

**DISEÑO E IMPLEMENTACIÓN DE UNA ESTRATEGIA METODOLÓGICA PARA LA APLICACIÓN DEL PROCESO UNIFICADO PARA EL DESARROLLO DE SOFTWARE EN LA EMPRESA “TECNOLOGÍAS DE INFORMACIÓN Y TELECOMUNICACIONES TYT LTDA.”**

**Carlos Andrés Rueda Ribero**

**Universidad Industrial de Santander**  
**Facultad de Ingenierías Físico-mecánicas**  
**Escuela de Ingeniería de Sistemas e Informática**  
**Bucaramanga**  
**2007**

**DISEÑO E IMPLEMENTACIÓN DE UNA ESTRATEGIA METODOLÓGICA PARA LA APLICACIÓN DEL PROCESO UNIFICADO PARA EL DESARROLLO DE SOFTWARE EN LA EMPRESA “TECNOLOGÍAS DE INFORMACIÓN Y TELECOMUNICACIONES TYT LTDA.”**

**Carlos Andrés Rueda Ribero**

**Código: 2012684**

**Fernando Antonio Rojas Morales**

**Director Práctica Empresarial**

**Docente UIS**

**Humberto Rueda Rivero**

**Tutor Práctica Empresarial**

**Director de Proyectos TyT Ltda.**

**Universidad Industrial de Santander**

**Facultad de Ingenierías Físico-mecánicas**

**Escuela de Ingeniería de Sistemas e Informática**

**Bucaramanga**

**2007**

**2**

Titulo: DISEÑO E IMPLEMENTACIÓN DE UNA ESTRATEGIA METODOLÓGICA PARA LA APLICACIÓN DEL PROCESO UNIFICADO PARA EL DESARROLLO DE SOFTWARE EN LA EMPRESA: TECNOLOGÍAS DE INFORMACIÓN Y TELECOMUNICACIONES TYT LTDA.\*

Autor: Carlos Andrés Rueda Ribero\*\*

Palabras Clave: RUP, metodología, implementación, proceso de software, iteración.

## RESUMEN

Esta práctica empresarial se dividió en cuatro fases principales, cada una de ellas con unos objetivos muy específicos los cuáles se cumplieron en su totalidad, a continuación se describen brevemente.

Análisis del proceso de desarrollo de software utilizado por la empresa que buscaba la identificación de los elementos que influían de manera negativa en el proceso, para enfocar el trabajo en torno a las actividades necesarias para lograr su mejoramiento.

Consulta acerca de la metodología RUP y su proceso de implementación; su objetivo era profundizar en los conocimientos acerca de los factores fundamentales del RUP y crear una base para la elaboración de un plan de capacitación y un material de trabajo que sirvieran como ejes directores del trabajo en la práctica empresarial.

Capacitación sobre los fundamentos del RUP y su proceso de implementación; se llevó a cabo un proceso de transferencia y construcción de conocimientos que buscaba establecer un vocabulario común, despejar las dudas acerca de la metodología y medir el grado de aceptación de la misma por parte del equipo humano de la empresa; esta fase representó un importante aporte para el futuro de la empresa.

Talleres prácticos que permitieron la aplicación de los conocimientos aprendidos durante la fase de capacitación en un proyecto real.

\* Trabajo de grado.

\*\* Facultad de Ingenierías Físico-Mecánicas  
Escuela de Ingeniería de Sistemas e Informática  
Tutor: Ing. Humberto Rueda Rivero  
Director: Ing. Fernando Antonio Rojas Morales

Title: DESIGN AND IMPLEMENTATION OF A METHODOLOGIC STRATEGY FOR THE APPLICATION OF THE UNIFIED PROCESS FOR SOFTWARE DEVELOPMENT IN THE COMPANY: "TECNOLOGÍAS DE INFORMACIÓN Y TELECOMUNICACIONES TYT LTDA".\*

Author: Carlos Andrés Rueda Ribero\*\*

Key Words: RUP, methodology, implementation, software process, iteration.

### ABSTRACT

This enterprise practice was divided in four main phases, each one of them with very specific objectives which were fulfilled in their totality, next they are briefly described.

Analysis of the software development process used by the company that looked for the identification of the elements that influenced the process in a negative way, to focus the work around the necessary activities to achieve their improvement.

Consultation about the RUP methodology and its implementation process; its aim was to conceptualize the fundamental factors of the RUP and to create a base for the elaboration of a training plan and a work material that be used like a principal tool to direct the work during the enterprise practice.

Training on RUP fundamentals and their implementation process; this activity involved a process of transfer and construction of knowledge that was seeking to establish a common vocabulary, to clear the doubts about the methodology and to measure the grade of its acceptance on the part of the human team of the company; this phase represent an important contribution for the future of the company.

Practical exercises that allowed the application of the knowledge learned during the training phase to a real project.

\* Graduation work.

\*\*Faculty of Physical-Mechanical Engineering  
School of Engineering of Systems and Computer science  
Tutor: Ing. Humberto Rueda Rivero  
Director: Ing. Fernando Antonio Rojas Morales

## DEDICATORIA

A mi hijo Nicolás por enseñarme día a día el significado del verdadero amor.

A mis padres Humberto y Carmenza por su infinito amor, comprensión, paciencia y por enseñarme el significado de la esperanza.

A mis amigos Sonia, Eliana y Salah quienes son parte fundamental de mi vida y por enseñarme el significado de la amistad.

A Andrea por su amor, cariño, ternura y por darle a mi vida un nuevo motivo para seguir adelante.

## **AGRADECIMIENTOS**

A todos los miembros de TyT Ltda por su confianza y su participación en el desarrollo de esta práctica empresarial.

A toda mi familia la cuál creyó en mí y me apoyo en los momentos más difíciles de mi vida, a mi tía Sor Lilian Rivero y muy especialmente a mi tío Luis María.

Al profesor Fabio Reyes Cordero porque más que un profesor es un verdadero maestro.

A mi hermano Humberto por su entrega, profesionalismo, compromiso y dedicación en el desarrollo y tutoría de esta práctica empresarial.

Al profesor Fernando Antonio Rojas Morales quien dirigió esta práctica empresarial con mucho interés y siempre con el deseo de aportar y de construir.

A todos mis compañeros de carrera, en especial Kata, Joan, Melissa, Eduardo, Diana, Diego y muchos mas, gracias.

## Tabla de contenido

<b>INTRODUCCION.....</b>	<b>10</b>
<b>1. IDENTIFICACIÓN DEL PROBLEMA .....</b>	<b>10</b>
<b>2. OBJETIVOS .....</b>	<b>11</b>
<b>2.1 OBJETIVOS ESPECÍFICOS.....</b>	<b>11</b>
<b>3. JUSTIFICACION.....</b>	<b>12</b>
<b>4. MARCO TEORICO .....</b>	<b>14</b>
<b>4.1 HISTORIA DEL PROCESO UNIFICADO.....</b>	<b>14</b>
4.1.1 EL METODO DE ERICSSON .....	14
4.1.2 OBJECTORY .....	15
<b>4.2 PROCESO UNIFICADO PARA EL DESARROLLO DE SOFTWARE (RUP) .....</b>	<b>16</b>
4.2.1 UN PROCESO DIRIGIDO POR LOS CASOS DE USO.....	17
4.2.2 UN PROCESO CENTRADO EN LA ARQUITECTURA.....	19
4.2.3 UN PROCESO ITERATIVO E INCREMENTAL.....	22
4.2.4 FASES DEL PROCESO .....	23
4.2.5 DISCIPLINAS O FLUJOS DE TRABAJO. ....	26
4.2.6 ROLES .....	30
<b>4.3 MEJORES PRÁCTICAS EN INGENIERÍA DEL SOFTWARE.....</b>	<b>31</b>
4.3.1 DESARROLLO ITERATIVO. ....	31
4.3.2 GESTIÓN DE REQUERIMIENTOS.....	33
4.3.3 DESARROLLO BASADO EN COMPONENTES. ....	34
4.3.4 MODELADO VISUAL (UML).....	36
4.3.5 VERIFICACIÓN CONTINÚA DE CALIDAD. ....	37
<b>5. METODOLOGIA EMPLEADA.....</b>	<b>39</b>
5.1 ANÁLISIS DEL ENTORNO. ....	39
5.2 CONSULTA DE INFORMACIÓN ACERCA DE LA METODOLOGÍA RUP.....	39
5.3 CAPACITACIÓN SOBRE LOS FUNDAMENTOS DEL RUP. ....	40
5.4 SOCIALIZACIÓN Y TALLERES. ....	40
<b>6. EXPERIENCIA DE LA PRÁCTICA EMPRESARIAL. ....</b>	<b>41</b>
6.1 PROCESO DE DESARROLLO DE SOFTWARE DE TYT LTDA. ....	43
6.2 CAPACITACIÓN SOBRE LOS FUNDAMENTOS DEL RUP. ....	47
<b>7. ESTRATEGIA DE IMPLEMENTACIÓN DE RUP EN LA EMPRESA TYT LTDA. ....</b>	<b>59</b>
<b>8. CONCLUSIONES .....</b>	<b>60</b>
<b>9. RECOMENDACIONES.....</b>	<b>62</b>
<b>10. BIBLIOGRAFIA .....</b>	<b>63</b>
<b>11. ANEXOS .....</b>	<b>64</b>

## Tabla de Ilustraciones

Ilustración 1: Historia del RUP.....	14
Ilustración 2: Bases del RUP.....	16
Ilustración 3: Ejemplo del Diagrama de Casos de Uso.....	19
Ilustración 4: Ejemplo de un Diagrama de Clases. ....	20
Ilustración 5: Vista general del modelo de desarrollo RUP.....	23
Ilustración 6: Ejemplo de los Roles en el RUP.....	30
Ilustración 7: Modelo de Desarrollo iterativo .....	31
Ilustración 8: Ejemplo de un diagrama de componentes. ....	35
Ilustración 9: Diagrama de Casos de Uso módulo de Facturación, Aquasoft.....	57

# INTRODUCCION

## 1. IDENTIFICACIÓN DEL PROBLEMA

Debido a la creciente complejidad de los sistemas de información y concientes de la necesidad, cada vez más crítica, de optimizar su proceso de desarrollo de software a fin de cumplir de una manera eficaz a las exigencias de calidad que imponen los clientes actualmente, la empresa Tecnologías de Información y Telecomunicaciones<sup>1</sup> ha decidido incursionar en el uso de esta metodología (RUP) que busca brindarle la dirección administrativa necesaria para un mejor aprovechamiento de su talento humano, a una reducción en los costos y tiempos de entrega de sus productos y que propenda por alcanzar un nivel elevado de calidad.

Teniendo en cuenta que el proceso unificado para el desarrollo de Software de Rational RUP<sup>2</sup> es *“una innovación basada en el conocimiento”*,<sup>3</sup> TyT Ltda ha venido adelantando un proceso de auto capacitación en estas metodologías, pero requiere apoyo para la adopción de las mismas dentro de su proceso de desarrollo de software.

---

<sup>1</sup> En adelante TyT Ltda.

