



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

03063
13

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**“La implementación y administración del proyecto,
cambios y configuración del Proceso Unificado;
adaptados a un caso práctico.”**

T E S I S

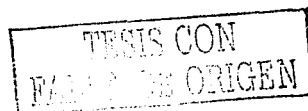
QUE PARA OBTENER EL GRADO DE:

**MAESTRA EN INGENIERÍA
(COMPUTACIÓN)**

P R E S E N T A:

ANA YURI RAMÍREZ MOLINA

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



MÉXICO, D.F.

2003.

A



Universidad Nacional
Autónoma de México



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.

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

NOMBRE: Ana Yun Ramírez

Molina

FECHA: 16/02/2003

FIRMA: 

GRACIAS.

A mi hijo Angel Cipatli, por comprender mi ausencia en algún momento.

A mi esposo Miguel Angel, por todo el cariño, apoyo y comprensión que siempre me dio aliento para seguir adelante.

A mis padres Alfonso Ramírez y Teresa Molina, por todo el apoyo que siempre me han dado, pues sin su ayuda, esto no hubiese sido posible.

A mis Amigos Antonio y Guadalupe, por su ayuda en los momentos difíciles.

A mi asesora Guadalupe Ibargüengoitia, por su paciencia y comprensión en todo momento.

A la UNAM por la formación que me ha dado.

Y finalmente, aun que no menos importante, a Dios por permitirme llegar hasta este momento y rodearme de una familia y amigos incomparables.

TESIS CON
VALOR DE ORIGEN

B

Índice

Introducción.....	I
Capítulo 1. Del modelado de procesos al modelado de software.....	1
1.1 El modelado de procesos.	
1.1.1 Definición.	
1.1.2 Objetivos del modelado de procesos.	
1.1.3 Conceptos básicos.	
1.1.4 Enfoques en la representación de procesos.	
1.1.5 Lenguajes de modelado.	
1.2 El modelado de procesos de desarrollo de software.	
1.2.1 Características.	
1.2.2 Expectativas.	
Capítulo 2. El Proceso Unificado.....	6
2.1. Antecedentes	
2.2 Características	
2.3 Fases y flujos de trabajo	
2.4 EL RUP	
2.5 UML	
Capítulo 3. Planteamiento del proyecto.....	12
3.1 Preámbulo.	
3.2 El caso práctico.	
3.3 Objetivos.	
Capítulo 4. Adaptación del flujo de trabajo Implementación del Proceso Unificado al caso práctico.....	15
4.1 El flujo de trabajo Implementación de acuerdo al Proceso Unificado y RUP	
4.1.1 Actividades	
4.1.2 Roles	
4.1.3 Artefactos	
4.2 El flujo de trabajo Implementación Adaptado al caso práctico	
4.2.1 Actividades	
4.2.2 Roles	
4.2.3 Artefactos	
Anexos.....	35
Capítulo 5. Adaptación del flujo de trabajo Administración del Proyecto del RUP al caso práctico.....	46
5.1 El flujo de trabajo Administración del Proyecto de acuerdo al RUP	
5.1.1 Actividades	
5.1.2 Roles	
5.1.3 Artefactos	
5.2 El flujo de trabajo Administración del Proyecto Adaptado al caso práctico	

C

TESIS CON
FALLA DE ORIGEN

5.2.1 Actividades	
5.2.2 Roles	
5.2.3 Artefactos	
Anexos.....	72
Capítulo 6. Adaptación del flujo de trabajo Administración de Cambios y Configuración del RUP al caso práctico.....	106
6.1 El flujo de trabajo Administración de Cambios y Configuración de acuerdo al RUP	
6.1.1 Actividades	
6.1.2 Roles	
6.1.3 Artefactos	
6.2 El flujo de trabajo Administración de Cambios y Configuración adaptado al caso práctico	
6.2.1 Actividades	
6.2.2 Roles	
6.2.3 Artefactos	
Anexos.....	126
Capítulo 7. Creación de la herramienta de consulta para el seguimiento del proceso de desarrollo de software.....	143
7.1 Objetivo	
7.2 Estructura	
7.3 Funcionamiento	
Conclusiones.....	153
Bibliografía.....	157
Ejemplo de llenado de artefactos.....	158

TESIS CON
FALLA DE ORIGEN

Introducción.

