



03063
16

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**“ADAPTACIÓN DEL PROCESO UNIFICADO
EN LOS FLUJOS DE TRABAJO
DE REQUERIMIENTOS, ANÁLISIS Y DISEÑO:
CASO PRÁCTICO”**

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRA EN INGENIERÍA
(COMPUTACIÓN)**

P R E S E N T A:

MARÍA GUADALUPE SÁNCHEZ MENDOZA

**DIRECTORA DE TESIS: M. EN C. MARÍA GUADALUPE E.
IBARGÜENGOITIA GONZÁLEZ**

MÉXICO, D.F.

TESIS CON
FALLA DE ORIGEN

2003.

A



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

PAGINACIÓN DISCONTINUA

A Dios

Por darme la maravillosa oportunidad de vivir

A mis padres y hermanas

Por su gran apoyo

Autorizo a la Dirección General de Bibliotecas de la
UNAM a difundir en formato electrónico e impreso el
contenido de mi trabajo receptorial.

NOMBRE: Ma. Guadalupe
Sánchez Mendoza

FECHA: 12 - Junio - 2003

FIRMA: [Firma]

Agradecimientos:

A la M. en C. Ma. Guadalupe E. Ibarguengoitia González

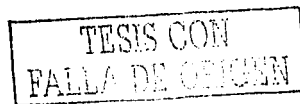
Por su guía, dedicación y entrega para hacer posible esta tesis.

A mis sinodales Dra. Hanna Oktaba, Dr. Fernando Gamboa Rodríguez, Ing. Mario Rodríguez Manzanera y M. en C. Reynaldo Alanís Cantú, por sus valiosos consejos.

A Pedro Velasco, Ana Yuri Ramírez y José Antonio Salazar por su amistad, paciencia e ideas para la conclusión de este trabajo en términos favorables.

Al Dr. Alejandro Cuevas

Por sus valiosas recomendaciones.



Ma. Guadalupe Sánchez Mendoza.

ÍNDICE

INTRODUCCIÓN.....	1
1 MODELADO DE PROCESOS DE SOFTWARE.....	3
1.1 El modelado de procesos.....	3
1.2 Marco conceptual del modelado de procesos de software.....	4
1.3 Características del modelado de procesos de software.....	5
1.4 Objetivos del modelado de procesos de software.....	6
1.5 Paradigmas del modelado de procesos de software.....	7
2 PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE, LENGUAJE DE MODELADO UNIFICADO Y RUP.....	9
2.1 El Proceso Unificado.....	9
2.2 Características del Proceso Unificado.....	10
2.3 Ciclo de vida del Proceso Unificado.....	13
2.4 Fases del Proceso Unificado.....	13
2.5 Flujos de trabajo del Proceso Unificado.....	16
2.6 Artefactos del Proceso Unificado.....	19
2.7 Lenguaje de Modelado Unificado (UML)	19
2.8 RUP (Rational Unified Process)	22
3 DEPARTAMENTO DE INFORMÁTICA DEL INSTITUTO DE INVESTIGACIONES BIBLIOGRÁFICAS	27
3.1 Antecedentes del Instituto de Investigaciones Bibliográficas.....	27
3.2 Estructura organizacional del Instituto de Investigaciones Bibliográficas.....	27
3.3 Funciones del Instituto de Investigaciones Bibliográficas.....	39
3.4 Panorama del Departamento de Informática.....	30
3.5 El proceso de desarrollo de software del Departamento de Informática.....	33
3.6 Hipótesis.....	34
3.7 Objetivos.....	34

TESIS CON
FALLA DE ORIGEN

4	ADAPTACIÓN DEL FLUJO DE TRABAJO DE REQUERIMIENTOS DEL PROCESO UNIFICADO AL DEPARTAMENTO DE INFORMÁTICA.....	37
4.1	Definición, clasificación y objetivos de los requerimientos.....	37
4.2	Flujo de requerimientos del Proceso Unificado.....	39
4.3	Adaptación del flujo de requerimientos del Proceso Unificado al Departamento de Informática.....	42
4.4	Comparación del flujo de requerimientos entre el Proceso Unificado y el Departamento de Informática.....	46
4.5	Detalle de las actividades de requerimientos en el Departamento de Informática.....	50
5	ADAPTACIÓN DEL FLUJO DE TRABAJO DE ANÁLISIS Y DISEÑO DEL PROCESO UNIFICADO AL DEPARTAMENTO DE INFORMÁTICA.....	65
5.1	Concepto y objetivo del análisis.....	65
5.2	Flujo de análisis del Proceso Unificado.....	66
5.3	Adaptación del flujo de análisis del Proceso Unificado al Departamento de Informática.....	69
5.4	Comparación del flujo de análisis entre el Proceso Unificado y el Departamento de Informática.....	72
5.5	Detalle de las actividades del análisis en el Departamento de Informática.....	75
6	ADAPTACIÓN DEL FLUJO DE TRABAJO DE DISEÑO DEL PROCESO UNIFICADO AL DEPARTAMENTO DE INFORMÁTICA.....	85
6.1	Concepto y objetivo del diseño.....	85
6.2	Flujo de diseño del Proceso Unificado.....	86
6.3	Adaptación del flujo de diseño del Proceso Unificado al Departamento de Informática.....	89
6.4	Comparación del flujo de diseño entre el Proceso Unificado y el Departamento de Informática.....	93
6.5	Detalle de las actividades del diseño en el Departamento de Informática.....	97
7	CONSTRUCCIÓN DE LA HERRAMIENTA "PDSIIB" (PROCESO DE DESARROLLO DE SOFTWARE DEL INSTITUTO DE INVESTIGACIONES BIBLIOGRÁFICAS).....	117
7.1	Definición y objetivos de la herramienta PDSIIB.....	117

7.2	Proceso de creación de la herramienta PDSIIB.....	118
7.3	Organización de la herramienta PDSIIB.....	120
7.4	Manejo de la herramienta PDSIIB.....	121
CONCLUSIONES.....		127
	Trabajos futuros.....	129
GLOSARIO.....		131
BIBLIOGRAFÍA		139
APÉNDICE A. Estándar de documentación		
APÉNDICE B. Estándar de requerimientos		
APÉNDICE C. Estándar de análisis y diseño		
APÉNDICE D. Estándar de diseño de base de datos		
APÉNDICE E. Artefactos del flujo de requerimientos		
APÉNDICE F. Artefactos del flujo de análisis		
APÉNDICE G. Artefactos del flujo de diseño		
APÉNDICE H. Ejemplo de la aplicación del proceso		

TESIS CON
FALLA DE ORIGEN

INTRODUCCIÓN

Existen diversos modelos y métodos para estudiar y analizar los procesos involucrados en el desarrollo de software; quizá uno de los más actuales e importantes en el ámbito de la ingeniería de software, es el denominado Proceso Unificado, puesto que poco a poco se van sumando más organizaciones que lo toman como guía para la construcción y desarrollo de software. No obstante lo anterior, existen organizaciones que aún no cuentan con una guía en su proceso de desarrollo de software; tal es el caso del Departamento de Informática del Instituto de Investigaciones Bibliográficas. Dicho lo anterior, se justifica la realización de la presente tesis en la que se realiza un estudio y análisis del Proceso Unificado, así como su adaptación a dicho departamento.

El Proceso Unificado contiene cuatro fases (inicio, elaboración, construcción y transición), y nueve flujos de trabajo (modelado del negocio; requerimientos, análisis y diseño, implementación, pruebas, distribución, ambiente, administración de la configuración; y, administración del proyecto), que involucran actividades, roles y artefactos; por lo que para su posible adaptación, se hace necesaria y deseable la división del trabajo. Es así como surge el proyecto llamado "Adaptación del Proceso Unificado a un Departamento de Informática" el cual se ha dividido en tres partes, tres trabajos de tesis independientes, pero a la vez unificados en su contenido, dado que cada tesis contiene de manera independiente la presentación de tres flujos de trabajo adecuándolos al departamento; esto supone la integración de los tres trabajos para posteriormente construir una herramienta que guíe el proceso completo.

Así, el objetivo principal es adaptar el Proceso Unificado en los flujos de requerimientos, análisis y diseño de dicho departamento.

TESIS CON
FALLA DE ORIGEN

Un segundo propósito es, el que una vez lograda la posible adaptación del Proceso Unificado, se construya la herramienta que guíe el proceso de desarrollo de software.

Para la adaptación del Proceso Unificado y la construcción de la herramienta se deben realizar ciertas actividades como: efectuar un análisis de la forma en que se realiza actualmente el proceso de desarrollo de software en el Departamento de Informática; estudiar los flujos de requerimientos, análisis y diseño del propio Proceso Unificado; adaptar y diseñar los flujos para el Departamento de Informática y finalmente construir la herramienta que guíe su aplicación.

El presente trabajo está estructurado de la siguiente forma: Introducción; capítulo 1, describe en forma general el modelado de procesos de software y se enuncian los conceptos, características, objetivos, y paradigmas del mismo; capítulo 2, resume la constitución del Proceso Unificado, UML (Lenguaje de Modelado Unificado) y RUP (Rational Unified Process); capítulo 3, trata del Departamento de Informática del Instituto de Investigaciones Bibliográficas; en los capítulos 4, 5 y 6 se detalla la adaptación de los flujos de requerimientos, análisis y diseño respectivamente; capítulo 7, detalla la estructura de la herramienta del proceso de desarrollo de software del Instituto de Investigaciones Bibliográficas; y, finalmente se incluyen las conclusiones, glosario, bibliografía y apéndices.

CAPÍTULO 1

1 MODELADO DE PROCESOS DE SOFTWARE.

A fin de comprender el modelado de procesos de software, el capítulo se encuentra dividido en cinco secciones. El modelado de procesos integra la primera de estas, en donde se presenta un esbozo general de lo que es el modelado; en la segunda se define puntualmente el significado de los conceptos y/o terminología utilizada en este dominio; la tercera denota sus características particulares; la cuarta enuncia sus objetivos, para finalmente en la quinta mencionar a groso modo la evolución del proceso de modelado de procesos de software en conjunto.

1.1. El modelado de procesos

En los inicios de la computación, al establecerse la importancia de dividir el trabajo de desarrollo en diferentes procesos, el enfoque a su modelado consistía únicamente en el análisis del flujo de la información utilizada por estos. La utilización de las computadoras en las tecnologías de información se encontraba restringida al procesamiento de transacciones dado que no se tomaban en cuenta todos los elementos que intervenían en los procesos, los cuales sólo se consideraban en forma independiente. La utilidad del modelado de procesos radica esencialmente en considerar todos los elementos que intervienen en el proceso y describirlos en forma detallada y precisa. Este enfoque inicial del modelado de procesos ha ido evolucionando necesariamente debido al continuo cambio que enfrentan las organizaciones.

Poco a poco se fueron integrando y aislando a la vez nuevos procesos, los cuales podían ser modelados independientemente, tales como: La reingeniería de procesos, la tecnología y los ambientes de desarrollo de software.

El primero se encarga del rediseño de los procesos de negocio de las organizaciones con el fin de hacerlos más eficientes; el segundo auxilia a los negocios en el manejo de las dependencias existentes entre los agentes que participan en sus procesos y provee soporte automático para los componentes de los procesos rutinarios; y, el último, que constituye un sistema que integra el trabajo de todos los elementos relacionados en el proceso de desarrollo de software.

1.2. Marco conceptual del modelado de procesos de software

A fin de poder establecer un solo lenguaje incluiremos a continuación el siguiente glosario de términos utilizados en el modelado de procesos de software. Al final del trabajo se incluye un glosario de términos completo sobre el vocabulario que se utiliza en todo el documento.

Modelo

Un modelo es una representación abstracta de la realidad. El propósito de un modelo es describir y entender el comportamiento de los fenómenos, para poder así predecir su comportamiento.

Proceso

Un proceso es un conjunto ordenado de pasos a seguir para alcanzar un objetivo.

Paso

Una acción que no puede dividirse en más acciones, por lo que se le considera atómica.

Actividad

Ejecución en curso, la conforman más de un paso. En el modelado de procesos de desarrollo de software se realiza una labor ardua en cuanto a la especificación y descripción de todas las partes que intervienen en el proceso, como lo es la

determinación de: actores, actividades, artefactos, elementos que definimos a continuación:

Actor o agente

Un agente puede definirse como una entidad que realiza una acción de un proceso.

Recurso

Un recurso es una entidad que puede ser requerida para realizar una acción de un proceso.

Artefacto

En términos simples podemos definir un artefacto como la salida o la entrada de un proceso.

Modelo de un proceso

Es la descripción de un proceso en términos de un grupo de actividades que son llevadas a cabo por actores utilizando recursos y generando artefactos.

1.3. Características del modelado de procesos de software

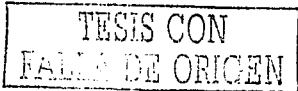
Entre las características específicas de los procesos de software pueden ubicarse las siguientes :

Concurrencia y distribución (Procesamiento Paralelo)

Los procesos de software se consideran ser concurrentes debido a que ejecutan o realizan múltiples tareas en paralelo, por lo que para su proceso es necesario disponer de múltiples equipos para que los ejecuten; esto es a lo que actualmente y ya en forma común se denomina procesamiento en paralelo.

Incertidumbre y no-determinismo

Esto se refiere básicamente a que durante la ejecución del proceso, algunas etapas no son factibles de predecirse por lo que existe incertidumbre ocasionando



a la vez que las alternativas que se toman sean no deterministas, es decir, alternativas no predeterminadas.

Errores, replanteamiento y otras contingencias

Durante el proceso de desarrollo de software se deben incluir aspectos como: creación, inspección, pruebas y revisión del desarrollo del software. Esto significa que es factible detectar errores realizados durante el proceso de creación, mismos que pueden ser corregidos, replanteados o bien incluir las posibles contingencias que se puedan presentar.

Evolución y cambio

Los procesos de software comprenden un "ciclo de vida" que incluye: especificación de requerimientos, análisis, diseño, implementación, pruebas; y, liberación y evaluación, lo cual significa que los procesos de software están expuestos a cambios continuos y que este ciclo de vida permita realizar los cambios necesarios pertinentes para que los procesos estén de acuerdo a las necesidades.

1.4. Objetivos del modelado de procesos de software

El entendimiento, la mejora y el análisis son los objetivos del modelado de procesos.

El entendimiento consiste en definir clara y perfectamente el rol de cada persona, en el proceso, las actividades que debe realizar y los artefactos que se deben genera.

La mejora significa el perfeccionamiento de los procesos realizando cambios adecuados, continuos y necesarios con base en la experiencia y las lecciones aprendidas.

El análisis es fundamental para el modelado correcto de los procesos de software tiene como objetivo el ubicar exactamente todos los elementos que intervienen en el proceso, tales como: actores, actividades, recursos y artefactos; del análisis depende el alcance del proceso de software.

1.5. Paradigmas del modelado de procesos de software

Cabe hacer notar que la utilización de modelos para la representación de los procesos de software no representa algo novedoso puesto que se ha venido utilizando desde hace varios años; se ha demostrado que los modelos son muy útiles para entender de manera clara y específica el comportamiento de cada elemento que interviene en el proceso de desarrollo. Así, existen varios paradigmas que pueden ser utilizados para el modelado de procesos de software, los cuales se han ido creando en base a las necesidades de cada proceso. A continuación se mencionan y describen brevemente algunos de ellos.

Modelos de programación

Esta abstracción, parte de la observación de que la especificación de un proceso es una forma de programación, por lo tanto un proceso puede ser modelado con todas las técnicas y herramientas de los programadores.

Modelos funcionales

En éste paradigma un proceso es representado como una colección de elementos con atributos de entrada y de salida. Específicamente, un proceso se define como un conjunto de funciones matemáticas que representan relaciones entre las entradas y las salidas. Además, cada una de estas funciones puede ser descompuesta jerárquicamente en sub-elementos del proceso donde los atributos de entrada y salida de un elemento padre deben ser satisfechos por los atributos de sus hijos.

TESIS CON
FALLA DE CARGEN

Modelos basados en planes

El paradigma provee mecanismos donde los operadores representan posibles acciones que son seleccionadas en base a sus precondiciones. Estos operadores son aplicados al estado actual del domino en el que el proceso opera, con el fin de acercar más ese estado al objetivo deseado.

Modelos de Redes de Petri

La aplicación de este paradigma modela la estructura de interacción de roles de un proyecto usando un lenguaje y una representación basados en redes, en particular Redes de Petri. Las redes de interacción de roles ayudan a la representación y ejecución de tareas estructuradas, que son aquellas que pueden ser planeadas con dependencias conocidas.

Modelos cuantitativos

Los modelos construidos intentan definir un conjunto de relaciones cuantitativas entre variables de interés que simulan el comportamiento observado del sistema. En el estudio de sistemas dinámicos es una de las pocas técnicas utilizadas de modelado dado que involucra representaciones cuantitativas, y aplica la retroalimentación y algunas técnicas de sistemas de control a fenómenos sociales e industriales.

Así, como puede observarse el modelado de procesos de software no es un tema nuevo es más bien un aspecto importante que se ha ido fortaleciendo conforme la situación cambiante del entorno de los procesos de software, hoy en día se siguen creando modelos de procesos de software para mejorar y hacer frente a lo que se ha dado por llamar cambio continuo. Sin embargo todos los modelos coinciden en ayudar a entender perfectamente el proceso identificando actores, actividades, artefactos y/o recursos. Tal es el caso del tema que nos ocupa, el Proceso Unificado, un modelo para procesos de desarrollo de software, al cual trataremos en detalle en el siguiente capítulo.

CAPÍTULO 2

2. PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE, LENGUAJE DE MODELADO UNIFICADO Y RUP.

En el presente capítulo se describe como esta constituido el Proceso Unificado enunciando sus características, ciclo de vida, fases, flujos y artefactos. También se puntualiza la relación entre el Proceso Unificado con respecto al Lenguaje de Modelado Unificado y RUP (Rational Unified Process).

2.1. El Proceso Unificado

“El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software, es el conjunto de actividades necesarias para transformar los requerimientos de un usuario en un sistema de software”. [Jacobson, 2000. p. 4.]. (Ver figura 2.1).

El Proceso Unificado de desarrollo de software surge de la unificación de las metodologías OMT (Object Modelling Technique), Objectory (Object Factory) y la de Booch, desarrolladas por James Rumbaugh, Ivar Jacobson y Grady Booch quienes también son los diseñadores originales del Lenguaje de Modelado Unificado.

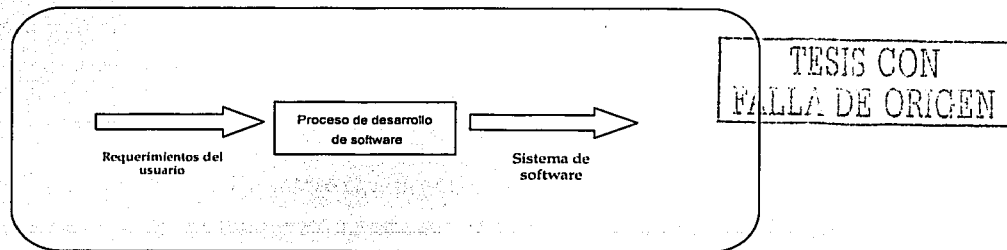


Figura 2.1 Modelo de un proceso de desarrollo de software

Según los creadores del Proceso Unificado, éste es mas que un simple proceso de software, ellos dicen que el proceso es "general" y puede ser utilizado en los diversos sistemas de software existentes para diferentes áreas de aplicación y distintos tipos de organizaciones.

2.2. Características del Proceso Unificado

El Proceso Unificado básicamente parte de tres ideas principales: está dirigido por casos de uso, se centra en la arquitectura, y es iterativo e incremental.

A continuación se explica el significado cada una de las características.

2.2.1. El Proceso Unificado está dirigido por casos de uso

El punto esencial para garantizar la funcionalidad de un sistema es conocer exactamente lo que los usuarios necesitan, desean y esperan del sistema.

En términos de ingeniería de software un usuario es alguien (persona) o algo (otro sistema) que interactúa con el sistema y es el que obtiene un resultado específico de sistema.

Interacción

A fin de ejemplificar la interacción que existe entre usuario y sistema supóngase una persona que captura información para que se almacene en una base de datos; primero la información se captura, luego se da la indicación de guardar y, finalmente el sistema indica a través de una leyenda que la información ha sido guardada o almacenada.

En este ejemplo el usuario y sistema interactúan llevando a cabo una secuencia de acciones lógicas y secuenciales que proporcionan un resultado específico. A este proceso o interacción se le llama "caso de uso".

En este caso de uso, el resultado es que la información ha sido almacenada.

Especificación Funcional

Puede decirse que una especificación funcional contesta a la pregunta: "¿qué debe hacer el sistema?" [Jacobson, 2000. p. 5].

"Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante." [Jacobson, 2000. p. 5].

Los casos de uso representan los requerimientos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso, el cual describe la funcionalidad total del sistema.

La importancia de los casos de uso radica no solo en establecer claramente los requerimientos de un sistema, sino también, la capacidad de guiar su diseño, implementación y pruebas, o lo que es lo mismo guiar el proceso de desarrollo de software.

2.2.2. El Proceso Unificado centrado en la arquitectura

El papel de la arquitectura en el proceso de desarrollo de software es equiparable a una obra arquitectónica para la construcción de una unidad habitacional; cuando se inicia un proyecto de éste tipo se deben tomar en cuenta factores como: drenaje, electricidad, vialidad, decoración, servicios, etc. La idea es que un arquitecto logre plasmar una imagen completa del proyecto de construcción antes de que se inicie la obra. Igualmente en el proceso de desarrollo de software se deben tomar en cuenta todos los elementos que intervienen en el desarrollo del sistema a fin de poder tener una vista integral que sea entendible para los usuarios y desarrolladores antes de llevar a cabo la obra. En nuestro caso los elementos a considerar son los requerimientos funcionales y no funcionales del sistema.

La importancia de la arquitectura radica en tener una vista integral del sistema. Ahora bien, los casos de uso están totalmente relacionados con la arquitectura; dado que "los casos de uso guían la arquitectura del sistema y la arquitectura del sistema influye en la selección de los casos de uso. Por tanto, la arquitectura del sistema como los casos de uso maduran según avanza el ciclo de desarrollo" [Jacobson, 2000. p. 5].

El arquitecto de software debe conocer los casos de uso y demás elementos antes de la creación del esquema arquitectónico, después debe representar cada una de las funciones del sistema especificando en detalle en términos de subsistemas, sus clases y componentes; de esta manera los casos de uso tienden a entenderse y madurarse. Por ende la arquitectura será estable.

2.2.3. El Proceso Unificado es iterativo e incremental

Se refiere a que cuando se está desarrollando un sistema demasiado grande y complejo es muy conveniente dividirlo en partes más pequeñas o mini proyectos. "Cada mini proyecto es una iteración del que resulta un incremento". . [Jacobson, 2000. p. 7.]. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto.

Para una efectividad máxima, las iteraciones deben estar controladas; esto es, deben seleccionarse y ejecutarse de una forma planificada. Por esta razón son considerados como mini proyectos.

La utilidad de que el Proceso Unificado sea iterativo e incremental es que se reducen costos, equivocaciones, riesgos y tiempo de entrega de los productos esperados.

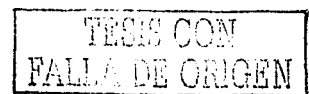
2.3. Ciclo de vida del Proceso Unificado

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen "el ciclo de vida del sistema". Cada ciclo concluye con una nueva versión del producto para los usuarios. Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase se subdivide en iteraciones.

2.4. Fases del Proceso Unificado

Como se mencionó, cada ciclo es desarrollado en un tiempo específico. Un ciclo se divide en cuatro fases; cada fase termina con un indicador, cuyo objetivo es el tomar la decisión para dar inicio o no a la siguiente fase estimando nuevamente tiempos, personas y asignando recursos, es decir, al término de cada fase, los involucrados en el desarrollo del producto se reúnen y toman las decisiones correspondientes de continuar o no con la fase siguiente; ésto con base en los artefactos, criterios y documentos que se han generado en la fase correspondiente.

Es importante señalar que cada fase atraviesa por los flujos de trabajo como son: requerimientos, análisis, diseño, implementación y pruebas. En la figura 2.2 se puede apreciar el ciclo de vida del Proceso Unificado que esta constituido por cuatro fases (parte superior) en donde cada fase debe atravesar por todos los flujos de trabajo (columna izquierda), finalmente para proyectos grandes el sistema se puede dividir en mini-proyectos que son las iteraciones. (Parte inferior de la figura).



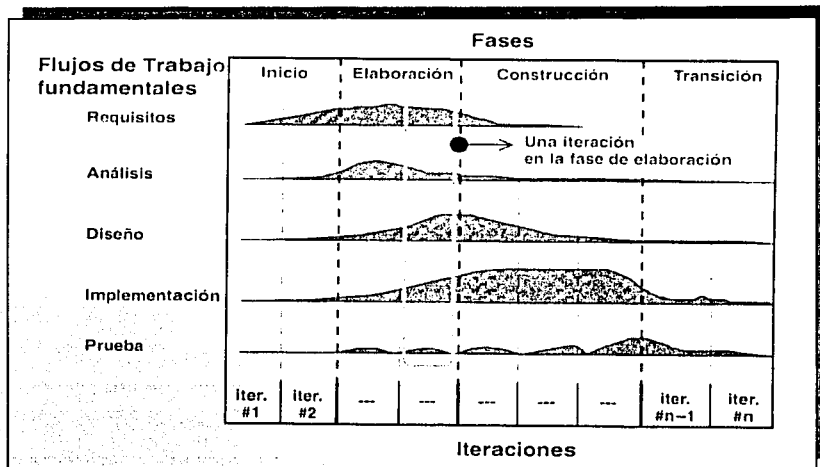


Figura 2.2. Imagen tomada del Proceso Unificado. Los flujos de trabajo, (requerimientos, análisis, diseño, implementación y prueba) tienen lugar sobre las fases del Proceso Unificado, (inicio, elaboración, construcción y transición).

A continuación se describen en forma resumida cada una de las fases y flujos de trabajo.

Las fases que conforman el Proceso Unificado son cuatro: inicio, elaboración, construcción y transición.

2.4.1. Inicio

Durante esta fase se debe realizar una descripción del producto partiendo de una idea. En esta fase se debe responder a las siguientes preguntas:

- ¿Cuáles son las principales funciones del sistema?,

- ¿Cómo podría ser la arquitectura del sistema?,
- ¿Cuál es el plan del proyecto?, y
- ¿Cuánto costará?.

2.4.2. Elaboración

En esta fase se deben especificar detalladamente todos los casos de uso y detallar la arquitectura del sistema. La pregunta a contestar es: ¿Están perfectamente definidos los casos de uso la arquitectura y el plan para comprometerse al desarrollo del sistema sin correr riesgos o bien controlarlos en forma adecuada?.

2.4.3. Construcción

“En esta fase, la línea base de la arquitectura crece hasta convertirse en un producto preparado para ser entregado a la comunidad de usuarios”. [Jacobson, 2000. p.12]. De hecho durante esta fase se emplea la mayoría de los recursos requeridos. La pregunta a contestar es: ¿Cubre el producto los requerimientos mínimos necesarios para hacer una primera entrega?.

2.4.4. Transición

En esta fase, el producto se convierte en una versión beta para que los usuarios puedan probarlo y detectar posibles fallas o deficiencias para poder sugerir los cambios necesarios. Aquí también se decide si se corrigen los defectos en la misma versión o bien si se corregirán en una nueva.

TESIS CON
FALLA DE ORIGEN

2.5. Flujos de trabajo del Proceso Unificado

Los flujos de trabajo del Proceso Unificado son cinco: requerimientos, análisis, diseño, implementación y prueba.

2.5.1. Requerimientos

La idea fundamental en este flujo es determinar cuáles son las necesidades de los usuarios, esta labor a menudo suele ser compleja puesto que a los propios usuarios les cuesta trabajo definir claramente lo que desean.

La mayoría de los usuarios no saben qué parte de su trabajo puede transformarse en software. La verdad es que no saben cuáles son los requerimientos y mucho menos especificarlos en forma precisa. Aquí es fundamental la intervención de los analistas que son quienes desmenuza el problema y tratan de describirlo lo más cercano posible a la realidad.

El objeto fundamental de este flujo de trabajo es guiar el proceso de desarrollo hacia el sistema correcto.

Un reto para conseguirlo es que el usuario, el cual es especialista en su campo y normalmente no informático, deberá ser capaz de leer y comprender el resultado de la captura de los requerimientos. Para que ésto sea posible se deberá utilizar un lenguaje entendible para el usuario. Los diagramas de UML que se utilizarán en este flujo serán los de casos de uso.

2.5.2. Análisis

En el análisis se deben estructurar los requerimientos de manera que se facilite su comprensión, preparación, modificación, y, en general su mantenimiento. En otras palabras en esta parte se debe generar el modelo de análisis el cual debe tener las siguientes características:

- "Estar escrito con el lenguaje del desarrollador"

- **Mostrar la vista interna del sistema**
- **Estar estructurado por clases y paquetes**
- **Ser utilizado principalmente por los desarrolladores para comprender cómo debería darse forma al sistema**
- **Estar libre de redundancias**
- **Presentar un esbozo de cómo llevar a cabo la funcionalidad del sistema**
- **Definir realizaciones de casos de uso". [Jacobson, 2000. p.167].**

El objetivo principal del análisis es identificar claramente paquetes y clases del análisis, y los requerimientos especiales necesarios para llevar a cabo los demás flujos de trabajo. Es por ésto que aquí se hará uso de los diagramas de casos de uso y de clases.

2.5.3. Diseño

En el diseño se debe modelar el sistema incluyendo todos los requerimientos; los funcionales y los no funcionales.

Los objetivos primordiales son:

- **Capturar interfaces entre subsistemas**
- **Visualizar y reflexionar sobre el diseño utilizando una notación común**
- **Crear una abstracción para la implementación del sistema**
- **Elaborar el modelo de diseño que se esfuerce en conservar la estructura del sistema impuesta por el modelo de análisis, y que sirva como base para la implementación.**

El diseño es el centro de atención al finalizar la fase de elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a que la arquitectura sea sólida, estable y permita crear un modelo para la implementación. No obstante, el modelo de diseño esta muy cercano a la implementación por lo que es natural

guardar y mantener el modelo de diseño a través del ciclo de vida del software porque éste puede ser utilizado para técnicas de programación gráfica y para la reingeniería si fuera necesaria.

En el flujo de diseño se utilizan los diagramas de secuencia y/o actividades.

2.5.4. Implementación

En la implementación se empieza con el resultado del diseño y se implementa el sistema en términos de componentes, es decir, código fuente, scrips, ejecutables, etc.

Los propósitos de la implementación son:

- Planificar las integraciones del sistema que son necesarias para cada iteración
- Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue
- Implementar las clases y subsistemas encontrados durante el diseño
- Probar los componentes de manera individual y luego integrados

El resultado principal de esta etapa es el modelo de implementación. Principalmente se utilizarán los diagramas de despliegue.

2.5.5. Prueba

En este flujo de trabajo se verifica el resultado de la implementación probando cada módulo o subsistema integrado.

Los principales objetivos son:

- Planificar las pruebas necesarias en cada iteración, incluyendo las pruebas de integración y de sistema.
- Diseñar e implementar las pruebas creando los casos de prueba que especifiquen qué probar, creando los procedimientos de prueba que detallen cómo realizar las pruebas y, finalmente crear los procedimientos automatizados de prueba.

2.6. Artefactos del Proceso Unificado

Cada ciclo produce una versión del sistema que incluye código fuente, modelos, diagramas, documentos, manuales y otros elementos que se deben ajustar por completo a las necesidades de los usuarios que utilizarán el sistema.

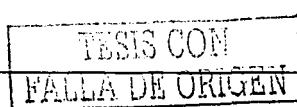
El producto terminado incluye los elementos que permiten a los interesados; clientes, usuarios, analistas, diseñadores, programadores, ingenieros de prueba y directores; especificar, diseñar, implementar, probar y utilizar un sistema. Es más, son esos elementos los que permiten modificar el sistema de versión en versión.

2.7. Lenguaje de Modelado Unificado (UML)

“El Lenguaje de Modelado Unificado (UML, Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema. UML proporciona una forma estándar de describir los planos de un sistema como cosas concretas, tales como las cosas escritas en un lenguaje de programación específico (programas), esquemas de bases de datos y componentes de software reutilizables.” [Booch, 1999. p. XIX].

“El UML es un lenguaje cuyo vocabulario y reglas se centran en la representación conceptual y física de un sistema. Un lenguaje de modelado como UML es por tanto un lenguaje estándar para modelar software.” [Booch, 1999. p. 12].

UML tiene las siguientes características:



- Es un lenguaje para visualizar y para comunicar dado que al realizar un modelo explícito en UML se facilita de inmediato la comunicación.
- Es un lenguaje gráfico bien definido porque detrás de cada elemento gráfico o símbolo existe una semántica bien definida.
- Es un lenguaje para especificar, lo que significa construir modelos precisos, no ambiguos ni incompletos.
- Es un lenguaje para construir porque sus modelos de abstracción pueden conectarse de forma directa a algunos lenguajes de programación.
- Es un lenguaje para documentar porque cubre toda la arquitectura del sistema y demás detalles.

UML consta de nueve tipos de diagramas que se pueden utilizar para visualizar cualquier sistema desde diferentes perspectivas, de forma que un diagrama es una proyección de un sistema.

Un diagrama permite representar una visión resumida de los elementos que conforman un sistema.

A continuación se mencionan los diferentes diagramas que se pueden realizar con UML.

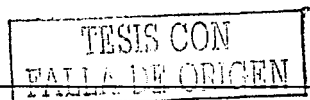
1. **Diagrama de clases;** muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Entendiendo por colaboración un conjunto de elementos, (clases e interfaces), que interactúan conjuntamente para obtener un resultado específico.
2. **Diagrama de objetos;** muestra un conjunto de objetos y sus relaciones. Un objeto es una entidad en la que pueden aplicarse un conjunto de operaciones.
3. **Diagrama de casos de uso;** muestra los casos de uso, actores y sus relaciones.

4. **Diagrama de secuencia;** son aquellos que muestran un conjunto de objetos y sus relaciones; haciendo énfasis en el orden del envío de mensajes entre objetos.
5. **Diagrama de colaboración;** muestra la organización estructural de los objetos que envían y reciben mensajes.
6. **Diagrama de estados;** permite apreciar los diferentes estados por los que atraviesan los objetos durante un proceso, identificando sus transiciones, eventos y actividades.
7. **Diagrama de actividades;** muestra el flujo de actividades dentro de un sistema.
8. **Diagrama de componentes;** muestra la organización y las dependencias entre un conjunto de componentes.
9. **Diagrama de despliegue;** muestra la configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos.

A través de estos diagramas se pueden representar la vista estática y la vista dinámica de los sistemas.

La vista estática se refiere a visualizar la estructura de un sistema, es decir ver los elementos que conforman el sistema.

La vista dinámica se refiere a visualizar el comportamiento de los elementos de un sistema.



**Diagramas para representar la vista
estática de los sistemas**

De clases
De objetos
De componentes
De despliegue

**Diagramas para representar la vista
dinámica de los sistemas**

De casos de uso
De secuencia
De colaboración
De estados
De actividades

Por último, es importante señalar que algunos diagramas de UML se utilizarán para representar la vista estática y dinámica de los sistemas en el proceso de desarrollo de software del Departamento de Informática.

2.8. RUP (Rational Unified Process)

RUP es el resultado de varias aportaciones, no solo de Jacobson, Booch y Rumbaugh sino también de otras empresas como Rational quien adquirió o se fusionó con otros fabricantes de herramientas de desarrollo de software para crear el RUP.

RUP es el Proceso Unificado de Rational; RUP es un software de ingeniería de procesos que permite asignar tareas y responsabilidades en el proceso de desarrollo del software de la organización.

RUP introduce flujos adicionales al Proceso Unificado como son: modelado del negocio; distribución; administración de cambios y configuración; administración del proyecto y ambiente. Por otra parte, conceptualiza en un solo flujo al análisis y el diseño. En la figura 2.3. se pueden apreciar los nueve flujos del RUP.

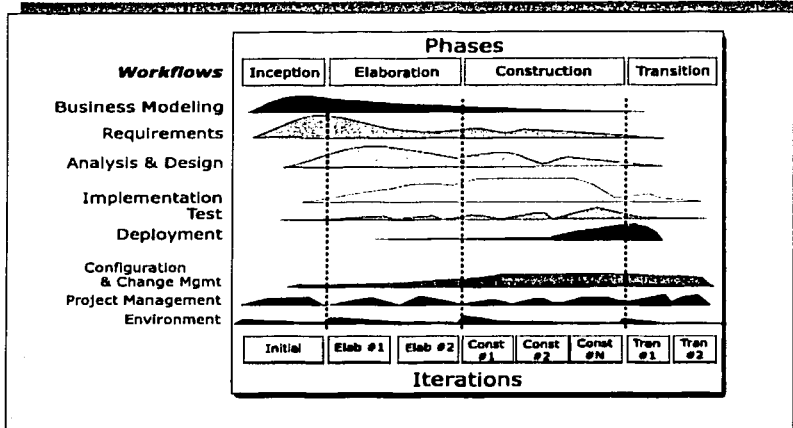


Figura 2.3. Imagen tomada del Proceso Unificado de Rational que incluye nueve flujos y cuatro fases.

A continuación se describen los flujos que introduce el RUP.

2.8.1. Modelado del negocio

El propósito del modelado del negocio es entender la estructura, problemática, dinámica y el tipo de sistemas de la organización; así como los usuarios de la misma.

