

**Universidad de las Ciencias Informáticas**

**Facultad 15**



**“Análisis y Diseño para la Automatización de las fichas generadas por la Metodología TOT”**

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

**Autor(es):** Marliés Pérez Torres  
Yoriangel Rivero González

**Tutor(es):** Ing. Leydisbel Jaime Robaina  
Msc. Mariano Flores López

**Ciudad de la Habana, junio de 2010**

# DECLARACIÓN DE AUTORÍA

---

## **Declaración de autoría:**

Declaramos que somos los únicos autores del trabajo “Análisis y Diseño para la Automatización de las fichas generadas por la Metodología TOT” y autorizamos al Centro para la Informatización de Gestión de Entidades (CEIGE) de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Marliés Pérez Torres

Autor

---

Yoriangel Rivero González

Autor

---

Msc. Mariano Flores López

Tutor

---

Ing. Leydisbel Jaime Robaina

Tutor



*...Dedicatoria del Trabajo de Diploma...*

*De Yoriangel:*

*...A mi abuela querida...*

*...A mis padres, en especial a mi mamá...*

*...A Parra...*

*...A mi hermano y a mi pequeña hermanita...*

*...A mi novia...*

*De Marliés:*

*...Por su amor incondicional,*

*a los mejores papás del mundo: Héctor y Anita,*

*convencida de que no es nada*

*comparado con lo que se merecen...*

## AGRADECIMIENTOS

---

### **Quiero agradecer:**

*...A toda mi familia, por estar pendiente de mí todo el tiempo...*

*...A mi papá, mi alma gemela, por ser el padre maravilloso que es...*

*...A mi mamá, por quererme tanto, por alarme las orejas cuando lo merezco...*

*...A mi hermano Maiquel, por demostrarme que uno puede llegar tan lejos como se proponga...*

*...A mi hermano Yanel, por las maldades, los apodos, porque trasmite felicidad y alegría...*

*... A Hirán, por los buenos momentos que hemos vivido juntos: “siempre me quedará...”*

*...A tía Maricela, Humberto, La Negra y Héctor, por acogerme en sus casas como a una hija...*

*...A Tika, Tamy, Tahy, Sury, Tai, Any y Nere, porque los amigos nos hacen la vida más bella...*

*... A Maire y Wen, por estar siempre a mi lado los cinco años en las buenas y en las malas...*

*... A Rafelito, por aguantar y perdonar mis malacrianzas...*

*... A Iván, por ser un magnífico amigo y profesor...*

*...A Yoriangel, por ser mi compañero de tesis, por trabajar junto a mí...*

*...A los tutores Leydi y Mariano, por guiarnos y ayudarnos en la última tarea como estudiantes...*

*...A Fidel y la Revolución, por hacer posible que me convierta en ingeniera...*

***...Marliés...***

## AGRADECIMIENTOS

---

### **Quiero agradecer:**

*... A todos los que de una forma u otra me han ayudado a ser lo que soy ahora...*

*... A la tutora Leydisbel por su apoyo y por la ayuda brindada, y a los profesores de la facultad que me ayudaron en mi formación profesional...*

*... Agradezco también a mi compañera de tesis, Marliés, por las tantas horas de trabajo consagrado a la realización de este estudio, y por las horas de sueño que le robé...*

*... A las amistades que aquí encontré...*

*...A mi familia, de manera especial a mi mamá, que tanto me ha apoyado y ha estado conmigo siempre en cualquier situación, mi razón de ser, mi vida entera...*

*... A Parra, que más que amigo ha sido un hermano, y cuyos consejos fueron decisivos a la hora de tomar una decisión correcta...*

*...A mi novia, para la cual no encuentro palabras que puedan expresar mi infinita gratitud para con ella, quien tanto me ha ayudado, tolerado, malcriado, consentido y querido...*

*... Y por último, pero no menos importante, a mi abuela del alma, cuyos días de vida terminaron un triste día de enero –el peor de todos-, y quien fue el empujón principal, el combustible del alma que me hizo querer ser mejor, quien más me apoyó y a quien le debo en mayor medida ser lo que ahora soy...*

**...Yoriangel...**

# RESUMEN

---

## **Resumen**

En el mundo lo único constante es el cambio en todas las esferas, una de las más cambiantes es la economía actual, por tal motivo surge la necesidad de mejorar los procesos de negocios de las compañías para reducir costos y tiempos de ciclo, mejorar la calidad de productos y servicios, flexibilizar los procesos para agilizar el negocio e innovar más que la competencia utilizando menos recursos. Sin una adecuada gestión de los procesos de negocios es imposible lograr estos objetivos. Con este fin de analizar y mejorar los procesos de negocio surge BPM con gran auge en los últimos años. En Cuba se intenta implantar la filosofía de trabajo de BPM en algunas entidades, para facilitar esta tarea se utiliza la metodología TOT para definir procesos claros y fáciles de utilizar. Surgida en nuestro país, es reciente y novedosa, por ese motivo todavía no cuenta con herramientas que faciliten el trabajo con la misma. El presente trabajo de diploma comprende el análisis y diseño para una herramienta que gestione y modele los procesos de negocio que se describen en las fichas generadas por la Metodología TOT.

## **Palabras claves:**

Proceso, Proceso de negocio, Modelado de procesos de negocio, Metodología TOT.

## Tabla de contenido

INTRODUCCIÓN .....	1
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....</b>	<b>5</b>
1.2    Conceptos básicos asociados al dominio del problema .....	5
1.3    Metodología TOT. ....	5
1.4    Tecnologías para la gestión de procesos de negocio.....	8
1.4.1    Gestión de Procesos de Negocio .....	8
1.4.2    Notación para el Modelado de Procesos de Negocio .....	9
1.4.3    Lenguaje de Definición de Proceso de Negocio de XML .....	10
1.4.4    Lenguaje de Ejecución de Procesos de Negocio con Servicios Web.....	13
1.4.5    Lenguaje de Definición de Procesos de Negocio de XML vs Lenguaje de Ejecución de Procesos de Negocio con Servicios Web .....	14
1.4.6    Sistemas de Administración de Procesos de Negocio .....	15
1.5    Estado del arte de los Sistemas de Administración de Procesos de Negocio.....	16
1.5.1    Intalio.....	16
1.5.2    Adonis .....	17
1.5.3    JBoss jBPM .....	17
1.5.4    Enhydra Shark .....	17
1.5.5    TIBCO .....	17
1.5.6    Conclusiones sobre los Sistemas de Gestión de Procesos de Negocio .....	18
1.6    Herramientas utilizadas.....	18
1.6.1    Lenguaje Unificado de Modelación (UML) .....	18
1.6.2    Herramienta Case a utilizar: Visual Paradigm .....	19
1.7    Metodología de desarrollo de software utilizada.....	19
1.7.1    Modelo de desarrollo del Centro para la Informatización de Gestión de Entidades (CEIGE).....	20
1.8    Conclusiones parciales .....	21
<b>CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....</b>	<b>22</b>
2.1    Introducción .....	22



# ÍNDICE DE CONTENIDO

---

<b>2.2</b>	<b>Modelo conceptual .....</b>	<b>22</b>
<b>2.3</b>	<b>Requisitos .....</b>	<b>23</b>
2.3.1	Técnicas empleadas para la captura de los requisitos .....	24
2.3.2	Requisitos funcionales .....	25
2.3.3	Especificaciones de requisitos y prototipos de interfaz de usuario .....	28
2.3.4	Validación de los requisitos funcionales .....	36
2.3.5	Conclusiones parciales .....	39
<b>CAPÍTULO 3: DISEÑO Y VALIDACIÓN DE LA SOLUCIÓN .....</b>		<b>40</b>
<b>3.1</b>	<b>Introducción .....</b>	<b>40</b>
<b>3.2</b>	<b>Diagrama de clases .....</b>	<b>40</b>
<b>3.3</b>	<b>Descripción de la arquitectura .....</b>	<b>42</b>
<b>3.4</b>	<b>Tecnologías y herramientas de desarrollo propuestas para la solución .....</b>	<b>44</b>
3.4.1	Software libre .....	44
3.4.2	Lenguaje de programación Java .....	45
3.4.3	Entorno de Desarrollo Integrado NetBeans 6.8 .....	45
3.4.4	Sistema de gestión de Base de Datos: PostgreSQL .....	46
<b>3.5</b>	<b>Diagrama de componentes .....</b>	<b>47</b>
<b>3.6</b>	<b>Diagramas de clases del diseño .....</b>	<b>47</b>
<b>3.7</b>	<b>Modelado de la base de datos .....</b>	<b>51</b>
<b>3.8</b>	<b>Patrones de diseño empleados en la solución .....</b>	<b>54</b>
<b>3.9</b>	<b>Resultados de aplicar las métricas del diseño .....</b>	<b>56</b>
<b>3.10</b>	<b>Conclusiones parciales .....</b>	<b>59</b>
<b>CONCLUSIONES GENERALES .....</b>		<b>60</b>
<b>RECOMENDACIONES .....</b>		<b>61</b>
<b>TRABAJOS CITADOS .....</b>		<b>62</b>
<b>BIBLIOGRAFÍA .....</b>		<b>64</b>

# ÍNDICE DE FIGURAS

---

## Índice de figuras

Figura 1 Prototipo de interfaz de usuario para Adicionar proceso.....	31
Figura 2 Prototipo de interfaz de usuario para Adicionar proceso.....	31
Figura 3 Prototipo de interfaz de usuario para Modificar proceso .....	34
Figura 4 Prototipo de interfaz de usuario para Modificar proceso .....	34
Figura 5 Prototipo de interfaz de usuario para Eliminar proceso.....	36
Figura 9 Estructura de la arquitectura .....	44
Figura 9 Diagrama de clase del diseño para Gestionar mejora .....	55
Figura 10 Cantidad de procedimientos .....	57
Figura 11 Responsabilidad por clases .....	57
Figura 12 Complejidad de implementación de las clases .....	58
Figura 13 Nivel de reutilización de las clases .....	58

## Índice de Tablas

Tabla 1 Conclusiones sobre BPMS .....	18
Tabla 2 Especificación del requisito Adicionar proceso .....	29
Tabla 3 Especificación del requisito Modificar proceso .....	32
Tabla 4 Especificación del requisito Eliminar Proceso .....	35
Tabla 6 Métricas aplicadas a la especificación de requisitos .....	38
Tabla 7 Resultado de la métricas aplicadas a la especificación de requisitos .....	38
Tabla 8 Descripción del diseño de clases de Gestionar proceso .....	50

## Introducción

En la actualidad las organizaciones se encuentran inmersas en entornos y mercados competitivos y globalizados; donde quienes deseen tener éxito necesitan alcanzar buenos resultados. Para lograrlo, deben gestionar sus actividades y recursos, lo que conlleva a la utilización de herramientas y metodologías para establecer, documentar y mantener un Sistema de Gestión. Con esta finalidad, se utilizan modelos o normas de referencias reconocidos, destacándose la familia ISO 9000 que se sustenta sobre ocho Principios de Gestión de la Calidad donde uno de estos es el "enfoque basado en procesos" el cual sostiene que *"un resultado se alcanza más eficientemente cuando las actividades y los recursos se gestionan como un proceso"* (1).

Con el objetivo de eliminar los problemas de los sistemas empresariales tradicionales surge el término Business Process Management o BPM, también conocido como Business Process Reengineering (BPR), el cual se refiere a las actividades que desarrollan las empresas para optimizar y adaptar sus procesos; en nuestro país su aplicación trajo consigo la necesidad de adoptar nuevas prácticas para el modelado y mejora de los procesos de negocio utilizándose para esto la Metodología TOT. Esta metodología está basada en modelos orientados al cliente en lenguaje natural y persigue la definición de un proceso claro y fácil de utilizar en la mejora del proceso de Modelado del Negocio, utiliza el paradigma del estudio de los procesos Dirigidos por Modelos y la inserción de una herramienta de gestión de procesos de negocios (BPMS). Como producto de aplicar la metodología TOT se generan tres artefactos (2):

1. Ficha de Captura de la Información Primaria (FCIP)
2. Ficha de Modelación Procesos (FMP). Ver **¡Error! No se encuentra el origen de la referencia..**
3. Ficha de Aprobación de Mejoras (FAM)

Estas fichas existen actualmente en documentos Word (.doc o .docx), y la información que en ellas se recoge debe ser luego modelada en una de las herramientas de Modelación de Procesos de Negocio. Esta tarea de traducir la información recogida en las fichas a los elementos gráficos de la herramienta resulta tediosa, además de que toma bastante tiempo cuando se trata de un volumen considerable de información, por esta razón se impone la necesidad de agilizar esta tarea mediante su automatización. Puesto que la metodología es relativamente reciente, novedosa y solo existe en nuestro país, aún no cuenta con una herramienta que modele los procesos de negocio a partir de la información de las fichas

# INTRODUCCIÓN

---

que se obtienen como resultado de aplicarla. Ante la situación existente, se impone el siguiente **problema científico**:

La información que se recoge en las fichas generadas por la Metodología TOT no se gestiona, además el proceso de generación del modelo de procesos a partir de las fichas es lento e ineficiente.

De ahí que el **objeto de investigación** se centra en: La gestión y modelado de los procesos de negocio.

Para dar solución al problema planteado anteriormente se ha trazado como **objetivo general** de la investigación: Realizar el análisis y diseño para una herramienta que permita gestionar la información que se recoge en las fichas generadas por la Metodología TOT y realizar el modelado de los procesos de negocio automáticamente.

Para lo cual se define el siguiente **campo de acción**: La gestión y el modelado de los procesos de negocio a partir de la información obtenida en las fichas generadas por la metodología TOT.

## **Objetivos específicos:**

1. Estudiar profundamente la Metodología TOT.
2. Investigar cuáles herramientas BPMS importan código en el Lenguaje de Definición de Procesos de negocio de XML (XPDL)
3. Definir todos los elementos de la Notación para el Modelado de Procesos de Negocio (BPMN) y su representación en el Lenguaje de Definición de Procesos de negocio de XML (XPDL)
4. Realizar el análisis y el diseño para la herramienta.

Para dar cumplimiento a todos los objetivos se proponen las siguientes **tareas de la investigación**:

1. Estudiar acerca de la metodología TOT, sus objetivos, principios, características, fases, trabajadores y artefactos. También sobre BPM, BPMN, BPMS y modelado de negocio.
2. Elaborar el estado del arte referente a la metodología TOT y las herramientas de modelado existentes actualmente.
3. Aprender el lenguaje XPDL: características, elementos, sintaxis, etc.
4. Realizar la descripción de los requisitos funcionales.
5. Definir los prototipos de interfaz de usuario.

# INTRODUCCIÓN

---

6. Realizar los diagramas de clases.
7. Diseñar la Base de Datos.

**Idea a defender:** El análisis y diseño de una herramienta para la gestión de la información que se recoge en las fichas generadas por la metodología TOT y que realice el modelado de los procesos automáticamente permitirá su posterior desarrollo.

Con la finalidad de llevar a cabo las tareas planteadas se emplearon métodos de investigación empíricos y teóricos, de modo tal que se pueda maximizar la validez y confiabilidad de la información, reducir los errores en los resultados y adquirir la solidez de conocimientos necesarios para el desarrollo de la herramienta propuesta.

## **Métodos teóricos:**

- **Histórico:** El estudio de la evolución en la gestión empresarial y el desarrollo de software demuestran la necesidad de utilizar sistemas informáticos con el fin de apoyar y facilitar todo el proceso de modelado y mejora de procesos de negocio.
- **Lógicos:**
  - ✓ **Hipotético deductivo:** A partir de la consecutiva aparición de problemas en el modelado de los procesos basados fundamentalmente en la traducción de la información empírica obtenida al diseño de los modelos de procesos se puede deducir la necesidad de traducir directamente esta información en los modelos deseados.
  - ✓ **Modelación:** El modelado de los procesos de negocio permite visualizar el comportamiento de un sistema, donde un proceso puede ser evaluado y corregido y además mostrar si un sistema dado es técnicamente factible. Además los requisitos del sistema serán mostrados previamente a través de prototipos no funcionales que permitirán mostrar cómo quedará el producto final.

## **Métodos empíricos:**

- **Generales**

# INTRODUCCIÓN

---

- ✓ **Observación:** Se observa el hecho de que el modelado de los procesos de negocio a partir de las fichas generadas por la metodología es un proceso muy lento y tedioso, y mucho más cuando en la entidad que se modela existen una cantidad considerable de mega procesos, procesos y subprocesos, puesto que para cada uno es necesaria la aplicación de monitoreo y mejoras.
  - ✓ **Experimental:** La realización de un experimento donde a partir de una situación real se ponga en funcionamiento el sistema con el objetivo de determinar las diferencias con resultados anteriores y además poder corregir cualquier ineficiencia del mismo nos permitirá comprobar la validez de la idea a defender.
- **Particulares:**
    - ✓ **Entrevista:** Permite realizar conversaciones planificadas con expertos funcionales, consultores, analistas de sistema. Se utiliza para obtener información necesaria para la especificación de los requisitos del sistema.