El objetivo de esta tesis es adaptar los flujos de trabajo del Proceso Unificado: Implementación, Administración del proyecto y Administración de cambios y configuración, al Departamento de Informática de una organización que no tienen como objetivo principal la producción de software, lo cual involucra la creación y el establecimiento de un *Proceso de Desarrollo de Software* para dicho departamento.

El hombre en forma natural se encuentra envuelto en un mundo de sistemas que involucran *procesos*, su propio cuerpo es un sistema compuesto de subsistemas en constante ejecución, procesos coordinados y todos con un mismo fin, la permanencia de la vida. Es por eso que hablar de procesos pudiera para el grueso de la población resultar poco relevante, ya que a diario realizamos procesos, la mayoría de las veces, sin darnos cuenta. Sin embargo no todos los procesos que ejecutamos nos dan siempre los resultados que queremos obtener o necesitamos, por lo que en ocasiones es necesario cambiar el proceso y volverlo a intentar.

Es por eso que el tener o mantener un proceso no es el problema, sino el tener el proceso que permita llegar al resultado esperado, en el tiempo esperado y con los recursos con que se cuenta, problemática a la que se enfrenta la Ingeniería de Software, ya que aún no es posible encontrar un *proceso de desarrollo de software único* que garantice para cualquier tipo de desarrollo y cualquier tipo de empresa, un resultado satisfactorio en el tiempo estimado y con los recursos disponibles. Por lo anterior, actualmente la intención es encontrar un *modelo genérico del proceso de desarrollo de software* que pueda ser particularizado para cada caso.

Es así que en 1998 un grupo de investigadores, apoyados por la empresa Rational, dio a conocer un modelo para el proceso de desarrollo de software conocido como *Proceso Unificado*, el cual tiene como premisa la adaptabilidad de un proceso a cualquier tipo de organización o proyecto.

Dado lo anterior, resulta la importancia de la presente tesis, que tiene como objetivo principal adecuar los flujos de trabajo del Proceso Unificado: Implementación, Administración del proyecto y Administración de cambios y configuración, a un caso práctico. Además de colaborar con el proyecto de *Adecuación del Proceso Unificado a un Departamento de Informática*, cuyo objetivo es demostrar la factibilidad de adaptar el modelo de procesos de software conocido como el Proceso Unificado (PU) a una área de informática de una organización, que no tiene como función principal la generación de software, en este caso el Instituto de Investigaciones Bibliográficas de la UNAM, en particular.

El proceso para obtener una respuesta al planteamiento anterior comienza con el modelado del proceso de desarrollo de software, para el departamento en cuestión, adecuado al trabajo cotidiano de la organización y que cumpla con lo establecido por el Proceso Unificado. Por ello, además de la elaboración de un marco teórico para el entendimiento del modelado de procesos y el funcionamiento del Proceso Unificado, es

necesario establecer un equipo de trabajo, ya que se requiere más de un punto de vista para abarcar los diferentes aspectos del proceso, además de dividir el proyecto de la siguiente forma:

- ⇒ Guadalupe Sánchez : Flujos de requerimientos, análisis y diseño;
- ⇒ Ana Yuri Ramírez: Flujos de Implementación, Administración del proyecto, cambios y configuración.
- ⇒ Antonio Salazar: Flujos de Pruebas, Distribución y Ambiente

Debido a la complejidad y factores que influyen en la creación del modelo de un proceso, la adecuación se hace en dos etapas: La primer etapa es en forma individual, en donde cada alumno se encargara de adecuar tres flujos de trabajo, con base a su experiencia y conocimiento, respetando los estándares establecidos por el equipo, coordinados por la Maestra Guadalupe Ibarquengoitia.

La segunda etapa es en forma grupal, donde cada flujo de trabajo es expuesto por su responsable, analizado por el equipo y en caso de ser necesario corregido, para asegurar su objetividad y su relación con el resto del proceso.

Este mecanismo de trabajo se establece por dos razones, primero por que todos los miembros del equipo del proyecto, a excepción de la coordinadora, laboran dentro del departamento de Informática en cuestión, y esto incrementa la posibilidad de que al intercambiar opiniones, el resultado de la adecuación se aproxime más a las necesidades de la organización, sin la necesidad de ir probando cada flujo como lo dice el PU, ya que no se tiene el suficiente tiempo para ello; y segundo por que el Proceso Unificado (PU) así lo recomienda.

