP, es el responsable de la arquitectura del software, la cual incluye las decisiones técnicas claves que conllevan a la mayor parte de la implementación en el proyecto y planeación de la integración.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según estos roles y en esta disciplina:

- Estructurar el modelo de implementación: tiene como objetivo establecer la estructura en la cual la implementación residirá y asignar responsabilidades para los subsistemas de implementación y su contenido.
- Planear la integración de subsistemas: tiene como objetivo planear el orden en el cual los elementos contenidos en un subsistema de implementación serán integrados.
- Integrar los subsistemas: tiene como objetivo integrar los elementos en un subsistema de implementación que luego será entregado para integrarlo al sistema.

- Diseñador: este rol abarca el rol de programador, es el responsable por el desarrollo y prueba de componentes, en acuerdo a los estándares adoptados por el proyecto.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según este rol y en esta disciplina:

- Implementar elementos del diseño: tiene como objetivo producir una implementación para parte del diseño o arreglar uno o más defectos.

- Revisor: este rol abarca el rol de revisor técnico, es realizado por los roles administrador del módulo, ingeniero de procesos y administrador de pruebas, es el responsable de contribuir a la retroalimentación del proyecto mediante la supervisión de los artefactos.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según estos roles y en esta disciplina:

- Revisar el código: tiene como objetivo verificar la implementación.

- Prueba

- Administrador de Pruebas: este rol abarca los roles analista, diseñador y administrador de pruebas del RUP, es el responsable del éxito de la prueba, este rol es defensor de prueba y calidad, planificación y administración de recursos, resolución de problemas que impiden el desarrollo de las pruebas, además es el responsable de identificar y definir las pruebas requeridas, el método y asegurar su implementación exitosa.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP de acuerdo a estos tres roles y en esta disciplina:

- Estar de acuerdo con la misión: tiene como objetivo negociar el uso más efectivo de recursos de prueba para cada iteración y aceptar un apropiado y factible conjunto de objetivos y entregables para la iteración.
- Identificar los objetivos de prueba: tiene como objetivo identificar los elementos individuales del sistema, hardware y software que necesitan ser probados.
- Definir necesidades de rastreabilidad y evaluación: tiene como objetivo definir la estrategia de evaluación para el esfuerzo de prueba.
- Definir el método de prueba: tiene como objetivo identificar cada técnica específica que será empleada para facilitar la experimentación deseada.
- Definir detalles de prueba: tiene como objetivo definir las condiciones individuales necesarias para realizar una prueba en un contexto específico.
- Definir elementos con posibilidad de evaluación: tiene como objetivo definir el diseño de los requerimientos del software que necesitan ser encontrados para permitirle al software ser físicamente probado.
- Estructurar la implementación de prueba: tiene como objetivo establecer la estructura en cual la implementación de la prueba residirá y asignar responsabilidades.
- Determinar resultados de pruebas: tiene como objetivo identificar y capturar el resultado de las pruebas detallado y proponer acciones correctivas para resolver fallas en la calidad.

- Probador: este rol se mantiene de acuerdo al RUP y es el responsable durante las actividades principales de las pruebas, el cual incluye la conducción de las pruebas necesarias y el registro del resultado de la prueba.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según este rol y en esta disciplina:

- Ejecutar prueba: tiene como objetivo ejecutar las colecciones apropiadas de pruebas requeridas para evaluar la calidad del producto y capturar los resultados.
- Elaborar documento de No conformidades: tiene como objetivo crear un documento con el resultado de las pruebas para que luego el Administrador de Prueba elabore las solicitudes de cambio.

- Despliegue

- Especificador: este rol abarca el rol de escritor técnico del RUP, es el responsable de generar el material de soporte para los usuarios finales tales como: guías de usuarios, textos de ayuda, etc.

Realiza la siguiente actividad de acuerdo a todas las propuestas por el RUP según este rol y en esta disciplina:

- Desarrollar materiales de soporte: tiene como objetivo desarrollar el material de soporte a usuarios finales.

- Diseñador: este rol abarca el rol de programador, es el responsable del desarrollo de los artefactos de instalación.

Realiza la siguiente actividad propuesta por el RUP según este rol y en esta disciplina:

- Desarrollar artefactos de instalación: tiene como objetivo producir todo el software requerido para instalar y desinstalar el producto rápido, fácil y con toda seguridad sin afectar otras aplicaciones o características de sistema.

- Administración de la Configuración y Cambios

- Administrador de la Configuración: este rol se mantiene al igual que el RUP, es el responsable de suministrar la completa infraestructura de administración de la configuración y ambiente al equipo de desarrollo del producto.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según este rol y en esta disciplina:

- Establecer el ambiente de administración de la configuración: tiene como objetivo establecer un ambiente donde el producto pueda ser construido y desarrollado, esto implica asignación de recursos de maquina (servidores y espacio en discos) e instalar las herramientas para esta configuración, además de la creación de repositorios.
 - Escribir el plan de administración de la configuración: tiene como objetivo describir todas las actividades de configuración y control de cambio a ser ejecutadas durante el ciclo de vida del proyecto y documentar como las actividades de la configuración relacionadas con el producto son planeadas, implementadas, controladas y organizadas.
 - Crear una unidad de despliegue: tiene como objetivo construir documentos y artefactos de instalación.
- Administrador del Control de Cambios: este rol se mantiene al igual que el RUP, se encarga de supervisar el proceso de control de cambios.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según este rol y esta disciplina:

- Revisión de solicitud de cambios: tiene como objetivo determinar si la solicitud de cambio puede ser aceptada o rechazada, para aceptarlas mediante esta actividad se evalúa la prioridad y así sucesivamente se determina si el cambio es alcanzado para la actual versión.
 - Establecer el proceso de control de cambios: tiene como objetivo crear un documento que lleve todo este control de cambios para que los clientes estén informados del estado del producto y sus cambios.
- Administrador de Prueba: este rol abarca el rol de Analista de Prueba, es el responsable de verificar cambios.

Solo realiza la siguiente actividad según este rol y esta disciplina:

- Verificar cambios en un Build: tiene como objetivo confirmar que una solicitud de cambio ha sido completada.

- Administración del Proyecto

- Administrador del Módulo: este rol abarca los roles de Administrador de Proyecto y Revisor de la Administración del RUP y se encarga de proyectar, administrar y asignar recursos, configurar prioridades, coordinar interacción con los clientes y usuarios, y mantener el equipo de proyecto enfocado. Además establece un conjunto de prácticas que aseguran la integridad y la calidad de los artefactos del proyecto.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP según este rol y esta disciplina y se le adiciona la última:

- Organizar fases e iteraciones: tiene como objetivo definir un conjunto de iteraciones dentro de las fases del proyecto e identifica los objetivos de cada una de ellas, además de desarrollar un cronograma de entrega para el proyecto.
- Identificar y evaluar riesgos: tiene como objetivo identificar, analizar y priorizar los riesgos del proyecto y crear un documento llamado Plan de Administración de Riesgos donde estén determinados los mismos.
- Desarrollar el plan de iteraciones: tiene como objetivo definir los entregables de cada iteración de acuerdo a las responsabilidades asignadas.
- Desarrollar el plan de control de la calidad: tiene como objetivo crear un plan documentado para controlar la calidad de las actividades en el proyecto.
- Desarrollar el plan de administración de riesgos: tiene como objetivo crear un plan para controlar los mismos.
- Recopilar documentos para conformar el Plan de Desarrollo del Software: tiene como objetivo coordinar el desarrollo de diferentes documentos para publicarlos en el documento Plan de Desarrollo del Software.
- Adquirir número de empleados: tiene como objetivo proveer al proyecto de recursos humanos.
- Definir la organización del proyecto y proveer su personal: tiene como objetivo definir una estructura organizacional para el proyecto en números en todas las iteraciones, es decir, establecer el número de personal del equipo de desarrollo, de acuerdo a roles y nivel de experiencia.