Es importante destacar que la adaptación de este flujo no se realizó debido a que los aspectos antes mencionados son conocidos por las personas que llevan a cabo el desarrollo de sistemas en el Departamento de Informática; con base en ésto, se realizó la adaptación de los flujos siguientes.

TESIS CON
FALLA DE ORIGEN

2.8.2. Distribución

Este flujo establece las actividades que se deben realizar una vez que el producto de software está disponible para los usuarios.

2.8.3. Administración de cambios y configuración

La finalidad de este flujo es mantener la integridad del proyecto, sistema o desarrollo. Debido a que durante el desarrollo del proyecto muchos artefactos son creados, modificados o actualizados por un trabajador específico y porque normalmente los artefactos son requeridos en más de un flujo, es por ello que se debe tener control sobre los cambios realizados o bien, los cambios requeridos para que siempre se trabaje con los elementos actuales y aprobados.

2.8.4. Administración del proyecto

El objetivo principal de la administración del proyecto radica en proveer los elementos necesarios de planeación, ejecución, monitoreo y control de los proyectos; así como el manejo de riesgos.

2.8.5. Ambiente

El propósito del ambiente consiste en brindar el soporte necesario durante el desarrollo del proyecto con respecto a los procesos y herramientas que sean necesarias.

El ambiente incluye:

- Selección y adquisición de herramientas
- Configuración de herramientas
- Procesos de configuración, y

- **Soporte técnico al proceso**

Para llevar a cabo la adaptación del Proceso Unificado al Departamento de Informática se tomó como referencia los cinco flujos del Proceso Unificado, recordando que en este trabajo, se presenta la adaptación de requerimientos, análisis y el diseño.

No obstante lo anterior , se consultó el RUP para tener una visión más amplia de lo que involucra cada flujo de trabajo.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO 3

3. DEPARTAMENTO DE INFORMÁTICA DEL INSTITUTO DE INVESTIGACIONES BIBLIOGRÁFICAS.

En el presente capítulo se describen algunos aspectos importantes del Instituto de Investigaciones Bibliográficas tales como: antecedentes; estructura organizacional; funciones y áreas académicas. A continuación se presenta la situación del Departamento de Informática en donde se describen las funciones principales y algunos elementos del proceso de desarrollo de software, finalmente se plantea en forma escrita la hipótesis y los objetivos del presente trabajo.

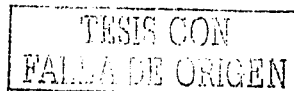
3.1. Antecedentes del Instituto de Investigaciones Bibliográficas

El Instituto de Investigaciones Bibliográficas tiene como antecedente al Instituto Bibliográfico Mexicano creado en 1899 a iniciativa de Francisco del Paso y Troncoso. Este Instituto se formó como parte integrante de la Biblioteca Nacional, con la misión de formar la bibliografía general de México, que incluyera tanto las obras de mexicanos escritas en el país y en el extranjero, como las obras de autores extranjeros que hablen sobre México. Sin embargo, por falta de presupuesto desapareció en 1909. No es sino hasta el 15 de diciembre de 1967, año en que se creó el Instituto de Investigaciones Bibliográficas dependiente de la Coordinación de Humanidades de la UNAM, con la función principal de dirigir y coordinar las labores de la Biblioteca y Hemeroteca Nacionales.

3.2. Estructura Organizacional del Instituto de Investigaciones Bibliográficas

El Instituto está integrado por:

- Un Consejo Interno;
- Un Director General;



- Personal Administrativo
- El Claustro del personal académico

Para el desempeño de sus funciones el Director se auxilia de:

- El Secretario Académico;
- Los Coordinadores de la Biblioteca y Hemeroteca Nacionales de México, y
- El Secretario Administrativo.

En la figura 3.1. se presenta el organigrama del IIB que muestra claramente la estructura organizacional del IIB.

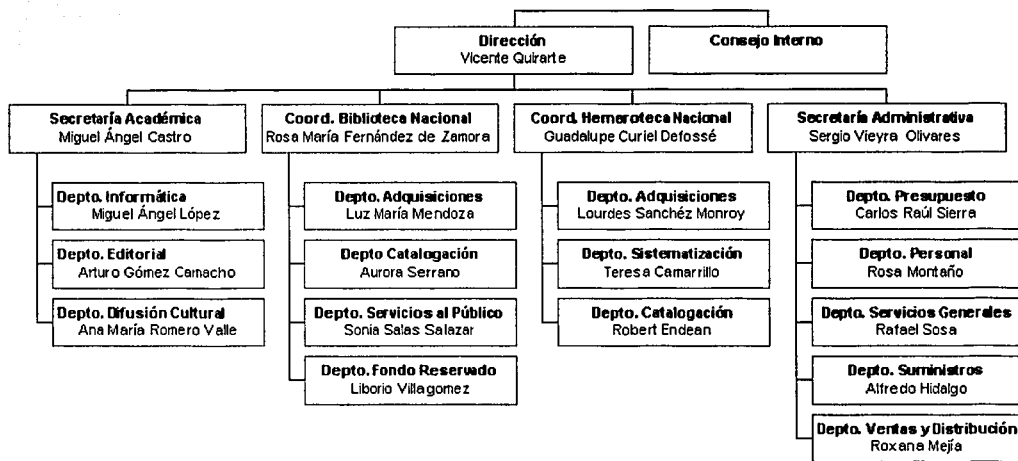
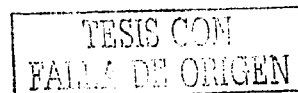


Figura 3.1. Estructura organizacional del Instituto de Investigaciones Bibliográficas.

3.3. Funciones del Instituto de Investigaciones Bibliográficas

- "Realizar e impulsar la investigación bibliográfica, hemerográfica, bibliotecológica, bibliológica, de archivos y manuscritos y de estudio de fuentes, para precisar los campos del conocimiento, facilitar y permitir su estudio;
- Administrar y coordinar el funcionamiento de la Biblioteca y Hemeroteca Nacionales de México; contribuir a la formación de investigadores y técnicos académicos;
- Organizar y promover actividades para la superación académica de sus miembros;
- Organizar y promover actividades relacionadas con sus áreas de investigación para fomentar la difusión de la cultura;
- Desarrollar las áreas de investigación de manera que se contribuya al rescate de la bibliografía y la Hemerografía mexicanas en los acervos de la Biblioteca y Hemeroteca Nacionales de México;
- Preparar, editar y distribuir libros, inventarios, guías, índices, catálogos, folletos, estudios o cualquier instrumento, retrospectivos o contemporáneos, impresos o capturados en cualquier tipo de soporte, que faciliten la investigación o el conocimiento de los acervos de la Biblioteca y Hemeroteca Nacionales de México;
- Desarrollar programas y actividades de carácter nacional e internacional relacionados con las investigaciones del Instituto de Investigaciones Bibliográficas y las colecciones de la Biblioteca y Hemeroteca Nacionales de México;
- Impulsar a la Biblioteca y Hemeroteca Nacionales como órganos normalizadores de las actividades bibliotecológicas y hemerográficas del país, y las demás que le confiera la Legislación Universitaria y el presente reglamento." [IIB, 1999. p.]



3.4. Panorama del Departamento de Informática

Debido a la complejidad del Instituto en cuanto a su estructura organizacional, la labor del Departamento de Informática es a su vez compleja; fundamentalmente por la multiplicidad de usuarios tanto internos como externos, es decir las áreas conformadas por investigadores y las respectivas coordinaciones de la Biblioteca y Hemeroteca Nacional de México las que presentan sus solicitudes de atención al Departamento y éste atiende a toda la Institución. A continuación se puntualizan algunos aspectos importantes del ámbito del Departamento.

3.4.1. Composición del Departamento de Informática

Desde 1994 el Departamento de Informática del IIB está integrado por cinco técnicos académicos y dos administrativos. El número de personas que labora en esta institución, siendo usuarios potenciales asciende aproximadamente a 400 .

3.4.2. Funciones del Departamento de Informática

Los técnicos académicos realizan varias funciones como son: capacitación de recursos humanos, soporte técnico y desarrollo de sistemas entre otras.

El personal administrativo realiza funciones específicas de digitalización de documentos, entendiéndose por captura y/o escaneo de diversos materiales documentales; En ocasiones los técnicos académicos realizan esta actividad interrumpiendo otras actividades.

Otra persona realiza funciones secretariales.

Las funciones del Departamento de Informática fundamentalmente son de apoyo a la investigación, a la Biblioteca y Hemeroteca Nacional de México y pueden ser divididas en las siguientes:

Función de desarrollo de software

Generalmente esta es una función que tiene mayor demanda por parte de la Secretaría Académica, de los investigadores y ocasionalmente de otras áreas como difusión cultural.

Consiste básicamente en desarrollar sistemas bibliográficos, hemerográficos y administrativos.

Las peticiones de este tipo de desarrollos normalmente incluyen sistemas de captura, recuperación, consulta y edición de discos compactos.

Función de apoyo interinstitucional

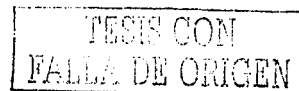
El tipo de actividades realizadas para cumplir con esta función se refieren básicamente al apoyo que el personal del Departamento de Informática periódicamente brinda a dependencias de la propia Universidad Nacional Autónoma de México y a otras Universidades de los Estados de la República, a diversas bibliotecas del D.F. e inclusive a otras organizaciones como el Fondo de Cultura Económico para colaborar en el desarrollo de sistemas, para impartir cursos y para brindar asesorías en determinados paquetes de software.

Formación de recursos humanos

Consiste en la capacitación proporcionada al personal del IIB en el manejo de programas y paquetes de software como: procesadores de texto; sistemas operativos; software para digitalización de documentos, edición de imágenes, manejo de correo electrónico, diseño de presentaciones, diseño de páginas electrónicas y manejadores de bases de datos.

Soporte técnico

Adquirir, instalar, configurar y probar el software y hardware proporcionado al personal del IIB para realizar sus actividades.



Administración de equipos (hardware y software)

Instalar, configurar, mantener y controlar servidores que corren en diferentes plataformas como son: Windows NT, Solaris, Linux y Novell que están destinados a servicios de correo electrónico, bases de datos, página electrónica, intranet e impresión de documentos.

Realizar respaldos en diferentes medios de almacenamiento como: discos duros y ópticos, snaps server, cintas y CD-Rom.

Seguridad

Investigar, instalar, configurar, probar e implementar firewalls y diversos programas de software que garantizan el funcionamiento adecuado de los equipos impidiendo el acceso a intrusos que causan daños a los equipos. La labor principal es la de garantizar el buen desempeño de los servidores para que la producción no se vea interrumpida.

Diseño

Diseñar portadas y contraportadas para la edición de discos compactos; presentaciones animadas para los sistemas; pantallas; logotipos; etc.

Otras

Digitalizar documentos; tomar fotografías; generar diapositivas; elaborar presentaciones, etc.

3.4.3. De la capacitación del personal del departamento

Básicamente el conocimiento que se adquiere es a través de los manuales de manera autodidacta, en pocas ocasiones se proporciona capacitación al personal de informática.

3.5. El proceso de desarrollo del Departamento de Informática

Actualmente el Departamento de Informática del IIB no cuenta con una metodología que guíe el proceso de desarrollo de software, no obstante desarrolla los sistemas que le son solicitados.

A continuación se describen algunos elementos que están directamente relacionados con el desarrollo de sistemas:

Personal

Para cubrir el desarrollo de todos los sistemas solicitados por las diferentes áreas del IIB están asignadas tres personas, mismas que en ocasiones realizan otras actividades.

Metodología

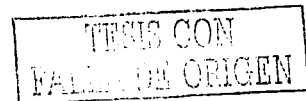
Actualmente el proceso de desarrollo de software no se guía por ninguna metodología ni se utiliza ningún proceso especial o metodológico cada quien realiza el desarrollo de la mejor manera posible.

Biblioteca de componentes

Cada sistema cumple con los requerimientos solicitados por los usuarios. Sin considerar o tomar en cuenta que hay sistemas que por la naturaleza del Instituto son muy parecidos y algunas de las funciones que se requieren en un sistema pudieran haber sido creadas para otro y en todo caso reutilizarse para otro proyecto. Actualmente no se tiene una biblioteca de funciones que permita su consulta y utilización en caso de que fuera conveniente.

Estándares

En el proceso de desarrollo de software no existen plantillas de documentos o estándares que se utilicen al desarrollar los sistemas. Por lo tanto, no hay modelos para realizar análisis, diseño, implementación y pruebas. Esto conlleva a



que cada persona tiene libre albedrío para el desarrollo de los sistemas. Es decir, cada quien realiza la captura de requerimientos, el análisis, el diseño, la implementación y las pruebas como mejor le conviene y en ocasiones no se realizan a detalle. De hecho no se invierte mucho tiempo en el análisis y diseño; en ocasiones se va directamente a la programación lo que ocasiona que los usuarios soliciten frecuentes cambios no solo de funcionalidad sino también de presentación y diseño.

Por estos elementos que se consideran importantes en el proceso de desarrollo de software es que se plantea la posibilidad de adaptar el Proceso Unificado al Departamento de Informática del IIB; ya que de ser posible su adaptabilidad se contaría con una herramienta que guíe este proceso. Por otro lado se tendrían plantillas de documentos que se utilizarían durante el desarrollo de los sistemas.

A continuación se plantea la hipótesis a responder, así como los objetivos que se pretenden alcanzar.

3.6. Hipótesis

Es posible adaptar el Proceso Unificado en los flujos de requerimientos, análisis y diseño en el Departamento de Informática del Instituto de Investigaciones Bibliográficas en el proceso de desarrollo de software.

3.7. Objetivos

- Adaptar el Proceso Unificado en los flujos de trabajo de requerimientos, análisis y diseño en el Departamento de Informática del Instituto de Investigaciones Bibliográficas.
- Crear formatos y estándares que sean útiles en el proceso de desarrollo de software.

- Integrar los flujos de trabajo que conforman el Proceso Unificado.
- Construir una herramienta que guíe el proceso de desarrollo de software del Departamento de Informática.
- Probar la herramienta con un ejemplo real del departamento de Informática.
- Detectar y corregir posibles errores de consistencia, omisión o repetición de roles, artefactos, actividades o pasos.

TESIS CON
FALLA DE ORIGEN

CAPÍTULO 4

4. ADAPTACIÓN DEL FLUJO DE TRABAJO DE REQUERIMIENTOS DEL PROCESO UNIFICADO AL DEPARTAMENTO DE INFORMÁTICA DEL IIB.

A continuación trataremos la definición, clasificación y objetivos de los requerimientos; el flujo de requerimientos del Proceso Unificado; la adaptación del flujo en el Departamento de Informática; la comparación entre los dos flujos identificando claramente, actividades, trabajadores y artefactos y, los pasos que integran cada actividad de los requerimientos en el Departamento de Informática.

El propósito de conocer con exactitud los requerimientos es entender claramente las necesidades de los usuarios y los requerimientos necesarios desde el punto de vista de sistemas, equipos, etc., para satisfacerlas. A menudo, ésta labor resulta compleja porque en varias ocasiones los usuarios no saben expresar claramente lo que necesitan o el informático no tiene los conocimientos para entenderlo. Es aquí en donde los analistas juegan un papel importante para definir y expresar en un lenguaje entendible para los usuarios los requerimientos del sistema.

4.1. Definición, clasificación y objetivos de los requerimientos.

Definición

En términos generales un requerimiento es lo que se necesita. Normalmente es el resultado de una petición realizada por una persona (usuario) para contar con un producto o servicio.

En términos de sistemas, un *requerimiento es un atributo de un sistema que satisface una o varias necesidades o problemas de un "usuario"*.

Clasificación

TESIS CON
FALLA DE ORIGEN

Existen sin duda varios tipos de clasificación de requerimientos, aquí solo se presenta aquella que los divide en funcionales y no funcionales porque consideramos que entre ellos cubren los aspectos que debe reunir un sistema.

Requerimientos funcionales

Son aquellos que tienen que ver directamente con la funcionalidad del sistema. Es decir, se usan para expresar el comportamiento de un sistema especificando tanto entradas como condiciones de salida para alcanzar los resultados esperados.

Requerimientos no funcionales

Los requerimientos no funcionales son tan importantes como los funcionales, estos están directamente relacionados con el usuario final y definen características tales como: uso, fiabilidad, desempeño y portabilidad. Ejemplos de este tipo de requerimientos pueden ser: frecuencia y atención de fallas, tiempos de corrida y respuesta, uso de memoria, código estándar, etc.

Objetivos de los requerimientos

- Establecer y mantener acuerdos entre los usuarios y los desarrolladores.
- Permitir al desarrollador comprender los requerimientos del sistema.
- Definir el límite o alcance del sistema.

Una parte muy importante de este flujo, sin duda, es que los requerimientos del sistema se presentarán a través de modelos de casos de uso; permitiendo entender a los usuarios si sus peticiones son correctas. Es por ello que en este

flujo del proceso unificado se utiliza el lenguaje de los usuarios; para definir el sistema mediante el modelo de casos de uso.

4.2. Flujo de requerimientos del Proceso Unificado.

El flujo inicial del Proceso Unificado comienza en los requerimientos.

A continuación se realizará una revisión cuidadosa de los elementos que integran este flujo para continuar con el proceso de adaptación en el Departamento de Informática.

En las tablas 4.1., 4.2. y 4.3. se especifican: actividades, trabajadores y artefactos del flujo de requerimientos del Proceso Unificado.

Actividades	Descripción
Encontrar actores y casos de uso	<p>La identificación de actores y casos de uso es una actividad muy importante porque permite obtener en forma adecuada los requerimientos del proyecto de software.</p> <p>Esta actividad permite delimitar el entorno del sistema, identificando quiénes y cómo interaccionarán con este y, especificar un glosario de términos que permita entender con detalle su funcionalidad.</p> <p>Esta actividad comprende: encontrar y definir los actores, encontrar los casos de uso, efectuar una descripción breve de los casos de uso y puntualizar el modelo de casos de uso completamente.</p>
Priorizar casos de uso	Determina cuáles casos de uso deben realizarse en la primera iteración y cuáles pueden dejarse para después.
Detallar casos de uso	Consiste en describir a detalle la secuencia de pasos desde el inicio hasta el fin de cada caso de uso. El

TESIS CON FALTA DE ORIGEN	resultado de esta actividad será puntualizar claramente cada caso de uso, ya sea en forma de texto, con diagramas, o con otros elementos que sirvan para tal fin.
Hacer el prototipo de la interfaz de usuario	El objetivo de esta actividad consiste en realizar un prototipo de la interfaz de usuario que permita llevar a cabo los casos de uso de manera eficiente. Este prototipo se hará con base al modelo de casos de uso y a la descripción detallada de los casos de uso.
Estructurar modelo de casos de uso	Esta actividad consiste en refinar el modelo de casos de uso, esto se hace tomando como referencia el detalle de los casos de uso. Aquí ya se deben incluir los casos de uso que están relacionados con el caso de uso general, así como las relaciones que puedan darse entre ellos.

Tabla 4.1. Especificación de las actividades principales del flujo de requerimientos en el Proceso Unificado.

Trabajadores	Descripción
Analista de sistemas	Define y delimita el sistema con base en requerimientos funcionales y no funcionales; identifica actores y casos de uso; y, genera un glosario de términos.
Especificador de casos de uso	Asume la responsabilidad de hacer la descripción detallada de uno o más casos de uso.
Diseñador de interfaz de usuario	El diseñador de interfaz de usuario es responsable de realizar el prototipo de la interfaz de usuario que debe incluir los casos de uso. En ocasiones, es común que por cada actor se realice un prototipo, sin embargo en ocasiones no es necesario incluir todos los casos de uso en el prototipo.
Arquitecto	En este flujo, el arquitecto debe realizar una descripción de la arquitectura del modelo de casos de uso. Una vez que se ha estructurado el modelo de casos de uso, se debe

	realizar una descripción de la arquitectura que se va a elegir.
--	-----------------------------------------------------------------

Tabla 4.2. Especificación de los trabajadores del flujo de requerimientos en el Proceso Unificado.

Artefactos	Descripción
Modelo de casos de uso	Es un modelo del sistema que permite visualizar a los actores, casos de uso y sus relaciones. El objeto de este modelo consiste en que los desarrolladores y usuarios se cercioren de que el sistema cubrirá todos los requerimientos.
Actor	Consiste en identificar todos y cada uno de los actores del sistema, ya sea internos, externos o de cualquier otro tipo. Simplemente se debe de puntualizar de que forma interaccionarán con el sistema.
Caso de uso	Los casos de uso representan fragmentos de funcionalidad del sistema integral. Un caso de uso consiste en un conjunto de pasos para realizar una actividad específica.
Descripción de la arquitectura	Contiene una vista de la arquitectura del modelo de casos de uso, debe contener los casos de uso que representen una actividad crítica de funcionalidad para el sistema.
Glosario	Un glosario o lista de los términos más comunes e importantes del entorno del sistema. La funcionalidad principal radica en que todos los participantes hablen el mismo idioma para evitar confusiones.
Prototipo de la interfaz de usuario	Es una interfaz gráfica que permite comprender más fácilmente la función de los casos de uso. Aquí se puede identificar que es lo que va a realizar cada actor. Durante este flujo, permite identificar todos los requerimientos que por cualquier razón se hubiesen omitido.

Tabla 4.3. Especificación de los artefactos del flujo de requerimientos en el Proceso Unificado.

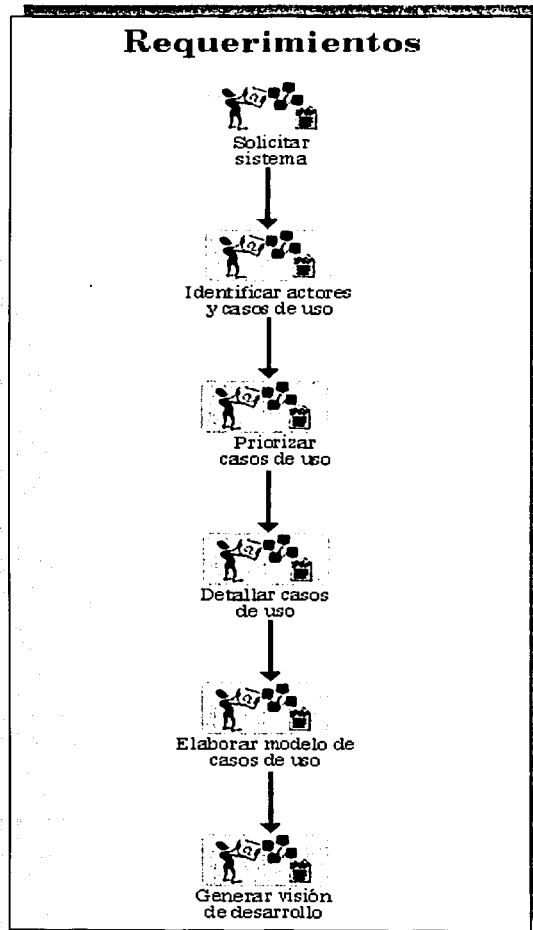
4.3. Adaptación del flujo de requerimientos del Proceso Unificado al Departamento de Informática del IIB.

Para llevar a cabo esta adaptación es importante señalar que se consideraron las actividades que son necesarias para el proceso de desarrollo de software del Departamento de Informática. Estas actividades fueron establecidas con base en la experiencia que se tiene en el desarrollo de sistemas del tipo requerido por el Instituto de Investigaciones Bibliográficas.

En la figura 4.2. se presenta el flujo de requerimientos adaptado al Departamento de Informática.

Es importante señalar que cada una de las actividades que conforman este flujo tienen a su vez una serie de pasos. Los pasos de cada actividad se explican en la sección 4.5. de éste capítulo.

TESIS CON
FALLA DE ORIGEN



TESIS CON
VALOR DE ORIGEN

Figura 4.2. Flujo de requerimientos adaptado al Departamento de Informática.

En las tablas siguientes se especifica la adaptación del flujo de requerimientos del Proceso Unificado en el Departamento de Informática con respecto a actividades (4.4), trabajadores (4.5) y artefactos (4.6).

Es importante señalar que para la realización de las actividades pueden utilizarse diferentes diagramas que proporciona UML. Para la elaboración de estos diagramas se utilizará el estándar de análisis y diseño.

Actividades	Descripción
Solicitar sistema	Manifestación por escrito, por parte del solicitante del desarrollo de un proyecto.
Identificar actores y casos de uso	Permite entender el problema mediante la identificación de actores (usuarios del sistema); y casos de uso. Para realizar esta actividad se utilizan los diagramas de casos de uso.
Priorizar casos de uso	Consiste en establecer cuáles serán los casos de uso que se deben realizar en una primera fase y cuáles se realizarán después. Esta decisión se toma con base a los requerimientos que el usuario establece, considerando la importancia de éstos y demás recursos necesarios.
Detallar casos de uso	Consiste en describir la forma en cómo se realizará una actividad mediante una secuencia de pasos, que va de inicio a fin. Esta descripción puede realizarse con diagramas de actividades o de secuencia.
Elaborar modelo de casos de uso	Consiste en la elaboración de un modelo que permita visualizar los actores, casos de uso y la relación que exista entre ellos.
Generar visión de desarrollo	Se refiere a llenar un formato denominado "visión de desarrollo" que incluye la especificación de los requerimientos del sistema.

Tabla 4.4. Adaptación de actividades del Departamento de Informática con base en el Proceso Unificado.

TESIS CON
FALLA DE ORIGEN

Trabajadores	Descripción
Coordinador	Recibe la solicitud de desarrollo de software y asigna el proyecto al arquitecto de software.
Analista de sistemas	Define y delimita el sistema en base a requerimientos funcionales y no funcionales; identifica actores y casos de uso. Con estos elementos, y la descripción detallada de los casos de uso, estructura el modelo de casos de uso. Por último, genera la visión de desarrollo.
Especificador de casos de uso	Es responsable de priorizar y detallar la funcionalidad de todos los casos de uso. Este personaje trabaja en coordinación con el analista de sistemas, pero debe trabajar directamente con los usuarios reales (actores) del sistema.
Arquitecto de software	Este personaje, es quien toma el proyecto bajo su custodia, contacta al solicitante para entregarle el formato de requerimientos y después recibir el mismo formato.

Tabla 4.5. Adaptación de los trabajadores del Departamento de Informática con base en el Proceso Unificado.

Artefactos	Descripción
Solicitud de desarrollo de software	Artefacto de salida que contiene información general sobre el proyecto de desarrollo de software.
Formato de requerimientos	Artefacto de salida. Es un formato establecido por el grupo de desarrollo, que contiene información específica del proyecto de desarrollo de software.
Diagramas de casos de uso	Artefacto de salida. Son diagramas que muestran el conjunto de casos de uso, actores y sus relaciones.
Formato de prioridad de	Artefacto de salida. Es un formato que permite ubicar

casos de uso	fácilmente el orden de los casos de uso que será el mismo orden en que se deben desarrollar.
Modelo de casos de uso	Artefacto de salida: Es un modelo que permite visualizar el conjunto de casos de uso, actores y sus relaciones.
Visión de desarrollo	Artefacto de salida. Formato creado por el grupo de desarrollo; contiene los requerimientos del sistema y deberá ser llenado por el analista de sistemas. A este artefacto se deberán anexar todos los artefactos anteriores.

Tabla 4.6. Adaptación de los artefactos del Departamento de Informática con base en el Proceso Unificado.

4.4. Comparación del flujo de requerimientos entre el Proceso Unificado y el Departamento de Informática.

En las siguientes tablas se presenta la convergencia y divergencia en los elementos que integran el flujo de requerimientos en el Proceso Unificado con su adaptación al flujo de requerimientos en el Departamento de Informática.

Puntos a considerar:

Indica que el elemento existe.

Indica que el elemento no existe.

Sólo cuando un elemento difiere entre el Proceso Unificado (PU) y el Departamento de Informática (DI) se anota la justificación; en otro caso no se anota nada porque significa que la actividad, el rol o el artefacto no difieren en lo absoluto.

Actividades	PU	DI	Justificación
Solicitar sistema	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Esta actividad es agregada porque debe existir un registro de los proyectos que se desarrollan en el IIB, para fines académicos.
Encontrar actores y casos de uso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Esta actividad solo cambia de nombre en el Departamento de Informática.
Identificar actores y casos de uso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Esta actividad es equivalente a "encontrar actores y casos de uso"; sólo se cambia la palabra "encontrar" por el sinónimo de "identificar", porque resulta más apropiado.
Priorizar casos de uso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Detallar casos de uso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Hacer el prototipo de la interfaz de usuario	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<p>Esta actividad no se lleva a cabo en el Departamento de Informática porque al detallar cada caso de uso, el especificador de casos de uso trabaja conjuntamente con los usuarios reales del sistema, y por lo tanto todos los requerimientos funcionales deben estar contemplados en los flujos siguientes.</p> <p>Por otra parte, en el siguiente flujo, que es el análisis se especifican los requerimientos del sistema en términos de los desarrolladores, y en caso de no existir algún requerimiento, en ese momento se captura.</p> <p>De hecho en el flujo de diseño, se elabora la interfaz de usuario que incluye todos los requerimientos.</p>
Elaborar modelo de casos de uso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Generar visión de desarrollo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<p>Esta actividad es necesaria porque a partir de ella surge la "visión de desarrollo", que es un artefacto de entrada indispensable para el flujo de administración del proyecto para poder elaborar la "visión inicial".</p> <p>Por tanto, esta actividad se agrega en este flujo de requerimientos del Departamento de Informática.</p>
------------------------------	-------------------------------------	-------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.7. Convergencias y divergencias en las actividades que integran el flujo de requerimientos del Proceso Unificado, así como su adaptación al flujo de análisis del departamento de informática.

Trabajadores	PU	DI	Justificación
Analista de sistemas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Especificador de casos de uso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Diseñador de interfaz de usuario	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Este actor no existe en el Departamento de Informática porque su función es diseñar el prototipo de la interfaz de usuario y como no se realiza dicho prototipo, entonces el trabajador no justifica su existencia en este flujo.
Arquitecto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Este trabajador solo cambia de nombre en el Departamento de Informática.
Arquitecto de software	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<i>Arquitecto</i> es igual a <i>arquitecto de software</i> , es decir, realizan las mismas actividades. Sólo que éste último es un concepto actualizado y más representativo para quienes construyen software. Por tanto, así se denominará en los siguientes flujos para unificar los criterios.
Coordinador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	El coordinador es importante en el flujo de requerimientos en el Departamento de Informática porque es el único e indicado para recibir la

			solicitud de desarrollo de software y asignar los proyectos al arquitecto de software.
--	--	--	----------------------------------------------------------------------------------------

Tabla 4.8. Convergencias y divergencias en los trabajadores que integran el flujo de requerimientos del Proceso Unificado, así como su adaptación al flujo de requerimientos del departamento de informática.

Artefactos	PU	DI	Justificación
Modelo de casos de uso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Actor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	En el Departamento de Informática no existe este artefacto; sin embargo los actores pueden identificarse claramente en los diagramas de casos de uso.
Caso de uso	<input checked="" type="checkbox"/>	<input type="checkbox"/>	En el Departamento de Informática no existe este artefacto, sin embargo los casos de uso pueden identificarse claramente en los diagramas de casos de uso.
Descripción de la arquitectura	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Este artefacto se omite en el flujo de requerimientos porque el patrón de arquitectura a seguir se define puntualmente en el estándar de análisis y diseño. Además en el siguiente flujo es cuando realmente se empieza a trabajar con la arquitectura.
Glosario	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Prototipo de la interfaz de usuario	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Este artefacto no se realiza en el Departamento de Informática, porque el diseño de la interfaz de usuario se realiza en el Flujo de Diseño. De esta forma se trabaja ya sobre algo más certero porque para entonces se habrá realizado un análisis minucioso de los requerimientos del sistema.

Solicitud de desarrollo de software	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Esta solicitud es indispensable en este flujo porque es a partir de ella como inicia un proyecto de desarrollo de software.
Formato de requerimientos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Este artefacto es necesario porque ayuda al solicitante a entender lo que necesita. En ocasiones los usuarios no saben transmitir con claridad lo que necesitan.
Diagramas de casos de uso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Los diagramas de casos de uso sustituyen a los actores y casos de uso que aparecen en los artefactos del PU. Esto porque la identificación de actores y casos de uso se realiza mediante los diagramas antes mencionados.
Formato de prioridad de casos de uso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	El artefacto se crea debido a la necesidad de saber y tener un registro del orden en que los casos de uso se desarrollarán.
Visión de desarrollo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Debe existir porque es un artefacto de entrada indispensable para el flujo de administración del proyecto para poder elaborar la "visión inicial" del proyecto.

Tabla 4.9. Convergencias y divergencias en los artefactos que integran el flujo de requerimientos del Proceso Unificado, así como su adaptación al flujo de requerimientos del departamento de informática.

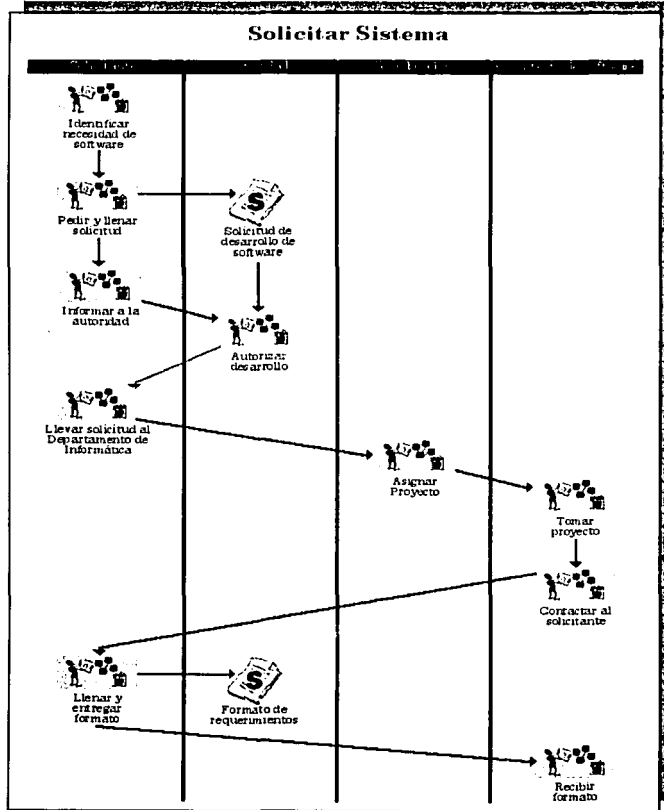
4.5. Detalle de las actividades de Requerimientos en el Departamento de Informática.

A continuación se presenta cada una de las actividades con sus respectivos pasos que conforman el flujo de requerimientos.

En la tabla 4.6. se puede ver la descripción de los artefactos del flujo de requerimientos y en el apéndice E se pueden observar los formatos.

Actividad: Solicitar Sistema.

En esta actividad se puede apreciar que existen dos artefactos de salida, (*Solicitud de desarrollo de software* y *Formato de requerimientos*).



TESIS CON
FALLA DE ORIGEN

Los pasos de esta actividad son:

Identificar necesidad de software. Consiste en que un solicitante o usuario se percata de que necesita un desarrollo de software que cubra ciertos requerimientos.

Pedir y llenar solicitud. Consiste en que el solicitante deberá requerir y llenar la solicitud de desarrollo de software. Esta solicitud estará disponible en la Secretaría Académica, en el Departamento de Informática o en la siguiente dirección electrónica: <http://mofeta.bibliog.unam.mx/pdsiib/flujos/requerimientos/sds.html>¹

Informar a la autoridad. Se refiere a que el solicitante acude con la autoridad correspondiente (Director o Secretario Académico), para hacerle saber de su necesidad.

Autorizar desarrollo. El Director o Secretario Académico se dan por enterados del desarrollo y dan el visto bueno a la solicitud.

Llevar solicitud al Departamento de Informática. El solicitante acude al Departamento de Informática y entrega la solicitud al coordinador de proyectos de software, (Jefe del Departamento de Informática).

Asignar proyecto. El coordinador recibe la solicitud de desarrollo de software, llena la sección que indica ser llenada por el Departamento de Informática y acto seguido asigna el desarrollo al arquitecto de software entregando la solicitud.

Tomar proyecto. Consiste en que el arquitecto de software toma o asume bajo su tutela el proyecto para continuar con el proceso de desarrollo.

¹ La dirección electrónica corresponde a la intranet del Instituto de Investigaciones Bibliográficas.

Contactar al solicitante. Consiste en que el arquitecto de software deberá contactar al solicitante para hacer entrega del formato de requerimientos e indicarle la forma en cómo deberá llenar el formato.

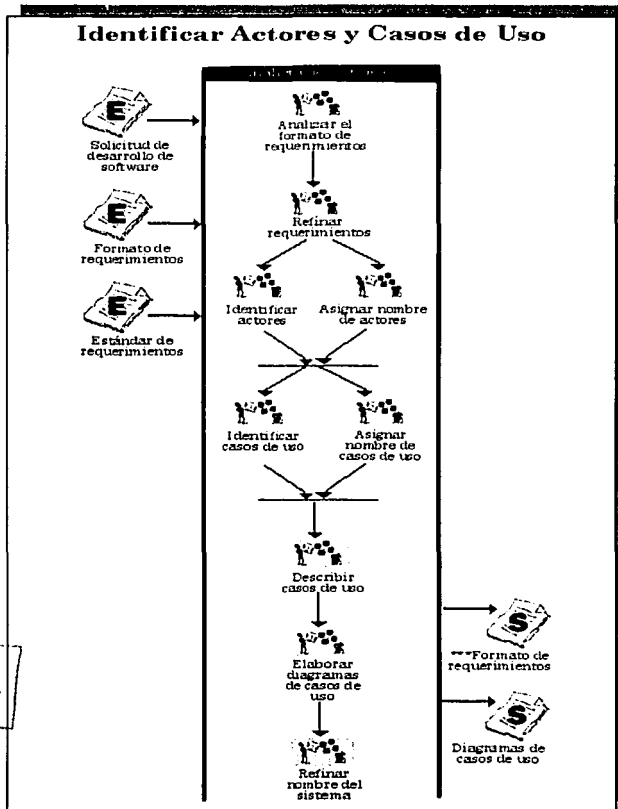
Llenar y entregar formato. El solicitante deberá llenar el formato de requerimientos que contiene información específica sobre el proyecto de desarrollo de software y lo entrega al arquitecto de software.