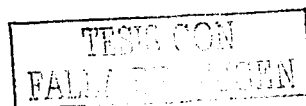
Es así, que este trabajo muestra el resultado de la adecuación de los flujos de trabajo: Implementación, Administración del proyecto y Administración de Cambios y configuración, al proceso de desarrollo de software del Departamento de Informática del Instituto de Investigaciones Bibliográficas, con el cual no sólo pretende establecer una forma de trabajo, sino además se espera sirva para capacitar a los ingenieros de software que laboran o pretendan laborar en dicho departamento.

La presente tesis se encuentra dividida en siete capítulos; en los tres primeros se describen los elementos de estudio necesarios para elaborar el trabajo de adecuación. Los siguientes tres capítulos detallan cada paso efectuado durante la adecuación de cada flujo de trabajo (Implementación, Administración del proyecto y Administración de Cambios y Configuración), en donde se mostrarán los elementos que se tuvieron que evaluar para decidir el contenido, además de los productos (formatos y estándares) que fueron creados para cada uno de los flujos.

Cabe mencionar, que en los capítulos 4, 5 y 6, para ilustrar las diferencias entre lo establecido por el RUP y el resultado de la adecuación, se utilizaron los diagramas del RUP tal como son, con la intención de mostrar que los cambios hechos no sólo están en la estructura de cada flujo de trabajo, sino que además en el hecho de crear un proceso descrito en nuestro propio idioma.

Finalmente después de realizar la adecuación, es necesario crear un instrumento que permita la consulta y seguimiento del nuevo proceso creado. De tal forma que el equipo de trabajo del proyecto creó una herramienta como medio único con el que se pueda consultar el resultado completo de la adecuación del Proceso Unificado. Este instrumento se detalla en el último capítulo, ya que se hace la descripción de la herramienta de consulta del Proceso de Desarrollo de Software del Instituto de Investigaciones Bibliográficas (PDSIB), que se espera sirva como guía para el desarrollo de software del departamento de informática de dicha institución.

Para finalizar podemos decir que al profundizar en el estudio de cada una de los flujos de trabajo del Proceso Unificado, se establece un marco de referencia sólido y concreto en el área de desarrollo de software que puede servir de ejemplo a empresas con áreas de informática pequeñas, que estén interesadas en establecer su propio proceso para la construcción de sus productos.



Capítulo 1. Del modelado de procesos al modelado de software.

1.1 El modelado de procesos.

1.1.1 Definición.

“El objetivo del modelado es la producción de una abstracción de un proceso (modelo) que sirva como base para la especificación de un mecanismo de flujo de trabajo.

El modelo de un proceso posibilita la comprensión de las actividades, dependencias entre estas y los papeles (roles) que son necesarios para efectuar el proceso”[UCM, 2003].

Se sabe que un modelo es la representación abstracta de un objeto o cosa, de tal forma que se resalten las características más importantes del mismo, esto es con la intención de eliminar todo aquello que no sea relevante para representar el objetivo del modelo. Dentro del área de la computación en lo referente a desarrollo de software la definición de modelos se ha usado tradicionalmente para representar flujos de información, lo que en la actualidad resulta insuficiente, pues las investigaciones han mostrado que para integrar productos de software de manera exitosa en las empresas u organizaciones, es necesario modelar también los procesos de creación. Esto nos conduce al modelado de procesos que no sólo se enfoca en el flujo de la transformación de la información, sino que además, hace hincapié en quién, cómo, dónde y cuándo interviene en el proceso.

Ahora, no sólo es necesario tener la colección de todos los pasos y elementos de un proceso para considerarlos como un modelo, además es necesario ponerlos en un medio que permita que un humano los entienda..

1.1.2 Objetivos del modelado de procesos.

A pesar de que el modelado de procesos tiene como objetivo principal mostrar todos los elementos que intervienen durante la ejecución de un proceso, existen otras razones por las que resulta ser útil e importante, como son:

- Permite descomponer las descripciones al detalle para facilitar su ejecución sin mayor explicación.
- Estructura el proceso en unidades independientes que puedan ser integradas para la construcción del modelo.
- Crea un soporte que permite la administración y monitoreo de la ejecución del proceso.
- Crea una base que sirve de ayuda en la capacitación e introducción del personal a la ejecución del proceso.
- Formaliza el proceso para que las personas puedan trabajar en forma más eficiente y en equipo.
- Crea un soporte que sirve de guía para quienes ejecutan el proceso.