## **Resultados esperados:**

1. Obtener el análisis y diseño de una herramienta que permita gestionar y modelar de los procesos de negocio de una organización a partir de la información que se recoge en las fichas generadas por la metodología TOT.
2. Facilitar la implementación de la herramienta para la elaboración de modelos de procesos y transformación de los mismos.
3. Brindar un mayor nivel de gestión y control de los procesos de negocio de una organización determinada.

## **Estructura del trabajo de diploma:**

El trabajo de diploma está compuesto por 3 capítulos:

- Capítulo 1: Fundamentación Teórica.
- Capítulo 2: Descripción de la solución propuesta.
- Capítulo 3: Diseño y validación de la solución.

## Capítulo 1: Fundamentación Teórica

### 1.1 Introducción

El objetivo de este capítulo es explicar y profundizar en los conceptos fundamentales que están asociados al problema, haciendo énfasis en su historia, evolución y estado del arte, elementos indispensables para la comprensión de los capítulos posteriores. Justifica la búsqueda de una correcta solución al problema que se planteó en la Introducción dando paso al cumplimiento del objetivo principal de nuestra investigación científica.

### 1.2 Conceptos básicos asociados al dominio del problema

**Proceso:** conjunto de actividades relacionadas o que interactúan las cuales transforman elementos de entradas en resultados.

**Proceso de negocio:** conjunto de tareas relacionadas lógicamente para lograr un resultado de negocio definido, definen entradas, funciones y salidas.

**Modelado de procesos de negocio:** representación de un grupo de actividades que una persona o empresa efectúan para alcanzar una meta.

**Metodología TOT:** metodología basada en modelos orientados al cliente en lenguaje natural. Persigue la definición de un proceso claro y fácil de utilizar para la organización de la mejora de proceso de Modelado del Negocio y Gestión de Requerimientos.

### 1.3 Metodología TOT.

Está basada en modelos orientados al cliente en lenguaje natural y por esta característica lleva el nombre del dios griego creador de las palabras y el lenguaje natural: Tot. Se encuentra dentro del campo de estudio de gestión de la complejidad empresarial. Tiene como propósito el análisis y mejora de procesos empresariales; pero como sus orígenes están enmarcados en el mundo de la informática, siendo utilizada desde sus inicios para la gestión de proyectos, se incluye dentro de los flujos de trabajo Modelado de Negocio y Gestión de Requisitos para el desarrollo de software. (2)

**Tiene cuatro características fundamentales:**



# CAPÍTULO 1

---

1. Uso del lenguaje natural.
2. Centrada en el conocimiento de técnicas de descripción de procesos y de comunicación con el cliente.
3. Orientada para que el desarrollo del software sea dirigido por caso de uso.
4. Iterativa e incremental.

## **Y está regida por cuatro principios básicos:**

1. Separa el espacio del problema del espacio de la solución donde El modelado de Negocios es el espacio del problema y La Ingeniería de Requisitos es el espacio de la solución, enfocándose al estudio del primero.
2. Orientada como una actividad previa y complementaria de la Ingeniería de Requisitos.
3. Promueve el estudio y reingeniería de los procesos como fase inicial y obligatoria para la optimización de los procesos del negocio.
4. La optimización de los procesos es la que justifica la inversión en Tecnologías de Información.

## **Fases de la Metodología TOT. Ver ¡Error! No se encuentra el origen de la referencia..**

Contiene fases marcadas por hitos, las cuales se fusionan entre sí y con la metodología que se elija, para el posterior progreso del software, pudiendo moverse tanto hacia delante como hacia atrás en el desarrollo del producto, de modo que se pueda comprobar el trabajo realizado y se puedan efectuar correcciones, estas fases son:

- Iniciación
- Diagnóstico
- Mejorar el proceso
- Implementación

## **Trabajadores de TOT:**

- Principales:
  - ✓ Propietario del proceso
  - ✓ Analista del proceso
  - ✓ Experto funcional
- De apoyo:

- ✓ Diseñador de UI
- ✓ Especificador
- ✓ Arquitecto
- ✓ Especialista de calidad
- ✓ Especialista en pruebas

## La aplicación de la metodología TOT genera tres artefactos significativos:

1. **Ficha de Captura de la Información Primaria (FCIP):** La FCIP fue creada con el objetivo de recopilar la mayor cantidad de información posible mediante preguntas orientativas al cliente, está dirigida a cualquier persona dentro del proceso que tenga conocimientos del mismo. Es la primera herramienta que se utiliza durante la aplicación de la metodología, la información que aquí se recoge es la entrada principal para la FMP. Con el cliente se realizan entre 2 ó 3 reuniones para ir conformando la ficha, si la información aquí descrita cambia, también cambia en las demás fichas. (3)
2. **Ficha de Modelación de Proceso (FMP):** La FMP fue creada con el objetivo de desglosar de forma clara los diferentes componentes por los que está formado, refiriéndose a componentes, a todos los procesos, subprocesos, eventos y actividades que intervienen en el macroproceso, así como el nivel de importancia que tiene el mismo. (4). Ver **¡Error! No se encuentra el origen de la referencia..**
3. **Ficha de Aprobación de Mejoras (FAM):** La Ficha de Aprobación de Mejoras refleja los procesos de la organización mejorados y optimizados listos para su futura implementación. Además es aquí donde el cliente se manifiesta de acuerdo o en desacuerdo con el trabajo realizado. (5)

## Ventajas de usar esta metodología: (2)

- Utiliza el lenguaje natural creando una mejor comunicación entre el cliente y el equipo de desarrollo.
- El cliente y el equipo de desarrollo trabajan en conjunto.
- Proporciona una mejora de procesos para lo cual realiza un estudio previo de los mismos, analizando la forma de hacerlos más óptimos. Después de estudiar su comportamiento y detectar los puntos débiles busca la forma de eliminarlos o disminuirlos.

- Obliga a que no se pueda pasar al espacio de la solución sin haber analizado antes el espacio del problema.
- Gana en agilidad, minimizando el riesgo de requisitos mal definidos.

## 1.4 Tecnologías para la gestión de procesos de negocio

### 1.4.1 Gestión de Procesos de Negocio

Toda entidad tiene múltiples procesos de negocio, del grado de eficiencia de estos depende su éxito. Con el fin de adaptar y modificar los procesos a medida que una organización crece para hacer frente a las nuevas necesidades del negocio surge el término Gestión de Procesos de Negocio, más conocido como BPM de su nombre en inglés Business Process Management (BPM). La gestión de procesos de negocio a través de flujos, lo que se suele denominar workflow (6). BPM trata los procesos como activos que contribuyen directamente al desempeño de la empresa llevándolos a la excelencia operacional y agilidad del negocio; permite analizar los flujos de trabajo existentes y optimizarlos, para proveer productos y servicios a los clientes de la manera más eficiente; a través de BPM se puede disponer de una visión integral de los procesos con el fin de mejorar el rendimiento del negocio.

Algunas aplicaciones de BPM o motivos de implantación pueden ser: (6)

- Implantación o mejora del sistema de calidad
- Comprensión y mejora de procesos
- Reingeniería de procesos
- Optimización de tareas
- Establecimiento de metodología a nivel de empresa
- Control y seguimiento de los procesos organizativos

La finalidad de un BPM es descomponer la actividad global de una empresa u organización en un conjunto de procesos para que puedan ser analizados con detalles y cuyas acciones repetitivas puedan ser automatizadas, tanto en lo concerniente a los sistemas como a las personas que intervienen, para optimizar tiempo, oportunidades y costes, sin perder la capacidad de adaptación constante y conservando la coexistencia de métodos seguros para facilitar la intervención activa y fundamental de las personas en los procesos.

Entre los beneficios que aporta a una organización la implantación de la filosofía de trabajo BPM están:

- Reducción de plazos en los procesos de soporte al negocio.
- Optimización de costes.
- Integridad y calidad de procesos.
- Integración de terceras partes en los procesos.

## 1.4.2 Notación para el Modelado de Procesos de Negocio

BPM engloba numerosas actitudes y tareas a desarrollar en una empresa, una de sus técnicas principales es la gestión de procesos de negocio a través de flujos o *workflow*, como es conocido también. Mediante un *workflow* se estudian los aspectos operacionales de una actividad de negocio: cómo se estructuran las tareas, cómo se realizan, cuál es su orden, cómo se sincronizan y la manera en que se relacionan unas con otras. Existe una notación gráfica estandarizada que permite el modelado de procesos de negocio en un formato de flujo de trabajo denominada Notación para el Modelado de Procesos de Negocio o BPMN, siglas por su nombre en inglés Business Process Modeling Notation.

En síntesis tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación. Actualmente hay una amplia variedad de lenguajes, herramientas y metodologías para el modelado de procesos de negocio (7). La cada vez mayor adopción de la notación BPMN como estándar ha ayudado a unificar la expresión de conceptos básicos de procesos de negocio así como conceptos avanzados de modelado. Está planeada para dar soporte únicamente a aquellos procesos que sean aplicables a procesos de negocios, esto significa que cualquier otro tipo de modelado realizado por una organización con fines distintos a los del negocio no estará en el ámbito de BPMN. El modelado se realiza mediante diagramas muy simples con un conjunto muy pequeño de elementos gráficos, con esto se busca que los usuarios del negocio y los desarrolladores técnicos entiendan fácilmente el flujo y el proceso.

Las cuatro categorías básicas de elementos son (6): Ver **¡Error! No se encuentra el origen de la referencia..**

- **Objetos de flujo:** Eventos, Actividades, Rombos de control de flujo (*Gateway*).
- **Objetos de conexión:** Flujo de secuencia, Flujo de mensaje, Asociación.

- **Carriles de piscina** (*Swimlanes*): *Pool, Lane*.
- **Artefactos**: Objetos de datos, Grupo, Anotación.

Estas cuatro categorías de elementos nos dan la oportunidad de realizar un diagrama simple de procesos de negocio, BPD de sus siglas en inglés (*Business Process Diagram*), en un BPD se permite definir un tipo personalizado de Objeto de flujo o un Artefacto, si con ello se hace el diagrama más comprensible.

BPMN no está pensado para modelar aplicaciones, sino procesos que correrán dentro de ellas, por este motivo su salida necesita ser expresada en algo que no sea un lenguaje programático, es aquí donde entra en juego BPML, encargado de realizar la traducción de un formato gráfico leído por personas a uno entendible por las máquinas y permitir el intercambio entre distintas herramientas. Algunos estándares han surgido en los últimos años como soluciones a esa traducción, dos de los más importantes son el Lenguaje de Definición de Procesos de Negocio de XML (XPDL) y el Lenguaje de Ejecución de Procesos de Negocio con Servicios Web (BPEL).

### **1.4.3 Lenguaje de Definición de Proceso de Negocio de XML**

El Lenguaje de Definición de Procesos de Negocio (XPDL) es un formato de archivo basado en XML que puede ser usado para intercambiar modelos de procesos de negocio entre distintas herramientas. Representa el “dibujo” de la definición del proceso, tiene el tamaño y las coordenadas X e Y del nodo, un concepto de líneas que señalan el camino a seguir, los nodos y las líneas tienen atributos que pueden especificar información ejecutable tales como roles, descripción de actividades, temporizadores, llamadas a *web services*, entre otros (7). El objetivo de XPDL es almacenar e intercambiar el diagrama de procesos. Permite que una herramienta de diseño de procesos escriba un diagrama y otra lo lea, y que en la imagen resultante sean lo más parecidas posibles, sin embargo, no otorga garantía de semántica de ejecución precisa. Intenta ofrecer una manera estándar para representar procesos de tal manera que puedan ser importados y/o exportados por cualquier editor que implemente el estándar. Con XPDL, un producto puede escribir una definición de proceso con total fidelidad, y otro producto puede leer y reproducir el mismo diagrama que fue enviado. XPDL es extensible de forma que permite a cada herramienta diferente almacenar la información de la aplicación específica en el XPDL, y conserva esos valores incluso cuando son manipulados por herramientas que no leen esas extensiones. Esta es la única manera de proveer un

"viaje a la redonda", utilizando múltiples herramientas y aún así ser capaz de regresar a la herramienta original con total fidelidad.

Para conseguir el propósito de XPDL se define un metamodelo que cubre:

- Las entidades de más alto nivel en el dominio de la definición de procesos.
- Atributos de procesos.
- Agrupaciones de diferentes procesos en modelos relacionados.
- Definiciones de datos comunes que puedan ser usados en variedad de modelos.

Para todos estos aspectos tenemos dos metamodelos principales: (8)

- El metamodelo *Package*, que se encarga de las agrupaciones de procesos, del intercambio de mensajes entre estos y de las diferentes características que poseen los mismos.
- El metamodelo *Process* que describe las principales entidades que componen un proceso así como los atributos de estas.

De las entidades que conforman los dos metamodelos, a continuación se describen brevemente las de más alto nivel, que son precisamente las que tienen relación directa con los elementos de la notación gráfica BPMN: (8)

- **Pool:** Contenedor de actividades y transiciones entre ellas.
- **Lane:** Entidad que permite subdividir un Pool; típicamente en relación a los roles participantes.
- **Process Definition:** Proporciona información contextual que se aplica a una serie de entidades a lo largo de un proceso.
- **Actividad:** Trabajo dentro de un proceso que será desempeñado por una combinación de recursos humanos y computacionales.
- **Task:** Unidades de trabajo que componen una actividad.
- **Event:** Cualquier suceso que ocurre mientras se está ejecutando un proceso y que normalmente afecta al flujo del mismo.
- **Transition:** Paso de una actividad a otra cuando se cumplen determinadas condiciones.
- **Participant:** El que realiza una serie de actividades, ya sea un elemento humano o un elemento computacional.

# CAPÍTULO 1

---

- **Relevant Data:** Los datos que son creados y usados por una instancia de un proceso durante su ejecución.
- **Application:** Elementos computacionales que nos van permitir automatizar, total o parcialmente, una o varias actividades.
- **Artifact:** Elementos del proceso que no pertenecen al conjunto de elementos básicos (actividades, secuencia, flujo de mensajes) y que se relacionan con objetos del flujo (flow objects) mediante asociaciones. Pueden ser (siguiendo la notación BPMN), un Data Object, una Annotation o un Group.
- **Message Flow:** Flujo de mensajes entre dos participantes y/o procesos que están preparados para enviar y recibir información.
- **Association:** Para asociar un Artifact con un objeto de flujo (Object Flow) y para mostrar las actividades que se usan para compensar otra actividad.
- **System and Environmental data:** Datos que son mantenidos por el sistema de workflow o el sistema de entorno local y que son accedidos por las actividades para ser usados de la misma manera que los Relevant Data.

Existen muchas herramientas que nos permiten trabajar con XPD, las reconocidas por la propia WfMC son las siguientes (8):

- ADVANTYS.
- Amazonas Workflow.
- BOC ADONIS 3.7.
- CARNOT Process Engine
- CHALEX BPM Framework
- Enhydra Shark
- Enhydra JaWE
- Finantix Studio (FXS)
- Fuego
- Fujitsu Interstage
- HOGA.PL

- IDS Scheer
- Integic e.POWER WorkManager Builder tool
- Interstage Business Process Manager Studio
- ITP-Commerce
- Jenz & Partner GmbH
- Flow Designer, Eclair Group
- Metoda S.p.A
- Tell-Eureka's
- OfficeObjects RWorkFlow
- Open Business Engine
- TIBCO R Staffware Process Suite
- TIBCO R Business Studio
- Together Workflow Editor
- Vignette Process Workflow Modeler
- ZAPLET 3 PROCESS BUILDER

#### **1.4.4 Lenguaje de Ejecución de Procesos de Negocio con Servicios Web**

El desarrollo del Lenguaje de Ejecución de Procesos de Negocio con Servicios Web (BPEL), nace de la necesidad de manejar lenguajes distintos entre la programación a gran escala<sup>1</sup> y la programación detallada<sup>2</sup>, ya que en su esencia ambos tipos de desarrollo requieren de distintos grados de comunicación con otros servicios. Este lenguaje fue concebido por los grandes de la informática como lo son Oracle, BEA Systems, IBM, SAP y Microsoft entre otros. Es un lenguaje de alto nivel que lleva el concepto de servicio un paso adelante al proporcionar métodos de definición y soporte para flujos de trabajo y procesos de negocio. Se desarrolló a partir de WSDL<sup>3</sup> y XLANG<sup>4</sup>, ambos lenguajes basados en XML orientados a la descripción de servicios Web. Básicamente consiste en un lenguaje basado en XML

---

<sup>1</sup> Se refiere al desarrollo de software de gran tamaño que involucran grandes procesos de desarrollo, evolución y mantenimiento.

<sup>2</sup> Se refiere a la construcción de componentes de software pequeño y autónomo.

<sup>3</sup> Web Services Description Language es un formato XML que se utiliza para describir servicios Web.