Recibir formato. El arquitecto de software recibe el formato de requerimientos para continuar con el desarrollo.

TESIS CON
FALLA DE ORIGEN

Actividad: Identificar Actores y Casos de Uso.

En esta actividad se puede apreciar que existe tres artefactos de entrada, (*Solicitud de desarrollo de software, Formato de requerimientos y Estándar de requerimientos*), y dos de salida (****Formato de requerimientos y Diagramas de casos de uso*).²



TESIS CON FALLA DE ORIGEN

² Cuando en un artefacto aparecen 3 asteriscos (***) , significa que el artefacto es de entrada y salida con la diferencia de que al salir ha sido modificado.

Los pasos de esta actividad son:

Conforme se realiza la descripción de los pasos que conforman cada una de las actividades de los flujos de trabajo³, se utilizarán letras cursivas en los nombres de los artefactos con la finalidad de ubicar fácilmente de dónde obtener la información o en dónde registrarla.

Analizar el formato de requerimientos. Consiste en que el analista de sistemas debe leer y analizar detenidamente la solicitud de desarrollo de software y el formato de requerimientos que el solicitante previamente ha llenado para formular y registrar una lista de posibles dudas o preguntas en la sección 11 del *Formato de requerimientos*.⁴

Refinar requerimientos. Consiste en hacer una cita con el responsable del proyecto que se especifica en la sección de datos generales del formato de requerimientos para dar respuesta a las dudas o preguntas formuladas sobre el proyecto de desarrollo. La respuesta a estas interrogantes deberá registrarse en el la sección 11 del *Formato de requerimientos*.

Identificar actores. Se refiere a ubicar quienes van a interactuar directamente con el sistema. Esta información se obtiene de la sección 6 del *Formato de requerimientos*.

Asignar nombre de actores. Aquí se debe asignar un nombre a los actores. Este nombre deberá asignarse con base a la sección 6 del *Formato de requerimientos* y su equivalente con el *Estándar de requerimientos* en la sección 1 que corresponde a los *Nombres de actores*.⁵ Esta información debe ser registrada en la sección 1 correspondiente a la *Especificación general de casos de uso* de los *Diagramas de casos de uso*.⁶

³ El proceso se realizará en los flujos de trabajo de requerimientos, análisis y diseño.

⁴ La sección 11 del formato de requerimientos deberá ser utilizada únicamente por el analista de sistemas.

⁵ Cuando el nombre de un actor no figure en el estándar de requerimientos, entonces deberá ser agregado en la sección correspondiente de dicho estándar.

Identificar casos de uso. Se refiere a ubicar cuales serán las funciones que el sistema debe realizar. Esta información se obtiene de la sección 7 del *Formato de requerimientos*.

Asignar nombre de casos de uso. Aquí se debe asignar un nombre a los casos de uso. Este nombre deberá asignarse con base a la sección 6 del *Formato de requerimientos* y su equivalente con el *Estándar de requerimientos* en la sección 2 que refiere los *Nombres de casos de uso generales*.⁷

La información debe ser registrada en la sección 1 correspondiente a la *Especificación general de casos de uso* de los *Diagramas de casos de uso*.

Describir casos de uso. Consiste en hacer una descripción breve de cada uno de los casos de uso. Esta información debe registrarse en los diagramas de casos de uso.

Elaborar diagramas de casos de uso. Consiste en plasmar en un primer modelo, (modelo de casos de uso), a los actores, casos de uso y sus relaciones. Este dibujo se hace en la sección 1 de la *Especificación general de casos de uso* de los *Diagramas de casos de uso*. Para ello se utilizan los símbolos de actor, caso de uso y relación.

Cabe señalar el modelo sólo es una primera visión que se tiene del proyecto, después, el mismo modelo se irá refinando hasta obtener un modelo de casos de uso que integre todos los requerimientos.

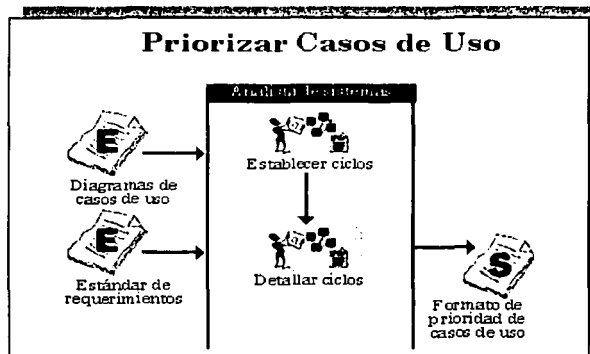
Refinar nombre del sistema. Consiste en valorar si el nombre sugerido en la sección 1 del *Formato de requerimientos* es representativo de la funcionalidad integral del proyecto. Si es así, conservar el nombre, sino modificar el nombre. En cualquier caso, el nombre definitivo y las siglas del sistema deberán registrarse en la sección 12 del *Formato de requerimientos*.

⁶ Se deberá realizar un diagrama por cada actor.

⁷ Cuando el nombre de un caso de uso no figure en el estándar de requerimientos, entonces deberá ser agregado en la sección correspondiente de dicho estándar.

Actividad: Priorizar Casos de Uso.

En esta actividad se puede apreciar que existen dos artefactos de entrada (*Diagramas de casos de uso y el Estándar de requerimientos*) y uno de salida (*Formato de prioridad de casos de uso*).



Los pasos que conforman esta actividad son:

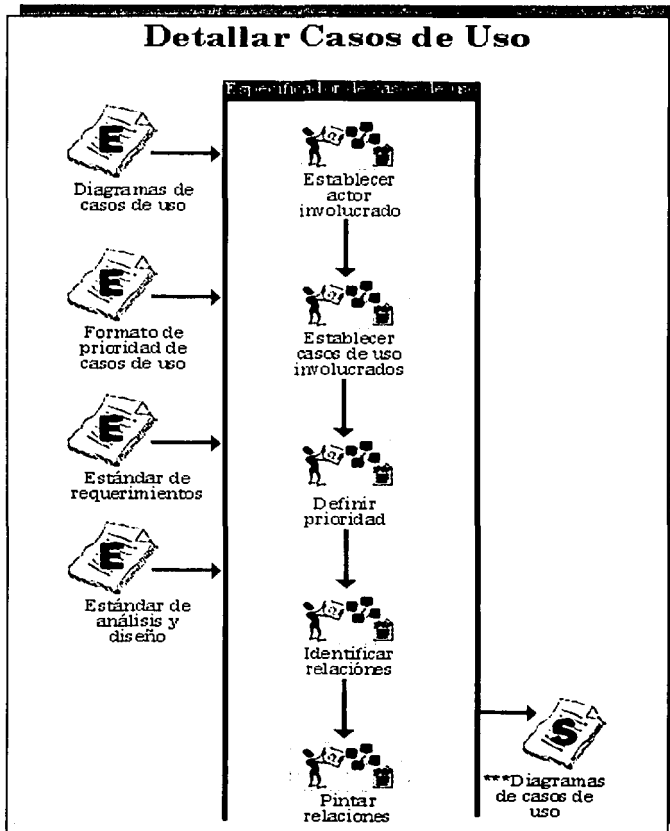
Establecer ciclos. Consiste en establecer en cuántos ciclos se construirá el sistema. Esta definición se debe realizar con base a los diagramas de casos de uso y al *Estándar de requerimientos* en la sección 3 correspondiente a la *Definición de ciclos*. La información que se genera de este paso, deberá registrarse en el *Formato de prioridad de casos de uso*.

Detallar ciclos. Si existe más de un ciclo, entonces se debe elegir los casos de uso más importantes que se van a desarrollar en el primer ciclo, y cuáles se deberán dejar para los siguientes ciclos.

Para detallar los ciclos, se deberá consultar el estándar de requerimientos en la sección 4 correspondiente al *Detalle de ciclos*. Esta información deberá registrarse en el *Formato de prioridad de casos de uso*.

Actividad: Detallar Casos de Uso.

En esta actividad tenemos como artefactos de entrada (*Diagramas de casos de uso, el Formato de prioridad de casos de uso, Estándar de requerimientos y Estándar de análisis y diseño*). Como artefacto de salida tenemos (****Diagramas de casos de uso*).



TESIS CON
FALLA DE ORIGEN

Los pasos que conforman esta actividad son:

Establecer actor involucrado. Se refiere a escribir el nombre del actor involucrado en el caso de uso. La información de este paso deberá registrarse en la sección 2 llamada *Detalle de casos de uso generales* en la parte inferior de los símbolos de actor⁸ de los *Diagramas de casos de uso*.

Establecer casos de uso involucrados. Consiste en anotar el nombre de todos los casos de uso involucrados con el caso de uso general.

La información de este paso se toma del *Formato de prioridad de casos de uso* y se registra en la sección 2 llamada *Detalle de casos de uso generales* en la parte inferior de los símbolos de casos de uso de los *Diagramas de casos de uso*.

Definir la prioridad. Se refiere a establecer el orden en que los casos de uso se van a realizar. El orden se asigna mediante la colocación de números en la esquina superior derecha del símbolo de casos de uso. Los números se toman del *Formato de prioridad de casos de uso*.

Identificar relaciones. Consiste en ubicar, con base a la descripción detallada de los casos de uso, en que forma se relacionan los casos de uso involucrados con el caso de uso general. Aquí solo se identifican las relaciones, no se marcan aún; esto para evitar en la medida de lo posible, relaciones erróneas.

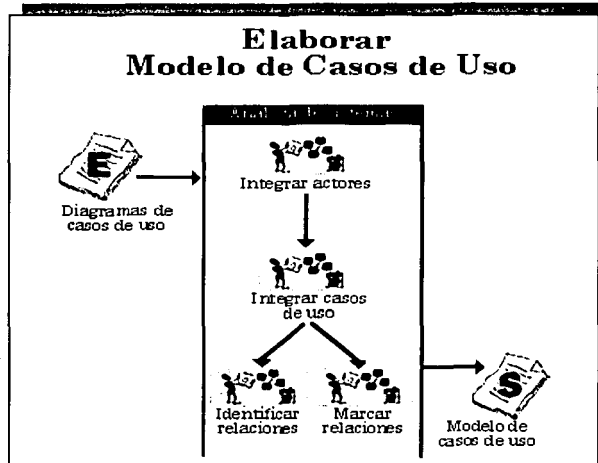
Pintar relaciones. Consiste en marcar la relación entre los casos de uso involucrados con el caso de uso general.

TESIS CON
FALLA DE ORIGEN

⁸ Revisar la sección 10 el estándar de análisis y diseño para identificar los símbolos de actor, caso de uso y relación.

Actividad: Elaborar Modelo de Casos de Uso.

En esta actividad tenemos como artefacto de entrada (*Diagramas de casos de uso*) y como artefacto de salida el (*Modelo de casos de uso*).



Los pasos que conforman esta actividad son los siguientes:

Integrar actores. Consiste en anotar el nombre de todos los actores involucrados en los casos de uso generales⁹. La información de los cuatro pasos que conforman la presente actividad deberá registrarse en el modelo de casos de uso.

Integrar casos de uso. Consiste en anotar el nombre de todos los casos de uso generales.

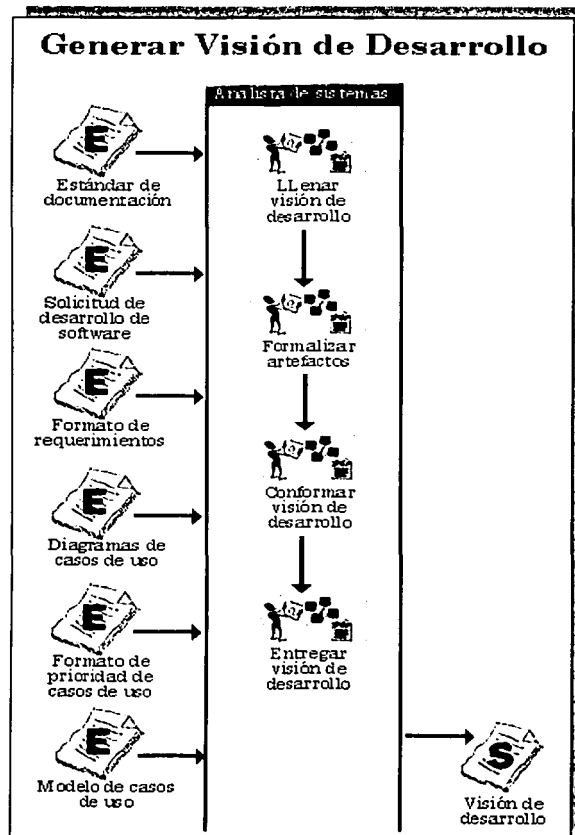
Identificar relaciones. Consiste en ubicar las relaciones que existan entre actores y casos de uso.

Marcar relaciones. Se refiere a pintar las relaciones entre actores y casos de uso.

⁹ Para colocar nombres de actores, casos de uso y para marcar las relaciones se sigue la misma dinámica que en la actividad anterior.

Actividad: Generar Visión de Desarrollo.

En esta actividad se puede apreciar que existen seis artefactos de entrada (*Estándar de documentación, Solicitud de desarrollo de software, Formato de requerimientos, Diagramas de casos de uso, Formato de prioridad de casos de uso y Modelo de casos de uso*) y uno de salida (*Visión de desarrollo*).



ANÁLISIS CON
FUENTE DE ORIGEN

Los pasos de esta actividad son:

Llenar visión de desarrollo. Consiste en que el analista de sistemas deberá llenar la sección 1 de la *Visión de desarrollo* que es un concentrado de los requerimientos que debe cubrir el sistema.

Formalizar artefactos. Se refiere a que el analista de sistemas deberá preparar todos los artefactos del flujo de requerimientos en formato electrónico, esto porque cuando se van realizando las actividades, existen ocasiones que algunos artefactos se llenan a mano.

Conformar visión de desarrollo. El analista de sistemas deberá reunir todos los artefactos del flujo de requerimientos y anexarlos a la visión de desarrollo.

Entregar visión de desarrollo. Una vez conformada la visión de desarrollo, deberá ser entregada al administrador del proyecto.¹⁰

TESIS CON
FALLA DE ORIGEN

¹⁰ La descripción del Administrador del proyecto puede ser consultada en la referencia [Ramírez, 2003].

Para la conclusión del capítulo se puede decir que el flujo de requerimientos tiene como objetivo fundamental, obtener y entender los requerimientos de los usuarios para el desarrollo de productos de software; identificando claramente actividades, trabajadores y artefactos.

Se establecieron y diseñaron los siguientes artefactos: *Solicitud de desarrollo de software, Formato de Requerimientos, Diagramas de casos de uso, Formato de prioridad de casos de uso, Modelo de casos de uso y Visión del desarrollo.*

El artefacto más importante de este flujo es *la visión de desarrollo* porque contiene todos los artefactos del flujo y sirve como *entrada principal* para el *flujo de análisis.*

TESIS CON
FALLA DE ORIGEN

CAPÍTULO 5

5 ADAPTACIÓN DEL FLUJO DE TRABAJO DE ANÁLISIS DEL PROCESO UNIFICADO AL DEPARTAMENTO DE INFORMÁTICA DEL IIB.

En este capítulo trataremos los siguientes puntos: definición y objetivos del análisis; el flujo de análisis del Proceso Unificado; la adaptación del flujo en el Departamento de Informática; y estableceremos la comparación entre los dos flujos identificando claramente, actividades, trabajadores y artefactos, y los pasos que integran cada actividad del análisis en el Departamento de Informática de IIB.

5.1 Concepto y objetivo del análisis.

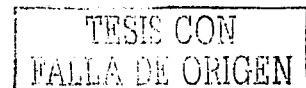
Concepto

El análisis consiste en presentar una especificación más precisa de los requerimientos y requerimientos que debe cubrir el sistema utilizando para ello el lenguaje de los desarrolladores.

Objetivo

El objetivo del análisis consiste en:

- Entender lo que se pretende a través de el análisis de los requerimientos.
- Construir el modelo de análisis que sirve como base para estructurar todo el sistema.



5.2 Flujo de análisis del Proceso Unificado.

Una vez que el flujo de requerimientos ha concluido, se procede a examinar y adaptar el flujo de análisis. En las siguientes tablas se especifican los elementos que integran el flujo de análisis del Proceso Unificado.

Actividades	Descripción
Análisis de la arquitectura	El propósito de esta actividad consiste en esbozar el modelo de análisis y la arquitectura mediante la identificación de paquetes del análisis, clases del análisis, y requerimientos especiales.
Analizar casos de uso	Permite identificar las clases del análisis cuyos objetos son necesarios para poder llevar a cabo el flujo de sucesos del caso de uso. Dicho de otra forma, consiste en refinar los casos de uso, que permitan identificar, si es que existen, los requerimientos especiales. De hecho, un caso de uso pasa a ser una clase del análisis.
Analizar clases	Consiste en Identificar y mantener las responsabilidades de una clase de análisis con base en la realización de los casos de uso. Identifica y mantiene atributos. Y aquí es en donde se capturan los requerimientos especiales.
Analizar paquetes	Esta actividad tiene como objetivo, garantizar que el paquete del análisis sea tan independiente como sea posible; que cumpla su objetivo; y, describa las dependencias entre paquetes, si es que estas existen, finalmente para visualizar el efecto que puedan ocasionar los cambios futuros.

Tabla 5.1. Especificación de las actividades del flujo de análisis en el Proceso Unificado.

Trabajadores	Descripción
Arquitecto de software	Es responsable de la integridad del modelo de análisis, garantizando que éste sea correcto, consistente y legible como un todo.
Ingeniero de casos de uso	Es responsable de la integridad de una o más realizaciones de caso de uso, garantizando que cumplen los requerimientos que recaen sobre ellos.
Ingeniero de componentes	<p>Define y mantiene las responsabilidades, atributos, relaciones y requerimientos especiales de una o varias clases del análisis asegurándose de que cumplen con los requerimientos que se esperan de ella o de acuerdo a la realización de los casos de uso.</p> <p>Este personaje también debe mantener la integridad de los paquetes del análisis, garantizando que su contenido (clases, relaciones y dependencias) sea correcto.</p>

Tabla 5.2. Especificación de los trabajadores del flujo de análisis en el Proceso Unificado.

TESIS CON
FALLA DE ORIGEN

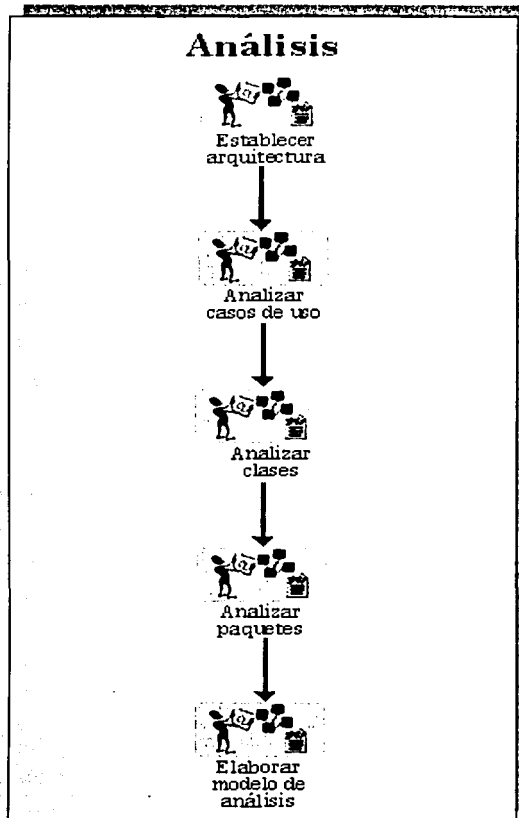
Artefactos	Descripción
Modelo de análisis	El modelo del análisis contiene las clases del análisis, los casos de uso y cualquier artefacto asociado. El modelo del análisis sirve como descripción conceptual del sistema.
Clase del análisis	Representa una abstracción de una o varias clases y/o subsistemas. Esta abstracción posee las siguientes características: Una clase de análisis se centra en el tratamiento de los requerimientos funcionales únicamente. Es por ello que la clase es evidente en el dominio del problema; define atributos y relaciones con otras clases o subsistemas.
Realización de casos de uso del análisis	Describe cómo se lleva a cabo y se ejecuta un caso de uso determinado en términos de las clases del análisis y de los objetos que interactúan con ellas.
Paquetes del análisis	Permite organizar los artefactos de tal forma que puedan ser manejados como piezas independientes dentro del modelo del análisis. Es por ello que un paquete del análisis puede constar de clases, casos de uso, entre otros.
Descripción de la arquitectura (vista del modelo de análisis)	Contiene una vista de la arquitectura del modelo de análisis que muestra los artefactos significativos para la arquitectura.

Tabla 5.3. Especificación de los artefactos del flujo de análisis en el Proceso Unificado.

TESIS CON
FALLA DE ORIGEN

5.3. Adaptación del flujo de análisis del Proceso Unificado al Departamento de Informática del IIB.

En la figura 5.2. se presenta el flujo de análisis del Departamento de Informática que fue adaptado con base al Proceso Unificado.



TESIS CON
FALLA DE ORIGEN

Figura 5.2. Flujo de análisis adaptado al Departamento de Informática.

En las tablas 5.4., 5.5. y 5.6. se especifican: actividades, trabajadores y artefactos del flujo de análisis del Departamento de Informática.

Actividades	Descripción
Establecer arquitectura	En esta actividad se debe establecer el patrón de arquitectura que se debe seguir con base en el modelo de casos de uso; se define y describe el patrón de arquitectura y se identifican los paquetes del análisis.
Analizar casos de uso	El objetivo de esta actividad es el identificar las clases del análisis y sus responsabilidades, así como llevar a cabo la captura de requerimientos especiales sobre la realización de los casos de uso del diseño. Esta actividad se puede decir, que es una refinación de los casos de uso para obtener las clases del análisis.
Analizar clases	Esta actividad consiste en describir y mantener las responsabilidades de la clase; identificar atributos y relaciones, y, capturar los requerimientos especiales sobre la realización de la clase del análisis. Al conjunto de clases del análisis, relacionadas con una misma funcionalidad, se les denomina paquetes de servicio. Porque todas ellas contribuyen a un mismo servicio.
Analizar paquetes	La finalidad de esta actividad es la de estructurar el funcionamiento del sistema en paquetes independientes y manejables. Con esto se pretende que al realizarse un cambio, éste se pueda identificar e implementarse adecuadamente. Aquí se debe definir el contenido y las dependencias de los paquetes del análisis.
Elaborar modelo de análisis	Consiste en realizar un modelo que permita visualizar la estructura del sistema incluyendo los respectivos paquetes sobre las clases del análisis.

Tabla 5.4. Adaptación de actividades del Departamento de Informática con base en el Proceso Unificado.

Trabajadores	Descripción
Arquitecto de software	<p>Establecer la arquitectura del sistema que será rectora durante el proceso de desarrollo del software.</p> <p>Construye el modelo de análisis el cual debe ser claro, consistente e integro.</p>
Ingeniero de casos de uso	Este personaje es responsable de la realización de los casos de uso.
Ingeniero de componentes	<p>Un ingeniero de componentes es el indicado para definir y mantener responsabilidades, atributos, relaciones y captura de requerimientos especiales de las clases del análisis.</p> <p>Es responsable de definir y mantener la integridad de los paquetes del análisis; garantizando que sus contenidos sean los adecuados y la dependencia entre paquetes sea correcta y mínima.</p>

Tabla 5.5. Adaptación de trabajadores del Departamento de Informática con base en el Proceso Unificado.

Artefactos	Descripción
Descripción de la arquitectura	Es un "documento" que contiene la descripción a detalle del patrón de arquitectura que se debe seguir durante el proceso de desarrollo de software.
Clases del análisis	Es un "modelo" que incluye las diferentes clases del análisis como son: clases de interfaz, de dominio del problema y del manejo de datos.
Realización de casos de uso del análisis	Son "diagramas" que describen el flujo completo de la forma en como se realizan los casos de uso, pero, mediante la utilización de clases del análisis.
Paquetes del análisis	Es un "modelo" que permite visualizar el contenido de los paquetes del análisis como son: clases, realización de casos de uso u otros paquetes.

Modelo de análisis	Es un "modelo" que se realiza con base al patrón de arquitectura establecido e paquetes y clases.
--------------------	---------------------------------------------------------------------------------------------------

Tabla 5.6. Adaptación de artefactos del Departamento de Informática con base en el Proceso Unificado.

5.4. Comparación del flujo de análisis entre el Proceso Unificado y el Departamento de Informática del IIB.

En las siguientes tablas se presenta la convergencia y divergencia en los elementos que integran el flujo de análisis en el Proceso Unificado; así como su adaptación al flujo de análisis en el Departamento de Informática del IIB.

Actividades	PU	DI	Justificación
Establecer arquitectura	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Con esta actividad inicia el flujo de Análisis en el Departamento de Informática. Esto, porque se consideró oportuno que, hasta el momento en que los usuarios manifiesten su conformidad con los requerimientos que solicitaron, se debe continuar, eligiendo el patrón de arquitectura que se seguirá y conservará durante el proceso de desarrollo de software; realizando las modificaciones pertinentes en cada flujo. Con esto se trabaja sobre una base ya establecida.
Análisis de la arquitectura	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	La razón por la que esta actividad no figura en el Departamento de Informática, es porque en la actividad anterior, los pasos que se realizan, después de elegir y describir el patrón de

			arquitectura, son los mismos que incluye el análisis de la arquitectura. Otra razón más, es porque es más adecuado tener todo lo que se refiere a la arquitectura, en una sola actividad. Que incluya, elección, descripción y análisis.
Analizar casos de uso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Analizar clases	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Analizar paquetes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Elaborar modelo de análisis	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<p>No existe como actividad independiente en el Proceso Unificado; sin embargo, el modelo de análisis es uno de los artefactos principales en ambos casos. Es por ello que se agrega como una actividad particular en el Departamento de Informática.</p> <p>Se debe generar una vez que las actividades anteriores han sido realizadas; la finalidad es que el modelo esté lo más completo posible porque es una entrada principal para el flujo de diseño.</p>

Tabla 5.7. Convergencias y divergencias en las actividades que integran el flujo de análisis del Proceso Unificado, así como su adaptación al flujo de análisis del Departamento de Informática.

TESIS CON
FALLA DE ORIGEN

Trabajadores	PU	DI	Justificación
Arquitecto de software	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Ingeniero de casos de uso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Ingeniero de componentes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Tabla 5.8. Convergencias y divergencias en los trabajadores que integran el flujo de análisis del Proceso Unificado, así como su adaptación al flujo de análisis del Departamento de Informática.

Artefactos	PU	DI	Justificación
Modelo de análisis	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Clases del análisis	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Realización de casos de uso del análisis	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Paquetes del análisis	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Descripción de la arquitectura (vista del modelo de análisis)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Este artefacto, en el PU, es un modelo jerárquico que representa el modelo de análisis. En el Departamento de Informática se denomina únicamente descripción de la arquitectura.
Descripción de la arquitectura	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Únicamente cambia de nombre.

Tabla 5.9. Convergencias y divergencias de los artefactos que integran el flujo de análisis del Proceso Unificado, así como su adaptación al flujo de análisis del Departamento de Informática.

TESIS CON
FALLA DE ORIGEN

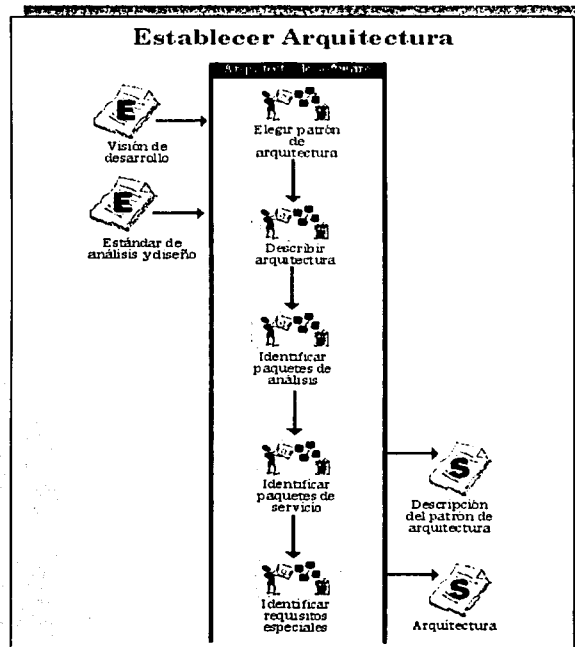
5.5 Detalle de las actividades del análisis en el Departamento de Informática del IIB.

A continuación se presenta cada una de las actividades con sus respectivos pasos que conforman el flujo de análisis.

Actividad: Establecer Arquitectura.

En esta actividad se puede apreciar que existen dos artefactos de entrada (*Visión de desarrollo* y *Estándar de análisis y diseño*), dos de salida (*Descripción del patrón de arquitectura* y *Arquitectura*).

En la tabla 5.6. se puede ver la descripción de todos los artefactos del flujo de análisis y en el apéndice F se pueden observar los formatos.



TESIS CON
FALLA DE ORIGEN

Los pasos de esta actividad son:

Elegir y describir patrón de arquitectura. Para el proceso de desarrollo de software del IIB se ha establecido como patrón de arquitectura el de capas, el cual puede ser consultado en la sección 3 del *Estándar de análisis y diseño*. La elección del patrón es debido a que los sistemas bibliohemerográficos y administrativos que se desarrollan en el IIB tienden a cumplir con la especificación del patrón de arquitectura por capas.

Cuando surge un nuevo proyecto que no se ajuste al patrón de arquitectura por capas, se deberá elaborar una descripción del patrón que se utilizará y la descripción se deberá agregar al estándar de análisis y diseño en la sección 3 sobre la *Descripción de la arquitectura*.

Identificar paquetes de análisis. Con base al modelo de casos de uso se deben identificar los paquetes del análisis y sus contenidos. Esta identificación se realiza en base a los requerimientos funcionales del sistema. Para integrar los paquetes del análisis se debe tomar en cuenta la relación que existe entre los actores, casos de uso y sus relaciones. Cada caso de uso general se convierte en un paquete del análisis. Todos los paquetes del análisis se deberán registrar en la arquitectura en la sección 1 denominada *Paquetes del análisis*.

Identificar paquetes de servicio. “La identificación adecuada de los paquetes de servicio se suele hacer cuando el trabajo de análisis está avanzado, momento en el que los requisitos funcionales se comprenden bien y existe la mayoría de las clases del análisis. Todas las clases del análisis dentro del mismo paquete de servicio contribuyen al mismo servicio.” [Jacobson, 2000, p.190].

“Debido a que los paquetes de servicio contienen clases relacionadas funcionalmente, obtendremos con ellos una estructura de paquetes que aísla la mayoría de los cambios en paquetes individuales.” [Jacobson, 2000, p.191].

Ejemplos de cuando una clase "X" y una clase "Y" están relacionadas funcionalmente.

- Un cambio en X muy probablemente requerirá un cambio en Y.
- Los objetos de X interactúan con los objetos de Y, probablemente a través de varios mensajes diferentes.

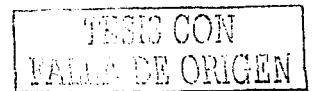
La identificación de los paquetes de servicio se realiza en la sección 2 de la arquitectura designada "Paquetes de servicio".¹¹

Identificar requerimientos especiales. Se refiere a ubicar las limitaciones o restricciones que pudieran ocurrir en cuanto a: persistencia, tolerancia a fallos, seguridad, entre otras. Es importante ubicarlos y tenerlos presentes al momento de realizar el diseño y la implementación.

Ejemplos:

- Rangos de tamaño
- Volumen
- Período de persistencia
- Frecuencia de actualización
- Fiabilidad

La identificación de requerimientos especiales se realiza en la sección 3 de la arquitectura llamada *Captura de requerimientos espaciales*.

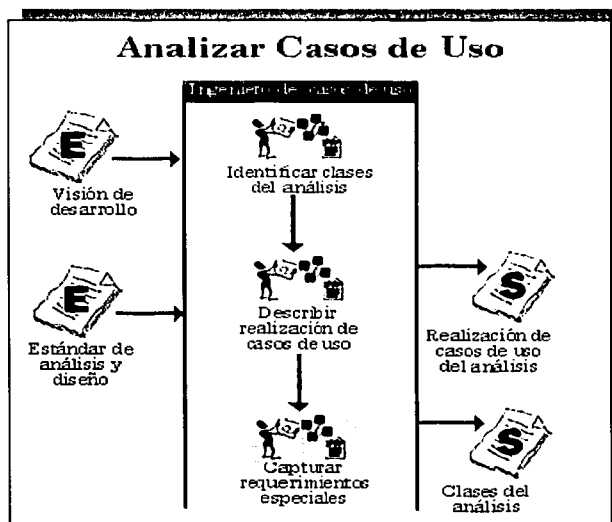


¹¹ Los paquetes de servicio pueden considerarse como paquetes del análisis debido a que contienen varias clases que están relacionadas para proporcionar una funcionalidad del sistema; en este caso un servicio.

Actividad: Analizar Casos de Uso.

En esta actividad se observan dos artefactos de entrada; (*Visión de desarrollo* y *Estándar de análisis y diseño*), y dos de salida, (*Realización de casos de uso del análisis* y *Clases del análisis*).

De ésta actividad en adelante se deberán realizar los pasos correspondientes sobre el ciclo que se está trabajando en ese momento.



Los pasos que conforman esta actividad son los siguientes:

Identificar clases del análisis. En este paso se deben identificar todas las clases que sean necesarias para realizar los casos de uso. Como en el proceso de desarrollo de software del IIB se ha establecido el patrón de arquitectura por

capas, entonces se deberán identificar las clases de dominio del problema, de interfaz humana y del manejo de datos. También se hace un primer esbozo de las clases sobre nombres, responsabilidades, atributos y relaciones.

En caso de que el patrón cambie, se deberá realizar la identificación sobre el patrón de arquitectura establecido.

Para la identificación de las clases del análisis se requieren los *Diagramas de casos de uso* y para registrar la información debe hacerse sobre en la sección 1 de las *Clases del análisis*, para pintar el símbolo de las clases se debe consultar el estándar de análisis y diseño en la sección 10 nombrada *Símbolos permitidos en la elaboración de artefactos*.

Describir realización de casos de uso. Consiste en realizar un refinamiento los casos de uso en donde se deberá especificar paso a paso lo que deberá hacer tanto el usuario como el sistema. La descripción debe hacerse sobre la *Realización de casos de uso del análisis* en la sección 1 denominada *Descripción de casos de uso*. Para cada caso de uso se deberá hacer su descripción.

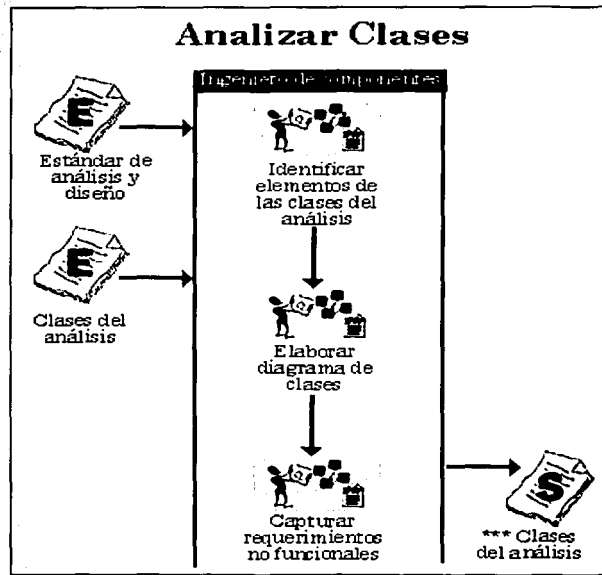
Capturar requerimientos especiales. Consiste en recoger todos los requerimientos necesarios para la realización de los casos de uso. Por ejemplo: orden de captura, color y tamaño, etc. La descripción de los requerimientos especiales se hace sobre la *Realización de casos de uso del análisis* en la sección 2 designada *Captura de requerimientos especiales*.

TESIS CON
FALLA DE ORIGEN

ESTA TESIS NO SALE
DE LA BIBLIOTECA

Actividad: Analizar Clases.

En esta actividad se puede apreciar que existen cuatro artefactos de entrada, (*Estándar de análisis y diseño y Clases del análisis*) y uno de salida, (****Clases del análisis*).



Los pasos que conforman esta actividad son los siguientes:

Identificar elementos de las clases del análisis. Consiste en identificar y nombrar los atributos y métodos de todas las clases del análisis.

Para nombrar los elementos se necesita consultar la sección 6 y 8 del *Estándar de análisis y diseño*. La información se deberá registrar en la sección 1 de las *Clases del análisis*.