<sup>2</sup> Sigla de Rational Unified Process, en adelante se establece la equivalencia con el título de “Proceso Unificado para el Desarrollo de Software”.

<sup>3</sup> BOOCH, Grady - RUMBAUGH, James - JACOBSON, Ivar. El Proceso Unificado de Desarrollo de Software. Madrid: Addison-Wesley, 2000

## **2. OBJETIVOS**

Diseñar e implementar una propuesta de estrategia metodológica que permita a la empresa crecer en el conocimiento y aplicación del proceso unificado para el desarrollo de software RUP, teniendo como base el lenguaje unificado de modelado UML.

### **2.1 OBJETIVOS ESPECÍFICOS**

2.1.1 Especificar y guiar, en forma incremental, el uso de los diagramas y modelos de UML en los proyectos que actualmente desarrolla TyT Ltda.

2.1.2 Definir Roles, actividades y artefactos para cada etapa de los proyectos informáticos de TyT, siguiendo la propuesta del RUP.

2.1.3 Ilustrar la forma de implementación de RUP-UML en un proceso de desarrollo de software por medio de un proceso de ingeniería inversa<sup>4</sup> aplicado a uno de los proyectos que actualmente desarrolla la empresa.

2.1.4 Transferir el conocimiento necesario para que TyT adopte UML en todas las fases existentes en su proceso de desarrollo de software.

---

<sup>4</sup> En el contexto del desarrollo de software, transformación del código en un modelo a través de su traducción a un determinado lenguaje de implementación

### 3. JUSTIFICACION

*“TyT Ltda es una compañía de base tecnológica que nace del espíritu emprendedor de un Grupo interdisciplinario de personas con amplios conocimientos y experiencia en sectores y actividades como las tecnologías de información, la gestión tecnológica y la consultoría estratégica empresarial que durante años aportaron , agregaron y lo siguen haciendo, valor a las empresas en donde prestaron y prestan sus servicios, adquiriendo experiencia y las competencias necesarias para transformar deseos, visiones y proyectos individuales en un pensamiento corporativo basado en la dimensionalidad de tres pilares fundamentales: Talento Humano (socios , empleados, aliados), Clientes y la Sociedad, permitiendo estructurar productos y servicios que generen valor a sus clientes, prácticas que generen riqueza a nuestro Talento Humano y principios que contribuyan a construir una sociedad más abierta y con mejor calidad de vida.”<sup>5</sup>*

Los cambios en el entorno social y económico del mundo en los últimos años han estimulado una creciente ola de innovación que impulsa a las compañías a ser competitivas para subsistir; la información se ha convertido en el activo más valioso para las organizaciones y las necesidades de esta índole han crecido a un ritmo acelerado. Como consecuencia, la complejidad de los sistemas de información es cada día mayor y por ello se hace necesario para los profesionales en Ingeniería de Sistemas contar con herramientas metodológicas que permitan que el desarrollo de un sistema de información se convierta en un proceso formal, definido e implementado que les permita cumplir de forma efectiva, predecible

---

<sup>5</sup> Tomado del documento de visión de la empresa Tecnologías de Información y Telecomunicaciones TyT Ltda.

y con un alto nivel de rendimiento a las exigencias de calidad que actualmente imponen los clientes.

Ante este panorama se plantean soluciones que permitan a TyT Ltda afrontar con éxito el reto que impone el ejercicio de su actividad haciéndose manifiesta la necesidad de implementar la metodología propuesta por el Proceso Unificado para el Desarrollo de Software (RUP), teniendo como base al Lenguaje Unificado de Modelado (UML), con el fin de obtener un manejo integro, controlado y repetible de sus procesos internos que permitan un mayor aprovechamiento de su talento humano que se vea reflejado en una optimización de los recursos financieros, en una mejora considerable en los tiempos de entrega de sus productos y en un alto grado de calidad que asegure su sostenibilidad en el mercado altamente competitivo de hoy.

## 4. MARCO TEORICO

### 4.1 HISTORIA DEL PROCESO UNIFICADO

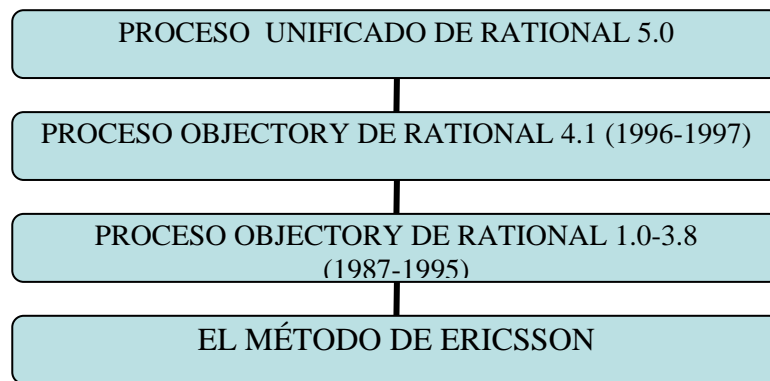


Ilustración 1: Historia del RUP.

#### 4.1.1 EL METODO DE ERICSSON

Ericsson modelaba el sistema entero como un conjunto de bloques interconectados (en UML se los conoce como *subsistemas* y son implementados mediante *componentes*). Después ensamblaba los bloques de más bajo nivel en bloques de más alto nivel con el fin de lograr un mayor control sobre el sistema. Los bloques eran identificados estudiando los casos del negocio, hoy conocidos como *casos de uso*, previamente identificados y especificados. Para cada caso de negocio se identificaban los bloques que debían cooperar para realizarlo; sus actividades de diseño producían un conjunto de diagramas de bloques estáticos

con sus interfaces, agrupados en subsistemas. Estos diagramas fueron la versión preliminar de los diagramas de clases de UML.

Como resultado del trabajo en las actividades de diseño se tenía una “*Descripción de la Arquitectura Software*”. Estaba basada en la comprensión de los requisitos más críticos, y de manera breve describía al bloque y su agrupamiento en subsistemas. Todos los mensajes que se transmitían entre los diferentes objetos cabe resaltar que se almacenaban en una biblioteca de mensajes, que, junto con la descripción de la arquitectura eran los documentos fundamentales que guiaban el trabajo de desarrollo y además eran utilizados para presentar el sistema a los clientes, los cuáles –para esa época (1968)- no estaban muy acostumbrados a este tipo de presentaciones para un proyecto de software.

En esencia era un desarrollo basado en componentes.

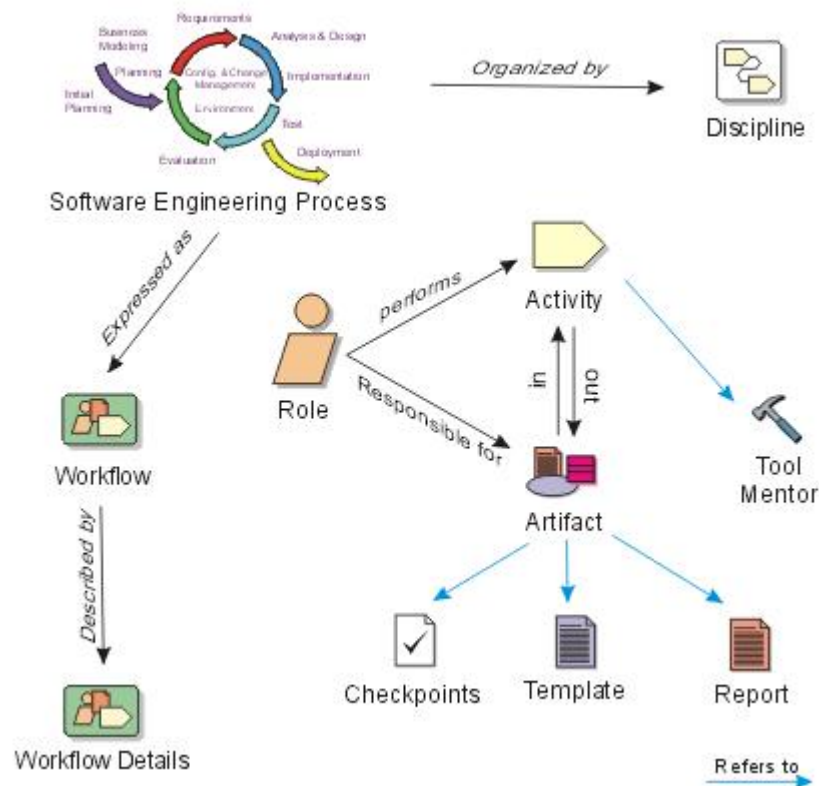
#### 4.1.2 OBJECTORY

Es la abreviatura de **Object Factory** o fábrica de objetos. Fundada en 1987 por Ivar Jacobson.

La idea del caso de uso, la cuál fue tratada en el método de Ericsson fue materializada y bautizada además en la conferencia OOPSLA de 1987. Los diferentes flujos de trabajos se representaron en una serie de modelos: requisitos-casos de uso, análisis, diseño, implementación y pruebas. La trazabilidad se convirtió en prerrequisito del desarrollo dirigido por casos de uso, de esta forma los desarrolladores podían seguir la traza de un caso de uso a través de la secuencia de modelos hasta el código o bien retroceder en caso de presentarse algún inconveniente.

El desarrollo del proceso de **Objectory** continuó en una serie de versiones, desde **Objectory 1.0** en 1988 hasta la versión 3.8 de 1995.

## 4.2 PROCESO UNIFICADO PARA EL DESARROLLO DE SOFTWARE (RUP)<sup>6</sup>



**Ilustración 2: Bases del RUP**

El RUP es un modelo de un proceso para el desarrollo de software que centra su atención en la adaptación de la empresa a una nueva forma de llevar a cabo sus

<sup>6</sup> Adaptado de BOOCH, Grady - RUMBAUGH, James - JACOBSON, Ivar. El Proceso Unificado de Desarrollo de Software. Madrid: Addison-Wesley, 2000

actividades, es muy interesante el hecho de no ser simplemente un modelo de proceso de desarrollo de software sino el ser un marco de trabajo genérico fácilmente adaptable a diversos tipos de organizaciones, equipos, áreas de trabajo y diferentes tamaños de proyectos informáticos.

Como parte fundamental de este nuevo enfoque se deben tener en cuenta los siguientes aspectos definitorios, son ellos:

#### 4.2.1 Un proceso Dirigido por los casos de uso.

Un caso de uso es un conjunto de secuencias de acciones, incluidas las variaciones, que un sistema realiza y que tienen como consecuencia un resultado de valor para el usuario. Se busca responder a la pregunta: "*¿Qué debe hacer el sistema para cada usuario?*". Esto indica que los casos de uso se utilizan como *artefacto*<sup>7</sup> principal para definir el comportamiento deseado para el sistema y para comunicar este comportamiento entre las personas involucradas en el sistema.

Surge la inquietud acerca de la necesidad de emplear los casos de uso y existen básicamente dos razones fundamentales para su uso:

- Proporcionan un medio sistemático e intuitivo de capturar los requisitos funcionales con el fin de enfocarse en el valor añadido para el usuario.
- Dirigen todo el proceso de desarrollo y sirven para la planificación del diseño, casos de prueba y arquitectura del sistema software.