<sup>4</sup> Es una extensión basada en XML de Web Services Description Language.



diseñado para el control centralizado de la invocación de diferentes servicios Web, con cierta lógica de negocio añadida que ayudan a la programación en gran escala. El enfoque sobre procesos de negocios modernos más el bagaje de los lenguajes WSDL y XLANG, guiaron a BPEL a adoptar los servicios Web como su mecanismo de comunicación externa. Así las facilidades de mensajería BPEL dependen del uso del WSDL para describir los mensajes entrantes y salientes. (9)

BPEL es un lenguaje de orquestación<sup>5</sup>, no un lenguaje coreográfico<sup>6</sup>, las diferencias entre orquestación y coreografía están basadas en analogías: la orquestación describe un control central del comportamiento como un director de orquesta, mientras que la coreografía trata sobre el control distribuido del comportamiento donde participantes individuales realizan procesos basados en eventos externos, como en una danza coreográfica donde los bailarines reaccionan a los comportamientos de sus pares. Cuando un flujo de negocio se compone de muchos procesos internos que se lanzarían dependiendo de una serie de valores y respuestas, BPEL sería el encargado de orquestar todo el proceso ordenando qué proceso ejecutar (servicio Web) y en qué momento.

Una de sus limitaciones es que para soportar interacciones humanas necesita de añadidos a la especificación, lo cual dificulta su portabilidad ya que suelen ser implementaciones propietarias de cada proveedor.

## **1.4.5 Lenguaje de Definición de Procesos de Negocio de XML vs Lenguaje de Ejecución de Procesos de Negocio con Servicios Web**

BPEL y XPDL no son competidores entre sí, ya que pueden ser complementarios según el uso que se haga de ellos. A partir de un modelo BPMN podemos pasar a un proceso BPEL ejecutable por un motor de una herramienta, igualmente a partir de un proceso BPMN podemos obtener una implementación XPDL, desde la versión XPDL 2.0 esto es inmediato en ambos sentidos que puede ser ejecutada por algunos motores de forma nativa. La desventaja del paso de BPMN a BPEL es que no siempre se puede hacer el camino inverso lo que dificulta la exportación y/o importación entre herramientas. BPEL y XPDL están pensados para cosas distintas pero pueden ser complementarias si se utilizan de esa manera. XPDL permite que una herramienta de diseño de procesos escriba un diagrama y otra lo lea, y que en la

---

<sup>5</sup> Provee un ámbito específicamente enfocado en la vista de un participante en particular.

<sup>6</sup> Abarca todos los participantes y sus interacciones asociadas, dando una vista global del sistema.

imagen resultante sean lo más parecidas posibles; BPEL no define el esquema gráfico, procesos orientados a humanos, subproceso, y muchos otros aspectos de un proceso empresarial moderno: simplemente no se ha definido para llevar el diagrama de proceso de negocio de una herramienta de diseño a otra (9).

## 1.4.6 Sistemas de Administración de Procesos de Negocio

Para lograr los objetivos que se propone BPM y adherirse al paradigma se implanta un sistema en las empresas denominado Sistema de Administración de Procesos de Negocio, BPMS (Business Process Management Systems), también se conoce como BPM Suite o BPM System. La definición más simple de un BPMS es "Conjunto de herramientas y servicios diseñados para administrar procesos de negocios BPM en todo su ciclo de vida: Análisis, diseño, simulación, implementación, monitoreo y optimización" (10), estas herramientas son indispensables para implantar BPM.

Un BPMS está compuesto por seis elementos principales (10):

1. **Motores de Orquestación:** permiten coordinar la secuencia de actividades según los flujos y reglas del modelo de negocio.
2. **Herramientas de análisis y Business Intelligence:** permiten analizar la información producto del análisis del proceso.
3. **Motores de regla:** ejecuta reglas que permiten abstraer las políticas y decisiones.
4. **Repositorios:** mantiene los componentes y recursos de los procesos disponibles para su reutilización.
5. **Herramientas de simulación y optimización:** permite comparar los nuevos diseños de procesos con el desempeño actual.
6. **Herramientas de integración:** permiten integrar el modelo con otros sistemas.

De forma general sus beneficios, tanto tangibles como intangibles son numerosos, resumiendo los de mayor importancia tenemos:

- Mejora la atención y servicio al cliente.
- Incrementa el número de actividades ejecutadas en paralelo.
- Minimiza el tiempo requerido por los participantes para acceder a la documentación, aplicaciones y bases de datos.

- Disminuye el tiempo de transferencia de trabajo, información y documentos entre actividades.
- Asegura la continua participación y colaboración de todo el personal en el proceso.
- Disminuye el tiempo que los participantes, supervisores y administradores necesitan para conocer la situación de un ítem de trabajo (P.ej.: Orden de compra, participación de siniestro, pedido de cliente).
- Simplificación de salidas automáticas. Documentos Word, Faxes, e-mails, mensajes cortos a móviles, etc.
- Disponibilidad de mecanismos para una mejor gestión y optimización de procesos.

## 1.5 Estado del arte de los Sistemas de Administración de Procesos de Negocio

En el estudio del estado del arte de los BPMS se consideran fundamentalmente tres aspectos que se consideran más importantes por el tipo de herramienta que se necesita:

1. **Licencia del software** centrándonos principalmente en las herramientas de distribución libre de código abierto.
2. **Las herramientas BPMS que importan/exportan código XPD**L puesto que es el lenguaje que se utilizará para la representación de los elementos que conforman el “dibujo” de la representación del proceso de negocio.
3. **Modelador Gráfico de Procesos** con Diseñador Gráfico de procesos que permita fácilmente crear e interpretar el modelado de los procesos de negocio.

### 1.5.1 Intalio

Intalio es un software Open Source basado en Java-J2EE, que implementa BPMS. Utiliza la notación para diseñar procesos de negocio establecida por el BPMN que puede adaptarse a los requisitos de las Arquitectura Orientada a Servicio (SOA). El Diseñador de Intalio (Intalio|Designer) es una herramienta simple tanto para analistas del negocio, ingenieros de software y administradores de sistemas como para ayudar el modelado de procesos de negocio. Su relación con sistemas externos e interfaces de usuario y su despliegue en el Intalio|Server facilitan que todos los stakeholders se beneficien teniendo un ambiente de trabajo común que fomenta la comunicación entre disciplinas y preserva la integridad de los procesos

de negocios a través de su ciclo de vida completo, desde el diseño hasta la implementación y optimización. Intalio es uno de los mejores BPMS que existen actualmente, pero no importa código XPD (10)

## 1.5.2 Adonis

Adonis es un software libre que emerge como una herramienta empresarial para la gestión de los procesos de negocio. Ofrece integración con herramientas de desarrollo de aplicaciones basadas en estándares como XMI y BPEL, pero no con XPD. En ADONIS Community Edition la cobertura funcional de las herramientas básicas de dibujo no son suficientes para establecer un sistema de gestión cuyo objetivo sea promover la mejora continua en la implantación de la gestión por procesos.

## 1.5.3 JBoss jBPM

JBoss jBPM es un software libre para la plataforma Java que permite diseñar procesos de negocio de manera gráfica con Eclipse e integrarlos en aplicaciones Java, también incluye una herramienta gráfica de diseño; el modelador de jBPM es una herramienta gráfica para crear los procesos de negocio el cual se incorpora como un plugin de eclipse, la cual es una instalación independiente de la herramienta de diseño. La característica más importante de la herramienta gráfica de modelado es que incluye soporte tanto para las tareas del analista de negocios como para el desarrollador técnico. Esto permite una transición armónica desde la modelación de procesos de negocio a la implementación práctica. El plugin incorpora por defecto la notación jPDL para modelar los procesos de negocio, teniendo rasgos similares a la notación BPMN. Análogamente se puede extender la extensión BPEL que viene por separado y que permite que jBPM soporte BPEL, pero no XPD.

## 1.5.4 Enhydra Shark

Es un software libre que posee una herramienta de definición de procesos llamada *Together Workflow Editor* (TWE) o *JaWE*; el editor utiliza XPD 1.0 como lenguaje de procesos. La definición de los procesos está hecha en XML. *Enhydra Shark* provee una notación gráfica de desarrollo donde un proceso es visualizado como un pool en la asignación de trabajo a un participante representado por un “swim-lane”.

## 1.5.5 TIBCO

TIBCO Business Studio es una herramienta BMP con la que modelar procesos de negocio siguiendo el estándar BPMN e intercambiando con XPD. Lo más interesante de esta herramienta, a parte de conseguirla gratuitamente, es lo fácil que resulta simular el comportamiento de los procesos BPM para ver cuellos de botella en procesos reales de negocio, dimensionar el personal, definir cupos de atención, etc. es fácil de usar.

## 1.5.6 Conclusiones sobre los Sistemas de Gestión de Procesos de Negocio

En la siguiente tabla se muestran los BPMS que cumplen total o parcialmente con los criterios de búsqueda considerados para el estado del arte. Se concluye que del resultado obtenido se puede utilizar Enhydra Shark y TIBCO, debido a que ambos tienen la capacidad de importar/exportar código XPD, es decir, estas herramientas son capaces de leer un código XPD y a partir de la información que estos ficheros .xpd contienen representar el dibujo de los procesos de negocio.

**Tabla 1 Conclusiones sobre BPMS**

BPMS	Software libre	Importa/exporta XPD	Modelador gráfico.
Intalio	x	–	x
Adonis	x	–	x
Jboss jBPM	x	–	x
Enhydra Shark	x	x	x
TIBCO	x	x	x

## 1.6 Herramientas utilizadas

### 1.6.1 Lenguaje Unificado de Modelación (UML)

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimientos sobre los sistemas que se deben construir. Se usa para entender, diseñar, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para emplearse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, y principios

generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. (11)

## **1.6.2 Herramienta Case a utilizar: Visual Paradigm**

El Visual Paradigm es una Suite de herramientas CASE que utiliza “UML” como lenguaje de modelado. Es independiente de plataforma, dotada de una buena cantidad de productos o subsistemas para facilitar el trabajo durante la confección de un software, lo cual garantiza la calidad del producto final. Permite:

- Entorno de creación de diagramas para UML.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Garantiza que el modelo y el código permanezcan sincronizados en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad en múltiples plataformas.

Otra de las características que posee es que resulta una herramienta amigable para el usuario, permite la sincronización entre Diagramas de Entidad Relación y Diagramas de Clases, la generación de Código / Ingeniería Inversa. Además permite la integración entre distintos ambientes de Desarrollo Integrados (IDE) como Visual Studio y Eclipse entre otros. (12)

## **1.7 Metodología de desarrollo de software utilizada**

Un proceso de desarrollo de software tiene como objetivo la producción eficiente de un producto de software que satisfaga los requisitos de un cliente con una planificación y una estimación de recursos predecibles; una metodología de desarrollo de software es una guía, un modelo a seguir para lograrlo, compuesta por un conjunto de procedimientos, técnicas y ayudas a la documentación, modelando las interacciones y relaciones que suceden entre las personas (roles), las actividades que estas desarrollan y los artefactos que se crean o actualizan durante el proceso, es decir, definen “Quién” va hacer “Qué”,

“Cuándo” y “Como” va hacerlo, lo cual ayuda a organizar el trabajo y orientar a todos los trabajadores. En la actualidad no existe una metodología universal para realizar el Proceso de Desarrollo, estas se dividen en dos grupos: las tradicionales y las ágiles, con marcadas diferencias. Un ejemplo de las tradicionales es el archiconocido Rational Unified Process (RUP) y entre las ágiles es muy utilizado el Extreme Programming (XP). Existen también las metodologías híbridas, formadas a partir de la combinación de distintas metodologías, tomando de cada una los elementos que se necesiten y se adapten al proceso de desarrollo de software que se realizará.

## **1.7.1 Modelo de desarrollo del Centro para la Informatización de Gestión de Entidades (CEIGE).**

El Centro para la Informatización de Gestión de Entidades (CEIGE) de la UCI utiliza su propio modelo de desarrollo de software que describe la secuencia de actividades de alto nivel para la construcción y desarrollo de soluciones. Se logra con la combinación entre los modelos basado en Componentes, el Iterativo y el Incremental. Emplea las técnicas de prototipado, si son requeridas, para los requerimientos del usuario de los que no existe una visión clara por parte de estos, con el objetivo de desarrollar una definición mejorada de los requisitos del usuario para el sistema. (15)

- **Desarrollo iterativo e incremental:** Es un enfoque en el que el ciclo de vida está compuesto por iteraciones, estas son pequeños procesos compuestos de varias actividades cuyo objetivo es entregar una parte del sistema parcialmente completo, probado, integrado y estable. Todo el software es integrado en cada entrega de cada iteración hasta obtener el producto de software completo en la última iteración. En cada iteración se obtiene como resultado un incremento.
- **Desarrollo basado en componentes:** lleva a alcanzar un mayor nivel de reutilización de software, aún en contextos distintos a aquellos para los que fue diseñado. Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados. Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema. Dado que un componente puede ser construido y luego mejorado continuamente, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.

Contempla las fases principales del desarrollo de un producto de software, adecuándolas a las necesidades del centro, como son: Modelación de proceso de negocio, Definición de requisitos, Diseño de la Arquitectura de software, Análisis de reutilización y Estrategia de desarrollo, Diseño detallado, Implementación e integración y Pruebas; además de analizar los procesos de Gestión de proyecto.

## **1.8 Conclusiones parciales**

En este capítulo se explican de manera entendible todos los conceptos a los que se harán referencia en el transcurso del Trabajo de Diploma y que son imprescindibles para poder comprenderlo. Se puede concluir que analizar y mejorar los procesos de negocio es de vital importancia para alcanzar resultados exitosos; en este sentido BPM brinda las técnicas y herramientas necesarias para lograrlo, implantar esta filosofía de trabajo depende en gran medida de las características específicas del lugar en donde se va a utilizar; en nuestro país, principalmente en proyectos informáticos, se ha logrado a través del uso de la Metodología TOT, pero debido a que la misma es relativamente joven y novedosa todavía no cuenta con una herramienta que automatice el modelado de los procesos de negocio y gestione los datos de las fichas que genera. Por esta razón se propone como solución realizar el análisis y el diseño para una herramienta, que una vez implementada permita gestionar toda la información que se recoge en la Ficha de Modelación de Procesos (FMP) y además crear un archivo en formato XPDL correcto que pueda ser importado por un BPMS y generar automáticamente el Modelo del Proceso.



### Capítulo 2: Descripción de la solución propuesta

#### 2.1 Introducción

A continuación se describe la propuesta solución, las herramientas a utilizar, la definición de la arquitectura, detallando la descripción de los procesos que lo componen mediante un modelo de dominio donde se abarcan los conceptos fundamentales y sus relaciones, determinando así cuáles pueden ser automatizados para obtener un producto final que satisfaga los requerimientos del cliente.

#### 2.2 Modelo conceptual

Para lograr una mayor comprensión del dominio del problema, se crea el artefacto Modelo conceptual o de Dominio, que no es más que una representación visual de los conceptos u objetos del mundo real que son significativos para el problema o el área que se analiza; representando las clases conceptuales, no los componentes de software. Puede verse como un modelo que comunica a los interesados, cuáles son los términos importantes y cómo se relacionan entre sí.

Debido a que la herramienta que se diseñe debe permitir además exportar parte de la información que en ella se gestiona a un fichero de extensión .xpdI, se debe realizar entonces un análisis del lenguaje con el fin de determinar la manera de representar los elementos definidos en BPMN, como las actividades, eventos de inicio o transiciones, en XPDL.