**PAGINACIÓN
DISCONTINUA**

1.1.3 Conceptos básicos.

El modelado de procesos se compone básicamente de cuatro elementos: actividades o pasos del proceso, artefactos, roles y agentes; los cuales pueden ser llamados genéricamente *elementos del proceso*.

Una *actividad ó paso del proceso* es aquella unidad atómica (mínima) en la que puede ser descompuesto el proceso. Este elemento no sólo debe ser una parte que permita completar el proceso, además debe contener en sí una posición jerárquica dentro del flujo del proceso que permita saber su dependencia o relación con las demás actividades.

Los *artefactos* son todos aquellos elementos generados, modificados y utilizados durante el proceso, es decir, documentos, diagramas, productos, etc. Por tanto, cada uno de estos artefactos debe tener una descripción precisa de la actividad o actividades en la que es ocupado o creado.

El *rol* es un papel o conjunto de responsabilidades que deberán ser cubiertas por un individuo. Un rol siempre tendrá asignado un nombre que lo distinga de los demás y que represente al conjunto de responsabilidades que cumple.

Un *agente* en una persona o individuo que ejecuta actividades del proceso, es decir un agente es aquel que ejecuta un rol o varios, en determinado momento.

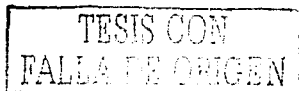
Como puede verse todos los elementos interactúan entre sí, por esa razón al elaborar el modelo cada elemento debe tener una definición clara de la relación que tiene con los demás.

1.1.4 Enfoques en la presentación de procesos.

Como se mencionó un modelo sólo muestra las características más importantes de un objeto. Por esa razón existen diferentes enfoques para realizar el modelado de un proceso, ya que con base a lo que se necesite abstraer, se hará hincapié en las características del modelo del proceso.

Bill Curtis, menciona cuatro como las perspectivas más comúnmente usadas para representar el modelado de procesos [Curtis, 1992]:

- *Funcional*. Muestra qué elementos del proceso son ejecutados y qué flujos de entidades de información (artefactos) son importantes, para estos elementos del proceso.
- *Comportamiento*. Se enfoca en cuándo (secuencia) y cómo son ejecutados los elementos del proceso durante las iteraciones, condiciones y decisiones, entradas y criterios de salida.
- *Organizacional*. Representa cuándo y por quién de la organización (qué agente), es ejecutado un elemento del proceso.
- *Informativo*. Hace relevancia en las entidades de información producidas o manipuladas por un proceso (datos, artefactos, productos finales e



intermedios, objetos, etc.). Esta perspectiva incluye tanto la estructura de las entidades de información como las relaciones entre ellas.

Cómo puede verse estos enfoques representan diferentes aspectos de un proceso, sin embargo, la mayoría de las veces es necesario tener un modelo que abarque más de un aspecto del proceso. Para esta problemática hay quienes han empleado textos que permitan narrar todo aquello que no está incluido en el modelo.

1.1.5 Lenguajes de modelado.

Uno de los problemas más difíciles del modelado de procesos es determinar cómo expresar el modelo, es decir qué lenguaje utilizar, pues es necesario primero saber qué nos interesa modelar, para decidir qué perspectiva es la que debemos tomar y, así seleccionar el lenguaje adecuado, ya que existen varios factibles de ser utilizados dependiendo del enfoque desde el que se quiere analizar el proceso. Por mencionar algunos tenemos:

- Modelos de programación. Este paradigma trabaja bajo el argumento de que todo proceso tiene una descripción y a su vez esa descripción puede ser modelada y analizada como un algoritmo. Este tipo de lenguaje resulta útil cuando se tienen procesos susceptibles de automatizar.
- Modelos funcionales. En este lenguaje un proceso es expresado textual y declarativamente como un conjunto de elementos con atributos de entrada y salida, ordenados jerárquicamente. Este tipo de modelo es mayormente matemático, pues se basa en la teoría de funciones.
- Modelos basados en un plan. Con este lenguaje los modelos se construyen basados en un plan o estrategia de decisión, es decir se tienen por un lado los pasos del proceso y por otro el conjunto de condiciones que darán pauta a la ejecución y selección de los pasos que deberán ser ejecutados del proceso.
- Modelos basados en Redes de Petri. Es un tipo de modelo gráfico que permite expresar las precondiciones y poscondiciones de cada actividad en un proceso, este tipo de lenguaje permite expresar las relaciones de precedencia, dependencia, congruencia y paralelismo entre los diversos tipos de elementos del proceso.