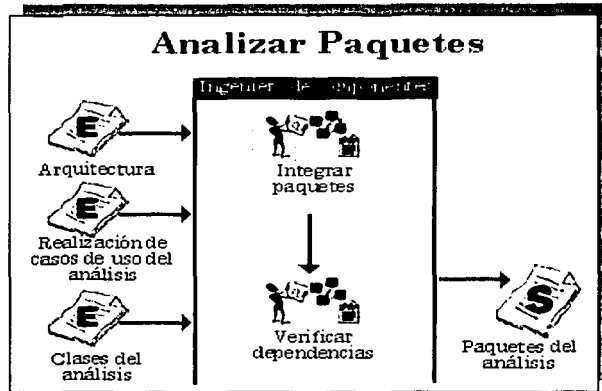
Elaborar modelo de clases. Consiste en tomar solo las clases principales, a partir de todas las clases del análisis. Después se deben establecer las relaciones de asociación, agregación y generalización pintando las líneas correspondientes de cada tipo de relación. Para generar el modelo se deben consultar las *Clases del análisis y el Estándar de análisis y diseño*. Como resultado se obtendrá el modelo de clases que deberá registrarse en la sección 2 de las *Clases del análisis* llamada *Diagrama de clases*.

Capturar requerimientos especiales. Consiste en recoger todos los requerimientos necesarios de las clases del análisis que previamente no se hayan considerado. La captura de requerimientos especiales deberá registrarse en la sección 3 de las *Clases del análisis* denominada *Captura de requerimientos especiales*.

TESIS CON
FALLA DE ORIGEN

Actividad: Analizar Paquetes.

En esta actividad se observan tres artefactos de entrada; (*Arquitectura*, *Realización de casos de uso del análisis* y *Clases del análisis*), y uno de salida, (*Paquetes del análisis*).



Los pasos que conforman esta actividad son los siguientes:

Integrar paquetes. Consiste en establecer cuáles son las clases que conformarán los paquetes identificados en la arquitectura. Para ello se verifican todas las clases que intervienen en la realización de casos de uso del análisis, y se van colocando dentro del símbolo de paquetes. Es importante que al establecer el contenido de los paquetes se evite, en la medida de lo posible, la dependencia entre ellos. Los paquetes se registran en el artefacto *Paquetes del análisis*.

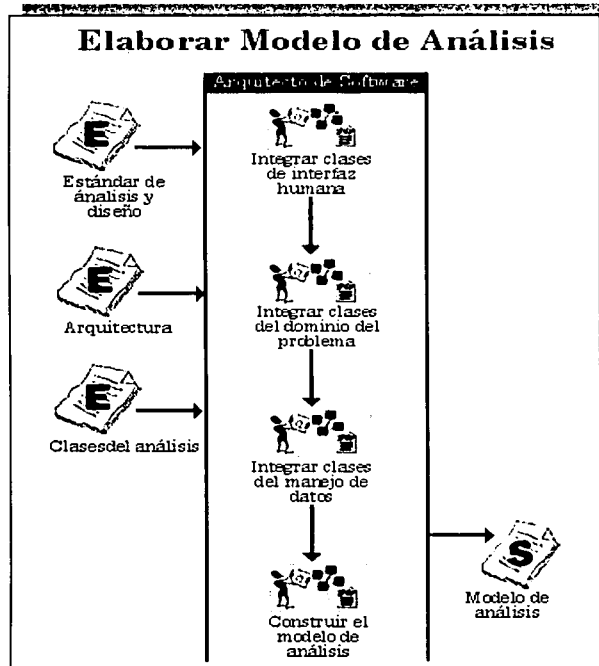
Verificar dependencias. Consiste en verificar que los paquetes solo contienen clases relacionadas funcionalmente. La pretensión es lograr que los paquetes sean tan independientes como sea posible.

En caso de que exista demasiada dependencia entre paquetes se debe considerar la reubicación de las clases contenidas en los paquetes que son demasiado dependientes de otros paquetes. Esto se hace sobre los *Paquetes del análisis*.

Actividad: Elaborar Modelo de Análisis.

Para realizar esta actividad se necesitan los siguientes artefactos de entrada:
(*Estándar de análisis y diseño, Arquitectura y Clases del análisis*).

El resultado de esta actividad es el *Modelo de análisis*.



Los pasos de esta actividad son:

Integrar clases de interfaz humana. Consiste en integrar todas las clases de interfaz humana en un solo paquete que se colocará en la primera capa. La integración se realiza con base al patrón de arquitectura elegido; para ello se utiliza la sección 1 del *Modelo de análisis* denominada *Clases de interfaz humana*.

TESIS CON
FALLA DE ORIGEN

Integrar clases del dominio del problema. Consiste en integrar todas las clases del dominio del problema en un solo paquete que se colocará en la capa intermedia. Esta integración se realiza con base al patrón de arquitectura elegido; para ello se utiliza la sección 2 del *Modelo de análisis* designada *Clases del dominio del problema*.

Integrar clases del manejo de datos. Consiste en integrar todas las clases del manejo de datos en un solo paquete que se colocará en la tercera. Esta integración se realiza con base al patrón de arquitectura elegido; para ello se utiliza la sección 3 del *Modelo de análisis* llamada *Clases del manejo de datos*.

Construir el modelo de análisis. Para construir el modelo de análisis se coloca a los paquetes en cada una de las capas de la arquitectura; se pinta con una línea punteada la división de cada capa y se establece la relación entre las capas. Para ello se debe consultar el *Estándar de análisis y diseño* para ver que símbolos se utilizan. El resultado será el *Modelo de análisis* ubicado en la última sección.

Para la conclusión del capítulo se puede decir que el flujo de análisis tiene como objetivo fundamental, refinar y expresar los requerimientos de los usuarios en términos de los desarrolladores.

Se establecieron y diseñaron los siguientes artefactos: *Arquitectura*, *Realización de casos de uso del análisis*, *Clases del análisis*, *Paquetes del análisis* y el *Modelo de análisis*.

El artefacto más importante de este flujo es el *Modelo de análisis* porque es la base para estructurar todo el sistema, por lo tanto, sirve como *entrada principal* para continuar con el flujo de diseño.

CAPÍTULO 6

6 ADAPTACIÓN DEL FLUJO DE TRABAJO DE DISEÑO DEL PROCESO UNIFICADO AL DEPARTAMENTO DE INFORMÁTICA DEL IIB.

En este capítulo trataremos los siguientes puntos: definición y objetivos del diseño; el flujo de diseño del Proceso Unificado; la adaptación del flujo en el Departamento de Informática; la comparación entre los dos flujos identificando claramente roles, actividades y artefactos; y los pasos que integran cada actividad del diseño en el Departamento de Informática.

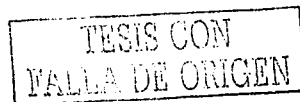
6.1 Concepto y objetivo del diseño.

Concepto

El diseño es un proceso que permite establecer la forma en que se va a implementar el sistema. Aquí debe tomarse en cuenta las decisiones, estrategias y tácticas útiles para que el sistema cubra absolutamente todos los requerimientos ya detectados de los usuarios.

Objetivo

El objetivo del diseño es crear un modelo que sirva como base para la implementación incluyendo sus restricciones con respecto al lenguaje de programación, componentes reutilizables, sistemas operativos, entre otros.



6.2 Flujo de diseño del Proceso Unificado.

El Proceso Unificado establece como tercer flujo, al diseño. A continuación se muestran los elementos que lo conforman.

En las tablas 6.1., 6.2. y 6.3. se especifican: actividades, trabajadores y artefactos del flujo de diseño del Proceso Unificado.

Actividades	Descripción
Diseñar arquitectura	Esboza los modelos de diseño y despliegue y su arquitectura mediante la identificación de clases del diseño, subsistemas, interfaces y nodos de red.
Diseñar casos de uso	Su objetivo principal es identificar las clases del diseño y subsistemas cuyas instancias son necesarias para llevar a cabo el flujo de sucesos del caso de uso.
Diseñar clases	Consiste en crear una clase que cumpla su papel en la realización de casos de uso del diseño y los requerimientos no funcionales que se aplican a estos. Esto también incluye el mantenimiento del diseño de las clases en aspectos de: operaciones, atributos, relaciones, métodos, estados, dependencias, requerimientos y una correcta realización de la interfaz requerida.
Diseñar subsistemas	Su objetivo es garantizar que el subsistema sea independiente de otros subsistemas; que las interfaces sean correctas y cumpla su propósito; es decir que realice en forma correcta las operaciones.

Tabla 6.1. Especificación de las actividades del flujo de diseño en el Proceso Unificado.

TESIS CON
FALLA DE ORIGEN

Trabajadores	Descripción
Arquitecto de software	Es responsable de la integridad del modelo de diseño garantizando que sea correcto, consistente y legible en su totalidad.
Ingeniero de casos de uso	Es responsable de la integridad de una o más realizaciones de casos de uso del diseño, y debe garantizar que cumplen los requerimientos que se esperan de ellos.
Ingeniero de componentes	Define y mantiene las operaciones, métodos, atributos, relaciones y requerimientos de implementación de una o más clases del diseño, garantizando que cada clase cumple con los requerimientos que se esperan de ella.

Tabla 6.2. Especificación de los trabajadores del flujo de diseño en el Proceso Unificado.

Artefactos	Descripción
Modelo de diseño	"El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requerimientos funcionales junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el modelo de diseño sirve como una entrada fundamental para las actividades de implementación." [Jacobson, 2000, p.208].
Clases del diseño	Es una abstracción de cómo la clase será representada en el lenguaje de programación. Por lo tanto; es por ello que sus parámetros, atributos, operaciones y tipo, deberán especificarse utilizando la sintaxis del lenguaje de programación.
Realización de casos de uso de diseño	Es un "diagrama" que describe la forma en cómo se realiza un caso de uso específico. Todos los casos de uso deben

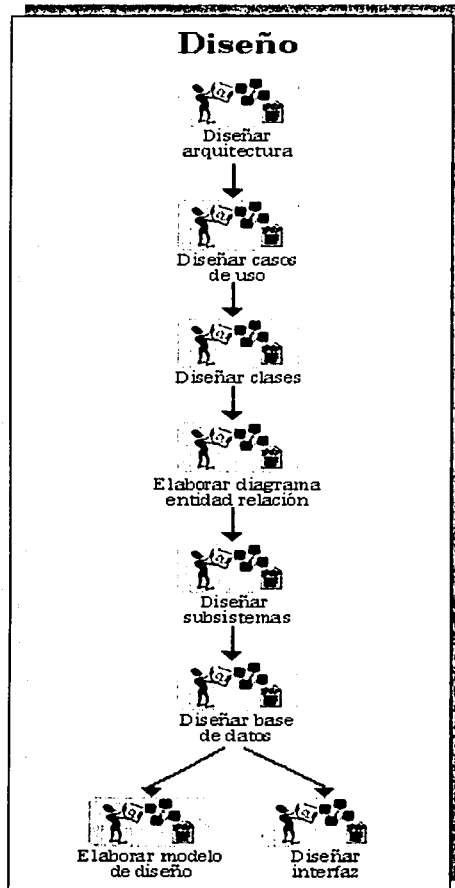
	describir el flujo de sucesos para que se puedan llevar a cabo.
Subsistemas de diseño	Es un "modelo" que permite organizar los elementos del diseño, en piezas manejables y en forma casi independiente. Un subsistema puede contener clases de diseño, realizaciones de casos de uso, interfaces, etc.
Interfaz	Es un "diseño", a través de pantallas, que permite visualizar las operaciones o métodos que deben realizar las clases y los subsistemas.
Descripción de la arquitectura (vista del modelo de diseño)	Es un "modelo" que contiene la una vista arquitectónica del modelo de diseño; muestra los artefactos relevantes para la arquitectura, como son: subsistemas, interfaces y las dependencias entre ellos; las clases del diseño y la realización de casos de uso del diseño.
Modelo de despliegue	Es un "modelo de objetos" que permite comprender la distribución física del sistema, esto, en términos de funcionalidad entre los nodos de red.
Descripción de la arquitectura (vista del modelo del modelo de despliegue)	"Documento" que contiene una vista de la arquitectura del modelo de despliegue que incluye los elementos más importantes.

Tabla 6.3. Especificación de los artefactos del flujo de diseño en el Proceso Unificado.

TESIS CON
FALLA DE ORIGEN

6.3. Adaptación del flujo de diseño del Proceso Unificado al Departamento de Informática.

En la figura 6.2. se presenta el flujo de diseño del Departamento de Informática que fue adaptado con base al Proceso Unificado.



TESIS CON
FALLA DE ORIGEN

Figura 6.2. Flujo de diseño adaptado al Departamento de Informática.

En las tablas 6.4., 6.5. y 6.6. se especifican: actividades, trabajadores y artefactos del flujo de diseño del Departamento de Informática.

Actividades	Descripción
Diseñar arquitectura	Esta actividad consiste en realizar un modelo que permita identificar subsistemas e interfaces; clases del diseño; nodos y configuraciones de red; requerimientos especiales y algunos elementos de otros sistemas que puedan ser reusables.
Diseñar casos de uso	En esta actividad se deben identificar las clases y subsistemas que se necesitan para llevar a cabo el caso de uso; la relación que exista entre ellos; especificar las operaciones que se van a realizar y establecer el flujo que se debe seguir. Para realizar esta actividad se utilizarán los diagramas de secuencia.
Diseñar clases	En esta actividad se deberán crear las clases de diseño con base en la realización de los casos de uso del diseño y los requerimientos no funcionales. Se deben mantener y/o refinar los elementos de las clases, como son: atributos, relaciones, métodos y dependencias.
Elaborar diagrama entidad relación	La finalidad de elaborar el diagrama es para facilitar la implementación de la base de datos independientemente del manejador de bases de datos del que se trate.
Diseñar subsistemas	El objeto de esta actividad consiste en garantizar que los subsistemas son tan independientes como sea posible; verificar que las interfaces que integran el subsistema sean correctas y verificar que cumplan su función.
Diseñar base de datos	Aquí se deben identificar y definir tablas; identificar y definir campos; modelar y definir el esquema completo de la base de datos para garantizar la integridad referencial.

Elaborar modelo de diseño	El propósito de esta actividad consiste en integrar los elementos del diseño en un solo modelo para apreciar la estructura completa del sistema. Los elementos a integrar deben ser: subsistemas y clases.
Diseñar interfaz	Consiste en realizar un diseño de la interfaz de usuario con base en las realizaciones de caso de uso y las clases de diseño.

Tabla 6.4. Adaptación de actividades del flujo de diseño del Departamento de Informática con base en el Proceso Unificado.

Trabajadores	Descripción
Arquitecto de software	Realiza el diseño de la arquitectura y construye el modelo de diseño que debe ser claro, consistente e integro.
Ingeniero de casos de uso	Este personaje es responsable de la realización de los casos de uso del diseño. También deberá realizar la interfaz de usuario.
Ingeniero de componentes	El ingeniero de componentes es el indicado para: Definir y mantener responsabilidades, atributos, relaciones y captura de requerimientos especiales de las clases del diseño y verificar las dependencias e integridad de los subsistemas.
Diseñador de bases de datos	Es el encargado de elaborar el diagrama entidad relación y el diseño la base de datos.

Tabla 6.5. Adaptación de trabajadores del flujo de diseño del Departamento de Informática con base en el Proceso Unificado.

TESIS CON
FALLA DE ORIGEN

Artefactos	Descripción
Diseño de la arquitectura	Es un esbozo del modelo de diseño que incluye subsistemas, clases, casos de uso, nodos y configuraciones de red; así como elementos que puedan ser reutilizados.
Realización de casos de uso de diseño	Son "diagramas" que describen el flujo completo de la forma en como se realizan los casos de uso, pero, mediante la utilización de clases del diseño.
Clases de diseño	Es un "modelo" que incluye las diferentes clases del diseño como son: clases de interfaz, de dominio del problema y del manejo de datos. Es importante destacar que este diseño debe realizarse con base al lenguaje de programación que ha sido establecido. Por lo que los atributos, métodos, etc. Deberán expresarse con la sintaxis adecuada.
Diagrama entidad relación	Diagrama que contiene todas las entidades y relaciones que deben ser consideradas al momento de diseñar la base de datos.
Subsistemas de diseño	Es un "modelo" que organiza a los elementos del diseño, (casos de uso, clases, interfaces, etc.), en piezas más pequeñas y manejables en forma independiente. De tal forma que el sistema puede ser abstraído en su totalidad.
Formato de diseño de base de datos	Este documento es un "modelo" que contiene los datos necesarios como: tablas, campos, atributos, etc. Que son necesarios para implementar la base de datos. Es importante señalar que los elementos de la base de datos deberán estar expresados en la sintaxis correspondiente al manejador de bases de datos establecido.
Modelo de diseño	Es un "modelo de objetos" que muestra: los subsistemas y clases de diseño. Este modelo contempla los requerimientos funcionales y los no funcionales. También es considerado como un elemento de entrada fundamental para el flujo de

	implementación.
Interfaz	Son diseños de pantallas que simulan la realización de los casos de uso. En esta interfaz aun no existe nada programado, simplemente se simula el comportamiento del sistema.

Tabla 6.6. Adaptación de artefactos del flujo de diseño del Departamento de Informática con base en el Proceso Unificado.

6.4. Comparación del flujo de diseño entre el Proceso Unificado y el Departamento de Informática del IIB.

En las siguientes tablas se presenta la convergencia y divergencia en los elementos que integran el flujo de diseño en el Proceso Unificado; así como su adaptación al flujo de diseño en el Departamento de Informática.

Actividades	PU	DI	Justificación
Diseñar arquitectura	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Diseñar casos de uso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Diseñar clases	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Elaborar diagrama entidad relación	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	La actividad es necesaria porque facilita el diseño de la base de datos independientemente del manejador que se valla a utilizar.
Diseñar subsistemas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Diseñar base de datos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	En el Departamento de Informática, esta actividad es agregada porque, el diseño de la base de datos deberá ser realizado con base al manejador de base de datos que se ha establecido. Esto, para facilitar la implementación de la misma.
Elaborar modelo de diseño	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	En PU, el modelo de diseño, se realiza conjuntamente con otros modelos en la primera actividad; esto ocasiona que dicho artefacto se esté modificando continuamente porque los elementos que lo integrarán se diseñan posteriormente. Es por esto que se agrega como una actividad particular en el Departamento de Informática. Se debe generar una vez que se ha realizado el diseño de todos los elementos; la finalidad es que el modelo este lo más completo posible porque es una entrada principal para el flujo de implementación.
Diseñar interfaz	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Esta actividad se agrega en forma independiente en el Departamento de Informática porque se considera oportuno que hasta este momento, en donde se ha realizado un análisis y diseño detallado de los requerimientos del sistema, la interfaz debe diseñarse; esto, para que contemple todos los elementos necesarios. En caso de que se deba modificar, los cambios serán mínimos.

Tabla 6.7. Convergencias y divergencias en las actividades que integran el flujo de diseño del Proceso Unificado, así como su adaptación al flujo de diseño del Departamento de Informática.

TESIS CON
FALLA DE ORIGEN

Trabajadores	PU	DI	Justificación
Arquitecto de software	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Ingeniero de casos de uso	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Ingeniero de componentes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Diseñador de bases de datos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Este personaje se agrega en el Departamento de Informática porque la labor de para la creación del diagrama entidad relación y el diseño de bases de datos es considerada como una actividad especializada y deberá ser realizada por un trabajador con conocimientos en el diseño de bases de datos y en el manejador que se valla a implementar.

Tabla 6.8. Convergencias y divergencias en los trabajadores que integran el flujo de diseño del Proceso Unificado, así como su adaptación al flujo de diseño del Departamento de Informática.

Artefactos	PU	DI	Justificación
Diseño de la arquitectura	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Artefacto realizado en lugar de: Descripción de la arquitectura (vista del modelo de diseño); Modelo de despliegue y Descripción de la arquitectura (vista del modelo del modelo de despliegue). Esto porque los tres artefactos se enfocan para el desarrollo de proyectos muy grandes que involucran a diversos subsistemas como pueden ser, otras grandes corporaciones. En el caso particular del Departamento de Informática, no es necesario generar los 3 artefactos mencionados porque

			con el diseño de la arquitectura se pueden identificar, representar y manejar todos los elementos que los sistemas del Departamento de Informática requieren.
Realización de casos de uso de diseño	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Clases del diseño	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Elaborar diagrama entidad relación	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Artefacto agregado para facilitar el diseño de la base de datos.
Subsistemas de diseño	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Formato de diseño de base de datos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	El artefacto no figura el Proceso Unificado, es creado en el flujo de diseño porque contiene los datos necesarios como: tablas, campos, atributos, etc.; que son necesarios para implementar la base de datos.
Descripción de la arquitectura (vista del modelo de diseño)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Artefacto sustituido por el de <i>diseño de arquitectura</i> .
Modelo de despliegue	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Artefacto sustituido por el de <i>diseño de arquitectura</i> .
Descripción de la arquitectura (vista del modelo del modelo de despliegue)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Artefacto sustituido por el de <i>diseño de arquitectura</i> .
Modelo de diseño	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Interfaz	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Tabla 6.9. Convergencias y divergencias en los artefactos que integran el flujo de diseño del Proceso Unificado, así como su adaptación al flujo de diseño del Departamento de Informática.

TESIS CON
FALLA DE ORIGEN

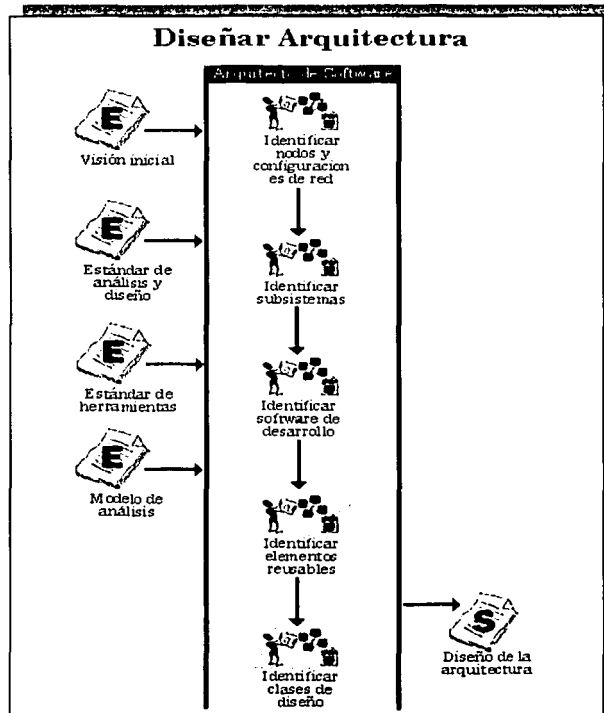
6.5 Detalle de las actividades del diseño en el Departamento de Informática.

A continuación se presenta cada una de las actividades con sus respectivos pasos que forman el flujo de diseño.

Actividad: Diseñar Arquitectura.

En esta actividad se puede apreciar que existen tres artefactos de entrada: (*Visión inicial*, *Estándar de análisis y diseño*, *Estándar de herramientas* y *Modelo de análisis*), y uno de salida (*Diseño de la arquitectura*).

En la tabla 6.6. se puede ver la descripción de todos los artefactos del flujo de diseño y en el apéndice G se pueden apreciar todos los formatos.



TESIS CON
FOLIO DE ORIGEN

Los pasos de esta actividad son:

Identificar nodos y configuraciones de red. Consiste en especificar las características de los equipos que se utilizarán para la puesta en marcha del sistema. Si el sistema se desarrolla en ambiente multiusuario, se deberá especificar el servidor, el cliente y el protocolo de comunicación; si el sistema se desarrolla en ambiente monousuario, entonces solo se debe especificar el equipo en el que correrá.

Para verificar el ambiente en el que se desarrollará el sistema se debe consultar la *visión inicial* y la información se registra en la sección 1 del *Diseño de la arquitectura* llamada *Identificación de nodos y configuraciones de red*.

Identificar subsistemas. Consiste en identificar todos los subsistemas de diseño para que el proyecto pueda contemplarse en piezas manejables, con la finalidad de establecer una división del trabajo; o bien, si es necesario incorporar nuevos elementos, entonces, será fácil la integración o refinamiento.

Los paquetes definidos en el análisis se convierten en subsistemas en el flujo de diseño; por lo tanto, los paquete establecidos en la arquitectura se deberán registrar como subsistemas en la sección 2 del *Diseño de la arquitectura* nombrada *Subsistemas de diseño*.

Identificar software de desarrollo. Aquí se debe especificar todo el software que se debe utilizar para el desarrollo del sistema. Entre el software que se debe definir tenemos: sistema operativo, lenguajes de programación, manejadores de bases de datos, herramientas de diseño, y demás elementos necesarios para la puesta en marcha del sistema. Para especificar el software de desarrollo se debe

TESIS CON
FALLA DE ORIGEN

consultar el *Estándar de herramientas*¹² en la sección denominada *Herramientas para la construcción*.¹³

Normalmente estos elementos los encontraremos en la capa intermedia de la arquitectura.

La información sobre el software que se utilizará en el desarrollo del sistema se debe especificar en la sección 3 del *Diseño de la arquitectura* designada *Software de desarrollo*.

Identificar elementos reusables. Consiste en consulta la sección denominada "Biblioteca de componentes" del estándar de herramientas con la finalidad de ubicar procedimientos, funciones y programas que puedan ser reutilizados para el proyecto en desarrollo; esto con la finalidad de aprovechar los recursos disponibles.¹⁴

Los elementos reusables identificados se deberán registrar en la sección 4 del *Diseño de la arquitectura* llamada *Componentes reusables*.

Identificar clases de diseño. Consiste en ubicar únicamente las clases principales o relevantes, con la finalidad de que se consideren para el diseño de la

¹² El estándar de codificación puede consultarse en la herramienta "PDSIIB" o bien, en la referencia [Salazar J.A., 2003].

¹³ Con respecto a la definición del software de desarrollo, la pretensión es ir integrando nuevo software al estándar de herramientas, debido a que los requerimientos de los sistemas no siempre pueden cubrirse con un solo lenguaje o manejador de base de datos. Eso aunado al cambio continuo o mejoras que ocurren con respecto a las herramientas de programación y definición de bases de datos entre otras.

¹⁴ La sección de componentes reusables del estándar de herramientas se deberá ir conformando en la medida que se liberen y agreguen nuevos elementos. Para que un componente reusable pueda agregarse al estándar deberá ser evaluado por al menos dos integrantes del equipo de desarrollo.

arquitectura, sin embargo, la identificación definitiva y el diseño de clases, se realizará con más detenimiento en el diseño de clases.¹⁵

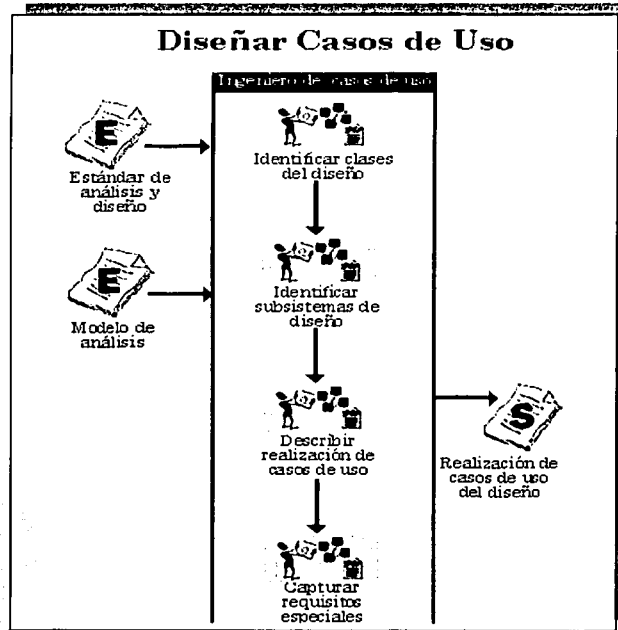
Esta información deberá registrarse en la sección 5 del *Diseño de la arquitectura* nombrada *Clases de diseño principales*.

TESIS CON
FALLA DE ORIGEN

¹⁵ Las clases de diseño principales normalmente son las que se representan en el diagrama de clases elaborado en el flujo de análisis; sin embargo conforme el trabajo de análisis y diseño avanza puede ocurrir que se agreguen nuevas clases relevantes que deberán incluirse en éste momento.

Actividad: Diseñar Casos de Uso.

En esta actividad se puede apreciar que existen dos artefactos de entrada: (*Modelo de análisis y Estándar de análisis y diseño*), y uno de salida (*Realización de casos de uso de diseño*).



Los pasos de esta actividad son:

Identificar clases del diseño. En este paso se deben identificar todas las clases de diseño necesarias para realizar los casos de uso. Para poder realizar esto,

TESIS CON
FALLA DE ORIGEN

tomamos como referencia las *Clases del análisis*, las estudiamos e identificamos las clases del diseño faltantes para la realización integral del caso de uso.¹⁶

Las clases de dominio del problema, de interfaz de usuario y del manejo de datos, resultantes en el análisis, aquí las denominaremos clases del diseño.

Las clases del diseño deben registrarse en la sección 1 de la *Realización de casos de uso de diseño* denominada *Clases de diseño por caso de uso*.

Identificar subsistemas de diseño. Consiste en identificar los subsistemas de diseño que intervienen en la realización de casos de uso del diseño.¹⁷

Los subsistema de diseño deberán registrarse en la sección 2 de la *Realización de casos de uso de diseño* denominada *Clases de diseño por subsistema*.¹⁸

Describir realización de casos de uso. Una vez que se han identificado las clases y los subsistemas de diseño que participan en la realización de un caso de uso, ahora se debe describir la forma en que interactúan para lograr el objetivo. Para realizar esta descripción se utilizarán los diagramas de secuencia que describen a detalle la forma en que se realizan los casos de uso. En este tipo de diagramas se pueden identificar claramente actores, objetos y secuencia de pasos para llevar a cabo el caso de uso.



¹⁶ Las realizaciones de casos de uso involucran la relación de varias clases de diseño con la finalidad de que el sistema proporcione los resultados esperados.

¹⁷ Un subsistema de diseño esta integrado por más de una clase.

¹⁸ El subsistema se deberá llamar igual que el caso de uso. Por ejemplo si el caso de uso se llama "bibliografía", entonces el subsistema se denominará "bibliografía".

Verificar la sección 10 del *Estándar de análisis y diseño* para identificar los símbolos (actor, actividad, secuencia, clase, etc.) que se utilizan los diagramas de secuencia.¹⁹

Los diagramas de secuencia se realizan en la sección 3 de la *Realización de casos de uso de diseño* denominada *Diagramas de secuencia por caso de uso*.

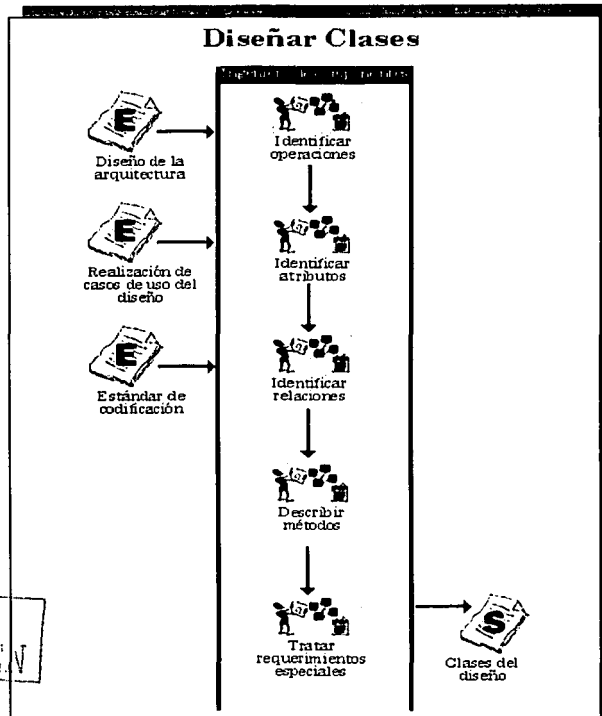
Capturar requerimientos especiales. Consiste en capturar los requerimientos especiales que se identificaron en el análisis, aquí corresponde decir como van a ser tratados. Los requerimientos pueden ser con respecto a: persistencia, seguridad, tolerancia a fallos, etc.

Los requerimientos especiales deberán capturarse en la en la sección 4 de la *Realización de casos de uso del diseño* designada *Captura de requerimientos especiales*.

TESIS CON
FALLA DE ORIGEN

¹⁹ Los diagramas de secuencia se debe realizar para cada caso de uso y puede ser que un caso de uso requiera de más de un diagrama.

Actividad: Diseñar Clases. En esta actividad se puede apreciar que existen dos artefactos de entrada: (*Diseño de la arquitectura, Realización de casos de uso del diseño y Estándar de codificación*), y uno de salida (*Clases del diseño*).



TESIS CON
FALLA DE ORIGEN

Los pasos de esta actividad son:

Identificar operaciones. Consiste en definir las operaciones de cada una de las clases. La definición se debe realizar mediante la utilización de la "sintaxis del lenguaje de programación" que se haya seleccionado; para saber cuál es, se consulta la sección 3 del *Diseño de la arquitectura*. Las clases deben realizar todas las operaciones necesarias para que soporten todas las realizaciones de casos de uso. Para identificar el tipo de operaciones consultar la sección 11 de

Estándar de análisis y diseño. La información derivada de este paso se registra en la sección 1 de las *Clases del diseño*.

Identificar atributos. Consiste en definir los atributos requeridos por las clases de diseño. Un atributo será siempre requerido por una o más operaciones de las clases. Al establecer los atributos de deben tomar en cuenta los siguientes puntos:

1. Si se usa un lenguaje de programación orientado a objetos entonces, se identifican conforme a las clases. Porque una clase se representa con un símbolo que permite identificar rápidamente los atributos.
2. Si no se utiliza un lenguaje de programación orientado a objetos entonces, la forma de identificarlos será la siguiente: Cada una de las clases se convertirá en un dato de tipo registro que contendrá como campos a los atributos.

Independientemente de la forma de identificar a los atributos, éstos deberán registrarse en el símbolo de clases en la parte que corresponde a los atributos de la sección 1 de las *Clases del diseño* respetando las normas del lenguaje de programación.

Identificar relaciones. Debido a que los objetos de diseño interactúan unos con otros, se debe identificar el tipo de relación que entre ellos se establece.

Si se usa un lenguaje de programación orientado a objetos entonces las relaciones pueden ser de: dependencia²⁰, asociación²¹ o generalización²²;

TESIS CON
FALLA DE ORIGEN

²⁰ Relación entre dos elementos, en la cual un cambio a un elemento (elemento independiente) puede afectar a la semántica del otro elemento (elemento dependiente).

²¹ La agregación es un tipo especial de asociación, que representa una relación estructural entre un todo y sus partes.

²² Tipo de especialización / generalización en la cual los objetos del elemento especializado (el hijo) pueden sustituir a los objetos del elemento general padre (el padre). De esta forma, el hijo comparte la estructura y el comportamiento del padre.

mismas que pueden consultarse en la sección 10 del *Estándar de análisis y diseño*.

En caso contrario, se deben tomar en cuenta los siguientes puntos:

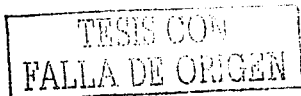
1. Cuando las clases son de Interfaz Humana permanecen de la misma forma porque al final serán ventanas.
2. Cuando las clases son del Dominio del Problema y existe algún tipo de relación con otras clases, se generará un nuevo dato de tipo registro.
3. Cuando las clases son de Manejo de Datos se convierten automáticamente en funciones o procedimientos; por lo que la relación se maneja a través del tipo de datos que manejan las funciones o procedimientos.

La información se registra en la sección 1 de las *Clases del diseño* a un costado del símbolo de la clase.

Describir métodos. Consiste en dar una descripción de las operaciones, identificando tipos de datos que reciben y tipos de datos que regresan.

La descripción de todos los métodos de las clase se hace con base en el *Estándar de codificación*²³ y debe registrarse en la sección 2 de las *Clases del diseño*.

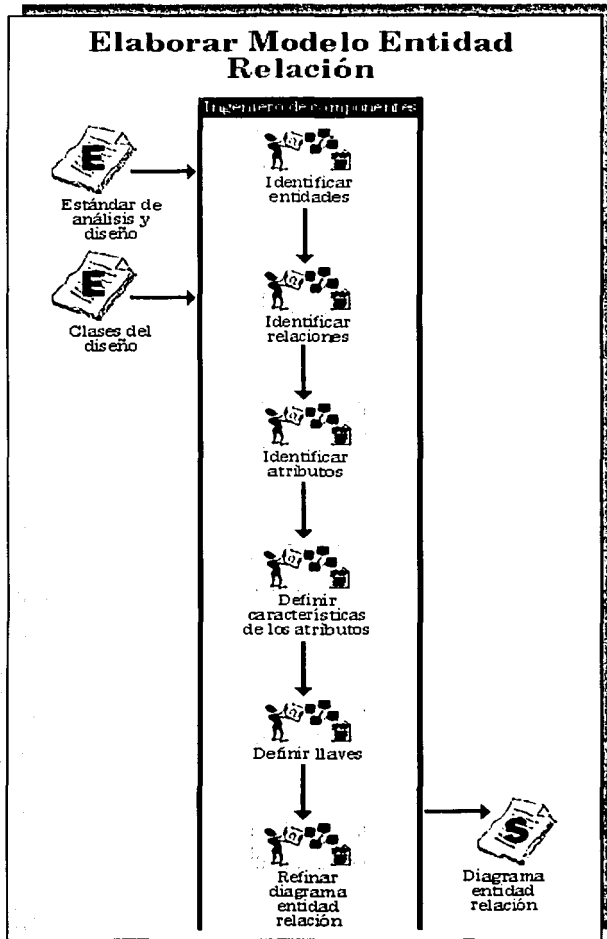
Tratar requerimientos especiales. Se refiere a que en esta parte se deben tomar en cuenta todos los requerimientos especiales que hubiesen surgido. Para esto se deberán agregar atributos, métodos o relaciones que se requieran para cubrir estos requerimientos. Esta información se deberá registrar en la sección 1 y 2 de las *Clases del diseño*.



²³ El estándar de codificación puede ser consultado en la herramienta "PDSIIB" o en la referencia de [Ramírez 2003].