- Revisar hitos del ciclo de vida: tiene como objetivo revisar el estado del proyecto al final de cada fase.
- Monitorear estado del proyecto: tiene como objetivo capturar y evaluar el estado actual del proyecto.
- Capacitar a los miembros del equipo: tiene como objetivo proporcionar diferentes cursos para que los miembros del equipo de desarrollo estén más preparados en la labor a desarrollar.

- Ambiente

- Ingeniero de procesos: este rol abarca el rol de ingeniero de procesos y tiene como meta equipar al equipo de proyecto de un eficiente proceso de desarrollo y asegurarse de que los miembros del equipo no están imposibilitados en la realización de su trabajo.

Solo realiza las siguientes actividades de acuerdo a todas las propuestas por el RUP de acuerdo a estos dos roles y en esta disciplina:

- Ajustar el proceso de desarrollo al proyecto: tiene como objetivo ajustar el tamaño del proceso de desarrollo de software de acuerdo a las necesidades del proyecto, proporcionando una relevante y accesible descripción del proceso para los miembros del proyecto.
- Preparar plantillas para el proyecto: tiene como objetivo disponer de las plantillas o desarrollar nuevas a utilizar en el proyecto.
- Preparar líneas directivas específicas para el proyecto: tiene como objetivo disponer de las guías existentes para que estén accesibles por los miembros del equipo.
- Implementar caso de desarrollo: tiene como objetivo desarrollar un caso de desarrollo que describa el proceso de desarrollo del software para el proyecto.

El proyecto RyN al adoptar el RUP clásico mantiene determinados roles para realizar diferentes actividades de igual similitud en los módulos que desarrolle, tales son los casos del Administrador de Despliegue, el Especialista en Herramientas y los Stakeholder, a continuación de dará una breve explicación de sus ocupaciones:

- Administrador de Despliegue: este rol abarca los roles administrador de despliegue y desarrollador de cursos del RUP, es el responsable de planificar la transición del producto a

la comunidad del usuario, además de desarrollar el material de adiestramiento para capacitar a los usuarios y enseñarlos a usar el producto.

- Especialista en herramientas: este rol abarca los roles de especialista en herramientas y administrador de sistemas del RUP, es el responsable de proveer las herramientas para el proyecto, su obtención y selección, así como su configuración y control de que las mismas funcionen, además del desarrollo de la infraestructura, que incluye hardware y software.
- Stakeholder: este rol abarca el rol de stakeholder del RUP, es el responsable de la representación de un grupo de necesidades que deben ser satisfechas por el proyecto.

2.5 Artefactos.

Como tercer término se analizarán los artefactos generados de acuerdo a los roles y sus actividades, los mismos son producto del trabajo final o intermedio y son producidos y usados durante el proyecto capturando diferentes informaciones.

El RUP propone un número grande de ellos, de los cuales para esta configuración se mantuvieron algunos y se agregaron otros como se explicó en el capítulo anterior, a continuación se detallarán los artefactos según el orden de las disciplinas del RUP:

- Modelado del Negocio
 - Glosario del negocio: define los términos importantes usados en el modelado del negocio, una parte del proyecto.
 - Reglas del negocio: las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio.
 - Modelo de casos de uso del negocio: es un modelo que describe los procesos de un negocio (casos de uso del negocio) y su interacción con elementos externos (actores), donde las funciones que el negocio pretende realizar y su objetivo básico es describir cómo el negocio es utilizado por sus clientes y socios. El mismo esta conformado por:
 - Actor del negocio: es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

- Caso de uso del negocio: un proceso o caso de uso del negocio es un grupo de tareas relacionadas lógicamente que se llevan a cabo en una determinada secuencia y manera y que emplean los recursos de la organización para dar resultados en apoyo a sus objetivos.
 - Modelo de análisis del negocio: describe las realizaciones de casos de uso del negocio por la interacción entre los trabajadores y entidades del negocio. Este sirve como abstracción de cómo los trabajadores y entidades del negocio deben relacionarse y cómo ellos colaboran para desarrollar los casos de uso del negocio. Este modelo está compuesto por:
 - Realización de casos de uso del negocio: Mientras un caso de uso del negocio describe que pasos deben realizarse para producir un resultado observable para ciertos actores del negocio, una realización de caso de uso describe cómo estos pasos se realizan dentro de la organización. Se describen los casos de uso del negocio desde una perspectiva externa, mientras que la realización de un caso de uso se describe desde una perspectiva interior. Las realizaciones pueden ser a través de:
 - ✓ Descripción textual: es un documento donde se hace la descripción del caso de uso en detalles.
 - ✓ Roles participantes: es una lista con nombres y breve descripción de los actores, trabajadores, entidades y eventos del negocio que aparecen como roles u objetos en los diagramas de actividad y clases.
 - ✓ Diagrama de actividad: describe un proceso que explora el ordenamiento de las tareas o actividades que cumplen los objetivos del negocio y que satisfacen el propósito entre los actores externos y los trabajadores internos del negocio.
 - ✓ Diagrama de clases: muestra las asociaciones, agregaciones y generalizaciones entre los trabajadores y entidades del negocio.
 - Trabajadores del negocio: es una abstracción de un humano o sistema del software que representa un papel dentro de las realizaciones de caso de uso. Un trabajador del negocio colabora con otros trabajadores, se notifica de eventos del negocio y manipula las entidades del negocio para realizar sus responsabilidades.

- Entidades del negocio: son una abstracción de elementos persistente dentro del negocio, estas representan algo que los trabajadores toman, manipulan, modifican y utilizan.
 - Sistemas de negocio: encapsula un conjunto de roles y recursos que juntos cumplen cabalmente un propósito específico y define un conjunto de responsabilidades con las cuales estos propósitos pueden ser alcanzados.
 - Eventos del negocio: representa una ocurrencia significativa en las actividades del negocio que requieren acción inmediata.
- Requerimientos
 - Glosario: define los términos importantes utilizados en el proyecto, es un refinamiento del glosario del negocio.
 - Visión: define la perspectiva de los involucrados del producto a desarrollar, detallando en sus términos las necesidades y rasgos principales, además de ser un refinamiento de la visión del negocio.
 - Requerimientos del software: condición o capacidad que el sistema debe cumplir.
 - Atributos de los requerimientos: es un almacén de requerimientos de proyecto, atributos y dependencias para asistir el cambio dirigido desde una perspectiva de los requerimientos.
 - Especificación de los requerimientos de software: captura los requerimientos de software para el sistema completo.
 - Plan de administración de requerimientos: describe los artefactos de requerimientos, los tipos y sus respectivos atributos, detallando la información a recolectar y los mecanismos de control a ser usados para la medición y cambios controlados en los requerimientos del producto.
 - Modelo de casos de uso: es un modelo de las funciones deseadas del sistema que contiene actores, paquetes, casos de uso y sus relaciones. Es usado para como una entrada esencial a actividades en diseño y prueba, esta conformado por:
 - Paquetes de casos de uso: es una colección de casos de uso, actores, relaciones, diagramas y otros paquetes, usados para estructurar el modelo de casos de uso y dividirlo en pequeñas partes.

- Casos de uso: se define como un conjunto de instancias de casos de uso donde cada instancia es una secuencia de acciones a ejecutar en un sistema que producen un resultado observable de importancia para un actor en particular.
 - Actores: se define como un conjunto lógico de roles que los usuarios del sistema pueden jugar cuando interactúan con él. Una instancia de un actor puede ser jugada por un individuo cualquiera o un sistema externo.
 - Realización de casos de uso del sistema:
 - Descripción textual: lleva el mismo contenido de la descripción textual del negocio lo que se le agrega al final el prototipo de la interfaz.
 - Roles participantes: es una lista con nombres y breve descripción de los actores y entidades del sistema que aparecen como roles u objetos en los diagramas de actividad.
 - Diagrama de actividad: describe un proceso que explora el ordenamiento de las tareas o actividades que cumplen los objetivos del sistema y que satisfacen el propósito entre los actores externos y el sistema.
 - Prototipo de interfaz de usuario (diseño): es un ejemplo de diseño de la interfaz de usuario.
 - Solicitudes de los Stakeholder: este artefacto contiene cualquier tipo de solicitudes de un Stakeholder que el sistema tiene que desarrollar.
- Diseño
 - Modelo del Diseño: es un modelo de objetos que describe la realización de los casos de uso y es usado como una abstracción del modelo de implementación y su código fuente y como una entrada esencial en actividades de implementación y prueba, además está conformado por:
 - Realización de casos de uso del diseño: describe como un caso de uso particular es realizado dentro del modelo del diseño en términos de colaboración de objetos.
 - ✓ Diagramas de interacción: son usados para mostrar como los objetos actúan para ejecutar el comportamiento de un caso de uso en particular. Existen dos formas para esto:
 - Diagrama de colaboración: destaca la organización de los objetos que participan en una interacción.