---

<sup>7</sup> Pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades. Ej: modelos, diagramas, documentos

Es de tener en cuenta que *“La perspectiva que proporcionan los casos de uso refuerza el objetivo último de la ingeniería del Software: la creación de productos que permitan a los clientes realizar un trabajo útil”*.<sup>8</sup>

Cuando se hace referencia a los requisitos funcionales debemos tener en mente a aquellos requisitos realmente importantes o relevantes, para esto nos valemos de algunas preguntas simples que deben discutirse al interior del equipo de trabajo tales como ¿A que tipo de usuarios ayudan estas funcionalidades? , ¿Qué necesidades del negocio satisfacen?, ¿Cuál resultado de valor añaden al negocio?; al responder estas simples preguntas podremos estar en capacidad de diferenciar aquellas funcionalidades realmente trascendentes que, a la larga se convertirán en la columna vertebral del desarrollo del proyecto informático en el que estemos involucrados.

El otro gran objetivo del enfoque de los casos de uso es lograr una adecuada presentación de los requerimientos del software a los **Stakeholders**<sup>9</sup>, esto teniendo en cuenta que no todas las personas relacionadas con el proyecto tienen las mismas necesidades, un desarrollador centrará su atención en los aspectos técnicos relacionados con el diseño mientras que un usuario final quiere conocer los aspectos funcionales que la aplicación le va a brindar, por otra parte la dirección de la empresa cliente tiene más interés en los aspectos de gerencia y control del proyecto; por tal motivo el RUP hace uso del Lenguaje Unificado para el Modelado UML como complemento esencial en todas las fases del proyecto , convirtiéndose en la herramienta de comunicación principal a lo largo del ciclo de vida del proyecto software.

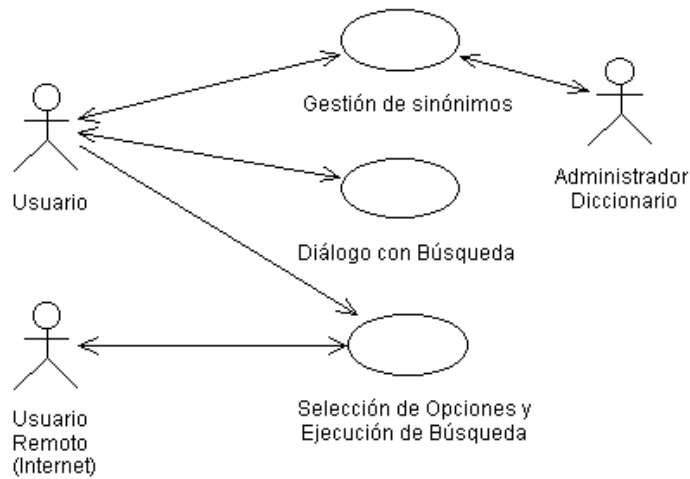
Pero el papel de los casos de uso no termina aquí ya que son encargados de dirigir todo el proceso del desarrollo de software. Todo el trabajo se edifica

---

<sup>8</sup> Wieger, Karl, “Use cases: Listening to customer’s voice”, Software Developmente, marzo 1997.

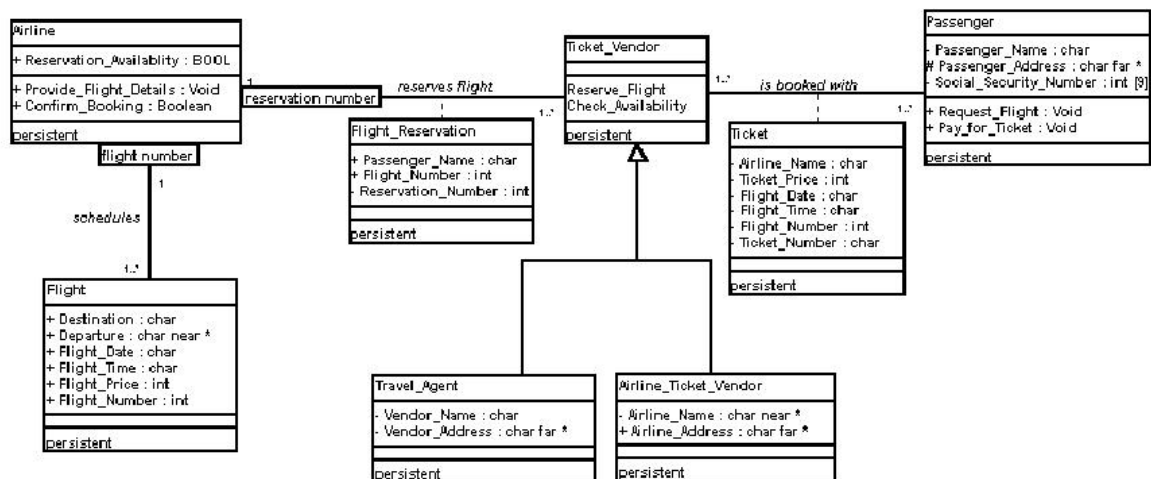
<sup>9</sup> Stakeholders: Son todos los grupos de personas que tienen algún grado de relación e interés en el software.

alrededor de una serie de flujos de trabajo que se inician a partir de los casos de uso y que concluyen con una versión operativa del subsistema, módulo o el sistema global que cumple con las funcionalidades requeridas por el usuario.



**Ilustración 3: Ejemplo del Diagrama de Casos de Uso.**

#### 4.2.2 Un Proceso Centrado en la arquitectura



#### **Ilustración 4: Ejemplo de un Diagrama de Clases.**

La arquitectura del Software es un concepto fácil de entender, y casi de manera intuitiva es usado por los ingenieros de sistemas aun con un nivel de experiencia mínimo, sin embargo es muy difícil de explicar con precisión y a menudo es confundido con el diseño.

Más allá de los algoritmos y las estructuras de datos propios de la programación la arquitectura del software tiene que lidiar con otro tipo de asuntos tales como protocolos de comunicación, sincronización, control, acceso a datos, distribución física, rendimiento, escalabilidad entre otros, lo cual deja ver la gran importancia de la arquitectura en el desarrollo de un proyecto software y por ende invita a dedicar especial atención a este punto en particular.

En el RUP la arquitectura del software se representa mediante un conjunto de vistas de arquitectura entre las cuales se tienen:

- Vista de Casos de Uso. Contiene los casos de uso y escenarios que guían los comportamientos y riesgos significativos de la aplicación.
- Vista Lógica. Contiene las realizaciones de los casos de uso, así como las clases de diseño más importantes y su agrupación en subsistemas.
- Vista de Implementación. Contiene la materialización de los elementos de diseño y su organización en funciones, programas, paquetes y módulos.
- Vista de Procesos. Es la descripción de todas las actividades que se deben llevar a cabo para cumplir con algún proceso determinado en la empresa cliente.

- Vista de Despliegue. Muestra la ubicación física de los diferentes componentes de la aplicación, las redes computacionales de la empresa, y la ubicación de los diferentes artefactos que son el resultado del proceso de desarrollo de software.

Con esta pequeña introducción a la arquitectura del software es posible ahora explicar el significado de este punto. El ser un proceso centrado en la arquitectura significa que la arquitectura de un sistema se usa como artefacto primordial para la conceptualización, construcción, gestión y evolución del sistema en desarrollo.

Teniendo en cuenta que la principal labor de la ingeniería es el diseño, cabe anotar que en la arquitectura software deben tenerse en cuenta:

- La organización del sistema.
- Los elementos estructurales, interfaces, comportamientos.
- La composición de los elementos estructurales.

Se hace necesaria la arquitectura para:

- Comprender el sistema
- Organizar el desarrollo
- Fomentar la reutilización
- Hacer evolucionar el sistema

### 4.2.3 Un proceso Iterativo e Incremental

Debido a que el desarrollo de un sistema de información puede tardar desde varios meses hasta algunos años es práctico dividir el trabajo en partes más pequeñas o *mini proyectos* siendo cada uno de ellos una *iteración*, que al completarse se convierte en un *incremento* en el crecimiento del producto.

Algunas de las ventajas que se pueden obtener con estas características son:

- Reducción de costos asociados al riesgo
- Permite el manejo adecuado de los riesgos críticos y significativos desde el principio.
- Permite la puesta en marcha de una arquitectura que guíe el desarrollo.
- Proporciona un marco de trabajo para la gestión de requisitos.
- Permite la re-evaluación de las necesidades de los usuarios ya que estas no se pueden definir totalmente al inicio del proyecto.
- Reducción de sobre tiempos.
- Aumento en el ritmo de trabajo debido a la motivación de metas a corto plazo.

#### 4.2.4 Fases del Proceso

El RUP es un proceso que divide el ciclo de vida del proyecto en cuatro fases: inicio, elaboración, construcción y transición, las cuales serán explicadas con mayor detalle a continuación.

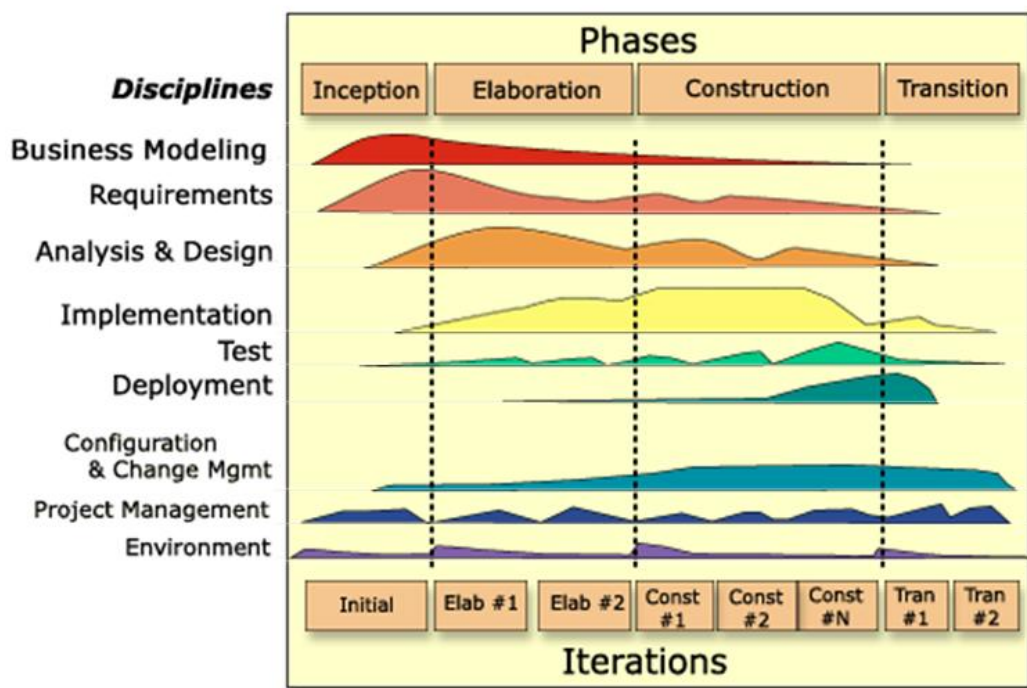


Ilustración 5: Vista general del modelo de desarrollo RUP.