Actividad: Realizar Diagrama Entidad Relación.

En esta actividad se observan dos artefactos de entrada; (*Estándar de análisis y diseño* y *Clases del diseño*), y uno de salida, (*Diagrama entidad relación*).



TESIS CON FALLA DE ORIGEN

Los pasos de esta actividad son:

Identificar entidades. Consiste en identificar todas aquellas entidades que participan en la ejecución de procesos internos del sistema; tienen como característica; que son elementos del mundo real y la información que manejan necesita ser almacenada en algún medio físico. Las entidades normalmente serán todas aquellas *Clases del dominio del problema* que tengan su clase equivalente en las *Clase del manejo de datos*. Para registrar las entidades se utiliza el símbolo de entidades y se pintan en el *Diagrama entidad relación*.

Identificar relaciones. Consiste en identificar la forma en cómo interactúan las entidades. La siguiente figura es un ejemplo en como se deben expresar las relaciones entre entidades.

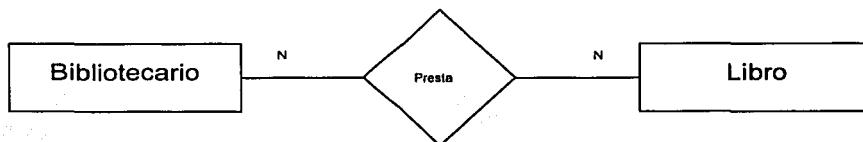


Figura 5.1. Relación entre la entidad bibliotecario y la entidad libro.

La entidad bibliotecario se relaciona con la entidad libro a través de la relación presta y se lee como sigue:

Muchos *bibliotecarios* pueden prestar muchos *libros* y muchos *libros* pueden ser prestados por muchos *bibliotecarios*. A este tipo de relación se le llama *relación de muchos a muchos*. Para establecer las relaciones se utiliza el símbolo de relación y el de asociación marcando todas las relaciones que existan; todo esto se debe hacer en el *Diagrama entidad relación*.

Identificar atributos. Los atributos son aquellos datos de las clases del diseño que necesitan ser almacenados. De la relación entre dos entidades puede generarse

uno o más atributos, es decir, información que requiera ser almacenada como producto de esa relación. Por ejemplo: la hora en la que se prestan los libros. Los atributos de cada relación se registran dentro del símbolo de relación en el *Diagrama entidad relación*.

Definir características de los atributos. Consiste en registrar en la sección 2 del diagrama entidad relación el tipo y el tamaño de los mismos. El registro se hace en el costado derecho de los atributos o campos como lo establece el estándar de análisis y diseño en la sección 7.

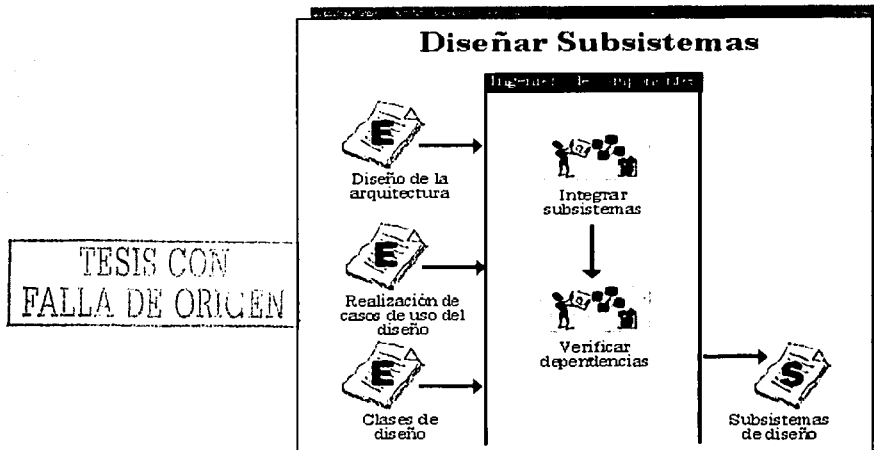
Definir llaves. Consiste en especificar las llaves primarias, secundarias y foráneas. Para definir los tipos de llaves se deberá consultar la *sección 9 del Estándar de análisis y diseño*. La designación de llaves se registra en la *sección 2 del diagrama entidad relación* en el costado izquierdo del atributo o campo que representará el tipo de llave.

Refinar diagrama entidad relación. Con base en todos los elementos se depura y pinta el diagrama definitivo, evitando en la medida de lo posible que las relaciones entre entidades se crucen; con ello se pretende que el diagrama sea lo más claro posible. El diagrama resultante es el denominado *Diagrama entidad relación*.

TESIS CON
FALLA DE ORIGEN

Actividad: Diseñar Subsistemas.

En esta actividad se puede apreciar que existen tres artefactos de entrada: (*Diseño de la arquitectura, Realización de casos de uso y Clases del diseño*), y uno de salida (*Subsistemas de diseño*).



Los pasos de esta actividad son:

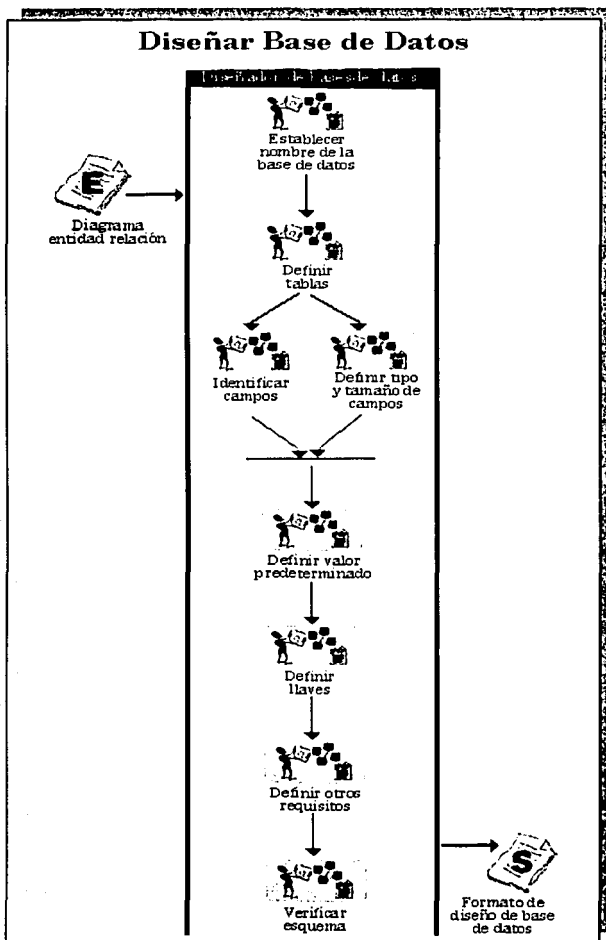
Integrar subsistema. Consiste en establecer cuáles son las clases que conformarán los subsistemas identificados en la arquitectura. Para ello se verifican todas las clases que intervienen en la realización de casos de uso del diseño, y se van colocando dentro del símbolo de paquetes. Es importante que al establecer el contenido de los subsistemas se evite, en la medida de lo posible, la dependencia entre ellos. Los subsistemas se registran en el artefacto *Subsistemas de diseño*.

Verificar dependencias. Consiste en verificar que los subsistemas solo contienen clases relacionadas funcionalmente. La pretensión es lograr que los subsistemas sean tan independientes como sea posible.

En caso de que exista demasiada dependencia entre ellos se debe considerar la reubicación de las clases contenidas en los subsistemas que son demasiado dependientes de otros. La verificación se hace sobre los *Subsistemas de diseño*.

Actividad: Diseñar Base de Datos.

En esta actividad se puede apreciar que existe un artefacto de entrada: (*Diagrama entidad relación*), y uno de salida (*Formato de diseño de base de datos*). La información que surja de todos los pasos de ésta actividad se debe registrar en las diferentes secciones del *Formato de diseño de base de datos*.



TESIS CON
VALOR DE ORIGEN

Los pasos de esta actividad son:

Establecer nombre de la base de datos. Consiste en asignar un nombre a la base de datos que sea representativo del tipo de información que va a almacenar. La información de cada uno de los pasos deberá registrarse en el *Formato de diseño de base de datos*.

Definir tablas. Consiste en establecer las tablas. Para ello se toman todas las entidades y las relaciones del diagrama entidad relación y se convierten en tablas.

Identificar campos. Consiste en tomar todos los atributos de las entidades y relaciones del diagrama entidad relación y convertirlos en campos. Los nombres deben respetarse por lo que se llamarán igual.

Definir tipo y tamaño de campos. Consiste en definir el tipo y tamaño de cada campo. Para ello se observan los tipos y tamaños de los atributos el diagrama entidad relación y se definen de la misma forma con base en el manejador de bases de datos que se va a utilizar.

Definir valor predeterminado. Consiste en asignar el "valor inicial" que deberá contener cada campo en caso de ser necesario. Si en el modelo entidad relación existe esta restricción para los atributos, entonces ahora se hace para los campos.

Definir llaves. Consiste identificar las llaves primarias y foráneas para cada tabla. Con base a los atributos definidos como cualquier tipo de llave en el diagrama entidad relación, se definen las mismas llaves pero ahora esta definición se hace sobre los campos considerando el manejador de bases de datos a utilizar.

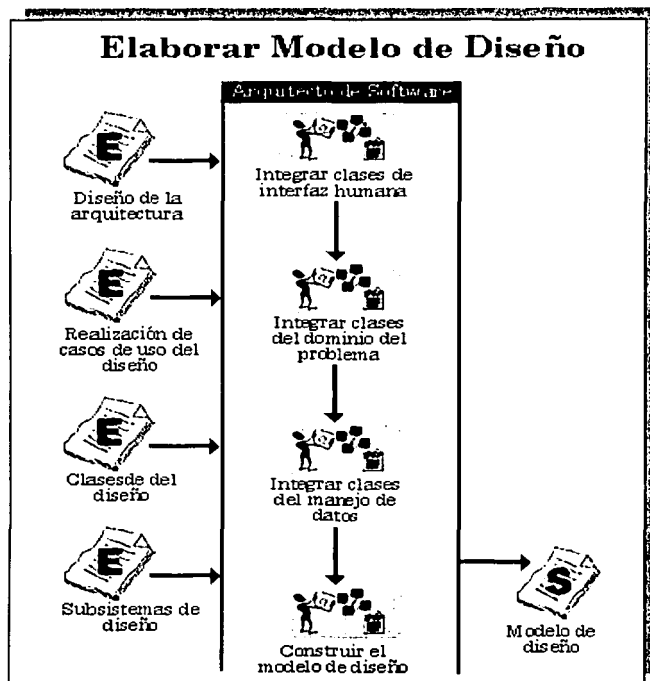
Definir otros requerimientos. Consiste en establecer otros requerimientos de los campos, como son: rangos permitidos, validaciones, restricciones, entre otros.

Verificar esquema. Consiste en verificar que las características de cada tabla establecidas en los pasos anteriores son correctas. La verificación se hace sobre *Formato de diseño de base de datos* y si existen errores se corrigen en el momento. Esto con la finalidad de garantizar la integridad referencial de la base de datos.

Actividad: Elaborar Modelo de Diseño.

Para realizar esta actividad se necesitan los siguientes artefactos de entrada:
(*Diseño de la arquitectura, Realización de casos de uso de diseño, Clases del diseño y Subsistemas de diseño*).

El resultado de esta actividad es el *Modelo de diseño*.



TESIS CON
FALLA DE ORIGEN

Los pasos de esta actividad son:

Integrar clases de interfaz humana. Consiste en integrar todas las clases de interfaz humana en un solo subsistema que se colocará en la primera capa. La integración se realiza con base al patrón de arquitectura elegido; para ello se utiliza la sección 1 del *Modelo de diseño* denominada *Clases de interfaz humana*.

Integrar clases del dominio del problema. Consiste en integrar todas las clases del dominio del problema en un solo subsistema que se colocará en la capa intermedia. Esta integración se realiza con base al patrón de arquitectura elegido; para ello se utiliza la sección 2 del *Modelo de diseño* designada *Clases del dominio del problema*.

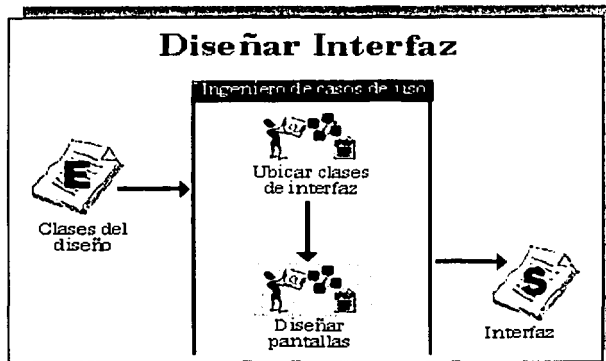
Integrar clases del manejo de datos. Consiste en integrar todas las clases del manejo de datos en un solo subsistema que se colocará en la tercera. Esta integración se realiza con base al patrón de arquitectura elegido; para ello se utiliza la sección 3 del *Modelo de diseño* llamada *Clases del manejo de datos*.

Construir el modelo de diseño. Para construir el modelo de diseño se coloca a los subsistemas en cada una de las capas de la arquitectura; se pinta con una línea punteada la división de cada capa y se establece la relación entre las capas. El resultado será el *Modelo de diseño* ubicado en la última sección.

TESIS CON
FALLA DE ORIGEN

Actividad: Diseñar Interfaz.

En esta actividad se puede apreciar que existe un artefacto de entrada: (*Clases del diseño*), y uno de salida (*Interfaz*).



Los pasos de esta actividad son:

Ubicar clases de interfaz. Consiste en seleccionar de todas las clases, aquellas que serán tomadas en cuenta para diseñar una interfaz. Es decir, exclusivamente se tomarán las clases de Interfaz Humana porque solamente se mostrarán los datos sin realizar programar la funcionalidad. Para distinguir las clases de Interfaz Humana basta con colocar una marca de un determinado color sobre las *Clases del diseño*.

Diseñar pantallas. Consiste en diseñar todas las pantallas necesarias para completar la interfaz de usuario, es decir, la forma visual a través de la cual los actores interactuarán con el sistema. Este diseño deberá realizarse en el lenguaje de programación establecido previamente. El artefacto resultante será la Interfaz compuesta por varias pantallas.

TESIS CON
FALLA DE ORIGEN

Para la conclusión del capítulo se puede decir que el flujo de diseño tiene como objetivo crear un modelo que sirva para la implementación incluyendo las restricciones del lenguaje de programación, componentes reutilizables, sistema operativo, entre otros.

Se establecieron y diseñaron los siguientes artefactos: *Diseño de la arquitectura, Realización de casos de uso del diseño, Clases del diseño, Subsistemas de diseño, Formato de diseño de base de datos, Modelo de análisis e Interfaz.*

Luego entonces se puede decir que la principal aportación del diseño es precisamente el *Modelo del diseño* que conserva la estructura del análisis y sirve como base para el *Modelo de implementación*, pues incluye detalles del ambiente de implementación.

Nota Aclaratoria

Si bien el flujo de Implementación no se considero en el presente trabajo puede ser consultado en la tesis denominada *"La implementación, administración del proyecto, cambios y configuración del Proceso Unificado; adaptados a un caso práctico"*; de Ramírez Molina, Ana Yuri; ya que ambos trabajos y una tesis más (*"Pruebas, Ambiente y Distribución basados en el Proceso Unificado (Un caso práctico)"*); de Salazar Carmona, José Antonio.) forman parte del proyecto "Adaptación del Proceso Unificado a un Departamento de Informática".



CAPÍTULO 7

7. CONSTRUCCIÓN DE LA HERRAMIENTA "PDSIIB" (PROCESO DE DESARROLLO DE SOFTWARE DEL INSTITUTO DE INVESTIGACIONES BIBLIOGRÁFICAS).

En este capítulo se desarrollan los siguientes puntos relacionados con la construcción de la herramienta "PDSIIB" (Proceso de Desarrollo de Software del Instituto de Investigaciones Bibliográficas): definición y objetivos; proceso de creación; organización y manejo de la herramienta.

En el proceso de desarrollo de software de cualquier organización, no basta con la existencia del proceso como tal, es conveniente construir una herramienta, que guíe este proceso.

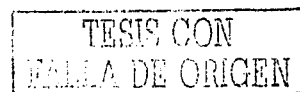
Para guiar el proceso, en el Departamento de Informática, se construyó la herramienta "PDSIIB" con base a la adaptación que se realizó del Proceso Unificado.

Los siguientes apartados permiten conocer las características de la herramienta.

7.1. Definición y objetivos de la herramienta PDSIIB.

Definición de PDSIIB

"PDSIIB" (Proceso de Desarrollo de Software del Instituto de Investigaciones Bibliográficas), es una herramienta que permite guiar el Proceso de Desarrollo de Software del Departamento de Informática del Instituto de Investigaciones



Bibliográficas. A través de ella se puede ubicar fase, flujo, actividad y los pasos que la componen.

Objetivos de PDSIIB

- Guiar el Proceso de Desarrollo de Software del Departamento de Informática del Instituto de Investigaciones Bibliográficas.
- Generar el flujo general del proceso integrando cada una de las actividades de los flujos de trabajo en las fases correspondientes.
- Revisar los nueve flujos de trabajo que conforman el Proceso de Desarrollo de software del Departamento de Informática.
- Conocer en que fase, flujo, actividad y paso se encuentra el proceso de desarrollo de sistemas.
- Unificar roles y diseño de documentación para la construcción de la herramienta.
- Establecer un estándar de diagramación para la elaboración de los nueve flujos de trabajo que integran el Proceso Unificado.
- Establecer un estándar de documentación (tipo, color y tamaño de fuente).
- Crear plantillas de los productos a generar.

7.2. Proceso de creación de la herramienta PDSIIB.

Es importante señalar que al construir la herramienta fue muy importante la colaboración, integración y contribución de los tres trabajos de tesis ya señalados; en donde cada uno trata del acoplamiento de tres flujos de trabajo del Proceso Unificado apoyándose en ellos para la construcción de la herramienta.

Esta herramienta está desarrollada en Dreamweaver (Tejedor de Sueños; herramienta especializada para la construcción de páginas y sitios web. Dreamweaver proporciona recursos muy variados, entre los que destacan:

impresión de libros, páginas de ayuda en línea, elaboración de tutoriales, entre otros.

La herramienta "PDSIIB" puede ser visualizada a través de un explorador como Explorer versión 5.0 o superior, y Netscape versión 4.0 o superior.

Para el diseño de las imágenes se utilizó Fireworks (Firewoks es una herramienta para la creación y edición de imágenes; además, permite crear menús, diseñar efectos para sitios web; y exportar archivos en un formato compatible para páginas y sitios en Internet.

Para la construcción de la herramienta fue necesario realizar las actividades que a continuación se listan.

- Revisar la forma en cómo se desarrolla el software en el Departamento de Informática.
- Estudiar el Proceso Unificado.
- Revisar el funcionamiento de RUP (Rational Unified Process).
- Analizar los flujos de trabajo del Proceso Unificado.
- Identificar qué actividades de la forma actual de hacer sistemas se asemejan con las actividades del Proceso Unificado.
- Adaptar los flujos de trabajo.
- Establecer actividades, roles y artefactos para cada uno de los flujos de trabajo.
- Generar los artefacto en cada flujo.
- Establecer estándares para la construcción de la herramienta.
- Unificar roles y artefactos entre los flujos de trabajo.
- Establecer herramientas de software para el desarrollo y construcción de la herramienta.
- Revisar periódicamente entre colegas la adaptación de los flujos, señalando en caso necesario las correcciones pertinentes.

TESIS CON
FALLA DE ORIGEN

- Establecer el flujo general del Proceso de Desarrollo de Software mediante la integración de los flujos de trabajo; resolviendo incongruencias e inconsistencias.
- Diseñar de las imágenes que se utilizan en la construcción de la herramienta.
- Establecer colores, tipo y tamaño de letra.
- Supervisar los avances de la construcción de la herramienta.
- Construir la herramienta.
- Probar la herramienta siguiendo en flujo general con un ejemplo real del Departamento de Informática.
- Corregir los errores identificados durante la prueba.

7.3. Organización de la herramienta PDSIIB.

La herramienta se encuentra estructurada a través de directorios y archivos que siguen una estructura de árbol.

Para el funcionamiento adecuado de la herramienta "PDSIIB" fue muy importante establecer el flujo general, mismo que se encuentra estructurado por fases y flujos. Es por ello que facilita identificar perfectamente en que parte del proceso se encuentra el desarrollo de software.

La herramienta está estructurada por directorios y archivos que facilitan su manejo. En la figura 7.1. muestra la estructura de la herramienta.

TESIS CON
FALLA DE ORIGEN

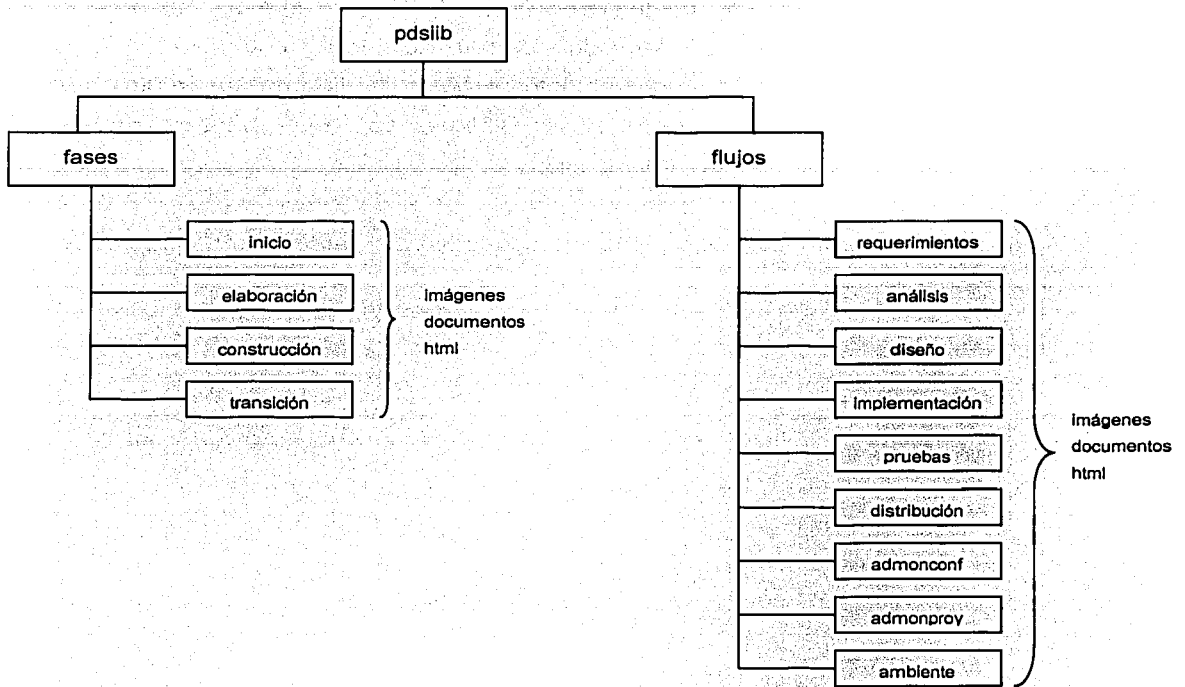


Figura 7.1. Estructura de la herramienta pdslib. Las fases y flujos contienen tres directorios: imágenes, documentos y html.

7.4. Manejo de la herramienta PDSIIB.

TESIS CON FALLA DE ORIGEN

Debido a que la estructura de la herramienta está pensada para no perderse en ella, en todo momento se puede ubicar fase, flujo, actividad, y paso en el que se está. Si en el paso en que se encuentra, se genera un artefacto, entonces puede visualizarse el artefacto.

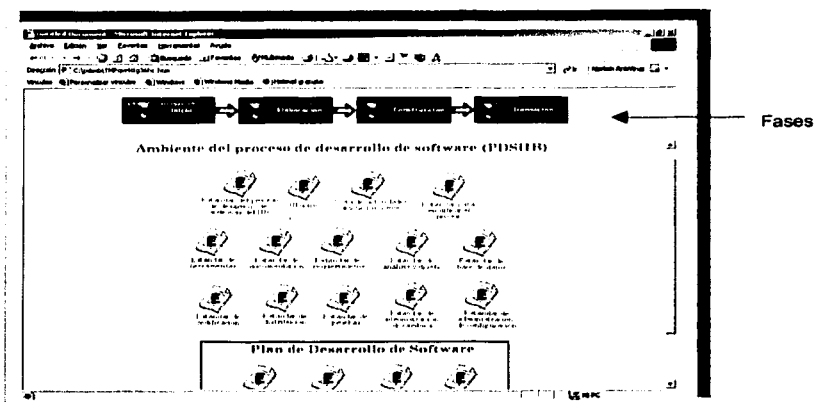
Esta característica es propia de la herramienta PDSIIB, dado que RUP (Proceso Unificado de Rational), cuenta con una herramienta que tiene una estructura mas general, ciertamente es muy completa, pero también es cierto que al estar navegando por ella, de repente no se sabe en que fase se está.

Otra característica particular de la herramienta PDSIIB, es que fue hecha a la medida del Departamento de Informática, es por ello que los estándares que se generan son con base a los requerimientos propios del tipo de desarrollos que realizan en el Instituto de Investigaciones Bibliográficas. RUP sin embargo, es general, para cualquier tipo de organización.

A continuación describiremos a grandes rasgos la forma para poder navegar en la herramienta.

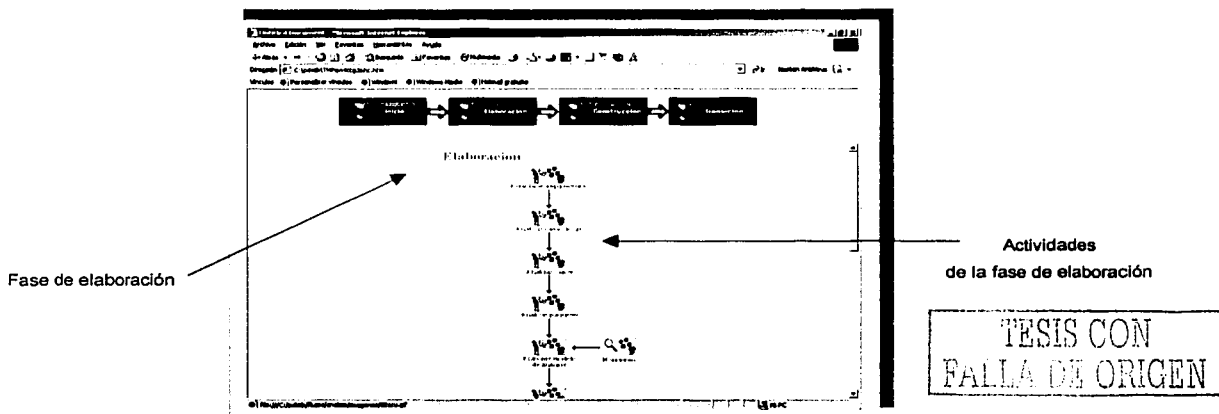
TESIS CON
FALLA DE ORIGEN

La presentación inicial de la herramienta es la siguiente.



La pantalla inicial muestra en la parte superior las cuatro fases que conforman el proceso de desarrollo de software. Para ingresar a las fases, sólo hay que hacer clic sobre la fase deseada. En el centro se encuentran los estándares creados para el proceso de desarrollo de software del IIB:

Si se hace clic en "Elaboración" se apreciará la siguiente ventana que muestra las actividades que conforman dicha fase.



En la siguiente pantalla podemos observar los siguientes elementos:

En la parte izquierda tenemos las actividades de la fase de "Elaboración". Por ello se sabe en que fase está.

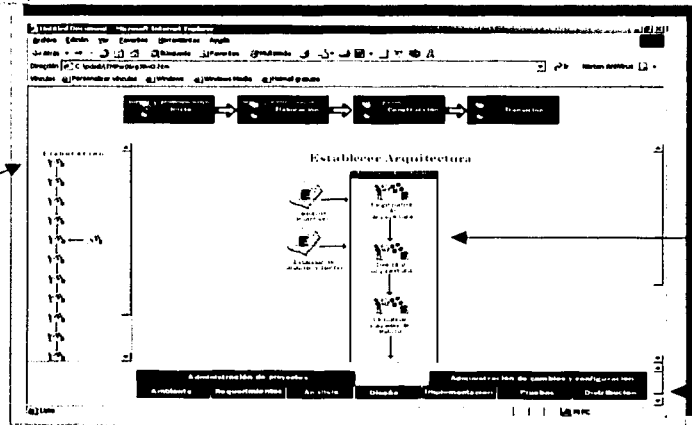
En la parte inferior aparece iluminado el flujo en el que se está.

En el centro se tiene los pasos de la actividad "Establecer Arquitectura".

Con los elementos mencionados se puede decir lo siguiente: Se está en la fase de elaboración; en la primera actividad que es "Establecer Arquitectura" y, también se puede saber que la actividad corresponde al flujo de "Análisis".

TESIS CON
FALLA DE ORIGEN

Actividades de la fase de elaboración. Para poder ver las actividades basta con hacer clic sobre cualesquiera de ellas.



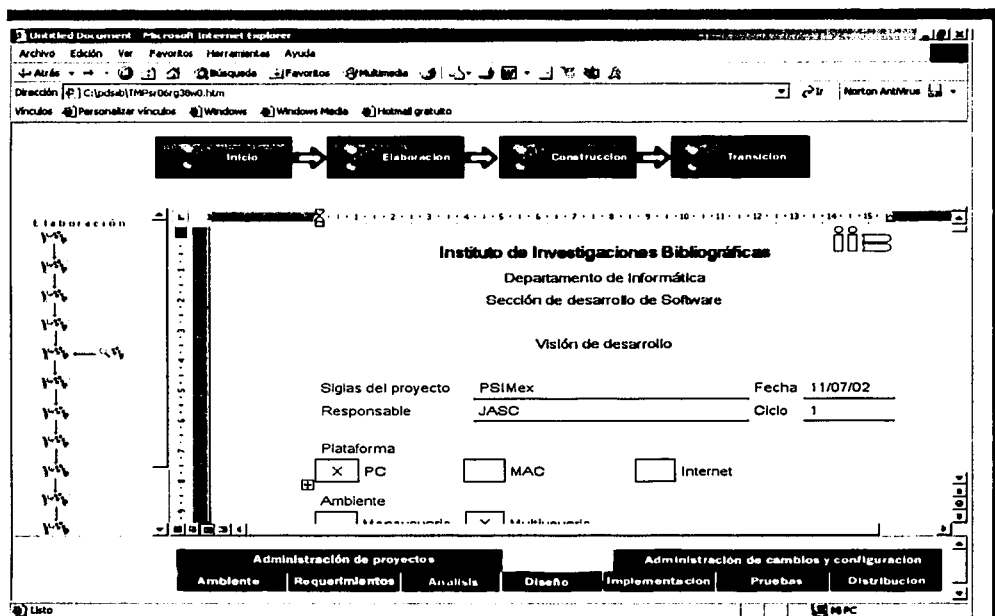
Pasos de la actividad establecer arquitectura

Flujos

Flujo de análisis

Si se desea ver la descripción de los pasos de una de las actividades, entonces se hace clic sobre el paso deseado.

Para poder tener acceso a un artefacto, basta con hacer clic sobre él.



La pantalla muestra el artefacto de entrada denominado "Visión de desarrollo" que se genera en última actividad del flujo de "Requerimientos" y que figura como entrada indispensable en el flujo de análisis.

TESIS CON
FALLA DE ORIGEN

En términos generales, basta con desplazar el mouse por los objetos y en donde aparezca el cursor en forma de manita (símbolo que indica más información), se puede dar clic para tener acceso a:

- Fases
- Flujos
- Actividades de fases o flujos
- Pasos de actividades
- Artefactos que se generan en cada actividad o paso
- Descripciones de actividades o pasos
- Detalle de los pasos representados a través de diferentes diagramas

Se puede concluir este apartado mencionando que es muy útil e importante el hecho de contar con un proceso de desarrollo de software a la medida; sin embargo, esta importancia se incrementa cuando existe una herramienta que guía este proceso.

TESIS CON
FALLA DE ORIGEN

CONCLUSIONES

A través del estudio realizado al Proceso Unificado se logró su adaptación al desarrollo de software del Departamento de Informática del Instituto de Investigaciones Bibliográficas.

La adaptación se realizó en tres flujos: requerimientos, análisis y diseño.

En el primer flujo se diseñaron y establecieron los siguientes artefactos: *Solicitud de desarrollo de software, Formato de Requerimientos, Diagramas de casos de uso, Formato de prioridad de casos de uso, Modelo de casos de uso y Visión de desarrollo.*

La importancia de dichos artefactos radica en que introducen al flujo de análisis.

En el segundo flujo se diseñaron y establecieron los siguientes artefactos: *Definición de la arquitectura, Clases del análisis, Realizaciones de casos de uso del análisis y Modelo de análisis.*

La importancia de dichos artefactos radica en que figuran como entradas indispensables para iniciar el flujo de diseño.

De igual forma en el tercer flujo se diseñaron y establecieron los siguientes artefactos: *Diseño de la arquitectura, Realización de casos de uso de diseño, Clases de diseño; Subsistemas de diseño, Interfaz, Formato de diseño de base de datos y Modelo de diseño.*

La importancia de dichos artefactos radica en que figuran como entradas indispensables para iniciar el flujo de implementación.

TESIS CON
FALLA DE ORIGEN

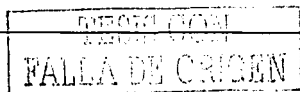
En cada flujo se diseñaron y establecieron estándares que se deberán utilizar en el proceso de desarrollo de software del Departamento de Informática. Dichos estándares tienen la finalidad de facilitar las actividades de los flujos de requerimientos, análisis y diseño. Los estándares son: de documentación; de requerimientos; de análisis y diseño; y, de diseño de bases de datos.

Mediante la integración de las tres tesis: **“Adaptación del Proceso Unificado en los flujos de trabajo de requerimientos, análisis y diseño: caso práctico”**; **“La implementación y administración del proyecto, cambios y configuración del Proceso Unificado adaptados a un caso práctico”**; y, **“Pruebas, Ambiente y Distribución basados en el Proceso Unificado (Un caso práctico)”**; se logró la integración de los nueve flujos de trabajo y la creación de la herramienta **“PDSIIB”** (Proceso de Desarrollo de Software del Instituto de Investigaciones Bibliográficas).

La herramienta creada demuestra que el Proceso Unificado se puede adaptar en forma satisfactoria a un caso práctico como es el Departamento de Informática del Instituto de Investigaciones Bibliográficas.

La herramienta **“PDSIIB”** fue probada en los flujos de requerimientos, análisis y diseño, en el sistema **“Sistema para las publicaciones sobre la Independencia de México: 1808-2003”**, del Departamento de Informática. Durante la prueba se detectaron errores (duplicación, modificación y ambigüedad de información), en el diseño de artefactos, los cuales fueron corregidos.

Todas y cada una de éstas actividades fueron de vital importancia en esta experiencia. Cada una de ellas contribuyó a alcanzar los objetivos de la adaptación del Proceso Unificado al Departamento de Informática; al establecimiento de estándares; a la elaboración de formatos; a la integración los nueve flujos del Proceso Unificado; y, a la construcción y prueba de la herramienta **“PDSIIB”**.



Finalmente, el producto de éste trabajo, aporta un nuevo concepto en la forma de crear sistemas tanto para el Departamento de Informática del Instituto de Investigaciones Bibliográficas, como para el desarrollo de otros sistemas.

Trabajos Futuros

- Se pretende seguir utilizando la herramienta "PDSIIB" en proyectos del Instituto de Investigaciones Bibliográficas.
- Con base en la utilización y actualización periódica de la herramienta; se pretende optimizarla y ponerla al alcance del usuario final; es decir, que éstos puedan llenar los formatos por vía Internet; por otra parte, se pretende que los artefactos se generen automáticamente; así como la administración de los proyectos.
- Se pretende elaborar una publicación en donde se integren los nueve flujos de trabajo adaptados al proceso de desarrollo de software.

TESIS CON
FALLA DE ORIGEN

GLOSARIO

Actividad

Ejecución no atómica en curso, la conforman más de un paso.

Actor

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo.

Análisis

Flujo de trabajo fundamental cuyo propósito principal es analizar los requerimientos mediante su refinamiento y estructuración.

Arquitecto de Software

Persona encargada de realizar el desarrollo de un producto de Software.

Arquitectura

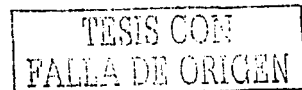
Conjunto de decisiones significativas acerca de la organización de un sistema de software, la selección de los elementos estructurales a partir de los cuales se compone el sistema, y las interfaces entre ellos, junto con su comportamiento.

Artefacto

En términos simples podemos definir un artefacto como la salida o la entrada de un proceso, como pueden ser: formatos, modelos, estándares, etc.

Artefactos

Son los documentos, formatos, productos, código, etc. generados al terminar una actividad.



Atributo

Una propiedad con nombre de un clasificador que describe el rango de valores que las instancias de una propiedad pueden tomar.

Campo

Unidad mínima de información dentro de la base de datos. Un campo puede contener una mezcla de caracteres alfabéticos, numéricos o ambos.

Capa

Parte bien definida de un sistema; se define a través de paquetes o subsistemas.

Caso de uso

Es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema; y que conduce a un resultado específico y observable de un actor.

Clase

Una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica.

Colaboración

Es un conjunto de elementos que interactúan entre sí para obtener un resultado específico.

Componente

Una parte física y reemplazable de un sistema que se ajusta y proporciona la realización de un conjunto de interfaces.