Para consultar el Diccionario de datos. Ver **¡Error! No se encuentra el origen de la referencia..**

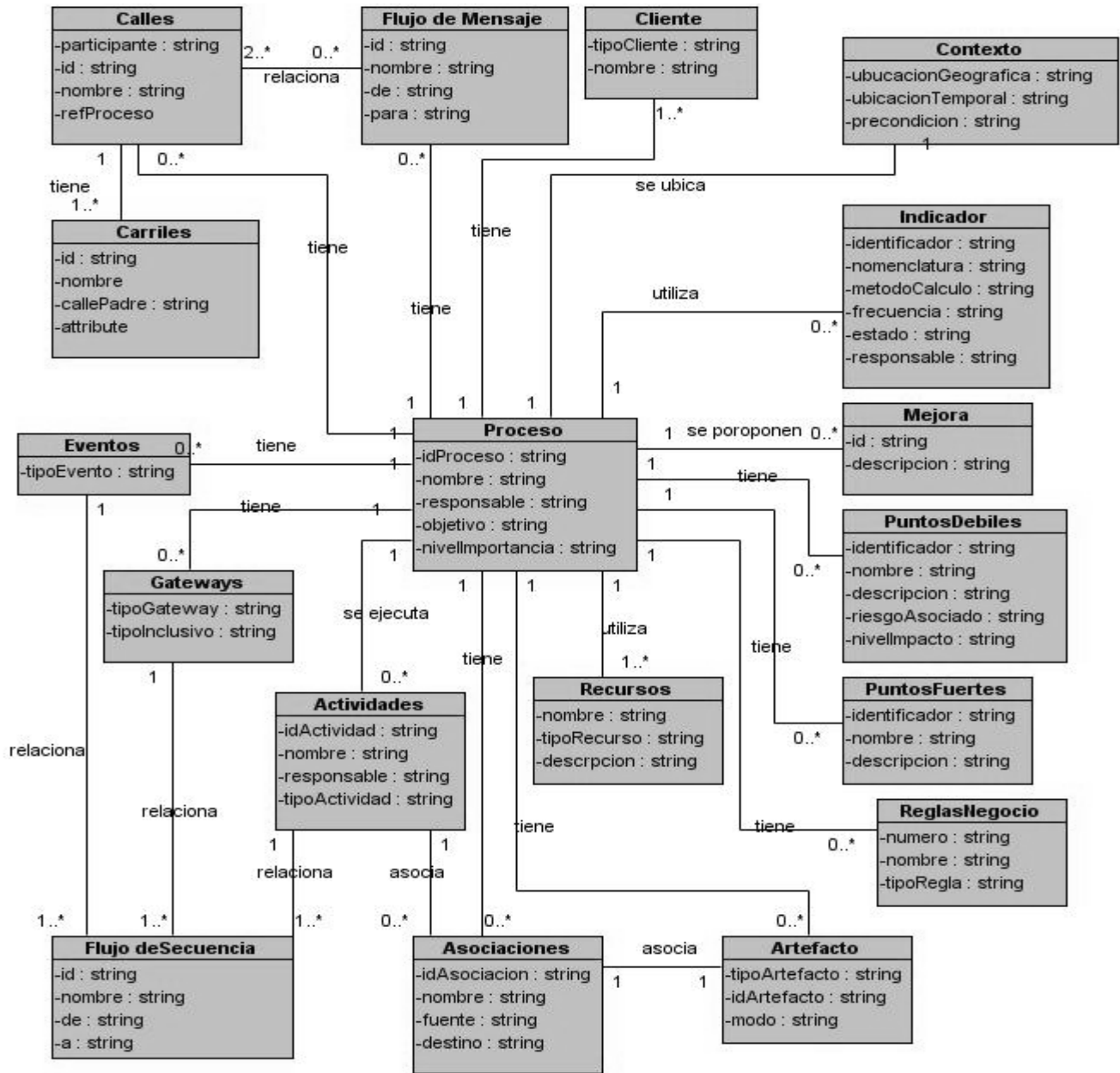


Figura 1 Modelo conceptual

2.3 Requisitos

Un requisito de software es una condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. (16)

Los requerimientos se clasifican en: (17)

- Requerimientos funcionales: son capacidades o condiciones que el sistema debe cumplir.
- Requerimientos no funcionales: son cualidades o propiedades que el producto debe tener, las cuales hacen al producto atractivo, usable, rápido o confiable.

Los requerimientos deben cumplir las siguientes características: (17)

- Especificados por escrito. Como todo contrato o acuerdo entre dos partes.
- Posibles de probar o verificar. Si un requerimiento no se puede comprobar, entonces ¿cómo saber si se cumple con él o no?
- Descritos como una característica del sistema a entregar, o sea deben decir qué es lo que el sistema debe hacer, no cómo hacerlo.
- Lo más general y conciso posible. Para evitar malas interpretaciones.

### 2.3.1 Técnicas empleadas para la captura de los requisitos

En el proceso de desarrollo de software cuando se realiza la actividad Captura de los requisitos el equipo de desarrollo extrae de diversas fuentes de información que estén disponibles, las necesidades que debe cubrir el sistema que se va a desarrollar. Por la complejidad que puede tener este proceso, se han creado técnicas que permitan realizarlo de una forma eficiente y precisa, de ellas se utilizaron las siguientes:

- **La Entrevista** es una técnica para recopilar información a partir de un intercambio directo entre personas o grupos (18). Esta fue la principal técnica empleada para la captura de los requisitos de esta herramienta, dirigida a nuestros principales clientes los Analistas de procesos de Consultoría.
- **La investigación documental** consiste en el estudio de documentos escritos sobre un objeto determinado, es decir son todos aquellos documentos registrados en diferentes dispositivos físicos a los que podemos tener acceso en forma directa o indirecta para su consulta (19). Esta técnica se utilizó para estudiar y analizar cómo funcionan los BPMS en sentido general, y más

específicamente los que importan/exportan código XPDL, lo que permitió tener conocimiento de otros requisitos que no pudieron ser capturados en las entrevistas.

- **Tormenta de ideas:** es también una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas (20). Para el empleo de esta técnica se hicieron reuniones entre los integrantes del equipo de desarrollo y los tutores para, entre todos, debatir acerca del tema y definir las principales funcionalidades.

### **2.3.2 Requisitos funcionales**

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, se mantienen invariables sin importar con que propiedades o cualidades se relacionen (16).

A continuación se presenta la Lista de Requisitos:

#### **RF 1 Gestionar proceso**

RF 1.1.1 Adicionar proceso.

RF 1.1.2 Modificar proceso.

RF 1.1.3 Eliminar proceso.

#### **RF 2 Gestionar actividades.**

RF 2.1 Adicionar actividad.

RF 2.2 Modificar actividad.

RF 2.3 Eliminar actividad

#### **RF 3 Gestionar clientes.**

RF 3.1 Adicionar cliente.

RF 3.2 Modificar cliente.

RF 3.3 Eliminar cliente.

#### **RF 4 Gestionar contexto.**

RF 4.1 Adicionar contexto.

RF 4.2 Modificar contexto.

RF 4.3 Eliminar Contexto.

### **RF 5 Gestionar entradas.**

RF 5.1 Adicionar entrada.

RF 5.2 Modificar entrada.

RF 5.3 Eliminar entrada.

### **RF 6 Gestionar eventos.**

RF 6.1 Adicionar evento.

RF 6.2 Modificar evento.

RF 6.3 Eliminar evento.

### **RF 7 Gestionar indicadores.**

RF 7.1 Adicionar indicador.

RF 7.2 Modificar indicador.

RF 7.3 Eliminar indicador.

### **RF 8 Gestionar mejoras.**

RF 8.1 Adicionar mejora.

RF 8.2 Modificar mejora.

RF 8.3 Eliminar mejora.

### **RF 9 Gestionar puntos débiles**

RF 9.1 Adicionar punto débil.

RF 9.2 Modificar punto débil.

RF 9.3 Eliminar punto débil.

### **RF 10 Gestionar puntos fuertes.**

RF 10.1 Adicionar punto fuerte.

RF 10.2 Modificar punto fuerte.

RF 10.3 Eliminar punto fuerte.

### **RF 11 Gestionar recursos.**

RF 11.1 Adicionar punto recurso.

RF 11.2 Modificar recurso.

RF 11.3 Eliminar recurso.

### **RF 12 Gestionar reglas de negocio.**

RF 12.1 Adicionar regla de negocio.

RF 12.2 Modificar regla de negocio.

RF 12.3 Eliminar regla de negocio.

### **RF 13 Gestionar salidas.**

RF 13.1 Adicionar salida.

RF 13.2 Modificar salida.

RF 13.3 Eliminar salida.

### **RF 14 Gestionar transición.**

RF 14.1 Adicionar transición.

RF 14.2 Modificar transición.

RF 14.3 Eliminar transición.

### **RF 15 Gestionar gateway.**

RF 15.1 Adicionar gateway.

RF 15.2 Modificar gateway.

RF 15.3 Eliminar gateway.

### **RF 16 Consultar información**

RF 16.1 Mostrar todos los procesos.

RF 16.2 Mostrar todos los clientes de un proceso.

RF 16.3 Mostrar todos los recursos de un proceso.

RF 16.4 Mostrar todos los indicadores de un proceso.

RF 16.5 Mostrar todas las reglas de negocio de un proceso.

RF 16.6 Mostrar todas las actividades de un proceso.

RF 16.7 Mostrar todos los eventos de un proceso.

RF 16.8 Mostrar todas las transiciones de un proceso.

RF 16.9 Mostrar todos los puntos fuertes de un proceso.

RF 16.10 Mostrar todos los puntos débiles de un proceso.

### **RF 17 Generar script**

#### **2.3.3 Especificaciones de requisitos y prototipos de interfaz de usuario**

Una especificación de requisitos del software es una descripción completa del comportamiento del sistema a desarrollar donde se registran las características y condiciones definidas con que debe cumplir cada requisito funcional, son artefactos que se generan durante el proceso de captura y descripción de los mismos.

La interfaz de usuario es el vínculo que se establece entre el usuario y la computadora, es decir es un conjunto de comandos o menús a través de los cuales el usuario se comunica con el programa. Es una de las partes más importantes de cualquier programa ya que determina que tan fácilmente es posible que el programa haga lo que el usuario quiere hacer (21). Un programa muy poderoso con una interfaz pobremente elaborada tiene poco valor para un usuario no experto. Un prototipo es una pequeña muestra,

## CAPÍTULO 2

de funcionalidad limitada, de cómo sería el producto final una vez terminado que ayuda a conocer la opinión de los usuarios y rectificar algunos aspectos antes de llegar al producto terminado.

Debido al gran espacio que ocupa una Especificación de Requisito con su prototipo de interfaz de usuario, en este documento se explicarán tres solamente, correspondientes a los requisitos que se consideran más importantes. Para consultar las especificaciones de requisitos con su prototipo de interfaz de usuario Ver **¡Error! No se encuentra el origen de la referencia..**

### Especificación del requisito RF 1 Gestionar proceso

- RF 1.1 Adicionar proceso

Tabla 2 Especificación del requisito Adicionar proceso

<b>Precondiciones</b>	Se ha creado un Paquete.
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
1	El usuario la opción adicionar un nuevo proceso.
2	El sistema muestra los campos que debe llenar el usuario: <b>Identificador:</b> para identificar el proceso <b>Nombre:</b> nombre del proceso donde se resume el objetivo del proceso. <b>Macro proceso:</b> macro proceso al que pertenece el proceso. <b>Producto:</b> resultado final que se obtiene una vez terminado el proceso. <b>Responsable:</b> entidades que se involucran activamente en el proceso, personas o estructuras organizacionales que tienen un rol en el mismo. <b>Nivel de importancia:</b> se puede clasificar en Alto cuando tiene vinculación directa con la razón de ser y el objeto social de la entidad, Medio cuando la vinculación que tiene es de apoyo y Bajo cuando puede ser o no requerido para lograr la razón de ser y el objeto social de la entidad.
3	El usuario llena todos los campos.
4	El sistema confirma el almacenamiento de los datos.
5	Concluye el requisito.
<b>Pos-condiciones</b>	
1	El proceso fue almacenado.
<b>Flujos alternativos</b>	



## CAPÍTULO 2

---

N/A		
<b>Flujo alternativo *.a El usuario cancela la acción</b>		
1	Concluye el requisito.	
<b>Pos-condiciones</b>		
1	No se almacena el proceso.	
<b>Validaciones</b>		
1	N/A	
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	
	<b>Extensiones</b>	N/A
<b>Conceptos</b>	Proceso de negocio	Conjunto de actividades relacionadas lógicamente para lograr un resultado de negocio definido.
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

- **Prototipos de interfaz de usuario para Adicionar proceso**

## CAPÍTULO 2

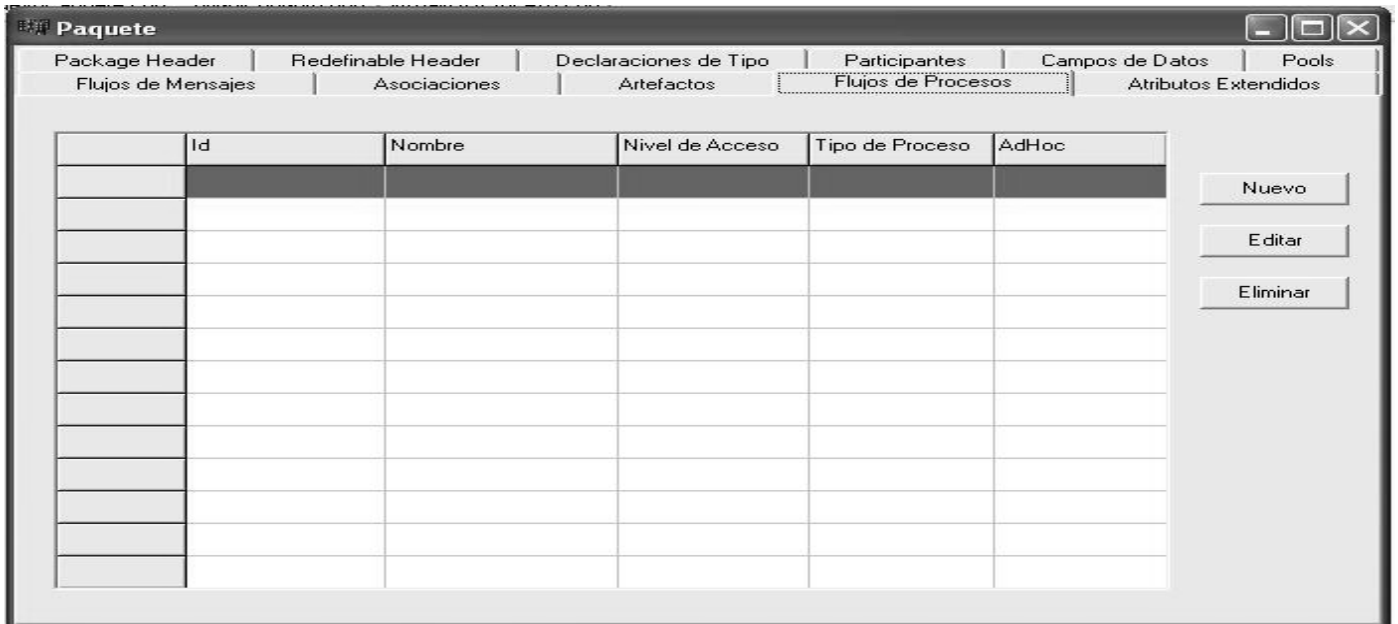


Figura 1 Prototipo de interfaz de usuario para Adicionar proceso

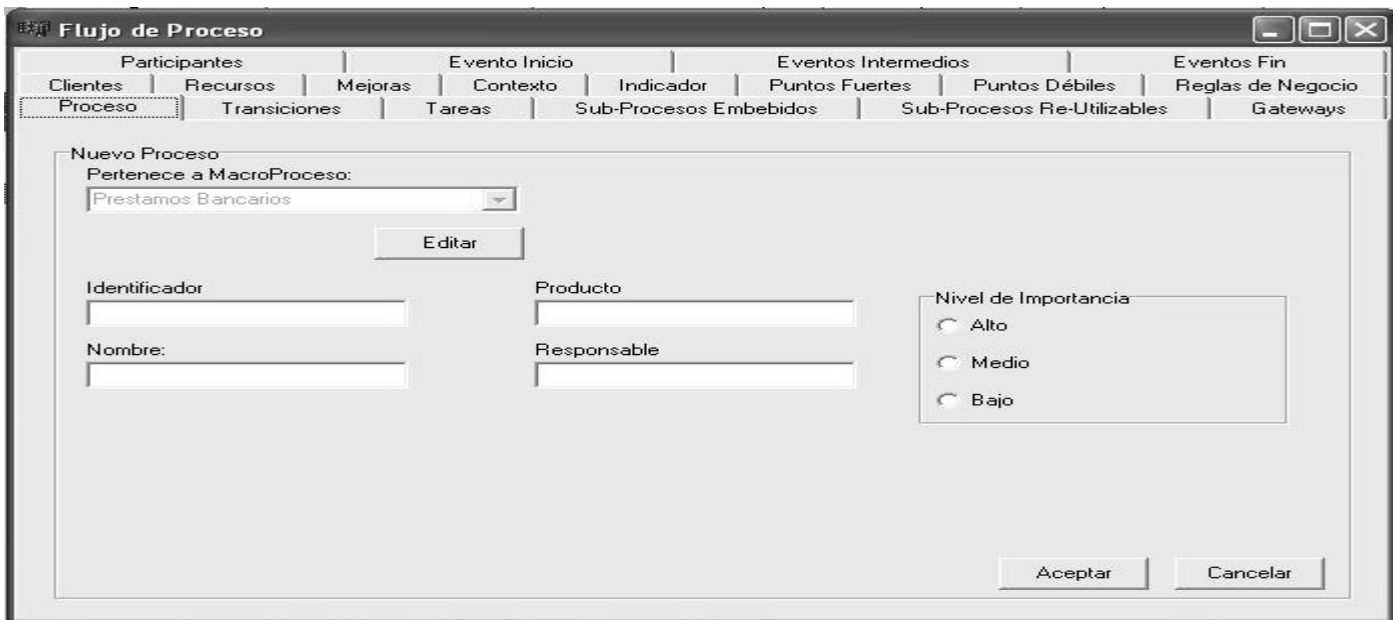


Figura 2 Prototipo de interfaz de usuario para Adicionar proceso

## CAPÍTULO 2

- **RF 1.2 Modificar proceso**

**Tabla 3 Especificación del requisito Modificar proceso**

<b>Precondiciones</b>	Existe un proceso que se desea modificar. El usuario tiene permisos para realizar la acción.
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
1	El usuario selecciona el proceso que desea modificar.
2	<p>El sistema muestra todos los datos del proceso editados en los elementos visuales de modo que el cliente pueda modificarlos. Los datos son:</p> <p><b>Identificador:</b></p> <p><b>Nombre:</b> nombre del proceso donde se resume el objetivo del proceso.</p> <p><b>Macro proceso:</b> macro proceso al que pertenece el proceso.</p> <p><b>Producto:</b> resultado final que se obtiene una vez terminado el proceso.</p> <p><b>Responsable:</b> entidades que se involucran activamente en el proceso, personas o estructuras organizacionales que tienen un rol en el mismo.</p> <p><b>Nivel de importancia:</b> se puede clasificar en Alto cuando tiene vinculación directa con la razón de ser y el objeto social de la entidad, Medio cuando la vinculación que tiene es de apoyo y Bajo cuando puede ser o no requerido para lograr la razón de ser y el objeto social de la entidad.</p> <p><b>Cliente:</b> elementos internos y externos a la empresa con los cuales interactúa el proceso.</p> <p><b>Contexto:</b> acciones previas necesarias para iniciar el proceso, son ubicación geográfica y temporal y el estado inicial del proceso.</p> <p><b>Indicador:</b> se evalúan los indicadores de Calidad y Gestión para medir el proceso.</p> <p><b>Mejora:</b> posibles mejoras que se le pueden aplicar al proceso.</p> <p><b>Puntos débiles:</b> problemas y debilidades del proceso que deben ser revisados y modificados.</p> <p><b>Puntos fuertes:</b> principales puntos y aspectos que deben mantenerse en el proceso.</p> <p><b>Recursos:</b> los recursos que utiliza el proceso en su desarrollo, se dividen en Recursos humanos que son las personas que intervienen en el proceso, Recursos materiales que es la materia prima necesaria para el desarrollo del proceso y los Recursos tecnológicos que son aquellos objetos pasivos utilizados por los actores durante el desarrollo del proceso.</p> <p><b>Reglas o políticas del negocio:</b> colección de políticas y restricciones del negocio de una organización que controlan su comportamiento.</p>
3	El usuario modifica los valores que desee contenidos dentro del proceso y selecciona la opción guardar