- Diagrama de secuencia: destaca la ordenación temporal de los mensajes.
 - ✓ Objetos participantes: breve descripción y clasificación de los objetos participantes.
 - Clases del diseño: es la descripción de un conjunto de objetos que comparten las mismas responsabilidades, relaciones, operaciones, atributos y semántica.
 - Paquetes del diseño: es una colección de clases, relaciones, realizaciones de casos de uso, diagramas y otros paquetes y es utilizado para estructurar el modelo del diseño en partes más pequeñas.
 - Subsistemas del diseño: son una forma de organizar los artefactos del modelo de diseño en piezas más manejables. Puede constar de clases del diseño, realizaciones de casos de uso y otros subsistemas.
 - Modelo de Datos: describe la representación lógica y física de los datos persistentes usados por la aplicación.
 - Documento de la arquitectura del software: proporciona una descripción comprensiva de la arquitectura del sistema, usando un número de vistas arquitectónicas diferentes para representar los aspectos desiguales del sistema.
 - Modelo de Despliegue: muestra la configuración de los nodos procesadores y los dispositivos a poner en funcionamiento y los enlaces de comunicación entre ellos mediante los protocolos.
 - Prototipo de interfaz de usuario (funcional): es un ejemplo de interfaz de usuario construido para explorar y/o validar el diseño de la interfaz de usuario.
- Implementación
 - Modelo de Implementación: representa la composición física de la implementación en términos de subsistemas de implementación y elementos de implementación (archivos de código fuente y ejecutables, datos, etc.)
 - Subsistemas de implementación: es un conjunto de elementos de implementación para estructurar el modelo de implementación en pequeñas partes que puedan ser separadamente integradas y probadas.

- Elementos de implementación: son la parte física que construye una implementación incluyendo archivos, directorios, archivos de código de software (código binario o ejecutable), ficheros de datos, archivos de documentación, como archivos de ayuda en línea.
 - Build: es una versión operacional del sistema o parte de un sistema que demuestra un subconjunto de capacidades que serán proporcionadas en el producto final.
 - Plan de Construcción e Integración: proporciona un plan detallado para la integración dentro de una iteración.
- Prueba
 - Plan de prueba: contiene la definición de metas y objetivos de prueba dentro del alcance del proyecto, los objetos a ser probados, la estrategia a ser tomada, los recursos requeridos y los entregables a producir, además de los responsables de las tareas previstas.
 - Resultados de prueba: es una colección de información resumida determinada del análisis de una o más anotaciones de prueba o solicitudes de cambio, proporcionando una valoración relativamente detallada de la calidad de los objetos de prueba.
 - Datos de prueba: son la definición de una colección de valores de entrada de prueba que son usados durante la ejecución de una prueba.
 - Casos de prueba: es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular ó una función esperada.
 - No conformidades: documento donde se muestra el incumplimiento de los requerimientos.
- Despliegue
 - Producto: es definido como una unidad de despliegue que debe ser empaquetada para la venta y distribución. El mismo esta conformado por:
 - Lista de materiales: describen las partes constituyentes y los cambios hechos de una versión determinada de un producto y donde pueden ser encontradas las partes físicas.
 - Artefactos de instalación: se refieren al software e instrucciones documentadas requeridas para instalar el producto.

- Unidad de despliegue: documentos y artefactos de instalación suficientemente completos para ser descargados, instalados y corridos en un nodo.
 - Material de soporte para usuarios finales: ayudan a los usuarios finales en conocimiento, uso, funcionamiento y manutención del producto.
 - Materiales de entrenamiento: hace referencia al material que es usado en capacitación o cursos para ayudar al usuario final con el uso del producto, funcionamiento y/o mantenimiento.
 - Plan del despliegue: describe el conjunto de tareas necesarias para instalar y probar el producto desarrollado tal que pueda ser transformado eficazmente a la comunidad del usuario.
- Administración de la configuración y control de cambios
 - Plan de administración de la configuración: describe todas las actividades de configuración y control de cambio a ejecutar durante el ciclo de vida del producto, además detalla el cronograma de las actividades, las responsabilidades asignadas y los recursos requeridos, incluyendo el número del personal del equipo de desarrollo y herramientas.
 - Repositorio del proyecto: almacena todas las versiones de archivos y directorios del proyecto, además es un recurso global que necesita ser accedido por los integrantes del equipo de desarrollo.
 - Solicitud de cambios: propone cambios a los artefactos a desarrollar, es usado para documentar y rastrear defectos, solicitud de mejoras y cualquier otro tipo de solicitud de cambio al producto.
- Administración del proyecto
 - Plan de desarrollo del software: es un artefacto compuesto, comprensivo que recoge toda la información requerida para administrar el proyecto, describe el acercamiento al desarrollo del software y es el plan de alto nivel generado y usado por los administradores para dirigir el esfuerzo de desarrollo. El incluye un numero de artefactos como:
 - Plan de administración de riesgos: detalla como administrar los riesgos asociados con un proyecto, las tareas de administración de riesgos que serán realizadas, las responsabilidades asignadas y cualquier recurso adicional requerido para esta actividad

- Plan de control de la calidad: es un artefacto que provee una vista clara de cómo el producto, artefacto y calidad del proceso deben ser asegurados.
 - Plan de iteraciones: es un conjunto de actividades y tareas en una secuencia de tiempo, con recursos asignados y comprendiendo dependencias de tareas para la iteración.
 - Lista de riesgos: es una lista clasificada de riesgos conocidos y abiertos para el proyecto, clasificada en orden de importancia decreciente y asociado con mitigaciones específicas o acciones de contingencia.
 - Registro de revisión: este es un documento creado para capturar los resultados de una actividad de revisión en la cual uno o más artefactos son revisados.
- Ambiente
 - Proceso de desarrollo: es una configuración de la plataforma fundamental RUP que reúne las necesidades del proyecto en ella, además provee una guía y apoyo para los miembros del equipo de desarrollo.
 - Caso de desarrollo: describe el proceso de desarrollo que ha sido seleccionado a seguir en el proyecto.
 - Plantillas específicas del proyecto: son las plantillas producidas para apoyar la producción de artefactos e informes usados en el proyecto.
 - Líneas directivas específicas del proyecto: proporcionan una orientación en como realizar una actividad segura o un conjunto de actividades en el contexto del proyecto.

Para un mejor entendimiento de la configuración propuesta ver el anexo 1.

CAPÍTULO 3: ANÁLISIS DE RESULTADOS

3.1 Introducción.

En el capítulo anterior se detalló la configuración propuesta de acuerdo a los roles, actividades y artefactos. En este capítulo se verá una experiencia de desarrollo en la que se utilizó esta configuración, en la misma el autor del presente trabajo se desempeñó con el rol de analista e ingeniero de procesos.

3.2 Proceso de desarrollo.

La adaptación del proceso de desarrollo obtenida es un artefacto para el módulo Bienes del proyecto RyN y sería la configuración propuesta en el capítulo II que cumple las necesidades del módulo en cuestión de acuerdo a disciplinas, roles, actividades y artefactos y además la adopción de las plantillas y guías específicas del proyecto de acuerdo a esa configuración planteada.