4.2.4.1 Fase de Inicio. Los objetivos generales de esta fase son:

- Definir el alcance del proyecto, sus límites, una versión operativa de lo que debe y lo que no debe hacer la aplicación.
- Encontrar los casos de uso más significativos, es decir aquellos que tengan un mayor riesgo asociado.
- Proponer una arquitectura candidata.

- Estimar riesgos potenciales.
- Estimar costos.
- Preparar el ambiente para la realización del proyecto.
- Desarrollar el documento de visión.

Como hito principal<sup>10</sup> de esta fase de inicio se espera al artefacto: “Objetivos del ciclo de vida”, el cuál básicamente busca analizar la viabilidad del proyecto en términos generales.

4.2.4.1 Fase de Elaboración. Los objetivos generales de esta fase son:

- Recopilar la mayor parte de los requisitos formulando aquellos que sean funcionales como casos de uso.
- Establecer la línea base de la arquitectura que guiará el trabajo en las fases subsecuentes.
- Análisis continuado de los riesgos críticos inherentes al proyecto, sus planes de mitigación y contingencia así como su impacto en el ámbito financiero.
- Realizar un plan de trabajo base para el desarrollo del proyecto.
- Refinar el documento de visión del proyecto iniciado en la fase anterior.

El hito principal esperado en esta fase se conoce como: “Línea base de Arquitectura del software”.

---

<sup>10</sup> Punto de sincronización en el que coinciden una serie de objetivos bien definidos, se completan artefactos y en los que las esferas técnicas y de gestión entran en conjunción.

4.2.4.2 Fase de Construcción. Tiene como propósito general la “construcción” del sistema y dejar lista una versión operativa preliminar que se conoce como “versión Beta”. Para ello es necesario llevar a cabo las siguientes actividades:

- Optimizar los recursos evitando el re-trabajo.
- Alcanzar niveles de calidad tanto en el producto como en los procesos.
- Producir de manera iterativa e incremental versiones ejecutables del sistema.

Como hito principal de esta fase se tiene la versión beta del sistema desarrollado.

4.2.4.3 Fase de Transición. Esta fase tiene como objetivo principal asegurar que el producto llegue a los usuarios finales, así como gestionar todos los aspectos relacionados a la operación en el entorno del usuario.

Los objetivos principales de esta fase son:

- Validar el sistema frente a las expectativas de los usuarios.
- Establecer concordancia entre el sistema anterior y el nuevo.
- Capacitar a los usuarios del sistema.
- Montaje e instalación del producto en su ambiente real.
- Entrega formal del producto y aceptación por parte de los **Stakeholders**.

#### 4.2.5 Disciplinas o flujos de Trabajo.

Una disciplina en el RUP hace referencia a un conjunto de actividades relacionadas a un motivo de interés importante dentro del proyecto global. Mediante la agrupación de dichas actividades se busca que el proyecto sea más fácilmente entendido desde una perspectiva en cascada.

4.2.5.1 Modelado del Negocio. Tiene como objetivo la identificación y comprensión de los procesos de negocio de la empresa cliente a fin de proponer mejoras en los mismos, se busca que todas las partes interesadas en el proyecto tengan una visión común de la empresa cliente que permita la comprensión de las necesidades a ser resueltas y como el producto software a ser desarrollado ayudaría a alcanzar estos objetivos.

Debido a la participación de diferentes grupos de interés (**stakeholders**) sobre el proyecto se hace necesario el modelado de los procesos de negocio de una forma lo suficientemente clara y concisa como para ser entendido por todos éstos.

Entre las actividades principales propias de esta disciplina se tienen:

- Captura de un vocabulario o glosario común.
- Encontrar actores<sup>11</sup> y casos de uso del negocio.

---

<sup>11</sup> Conjunto coherente de roles que los usuarios de casos de uso desempeñan cuando interactúan con los casos de uso.

4.2.5.2 Recopilación de requisitos. Los objetivos fundamentales de esta disciplina son el definir y mantener un acuerdo entre la empresa cliente y la empresa desarrolladora respecto a lo que el sistema debe hacer, brindar al equipo de trabajo un mejor entendimiento acerca de los requerimientos del software, delimitar el alcance del sistema, proveer una base para el planeamiento de las iteraciones en las diferentes fases, así como definir una interfaz de usuario que cumpla con las necesidades de los usuarios.

Para alcanzar estos objetivos es extremadamente importante entender la definición y alcance del problema, y para ello se cuenta con artefactos tales como el documento de visión el cual brinda un panorama completo del sistema software a la vez que da soporte al contrato entre la empresa cliente y la empresa desarrolladora, cabe resaltar que éste es un documento escrito en un lenguaje no técnico y desde la perspectiva del usuario.

Otro artefacto importante que ayuda a alcanzar los objetivos de esta disciplina es el diagrama de casos de uso, el cuál muestra un conjunto de casos de uso, los actores y sus relaciones, convirtiéndose en un poderoso medio de comunicación entre los usuarios y los desarrolladores, permitiendo que los usuarios finales validen las funcionalidades esperadas del sistema software a ser desarrollado.

La definición completa de los requerimientos del software es encapsulada en el Documento de Especificación de Requerimientos del Software (SRS).

Dentro de las actividades a realizar se encuentran la priorización de casos de uso, el desarrollo de una versión ampliada del documento de visión y del modelo de casos de uso.

4.2.5.3 Análisis y Diseño. El análisis es un flujo de trabajo fundamental que tiene como principal objetivo analizar los requisitos capturados o descritos en la captura de requerimientos, por medio de su refinamiento y estructuración, buscando una mayor comprensión de los mismos.

De igual forma el diseño, como flujo de trabajo fundamental, busca formular modelos que se centren en los requisitos no funcionales de la aplicación y que sirva de base para la implementación y las pruebas.

Algunas de las actividades que se realizan en este flujo de trabajo son las siguientes:

- Análisis y refinamiento de casos de uso.
- Diseño de clases.
- Diseño de interfaz de usuario.
- Diseño de la base de datos.

4.2.5.4 Implementación. Su principal objetivo es la elaboración del sistema en términos de componentes, es decir, código fuente, programas ejecutables, paquetes, entre otros.

Además, definir la organización de dichos componentes en los diferentes subsistemas que conformarán cada uno de los módulos que compondrán la aplicación.

Por último integrar los diferentes componentes y así constituir las diferentes versiones ejecutables de la aplicación conocidas como ***builds***<sup>12</sup>.

4.2.5.5 Pruebas. Su objetivo primordial es encontrar y documentar los errores encontrados en el sistema software, de igual forma validar los diseños mediante demostraciones concretas que muestren que se cumple con las funcionalidades expresadas en el modelo de casos de uso.

Es muy importante tener en cuenta que es de vital importancia el escoger el enfoque que dirigirá el trabajo en lo referente a las pruebas, ya que un enfoque muy permisivo no permitirá encontrar las debilidades del sistema, por otra parte, un enfoque demasiado pesimista no dará lugar a la realización del sistema.

Tradicionalmente las pruebas han sido realizadas sin seguir ninguna metodología y esto ha conllevado a una serie de problemas que tienen que ver con la falta de elementos que permitan realizar un seguimiento de los errores causando además sobre-costos y teniendo un impacto alto en el detrimento de la calidad del software.

Para remediar esta situación el RUP establece la realización, desde las fases tempranas del proyecto, de una serie de actividades relacionadas con las pruebas y por supuesto guiadas por los casos de uso, generando lo que se conoce como los casos de prueba.

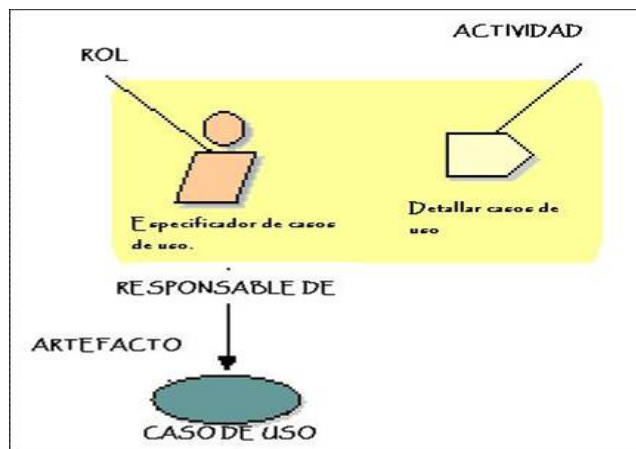
---

<sup>12</sup> Versión operativa de un sistema software o parte de éste, la cuál demuestra un subconjunto de funcionalidades incluidas en la versión final del mismo.

Estos artefactos buscan que las pruebas se realicen de una forma sistemática, controlada, iterativa e incremental asegurando de esta manera la calidad general en el producto software.

4.2.5.6 Gerencia del Proyecto. Esta disciplina tiene como objetivo fundamental el proveer guías para gestionar de manera efectiva todas las variables que se presentan en un proceso de desarrollo de software, tales como la contratación de personal, manejo de presupuestos, manejo de riesgos, planeación de iteraciones, y en general todo lo concerniente a la ejecución, control y monitoreo del proyecto.

#### 4.2.6 Roles



**Ilustración 6: Ejemplo de los Roles en el RUP.**

Este es uno de los conceptos clave del RUP. Un rol define un comportamiento y un conjunto de responsabilidades atribuibles a un individuo o a un conjunto de individuos que realizan actividades similares como equipo.

Es muy importante aclarar que un rol no necesariamente está relacionado con un individuo ni con una profesión o cargo en particular al interior de la empresa desarrolladora de software, incluso es necesario que un individuo desempeñe más de un rol dentro de un proyecto de desarrollo.

## 4.3 MEJORES PRÁCTICAS EN INGENIERÍA DEL SOFTWARE.

El RUP muestra cómo aplicar las 5 mejores prácticas en ingeniería del software, al tiempo que da guías de cómo utilizar herramientas informáticas para automatizar las diversas actividades que se desprenden de su aplicación. A continuación se describirán brevemente.

### 4.3.1 Desarrollo Iterativo.



**Ilustración 7: Modelo de Desarrollo iterativo**

Un proyecto que usa un modelo de desarrollo iterativo se caracteriza por llevar un ciclo de vida consistente en múltiples iteraciones. Cada iteración se compone de una serie de actividades en diferentes disciplinas propias del desarrollo de

software tales como análisis, diseño, implementación y pruebas en diferentes proporciones de acuerdo a la fase en que se encuentre el proyecto.

Las iteraciones de las fases de inicio y elaboración se centran en la captura de requerimientos, gerencia del proyecto y análisis principalmente; en la fase de construcción las actividades están enfocadas hacia el diseño, la implementación y las pruebas; mientras que en la fase de transición se centran los esfuerzos en las pruebas y el despliegue.

El desarrollo iterativo tiene la particularidad de tener un cronograma fijo para cada iteración lo cual permite la re-evaluación de metas y objetivos propuestos al equipo de trabajo al tiempo que permite la identificación de problemas y permite la búsqueda de una solución oportuna.