## CAPÍTULO 2

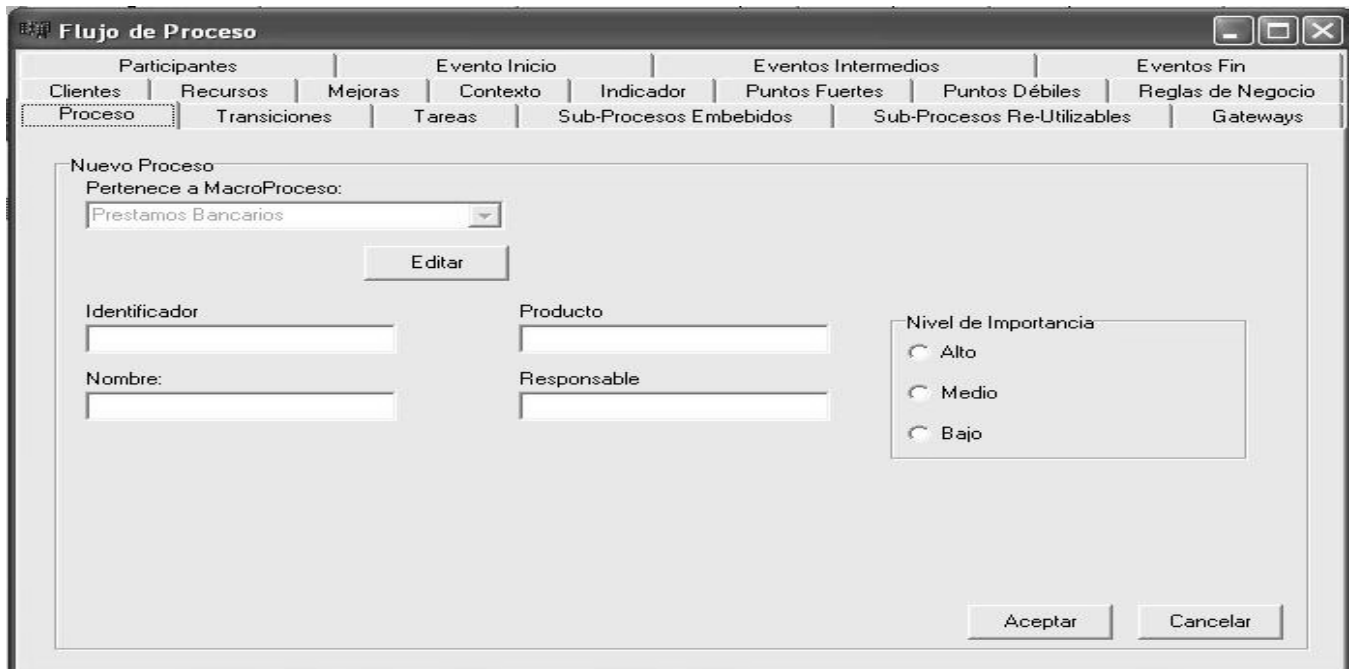
---

	cambios.	
4	El sistema muestra un mensaje para confirmar que el usuario desea modificar el proceso.	
5	El usuario selecciona aceptar.	
6	El sistema guarda los cambios realizados por el usuario.	
7	Concluye el requisito.	
<b>Pos-condiciones</b>		
1	El Proceso fue modificado.	
<b>Flujos alternativos</b>		
<b>Flujo alternativo *.a El usuario Cancela la acción</b>		
1.	Concluye el requisito.	
<b>Pos-condiciones</b>		
1.	El proceso no se modifica.	
<b>Validaciones</b>		
1		
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	
	<b>Extensiones</b>	
<b>Conceptos</b>	Proceso de negocio	Conjunto de actividades relacionadas lógicamente para lograr un resultado de negocio definido.
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	

- **Prototipos de interfaz de usuario para Modificar proceso**



**Figura 3 Prototipo de interfaz de usuario para Modificar proceso**



**Figura 4 Prototipo de interfaz de usuario para Modificar proceso**

## CAPÍTULO 2

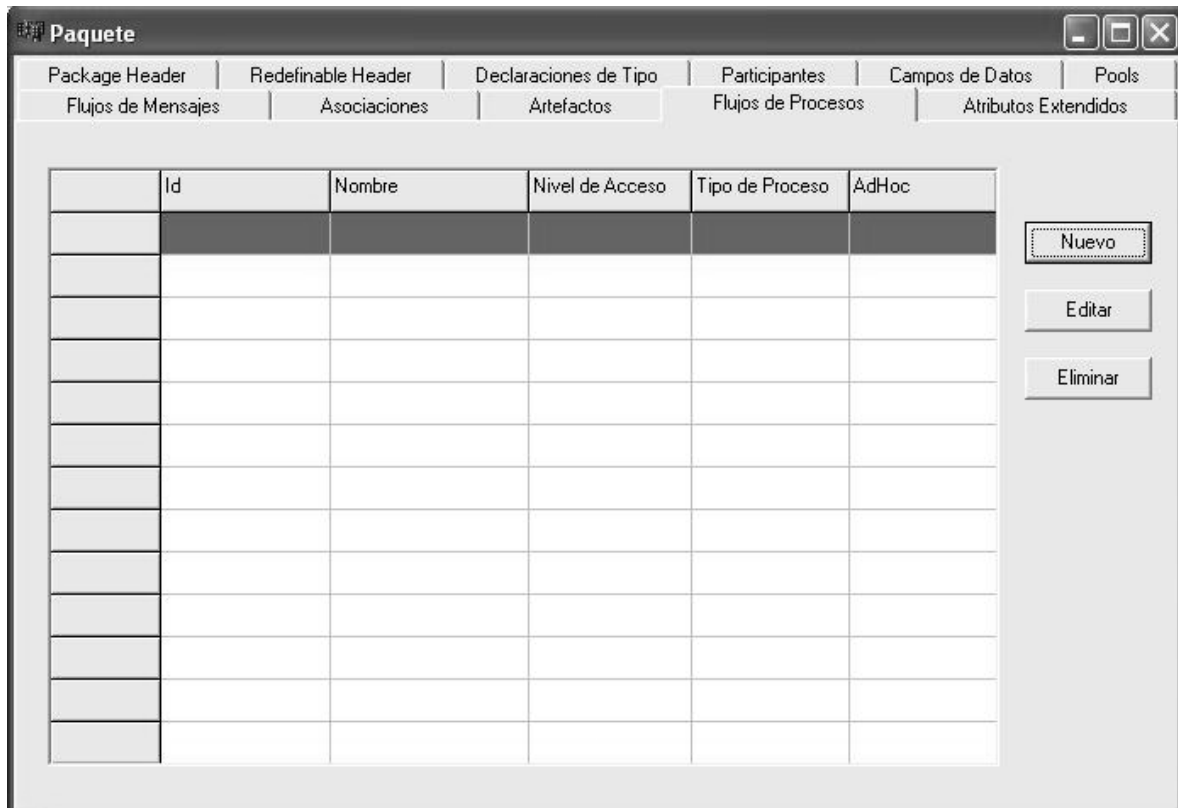
- **RF 1.3 Eliminar Proceso**

**Tabla 4 Especificación del requisito Eliminar Proceso**

<b>Precondiciones</b>		Se desea eliminar un proceso. El usuario tiene permisos para realizar esta acción.
<b>Flujo de eventos</b>		
<b>Flujo básico</b>		
1	El usuario selecciona la opción eliminar proceso.	
2	El sistema muestra todos los procesos que se han adicionado anteriormente.	
3	El usuario selecciona el proceso que desea eliminar y selecciona la opción eliminar.	
4	El sistema muestra un mensaje para confirmar que el usuario desea eliminar el proceso.	
5	El usuario selecciona la opción aceptar.	
6	El sistema elimina el proceso.	
7	Concluye el requisito.	
<b>Pos-condiciones</b>		
1	El Proceso fue eliminado.	
<b>Flujos alternativos</b>		
<b>Flujo alternativo *.a</b>		
1	El usuario cancela la acción.	
<b>Pos-condiciones</b>		
1	El proceso no fue eliminado.	
<b>Validaciones</b>		
1		
<b>Relaciones</b>	<b>Requisitos Incluidos</b>	N/A
	<b>Extensiones</b>	N/A
<b>Conceptos</b>	Proceso de negocio	Conjunto de actividades relacionadas lógicamente para lograr un resultado de negocio definido.
<b>Requisitos especiales</b>	N/A	
<b>Asuntos</b>	N/A	

pendientes

- **Prototipo de interfaz de usuario para Eliminar proceso**



**Figura 5 Prototipo de interfaz de usuario para Eliminar proceso**

### 2.3.4 Validación de los requisitos funcionales

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea. Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de requisitos son correctos, para ello se evalúa su calidad en la fase de validación. La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema que han

sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos. Una especificación es considerada con calidad por el estándar IEEE 830 cuando es correcta, no ambigua, completa, consistente, ordenada por importancia y estabilidad, verificable, modificable y trazable (16).

Existen algunas técnicas que pueden aplicarse para ello, por ejemplo: Reviews o Walk-throughs, Auditorías, Matrices de Trazabilidad, Revisión técnica formal y la utilización de Prototipos. Para validar los requisitos se utilizaron dos de estas técnicas:

1. **Revisión técnica formal:** es una actividad que garantiza la calidad, su objetivo primario es encontrar errores durante el proceso, incluye recorridos, inspecciones y revisiones cíclicas. Cada Revisión Técnica Formal (RTF) se lleva a cabo mediante una reunión y solo tiene éxito si esta es bien planificada, controlada y atendida. Una vez terminadas las especificaciones de requisitos se efectuó una reunión para revisarlos con el cliente.
2. **Prototipos:** se definieron los prototipos de interfaz de usuario que sin tener la funcionalidad del sistema permitieron al usuario hacerse una idea de la estructura sistema. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

También existen algunas métricas auxiliares que ayudan a medir el grado en que algunas de las características antes mencionadas se reflejan en las especificaciones realizadas con valores cuantitativos, por ejemplo: (22)

- **Especificidad:** Mide la relación entre los requisitos que han sido interpretados de igual forma por los revisores y los requisitos totales. Esta métrica permite determinar el grado de ambigüedad que pueda llegar a tener una especificación de requisitos.
- **Estabilidad:** Se refiere a los cambios que sufren en los requisitos a lo largo de todo el ciclo de vida del software incluyendo la eliminación, inserción y modificación, estos continuos cambios en la especificación de los requisitos traen consigo un atraso en el cronograma de trabajo.
- **Requerimientos en conflictos (Consistencia):** Se refiere a la consistencia que deben presentar las especificaciones. Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorio no se pueden implementar.
- **Redundancia:** Esta métrica se refiere a la concordancia que tiene que existir en la especificación.



## CAPÍTULO 2

---

- **Requerimientos por necesidades (Corrección):** Verifica que las necesidades de los clientes se encuentren cubiertas por los requerimientos capturados y especificados. Un conjunto de requisitos software es correcto sólo si todos los requisitos contenidos representan algo que es requerido para la construcción del sistema y no hay errores que afecten al diseño.

**Tabla 5 Métricas aplicadas a la especificación de requisitos**

	Descripción	Valor
<b>RT</b>	Total de requerimientos	17
<b>NUI</b>	Número de requisitos para los que todos los revisores tuvieron una misma interpretación.	17
<b>RC</b>	Cantidad de requisitos cambiados (insertados, modificados y eliminados)	0
<b>RCI</b>	Cantidad de requisitos en conflicto internamente con la especificación.	0
<b>RCE</b>	Cantidad de requisitos en conflicto con otra documentación.	0
<b>CN</b>	Cantidad de necesidades de los clientes	3
<b>CNR</b>	Cantidad de necesidades sin requerimientos	0
<b>CRR</b>	Cantidad de requisitos repetidos	0

**Tabla 6 Resultado de la métricas aplicadas a la especificación de requisitos**

No	Métrica	Fórmula	Propiedad	Valor
1	Especificidad	$Q = NUI / RT * 100$	No ambigüedad	100%
2	Estabilidad	$ETR = (RT-RC)/RT*100$	Estabilidad	100%
3	Requerimientos en conflicto	$PRC = (RCI+RCE)/RT* 100$	Consistencia externa e interna	0

## CAPÍTULO 2

---

4	Por ciento de redundancia	$RR = CRR / RT * 100$	No redundancia	0
5	Promedio de requerimientos por necesidades	$CNR = RT / CN$	Corrección	5

- **Resultado que arrojan las métricas:**

Cuando una especificidad de requerimientos es de un 100% significa que es realizada con una alta calidad, con ausencia de ambigüedades y evidencia que todas las personas involucradas en el proceso de revisar los requisitos tuvieron la misma interpretación de los mismos. La estabilidad de un 1000% demostró que no hubo cambios en los requisitos con respecto al total de ellos. Entre los requerimientos no hubo conflictos lo que evidencia buena consistencia externa e interna. Existe concordancia entre los requerimientos porque no se presentaron redundancia entre ellos y por último el promedio de requerimientos por necesidades tiene valor 5 significa que existen 5 requisitos por cada necesidad real del usuario.

### 2.3.5 Conclusiones parciales

En este capítulo se ha descrito qué debe hacer y cómo debe ser la herramienta a través del Modelo de Dominio, los Requisitos Funcionales y los No Funcionales, generando como artefactos significativos las especificaciones de requisitos y los prototipos de interfaz de usuario, estos resultados obtenidos en este capítulo serán imprescindibles para realizar el capítulo posterior que contempla el diseño de la aplicación.

### Capítulo 3: Diseño y validación de la solución

#### 3.1 Introducción

Con el resultado del análisis se procede a realizar el diseño de la aplicación para modelarla de forma tal que soporte todos los requisitos que se han definido anteriormente y que sienta las bases para que la implementación satisfaga las necesidades del cliente. Como parte del diseño se encuentran los patrones de diseño que se van a emplear en la solución, los diagramas de clases y de clases del diseño, así como el modelado de la base de datos.

#### 3.2 Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos, son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará y los componentes que se encargaran del funcionamiento y la relación entre uno y otro. A continuación se muestra el diagrama de todas clases de la aplicación a desarrollar y las relaciones que tienen entre sí.