Este nuevo proceso de desarrollo ajustado a las necesidades del proyecto es de gran importancia realizarlo al comienzo del mismo pues provee una guía de soporte y organización para la realización de actividades y obtención de artefactos por los miembros del equipo.

A continuación se detallarán 3 artefactos que recogen la configuración del módulo. Estos son las plantillas y líneas directivas específicas del proyecto y el caso de desarrollo.

- Plantillas específicas del proyecto: son para documentar artefactos e informes usados en el proyecto, y son elegidas de acuerdo a los entregables del mismo.
- Líneas directivas específicas del proyecto: son escogidas de acuerdo a los entregables del proyecto también, las mismas suministran una guía de cómo ejecutar una actividad o un conjunto de actividades.

Estas plantillas y líneas directivas proveen una guía de como producir artefactos seguros de acuerdo al contexto del proyecto, se aseguran de que los artefactos son desarrollados consistentemente y siguen las convenciones y estilos definidos. Por consiguiente para los nuevos módulos del proyecto RyN se mantienen las mismas plantillas y líneas directivas definidas a nivel organizacional (proyecto RyN) pero de acuerdo entonces a los entregables definidos.

- Caso de desarrollo: describe el proceso de desarrollo seleccionado en el proyecto, en su interior muestra los artefactos a utilizar y por consecuencias las plantillas y líneas directivas a seguir.

3.3 Módulo Bienes.

El módulo Bienes, del proyecto RyN es dentro del software de “Administración Contable” uno de los más importantes, pues logra el control económico de los bienes muebles e inmuebles, así como el cálculo de su depreciación. Esto se desarrollará en la oficina de Administración Contable y en los registros (en cada estado) con diferentes funcionalidades en estos dos lugares.

3.4 Artefacto: Caso de desarrollo, del módulo Bienes.

Ver Anexo 2.

3.5 Opinión del Jefe de Proyecto.

El proyecto de modernización de los Registros y Notarías de la República Bolivariana de Venezuela constituye una solución llave en mano que aporta todos los elementos para la gestión de los procesos que se desarrollan en las oficinas registrales adscritas al Ministerio del Poder Popular para Relaciones Interiores y Justicia. La solución incluye:

2 módulos registrales (Mercantil y Público).

1 módulo de Servicio Autónomo para las gestiones propias de configuración, control y estadísticas de los servicios que se prestan.

1 Sistema de Administración Financiera que está formado por varios módulos que automatizan la contabilidad, las compras, el presupuesto, la recaudación y el control de los Bienes muebles e inmuebles, entre otros.

En la primera fase de desarrollo e implantación de la solución ha sido requerido, por parte de la contraparte cliente, un grupo de módulos que por su importancia son imprescindibles para la puesta en marcha del proyecto. El módulo para el control y la gestión de Bienes es uno de los imprescindibles y la calidad del proceso de desarrollo garantizó que en las presentaciones y validaciones con los clientes se alcanzara la satisfacción de estos. Es de destacar que el proyecto no ha comenzado la fase de despliegue y por lo tanto no existe una estadística real de resultados; sin embargo, se han efectuado laboratorios de aceptación que han sido superados y se ha destacado la calidad del producto que se propone al cliente.

La dirección del proyecto y los clientes manifiestan su conformidad con la modelación e implementación del módulo de Bienes.

Ing. Oscar Camacho Acosta
Líder Sub proyecto de Software
Proyecto Registros y Notarías

CONCLUSIONES

Con el desarrollo de este trabajo se han visto cumplidos los objetivos planteados para los nuevos módulos del proyecto RyN pues:

- Se ha propuesto una configuración de la metodología RUP que soluciona el problema declarado.
- Se realizó un ajuste a las disciplinas propuestas por el RUP.
- Se definieron los roles a desempeñar por los miembros del equipo.
- Se definieron las actividades a realizar y por consecuencia los artefactos a obtener por las disciplinas del RUP.

RECOMENDACIONES

En los objetivos planteados no se abarcaron todos los temas relacionados para la propuesta de configuración, por lo que se hacen las siguientes recomendaciones:

- Investigar sobre las herramientas de apoyo al proceso de desarrollo de software como complemento de esta configuración.
- Adquirir la herramienta RPW para poder insertar los nuevos artefactos, con sus plantillas y las nuevas directivas a seguir en plug-ins.
- Cargar estos plug-ins en la herramienta RUP Builder y configurar el RUP mediante la misma para poder realizar la ayuda online, pues es la forma de poner en práctica la configuración propuesta para el uso de los miembros del equipo de desarrollo.

Además se recomienda desarrollar la configuración propuesta al inicio del proyecto, pues hace posible seguir una organización del trabajo del equipo.

BIBLIOGRAFÍA

- Metodologías de desarrollo de Software. **Sanchez, María A. Mendoza. 2004.** s.l. : Grupo Informatizate, 2004.
- The New Methodology . **Martin Fowler. 2000.** s.l. : Martin Fowler, 2000.
- **Ambler, Scott W. 2002.** History of the Unified Process. Enterprise Unified Process (EUP) Home Page. [Online] Ambyssoft Inc, 2002. [Consult: Mayo 7, 2007.]
<http://www.enterpriseunifiedprocess.com/essays/history.html>.
- **Cingo.** Cingo. [Online] [Consult: Abril 28, 2007.] <http://www.cingo.com.br/>.
- **Datasul. 1978.** Datasul. [Online] Datasul S.A, 1978. [Consult: Abril 28, 2007.]
<http://www.datasul.com.mx/aliancas/index.php>.
- **Dybox.** Nuestra empresa. Dybox. [Online] [Consult: Abril 28, 2007.] <http://www.dybox.cl/index.html>.
- **Kada Software.** Kada Software. [Online] Kada Software. [Consult: Abril 28, 2007.]
http://www.kadasoftware.com/quienes_somos.html.
- **Rational Software Corporation. 2003.** Rational Unified Process. [Online] Rational Software Corporation, 2003. [Consult: Marzo 1, 2007.] \Rational\RationalUnifiedProcess\index.htm.
- **Ambler, Scott W. 2005.** The Agile Unified Process v1.1. Agile UP. [Online] Ambyssoft Inc, 2005. [Consult: Abril 28, 2007.] <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.
- Simple Tools for Software Modeling. **Ambler, Scott W. 2007.** s.l. : Ambyssoft Inc, 2007.
- Basic Unified Process: A Process for Small and Agile Projects. **Balduino, Ricardo. 2005.** s.l. : IBM, 2005.
- Extending the Rational Unified Process with the Enterprise Unified Process. **Ambler, Scott W. 2006.** 2006.
- Adopting the Enterprise Unified Process (EUP). **Ambler, Scott W. 2004.** s.l. : Ambyssoft Inc, 2004.
- OpenUP/Basic. **Sewerin, Michael. 2007.** 2007.
- **Eclipse Contributor. 2004.** OpenUP/Basic. [Online] Eclipse contributor, 2004. [Consult: Abril 28, 2007.]

REFERENCIAS BIBLIOGRÁFICAS

- **Alexei Zubizarreta. 2005.** Visión del Proyecto Registros y Notarías. Cuba : UCI, 2005.
- **Ambler, Scott W. 2005.** The Agile Unified Process (AUP). s.l. : Ambyssoft Inc, 2005.
- **Armstrong Process Group. 2005.** OpenUP/Basic. 2005.
- Basic Unified Process: A Process for Small and Agile Projects. **Balduino, Ricardo. 2005.** s.l. : IBM, 2005.
- **IBM. 2003.** Rational Unified Process: Best practices for software development teams. 2003.
- Introducing IBM Rational Method Composer. **Per Kroll. 2005.** s.l. : IBM Rational, 2005.
- **Ivar Jacobson, Grady Booch and James Rumbaugh. 2000.** El proceso unificado de desarrollo de software. s.l. : Addison-Wesley, 2000. ISBN 84-7829-036-2.
- **James Rumbaugh, Ivar Jacobson and Grady Booch. 2000.** El lenguaje unificado de modelado. Manual de referencia. s.l. : ADDISSON-WESLEY IBEROAMERICANA, 2000. 978-84-7829-037-6.
- Methodologies in IT: Comprehension, Selection, and Implementation. **Richard Murch. 2005.** s.l. : IBM Press, 2005.
- **Patón, Dr. Eduardo Fernández-Medina. 2006.** Alarcos. [Online] 2006. [Cited: Mayo 6, 2007.] <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>.
- **Pressman, Roger S. 2001.** Ingeniería del Software, Un enfoque práctico. s.l. : MacGraw-Hill, 2001.
- The Eclipse Process Framework project. **Per Kroll. 2005.** s.l. : IBM Rational, 2005.
- **Rational Software Corporation. 2003.** Rational Unified Process. [Online] Rational Software Corporation, 2003. [Cited: Marzo 1, 2007.] \Rational\RationalUnifiedProcess\index.htm.
- **WordReference.com. 2005.** Diccionario de la lengua española. Madrid : Espasa-Calpe S.A., 2005.