4.3.1.1 ¿Por qué Desarrollar software de forma iterativa? Todos los proyectos informáticos tienen riesgos inherentes a la naturaleza de dicha actividad, entre más temprano en el ciclo de vida del proyecto éstos sean identificados más factible será la elaboración de un plan de mitigación y contingencia y con seguridad será más probable la elaboración de planes de trabajo acertados.

Muchos de los riesgos presentes en un proyecto software no son descubiertos sino hasta la etapa de implementación, siendo extremadamente costosos e incluso en ocasiones pueden llevar a la cancelación del proyecto. Nunca se podrán predecir ni evitar todos los riesgos presentes sin importar el grado de experiencia del equipo de trabajo.

#### 4.3.1.2 Beneficios del enfoque iterativo. Se nombran a continuación:

- Los riesgos son mitigados desde el inicio del proyecto.
- Los requerimientos “cambiantes” son acomodables a lo largo del ciclo de vida del proyecto.
- Facilita el mejoramiento y refinamiento del producto dando como resultado una versión final más robusta.
- Se genera un aprendizaje en el equipo de trabajo propiciando el mejoramiento en los procesos.
- Da soporte a la gerencia del proyecto.

#### 4.3.2 Gestión de Requerimientos.

Es una aproximación sistemática para encontrar, documentar, organizar y hacer seguimiento a los requerimientos cambiantes de un sistema de información, tiene como objetivo principal mantener un estado claro de los requerimientos junto con unos atributos apropiados y la trazabilidad a otros requerimientos y artefactos.

La captura de requerimientos puede parecer una tarea fácil pero en la realidad se ve que muchas empresas de desarrollo de software atraviesan por serias dificultades debidas principalmente a las siguientes razones:

- Los requerimientos no siempre son obvios y provienen de diferentes fuentes.

- Los requerimientos no siempre son claramente expresados con palabras.
- Existen diferentes tipos de requerimientos cada uno con un nivel de detalle diferente.
- Al no estar controlados los requerimientos se vuelven inmanejables.
- Al existir diferentes partes interesadas en el proyecto es necesario que los requerimientos sean gestionados por grupos interdisciplinarios.

Sin importar que tan cuidadosamente se definan los requerimientos siempre existirán cambios. Desde el punto de vista de la gestión de requerimientos lo que causa un verdadero impacto es la incidencia de dichos cambios en otros requerimientos ya establecidos.

#### 4.3.3 Desarrollo basado en componentes.

Un desarrollo basado en componentes es una variación del modelo tradicional de desarrollo de software en el cual la aplicación es construida en torno a unidades ejecutables discretas desarrolladas casi de manera independiente y por equipos de trabajo diferentes.

Con este modelo de desarrollo la aplicación es fácilmente escalable a través de pequeños incrementos, en el RUP esto se logra ya que el resultado de cada iteración logra un incremento en la funcionalidad prestada por la aplicación.

Estos componentes son el resultado de la definición de una arquitectura modular donde estos se identifican, definen, diseñan, desarrollan y prueban logrando construir la aplicación como si se tratara de simplemente unir por medio de

interfaces a estos bloques de componentes que conforman los diferentes módulos del sistema.

Una de las ventajas más importantes derivadas de la correcta aplicación de el desarrollo basado en componentes es la posibilidad de reutilizar dichos componentes para diferentes proyectos informáticos logrando con ello que la empresa desarrolladora sea más eficiente y contribuye a la búsqueda de calidad tanto en sus productos software como en sus procesos internos.

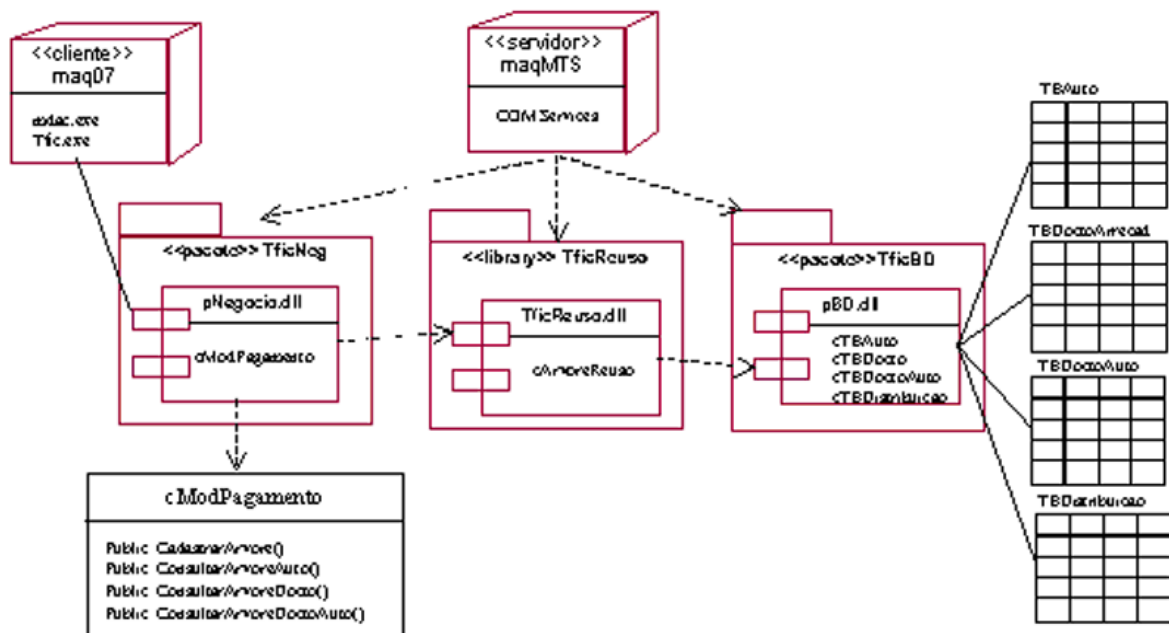


Ilustración 8: Ejemplo de un diagrama de componentes.

4.3.3.1 ¿Cómo soporta el RUP al desarrollo basado en componentes? Uno de los pilares fundamentales de la filosofía del RUP es la característica de ser iterativo e incremental, pues bien al llevar a cabo un modelo de

desarrollo de software iterativo es posible la identificación y refinamiento de los componentes que conformarán los distintos módulos del sistema, al mismo tiempo que permite a la gerencia del proyecto decidir cuales componentes son más factibles de ser comprados a un tercero.

Cabe anotar que la aplicación del desarrollo basado en componentes facilita las labores del equipo de trabajo en la disciplina de las pruebas toda vez que éstas se realizan de manera incremental, primero sobre los componentes, siguiendo con cada módulo para terminar con las pruebas generales a los subsistemas y a la aplicación como un todo.

#### 4.3.4 Modelado Visual (UML).

El modelado visual es el uso de notaciones de diseño ricas en semántica, gráficas y texto que en conjunto con una notación especializada como la provista por UML permite alcanzar un nivel de abstracción al tiempo que mantiene una sintaxis y semántica rigurosas. De esta forma se mejora la comunicación en el equipo de trabajo.

4.3.4.1 ¿Por qué modelamos? Un modelo es una vista simplificada de un sistema. Muestra los elementos esenciales de un sistema desde una perspectiva determinada y esconde los no esenciales.

Los modelos pueden ser útiles en las siguientes formas:

- Ayudar a la comprensión de sistemas complejos.

- Ejemplos:
  - Casos de uso: especifican inequívocamente un comportamiento del sistema.
  - Diagrama de clases y modelo de datos: especifican el diseño.
  - Diagrama de Estados: modela el comportamiento dinámico del sistema.
  - Explorar y comparar alternativas de diseño a un bajo costo.
  - Crean una base para la implementación.
  - Ayudan a un proceso de captura de requerimientos más preciso.
  - Se constituyen como una herramienta fundamental a la hora de comunicar decisiones inequívocamente.

En la actualidad el UML representa la mejor práctica en modelado de software.

#### 4.3.5 Verificación continua de Calidad.

4.3.5.1 ¿Qué es calidad? La calidad es un concepto que, de manera similar a la arquitectura del software, es muy intuitivo y todas las personas tienen un concepto que difícilmente pueden explicar de una manera clara y contundente, algunas definiciones se presentan a continuación:

- *“Propiedades inherentes a un objeto que permiten apreciarlo como mejor, igual o peor que otros objetos de su especie.”* (RAE).
- “Grado en el que un conjunto de características inherentes cumplen con los requisitos”. (ISO 9000).
- “El grado en el que un sistema, componente o proceso cumple con:
  - Requisitos
  - Necesidades o expectativas de los clientes. (IEEE)

Algunas de las métricas que se tienen en cuenta para verificar la calidad en un proyecto informático son:

- Progreso.
- Variabilidad.
- Confiabilidad.
- Funcionalidad.
- Rendimiento.

En el RUP el manejo y la búsqueda de la calidad es implementada a lo largo del ciclo de vida del proyecto y en cada una de las disciplinas.

## **5. METODOLOGIA EMPLEADA.**

### **5.1 Análisis del entorno.**

En esta fase además de llevar a cabo un proceso de adaptación al nuevo entorno de trabajo se realizó un análisis de la empresa, en especial de su modelo de desarrollo de software a fin de establecer una base para una posterior comparación entre este modelo y el propuesto por el RUP.

### **5.2 Consulta de información acerca de la metodología RUP.**

A continuación de la fase de análisis del entorno se dio inicio a una etapa de consulta y recolección de información que buscaba profundizar y ampliar los conocimientos acerca del RUP la cuál buscaba crear una base de conocimiento que sirviera de plataforma para la elaboración de un material de trabajo con el cual se llevaría a cabo un proceso de transferencia tecnológica y conceptual que tendría lugar en la siguiente fase a manera de capacitación.

### 5.3 Capacitación sobre los Fundamentos del RUP.

Una vez finalizada la fase de consulta se procedió a la elaboración de un material de trabajo que sirviera para dar soporte al proceso de capacitación sobre los fundamentos del RUP.

El objetivo principal de esta fase era transferir el conocimiento construido acerca del RUP y medir el grado de aceptación de los nuevos conceptos aportados por la metodología del RUP al interior del equipo humano de la empresa.

Se buscaba por medio de esta capacitación que los ingenieros de la empresa despejaran dudas respecto a la metodología y de esta manera poder concluir si en realidad ésta era la más indicada para la realidad de la empresa.

### 5.4 Socialización y Talleres.

La fase de capacitación se dividió en torno a los diferentes aspectos fundamentales del RUP, al finalizar cada uno de estos temas se propiciaba un espacio de discusión que tenía como objetivo fundamental la proposición de ideas respecto a la posible implementación de dichos temas en la empresa, analizando los pros y los contras y siempre estableciendo un paralelo con la forma con la que tradicionalmente se han enfrentado dichas situaciones en la empresa.

Con el fin de poner en práctica los diferentes conceptos aprendidos en las capacitaciones se realizaron al final de cada una de las sesiones unos talleres prácticos los cuáles fueron una experiencia enriquecedora para todos los

participantes, validando el trabajo teórico llevado a cabo durante toda esta fase de la práctica empresarial.