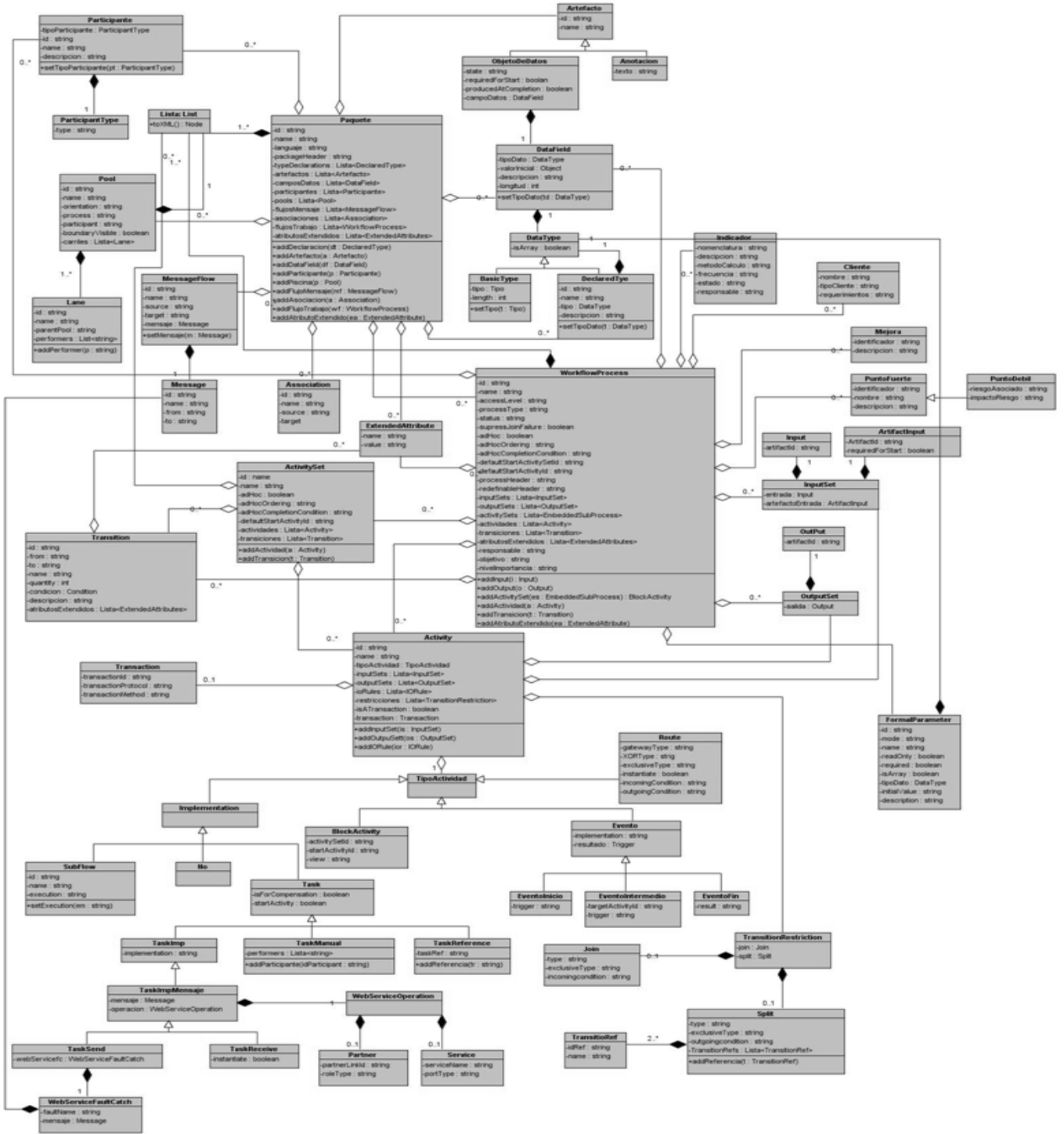


Figura 1 Diagrama de clases

### 3.3 Descripción de la arquitectura

La arquitectura de software es una disciplina que tiene por objetivo definir la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre estos y el entorno y los principios que orientan su diseño y evolución (Arquitectura - IEEE 1471-2000). El objetivo primario de la arquitectura de software es el de aportar elementos que ayuden en la toma de decisiones significativas y al mismo tiempo, proporcionar conceptos y un lenguaje común que permitan la comunicación entre todos los que tienen parte en el proceso de desarrollo del software. Estas decisiones significativas se toman en cuanto a:

- La organización del sistema de software.
- La selección de los elementos estructurales y sus interfaces a través de los cuales se constituye el sistema.
- Su comportamiento, según resultados de las colaboraciones entre esos elementos.
- El estilo arquitectónico que guía esta organización: los elementos estáticos y dinámicos; sus interfaces, su colaboración y su composición.

#### **Arquitectura en capas:**

La arquitectura basada en capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica proveyendo una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada.

El estilo de arquitectura basado en capas se identifica por las siguientes características:

- Describe la descomposición de servicios de forma que la mayoría de la interacción ocurre solamente entre capas vecinas.
- Las capas de una aplicación pueden residir en la misma maquina física (misma capa) o puede estar distribuido sobre diferentes computadores (n-capas).

## CAPÍTULO 3

---

- Los componentes de cada capa se comunican con otros componentes en otras capas a través de interfaces muy bien definidas.
- Este modelo ha sido descrito como una “pirámide invertida de re-uso” donde cada capa agrega responsabilidad y abstracción a la capa directamente sobre ella.

Los principios comunes que se aplican cuando se diseña para usar este estilo de arquitectura incluyen:

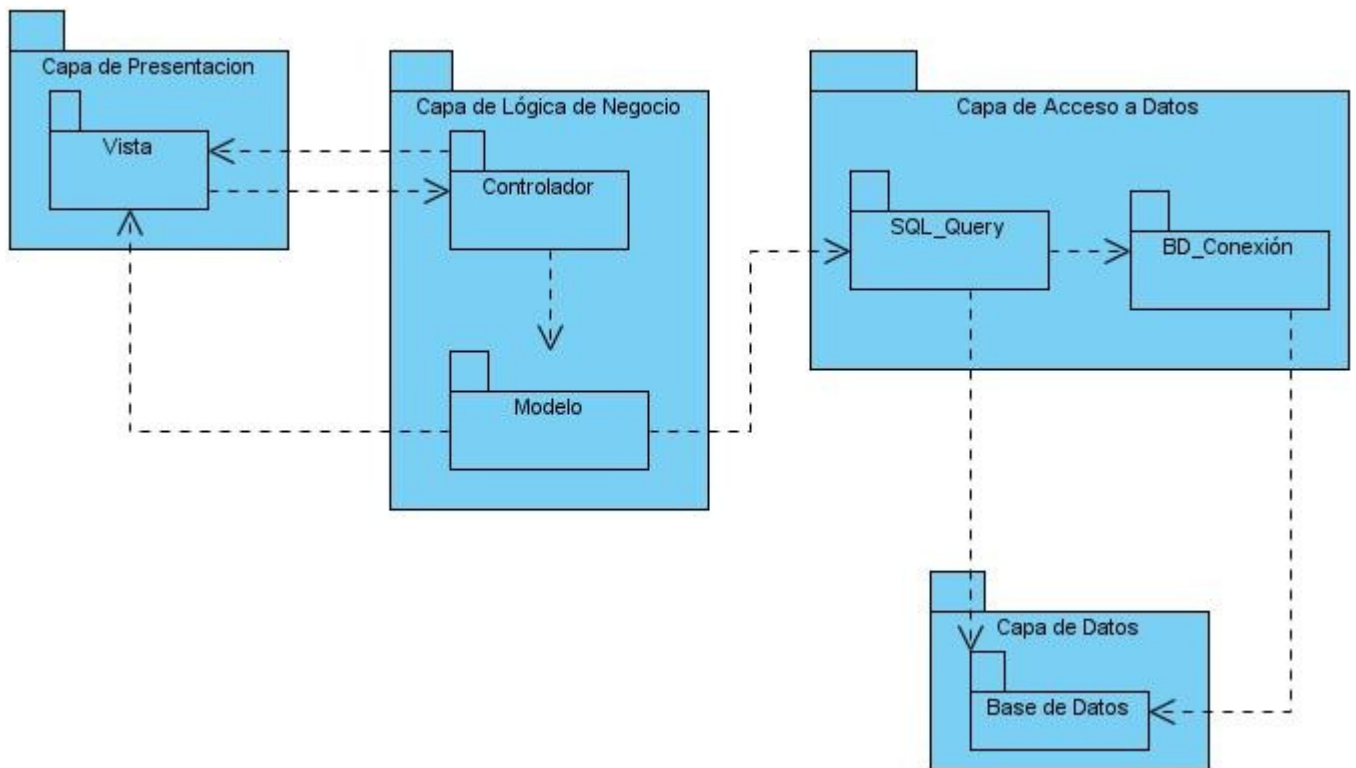
- **Abstracción:** La arquitectura basada en capas abstrae la vista del modelo como un todo mientras que provee suficiente detalle para entender las relaciones entre capas.
- **Encapsulamiento:** El diseño no hace asunciones acerca de tipos de datos, métodos, propiedades o implementación.
- **Funcionalidad claramente definida:** El diseño claramente define la separación entre la funcionalidad de cada capa. Capas superiores como la capa de presentación envía comandos a las capas inferiores como la capa de negocios y la capa de datos y los datos fluyen hacia y desde las capas en cualquier sentido.
- **Alta cohesión:** Cada capa contiene funcionalidad directamente relacionadas con la tarea de dicha capa.
- **Reutilizable:** Las capas inferiores no tienen ninguna dependencia con las capas superiores, permitiéndoles ser reutilizables en otros escenarios.

La descomposición más habitual es dividir la aplicación en tres capas:

- **Capa de presentación:** se ocupa de toda la interacción entre el usuario y el software, pudiendo tratarse de un sistema de menús muy simple o una interfaz gráfica de usuario relativamente compleja.
- **Capa del dominio:** también conocida como la lógica de negocio, que es todo lo que necesita conocer la aplicación para poder trabajar con el dominio en cuestión. Implica realizar cálculos basados en datos de entrada o almacenados, validación de cualquier dato proveniente de la capa

de presentación y la ejecución de algoritmos específicos en función de los comandos de la presentación.

- **Capa de acceso a datos:** se encarga de manejar el mecanismo de almacenamiento.



**Figura 6 Estructura de la arquitectura**

### 3.4 Tecnologías y herramientas de desarrollo propuestas para la solución

#### 3.4.1 Software libre

El Software libre es aquel en que el usuario tiene la libertad de una vez que lo obtiene, distribuirlo con o sin modificaciones. El Software libre tiene esa denominación, debido a que posee cuatro libertades básicas:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades (libertad 1).

- El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2). (13)

### 3.4.2 Lenguaje de programación Java

Java es un lenguaje de alto nivel y entre sus características más importantes sobresalen que es orientado a objetos, sencillo, independiente de plataforma, brinda un gran nivel de seguridad, capacidad multihilo, gran rendimiento, creación de aplicaciones distribuidas y robustas. Está diseñado para que un programa escrito en este lenguaje sea ejecutado independientemente de la plataforma ya sea de hardware, software y sistema operativo en la que se esté actuando.

### 3.4.3 Entorno de Desarrollo Integrado NetBeans 6.8

El Entorno de Desarrollo Integrado (Integrated Development Environment, IDE) NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans; es un producto libre y gratuito sin restricciones de uso; de código abierto escrito completamente en Java usando la plataforma NetBeans. Soporta el desarrollo de todos los tipos de aplicación Java ya sean J2SE (la plataforma Java SE), web, EJB (Enterprise JavaBeans, son una de las API que forman parte del estándar de construcción de aplicaciones) y aplicaciones móviles. Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring. Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.

NetBeans IDE dispone de:

- Soporte para crear interfaces gráficas de forma visual.
- Desarrollo de aplicaciones web.
- Control de versiones.
- Resaltado de sintaxis.



- Sus funcionalidades son ampliables mediante la instalación de packs.

### 3.4.4 Sistema de gestión de Base de Datos: PostgreSQL

PostgreSQL es un Sistema de Gestión de Bases de Datos objeto-relacional, que está distribuido bajo licencia BSD<sup>7</sup> y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

A continuación las características más importantes soportadas por PostgreSQL (14):

- Es una base de datos 100% ACID<sup>8</sup>
- Integridad referencial
- Replicación asíncrona
- Copias de seguridad en caliente (Online/hot backups)
- Unicode
- Juegos de caracteres internacionales
- Completa documentación
- Licencia BSD
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows.
- El tamaño de la base de datos es ilimitado.

---

<sup>7</sup> Berkeley Software Distribution se utiliza para identificar un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.

<sup>8</sup> Un Sistema de Gestión de bases de datos es ACID si cuenta con las funcionalidades necesarias para que sus transacciones tengan las características ACID. En concreto **ACID** es un acrónimo de **A**tomicity, **C**onsistency, **I**solation and **D**urability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español.

3.5 Diagrama de componentes

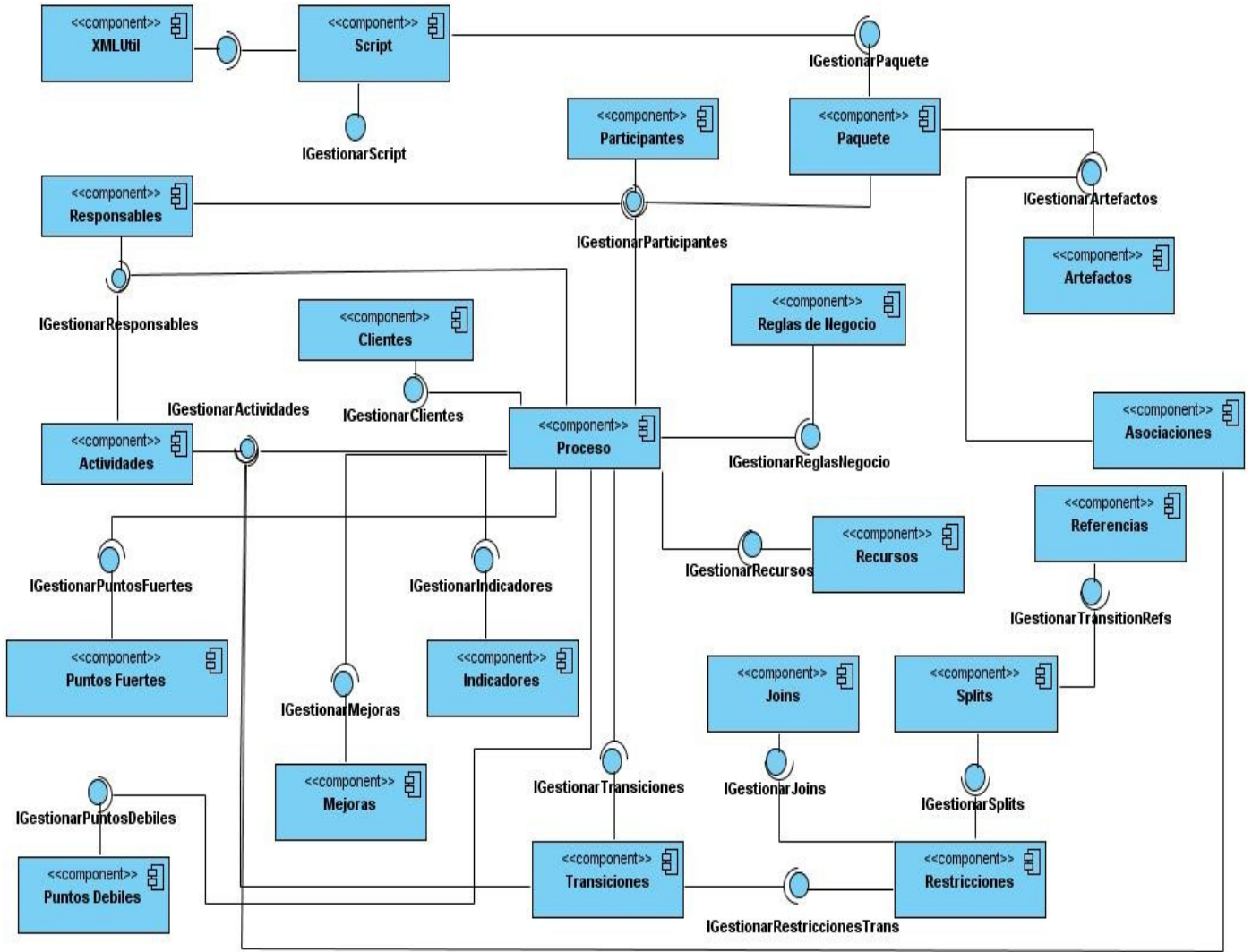


Figura 7 Diagrama de componentes

3.6 Diagramas de clases del diseño

Los diagramas de clases del diseño se realizan con el objetivo de tener en cuenta los detalles concretos de la implementación del sistema, son la base para los diagramas de componentes y los diagramas de despliegue, los diagramas de clases del diseño son importantes no sólo para visualizar, especificar y

## CAPÍTULO 3

---

documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa. Son muy útiles porque muestran a través de atributos y métodos la estructura de las clases que después serán escritas en algún lenguaje de programación.