ANEXO 1. CONFIGURACIÓN PROPUESTA DE ROLES, ACTIVIDADES Y ARTEFACTOS DE ACUERDO A LAS DISCIPLINAS DEL RUP.

DISCIPLINA	ROLES	ACTIVIDADES	ARTEFACTOS
Modelación del Negocio	Analista	<ul style="list-style-type: none"> ▪ Capturar el vocabulario común del negocio. ▪ Identificar reglas del negocio. ▪ Encontrar actores y casos de usos del negocio. ▪ Encontrar trabajadores y entidades del negocio. ▪ Definir los requerimientos a automatizar. ▪ Estructurar el modelo de casos de uso del negocio. 	<ul style="list-style-type: none"> • Glosario negocio. • Reglas del negocio. • Modelo de casos de uso del negocio. <ul style="list-style-type: none"> ○ Actor. ○ Caso de uso. • Modelo de análisis del negocio. <ul style="list-style-type: none"> ○ Realización de casos de uso del negocio. <ul style="list-style-type: none"> ▪ Descripción textual. ▪ Roles participantes. ▪ Diagrama de actividad. ▪ Diagrama de clases. ○ Trabajadores del negocio. ○ Entidades del negocio. ○ Sistemas de negocio. ○ Eventos del negocio.
	Especificador	<ul style="list-style-type: none"> • Detallar un caso de uso del negocio. • Detallar trabajadores y entidades del negocio. 	
	Revisor	<ul style="list-style-type: none"> • Revisar el modelo de casos de uso y análisis. 	

Requerimientos

Analista	<ul style="list-style-type: none">• Desarrollar la visión y capturar el vocabulario común.• Encontrar actores y casos de usos del sistema.• Desarrollar el plan de administración de requerimientos.• Detallar los requerimientos de software.• Producir solicitudes de los Stakeholder.• Estructurar el modelo de casos de uso del sistema.• Diseñar la interfaz de usuario.• Administrar dependencias.	<ul style="list-style-type: none">• Glosario.• Visión.• Requerimientos del software.• Atributos de los requerimientos.• Especificación de los requerimientos del software.• Plan de administración de requerimientos.• Modelo de casos de uso.<ul style="list-style-type: none">○ Paquetes.○ Casos de uso.○ Actores.• Realización de casos de uso.<ul style="list-style-type: none">○ Descripción textual.○ Roles participantes.○ Diagrama de actividad.• Prototipo de interfaz de usuario (diseño).• Solicitudes de los Stakeholder.
Especificador	<ul style="list-style-type: none">• Detallar un caso de uso del sistema.	
Arquitecto de Software	<ul style="list-style-type: none">• Priorizar casos de uso.	
Revisor	<ul style="list-style-type: none">• Revisar los	

		requerimientos.	
Diseño	Diseñador	<ul style="list-style-type: none"> • Diseñar clase. • Diseñar subsistema. • Diseñar caso de uso. • Prototipar la interfaz de usuario. 	<ul style="list-style-type: none"> • Modelo del Diseño. <ul style="list-style-type: none"> ○ Realización de casos de uso. <ul style="list-style-type: none"> ▪ Diagramas interacción. <ul style="list-style-type: none"> • Colaboración. • Secuencia. ▪ Objetos participantes. ○ Clases del diseño. ○ Paquetes del diseño. ○ Subsistemas del diseño.
	Diseñador BD	<ul style="list-style-type: none"> • Diseñar la base de datos. 	
	Arquitecto de Software	<ul style="list-style-type: none"> • Análisis de la arquitectura. • Describir distribución. • Identificar e incorporar los elementos del diseño. 	<ul style="list-style-type: none"> • Modelo de Datos. • Documento de la arquitectura del software. • Modelo de Despliegue. • Prototipo de interfaz de usuario (funcional).
	Revisor	<ul style="list-style-type: none"> • Revisar la arquitectura. • Revisar el diseño. 	
Implementación	Arquitecto de Software	<ul style="list-style-type: none"> • Estructurar el modelo de implementación. • Planear la integración de subsistemas. • Integrar los subsistemas. 	<ul style="list-style-type: none"> • Modelo de Implementación <ul style="list-style-type: none"> ○ Subsistemas de implementación ○ Elementos de implementación • Build.

	Diseñador	<ul style="list-style-type: none"> • Implementar elementos del diseño. 	<ul style="list-style-type: none"> • Plan de Construcción e Integración.
	Revisor	<ul style="list-style-type: none"> • Revisar el código. 	
Prueba	Administrador de Prueba	<ul style="list-style-type: none"> • Estar de acuerdo con la misión. • Identificar los objetivos de prueba. • Determinar resultados de pruebas. • Definir necesidades de rastreabilidad y evaluación. 	<ul style="list-style-type: none"> • Plan de prueba. • Resultados de prueba. • Datos de prueba. • Casos de prueba. • No conformidades.
		<ul style="list-style-type: none"> • Definir el método de prueba. • Definir elementos con posibilidad de evaluación. • Definir detalles de prueba. • Estructurar la implementación de prueba. 	
		<ul style="list-style-type: none"> • Ejecutar prueba. • Elaborar documento de No conformidades. 	
		<ul style="list-style-type: none"> • Ejecutar prueba. • Elaborar documento de No conformidades. 	

Despliegue	Especificador	<ul style="list-style-type: none"> • Desarrollar materiales de soporte. 	<ul style="list-style-type: none"> • Producto. <ul style="list-style-type: none"> ○ Lista de materiales. ○ Artefactos de instalación. ○ Unidad de despliegue.
	Diseñador	<ul style="list-style-type: none"> • Desarrollar artefactos de instalación. 	<ul style="list-style-type: none"> • Material de soporte para usuarios finales. <ul style="list-style-type: none"> ○ Materiales de entrenamiento. • Plan del despliegue.
Administración de la Configuración y Cambios	Administrador de Configuración	<ul style="list-style-type: none"> • Establecer el ambiente de administración de la configuración. • Crear una unidad de despliegue. • Escribir el plan de administración de la configuración. 	<ul style="list-style-type: none"> • Plan de administración de la configuración. • Repositorio del proyecto. • Solicitud de cambios.
	Administrador de Control de Cambios	<ul style="list-style-type: none"> • Revisión de solicitud de cambios. • Establecer el proceso de control de cambios. 	
	Administrador de Prueba	<ul style="list-style-type: none"> ▪ Verificar cambios en un Build. 	

**Administración
del Proyecto**

Administrador
de
Módulo

- Organizar fases e iteraciones.
 - Identificar y evaluar riesgos.
 - Desarrollar el plan de iteraciones.
 - Desarrollar un plan de control de la calidad.
 - Desarrollar un plan de administración de riesgos.
 - Recopilar documentos para conformar el Plan de Desarrollo del Software.
 - Adquirir número de empleados.
 - Definir la organización del proyecto y proveer su personal.
 - Revisar hitos del ciclo de vida.
 - Monitorear estado del proyecto.
 - Capacitar a los miembros del equipo.
- Plan de desarrollo del software.
 - Plan de administración de riesgos.
 - Plan de control de la calidad.
 - Plan de iteraciones.
 - Lista de riesgos.
 - Registro de Revisión.