Al final de la práctica empresarial se llevó a cabo por parte de la empresa un proceso de planeación interna que buscaba definir la forma más adecuada de iniciar con el proceso de implementación del RUP en firme, es decir, se buscaba definir aquellos aspectos críticos donde se hacía necesaria la asesoría de empresas especializadas, por ejemplo en el manejo y gestión de riesgos, en la estimación de costos entre otros, aspectos que sin lugar a dudas son claves en el inicio de cualquier proyecto software.

## **6. EXPERIENCIA DE LA PRÁCTICA EMPRESARIAL.**

A continuación se presenta una visión general del proyecto AQUASOFT.

Sistema integrado de información para acueductos metropolitanos que consta de los siguientes módulos:

- Seguridad
- Auditoria
- Conciliación de operadores
- Estadísticas
- Mercadeo
- Daños
- Redes

- Cartera
- Atención al cliente
- PQR
- Tarifas
- Facturación

**Características generales:**

- Solución basada en componentes y reglas del negocio.
- Interfaz gráfica que ofrece un ambiente de alta productividad y elevada estética.
- Estandarización y parametrización sencilla y totalmente abierta
- Diseñada bajo ambiente Web.
- Sistema de seguridad a nivel de aplicación, base de datos y componentes de administración.

En primer lugar se programó una reunión donde participaron el Gerente de la Empresa en ese momento, Ingeniero Oscar González Pimentel, el Director de proyectos y tutor de esta práctica empresarial Ingeniero Humberto Rueda Rivero, y el autor de este documento con el fin de dar un vistazo general al proyecto software escogido para la realización del trabajo de la práctica empresarial el proyecto AQUASOFT, así como para mostrar un panorama general del proceso de desarrollo de software empleado por la empresa, el cuál se presenta a continuación.

## 6.1 Proceso de Desarrollo de Software de TyT Ltda.

### 1. Captura de requerimientos

#### 1.1 Elaboración del Listado General de Requerimientos.

Este documento recopila los requerimientos técnicos y funcionales generales del sistema en general. En este documento se puede ver una versión preliminar de la arquitectura a ser empleada en el desarrollo de la aplicación, haciendo énfasis en la modularidad, y el carácter de ser una aplicación multiempresa.

#### 1.2 Elaboración Documento de especificación de Requerimientos.

#### 1.3 Elaboración Formato de objetivos de los requerimientos.

Es un documento que busca explicar de manera muy específica cuales son los objetivos que deben ser cumplidos por los requerimientos funcionales listados en el documento citado en el punto anterior.

### 2 Definición de los Casos de Uso.

Este proceso se hace siguiendo la experiencia de los ingenieros de la empresa. Cada caso de uso es extraído del listado de requerimientos, para posteriormente ser listado en el documento de casos de uso candidatos que se explica a continuación.

### 2.1 Listado de posibles casos de uso.

Es un documento que, como lo explica su nombre, contiene aquellos casos de uso que serán analizados a fin de decidir si se omiten, o si serán objeto de mayor análisis.

### 2.2 Establecimiento de prioridad para los Casos de Uso.

Esta labor se lleva a cabo teniendo en cuenta factores tales como las necesidades del mercado, y en líneas generales tratando de cubrir en principio las funcionalidades básicas y posteriormente las complementarias. Esta es una actividad que es realizada por el equipo de ingenieros de mayor experiencia en la empresa, teniendo en cuenta su importancia en la organización del trabajo subsiguiente.

### 2.3 Documento de especificación de Casos de uso.

La empresa, por propia cuenta había adelantado un estudio preliminar acerca del RUP centrandose su atención principalmente en el tema referente a los casos de uso. Como resultado de dicha experiencia se implementó el documento de especificación de casos de uso el cual fue motivo de estudio durante la práctica empresarial.

Dicho documento contiene los siguientes campos: Identificación, Nombre, Versión, Autor, Fuentes, Objetivos asociados, Requisitos asociados, Descripción, Precondición, Secuencia normal, Post condición, Excepciones, Rendimiento,

Frecuencia Esperada, Importancia, Prioridad, Estado, Estabilidad, Comentarios, Chequeo.

### 3 Análisis y Diseño.

En esta área el equipo de TyT Ltda cuenta con una reconocida experiencia adquirida por más de 15 años de trabajo en diferentes proyectos informáticos de gran magnitud. A continuación se describe de manera muy general el proceso seguido en esta etapa.

#### 3.1 Elaboración de los diagramas de Entidad-Relación.

Es la actividad principal de esta etapa; para esto se emplea la herramienta CASE de Oracle: Designer. En este momento es pertinente decir que la empresa TyT Ltda hace parte del **Oracle PartnerNetwork**<sup>13</sup> y por ende el fuerte de su actividad está orientado al desarrollo de aplicaciones bajo esta tecnología.

#### 3.2 Elaboración de los Diagramas de Datos.

Se constituyen como la base del sistema a ser desarrollado; estos diagramas reposan en un repositorio de datos de Oracle.

---

<sup>13</sup> Oracle PartnerNetwork (OPN) es una comunidad de más de 19.500 Partners o asociados alrededor del mundo, que usan y comercializan los productos de Oracle.

### 3.3 Bosquejos de interfaz de usuario

Para esta actividad no se utiliza herramienta alguna, se realiza manualmente y es para uso interno, estos bosquejos no son presentados al usuario.

## 4 Desarrollo.

Se conformaron diferentes grupos de trabajo, cada uno liderado por uno de los ingenieros socios de la empresa. Cada grupo tenía bajo su responsabilidad el desarrollo de diferentes componentes para un módulo en particular.

Siguiendo su experiencia, el trabajo en todos los grupos se iniciaba con el desarrollo de una serie de funciones y programas básicos necesarios en cada módulo, la cual se denominó como la biblioteca general de funciones que se describe a continuación.

### 4.1 Biblioteca general de funciones.

Consta de diferentes funciones y programas organizados por paquetes los cuales proporcionan funcionalidades básicas como por ejemplo manejo de fechas, operaciones aritméticas, operaciones sobre cadenas de caracteres y validaciones en general.

#### 4.1.1 Especificación de rutinas a construir.

Para cada uno de los programas y rutinas a desarrollar la empresa crea un documento de especificación donde se muestra la siguiente información:

- Paquete. Información del paquete que contiene al programa o función, su nombre, parámetros de entrada.
- Descripción. Breve descripción de su objetivo y resultados esperados.
- Rutinas/ funciones. Nombre, descripción y objetivos de cada rutina o función.
- Parámetros. Parámetros de entrada y de salida de cada rutina o función.
- Funcionalidad. Descripción de la funcionalidad prestada y su utilidad.

#### 4.2 Programación.

Ya entrados en materia se programan por una serie de metas para cada período y se asignan una serie de responsabilidades a cada ingeniero desarrollador siendo su guía el listado de casos de uso previamente priorizados y contando con la supervisión del gerente de proyecto.

#### 6.2. Capacitación sobre los fundamentos del RUP.

Como resultado de la fase de consulta se recopiló un material de apoyo que trataba de una manera muy concreta los principales aspectos a tener en cuenta a la hora de iniciar un proceso de implementación del RUP.

Atendiendo los consejos de expertos en el tema se hizo énfasis en el estudio de los pilares fundamentales de la filosofía del RUP, que como se ha venido mencionando en el presente informe son sus tres características principales, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental; de la misma forma con los temas referentes a la aplicación de las mejores prácticas en ingeniería del software.

Otro punto muy importante que se tuvo en cuenta durante la fase de consulta y consecuentemente en la fase de capacitación fue conocer aquellos factores críticos que dificultan un proceso de implementación de ésta naturaleza con el propósito de establecer la mejor forma de evitar dichas situaciones y sacar el mayor provecho de este proceso.

El objetivo de las primeras jornadas de capacitación era captar la atención de parte de los ingenieros de la empresa y de esta forma motivarlos a participar de manera activa en el desarrollo de las diferentes charlas, objetivo que se cumplió desde muy temprano para beneficio de todos.

Uno de los temas que más presente estuvo durante todo la práctica empresarial es la necesidad que el proceso de implementación del RUP al igual que el RUP en sí se lleve a cabo de manera iterativa e incremental, aplicando esto al contexto de la fase de capacitación fue necesario diseñar un plan de estudio que abarcara los temas más importantes y estos fueron estudiados de manera general. Lo anterior teniendo en cuenta que toda la fase de capacitación buscaba dar un panorama general del RUP sin entrar en detalles que hicieran caer al auditorio en pánico, ya que de acuerdo a los expertos, el tratar de estudiar y aplicar en su totalidad el RUP es una de las causas más frecuentes de fracaso en los procesos de implementación del mismo, ya que el RUP en su versión extendida llega a ser muy complejo.

A continuación se muestran los diferentes temas tratados en esta fase.

### 1. Fundamentos del RUP.

Se trabajó sobre los tres pilares fundamentales de la filosofía propuesta por el RUP son estos:

- Dirigido por casos de uso.
- Centrado en la arquitectura.
- Iterativo e incremental.

### 2. Fases y Flujos de Trabajo.

Se realizó una comparación entre Modelo en cascada vs iterativo, analizando sus similitudes y diferencias.

### 3. Mejores prácticas en Ingeniería del Software.

Se estudiaron los aspectos más relevantes de cada una de estas prácticas en ingeniería del software.

Los temas tratados fueron:

- Gestión de requerimientos.
- Desarrollo basado en componentes.
- Modelado Visual.
- Verificación continua de calidad

#### 4. Disciplinas en el RUP.

Se analizó el trabajo a realizar en cada una de las disciplinas usadas en el desarrollo de software, se listan a continuación:

- Modelado de Negocio.
- Análisis.
- Diseño.
- Implementación.
- Pruebas.

En cada uno de estos ítems se mostró la filosofía de trabajo propuesta por el RUP.

#### 5. Comparativa entre el modelo de análisis y el modelo de diseño.

Esta fue una sesión extra que se programó a fin de esclarecer algunas dudas surgidas durante la presentación del tema inmediatamente anterior en los apartados de análisis y diseño. Se llevó a cabo una comparación entre los dos modelos analizando cada una de sus características.

## 6. Roles en el RUP.

Este tema se dividió en dos sesiones a fin de abarcar cada rol con una mayor profundidad, en esta primera sesión se trabajó en torno a los siguientes temas:

- Analistas
- Desarrolladores

Para cada uno de estos roles se trataron los siguientes temas:

- Requisitos.
- Formas de asignar este rol.
- Actividades principales.

## 7. Roles en el RUP (Continuación).

Continuando con el tema se trabajó en base los siguientes temas:

- Managers o Directores.
- Testers o Personal de pruebas.

Los temas estudiados fueron:

- Requisitos.
- Formas de asignar este rol.
- Actividades principales

## 8. Antipatrones en el RUP.

El objetivo de esta charla fue en primer lugar hacer un sondeo general del grado de aceptación y entendimiento de los temas vistos durante todo el proceso de capacitación, y mostrar algunas fallas que frecuentemente se cometen en un proceso de implementación de éste tipo, a fin de crear conciencia entre el equipo y tomar las medidas correctivas en caso de presentarse alguno de estos síntomas.