1.2 El modelado de procesos de desarrollo de software.

El modelado de procesos de desarrollo de software se centra en la abstracción de los elementos que intervienen en el conjunto de etapas que tienen como objetivo la generación de un producto de software, por esa razón el ciclo de vida del software es una parte primordial del modelo, ya que el ciclo de vida contempla desde la recopilación de los requerimientos hasta elaboración y entrega del producto solicitado. En otras palabras, el proceso de desarrollo de software tiene como objetivo establecer la forma, el orden y los elementos necesarios para determinar lo que el usuario necesita y de esa manera reunir y crear los componentes que conformarán el producto final.

Por lo anterior, no sólo es necesario modelar el ciclo de vida, también es necesario establecer una metodología de desarrollo, que se adecue a las necesidades de la

organización; un lenguaje y conceptos propios; y un medio que permita el control, ejecución y actualización del proceso de desarrollo de software, ya que el resultado de este proceso aunque siempre es software, siempre es diferente.

1.2.1 Características.

Es importante mencionar algunas de las características propias del proceso de desarrollo de software pues este combina tanto trabajo rutinario como trabajo altamente creativo, ya que siempre el resultado ó producto tiene características diferentes, a pesar de que es creado con el mismo proceso.

El proceso de software es por naturaleza concurrente y distribuible, es decir, existen actividades que pueden ser realizadas por una o varias personas, lo cual significa que en determinado momento varias personas pueden dedicarse a cumplir con una actividad y en otra ocasión esa misma actividad puede llevarse a cabo por una sola persona. Además, algunas actividades pueden ser realizadas en diferentes lugares, fuera del lugar común de trabajo, o al mismo tiempo que otras, sin que deba hacerse así, cada que se ejecuta el proceso.

Por lo anterior, este proceso resulta ser en incierto y no-determinista, pues no es posible predecir para cada tipo de software, cuál será el avance en la ejecución de los pasos del proceso, ya que existen elecciones que dependen de importantes decisiones humanas, que la mayoría de las veces pueden generar ciclos interminables en el proceso.

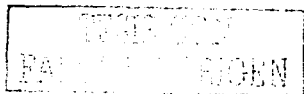
Otra característica importante del proceso de desarrollo de software es la alta probabilidad de errores, cambios y contingencias en el producto, ya que muchas de las veces esto crea retrasos en el proceso, pues al darse el caso de encontrar cambios o errores en el producto es necesario efectuar cambios, lo cual implica ir hacia atrás en el proceso.

Por último la característica más importante del proceso de desarrollo de software, es su evolución constante, la cual provoca la actualización del proceso y que puede ser producida por cambios en las organizaciones o en el contexto, o en la tecnología o en la teoría del proceso..

1. 2. 2 Expectativas

Como se mencionó el desarrollo de software no sólo involucra un conjunto de pasos, artefactos y roles, además requiere adaptarse y evolucionar de acuerdo a las características del producto y a la propia empresa desarrolladora del software. Por esa razón las investigaciones en el área de ingeniería de software están enfocadas en la creación de lenguajes de modelado que permitan una mezcla eficiente entre las múltiples perspectivas de modelado.

Además se espera contar con nuevas técnicas que permitan combinar la creación del modelo con su administración y control de ejecución, además de permitir su propia evolución. Si bien, ya ha habido varios intentos, de hecho las herramientas CASE son un ejemplo, estos productos sólo se enfocaron en el Ciclo de vida del software, dejando a un



lado todos los demás elementos de Administración. Actualmente, existen ambientes de desarrollo de software, los cuales combinan estos elementos.

Otro ejemplo es el Proceso Unificado(PU) que es el resultado de años de investigación y que actualmente sigue teniendo nuevas aportaciones, tal es el caso del Rational Unified Process (RUP), que es una versión más del PU.

Capítulo 2. El Proceso Unificado.

2.1 Antecedentes

El Proceso Unificado tiene su origen en 1967 cuando Ivar Jacobson, creó un método para diseñar los sistemas a partir de bloques interconectados que eran ensamblados y hacían del desarrollo de un sistema, algo más manejable. Pero fue hasta 1998, 31 años después del primer intento, que un grupo de personas encabezados por Booch, Jacobson y Rumbaugh, apoyados por un conjunto de empresas (Rational, Requisite Inc., SQA Inc., Pure-Atria, Performance Awareness y Vigortech), crearon algo llamado *Proceso Unificado* que fue distribuido y puesto a disposición a través del libro *El Proceso Unificado de desarrollo de software*.