Ambiente

Ingeniero
de
Procesos

1. Ajustar el proceso de desarrollo al proyecto.
2. Preparar plantillas para el proyecto.
3. Preparar líneas directivas específicas para el proyecto.
4. Implementar caso de desarrollo.
5. Proceso de desarrollo.
 - 5.1. Caso de desarrollo.
 - 5.2. Plantillas específicas del proyecto.
 - 5.3. Líneas directivas específicas del proyecto.

ANEXO 2. CASO DE DESARROLLO DEL MÓDULO BIENES.

**<Módulo Bienes>
Caso de Desarrollo**

Versión <1.0>

Revisión Histórica

Día	Versión	Descripción	Autor
02/12/2006	<1.0>	Confección y descripción del documento.	Elizabeth Martínez

Tabla de Contenidos

1.	INTRODUCCIÓN.....	58
1.1.	Propósito.....	58
1.2.	Definiciones, acrónimos y abreviaturas.....	58
1.3.	Referencias.....	58
1.4.	Visión General.....	58
2.	VISIÓN GENERAL DEL CASO DE DESARROLLO.....	58
2.1.	Modelo del Ciclo de Vida.....	58
2.2.	Disciplinas.....	58
2.3.	Configuración de las disciplinas.....	59
2.4.	Flujo de trabajo.....	59
2.5.	Artefactos.....	59
2.6.	Notas en Artefactos.....	60
2.7.	Informes.....	60
2.8.	Nota en Informes.....	61
2.9.	Revisión de Procedimientos Adicionales.....	61
2.10.	Otros temas.....	61
2.11.	Clasificación de Artefactos.....	61
2.12.	Revisión de Procedimientos.....	61
3.	DISCIPLINAS.....	61
3.1.	Modelación del Negocio.....	62
3.1.1.	Flujo de Trabajo.....	62
3.1.2.	Artefactos.....	62
3.1.3.	Notas en Artefactos.....	63
3.1.4.	Informes.....	63
3.1.5.	Notas en los Informes.....	63
3.2.	Requerimientos.....	63
3.2.1.	Flujo de Trabajo.....	63
3.2.2.	Artefactos.....	63

3.2.3.	Notas en Artefactos.....	65
3.2.4.	Informes.	65
3.3.	Diseño.....	65
3.3.1.	Flujo de Trabajo.	65
3.3.2.	Artefactos.	65
3.3.3.	Notas en Artefactos.....	66
3.3.4.	Informes.	66
3.4.	Implementación.....	67
3.4.1.	Flujo de Trabajo.	67
3.4.2.	Artefactos.	67
3.4.3.	Notas en Artefactos.....	67
3.4.4.	Revisión de procedimientos adicionales.	68
3.5.	Prueba.	68
3.5.1.	Flujo de Trabajo.	68
3.5.2.	Artefactos.	68
3.5.3.	Notas en Artefactos.....	68
3.6.	Despliegue.....	68
3.6.1.	Flujo de Trabajo.	68
3.6.2.	Artefactos.	69
3.6.3.	Notas en Artefactos.....	69
3.7.	Administración de la Configuración y Cambios.....	69
3.7.1.	Flujo de Trabajo.	69
3.7.2.	Artefactos.	70
3.7.3.	Notas en Artefactos.....	70
3.8.	Administración del Proyecto.....	70
3.8.1.	Flujo de Trabajo.	70
3.8.2.	Artefactos.	70
3.8.3.	Notas en Artefactos.....	71
3.9.	Ambiente.....	71
3.9.1.	Flujo de Trabajo.	71

3.9.2. Artefactos	71
3.9.3. Notas en Artefactos.....	72
3.10. Roles.....	72

1. INTRODUCCIÓN.

1.1. Propósito.

El propósito de este documento es describir el proceso de desarrollo para el módulo Bienes del Proyecto RyN.

1.2. Definiciones, acrónimos y abreviaturas.

-

1.3. Referencias.

-

1.4. Visión General.

El resto de este documento describe la forma en que el RUP será configurado para el módulo Bienes, específicamente la sección 2 de este documento contiene una visión del proceso de desarrollo, incluyendo la administración del proyecto y el control de la calidad de las actividades, en la sección 3 se describe por cada disciplina la visión planteada en la sección 2.

2. VISIÓN GENERAL DEL CASO DE DESARROLLO.

2.1. Modelo del Ciclo de Vida.

El ciclo de vida del proyecto abarca las 4 fases descritas por el RUP: inicio, elaboración, construcción y transición y la obtención de cada hito al final de las mismas.

2.2. Disciplinas.

El proceso de desarrollo cubre las siguientes disciplinas: Modelación del Negocio, Requerimientos, Diseño, Implementación, Prueba, Despliegue, Administración de la Configuración y Cambios, Administración del Proyecto y Ambiente.

2.3. Configuración de las disciplinas.

El propósito de esta sección es explicar como trabaja la configuración de la disciplina. Esto incluye la explicación del propósito para diferentes tablas y secciones que describen cada disciplina listada en la sección titulada Disciplina.

2.4. Flujo de trabajo.

Esta sección detalla cualquier cambio hecho a la estructura de la disciplina. El cambio típico incluye la adición de actividades para describir vías de trabajo específicas de la compañía o la eliminación de actividades de la disciplina.

2.5. Artefactos.

Esta sección describe en una tabla como el artefacto será utilizado, los artefactos adicionales pueden ser añadidos a esta tabla según las necesidades.

Artefactos	Como usarlos				Detalles de Revisión	Herramientas utilizadas
	Inic	Elab	Const	Trans		

Explicación de la Tabla		
Nombre Columna	Propósito	Contenidos / Comentarios
'Artefactos'	[Nombre del artefacto]	[Referencia al artefacto en el RUP o a una definición de artefacto local como parte del caso de desarrollo]
'Como usarlos'	[Calificar como el artefacto es utilizado a lo largo del ciclo de vida]	Por cada fase, decidir la clasificación de sus artefactos en: <ul style="list-style-type: none"> • Debe tener. • Debería tener. • Pudiera tener.

		<ul style="list-style-type: none"> No debe tener. <p>Estos están definidos en: Guidelines: Classifying Artifacts del RUP.</p>
' Detalles de Revisión '	[Definir el nivel y los procedimientos de revisión aplicados a los artefactos]	<p>Decidir que nivel de revisión aplicar:</p> <ul style="list-style-type: none"> Formal-Externo. Formal-Interno. Informal. Ninguno. <p>Los detalles en: Guidelines: Review Levels del RUP.</p>
'Herramientas a utilizar'	[Definir la herramienta o herramientas usadas para producir el artefacto]	Referencia los detalles de las herramientas usadas para desarrollar y mantener este artefacto.

2.6. Notas en Artefactos.

Contiene una lista de todos los artefactos que no se utilizan y los motivos.

2.7. Informes.

Esta sección lista los informes a ser usados e informes adicionales que pueden ser añadidos a la tabla según la necesidad.

Informes	Como usarlo	Herramienta usada.

2.8. Nota en Informes.

Esta sección tiene 2 propósitos principales. Listar todos los informes que el proyecto decidió no utilizar y los motivos.

2.9. Revisión de Procedimientos Adicionales.

Esta sección captura cualquier procedimiento adicional de revisión requerido para los artefactos usados en esta disciplina.

2.10. Otros temas.

Esta sección captura cualquier tema sobresaliente con la configuración de la disciplina.

2.11. Clasificación de Artefactos.

Un artefacto es un entregable del proceso. El es con frecuencia desarrollado dentro de una disciplina aunque hay excepciones. Para describir como un artefacto será usado, considere la siguiente clasificación y vea [Guidelines: Classifying Artifacts](#) por detalles.

- Debe.
- Debería.
- Pudiera.
- No debe.

2.12. Revisión de Procedimientos.

El proyecto usa los siguientes niveles de revisión:

- Formal-Externo.
- Formal-Interno.
- Informal.
- Ninguno.