Los temas tratados fueron:

- RUP en cascada. Se estudió como en algunos casos se estudia la propuesta del RUP, se asignan responsabilidades, se generan artefactos pero se descuida uno de los temas más importantes del RUP: el hecho de ser un proceso iterativo e incremental y simplemente se disfraza un modelo en cascada.
- Primera experiencia: RUP al 100%. Este es un tema en el que se hizo mucho énfasis durante toda la práctica empresarial ya que es uno de los más nocivos a la hora de implementar el RUP, como bien se ha dicho la implementación del RUP debe hacerse gradualmente a fin de sentar las bases del cambio cultural que supone dicho proceso.
- RUP: Solo comprar herramientas. El RUP se centra más en las actividades que en los artefactos y en propende por la aplicación de las mejores prácticas en ingeniería del software, con solo comprar herramientas tales como gestores de requerimientos o paquetes de modelado en UML no es

suficiente, el RUP busca un cambio organizacional enfocado a mejorar el proceso de desarrollo de software.

- Departamento de sistemas: Una isla. En este tema se habló acerca de la necesidad de contar con todos los niveles de la empresa a la hora de implementar el RUP, no solamente el departamento de sistemas, ya que el RUP afecta todos los procesos internos de la empresa.
- ¿Cuándo empezamos la programación?. Por lo general al iniciar el proceso de implementación del RUP el equipo de desarrollo quiere iniciar lo más pronto posible sus actividades de programación desatendiendo la propuesta del RUP y evitando las actividades correspondientes a las fases anteriores a la de construcción, lo cual se convierte en un problema que debe afrontarse con el fin de lograr una implementación del RUP exitosa.
- RUP: Receta de cocina. Este es otro error frecuente entre las empresas que deciden iniciar el proceso, como se mencionaba en la sección 4.2 el RUP es un modelo de desarrollo de software que se debe adaptar a las necesidades y a la realidad de cada empresa, ya que no es lo mismo implementarlo en una gran empresa con un grupo humano numeroso y un esquema organizativo complejo a hacerlo en una pequeña empresa con un grupo humano reducido como es el caso de TyT Ltda.
- ¿Cómo saber si no se entendió el RUP?. Por último se mostraron algunas señales de alerta que permiten evaluar el grado de entendimiento de la filosofía del RUP con el fin de mejorar en dichas áreas y continuar adelante con el proceso.

9. Resumen de la capacitación y proyección del proceso de implementación.

10. Fase de inicio: diagrama de casos de uso.

Se estudiaron con mayor detenimiento las actividades realizadas por el analista de sistemas y el arquitecto de software en la Fase de inicio del proyecto; dado que se presentaba como un producto de estas actividades la elaboración de una primera versión del diagrama de casos de uso, se llevó a cabo una introducción al mismo con el objetivo de sentar las bases para un ejercicio práctico que se menciona a continuación.

11. Taller: Elaboración del diagrama de Casos de Uso.

Esta fue en realidad una actividad muy enriquecedora para todos los participantes. Se propuso la elaboración de un diagrama de casos de uso para un subconjunto de estos que agrupaban una funcionalidad en particular del módulo de facturación para el proyecto AQUASOFT.

Los temas tratados fueron:

- Justificación de la necesidad del diagrama.
- Identificación de los elementos del diagrama.
- Pasos para la realización del diagrama.
- Refinamiento del diagrama.
- Expansión de los casos de uso de alto nivel.

- Herencia.
- Especialización de los casos de uso.
- Dependencia *Include*.
- Dependencia *Extend*.

El diagrama fue elaborado por los ingenieros de la empresa utilizando para ello el tablero de la sala de reuniones de la empresa. Allí fue posible observar y corregir algunos de los errores más frecuentes que se cometen al llevar a cabo por primera vez este diagrama, fueron estos:

- La falta de una plena identificación de los diferentes ACTORES que interactúan con el sistema.

Se hizo evidente la necesidad de utilizar dicho artefacto ya que este no había sido usado en las experiencias previas de la empresa, lo cual no contribuye a lograr una adecuada comunicación entre la empresa desarrolladora y el cliente respecto al comportamiento esperado del sistema. Gracias a la realización de éste taller se hizo evidente la necesidad y la importancia de la inclusión de dicho diagrama para futuros proyectos.

- La inclusión en el diagrama de casos de uso de muy alto nivel.

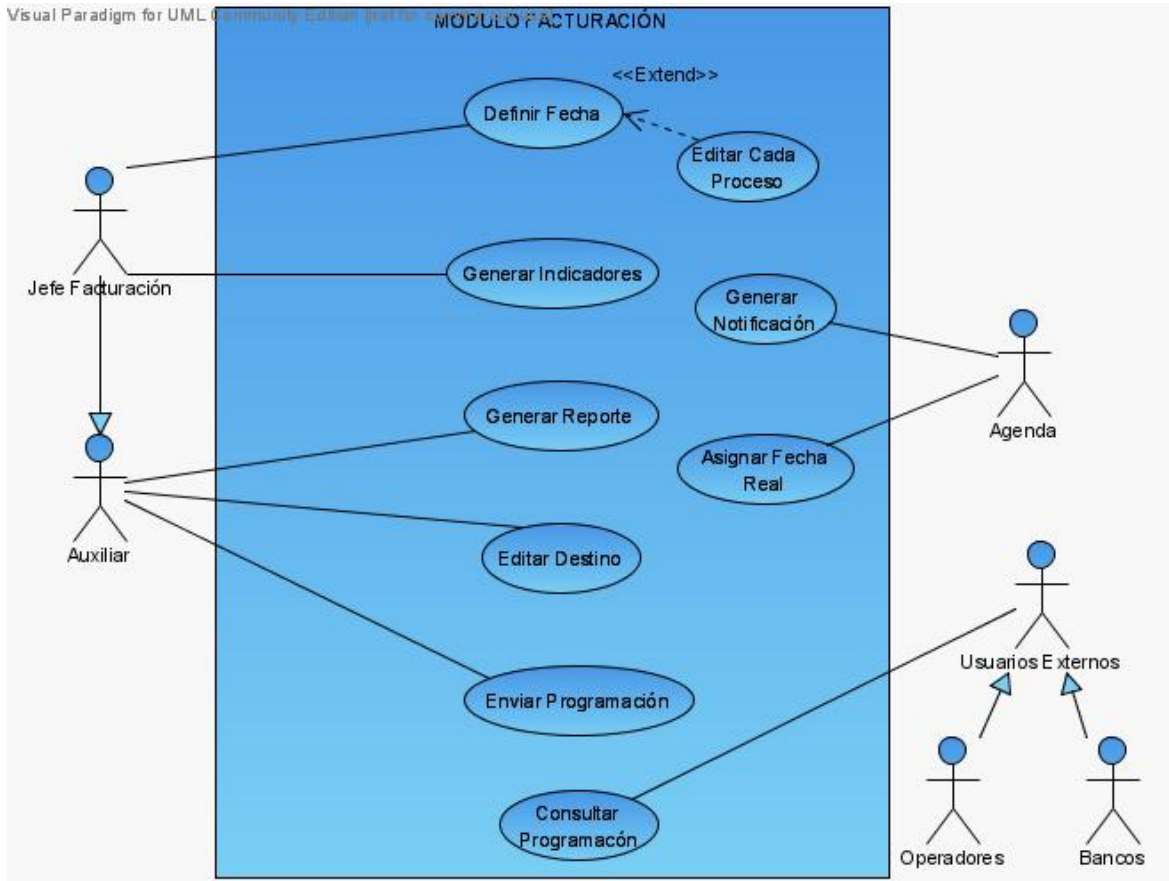
Se presenta cuando se incluyen en el diagrama casos de uso cuya especificación es muy general y no se refleja claramente la funcionalidad esperada del software.

- La confusión entre las relaciones de dependencia `<<include>>` y `<<extend>>`.

Es un error común que se presenta en las primeras experiencias con el diagrama; consiste en confundir cuándo un caso de uso necesariamente debe *incluir* la funcionalidad provista por otro caso de uso y cuándo dicha utilización es opcional y por tanto se *extiende* su funcionalidad a fin de cumplir con éxito su objetivo.

Fue tan productiva la sesión que incluso surgió un nuevo caso de uso que no se había contemplado para el módulo de facturación del proyecto en cuestión, se trató de la inclusión de una pantalla de consulta de la programación de los diferentes ciclos de facturación vía Web motivada por la necesidad existente por parte de los agentes externos recaudadores de los pagos así como los bancos de consultar dicha información sin tener que ingresar directamente al sistema.

A continuación se muestra dicho diagrama.



**Ilustración 9: Diagrama de Casos de Uso módulo de Facturación, Aquasoft**

12. Fase de inicio: Técnica CRC, diagramas de actividades, colaboración y secuencia.

Se exploró la técnica de definición de clases CRC (Class, Responsibility, Collaborator Card), la cuál consiste en la elaboración de una tarjeta que contiene la clase candidata, sus responsabilidades (operaciones y atributos) y las clases que “colaboran” en la realización de un caso de uso específico.

Dicha técnica es útil cuando el equipo de trabajo tiene poca experiencia en el dominio del negocio. Ahora bien, dado el alto grado de experiencia del equipo de trabajo de TyT Ltda en el área de identificación de las clases para proyectos de

este tipo se decidió la no inclusión de dicha técnica en el proceso de desarrollo de software de la empresa.

Una de las formas de refinar el modelo de casos de uso es mediante el uso de algunos diagramas auxiliares, tal es el caso del diagrama de actividades.

Éste es un diagrama opcional, su uso es recomendado cuando no existe total claridad respecto al proceso de realización de un caso de uso; para el caso en estudio (AQUASOFT) no se consideró pertinente su utilización.

Como introducción al modelo de análisis se abarcó el tema de los diagramas de colaboración y secuencia haciendo énfasis en las ventajas que se derivan de su utilización.

### 13. Taller: Elaboración de los diagramas de colaboración y secuencia.

Como complemento a las sesiones de capacitación anteriormente citadas se realizó un taller que buscaba despejar las dudas y mostrar las ventajas de la elaboración de estos diagramas, para ello se trabajó sobre un subconjunto de la funcionalidad general del módulo de facturación del proyecto AQUASOFT. Como resultado de este taller se decidió la inclusión del diagrama de secuencias ya que a juicio de los ingenieros de la empresa éste representaba una mayor utilidad y beneficio para el equipo de desarrollo.

Los temas tratados fueron:

- Elementos del diagrama.
- Tipos de Componentes.
- Pasos para la elaboración del diagrama.

## **7. ESTRATEGIA DE APLICACIÓN DE RUP EN LA EMPRESA TYT LTDA.**

En primer lugar analizar los diferentes procesos y actividades realizados por la empresa en torno a recopilación de requerimientos, análisis, diseño, implementación y pruebas; prestando especial atención a la identificación de los factores que influyen de manera negativa en su correcta realización.

En segundo lugar capacitar al personal en los temas que sean requeridos a fin de establecer las bases del mejoramiento de dichos procesos, dejando claro que el proceso de capacitación tomará el tiempo que sea necesario para dejar totalmente claros los diferentes conceptos propuestos por la metodología RUP.