Coordinador

Persona encargada de asignar responsabilidad para el desarrollo de Software, así como, supervisar el cumplimiento de los planes establecidos para cada uno de los desarrollos del Departamento de Informática

Diagrama

La presentación física de un conjunto de elementos, usualmente representado como un grafo conectado de vértices (elementos) y arcos (relaciones).

Diseño

Flujo de trabajo fundamental cuyo propósito principal es el de formular modelos que se centren en los requerimientos y que prepara para la implementación y pruebas del sistema.

Distribución

Subproceso que se encarga de la entrega, instalación y preparación (documentación) del producto final de Software.

Entidad

Es un elemento del mundo real con características propias.

Entregable

Es el sistema con las funcionalidades establecidas para un ciclo de desarrollo, artefactos adicionales y herramientas de software que acompañan al sistema.

Fase

Período de tiempo entre dos hitos de un proceso de desarrollo.

Flujo

Conjunto de actividades orientada a cumplir objetivos específicos.

Flujo de trabajo ambiente

El propósito de este flujo es realizar las actividades para configurar el proceso, crear los artefactos acordes a la empresa de desarrollo de software y proporcionar soporte al equipo de desarrollo.

TESIS CON
FALLA DE ORIGEN

Flujo de trabajo distribución

El propósito de este flujo de trabajo es asegurar que el sistema estará disponible para los usuarios finales.

Flujo de trabajo pruebas El propósito de este flujo es asegurar que el sistema cumple con los requerimientos establecidos.

Formatos

Son las plantillas creadas para la generación de artefactos o para la recopilación de información importante en la ejecución de una actividad.

Herramientas que acompañan al sistema

Son todo tipo de software necesarios para que el sistema pueda ejecutarse de acuerdo a los requerimientos especificados.

Hito

Punto en el que han de tomarse principales decisiones de negocio. Cada fase acaba en un hito en el cual los gestores han de tomar decisiones cruciales de continuar o no el proyecto. Un hito es un punto de sincronización en el que coinciden una serie de objetivos bien definidos, se completan artefactos y se toma la decisión de pasar o no a la siguiente fase.

Interfaz

Una colección de operaciones que son utilizadas para especificar un servicio de una clase o de un componente.

MARC (Machine Readable Cataloging Records; Registro Catalográfico Legible por Máquina).

Estándar que se utiliza en casi todo el mundo, por lo tanto, amplía las posibilidades de comunicación.

Modelo

Un modelo es una representación abstracta de la realidad. El propósito de un modelo es describir y entender el comportamiento de los fenómenos para que así se pueda predecir su comportamiento.

Modelo de un proceso

Es la descripción de un proceso en términos de un grupo de actividades que son llevadas a cabo por actores utilizando recursos y generando artefactos.

Nivel de integración

Define una etapa en el proceso de integración de un módulo o componente.

Nodo

Es un elemento físico que existe en tiempo de ejecución y que representa un recurso computacional.

Objeto

Entidad con atributos y operaciones perfectamente definidas.

Paquete

Mecanismo de propósito general para organizar elementos en grupos.

Paso

Es una acción atómica, es decir, una acción que ya no puede dividirse en más acciones.

Portabilidad

Grado en el que un sistema, en funcionamiento en un determinado entorno de ejecución, puede ser convertido fácilmente en un sistema funcionando en otro entorno de ejecución.

TESIS CON
FALLA DE ORIGEN

Proceso

Un proceso es un conjunto ordenado de pasos a seguir para alcanzar un objetivo.

Proceso Unificado

Proceso de desarrollo de software que esta dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental.

Recurso

Un recurso es una entidad que es requerida para realizar una acción de un proceso.

Recursos

Son todo tipo de material indispensable para apoyar el desarrollo de un sistema y sus artefactos.

Registro

Conjunto de campos relacionados.

Requerimiento

Condición esencial que tiene que cumplir un sistema.

Rol

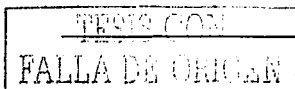
Comportamiento específico de una entidad que participa en un contexto particular.

RUP (Rational Unified Process)

Versión del Proceso unificado de Rational con su propia herramienta.

Solicitante

Persona u Proyecto de investigación, que hace una solicitud de software al Departamento de Informática



Subsistema

Una agrupación de elementos, de los que algunos constituyen una especificación del comportamiento ofrecido por otros elementos contenidos.

Tabla

Disposición de datos en filas y columnas. Las filas corresponden a los registros de datos y las columnas a los campos de datos.

Trabajador

Persona a la que se le asigna un rol para realizar actividades relacionadas con el rol. Para asignar los roles deben considerarse las habilidades de las personas.

UML (Unified Modeling Language).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML proporciona una forma estándar de describir los planos de un sistema, como las cosas concretas, tales como las cosas escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes de software reutilizables.

Usuario

Ver actor.

TESIS CON
FALLA DE ORIGEN

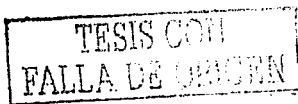
TESIS CON
FALLA DE ORIGEN

BIBLIOGRAFÍA

- Huff, 1996. Huff, K. E. *Software Process Modeling*. Trends in Software, 1996, pp. 1-24.
- Curtis, 1992. Curtis M. Bill, Kellner, I. and Over, J. *Process Modeling*. Communications of the ACM, vol. XXXV, núm. 9, 1992, pp. 75-90.
- Booch, 1999. Booch, G. *El lenguaje unificado de modelado*. Madrid : Addison Wesley Iberoamericana, 1999. 438 p.
- Flores, 1999. Flores Ríos B. L., and Martínez García A. I. *A Computer Aided Process Engineering Tool for the Study of Organizational Process*. Sociedad Mexicana de Ciencia de la Computación, Memorias del 3er encuentro internacional de ciencias de la computación, núm. 01, 2001, pp. 939-948.
- Gil, 1999. Gil Pizarro, Julia, *Diccionario general de informática*. España : Abeto editorial, 1999. 351 p.
- Instituto, 1999. Instituto de Investigaciones Bibliográficas. *Reglamento Interno*. México : Universidad Nacional Autónoma de México, 1999. 26 p.
- Jacobson, 2000. Jacobson, I., Booch G. y Rumbaugh J. *El proceso unificado de desarrollo de software*. Madrid : Pearson Educación, S.A., 2000. 464 p.

TESIS CON
FALLA DE ORIGEN

- Jazayeri, 1996. Jazayeri, M. and Garg, P. *Process-centered software engineering environments: a grand tour*. Trends in Software, 1996, pp. 25-52.
- Kruchten, 2000. Kruchten P. *The rational unified process an introduction second edition*. Boston : Addison Wesley, 2000. 299 p.
- Pfaffenberger, 1999 Pfaffenberger, Bryan. *Diccionario de términos de computación*. México: Prentice may, 1999. 576 p.
- Pressman, 1992. Pressman, S. Roger. *Ingeniería del software*. México : McGraw-Hill, 1992, pp. 824.
- Ramírez, 2003. Ramírez Molina, Ana Yuri. *La implementación y administración del proyecto, cambios y configuración del Proceso Unificado adaptados a un caso práctico*. México : UNAM, 2003, pp. 133.
- Robillard, 2003. Robillard N. Pierre; Kruchten, Philippe y D' Astous, Patrick. *Software Engineering Process with the UPEDU*. Estados Unidos : Addison Wesley, 2003. 346 p.
- Salazar, 2003. Salazar Carmona, José Antonio. *Pruebas, Ambiente y Distribución basados en el Proceso Unificado (Un caso práctico)*. México : UNAM, 2003, pp. 147.



Direcciones en Internet

- Mancilla, 200 http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/marc/09_definicion.htm
- Metamodel, 2002. <http://metamodel.com>
- Rational, 2000. <http://www.rational.com>
- Yoopedoo, 2002. <http://www.yoopeedoo.org>

TESIS CON
FALLA DE ORIGEN

APÉNDICE A

ESTÁNDAR DE DOCUMENTACIÓN

TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas



**Departamento de Informática
Sección de desarrollo de Software**

Estándar de documentación

La documentación será generada de acuerdo a los siguientes criterios y parámetros:

- Uso de Microsoft Word 97 o superior.
- El documento será identificado de la siguiente manera NombreDelArchivo.xxx.
Donde NombreDelArchivo será el nombre del documento, la letra inicial de cada palabra deberá ser mayúscula por lo cual no existirán espacios entre cada palabra y XXX indica la extensión o tipo de archivo. Por ejemplo este archivo debe llamarse EstandarParaElaborarDocumentacion.doc
- Tipo de letra Times New Roman de tamaño 10 para encabezado y pie de página, tamaño 12 para texto normal, de tamaño 12 negrita para subtítulos y de tamaño 14 negrita mayúscula para título principal.
- Dejar 2 espacios entre el subtítulo una nueva línea de texto.
- Dejar un espacio para la separación entre párrafos.
- Interlineado sencillo.
- Tabuladores a 1.25 cm.
- Márgenes superior e inferior a 2.5 cm del borde.
- Márgenes izquierdo y derecho a 3 cm del borde.
- El pie de página ira 1.5 cm del borde.
- El pie de página contendrá el número de página en el centro del documento bajo una línea horizontal.
- Todo el documento deberá estar justificado.
- Todas las palabras mayúsculas deberán acentuarse.

TESIS CON
FALLA DE ORIGEN

- Impresión a tinta negra en hojas tamaño carta blancas.
- Espacio entre párrafos de 1 interlineado.
- El formato para las fechas será el siguiente DD/MM/AAAA donde DD es el día, MM el mes correspondiente y AAAA el año.
- Las viñetas que se emplearan son las circulares rellenas.
- Los números que se utilicen deberán los que van seguidos de un punto, espacio e información.
- La elaboración de diagramas se realizará de la siguiente manera:
El tamaño de fuente será variable para ajustar y distribuir el diagrama a una sola página.
El título de cada diagrama deberá ir centrado.
- La cabecera del documento debe contener la siguiente leyenda:
Nombre del Instituto, Nombre del departamento, Nombre de la sección, título del documento, (todos estos puntos centrados) las siglas del proyecto deberán ir con mayúsculas en la parte izquierda y la fecha en la parte derecha.
- En la parte superior derecha deberá ir el escudo del IIB.
- En el pie del documento, en caso de requerirlo, deberán aparecer las firmas de los responsables; de lado izquierdo nombre y firma del responsable de informática y de lado derecho nombre y firma del solicitante.

TESIS CON
FALLA DE ORIGEN

APÉNDICE B

ESTÁNDAR DE REQUERIMIENTOS

TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas



Departamento de Informática
Sección de desarrollo de Software

Estándar de requerimientos

1. Nombres de actores.

Para designar los nombres de actores que interactúan con los sistemas se ha establecido la siguiente tabla de equivalencias.

En la primera columna se encuentran los diferentes actores que se han definido para operar el sistema y en la segunda su equivalente con los usuarios proporcionados por los solicitantes.

Actores del sistema	Usuarios proporcionados por el solicitante
Administrador	Investigador, Técnico Académico.
Capturista	Bibliotecario, becario, secretaria.
Consultante	Cualquier persona que requiera consultar información relacionada con los sistemas dentro o fuera de la institución.

2. Nombres de casos de uso generales.

Con base en la funcionalidad de los sistemas se han establecido los siguientes casos de uso generales.

Casos de uso generales	Descripción
Entrar	Proporciona la entrada para interactuar con el sistema.
Captura	Proporciona la funcionalidad para ingresar y almacenar información en una base de datos.
Consulta	Proporciona la funcionalidad para recuperar información por diferentes mecanismos.
Impresión	Proporciona la funcionalidad para imprimir fichas bibliográficas o catalográficas, reportes y etiquetas.

TESIS CON
FALLA DE ORIGEN

Administración	Proporcionar la funcionalidad para poder realizar respaldos de información y generación de índices, reportes y etiquetas.
Salir	Permite salir del sistema.

3. Definición de ciclos.

Con base en las necesidades de los solicitantes de los proyectos del Instituto de Investigaciones Bibliográficas éstos, serán desarrollados en tres ciclos, sin embargo puede ocurrir que un proyecto atraviese por solo un ciclo o dos. Los ciclos definidos son los siguientes.

No. de ciclo	Nombre del ciclo
1	Captura
2	Consulta
3	Impresión
4	Administración

4. Detalle de ciclos.

Cada ciclo puede estar integrado por más de un caso de uso general y, cada caso de uso general tiene una serie de casos de uso relacionados.

A continuación se muestra la forma en como está integrado cada ciclo.

No. de ciclo	Nombre del ciclo	Casos de uso involucrados
1	Captura	Entrar, nuevo, guardar, actualizar, consultar y salir.
2	Consulta	Entrar, búsqueda por autor, búsqueda por título, búsqueda por palabra clave y salir.
3	Impresión	Entrar, imprimir tarjetas, imprimir reportes, imprimir etiquetas y salir.
4	Administración	Entrar, respaldos, generación de índices, generación de reportes y salir.

TESIS CON
FALLA DE ORIGEN

APÉNDICE C

ESTÁNDAR DE ANÁLISIS Y DISEÑO

TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software



Estándar de análisis y diseño

La importancia de contar con un modelo fijo y normas de documentación, es el obtener un concepto uniforme del proyecto, permitiendo una mejor comprensión del mismo a los arquitectos de software y posteriormente agilizando las actualizaciones que se realicen al proyecto.

En el estándar se detallan las convenciones necesarias para producir el análisis y diseño de los sistemas del Departamento de Informática.

1. Características del Documento

Las características del documento en general deben seguir los lineamientos definidos en el estándar de documentación del Apéndice "A".

2. Características de los Diagramas

Para el caso de los diagramas el estándar será de la siguiente forma:

TESIS CON
FALLA DE ORIGEN

Caso de Uso: [Nombre del caso de Uso].

Identificador: [Identificador del caso de uso asignado en la fase de requerimientos].

Diagrama de Realización: [Diagrama de clases que intervienen en la resolución del caso de uso].

[Diagrama de Realización]

Diagrama de Secuencia: [Diagrama de interacción que muestra la ordenación temporal de los mensajes entre objetos]

[Diagrama de Secuencia]

TESIS CON
FALLA DE ORIGEN

3. Descripción de la arquitectura

Para el desarrollo de productos de software del IIB se ha seleccionado una arquitectura que abarca una interfaz para el usuario y el almacenamiento persistente de datos, conocida con el nombre de arquitectura de tres capas. Está constituida por las siguientes capas verticales:

- Capa de Interfaz Humana, en esta capa se ubicarán los elementos con los que interactuará el usuario. Por ejemplo: menús, ventanas, botones, etc.
- Capa Lógica de la Aplicación, en esta capa se definen las tareas y reglas que rigen el negocio. Está constituida por los siguientes elementos:
 - Objetos del Dominio del Problema, que son las clases que representan los conceptos del dominio, por ejemplo: horarios, respaldos, número de préstamos, etc.
 - Servicios, que son aquellos objetos de dominio no relacionados con el problema que prestan el servicio de soporte, como la interacción con la base de datos, los reportes, las comunicaciones, etc.

La capa lógica de aplicación es la capa intermedia que interactúa con las otras dos capas.

- Capa del Manejo de Datos, se encuentran los mecanismos de almacenamiento persistente como puede ser la base de datos.

TESIS CON
FALLA DE ORIGEN

4. Normas para nombrar clases

La asignación de nombre de las clases se realizará empleando una letra mayúscula al inicio y en cada inicio de subpalabra dentro de la propia definición.

- Las clases correspondientes al Dominio del Problema (DP) además deberán ser nombradas de acuerdo al siguiente formato:
NombreClaseDP : donde NombreClase es la identificación de la clase y DP es el posfijo que corresponde al dominio de problema.
- Las clases que pertenezcan a la interacción con el usuario o clases límite serán designadas de la manera siguiente:
VentanaNombreClase : donde NombreClase es el identificador de la clase y Ventana es el prefijo correspondiente a las interfaces de interacción.
- Las clases correspondientes al Manejo de Datos (MD) tienen que ser identificadas de acuerdo al criterio siguiente:
NombreClaseMD : donde NombreClase es el nombre de la clase y MD es el posfijo que identifica a las clases que realizan manejo de los datos de catálogos.

5. Normas para nombrar las entidades

Una entidad será denominada como tal siempre y cuando cumpla con las siguientes características:

- Que sea un objeto o elemento del mundo real, y
- Que la información que maneje necesite ser almacenada en algún medio físico

TESIS CON
FALLA DE ORIGEN

6. Normas para nombrar atributos

Los nombres de los atributos iniciarán con una letra mayúscula y se intercalará una letra mayúscula por cada inicio de subpalabra. Por ejemplo:

- Nombre
- ApellidoPaterno
- ApellidoMaterno
- Etc.

Al identificar los atributos se deberá tener presente en que lenguaje se va a implementar el sistema; con base en esto, se pueden tener dos formas de identificar los atributos:

7. Normas para definir las características de los atributos

Las características de los atributos de tipo, rango y tamaño pueden tomar alguno de los siguientes valores:

Tipo	Abreviatura	Rango	Tamaño
Alfabético	a	A...Z a...z	1...N
Numérico	n	1...N N = M N < M N > M N < M > P	Depende del rango
Alfanumérico	an	A...Z a...z 1...N	1...N

La forma de especificarlos será anotando en el costado derecho del atributo las características de la siguiente forma.

TEMA CON
PALA DE ORIGEN

- titulo(c, 150)
- paginas(n, 4)
- clasificacion(an, 30)
- precio (n, <1000>100)

8. Normas para nombrar métodos

Los nombres de los métodos iniciarán con una letra mayúscula y se intercalara una letra mayúscula por cada inicio de subpalabra dentro de la identificación del método. A continuación del nombre del método seguirá el listado de los parámetros correspondientes dentro de paréntesis. Por ejemplo:

- GuardarRegistro()
- ActualizarRegistro()
- ImprimirRegistro()
- Etc.

9. Normas para definir tipos de llaves

La definición de llaves se debe realizar para clases y para entidades; la forma en como se debe diferenciar el tipo de llave es la siguiente:

Llave primaria. Es aquél atributo o conjunto de atributos que identifican de manera única a las entidades, clases o tablas. En caso de que no existiera un atributo o un conjunto de atributos con esta característica, deberá ser creado en ese momento. Para indicar que un atributo o campo es una llave primaria se colocan las letras "CveP" (Clave Primaria) a un costado del atributo designado como tal.

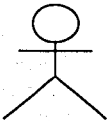




Llave foránea. Es aquél atributo que aparece en dos clases o en dos entidades de la siguiente forma: En una clase o entidad el atributo aparece como llave primaria y en la otra clase o entidad aparece como un atributo más.






Para indicar que un atributo o campo es una llave foránea se colocan las letras "CveF" (Clave Foránea) a un costado del atributo designado como tal.






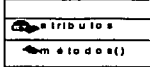


Llaves secundarias. Son aquellos atributos de las clases o relaciones que se definen como llaves secundarias con la finalidad de realizar búsquedas sobre la información que almacenan.

Para indicar que un atributo o campo es una llave secundarias se colocan las letras "CveS" (Clave Secundaria) a un costado del atributo designado como tal.


10. Símbolos permitidos en la elaboración de artefactos.

Símbolo	Descripción
	Es utilizado para representar a los actores. Este símbolo puede utilizarse en diagramas de casos de uso, modelo de casos de uso, diagramas de secuencia, etc.
	Simboliza casos de uso. Este símbolo puede utilizarse en diagramas de casos de uso y modelo de casos de uso
	Representa nota. Se puede usar en cualquier parte con la finalidad de aclarar alguna cuestión.
	Se utiliza para conectar una nota. Se utilizará en la elaboración de los modelos y en cualquier parte que se use el símbolo de nota.
	Se utiliza para representar dependencia. Una dependencia es una relación semántica entre dos elementos, en la cual un cambio a un elemento (elemento independiente) puede afectar a la semántica del otro elemento (elemento dependiente). Gráficamente, una dependencia se representa como una línea discontinua, posiblemente dirigida, que incluye a veces una etiqueta." [Booch, 1999. p. 20.]
	Se utiliza para representar una asociación. "Una asociación es una relación estructural que describe un conjunto de enlances, los cuales son conexiones entre objetos . La

	<p>agregación es un tipo especial de asociación, que representa una relación estructural entre un todo y sus partes. Gráficamente, una asociación se representa como una línea continua, posiblemente dirigida, que a veces incluye una etiqueta." [Booch, 1999. p. 20].</p> <p>Éste símbolo también se usará para unir la relación que exista entre entidades.</p>
	<p>Se utiliza para representar una generalización. "Una generalización es un tipo de especialización / generalización en la cual los objetos del elemento especializado (el hijo) pueden sustituir a los objetos del elemento general padre (el padre). De esta forma, el hijo comparte la estructura y el comportamiento del padre. Gráficamente, una relación de generalización se representa como una línea continua con una punta de flecha vacía apuntando al padre." [Booch, 1999. p. 20].</p>
	<p>Se utiliza para representar una realización. "Una realización es una relación semántica entre clasificadores, en donde un clasificador especifica un contrato que otro clasificador garantiza que cumplirá. Se pueden encontrar relaciones de realización en dos sitios: entre interfaces y las clases y componentes que las realizan, y entre los casos de uso y las colaboraciones que los realizan. Gráficamente, una relación de realización se representa como una mezcla entre una generalización y una relación de dependencia." [Booch, 1999. p. 20].</p>
	<p>Se utiliza para indicar que dos o más actividades se sincronizan. Este símbolo es utilizado al realizar diagramas de actividades y de secuencia.</p>
 <p>Nombre del artefacto de entrada</p>	<p>Se utiliza para representar un artefacto de entrada. Se usará al momento de elaborar diagramas de actividades y de secuencia. El símbolo no aparece en ninguna versión de UML, es un diseño elaborado por las personas que participamos en la adaptación del Proceso Unificado.</p>

 <p>Nombre del artefacto de salida</p>	<p>Se utiliza para representar un artefacto de salida. Se usará al momento de elaborar diagramas de actividades y de secuencia. El símbolo no aparece en ninguna versión de UML, es un diseño elaborado por las personas que participamos en la adaptación del Proceso Unificado.</p>
<p>Entidad o clase</p>	<p>Símbolo para representar a las entidades o las clases que participan en la realización de casos de uso. Se utiliza cuando se elaboran diagramas de entidad relación y de secuencia.</p>
	<p>Representa una decisión. Se usará en donde haya que tomar una decisión. Otra aplicación del símbolo es cuando se realiza el diagrama entidad relación para representar las relaciones entre entidades.</p>
	<p>Representa actividades y pasos. Se usará para construir los flujos y para diagramas de actividades. El símbolo no aparece en ninguna versión de UML, es un diseño elaborado por las personas que participamos en la adaptación del Proceso Unificado.</p>
	<p>Símbolo de monitoreo, se utiliza cuando se realiza cualquier tipo de monitoreo en cualquier fase.</p>
	<p>Representa paquetes y subsistemas. Será usado para construir la arquitectura, generar el modelo de análisis y el modelo de diseño.</p>
	<p>Es utilizado para representar las clases. Se usará para pintar clases y para elaborar los diagramas de clases.</p>
	<p>Representa almacenamiento en una base de datos. Se utilizará para representar la base de datos al momento de construir la arquitectura.</p>
	<p>Símbolo para representar componentes, se utiliza para elaborar el diagrama diagramas de componentes.</p>

TEXIS CON
FALLA DE ORIGEN

	Símbolo de interfaz, se utiliza para representar una interfaz al momento de realizar los diagramas de componentes.
-----------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------

11. Operaciones de clases.

Las operaciones que las clases realizan pueden ser de los siguientes tipos.

Operación	Descripción
Pública	Puede ser accesada por cualquier clase en cualquier momento.
Privada	La operación es exclusiva de la clase y ninguna otra la podrá utilizar.
Protegida	Puede ser accesada solo por algunas clases.

TESIS CON
FALLA DE ORIGEN

APÉNDICE D

ESTÁNDAR DE BASE DE DATOS

TESIS CON
FALLA DE ORIGEN

D-1

Instituto de Investigaciones Bibliográficas



Departamento de Informática
Sección de desarrollo de Software

Estándar de base de datos

Para la creación de una base de datos estándar que contemple los campos de datos necesarios para la definición y creación de las bases de datos de los sistemas del Instituto de Investigaciones Bibliográficas, se ha establecido un esquema con base al formato MARC.

MARC (Machine Readable Cataloging Records; Registro Catalográfico Legible por Máquina). Es un estándar que se utiliza en casi todo el mundo, por lo tanto, amplía las posibilidades de comunicación. MARC maneja los siguientes conceptos.

Registro catalográfico: Significa un registro bibliográfico o de información presentada tradicionalmente en una ficha.

Legible por máquina: Significa un tipo particular de máquina, una computadora, puede leer e interpretar los datos de un registro catalográfico.

Algunos campos pueden estar divididos en subcampos, cada uno tiene un código de subcampo con un delimitador 1, @, etc.

Campo: Cada registro bibliográfico se divide en campos. Campo: autor, título y así sucesivamente. Estos pueden dividirse en dos o más subcampos. Los nombres de los campos son muy largos para ser reproducidos en cada registro MARC, por lo que son representados en una etiqueta de tres dígitos.

Etiqueta: En un campo primero va el tag (020, 100, etc.) luego siguen dos caracteres llamados indicadores (en nuestro caso no los usaremos). Etiqueta es la representación de cada campo por un número de tres dígitos

llamado "tag" La etiqueta identifica el campo y la clase de datos que contiene.

020 es el tag que identifica el campo ISBN

Las etiquetas (tags) son seguidas por los nombres de los campos. Si una etiqueta aparece más de una vez en un registro bibliográfico, es una etiqueta repetible (R).

Si ella puede ser usada una sola vez se llama no repetible (NR).

Indicadores: Algunos campos son, además, definidos por indicadores. En nuestro caso no los vamos a usar. Los indicadores son dos caracteres que siguen a cada tag.

En caso que se usen, están especificados en cada campo.

Subcampo: Muchos campos tienen datos relacionados entre sí. Cada tipo de dato de un campo es llamado subcampo y está precedido de un código de subcampo(^).

Con base en los conceptos anteriores a continuación se presenta una tabla en donde se pueden identificar los campos de datos con su respectivo tipo de información que manejan.

La información que contiene la tabla fue extraída de un manual de MARC y se realizó en base al tipo de información que se requiere para los sistema. Todas las etiquetas contenidas en la tabla son las más generales y son las que se usan para el proceso bibliohemerográfico en el Instituto de Investigaciones Bibliográficas.

TESIS CON
FALLA DE ORIGEN

Estándar de base de datos

440	Mención de serie	0-9	Número de caracteres no alfabéticos	a b n p v	Título Unidad subordinada Número de la parte / sección de la obra Nombre de la parte / sección de la obra Designación secuencial / número del volumen
500	Nota general			a	Nota general
501	Nota de "con"			a	Nota de "con"
502	Nota de tesis			a	Nota de tesis
504	Nota bibliográfica			a	Nota bibliográfica
505	Nota formateada de contenido	0 1 2	Contenido completo Contenido incompleto Contenido parcial	a	Nota formateada de contenido
600	Nombre personal (Tema)	0 1 2 3	Nombre de pila Apellido simple Apellido múltiple Nombre de familia	a d c x y z	Nombre personal Título y otras palabras asociadas con el nombre Fechas asociadas con el nombre Subdivisión general Subdivisión cronológica Subdivisión geográfica
610	Nombre corporativo (Tema)	0 1 2	Nombre invertido Nombre de jurisdicción Nombre en orden directo	a b x y z	Nombre corporativo Unidad subordinada Subdivisión general Subdivisión cronológica Subdivisión geográfica
650	Término tópico (Tema general)	0 1 2	Nivel no especificado Tema principal Tema secundario	x y z	Término o tópico Subdivisión general Subdivisión cronológica Subdivisión geográfica
651	Nombre geográfico	8	Lista de Encabezamientos de Materia de la Biblioteca Nacional	a x y z	Nombre geográfico Subdivisión general Subdivisión cronológica Subdivisión geográfica
700	Nombre personal (Secundario)	0 1 2 3 Segundo 0 1 2	Nombre de pila Apellido simple Apellido múltiple Nombre de familia Asiento alternativo Asiento secundario Asiento analítico	a d c e	Nombre personal Títulos y otras palabras asociadas con el nombre Fechas asociadas con el nombre Relacionador
710	Nombre corporativo (Secundario)	0 1 2	Nombre invertido Nombre de jurisdicción Nombre en orden directo	a b	Nombre corporativo o nombre de la jurisdicción Unidad subordinada
740	Título diferente (Secundario)	0-9 Segundo 0 1 2	Número de caracteres no alfabéticos Asiento alternativo Asiento secundario Asiento analítico	a p	Título Nombre de la parte / Sección de la obra
980	Identificación (Nombre del catálogo)				

TESIS CON
FALLA DE ORIGEN

APÉNDICE E

ARTEFACTOS DEL FLUJO DE REQUERIMIENTOS

TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software



Solicitud de desarrollo de software

Esta área será llenada por el solicitante (Investigador, técnico, proyecto o departamento).

Nombre del solicitante: _____
Teléfonos: _____
Departamento o área: _____
Correo electrónico: _____

Descripción de las características del software que se solicita.

Justificación (si existen límites de tiempo de entrega favor de mencionarlo).

Esta área será llenada por el jefe del Departamento de Informática.

Fecha de recepción de la solicitud: _____
No. de Proyecto: _____
Persona asignada al desarrollo: _____

**TESIS CON
FALLA DE ORIGEN**

Nombre y firma del solicitante

VoBo de la autoridad



Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software

Formato de requerimientos

1. Datos Generales.

Nombre del responsable del proyecto de software para la entrega de avances.

Teléfonos:

Departamento o área:

Correo electrónico:

Nombre del sistema a desarrollar (sugerencia).

2. Tipo de desarrollo (Deberá marcar sólo una de las opciones para indicar las características del sistema que está solicitando). Cuando el formato se llena a mano, marcar con una "X", cuando se llena en la PC. Resaltar las opciones con color rojo.

- Un sistema que únicamente permita consultar la información de una base de datos previamente creada.
- Un sistema que permita ingresar, actualizar y consultar la información al solicitante.
- Un sistema que cubra los dos puntos anteriores.
- Creación de una Base de Datos y su correspondiente sistema de consulta para usuarios externos.
- Creación de una Base de Datos y su correspondiente sistema que permita ingresar, actualizar y consultar la información al solicitante.

Otro, explique:

TESIS CON
FALLA DE ORIGEN

3. Tipo de Datos.

- Bibliográficos formato MARC
- Hemerográficos formato MARC
- Biográficos formato MARC
- Archivo
- Material Visual formato MARC
- Bibliográficos formato propio
- Hemerográficos formato propio
- Biográficos formato propio
- Material Visual formato propio
- La combinación de las anteriores, especifique:

Otro, explique:

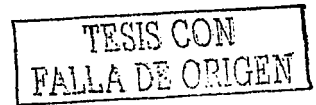
4. Tipo de Información.

- Varios autores.
- Un sólo autor.
- Un período de tiempo.
- Varios períodos de tiempo.
- La combinación de las anteriores, especifique:

Otro, explique:

5. Forma de trabajo deseado.

- Monousuario. (Un solo usuario en una computadora personal).



- **Multiusuario.** (Un grupo de usuarios dentro del IIB en varias computadoras personales).
- **Internet.** (Acceso desde diferentes sitios).

Otro, explique:

6. Usuarios del sistema (Puede marcar más de uno y deberá anotar la información solicitada de cada uno).

Usuario	Nombre	Teléfono	Correo electrónico	Función que desempeñará en el sistema
Investigador				
Técnico Académico				
Bibliotecario				
Becario				
Internos				
Externos				
Otro				

7. Funcionalidades del sistema (Se refiere a las acciones que el sistema deberá realizar; puede marcar más de una y es recomendable agregar ejemplos).

- Que permita importar y exportar datos de otros formatos, especifique:
- Impresión en diferentes formatos del contenido de la base, especifique:
- Búsquedas de registros por distintos campos, especifique:
- Visualización de imágenes, especifique.

TESIS CON
FALLA DE ORIGEN

Otras, explique:

8. Restricciones (En caso de anotar alguna restricción indique el por qué).

- Si existe alguna funcionalidad que tenga mayor prioridad indique cual.
- Manejador de Base de datos específico.
- Sistema operativo específico.
- Lenguaje de Programación específico.
- Otra.

TESIS CON
FALLA DE ORIGEN

9. Descripción de datos (sólo en caso de tener un formato propio de datos). Indique nombre, longitud, tipo de datos y repetibilidad de cada campo. En repetibilidad utilice **S1** para indicar sólo uno y **M1** para indicar más de uno, ejemplo:

Nombre del campo	Tipo de datos	Repetibilidad
Nombre del Autor	250 caracteres	M1
Código postal	5 números	S1
Teléfono	8 números y guiones	M1
Publicaciones	500 caracteres	M1

- **Bibliográfico:**

Nombre del campo	Tipo de datos	Repetibilidad

- Hemerográfico:

Nombre del campo	Tipo de datos	Repetibilidad

- Biográfico:

Nombre del campo	Tipo de datos	Repetibilidad

- Material Visual ¿Cuál?:

Nombre del campo	Tipo de datos	Repetibilidad

- Otro (indique nombre)

Nombre del campo	Tipo de datos	Repetibilidad

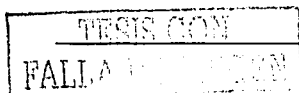
- Otro (indique nombre)

Nombre del campo	Tipo de datos	Repetibilidad

10. Recursos con que cuenta (Anote las características de cada elemento).

Recurso	Características

NOTA IMPORTANTE: FAVOR DE AGREGAR EJEMPLOS DE LA INFORMACIÓN QUE EL SISTEMA VA A MANEJAR.



***Las siguientes secciones serán utilizadas por el Departamento de Informática.
Por favor no anote nada.***

11. Lista de preguntas sobre el llenado de la solicitud de desarrollo de software y el formato de requerimientos.

Pregunta	Sección	Respuesta

Lista final de requerimientos.

12. Nombre y siglas del sistema.

Nombre definitivo del sistema:

Siglas:

Fecha de recepción del formato:



Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software

Diagramas de casos de uso

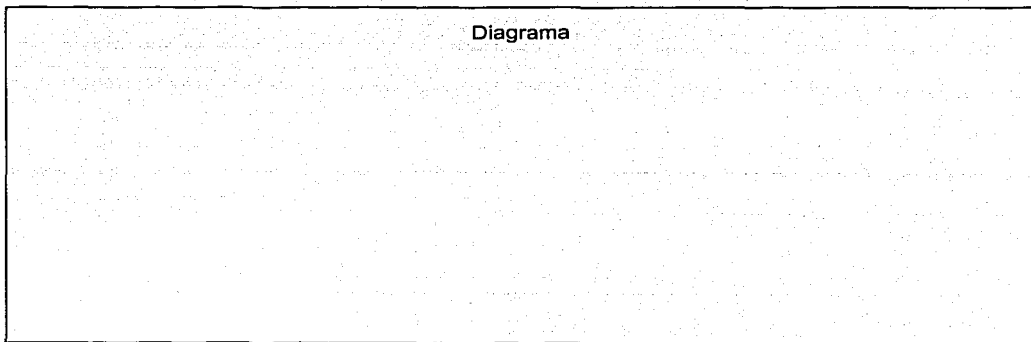
Siglas del proyecto _____ Fecha _____
Responsable _____ Ciclo _____

1. Especificación general de casos de uso.

Nombre del actor: _____

Nombre del caso de uso: _____

Descripción del caso de uso: _____



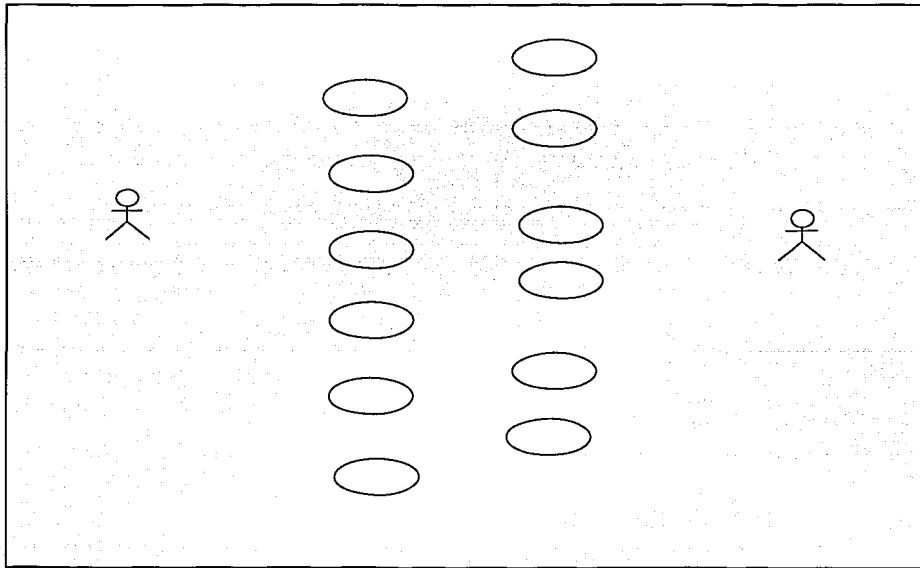
TESIS CON
FALLA DE ORIGEN

2. Detalle de casos de uso generales.

Caso de uso general:

Casos de uso involucrados:

Coloque debajo del símbolo de actor el nombre de éste, y debajo del símbolo de caso de uso los nombres de los casos de uso relacionados al caso de uso general. Con el símbolo de relación deberá marcar las relaciones entre actores y casos de uso.