Por detalles ver: [Guidelines: Review Levels](#).

3. DISCIPLINAS.

3.1. Modelación del Negocio.

3.1.1. Flujo de Trabajo.

Se siguió solo el detalle del Modelado del Negocio. Ver: [Business Modeling: Workflow](#) del RUP.

3.1.2. Artefactos.

Artefactos	Como usarlos				Detalles de Revisión	Herramientas utilizadas
	Inic	Elab	Const	Trans		
Actor Negocio	Debe	Pudiera	Pudiera	Pudiera	Formal-Externa	Rational Rose
Entidad del Negocio	Debe	Pudiera	Pudiera	Pudiera	Formal-Externa	Rational Rose
Evento del Negocio	Debe	Pudiera	Pudiera	Pudiera	Formal-Externa	Rational Rose
Glosario del Negocio	Debe	Pudiera	Pudiera	Pudiera	Formal-Externa	Microsoft Word
Modelo de Análisis del Negocio	Debe	Pudiera	Pudiera	Pudiera	Formal-Externa	Rational Rose
Reglas del Negocio	Debe	Pudiera	Pudiera	Pudiera	Formal-Externa	Rational Rose
Caso de uso del Negocio	Debe	Pudiera	Pudiera	Pudiera	Formal-Externa	Rational Rose
Modelo de casos de uso del Negocio	Debe	Pudiera	Pudiera	Pudiera	Formal-Externa	Rational Rose
Realización de los casos de uso del Negocio	Debe	Pudiera	Pudiera	Pudiera	Formal-Externa	Microsoft Word Rational Rose

Trabajador del Negocio	Debe	Pudiera	Pudiera	Pudiera	Formal-Externa	Rational Rose
Sistema del Negocio	Debe	Pudiera	Pudiera	Pudiera	Formal-Externa	Rational Rose

3.1.3. Notas en Artefactos.

El proyecto decidió hacer solo el Modelado del Negocio, lo que significa que hay una serie de artefactos que no fueron desarrollados como: Documento de la Arquitectura del Negocio, Meta del Negocio, Visión del Negocio, Especificaciones Suplementarias del Negocio y Evaluación del Objetivo de la Organización.

3.1.4. Informes.

Informes	Como usarlo	Herramienta usada.
Caso de uso del Negocio	Pudiera	Microsoft Word
Revisión del Modelo de caso de usos del Negocio	Pudiera	Microsoft Word
Realización del caso de uso del Negocio	Pudiera	Microsoft Word
Revisión de la regla del negocio	Pudiera	Microsoft Word

3.1.5. Notas en los Informes.

Se decidieron utilizar solo algunos informes de acuerdo a los entregables de esta disciplina con vistas a agilizar el proceso.

3.2. Requerimientos.

3.2.1. Flujo de Trabajo.

No hubo cambios. Ver: [Requirements: Workflow](#) del RUP.

3.2.2. Artefactos.

Artefactos	Como usarlos	Detalles	Herramientas
------------	--------------	----------	--------------

	Inic	Elab	Const	Trans	de Revisión	utilizadas
Actor	Debe	Debe	Debe	Debe	Formal- Interna	Rational Rose
Glosario	Debe	Debe	Debe	Debe	Formal- Externa	Microsoft Word
Atributos de los Requerimientos	Debe	Debe	Debe	Debe	Formal- Interna	Rational Rose
Plan de Administración de Requerimientos	Debe	Debe	Debe	Debe	Formal- Interna	Microsoft Word
Solicitud de los Stakeholder	Debe	Debe	Debe	Debe	Formal- Externa	Microsoft Word
Especificación de los requerimientos del Software	Debe	Debe	Debe	Debe	Formal- Externa	Microsoft Word
Caso de uso	Debe	Debe	Debe	Debe	Formal- Externa	Rational Rose
Modelo de casos de uso	Debe	Debe	Debe	Debe	Formal- Externa	Rational Rose
Paquetes de caso de uso	Debe	Debe	Debe	Debe	Informal	Rational Rose
Prototipo de interfaz de usuario (diseño)	No debe	Debe	Pudiera	Pudiera	Formal- Externa	Microsoft Visio
Visión	Debe	Debe	Debe	Debe	Formal- Externa	Microsoft Word
Realización de casos de uso	Debe	Debe	Debe	Debe	Formal- Interna	Rational Rose
Requerimientos del	Debe	Debe	Debe	Debe	Formal-	Microsoft Word

software					Externa	
----------	--	--	--	--	---------	--

3.2.3. Notas en Artefactos.

Se decidieron utilizar solo algunos artefactos con vistas a agilizar el proceso. No se utilizaron aquellos como: Storyboard y Especificaciones suplementarias, etc.

3.2.4. Informes.

Informes	Como usarlo	Herramienta usada.
Actor	Pudiera	Microsoft Word
Caso de uso	Pudiera	Microsoft Word
Revisión del Modelo de caso de uso	Pudiera	Microsoft Word
Realización del caso de uso	Pudiera	Microsoft Word

3.3. Diseño.

3.3.1. Flujo de Trabajo.

En esta disciplina se elimina el detalle de flujo de trabajo Ejecutar Síntesis de Arquitectura pues en la fase de inicio no se decide analizar la arquitectura, además se eliminan las actividades que tienen que ver con el análisis en esos detalles de flujos de trabajo, además como no se desarrolla una aplicación en tiempo real los roles de diseñador de capsula y su actividad son excluidos.

Ver Design: Workflow del RUP para más detalles.

3.3.2. Artefactos.

Artefactos	Como usarlos				Detalles de Revisión	Herramientas utilizadas
	Inic	Elab	Const	Trans		
Modelo de datos	No debe	Debe	Debe	Pudiera	Informal	Erwin
Modelo de despliegue	No debe	Debe	Debe	Debe	Formal-	Rational Rose

					Interno	
Clase del diseño	No debe	Debe	Debe	Debe	Informal	Rational Rose
Modelo del diseño	No debe	Debe	Debe	Debe	Formal-Interno	Rational Rose
Paquete del diseño	No debe	Debe	Debe	Debe	Formal-Interno	Rational Rose
Subsistema de diseño	No debe	Debe	Debe	Debe	Formal-Interno	Rational Rose
Realización del caso de uso	No debe	Debe	Debe	Debe	Formal-Interno	Rational Rose
Documento de la Arquitectura de Software	Pudiera	Debe	Debe	Debe	Formal-Externo	Microsoft Word
Prototipo de interfaz de usuario (funcional)	No debe	Debe	Debe	Debe	Formal-Interno	C#

3.3.3. Notas en Artefactos.

El proyecto no esta desarrollando un producto en tiempo real, esto significa que los siguientes artefactos no fueron desarrollados: capsula, protocolo, evento y señal. Además decidió no mantener un modelo de análisis, por lo que los artefactos clases y modelo del análisis no fueron desarrollados. Otros no fueron desarrollados con vistas a agilizar el proceso.

3.3.4. Informes.

Informes	Como usarlo	Herramienta usada.
Clase	Pudiera	Microsoft Word
Modelo del diseño	Pudiera	Microsoft Word
Paquete / Subsistema del diseño	Pudiera	Microsoft Word

Realización de caso de uso	Pudiera	Microsoft Word
----------------------------	---------	----------------

3.4. Implementación.

3.4.1. Flujo de Trabajo.

En esta disciplina se elimina el detalle de flujo de trabajo Integrar el Sistema pues como se esta trabajando a nivel de módulo lo que se hará es integrar los subsistemas.

Ver Implementation: Workflow del RUP para más detalles.

3.4.2. Artefactos.

Artefactos	Como usarlos				Detalles de Revisión	Herramientas utilizadas
	Inc	Elab	Const	Trans		
Build	No debe	Debe	Debe	Debe	Informal	C#
Elemento de implementación	No debe	Debe	Debe	Debe	Informal Revisión del Código	C#
Modelo de implementación	No debe	Debe	Debe	Debe	Informal	Rational Rose
Subsistema de implementación	No debe	Debe	Debe	Debe	Formal-Interno	C#
Plan de construcción e integración	No debe	Debe	Debe	Debe	Formal-Interno	Microsoft Word

3.4.3. Notas en Artefactos.

Se decidieron utilizar solo algunos artefactos con vistas a agilizar el proceso. No se utilizaron aquellos como: Stub de prueba, etc.