En tercer lugar implementar los nuevos conceptos de manera gradual con el fin de minimizar el impacto propio del cambio en el esquema de trabajo de los miembros de la empresa.

Se hará un seguimiento a los procesos mejorados con el fin de verificar que cumplen los objetivos propuestos.

Por último al finalizar cada fase del proyecto de desarrollo en cuestión, para éste caso específico AQUASOFT, evaluar los resultados globales obtenidos, generar las propuestas de mejoramiento que sean necesarias y continuar con la ejecución del proyecto.

El documento “Caso de Desarrollo” muestra los diferentes productos esperados para la fase de inicio del proyecto AQUASOFT. Ver anexo Número 2.

## **8. CONCLUSIONES**

Es imperativo llevar a cabo un estudio y análisis del proceso de desarrollo de software utilizado al interior de las empresas del sector, con el fin de identificar aquellos aspectos que influyen de manera negativa en el resultado esperado por parte de los involucrados en un proyecto informático.

Dadas las actuales condiciones del mercado los profesionales de Ingeniería de Sistemas y en general todo profesional que se desempeñe en el área del desarrollo de software debe capacitarse y adaptar su modelo de trabajo a las mejores prácticas en Ingeniería del Software; son estas el desarrollo basado en componentes, modelado visual, gestión de requerimientos, desarrollo iterativo y la verificación continua de la calidad.

El asesoramiento de expertos se constituye en pieza fundamental que sustenta cualquier proceso de implementación de una nueva metodología para el desarrollo de software como es el caso del RUP.

Se hace necesaria la participación activa de todo el equipo humano de la empresa para asegurar el éxito del proceso.

La modalidad de práctica empresarial como medio para optar por el título de Ingeniero de Sistemas es una excelente oportunidad para que el estudiante haga una adecuada transición del ambiente académico al ambiente de trabajo real de la industria, aprendiendo de la experiencia de profesionales con un alto nivel de experiencia, al tiempo que brinda un espacio propicio para la exploración de los temas que son de interés y que marcarán el camino del estudiante como profesional.

## **9. RECOMENDACIONES**

Se recomienda a la empresa TyT Ltda continuar con el proceso de implementación del RUP y de manera iterativa seguir incrementando su conocimiento y aplicación de esta metodología.

Es necesario fortalecer los vínculos de cooperación interinstitucional entre la Escuela de Ingeniería de Sistemas de la UIS y las empresas del sector productivo no solo de la región sino del país ya que esta es una excelente oportunidad tanto para dar a conocer la calidad de nuestro recurso humano como para recibir Realimentación acerca de las necesidades presentes en el sector del desarrollo de software y de esta manera poder hacer las correcciones y mejoras en el enfoque que se le está dando a la carrera.

## 10. BIBLIOGRAFIA

BOOCH, Grady - RUMBAUGH, James - JACOBSON, Ivar. El Proceso Unificado de Desarrollo de Software. Madrid: Addison-Wesley, 2000.

EVANS, Gary ,A Simplified Approach to RUP,

<http://www-128.ibm.com/developerworks/rational/library/354.html>.

CRAIN, Anthony ,Understanding Rup Roles,

<http://www128.ibm.com/developerworks/rational/library/content/RationalEdge/apr05/crain/>

BROWN, Barklay, Top Five RUP implementation killers,

<http://www-128.ibm.com/developerworks/rational/library/1797.html>.

FILHO, Julia, Rational Unified Process Anti-Patterns. Black Diamond Software,2002.

Object-Oriented Analysis and Design Using UML Preface, 2003 Sun Microsystems, Inc.. Sun Services

## 11. ANEXOS

### ANEXO 1. Carta de certificación de la empresa TyT Ltda.



Tecnologías de Información y Telecomunicaciones LTDA

Bucaramanga, 27 de Agosto de 2007

**Señores:**

**Escuela de Ingeniería de Sistemas e Informática  
Universidad Industrial de Santander  
L.C**

Certificamos que en el marco del desarrollo de la práctica empresarial realizada por el estudiante Carlos Andrés Rueda Ribero, código: 2012684 la cual se titula:

**“DISEÑO E IMPLEMENTACIÓN DE UNA ESTRATEGIA METODOLÓGICA PARA LA APLICACIÓN DEL PROCESO UNIFICADO PARA EL DESARROLLO DE SOFTWARE EN LA EMPRESA TECNOLOGÍAS DE INFORMACIÓN Y TELECOMUNICACIONES TYT LTDA”**

Se llevó a cabo un proceso de transferencia de conocimiento acerca de los fundamentos del Proceso Unificado para el Desarrollo de Software RUP, el cual se constituyó en pieza fundamental dentro del proceso de actualización de nuestra metodología de construcción de aplicaciones.

Es importante recalcar que el trabajo realizado de parte del estudiante durante dicho proceso estuvo a la altura de nuestras expectativas y nos deja altamente satisfechos, ya que le permitió a nuestro equipo de ingenieros construir un conocimiento sólido acerca de las características más relevantes acerca del RUP, siendo éste un aporte muy significativo para nosotros debido en gran medida a que próximamente tendremos que hacer uso de estos conocimientos para transmitir nuestros diseños a la fábrica de software que llevará a cabo la labor de programación de un importante proyecto.

Destacamos igualmente que el producto recibido y la metodología que se ha utilizado para ello, nos ha permitido adoptar la metodología RUP al interior de

Cra. 19 No 35 – 02 Oficina 305 Antiguo Hotel Bucarica Tel: 670 1185 Bucaramanga  
Web site: [www.tyt.com.co](http://www.tyt.com.co) e-mail: [info@tyt.com.co](mailto:info@tyt.com.co)



Tecnológicas de Información y Telecomunicaciones LTDA

nuestra empresa, labor que nos había resultado infructuosa en intentos previos que habíamos adelantado sin la debida asesoría y/o acompañamiento; igualmente destacamos el compromiso del estudiante en lograr el cumplimiento y satisfacción de nuestras inquietudes y necesidades.

Sin otro particular y agradeciendo su atención,

Atte:

**JULIAN ANDRES LOPEZ URBINA**  
CC: 91.510.077 B/manga  
Gerente TyT Ltda.

**HUMBERTO RUEDA RIVERO**  
CC: 91.263.541 B/maga  
Tutor Práctica Empresarial  
Director de Proyectos TyT Ltda

## ANEXO 2. Documento Caso De Desarrollo

**AQUASOFT**

**Caso de Desarrollo**

**Versión 1.0**

### Historial de Revisiones

Fecha	Versión	Descripción	Autor
Mayo 15 de 2006	1.0	Versión Inicial	Carlos Andrés Rueda Ribero

### Tabla de Contenidos

- ☐ Introducción
- ☐ Generalidades del proceso de desarrollo de software
- ☐ Fases
- ☐ Modelado del Negocio
- ☐ Requerimientos
- ☐ Análisis y Diseño
- ☐ Gerencia del Proyecto

### Introducción

### Propósito

Este documento recoge las impresiones acerca de la primera fase exploratoria en la implementación del RUP en lo concerniente al proceso de desarrollo de software para el proyecto AQUASOFT.

### Alcance

Este documento aplica a la fase de inicio del proyecto AQUASOFT.

### Referencias

RUP Versión 2003.06.12.

### Generalidades

El presente documento describe la forma en que el RUP será implementado en las fases de inicio del proyecto AQUASOFT.

## **Generalidades del proceso de Desarrollo de Software**

En este momento se plantea el desarrollo del proyecto hasta la fase de elaboración, las fases subsecuentes estarán sujetas a la capacitación previa en los temas referentes a cada una de ellas.

### **Fases**

#### **Inicio**

##### *Refinar el alcance y el documento de Visión*

Teniendo en cuenta los diferentes temas tratados a lo largo de la capacitación sobre los fundamentos del RUP se revisará el documento de visión del proyecto AQUASOFT con el fin de generar una nueva versión del mismo, por otra parte se buscará asesoría especializada en el tema de identificación y gestión de riesgos con el fin de ganar conocimiento en esta área la cuál hasta ahora no había sido tratada formalmente al interior de la empresa.

#### **Modelado del Negocio**

Se generará una versión mejorada del Glosario de Términos.

### **Artefactos**

Los siguientes artefactos serán producidos en este flujo de trabajo.

<b>Artefacto</b>	<b>Herramientas utilizadas</b>	<b>Entrega formal</b>
Glosario	Microsoft® Word®	Sí

#### **Artefactos de entrada**

Ninguno.

#### **Flujo de Trabajo**

## Requerimientos

Teniendo en cuenta el resultado del taller sobre el uso de los diagramas de casos de uso, y en vista de la no utilización en ninguno de los documentos que previamente había realizado la empresa referentes a la especificación de los casos de uso del artefacto ACTOR, y habiendo notado la importancia de su inclusión en dichos documentos se hará una revisión de los mismos al tiempo que se generarán las nuevas versiones incluyendo a los actores.

De igual manera se harán las distinciones entre los diferentes componentes del sistema a medida que éstos interactúen con los actores, ya que hasta el momento sólo se utilizaba la palabra “sistema”.

## Artefactos

Los siguientes artefactos se producirán durante este flujo de trabajo:

Artefacto	Herramientas utilizadas	Entrega Formal?
Visión	Microsoft Word	Sí
Actores	JDeveloper	No
Glosario	Microsoft Word	Sí
Especificaciones suplementarias	Microsoft Word	Sí
Casos de Uso (actualizados)	JDeveloper, MS Word	Sí
Prototipo de interfaz de usuario	Microsoft® PowerPoint®, Microsoft Word	Sí

## Análisis y Diseño

Con base en la versión actualizada de los casos de uso se revisarán los diseños existentes a fin de implementar los cambios a que hubiese lugar.

## Artefactos

Los siguientes artefactos se producirán durante este flujo de trabajo

Artefacto	Herramientas utilizadas	Entrega Formal?
-----------	-------------------------	-----------------

Modelo de Datos	Oracle Designer	Sí
Diseño de Base de Datos	Oracle Designer	Sí
Documento de Arquitectura del Software (actualizado)	Microsoft Word	Sí

## Artefactos de Entrada

Artefacto	Herramientas Utilizadas	Entrega Formal?
Glosario	Microsoft Word	Sí
Especificaciones Suplementarias	Microsoft Word	Sí
Casos de uso (actualizados)	JDeveloper, Microsoft Word	Sí
Prototipo de Interfaz de Usuario	Microsoft PowerPoint, Microsoft Word	Sí

## Gerencia del Proyecto

### Artefactos

Los siguientes artefactos se producirán durante este flujo de trabajo:

Artefacto	Herramientas Utilizadas	Entrega Formal?
Listado de Riesgos	Microsoft Word	No
Plan de Desarrollo de Sw.	Microsoft Word	Sí
Plan de Iteración	Microsoft Word	Sí

### Artefactos de Entrada

Artefacto	Herramientas Utilizadas	Entrega Formal?
Visión	Microsoft Word	Sí
Caso de Desarrollo	Microsoft Word	Sí