2.2 Características

Como vemos, el Proceso Unificado ha sido el resultado de años de investigación, estudio y experiencias acumuladas por sus creadores Booch, Jacobson y Rumbaugh. Y "Tiene como objetivo esquematizar todas las actividades necesarias para la transformación de los requisitos de un usuario en un sistema de software"[Jacobson, 2000]. Pero además, este proceso resulta ser genérico, ya que es capaz de adaptarse o especializarse para cualquier tipo de desarrollo y cualquier tipo de organización.

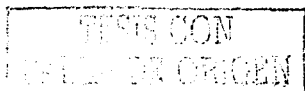
Sus autores lo describen como "Un proceso basado en componentes, lo cual quiere decir que el sistema ó software en construcción está formado por componentes de software interconectados a través de interfaces bien definidas" [Jacobson, 2000].

Este proceso tiene tres ideas centrales que lo identifican y describen en general, y son:

- Esta dirigido por casos de uso
- Se centra en la arquitectura y
- Es iterativo e incremental.

Se puede decir que el Proceso Unificado está *dirigido por casos de uso*, pues tiene como inicio la construcción de un diagrama de casos de uso, que se genera exactamente al inicio del proyecto y sirve como guía durante todo el desarrollo de este. El diagrama de casos de uso describe dos elementos, por un lado lo que se llama *casos de uso* del cual recibe el nombre, que responde a la pregunta ¿Qué debe hacer el sistema? y por otro lado, *los actores* que se enfocan en ¿Para quién?; de tal forma que estos dos elementos permiten abstraer de manera inicial y genérica la primer vista del sistema, que finalmente marcan la dirección durante el proceso de desarrollo del software.

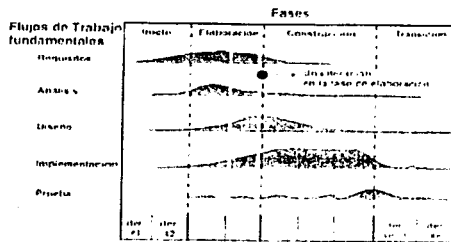
Por otro lado, el Proceso Unificado *se centra en la arquitectura*, pues finalmente la *arquitectura* representa la estructura interna del producto que se pretende construir, de ahí



que obtener el diagrama de casos de usos sea el primer paso, ya que marca el comienzo en la concepción de la arquitectura del sistema.

Por otra parte, el Proceso Unificado *es iterativo e incremental* por que permite dividir el desarrollo del producto final en subproductos (que son versiones del mismo) de tal manera que cada subproducto es desarrollado con el mismo proceso, ocasionando la ejecución iterativa del proceso hasta obtener cada una de las versiones que conforme son creadas van concretando el desarrollo del producto final en forma incremental.

Figura 1. Fases y flujos de trabajo en el Proceso Unificado



El Proceso Unificado está dividido en cuatro fases: Inicio, Elaboración, Construcción y Transición, las cuales son repetidas un número de veces finito, dependiendo del producto. Además existen cinco flujos de trabajo: Requerimientos, Análisis, Diseño, Implementación y Pruebas; que son ejecutados durante las fases. La figura 1, muestra las relaciones entre las fases y los flujos de trabajo.

2.3 Fases y flujos de trabajo

Fases

El Proceso Unificado es ejecutado en cuatro fases las cuales serán descritas a continuación.

Durante la *Fase de Inicio* se tiene como objetivo hacer el análisis del negocio para introducir en el giro de la empresa y de esa forma conocer el entorno y el lenguaje con el que trabajara el software. Esto también permitirá, entender con mayor facilidad lo que el cliente necesita y de esa forma, realizar una descripción de lo que se pretende hacer. De esta manera será posible obtener al final de la fase lo necesario para dar comienzo al desarrollo, como:

- Las funciones principales del sistema.
- Los usuarios más importantes.
- Las prioridades durante el desarrollo.
- Un esbozo de la arquitectura.
- Un plan de desarrollo del proyecto que incluya los costos.

En la *Fase de Elaboración* se construye la arquitectura del sistema, es decir se crea lo que es el esqueleto o estructura interna del software, esto se hace construyendo las diferentes vistas del sistema como son:

- El modelo de análisis.