3.4.4. Revisión de procedimientos adicionales.

Son realizadas las revisiones de código informales.

3.5. Prueba.

3.5.1. Flujo de Trabajo.

No hubo cambios es los flujos de trabajo de esta disciplina. Ver Test: Workflow del RUP para más detalles.

3.5.2. Artefactos.

Artefactos	Como usarlos				Detalles de Revisión	Herramientas utilizadas
	Inic	Elab	Const	Trans		
Caso de prueba	No debe	No debe	Debe	Debe	Informal	Microsoft Word
Datos de prueba	No debe	No debe	Debe	Debe	Informal	Rational Rose
Plan de prueba	Pudiera	Pudiera	Debe	Debe	Informal	Microsoft Word
Resultados de prueba	No debe	No debe	Debe	Debe	Informal	Microsoft Word
No conformidades	No debe	No debe	Debe	Debe	Informal	Microsoft Word

3.5.3. Notas en Artefactos.

Se decidieron utilizar solo algunos artefactos con vistas a agilizar el proceso. No se utilizaron aquellos como: Resumen de Evaluación de prueba, Script de prueba, Log de pruebas, Lista de ideas de prueba, Estrategia de prueba, etc.

3.6. Despliegue

3.6.1. Flujo de Trabajo.

Hubo cambios en esta disciplina, pues al ser el administrador despliegue definido en la organización no se desarrollaron sus actividades en esta configuración. Ver Deployment: Workflow del RUP para más detalles.

3.6.2. Artefactos.

Artefactos	Como usarlos				Detalles de Revisión	Herramientas utilizadas
	Inic	Elab	Const	Trans		
Lista de materiales	No debe	Pudiera	Pudiera	Debe	Formal-Interno	Microsoft Word
Plan de despliegue	No debe	Pudiera	Pudiera	Debe	Informal	Rational Rose
Unidad de despliegue	No debe	Pudiera	Pudiera	Debe	Informal	Microsoft Word
Material de soporte para usuarios finales	No debe	Pudiera	Pudiera	Debe	Informal	Microsoft Word
Artefactos de instalación	No debe	Pudiera	Pudiera	Debe	Informal	Microsoft Word
Producto	No debe	Pudiera	Pudiera	Debe	Formal-Externo	
Materiales de entrenamiento	No debe	Pudiera	Pudiera	Debe	Informal	Microsoft Word

3.6.3. Notas en Artefactos.

Se decidieron utilizar solo algunos artefactos con vistas a agilizar el proceso. No se utilizaron aquellos como: Release de Notas, Trabajo de arte para el producto.

3.7. Administración de la Configuración y Cambios.

3.7.1. Flujo de Trabajo.

En esta disciplina no hubo cambios en los flujos de trabajo. Ver [Configuration & Change Management: Workflow](#) del RUP para más detalles.

3.7.2. Artefactos.

Artefactos	Como usarlos				Detalles de Revisión	Herramientas utilizadas
	Inic	Elab	Const	Trans		
Solicitud de cambios	Pudiera	Pudiera	Debe	Debe	Informal	Microsoft Word
Plan de administración de la configuración	Debe	Debe	Debe	Debe	Informal	Microsoft Word
Repositorio del proyecto	Debe	Debe	Debe	Debe	Ninguno	Trac 0.9.5

3.7.3. Notas en Artefactos.

Se decidieron utilizar solo algunos artefactos con vistas a agilizar el proceso. No se utilizaron aquellos como: Resultados de Auditorias de Configuración, Workspace etc.

3.8. Administración del Proyecto.

3.8.1. Flujo de Trabajo.

No hubo cambios en esta disciplina. Ver [Project Management: Workflow](#) del RUP para más detalles.

3.8.2. Artefactos.

Artefactos	Como usarlos				Detalles de Revisión	Herramientas utilizadas
	Inic	Elab	Const	Trans		
Plan de iteración	Debe	Debe	Debe	Debe	Informal	Microsoft Word
Plan de control de calidad	Debe	Debe	Debe	Debe	Informal	Microsoft Word
Registro de revisiones	Debe	Debe	Debe	Debe	Informal	Microsoft Word

Lista de riesgos	Debe	Debe	Debe	Debe	Formal-Interno	Microsoft Word
Plan de administración de riesgos	Debe	Debe	Debe	Debe	Formal-Interno	Microsoft Word
Plan de desarrollo del software	Debe	Debe	Debe	Debe	Formal-Interno	Microsoft Word Microsoft Visio

3.8.3. Notas en Artefactos.

Se decidieron utilizar solo algunos artefactos con vistas a agilizar el proceso. No se utilizaron aquellos como: Plan de mediciones, Plan de aceptación del producto, Plan de resolución de problemas, etc.

3.9. Ambiente.

3.9.1. Flujo de Trabajo.

Hubo cambios en esta disciplina, pues al ser el especialista en herramientas definido en la organización no se desarrollaron sus actividades en esta configuración, tal es el caso del detalle de flujo de trabajo: Soportar el ambiente durante una iteración. Ver [Environment: Workflow](#) del RUP para más detalles.

3.9.2. Artefactos.

Artefactos	Como usarlos				Detalles de Revisión	Herramientas utilizadas
	Inic	Elab	Const	Trans		
Caso de desarrollo	Debe	Debe	Debe	Debe	Informal	Microsoft Word
Plantillas específicas del proyecto	Debe	Debe	Debe	Debe	Informal	Microsoft Word
Líneas directivas de pruebas	No debe	Debe	Debe	Debe	Informal	Microsoft Word
Líneas directivas de diseño	No debe	Debe	Debe	Debe	Informal	Microsoft Word

Líneas directivas de modelación del negocio	Debe	Pudiera	Pudiera	Pudiera	Informal	Microsoft Word
Líneas directivas de programación	No debe	Debe	Debe	Debe	Informal	Microsoft Word
Líneas directivas de herramientas	Debe	Debe	Debe	Debe	Informal	Microsoft Word
Líneas directivas de modelación de caso de uso	Debe	Debe	Debe	Debe	Informal	Microsoft Word
Líneas directivas de interfaz de usuario	No debe	Debe	Debe	Debe	Informal	Microsoft Word

3.9.3. Notas en Artefactos.

Se decidieron utilizar solo algunos artefactos con vistas a agilizar el proceso. No se utilizó el Manual de Estilo de Guía. Además no se seleccionaron las herramientas ni la infraestructura de desarrollo.

3.10. Roles.

ROLES RUP	ROLES PROPUESTOS
Analista de proceso de negocio.	Analista
Analista del sistema.	
Diseñador de interfaz de usuario.	
Diseñador del negocio.	Especificador
Escritor Técnico.	
Especificador de requerimientos.	
Arquitecto de software.	Arquitecto de software
Integrador.	

Diseñador.	Diseñador
Programador.	
Diseñador de Base Datos.	Diseñador de Base Datos
Administrador de prueba.	Administrador de Prueba
Analista de prueba.	
Diseñador de prueba.	
Probador.	Probador
Administrador del Proyecto.	Administrador de Módulo
Revisor de la Administración.	
Administrador del control de Cambios.	Administrador de control de Cambios
Administrador de la Configuración.	Administrador de la Configuración
Ingeniero de procesos.	Ingeniero de Procesos
Revisor Técnico.	Revisor
Administrador de despliegue.	Administrador de Despliegue
Desarrollador de cursos.	
Especialista en Herramientas.	Especialista en Herramientas
Administrador del sistema	
Stakeholder.	Stakeholder

Artista grafico.	-
Revisor.	-
Coordinador revisor.	-
Diseñador de capsula.	-
Cualquier rol.	-