Se realizó un diagrama de clases del diseño por funcionalidad, a continuación se muestran los correspondientes a Gestionar proceso, con la descripción de las clases. Para consultar los demás diagramas de clases del diseño ver el **¡Error! No se encuentra el origen de la referencia..**

• Diagrama de clase del diseño de Gestionar proceso

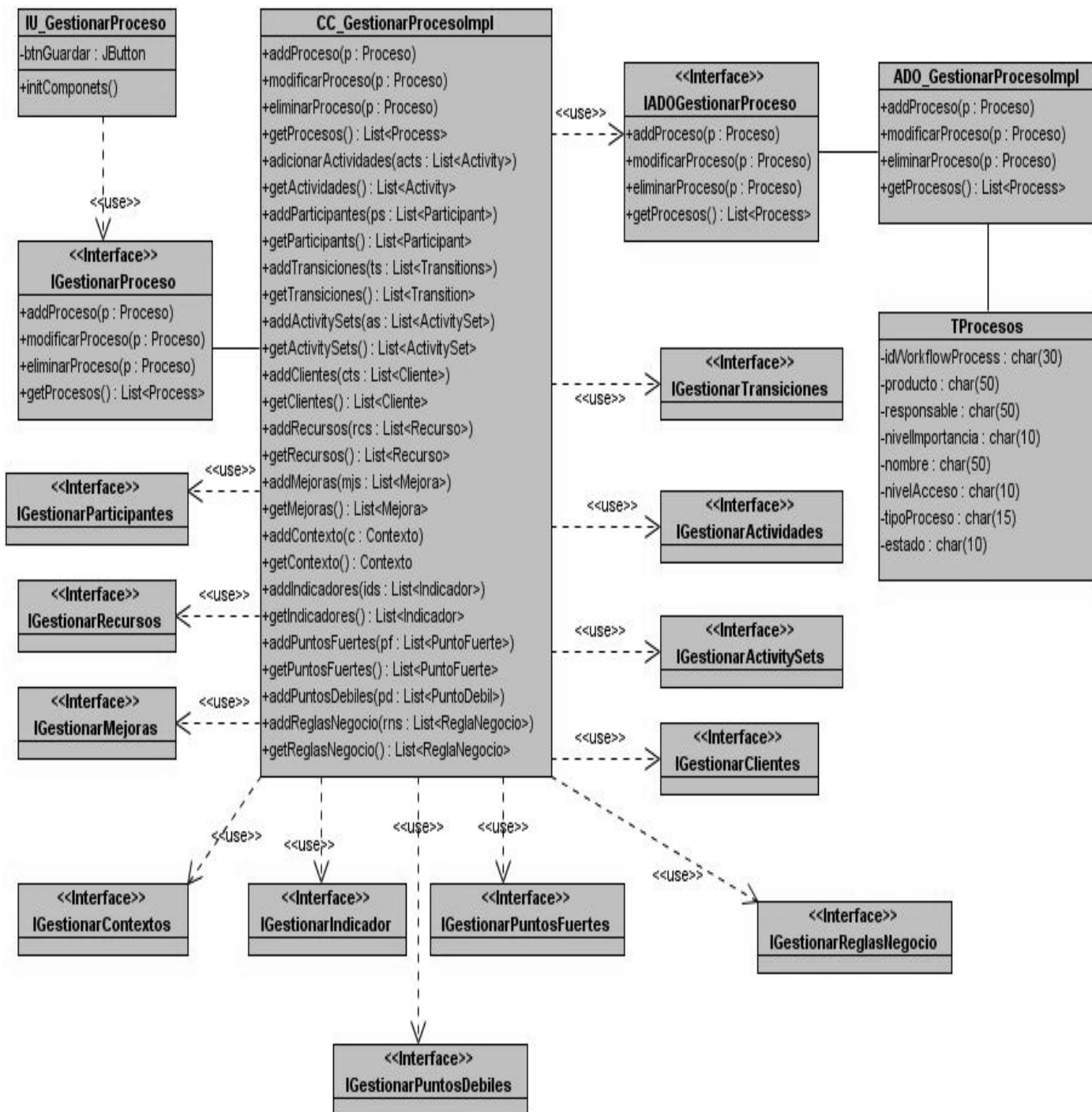


Figura 2 Diagrama de clase del diseño Gestionar proceso

### Descripción del diseño de clases de Gestionar proceso:

**Tabla 7 Descripción del diseño de clases de Gestionar proceso**

Clases	Descripción
IU_GestionarProceso	Crea los elementos visuales que muestran y reciben datos del usuario.
<<interface>> IGestionarProceso	Proporciona las operaciones que deben ser implementadas por la clase CC_GestionarProcesoImpl.
<<interface>> IGestionarParticipantes	Proporciona las operaciones que deben ser implementadas por la clase CC_GestionarParticipanteImpl.
<<interface>> IGestionarRecursos	Proporciona las operaciones que deben ser implementadas por la clase CC_GestionarRecursoImpl.
<<interface>> IGestionarMejoras	Proporciona las operaciones que deben ser implementadas por la clase CC_GestionarMejorasImpl.
<<interface>> IGestionarContexto	Proporciona las operaciones que deben ser implementadas por la clase CC_GestionarContextoImpl.
<<interface>> IGestionarIndicador	Proporciona las operaciones que deben ser implementadas por la clase CC_GestionarIndicadorImpl.
<<interface>> IGestionarPuntosDebiles	Proporciona las operaciones que deben ser implementadas por la clase CC_GestionarPuntosDébilesImpl.
<<interface>> IGestionarPuntosFuentes	Proporciona las operaciones que deben ser implementadas por la clase CC_GestionarPuntosFuertesImpl.
<<interface>> IGestionarReglasDelNegocio	Proporciona las operaciones que deben ser implementadas por la clase CC_GestionarReglasDeNegocioImpl.

## CAPÍTULO 3

---

<<interface>> IGestionarClientes	Proporciona las operaciones que deben ser implementadas por la clase CC_GestionarClientesImpl.
<<interface>> IGestionarActividad	Proporciona las operaciones que deben ser implementadas por la clase CC_GestionarActividadImpl.
<<interface>> IGestionarActivitySet	Proporciona las operaciones que deben ser implementadas por la clase CC_GestionarActivitySetImpl.
<<interface>> IGestionarTransición	Proporciona las operaciones que deben ser implementadas por la clase CC_GestionarTransiciónImpl.
CC_GestionarProcesoImpl	Clase controladora encargada de gestionar cada uno de los elementos que componen al Proceso según las peticiones del usuario.
<<interface>> IADO_GestionarProceso	Proporciona las operaciones que deben ser implementadas por la clase ADO_GestionarProceso.
ADO_GestionarProcesoImpl	Implementa las funcionalidades encargadas de operar con la base de datos para adicionar, modificar y eliminar datos del Proceso
TProceso	Es una tabla de la base de datos que contiene los datos de un Proceso.

### 3.7 Modelado de la base de datos

Una Base de Datos es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo. Una base de datos correctamente diseñada permite obtener acceso a información exacta y actualizada. Puesto que un diseño correcto es esencial para lograr los objetivos fijados para la base de datos, parece lógico emplear el tiempo que sea necesario en aprender los principios de un buen diseño ya que, en ese caso, es mucho más probable que la base de datos termine adaptándose a sus necesidades y pueda modificarse fácilmente.

## CAPÍTULO 3

---

El objetivo del diseño lógico es convertir los esquemas conceptuales locales en un esquema lógico global que se ajuste al modelo de SGBD (Sistema de Gestión de Base de Datos) sobre el que se vaya a implementar el sistema. Mientras que el objetivo fundamental del diseño conceptual es la compleción y expresividad de los esquemas conceptuales locales, el objetivo del diseño lógico es obtener una representación que use, del modo más eficiente posible, los recursos que el modelo de SGBD posee para estructurar los datos y para modelar las restricciones.

Teniendo en cuenta los requisitos funcionales Gestionar proceso, Gestionar actividades, Gestionar clientes, Gestionar contexto, Gestionar entradas, Gestionar eventos, Gestionar indicadores, Gestionar mejoras, Gestionar puntos débiles, Gestionar puntos fuertes, Gestionar recursos, Gestionar reglas de negocio, Gestionar salidas, Gestionar transición y Gestionar gateway se definieron un total de 34 tablas. A continuación se muestra el diseño lógico inicial de la base de datos.





### 3.8 Patrones de diseño empleados en la solución

Los patrones de diseño son una solución estándar para un problema común de diseño dentro de un contexto dado y constituyen una manera más práctica de describir ciertos aspectos de la organización de un programa. Hay patrones que abarcan las distintas etapas del desarrollo: desde el análisis hasta el diseño y desde la arquitectura hasta la implementación.

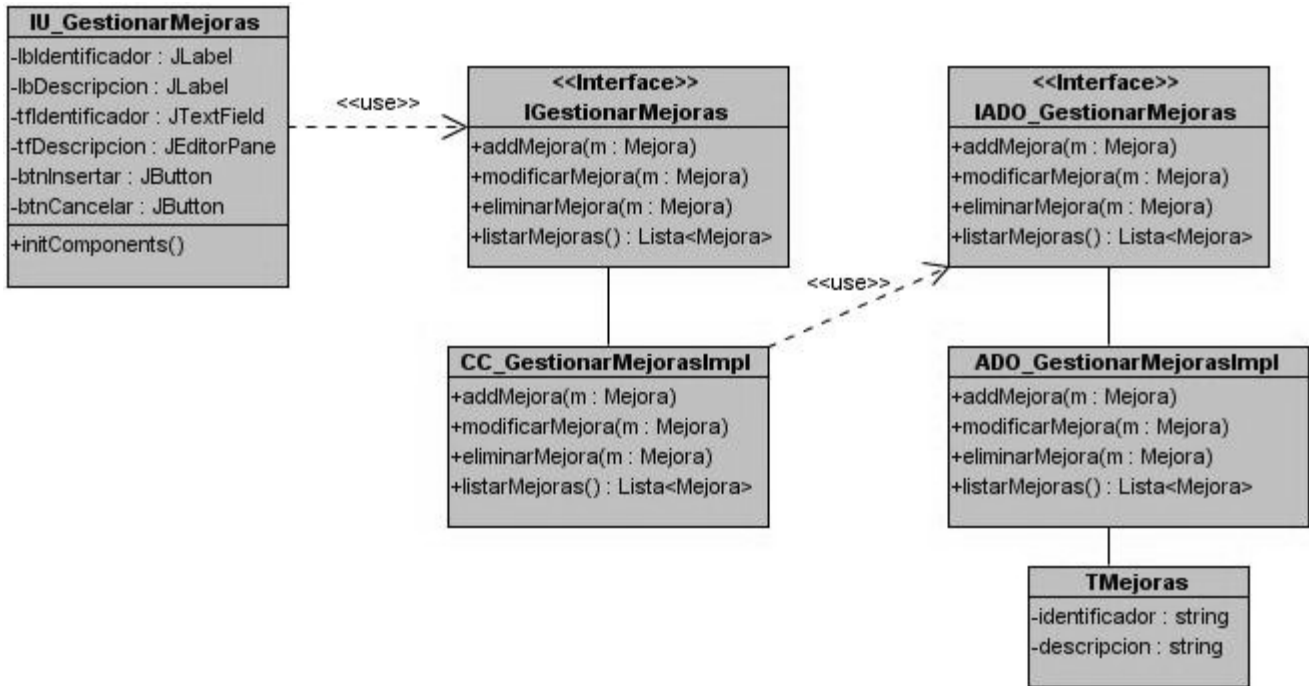
Estos patrones se emplearon por su capacidad de brindarle al diseño flexibilidad y extensibilidad. A continuación se explican brevemente en qué consisten y como se utilizaron:

#### **Patrón Modelo-Vista-Controlador:**

Es un patrón de diseño que plantea la separación de diferentes clases en dependencia de la función que realizan de manera tal que sea posible manejar dinámicamente la forma en que se procesan solicitudes y gestionar cómo se mostrarán los resultados al usuario final. En otras palabras separa la presentación del dominio de la aplicación.

- **Modelo:** administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).
- **Vista:** Maneja la visualización de la información. Esta existe para mantener separado el script de la vista, del modelo y del controlador. Provee un sistema de ayuda, filtros de salida y variables de escape.
- **Controlador:** Son las clases que gestionan el manejo de la lógica del negocio. Por lo general incluyen las restricciones y validaciones fundamentales determinadas por las reglas. Pueden utilizarse más de una.

Por ejemplo, en el siguiente diagrama de diseño para Gestionar mejora se pueden apreciar la vista conformada por la clase IU\_GestionarMejora a través de la cual se muestra información al usuario y se reciben datos; la clase CC\_GestionarMejoraImpl para manejar la lógica del negocio sería el controlador y por último la clase ADO\_GestionarMejorasImpl sería la clase para acceder a los datos y responder a los requerimientos de la vista y a las instrucciones del controlador. Así sucede para todos los demás casos.



**Figura 8 Diagrama de clase del diseño para Gestionar mejora**

### Patrones GRASP:

Son los patrones generales de software para asignar responsabilidades, GRASP por sus siglas en ingles (General Responsibility Assignment Software Paterns), describen en su totalidad los principios fundamentales sobre la asignación de responsabilidades a objetos, todo en forma de patrones.

- **Experto:** se evidencia en la definición de las clases de acuerdo a las funcionalidades que deben realizar a partir de la información que se maneja, como por ejemplo las clases controladoras y las del modelo. Por ejemplo la clase **ADO\_GestionarProcesoImpl** será la responsable de efectuar las operaciones adicionar, modificar y eliminar los conceptos asociados a un Proceso.
- **Creador:** Este patrón se evidencia en las clases encargadas de crear objetos, por ejemplo la clase **ActivitySet** es responsable de crear instancias de las clases **Actividad** y **Transición**.
- **Bajo acoplamiento:** El uso de este patrón se evidencia en el caso de la relación que se establece entre la clase controladora **CC\_GestionarProcesoImpl** y la clase **Proceso**. En este caso, la clase

controladora no depende de un proceso, porque recibe como parámetro para las operaciones de adición o eliminación un proceso, y los cambios que hayan en la clase proceso no afecta la manera en que la clase controladora gestiona los proceso toda vez que sigue recibiendo como parámetro un proceso, y los cambios en la clase controladora no afectan al proceso.

- **Alta cohesión:** este patrón fue utilizado para agrupar las clases por requerimientos, de modo que las pertenecientes a los requisitos funcionales de tipo gestionar se encargan solamente de adicionar, modificar y eliminar datos, no realizan otra operación, y las clases pertenecientes al requisito funcional Generar Script se encargan solamente de realizar esta operación, de modo que cada clase implementa las operaciones que se encuentran en su misma área funcional asegurando así que no realicen un trabajo enorme.

### **Patrones GOF (Gang of Four, Pandilla de los cuatros):**

- **Mediador:** este patrón se aplicó en la base de datos, cuando existen dependencias entre las tablas que la componen resulta difícil acceder a un determinado valor. La solución a este inconveniente viene dada por la utilización de un mediador que sería una nueva tabla entre todas las tablas que tengan una relación de muchos a muchos con otra. Esta tabla mediadora va a poseer una relación de uno a muchos con las vinculadas a ella. De esta forma, se eliminan las relaciones de muchos a muchos existentes y el comportamiento distribuido entre las clases queda adaptado a las circunstancias y necesidades del diseño.

Por ejemplo, las tablas Actividad y Responsables tienen relación de muchos a muchos, de ahí que surja la tabla mediadora ActividadResponsables.