- El modelo de diseño.
- El modelo de implementación.
- El modelo de instalación.
- El plan de pruebas.
- El plan de integración.

Al finalizar esta fase se tiene como resultado lo que el Proceso Unificado llama *la línea base*, pues todo esto marcará el rumbo durante la construcción del producto.

La *Fase de Construcción* permite hacer realidad lo que se tiene en el papel, es decir, durante esta etapa se crea el producto siguiendo el plano (arquitectura) que se elaboró en la fase anterior, esto quiere decir que en esta fase los cambios que pudieran darse al sistema no deben ser de gran trascendencia, pues en la fase anterior debió trabajarse mucho para obtener lo que denominamos *la línea base*, la cual recibe este nombre por que establece la base de la construcción. Al finalizar la construcción se obtiene el software.

Durante la última fase la de *Transición*, se tiene como principal objetivo el probar el funcionamiento del sistema para corregir los errores detectados y en caso de encontrar errores de gran magnitud incluir su eliminación en el siguiente ciclo del proyecto (versión), este último tipo de errores puede ocasionar que se corrija inmediatamente o se espere al siguiente ciclo, es decir hasta que se haga la siguiente iteración y se consiga una versión incrementada del producto. También en esta última fase se hacen los preparativos para la entrega del producto final.

Flujos de trabajo.

Requerimientos.

El primer flujo de trabajo es "*Requerimientos*" y se refiere a la obtención y recopilación de las necesidades del cliente. A través de este conjunto de actividades debe ser posible conocer el negocio, las necesidades principales del mismo según su giro, el objetivo del software que se quiere crear, adaptar, modificar, etc. y quienes son o serán su principales usuarios. Además en este no sólo se recopila información si no que también se crea un modelo, el cual se expresa a través del diagrama de casos de uso, que permite esbozar las características antes mencionadas de tal forma que sea entendible para el cliente y manejable para el equipo de desarrolladores.

Los requerimientos no sólo permiten saber qué es lo que se va a hacer, además permite-planear cómo debe de trabajarse para obtener el producto deseado, esto quiere decir que una vez que se sabe qué es lo que se quiere, es posible decir en cuántos ciclos debe y puede dividirse el proceso, qué recursos necesitamos, por dónde es posible comenzar, cuánto nos puede costar y aproximadamente cuánto tiempo podemos tardar.

Las actividades de este flujo por lo regular son ejecutadas durante las fases de iniciación, pues permiten cumplir con el objetivo de esa primera fase.



Análisis y Diseño.

Los flujos de trabajo *Análisis y Diseño* tienen como objetivo detallar todos los elementos que forman parte del software, tanto requerimientos funcionales como no funcionales, sin llegar a su construcción, por lo que es posible decir que todas las actividades que forman estos flujos de trabajo están orientadas a la "creación" de la arquitectura del sistema.

En el análisis de los componentes están involucradas una serie de actividades que permiten comprender qué elementos intervienen en cada uno de los casos de uso obtenidos de los requerimientos, su relación, dependencias y atributos. Básicamente el análisis se enfoca en el estudio de los requerimientos para la construcción del modelo de análisis del sistema.

Una vez hecho el análisis lo que sigue es el diseño, el cual involucra actividades encaminadas a encontrar la forma que el sistema debe tener para soportar lo que se planteó en los requerimientos y se estudio en el análisis, es decir se estudian todos los requerimientos no funcionales y restricciones que fueron detectados y que pudieron surgir en el análisis. Todo esto, con el objeto de dejar las bases en el modelo de diseño para la implementación.

Implementación.

La Implementación se encarga de crear todo lo que ya fue previamente analizado y diseñado. A través del establecimiento de estándares de codificación se crea un control sobre la forma en que deben ser implementados cada componente y el orden o secuencia de implementación que deberá de seguirse durante la implementación según el modelo de la implementación.

Durante la implementación cada programador deberá de probar individualmente el funcionamiento que se espera de cada componente generado; esto es por que una vez generados los componentes deben ser integrados para formar el producto final del ciclo.

Pruebas.