TESIS CON
FALLA DE ORIGEN



Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software

Formato de prioridad de casos de uso

Siglas del proyecto _____ Fecha _____
Responsable _____ Ciclo _____

Ciclo	Caso de uso general	Casos de uso involucrados	Prioridad

TESIS CON
FALLA DE ORIGEN



Instituto de Investigaciones Bibliográficas

Departamento de Informática

Sección de desarrollo de Software

Modelo de casos de uso

Siglas del proyecto _____

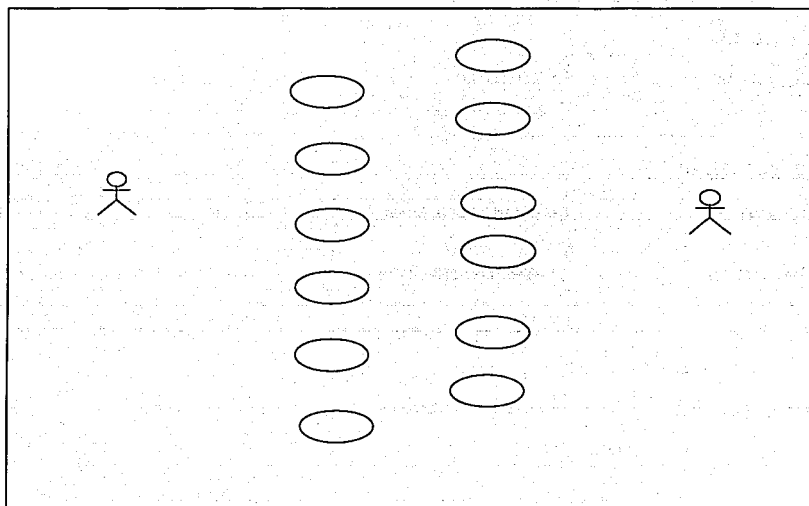
Fecha _____

Responsable _____

Ciclo _____

Actor:

Casos de uso generales:



TESIS CON
FALLA DE ORIGEN



Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software

TESIS CON
FALLA DE ORIGEN

Visión de desarrollo

Siglas del proyecto _____ Fecha _____
Responsable _____ Ciclo _____

Plataforma

PC MAC Internet

Ambiente

Monousuario Multiusuario

Tipo de sistema

Consulta Captura

Otro: _____

Módulos del sistema

Captura Modificaciones Bajas Consultas
 Reportes Administración Índices Impresión

Otro: _____

Información que se maneja

Bibliográfica Hemerográfica Biográfica Imágenes

Otra: Folletos

Productos

CD de instalación CD de recuperación Sitio en Internet

Otro: Manual de usuario

Anexar: Solicitud de desarrollo de software, Formato de requerimientos, Diagramas de casos de uso, Formato de priorización de casos de uso y Modelo de casos de uso.

APÉNDICE F

ARTEFACTOS DEL FLUJO DE ANÁLISIS

TESIS CON
FALLA DE ORIGEN

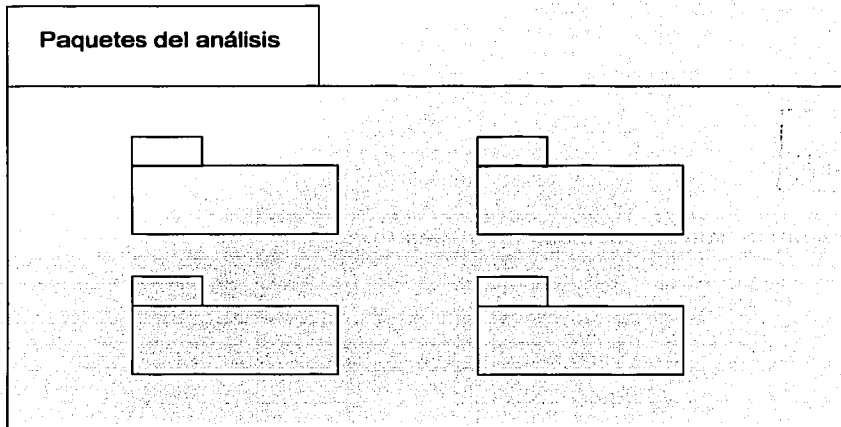
Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software



Arquitectura

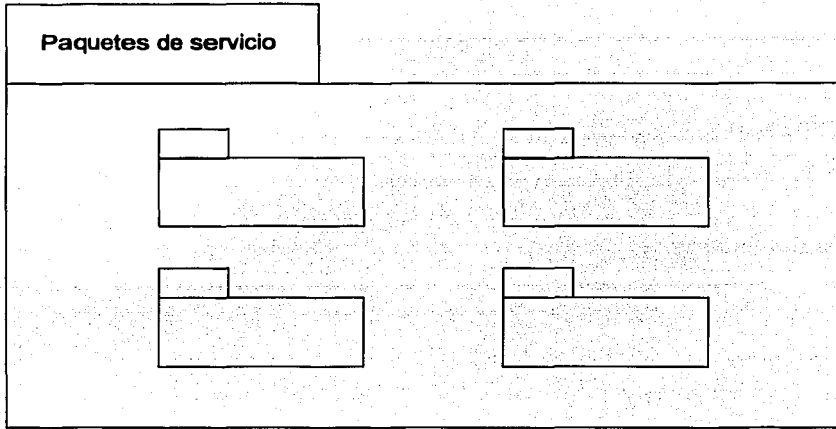
Siglas del proyecto _____ Fecha _____
Responsable _____ Ciclo _____

1. Paquetes del análisis.



TESIS CON
FALLA DE ORIGEN

2. Paquetes de servicio.



TESIS CON
FALLA DE ORIGEN

3. Identificación de requerimientos especiales.

Requerimiento	Descripción

Instituto de Investigaciones Bibliográficas



Departamento de Informática
Sección de desarrollo de Software

Realización de casos de uso del análisis

Siglas del proyecto _____ Fecha _____
Responsable _____ Ciclo _____

1. Descripciones de casos de uso.

Nombre del subcaso de uso:

Flujo:

Actores		Módulo		
Paso	Acción	Paso	Acción	Exp.

Excepciones:

Id	Nombre	Acción

2. Captura de requerimientos especiales.

Requerimiento	Descripción

TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software

TESIS CON
FALLA DE ORIGEN

Clases del análisis

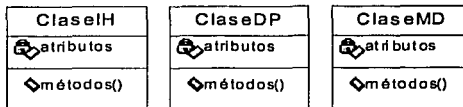
Siglas del proyecto _____

Fecha _____

Responsable _____

Ciclo _____

1. Clases del análisis.



2. Diagrama de clases.

3. Captura de requerimientos especiales.

Requerimiento	Descripción



Instituto de Investigaciones Bibliográficas

Departamento de Informática

Sección de desarrollo de Software

Paquetes del análisis

Siglas del proyecto _____ Fecha _____
Responsable _____ Ciclo _____

Paquetes del análisis	
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>

**TESIS CON
FALLA DE ORIGEN**



Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software

TESIS CON
FALLA DE ORIGEN

Modelo de análisis

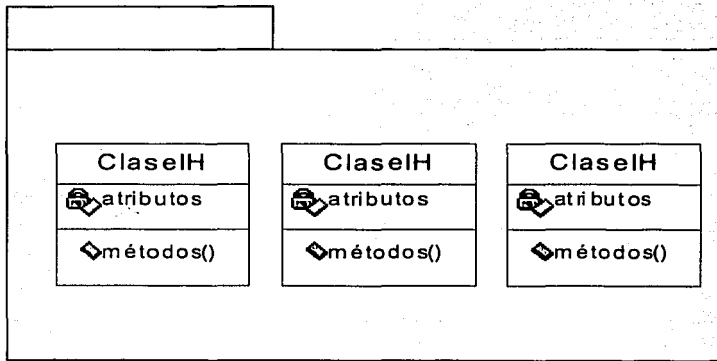
Siglas del proyecto _____

Fecha _____

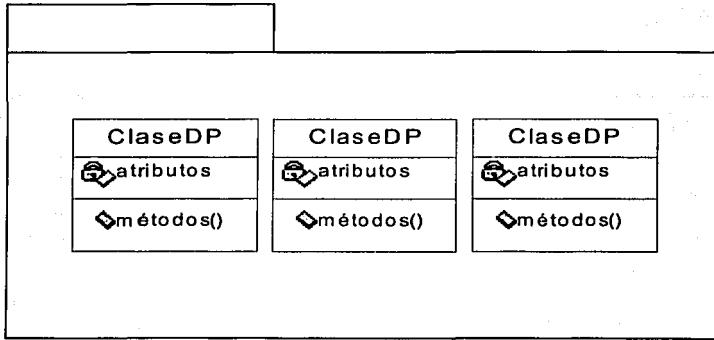
Responsable _____

Ciclo _____

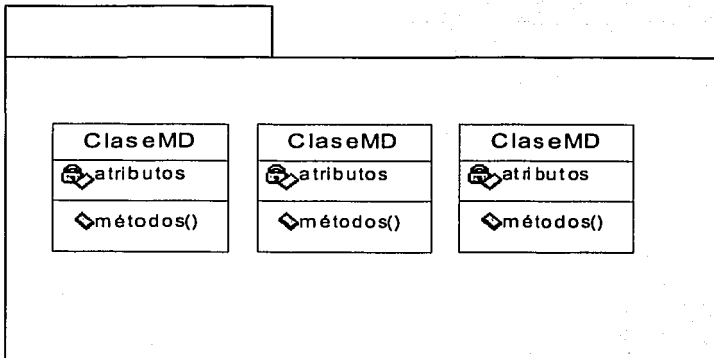
1. Clases de interfaz humana.



2. Clases de dominio del problema.

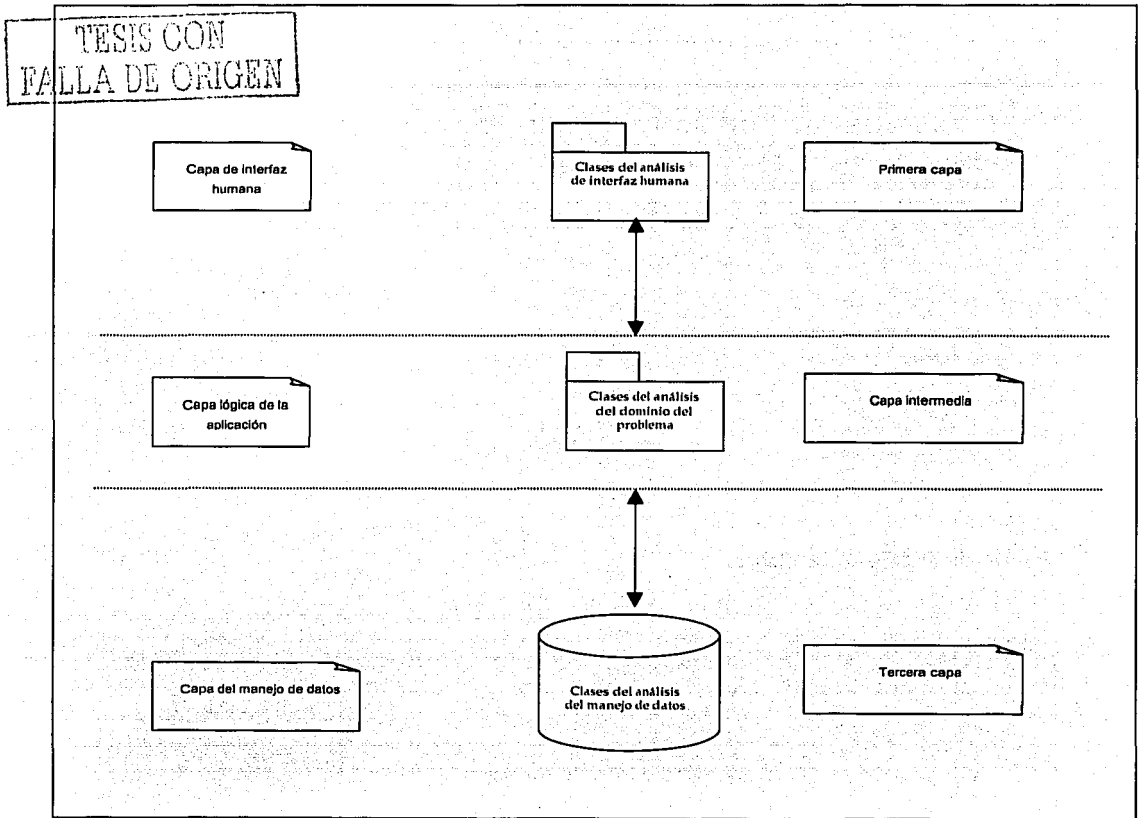


3. Clases de manejo de datos.



TESIS CON
FALLA DE ORIGEN

4. Modelo de análisis.



Anexar: Arquitectura, Realización de casos de uso, Clases y Paquetes del análisis.

APÉNDICE G

ARTEFACTOS DEL FLUJO DE DISEÑO

TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas



Departamento de Informática

Sección de desarrollo de Software

Diseño de la arquitectura

Siglas del proyecto _____

Fecha _____

Responsable _____

Ciclo _____

1. Identificación de nodos y configuraciones de red.

Si corre en ambiente monousuario marque el sistema operativo requerido.

Win9x

Win2000

WinXP

WinML

Si corre en ambiente multiusuario marque servidor y cliente requerido.

WinNT

Mandrake

Unix

Netware

WinXX

Web

Unix

Netware

2. Subsistemas de diseño.

Subsistemas de diseño

TESIS CON
FALLA DE ORIGEN

3. Software de desarrollo.

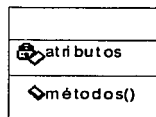
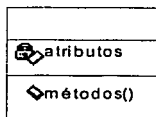
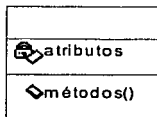
Software	Nombre

TESIS CON
FALLA DE ORIGEN

4. Componentes reusables.

Componente reusable	Equivalencia con la clase de análisis

5. Clases de diseño principales.



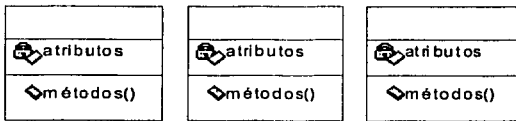
Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software



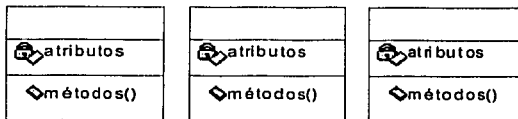
Realización de casos de uso del diseño

Siglas del proyecto _____ Fecha _____
Responsable _____ Ciclo _____

1. Clases del diseño por caso de uso.



2. Clases del diseño por subsistema.



TESIS CON
FALLA DE ORIGEN

3. Diagramas de secuencia por caso de uso.

Caso de uso:	Diagrama
Prioridad:	

TESIS CON FALLA DE ORIGEN

4. Captura de requerimientos especiales.

Requerimiento	Descripción

Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software



Clases del diseño

Siglas del proyecto _____ Fecha _____
 Responsable _____ Ciclo _____

1. Operaciones de la clase.

Operación:

Crear

<input type="checkbox"/>	Pública
<input type="checkbox"/>	Pública
<input type="checkbox"/>	Pública
<input type="checkbox"/>	Pública
<input type="checkbox"/>	Pública

<input type="checkbox"/>	Privada
<input type="checkbox"/>	Privada
<input type="checkbox"/>	Privada
<input type="checkbox"/>	Privada
<input type="checkbox"/>	Privada

<input type="checkbox"/>	Protegida
<input type="checkbox"/>	Protegida
<input type="checkbox"/>	Protegida
<input type="checkbox"/>	Protegida
<input type="checkbox"/>	Protegida

Guardar

Recuperar

Actualizar

Eliminar

Por participación en la realización de casos de uso.

Caso de uso _____ Dependencia Asociación Generalización

Requerimientos especiales:

TESIS CON
FALLA DE ORIGEN

2. Descripción de métodos.

Método	Datos que Recibe		Datos que regresa		Descripción
	Nombre	Tipo	Tipo	Nombre	

Instituto de Investigaciones Bibliográficas



Departamento de Informática

Sección de desarrollo de Software

Diagrama entidad relación

Siglas del proyecto _____

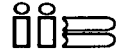
Fecha _____

Responsable _____

Ciclo _____

TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas



Departamento de Informática
Sección de desarrollo de Software

Subsistemas de diseño

Siglas del proyecto

Fecha

Responsable

Ciclo

Clases del subsistema captura

TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas



Departamento de Informática

Sección de desarrollo de Software

TESIS CON FALLA DE ORIGEN

Formato de diseño de base de datos

Siglas del proyecto _____ Fecha _____
 Responsable _____ Ciclo _____

Nombre de la base de datos:

Nombre de la tabla	Descripción
autores	Tabla que almacena toda la información con respecto a los autores.

Campos que la integran	Descripción	Tipo	Tamaño	Valor predeterminado

Identificación de llaves			Viene de tabla
Primaria	Foránea	Secundaria	

Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software



Modelo de diseño

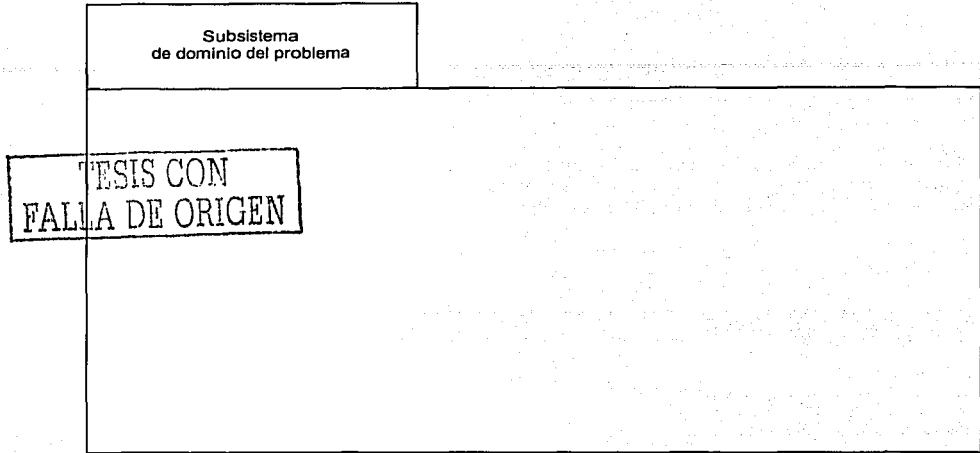
Siglas del proyecto _____ Fecha _____
Responsable _____ Ciclo _____

1. Clases de interfaz humana.

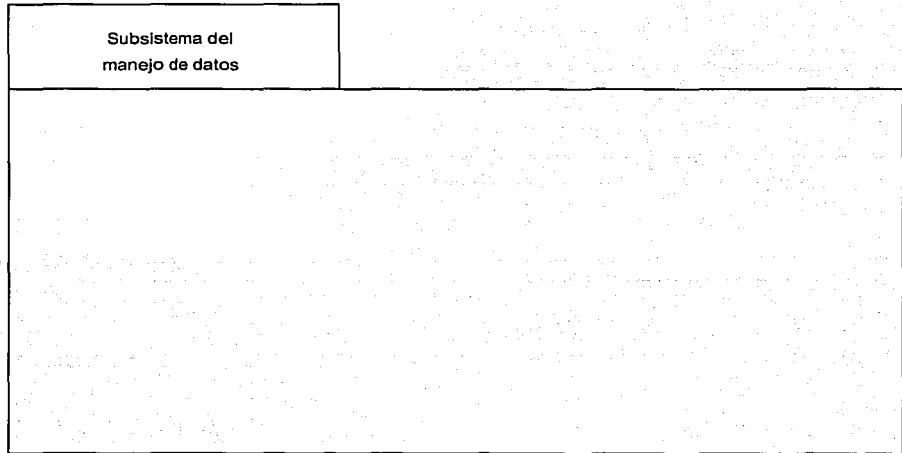
Subsistema de interfaz humana

TESIS CON
FALLA DE ORIGEN

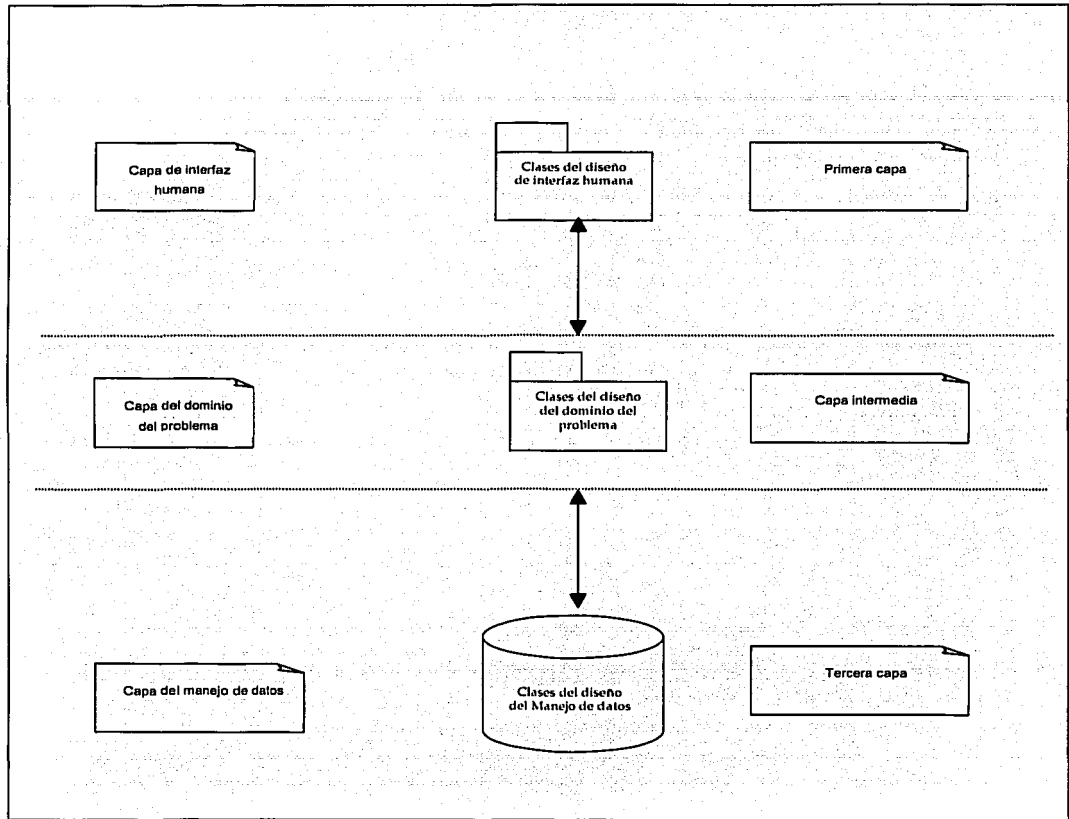
2. Clases de dominio del problema.



3. Clases de manejo de datos.



4. Modelo de diseño.



TESIS CON
VALOR DE ORIGEN

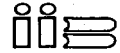
Anexar: Diseño de la arquitectura, Realización de casos de uso de diseño, Clases de diseño, Diagrama entidad relación, Subsistemas del diseño y Formato de diseño de base de datos.

APÉNDICE H

EJEMPLO DE LA APLICACIÓN DEL PROCESO

TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas



Departamento de Informática
Sección de desarrollo de Software

Solicitud de desarrollo de software

Esta área será llenada por el solicitante (Investigador, técnico, proyecto o departamento).

Nombre del solicitante: Dr. Tarsicio García Díaz
Teléfonos: 56 22 68 16
Departamento o área: Investigación
Correo electrónico: tarsicio@biblional.bibliog.unam.mx

Descripción de las características del software que se solicita.

Base de datos bibliográfica con posibilidades de actualización constante, con capacidad para mas de 10 000 mil registros. Son indispensables varias salidas: 1) Pantalla de captura, 2) Pantalla para consulta, 3) Impresión, 4) web. Los campos de captura propuestos son: número consecutivo único por obra, autor, título de la obra, edición, pie de imprenta, descripción física, contenido, notas, temas, otros autores, datos del autor, juicio crítico, localización en otros repositorios y responsable de la captura.

Justificación (si existen límites de tiempo de entrega favor de mencionarlo).

Dentro del Proyecto del Seminario Nacional que se tiene con PAPIIT-DGAPA (IN405201) esta concertada la elaboración de una base de datos bibliográfica que tiene el objetivo de recuperar la bibliografía sobre el tema de la Independencia Mexicana en todos los repositorios bibliográficos del país.

Esta área será llenada por el jefe del Departamento de Informática.

Fecha de recepción de la solicitud: 14 de junio de 2002
No. de Proyecto: 001
Persona asignada al desarrollo: José Antonio Salazar Carmona

YESIS CON
FALLA DE ORIGEN

Nombre y firma del solicitante

VoBo de la autoridad

Instituto de Investigaciones Bibliográficas



**Departamento de Informática
Sección de desarrollo de Software**

**TESIS CON
FALLA DE ORIGEN**

Formato de requerimientos

1. Datos Generales.

Nombre del responsable del proyecto de software para la entrega de avances.

Faviola Monroy Valverde

Teléfonos:

56 22 68 16

Departamento o área:

Correo electrónico:

Nombre del sistema a desarrollar (sugerencia).

Bibliografía General de la Independencia de México (1808-2003) del Seminario de Independencia Nacional.

2. Tipo de desarrollo (Deberá marcar sólo una de las opciones para indicar las características del sistema que está solicitando). Cuando el formato se llena a mano, marcar con una "X", cuando se llena en la PC. Resaltar las opciones con color rojo.

- Un sistema que únicamente permita consultar la información de una base de datos previamente creada.
- Un sistema que permita ingresar, actualizar y consultar la información al solicitante.
- Un sistema que cubra los dos puntos anteriores.
- Creación de una Base de Datos y su correspondiente sistema de consulta para usuarios externos.
- Creación de una Base de Datos y su correspondiente sistema que permita ingresar, actualizar y consultar la información al solicitante.

Otro, explique:

3. Tipo de Datos.

- Bibliográficos formato MARC
- Hemerográficos formato MARC
- Biográficos formato MARC
- Archivo
- Material Visual formato MARC
- Bibliográficos formato propio
- Hemerográficos formato propio
- Biográficos formato propio
- Material Visual formato propio
- La combinación de las anteriores, especifique:

La combinación de bibliográficos con base en el formato MARC y bibliográficos con formato propio.

Otro, explique:

4. Tipo de Información.

- Varios autores.
- Un sólo autor.
- Un período de tiempo.
- Varios períodos de tiempo.
- La combinación de las anteriores, especifique:
Combinación de autor, título, tiempo, juicio crítico.

Otro, explique:

TESIS CON
FALLA DE ORIGEN

5. Forma de trabajo deseado.

- Monousuario. (Un solo usuario en una computadora personal).
- Multiusuario. (Un grupo de usuarios dentro del IIB en varias computadoras personales).
- Internet. (Acceso desde diferentes sitios).

Otro, explique:

6. Usuarios del sistema (Puede marcar más de uno y deberá anotar la información solicitada de cada uno).

Usuario	Nombre	Teléfono	Correo electrónico	Función que desempeñará en el sistema
Investigador	X			
Técnico Académico				
Bibliotecario	X			
Becario	X			
Internos	X			
Externos	X			
Otro				

7. Funcionalidades del sistema (Se refiere a las acciones que el sistema deberá realizar; puede marcar más de una y es recomendable agregar ejemplos).

- Que permita importar y exportar datos de otros formatos, especifique:
- Impresión en diferentes formatos del contenido de la base, especifique:
De acuerdo a los resultados de la búsqueda e impresión general de todos los campos y registros.
- Búsquedas de registros por distintos campos, especifique:

RECIBO CON
FALLA DE ORIGEN

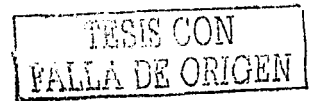
Que se pueda buscar por palabra clave en los distintos campos. Por ejemplo: buscar palabra Hidalgo en campos de título, contenido, notas, temas, juicio crítico, etc.

- Visualización de imágenes, especifique.

Otras, explique:

8. Restricciones (En caso de anotar alguna restricción indique el por qué).

- Si existe alguna funcionalidad que tenga mayor prioridad indique cual.
Inicio de captura 15 de julio.
- Manejador de Base de datos específico.
- Sistema operativo específico.
- Lenguaje de Programación específico.
- Otra.



9. Descripción de datos (sólo en caso de tener un formato propio de datos).

Indique nombre, longitud, tipo de datos y repetibilidad de cada campo. En repetibilidad utilice **S1** para indicar sólo uno y **M1** para indicar más de uno, ejemplo:

Nombre del campo	Tipo de datos	Repetibilidad
Nombre del Autor	250 caracteres	M1
Código postal	5 números	S1
Teléfono	8 números y guiones	M1
Publicaciones	500 caracteres	M1

• Bibliográfico:

Nombre del campo	Tipo de datos	Repetibilidad
Num. Consecutivo	5 números	S1
Autor	150 caracteres	M1
Título	800 caracteres	S1
Edición	10 caracteres	S1
Pie de imprenta	110 caracteres	S1
Descripción física	40 caracteres	S1
Notas	130 caracteres	S1
Temas	100 caracteres	S1
Otros autores	150 caracteres	M1
Responsable	5 caracteres	M1
Localización	50 caracteres	M1
Datos del autor	1500 caracteres	S1
Resumen	1500 caracteres	S1
Juicio crítico	1500 caracteres	S1

• Hemerográfico:

Nombre del campo	Tipo de datos	Repetibilidad

• Biográfico:

Nombre del campo	Tipo de datos	Repetibilidad

• Material Visual ¿Cuál?:

Nombre del campo	Tipo de datos	Repetibilidad

• Otro (indique nombre)

Nombre del campo	Tipo de datos	Repetibilidad

TESIS CON
FALLA DE ORIGEN

- Otro (indique nombre)

Nombre del campo	Tipo de datos	Repetibilidad

10. Recursos con que cuenta (Anote las características de cada elemento).

Recurso	Características
Computadora	1PC. Pentium IV
Escáner	1 HP
Cámara digital	
Quemador	1 HP
Otro	

Nota importante: favor de agregar ejemplos de la información que el sistema va a manejar.

TESIS CON
FALLA DE ORIGEN

Las siguientes secciones serán utilizadas por el Departamento de Informática.

Por favor no anote nada.

11. Lista de preguntas sobre el llenado de la solicitud de desarrollo de software y el formato de requerimientos.

Pregunta	Sección	Respuesta
¿Cómo debe ser la impresión de acuerdo al resultado de la búsqueda?	7	En diferentes formatos como: ficha bibliográfica, formato MARC, etc.
¿Existen más campos para hacer la búsqueda por palabras?	7	Por el momento sólo esas.
¿Qué significa diferencia entre salidas de impresión?	7	Diferentes tipos de reportes
¿Cuáles son los datos del autor?	9	Datos generales
¿Por qué en el campo de autores es más de uno?	9	Porque se debe poder ingresar más de un autor para una bibliografía.
¿Puede haber más responsables en un registro?	9	No
¿Qué es juicio crítico?	9	Crítica del libro
¿Qué significa localización en otros repositorios?	9	Localización en otras fuentes
¿Qué van a capturar en responsable?	9	Iniciales de una persona.
¿Qué van a poner en localización?	9	Clasificación
¿Por qué no entrego ejemplos?	10	Se me olvido.

TESIS CON
FALLA DE ORIGEN

Lista final de requerimientos.

El seminario de Independencia Nacional solicita un proyecto conformado por tres sistemas (Administración de información, consulta en Web y consulta en disco compacto).
Desean que la captura inicie el 15 de julio.
Desean que en octubre se puedan realizar impresiones.
En octubre pretender abrir la consulta en la RED para usuarios internos.
La característica del proyecto es el manejo de información bibliográfica, Hemerográfica y folletos de la Independencia de México.
El período que debe cubrir es de 1808 hasta el 2003.
El sistema deberá desarrollarse en red.
Se desea un CD de recuperación de información, sin la necesidad de que se instale.
Cuentan con un equipo Pentium IV a 1.6 MHZ. 40 GB en disco duro, con sistema operativo Windows XP.
Requieren un módulo de altas, bajas y modificaciones.
Desean capturar más de un autor.
El volumen de información a manejar asciende a más de 10 000 registros.
Las salidas de información serán a pantalla, consultas a través del WEB e impresiones.
Se capturará todo el material documental que contenga información de la Independencia de México.

12. Nombre y siglas del sistema.

Nombre definitivo del sistema: Sistema para las publicaciones sobre la Independencia de México:1808-2003.

Siglas: PSIMex.

Fecha de recepción del formato: 14 de junio de 2002.

TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas



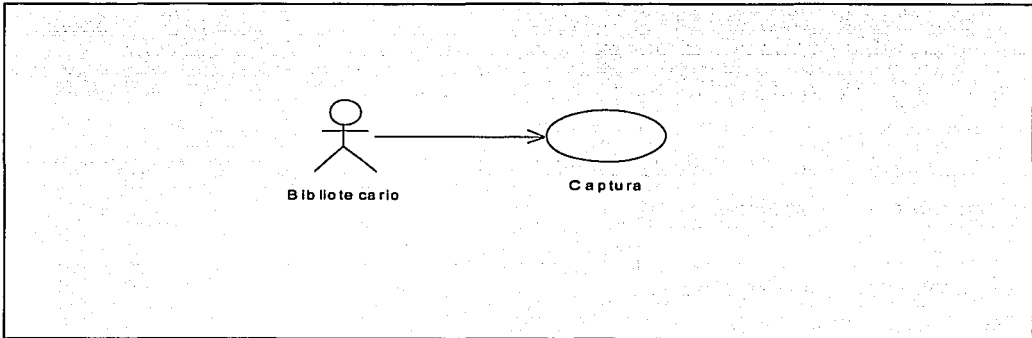
Departamento de Informática
Sección de desarrollo de Software

Diagramas de casos de uso

Siglas del proyecto	<u>PSIMex</u>	Fecha	<u>11/05/2003</u>
Responsable	<u>JASC, AYRM y MGSM</u>	Ciclo	<u>1</u>

1. Especificación general de casos de uso.

Nombre del actor: Bibliotecario
Nombre del caso de uso: Captura
Descripción del caso de uso: El bibliotecario realizará la captura de información.



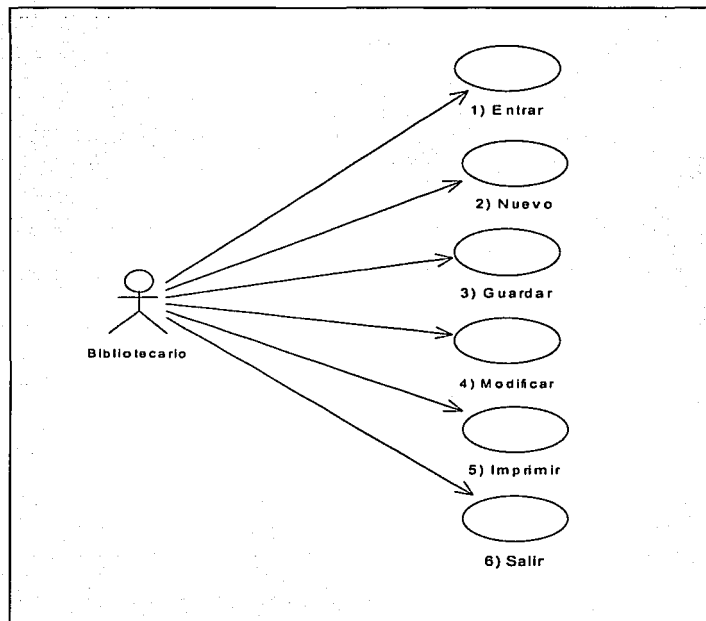
TESIS CON
FALLA DE ORIGEN

2. Detalle de casos de uso generales.

Caso de uso general: Captura

Casos de uso involucrados: Entrar, nuevo, guardar, modificar, imprimir, salir.

Coloque debajo del símbolo de actor el nombre de éste, y debajo del símbolo de caso de uso los nombres de los casos de uso relacionados al caso de uso general. Con el símbolo de relación deberá marcar las relaciones entre actores y casos de uso.



TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas

Departamento de Informática

Sección de desarrollo de Software

Formato de prioridad de casos de uso

Siglas del proyecto PSIMex Fecha 11/05/2003
 Responsable JASC, AYRM y MGSM Ciclo 1

Ciclo	Caso de uso general	Casos de uso involucrados	Prioridad
1	Captura	Entrar	1
		Nuevo	2
		Guardar	3
		Modificar	5
		Imprimir	6
		Salir	4

TESIS CON
FALLA DE ORIGEN

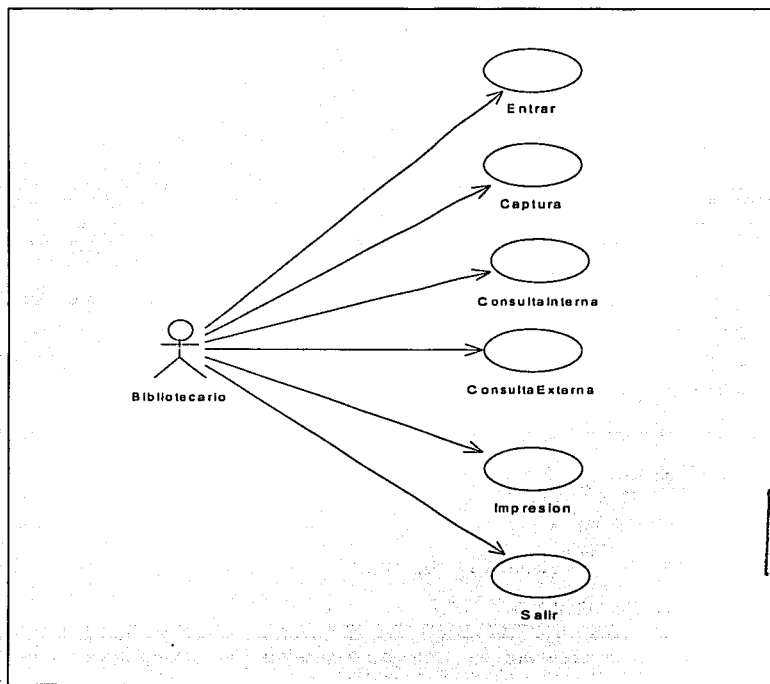
Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software



Modelo de casos de uso

Siglas del proyecto	PSIMex	Fecha	11/05/2003
Responsable	JASC, AYRM y MGSM	Ciclo	1

Actor: Bibliotecario
Casos de uso generales: Entrar, Captura, ConsultaInterna, ConsultaExterna, Impresión y Salir.



TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas



Departamento de Informática

Sección de desarrollo de Software

TESIS CON
FALLA DE ORIGEN

Visión de desarrollo

Siglas del proyecto

PSIMex

Fecha 11/05/2003

Responsable

JASC, AYRM y MGSM

Ciclo 1

Plataforma

PC

MAC

Internet

Ambiente

Monousuario

Multiusuario

Tipo de sistema

Consulta

Captura

Otro:

Módulos del sistema

Captura

Modificaciones

Bajas

Consultas

Reportes

Administración

Índices

Impresión

Otro:

Información que se maneja

Bibliográfica

Hemerográfica

Biográfica

Imágenes

Otra:

Folletos

Productos

CD de instalación

CD de recuperación

Sitio en Internet

Otro:

Manual de usuario

Anexar: Solicitud de desarrollo de software, Formato de requerimientos, Diagramas de casos de uso, Formato de priorización de casos de uso y Modelo de casos de uso.



Instituto de Investigaciones Bibliográficas

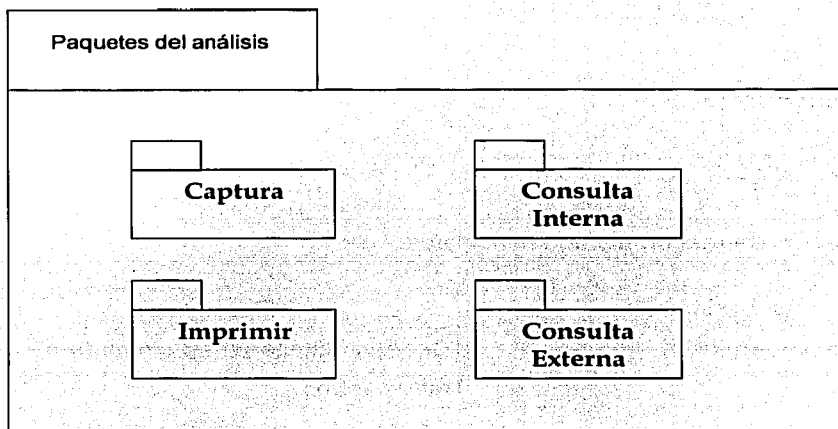
Departamento de Informática

Sección de desarrollo de Software

Arquitectura

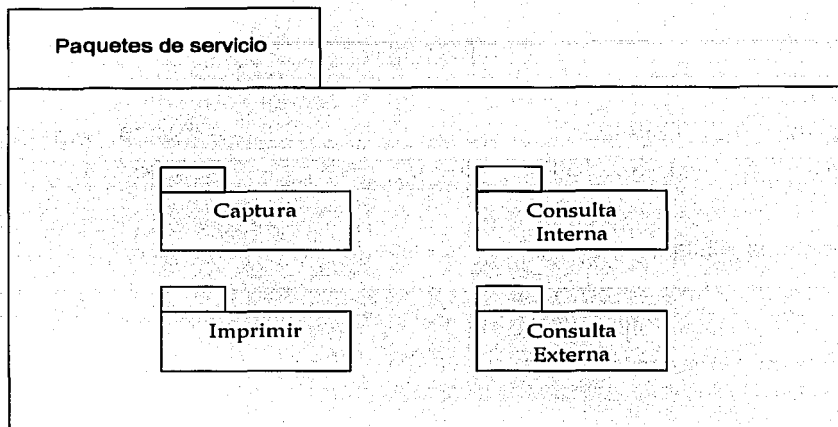
Siglas del proyecto	PSIMex	Fecha	12/05/2003
Responsable	JASC, AYRM y MGSM	Ciclo	1

1. Paquetes del análisis.



TESIS CON
FALLA DE ORIGEN

2. Paquetes de servicio.



TESIS CON
FALLA DE ORIGEN

3. Identificación de requerimientos especiales.

Requerimiento	Descripción
Volumen de información.	Desean almacenar aproximadamente 10 mil registros.
Tamaño de campo título.	Deberá ser aproximadamente de 900 caracteres.
Período a cubrir.	La información que se capture será del período 1803 a 2003.

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software

Realización de casos de uso del análisis

Siglas del proyecto PSIMex Fecha 13/05/2003
Responsable JASC, AYRM y MGSM Ciclo 1

1. Descripciones de casos de uso.Nombre del subcaso de uso: **Nuevo**

Flujo:

Actores		Módulo		
Paso	Acción	Paso	Acción	Exp.
1	Pulsa botón nuevo	2	Limpia campos en pantalla	
3	Ingresa datos a guardar	4	Valida datos	E1
		5	Guarda nuevo registro	

Excepciones:

Id	Nombre	Acción
E1	Datos inválidos	Vuelve a pedir datos

2. Captura de requerimientos especiales.

Requerimiento	Descripción
Validar campo de título.	Debe permitir hasta 900 caracteres.
Validar campo de fecha.	Debe permitir fechas entre 1803 y 2003.

TESIS CON
FALLA DE ORIGEN

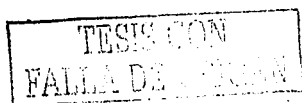
Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software



Clases del análisis

Siglas del proyecto	PSIMex	Fecha	13/05/2003
Responsable	JASC, AYRM y MGSM	Ciclo	1

1. Clases del análisis.



EntrarIH	SalirIH	MensajeIH	CapturistaIH	CapturistaDP	CapturistaMD
PalabrasClaveIH	PalabrasClaveDP	PalabrasClaveMD	HemerografiaIH		
HemerografiaDP	HemerografiaMD	BibliografiaIH	BibliografiaDP		
BibliografiaMD	FolletoIH	FolletoDP	FolletoMD	AutorIH	
AutorDP	AutorMD				

Ejemplo de la aplicación del proceso

BibliografíaH
CveBibliografía
CveAutor
CvePalClave
Título
Lugar
Editorial
Año
Serie
Páginas
Clasificación
JuicioCrítico
ISBN
Mostrar()
Ocultar()

HemerografíaH
CveHemerografía
CvePalClave
Fuente
Artículo
Responsable
Periodo
Año
Volumen
Numero
Fecha
JuicioCrítico
ISSN
Mostrar()
Ocultar()

FolletoH
CveFolleto
CveAutor
CvePalClave
Título
Páginas
JuicioCrítico
Numero
Año
Mostrar()
Ocultar()

AutorH
CveAutor
Nombre
Mostrar()
Ocultar()

PalabrasClaveH
CvePalabraClave
PalabraClave
Mostrar()
Ocultar()

CapturistaH
CveCapturista
Nombre
Mostrar()
Ocultar()

MensajeH
Mensaje
Opcion
Mostrar()
Ocultar()

SalirH
Salir()

EntrarH
Nombre
Clave
Entrar()
VerificarClave()
Salir()

BibliografíaDP
CveBibliografía
CveAutor
CvePalClave
Título
Lugar
Editorial
Año
Serie
Páginas
Clasificación
ISBN
JuicioCrítico
SetCveBibliografía()
getCveBibliografía()
SetCveAutor()
GetCveAutor()
SetCvePalClave()
GetCvePalClave()
SetTitulo()
getTitulo()
SetLugar()
GetLugar()
SetEditorial()
GetEditorial()
SetAño()
GetAño()
SetSerie()
GetSerie()
SetPaginas()
GetPaginas()
SetClasificación()
GetClasificación()
SetISBN()
GetISBN()
SetJuicioCrítico()
GetJuicioCrítico()

HemerografíaDP
CveHemerografía
CvePalClave
Fuente
Artículo
Responsable
Periodo
Año
Volumen
Numero
Fecha
Páginas
ISSN
JuicioCrítico
SetCveHemerografía()
GetCveHemerografía()
SetFuente()
GetFuente()
SetArtículo()
GetArtículo()
SetResponsable()
GetResponsable()
SetPeriodo()
GetPeriodo()
SetAño()
GetAño()
SetVolumen()
GetVolumen()
SetNumero()
GetNumero()
SetFecha()
GetFecha()
SetPaginas()
GetPaginas()
SetISSN()
GetISSN()
SetJuicioCrítico()
GetJuicioCrítico()

FolletoDP
CveFolleto
CveAutor
CvePalClave
Título
Páginas
Numero
Fecha
JuicioCrítico
SetCveFolleto()
GetCveFolleto()
SetAutor()
GetAutor()
SetCvePalClave()
GetCvePalClave()
SetTitulo()
GetTitulo()
SetPaginas()
GetPaginas()
SetNumero()
GetNumero()
SetFecha()
GetFecha()
SetJuicioCrítico()
GetJuicioCrítico()

AutorDP
CveAutor
Nombre
SetCveAutor()
GetCveAutor()
SetNombre()
GetNombre()

PalabrasClaveDP
CvePalabraClave
PalabraClave
SetCvePalabraClave()
GetCvePalabraClave()
SetPalabraClave()
GetPalabraClave()

CapturistaDP
CveCapturista
Nombre
SetCveCapturista()
GetCveCapturista()
SetNombre()
GetNombre()

BibliografíaMD
CveBibliografía
CveAutor
CvePalClave
Título
Lugar
Editorial
Año
Serie
Páginas
Clasificación
ISBN
JuicioCrítico
Crear()
Guardar()
Recuperar()
Actualizar()
Eliminar()

HemerografíaMD
CveHemerografía
Fuente
Artículo
Responsable
Periodo
Año
Volumen
Numero
Fecha
Páginas
ISSN
JuicioCrítico
Crear()
Guardar()
Recuperar()
Actualizar()
Eliminar()

FolletoMD
CveFolleto
CveAutor
CvePalClave
Título
Páginas
Numero
Fecha
JuicioCrítico
Crear()
Guardar()
Recuperar()
Actualizar()
Eliminar()

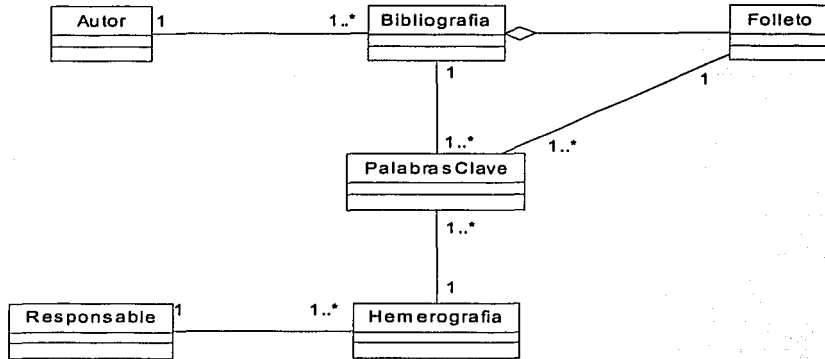
AutorMD
CveAutor
Nombre
Crear()
Guardar()
Recuperar()
Actualizar()
Eliminar()

PalabrasClaveMD
CvePalabraClave
PalabraClave
Crear()
Guardar()
Recuperar()
Actualizar()
Eliminar()

CapturistaMD
CveCapturista
Nombre
Crear()
Guardar()
Recuperar()
Actualizar()
Eliminar()

TESIS CON
FALLA DE ORIGEN

2. Diagrama de clases.

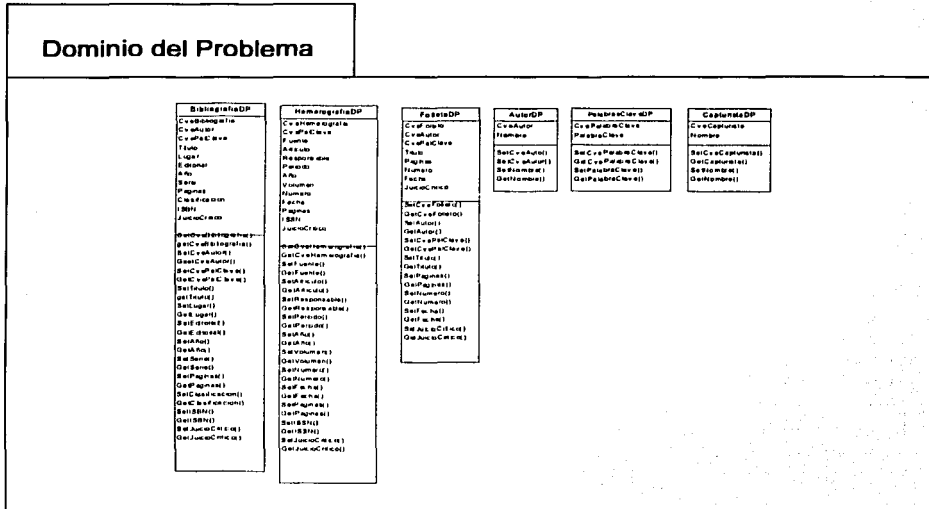


TESIS CON
FALLA DE ORIGEN

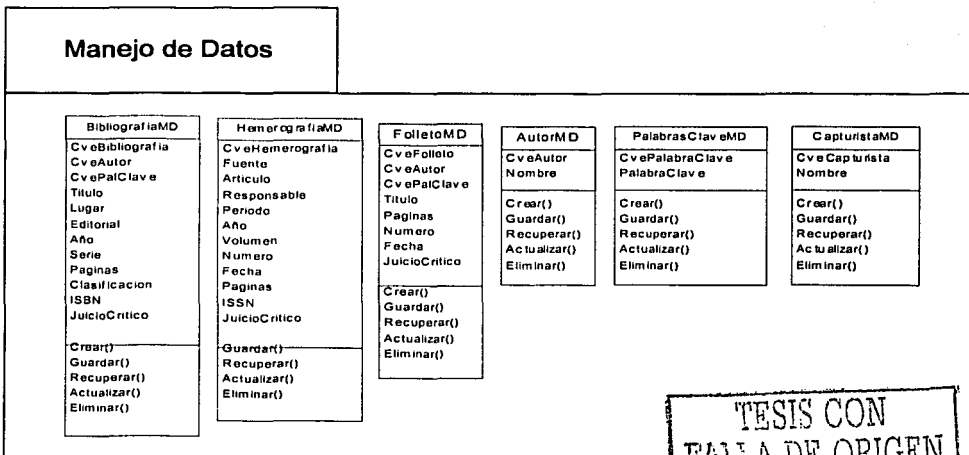
3. Captura de requerimientos especiales.

Requerimiento	Descripción
Palabras clave	El máximo de palabras clave a capturar por cada bibliografía, hemerografía o folleto no deberá exceder de 4.

2. Clases de dominio del problema.

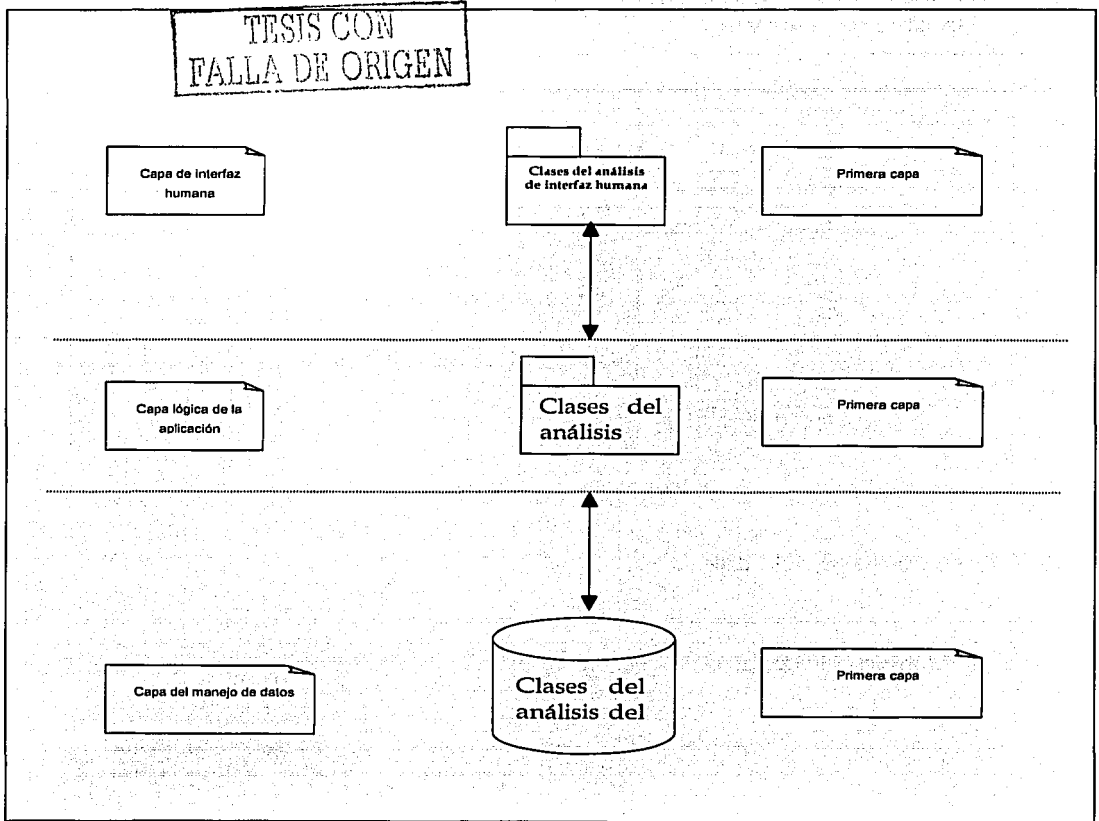


3. Clases de manejo de datos.



TESIS CON
 FECHA DE ORIGEN

4. Modelo de análisis.



Anexar: Arquitectura, Realización de casos de uso, Clases y Paquetes del análisis.

Instituto de Investigaciones Bibliográficas



Departamento de Informática
Sección de desarrollo de Software

Diseño de la arquitectura

Siglas del proyecto	PSIMex	Fecha	15/05/2003
Responsable	JASC, AYRM y MGSM	Ciclo	1

1. Identificación de nodos y configuraciones de red.

Si corre en ambiente monousuario marque el sistema operativo requerido.

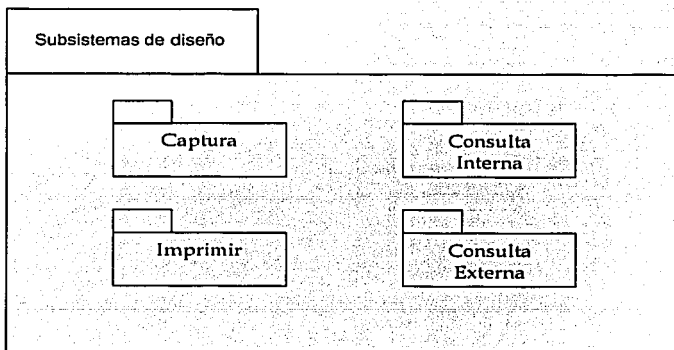
Win9x
 Win2000
 WinXP
 WinML

Si corre en ambiente multiusuario marque servidor y cliente requerido.

WinNT
 Mandrake
 Unix
 Netware

WinXX
 Web
 Unix
 Netware

2. Subsistemas de diseño.



TESIS CON
FALLA DE ORIGEN

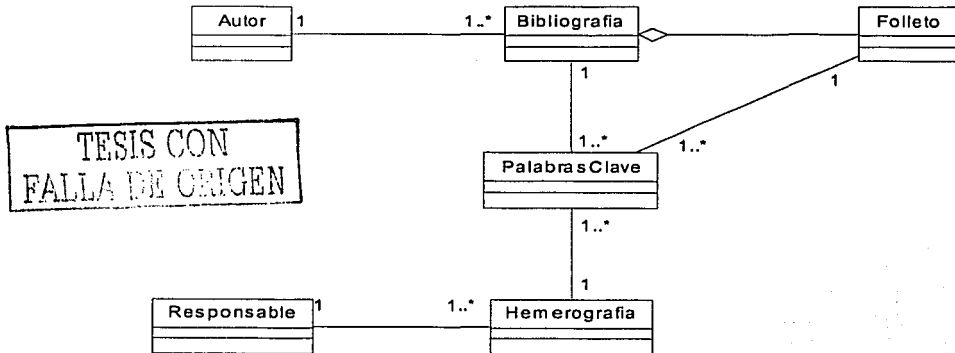
3. Software de desarrollo.

Software	Nombre
Lenguaje de programación	Visual Basic
Manejador de bases de datos	Microlsis
Sistema operativo	Windows
Protocolo de comunicación	Netware
Otros paquetes	Rational
Otras herramientas	Winlisis
Otros	

4. Componentes reusables.

Componente reusable	Equivalencia con la clase de análisis

5. Clases de diseño principales.



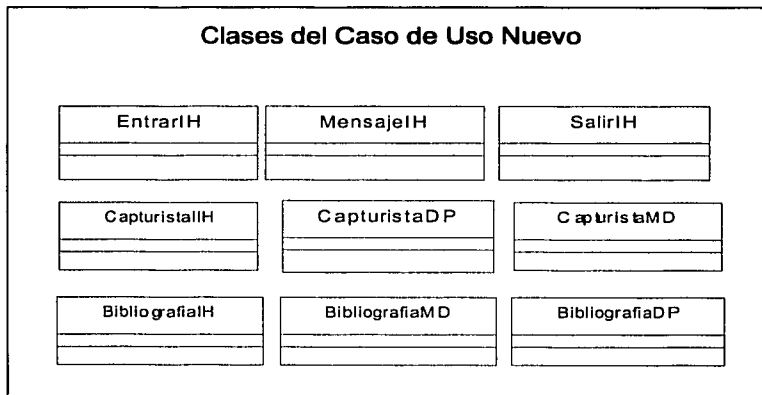
Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software



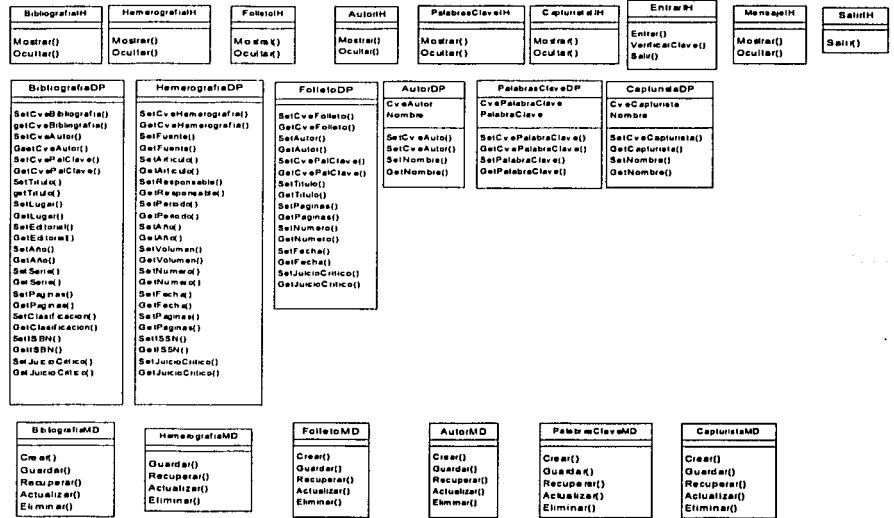
Realización de casos de uso del diseño

Siglas del proyecto	<u>PSIMex</u>	Fecha	<u>15/05/2003</u>
Responsable	<u>JASC, AYRM y MGSM</u>	Ciclo	<u>1</u>

1. Clases del diseño por caso de uso.

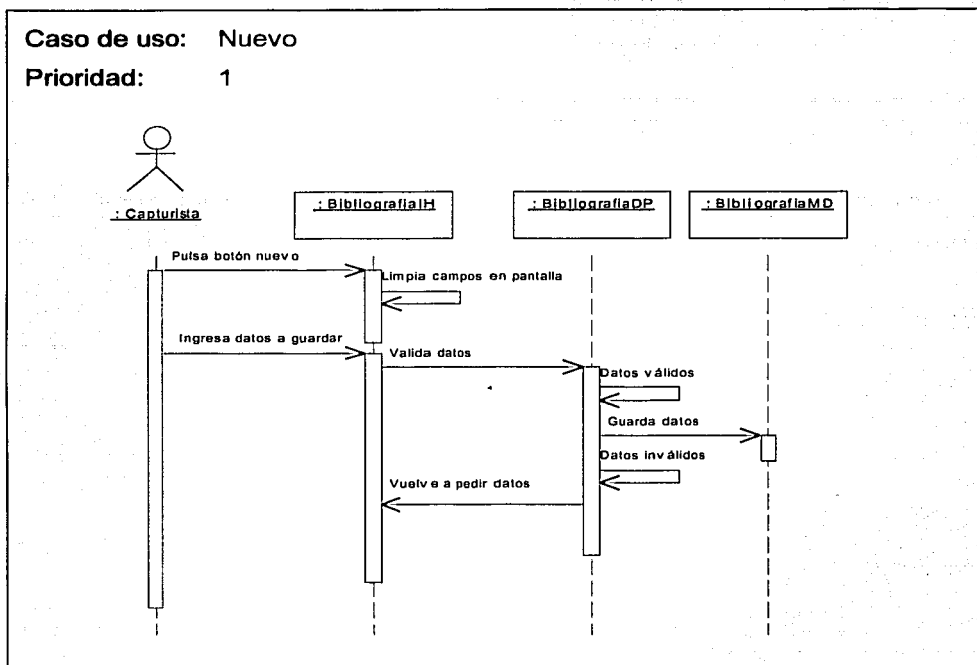


2. Clases del diseño por subsistema.



TESIS CON
FALLA DE ORIGEN

3. Diagramas de secuencia por caso de uso.



4. Captura de requerimientos especiales.

Requerimiento	Descripción
Letras mayúsculas	Todos los datos deberán quedar almacenados en mayúsculas.

TESIS CON
FALLA DE ORIGEN

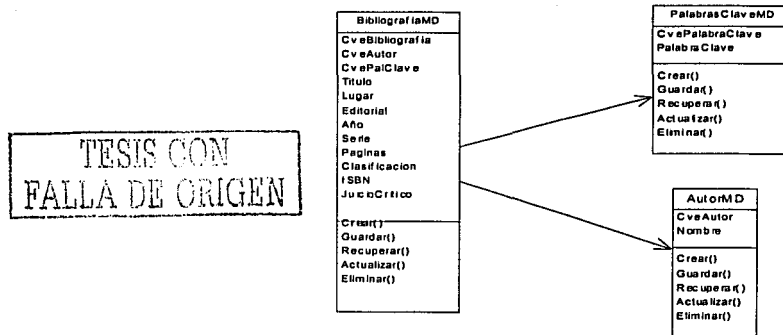
Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software



Clases del diseño

Siglas del proyecto	PSIMex	Fecha	16/05/2003
Responsable	JASC, AYRM y MGSM	Ciclo	1

1. Operaciones de la clase.



Operación:

Crear	<input checked="" type="checkbox"/> Pública	<input type="checkbox"/> Privada	<input type="checkbox"/> Protegida
Guardar	<input checked="" type="checkbox"/> Pública	<input type="checkbox"/> Privada	<input type="checkbox"/> Protegida
Recuperar	<input checked="" type="checkbox"/> Pública	<input type="checkbox"/> Privada	<input type="checkbox"/> Protegida
Actualizar	<input checked="" type="checkbox"/> Pública	<input type="checkbox"/> Privada	<input type="checkbox"/> Protegida
Eliminar	<input checked="" type="checkbox"/> Pública	<input type="checkbox"/> Privada	<input type="checkbox"/> Protegida

Por participación en la realización de casos de uso.

Caso de uso Nuevo Dependencia Asociación Generalización

Requerimientos especiales:

Al capturar datos de la bibliografía se requiere forzosamente proporcionar el autor y las palabras clave asociadas a la bibliografía.

2. Descripción de métodos.

Método	Datos que Recibe		Datos que regresa		Descripción
	Nombre	Tipo	Tipo	Nombre	
FnCrearRegistro	NumConsecutivo Autor Título Edición Pie de imprenta Descripción física Notas Temas OtroAutor Responsable Localización DatosAutor Resumen JuicioCrítico	n an an an an an an an an an an an an an	NumConsecutivo Autor Título Edición Pie de imprenta Descripción física Notas Temas OtroAutor Responsable Localización DatosAutor Resumen JuicioCrítico	n an an an an an an an an an an an an an	Crea un registro nuevo en la base de datos.
FnGuardarRegistro	" "	" "	" "	" "	Guarda un registro bibliográfico en la base de datos
FnRecuperarRegistro	" "	" "	" "	" "	Busca un registro bibliográfico en la base de datos.
FnActualizar	" "	" "	" "	" "	Actualiza un registro bibliográfico en la base de datos.
FnEliminarRegistro	" "	" "	" "	" "	Elimina un registro bibliográfico de la base de datos.

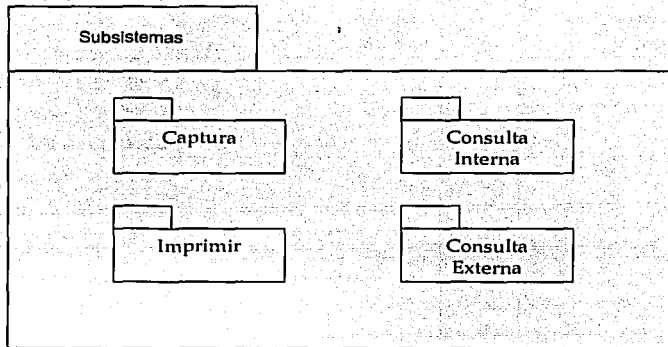
TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software



Subsistemas de diseño

Siglas del proyecto	PSIMex	Fecha	17/05/2003
Responsable	JASC, AYRM y MGSM	Ciclo	1



TESIS CON
FALLA DE ORIGEN

Instituto de Investigaciones Bibliográficas



Departamento de Informática

Sección de desarrollo de Software

Formato de diseño de base de datos

Siglas del proyecto PSIMex Fecha 17/05/2003

Responsable JASC, AYRM y MGSM

Nombre de la base de datos: PSIMB

Nombre de la tabla	Descripción
PSIMB	Tabla que almacena toda la información con respecto a los autores.

Campos que la Integran	Descripción	Tipo	Tamaño	Valor predeterminado
NumConsecutivo	Número con el que se almacena el registro	n	5	
Autor	Persona que escribió la obra	an	150	
Título	Título de la obra	an	900	
Edición	Edición de la obra	an	10	
Pie de Imprenta	Editorial que publica la obra	an	110	
Descripción física	Descripción del estado físico de la obra	an	40	
Notas	Observaciones sobre la obra	an	130	
Temas	Temas relacionados con el contenido de la obra	an	100	
OtroAutor	Nombre de otros autores de la obra	an	150	
Responsable	Persona que ingresa información en la base.	an	5	
Localización	Lugar físico de la obra, igual a clasificación.	an	50	
DatosAutor	Datos relevantes de la persona que escribió la obra	an	1500	
Resumen	Resumen breve de lo que trata la obra	an	1500	
JuicioCrítico	Opinión acerca de la obra que realiza el responsable.	an	1500	

Identificación de llaves			Viene de tabla
Primaria	Foránea	Secundaria	
NumConsecutivo			PSIMB

TESIS CON
FALLA DE ORIGEN

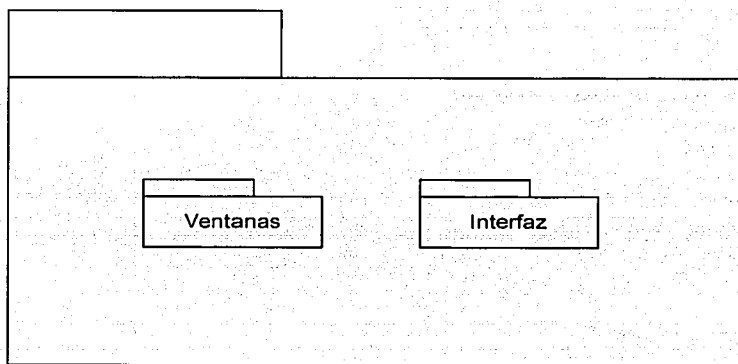
Instituto de Investigaciones Bibliográficas
Departamento de Informática
Sección de desarrollo de Software



Modelo de diseño

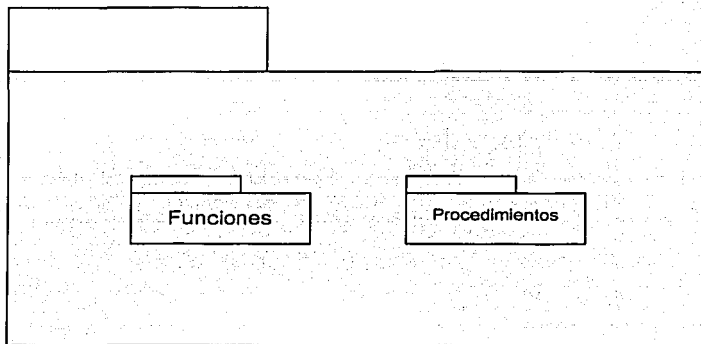
Siglas del proyecto	<u>PSIMex</u>	Fecha	<u>17/05/2003</u>
Responsable	<u>JASC, AYRM y MGSM</u>	Ciclo	<u>1</u>

1. Clases de interfaz humana.

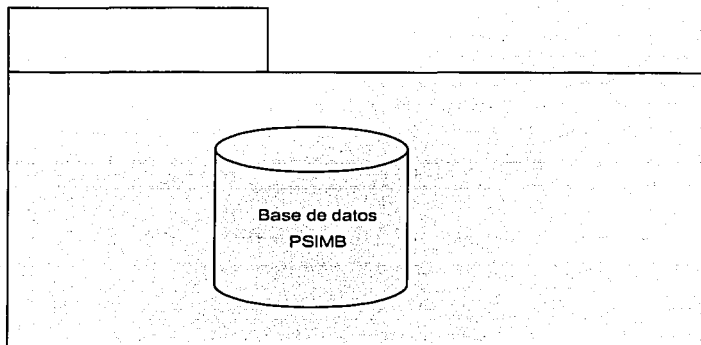


TESIS CON
FALLA DE ORIGEN

2. Clases de dominio del problema.

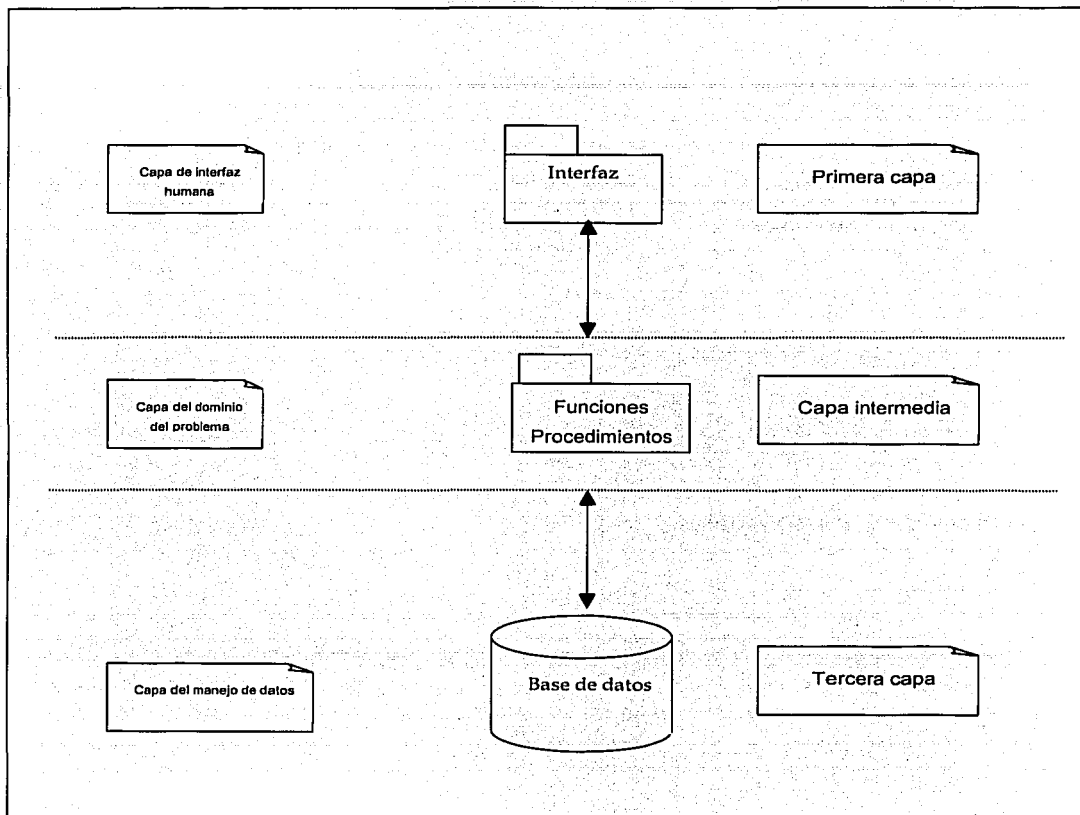


3. Clases de manejo de datos.



TESIS CON
FALLA DE ORIGEN

4. Modelo de diseño.



TESIS CON
FALLA DE ORIGEN

Anexar: Diseño de la arquitectura, Realización de casos de uso de diseño, Clases de diseño, Diagrama entidad relación, Subsistemas del diseño y Formato de diseño de base de datos.