### **3.9 Resultados de aplicar las métricas del diseño**

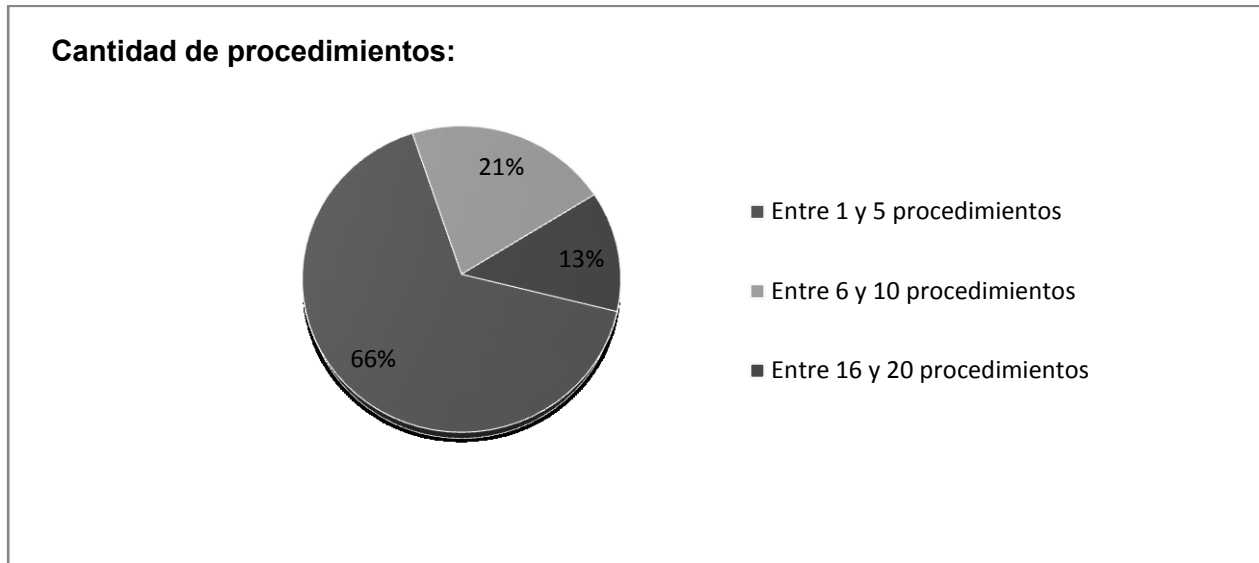
Con el objetivo de determinar el grado de calidad y fiabilidad del diseño que se propone se establecieron algunas métricas de diseño basadas en clases para medir categorías tales como tamaño, herencia y nivel de profundidad, ya que son aspectos orientados al código, a la cohesión, al acoplamiento y la reutilización. A continuación se muestran las gráficas con los resultados de la aplicación de las métricas:

- **Métrica Tamaño operacional de la clase (TOC):**

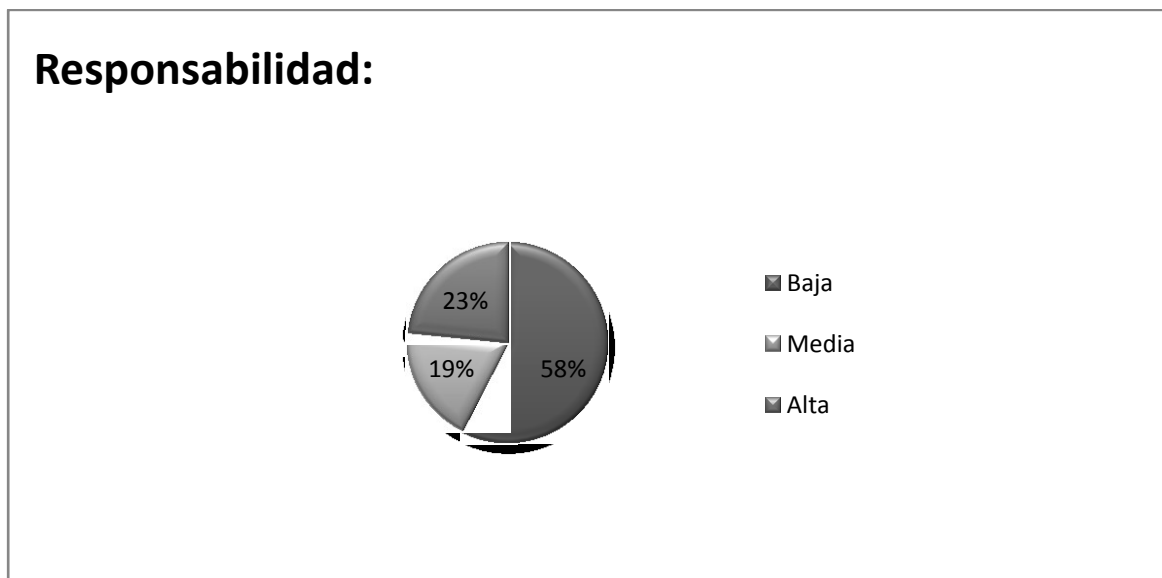
## CAPÍTULO 3

---

Se utilizó para saber el tamaño de una clase conociendo el número total de operaciones, tanto las heredadas como privadas de la instancia, que están encapsuladas dentro de la clase. Con esta métrica se evalúa la Responsabilidad, la Complejidad de implementación y la Reutilización.



**Figura 9 Cantidad de procedimientos**



**Figura 10 Responsabilidad por clases**

### Complejidad de implementación:

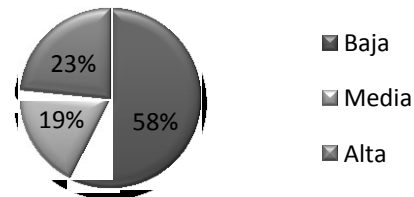


Figura 11 Complejidad de implementación de las clases

### Reutilización:

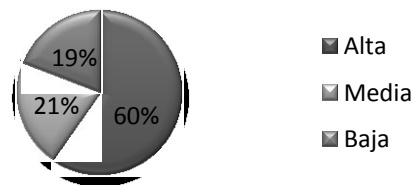


Figura 12 Nivel de reutilización de las clases

En estas gráficas se evidencia que la cantidad de procedimientos por clase es buena, ya que las clases que poseen mayor número se encuentran en el rango de 16 a 20, representando sólo un 13% del total de las clases. La Responsabilidad de las clases se manifiesta también de forma positiva, siendo el 23% del total evaluadas de Alta. Esto mismo sucede en el caso de la Complejidad de implementación. Y la Reutilización se comporta Baja sólo en el 19% del total de las clases.

### **3.10 Conclusiones parciales**

El diseño del sistema ha sido realizado con el objetivo de responder correctamente todos los requisitos del cliente y para guiar la implementación. Se pueden encontrar los patrones de diseño que se van a emplear en la solución y además los artefactos más importantes como son los diagramas de clases y de clases del diseño y el modelado de la base de datos.

### **Conclusiones Generales**

El modelado de los procesos de negocios es una actividad de gran importancia para la comprensión de actividades de una organización. Una organización que desee tener éxitos debe adoptar un sistema de gestión de la calidad, los que en principio establecen que las actividades que se realizan en una entidad tendrían mejores resultados si se modelan como un proceso. La metodología TOT se presenta como una solución a la gestión organizacional de manera efectiva permitiendo que se logre tener un entendimiento común de las principales actividades que se desarrollan en una entidad. Esta metodología incluye entre sus enunciados el uso de un Sistema de Modelado de Procesos de Negocio (BPMS) para que se logre mayor comprensión de los procesos y para que los mismos se puedan optimizar, mejorar y adaptar. Sin embargo, presenta como mayor dificultad que para cada proceso que se describe en los artefactos de la metodología no se administran ni se pueden consultar de manera automática los elementos que lo componen. El análisis y el diseño de una herramienta que permita realizar lo anteriormente explicado ha sido el principal objetivo de este trabajo que aquí se presenta, y ha incluido para su realización el estudio de la gestión por procesos, el estudio de la notación empleada para el modelado de procesos de negocio, las principales herramientas que se utilizan, las metodologías de desarrollo de software, los artefactos que se producen, así como el diseño lógico de una base de datos a través de la cual se puedan gestionar los elementos más importantes presentes en los procesos que se gestionan.

## Recomendaciones

Se recomienda:

- Realizar la implementación de la herramienta haciendo uso del análisis y el diseño resultante de este trabajo de diploma para que sea utilizada para la Gestión y el Modelado de los procesos de negocio.
- Seguir mejorando la metodología TOT para que se adapte más a los elementos de BPMN, una vez realizado esto se hará necesario añadir otras funcionalidades a la herramienta.



## Trabajos citados

1. Enfoque basado en procesos y la norma ISO 9000. [En línea] [Citado el: 1 de noviembre de 2009.] <http://normas-iso-9000.blogspot.com/2007/11/enfoque-basado-en-procesos-y-la-norma.html>.
2. **Flores, Mariano.** *La metodología TOT- La Mejora en la Gestión de los Procesos Empresariales.* 2010.
3. —. Ficha de Captura de la Información Primaria (FCIP). 2010.
4. —. Ficha de Modelación de Procesos (FMP). 2010.
5. —. *Ficha de Aprobación de Mejoras (FAM).* 2010.
6. *Introducción a un BPM.* 2008.
7. **Barrientos, Manuel Sánchez.** Aprender Gratis. [En línea] 2 de noviembre de 2008. [Citado el: 12 de diciembre de 2009.] [www.aprendergratis.com](http://www.aprendergratis.com).
8. **Pérez, Juan Diego.** *Notaciones y lenguajes de procesos. Una visión global.* Sevilla : s.n.
9. BPEL Business Process Execution Language. [En línea] [Citado el: 2 de noviembre de 2009.] <http://www.gestiopolis.com/delta/ads/incrustado-AS-delta-1.htm>.
10. **D., José Guillermo Espina.** *Estado del Arte Suites de BPM Open Source.*
11. **Rumbaugh, James, Jacobson, Ivar and Booch, Grady.** 2000. *El Lenguaje Unificado de Modelado. Manual de Referencia.* 2000.
12. **International, V.P.** 2006. *Introducción a los sistemas y herramientas CASE.* 2006.
13. **FOUNDATION, F. S.** La Definición de Software Libre. *La Definición de Software Libre.* [En línea] [Citado el: 30 de 01 de 2010.] <http://www.gnu.org/philosophy/free-sw.es.html>.
14. <http://www.postgresql-es.org/>. [En línea]
15. **ERP.** *Proceso de Desarrollo y Gestión de Proyectos de Software.* 2009.
16. **IEEE.** *Standards Collection: Software Engineering.* 1993.
17. **Arias, Michael Chaves.** La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. [En línea] 2006. [Citado el: 12 de enero de 2010.] [http://intersedes.ucr.ac.cr/pdfs\\_10/10-art\\_11.pdf](http://intersedes.ucr.ac.cr/pdfs_10/10-art_11.pdf).

## TRABAJOS CITADOS

---

18. **López, Jorge Quesada.** La Informática y la Empresa. . [En línea] 2004. [Citado el: 21 de noviembre de 2009.] <http://www.fec.uh.cu/info3/RECOPILACION.htm>.
19. **González, Jose Luis Hernández.** Instituto Tecnológico de Apisaco. [En línea] <http://www.itapizaco.edu.mx/~joseluis/apuntes/estadistica/definiciones%20y%20muestreos.pdf..>
20. **Nora Escalona, María José and Koch.** *Ingeniería de Requisitos en Aplicaciones para la Web-Un estudio comparativo.* 2002.
21. Interfaz de usuario. [En línea] [Citado el: 12 de febrero de 2010.] <http://www.fismat.umich.mx/~crivera/tesis/node6.html>.
22. **autores, Colectivo de.** *METRICAS TECNICAS DEL SOFTWARE.* s.l. : UCI.
23. Intalio community. [En línea] [Citado el: 15 de diciembre de 2009.] <http://community.intalio.com/>.

## Bibliografía

1. **ERP.** *Proceso de Desarrollo y Gestión de Proyectos de Software.* 2009.
2. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison Wesley, 2000.
3. **FOUNDATION, F. S.** La Definición de Software Libre. *La Definición de Software Libre.* [En línea] [Citado el: 30 de 01 de 2010.] <http://www.gnu.org/philosophy/free-sw.es.html>.
4. Metodología XP Vs. Metodología RUP. [En línea] [Citado el: 15 de 02 de 2010.] <http://metodologíaxpvsmetodologíarup.blogspot.com/>.
5. **Adonis.** ADONIS: community Edition. [En línea] [Citado el: 09 de 02 de 2010.] <http://www.es.adonis-community.com>.
6. **AuraPortal.** AuraPortal. [En línea] [Citado el: 08 de 02 de 2010.] <http://www.auraportal.com>.
7. **JAVA.** Conozca más sobre la tecnología Java. [En línea] [Citado el: 11 de 02 de 2010.] <http://www.java.com>.
8. PROGRAMACIÓN, L. D. *Programación Java, 2006.* [En línea] [Citado el: 12 de 12 de 2009.] <http://www.lenguajes-de-programacion.com/programacion-java.shtml>.
9. ROMÁN, C. A. *La Tecnología Java, 2007.* [En línea] [http://profesores.fi-b.unam.mx/carlos/java/java\\_basico1\\_1.html](http://profesores.fi-b.unam.mx/carlos/java/java_basico1_1.html).
10. **Adonis.** ADONIS: community Edition. *ADONIS: community Edition.* [En línea] [Citado el: 09 de 02 de 2010.] <http://www.es.adonis-community.com>.
11. **AuraPortal.** AuraPortal. *AuraPortal.* [En línea] [Citado el: 08 de 02 de 2010.] <http://www.auraportal.com>.
12. **JAVA.** Conozca más sobre la tecnología Java. *Conozca más sobre la tecnología Java.* [En línea] [Citado el: 11 de 02 de 2010.] <http://www.java.com>.
13. **WfMC.** Workflow Management Coalition. *Xpdl tools.* [En línea] [Citado el: 12 de 03 de 2010.] <http://www.wfmc.org/standards/xpdl.htm>.
14. **Pérez, Juan Diego.** *Notaciones y lenguajes de procesos. Una visión global.* Sevilla : s.n.

## BIBLIOGRAFÍA

---

15. <http://www.postgresql-es.org/>. [En línea]
16. **Flores, Mariano.** *La metodología TOT- La Mejora en la Gestión de los Procesos Empresariales.* 2010.
17. —. Ficha de Captura de la Información Primaria (FCIP). 2010.
18. —. Ficha de Modelación de Procesos (FMP). 2010.
19. —. *Ficha de Aprobación de Mejoras (FAM).* 2010.
20. *Introducción a un BPM.* 2008.
21. **Barrientos, Manuel Sánchez.** Aprender Gratis. [En línea] 2 de noviembre de 2008. [Citado el: 12 de diciembre de 2009.] [www.aprendergratis.com](http://www.aprendergratis.com).
22. XPD.org. *Welcome to XPD.* [En línea] [Citado el: 14 de noviembre de 2009.] XPD.org.
23. **Baeyens, Tom.** JBoss jBPM Cumple WfMC: Tienen mi respeto! . [En línea] [Citado el: 11 de diciembre de 2009.]
24. XPD Support and Resources. [En línea] [Citado el: 14 de noviembre de 2009.] <http://www.wfmc.org/xpd.html>.
25. **D., José Guillermo Espina.** *Estado del Arte Suites de BPM Open Source.*
26. **Pin Nie, Riku Seppälä, Måns Hafrén.** *Open Source Power on BPM - A Comparison of JBoss jBPM and Intalio BPMS.*
27. **Gianni, Renato de Laurentiis.** *BPMS orquestación y agilidad empresarial.*
28. **autores, Colectivo de.** *Business Process Management Systems.* 2009.
29. SOA Agenda. *Qué es BPM, qué es BPMS.* [En línea] [Citado el: 12 de diciembre de 2009.] <http://soaagenda.com/journal/articulos/que-es-bpm-que-es-bpms/>.
30. Azurian. *BPM.* [En línea] [Citado el: 13 de diciembre de 2009.] <http://www.azurian.com/Colombia/productos/producto4.htm>.
31. **Martín, Angel Fernández.** *Madurez del mercado BPM.*
32. **Javier Luis Cánovas Izquierdo, Óscar Sánchez Ramón, Jesús García Molina, Carlos Castillo Alarcón.** *Un caso de estudio para la adopción de un BPMS.*
33. **EMALDIA, JAIME CRISTIAN ACEVEDO.** *Criterios de selección para las herramientas de orquestación de servicios web.*

## BIBLIOGRAFÍA

---

34. Intalio community. [En línea] [Citado el: 15 de diciembre de 2009.] <http://community.intalio.com/>.
35. **Rumbaugh, James, Jacobson, Ivar and Booch, Grady. 2000.** *El Lenguaje Unificado de Modelado. Manual de Referencia. 2000.*
36. **International, V.P. 2006.** *Introducción a los sistemas y herramientas CASE. 2006.*
37. Historia del lenguaje java. [En línea] [http://www.cad.com.mx/historia\\_del\\_lenguaje\\_java.htm](http://www.cad.com.mx/historia_del_lenguaje_java.htm).
38. **López, Jorge Quesada.** La Informática y la Empresa. . [En línea] 2004. [Citado el: 21 de noviembre de 2009.] <http://www.fec.uh.cu/info3/RECOPILACION.htm>.
39. **González, Jose Luis Hernández.** Instituto Tecnológico de Apisaco. [En línea] <http://www.itapizaco.edu.mx/~joseluis/apuntes/estadistica/definiciones%20y%20muestreos.pdf>.
40. **Nora Escalona, María José and Koch.** *Ingeniería de Requisitos en Aplicaciones para la Web-Un estudio comparativo. 2002.*
41. **IEEE.** *Standards Collection: Software Engineering. 1993.*
42. **Arias, Michael Chaves.** La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. [En línea] 2006. [Citado el: 12 de enero de 2010.] [http://intersedes.ucr.ac.cr/pdfs\\_10/10-art\\_11.pdf](http://intersedes.ucr.ac.cr/pdfs_10/10-art_11.pdf).
43. Origen y evolución de la familia de normas de la Serie IRAM-ISO 9000:2000. [En línea] [Citado el: 1 de noviembre de 2009.] <http://www.slideshare.net/xeitoandres/enfoque-basado-en-procesos-presentation>.
44. Enfoque basado en procesos y la norma ISO 9000. [En línea] [Citado el: 1 de noviembre de 2009.] <http://normas-iso-9000.blogspot.com/2007/11/enfoque-basado-en-procesos-y-la-norma.html>.
45. BPEL Business Process Execution Language. [En línea] [Citado el: 2 de noviembre de 2009.] <http://www.gestiopolis.com/delta/ads/incrustado-AS-delta-1.htm>.
46. Interfaz de usuario. [En línea] [Citado el: 12 de febrero de 2010.] <http://www.fismat.umich.mx/~crivera/tesis/node6.html>.
47. **autores, Colectivo de.** *METRICAS TECNICAS DEL SOFTWARE.* s.l. : UCI.
48. Mundo Informático. *PATRONES GRASP (Patrones de Software para la asignación General de Responsabilidad).* [En línea] [Citado el: 23 de mayo de 2010.] <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>.