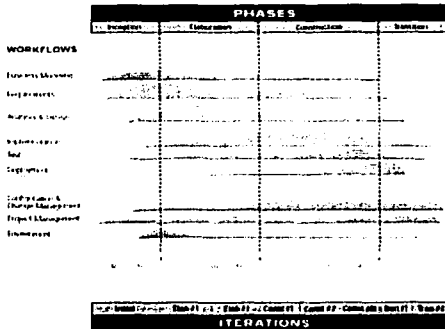
El último flujo de trabajo es Pruebas y como su nombre lo dice tiene como objetivo la realización de todo tipo de pruebas para verificar la calidad del software. Este proceso de detección de errores, esta ligado completamente a la integración de los componentes del sistema, ya que al ir integrando cada componente, ya sea para generar un sub-módulo o el sistema completo, se van realizando pruebas, con base al plan de pruebas, para disminuir el riesgo de errores en el software final.

Por lo anterior los errores detectados deberán de ser corregidos por los programadores en el momento, en caso de ser posible, o en caso contrario identificados para su corrección en los próximos ciclos.



2.4 El RUP

Figura 2. Grafica tomada del RUP



El Proceso Unificado está ligado a la empresa Rational desde 1998, es por eso que en el año 2000 crea una herramienta llamada Rational Unified Process (RUP) que tiene como objetivo guiar en el proceso de desarrollo de software y se basa en el Proceso Unificado. Esta herramienta puede considerarse como una nueva versión del Proceso Unificado, pues como se muestra en la figura 2, agrega cinco flujos más al Proceso Unificado: modelado del negocio, administración del proyecto, administración de cambios y configuración, distribución y ambiente.

Modelado del negocio

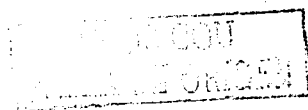
El primer flujo de trabajo llamado *Modelado del negocio*, tiene como objetivo conocer el negocio a mayor profundidad, de tal forma que el equipo de desarrolladores sea capaz de involucrarse en el ambiente del negocio y así comprender con mayor facilidad las necesidades reales del cliente. Por lo regular esta actividad resulta ser muy importante cuando la intención del cliente es automatizar las actividades de su organización y este no distingue entre necesidades y deseos.

Administración del proyecto

La *Administración del proyecto* cumple un papel muy importante en la ejecución del Proceso Unificado, pues tiene como objetivo llevar el control de la ejecución del proceso, asignar o repartir el trabajo, proporcionar los recursos y además contempla todas las actividades que tienen que ver con la interacción con el cliente. Esta nueva aportación al Proceso Unificado permite llevar un registro de las actividades realizadas que al final sirven como experiencia para los siguientes desarrollos y permiten evaluar y corregir errores en el desempeño del proceso de desarrollo de software.

Administración de cambios y configuración

El flujo de trabajo *Administración de cambios y configuración* lleva el control sobre el número de modificaciones hechas al producto durante el proceso de desarrollo, es decir lleva el control de versiones y proporciona las últimas versiones válidas al equipo de desarrolladores de cada uno de los documentos o artefactos generados durante el proceso. Este flujo tiene como objetivo resolver el problema de inconsistencia en la documentación que es muy común en las empresas desarrolladoras de software, pues la mayoría de las



veces al terminar el producto final no se cuenta con la documentación real del mismo, ya que durante el proceso surgen modificaciones que sólo se ven reflejadas en el software pero que no existen en ningún momento en la documentación almacenada.

Ambiente

El flujo de trabajo *Ambiente* contempla todas las actividades necesarias para la preparación del ambiente de trabajo durante el desarrollo (análisis, diseño, implementación y pruebas), por lo que se encarga de reunir los elementos y herramientas, como compiladores y otras herramientas de apoyo, y ponerlos a disposición del equipo de trabajo.

Por otro lado el ambiente también se refiere al entorno de desarrollo, es decir al establecimiento del proceso de desarrollo de software. Es por eso que algunas de las actividades del ambiente (analizar a la organización, levantar caso del negocio, etc.) que se refieren al ambiente organizacional, se realizan sólo una vez mientras se define el proceso de desarrollo de software en la empresa.

Distribución

Distribución como su nombre lo dice contempla todas las actividades que tienen que ver con la preparación del entregable, pues al liberar un sistema o producto no sólo es necesario copiar el software a un disco y darlo al cliente; además tendrá que prepararse documentación que diga cómo fue hecho el sistema, cómo debe usarse, qué necesidades tiene, etc.

2.5 UML

El lenguaje de modelado unificado(UML) fue creado para el diseño orientado a objetos, sin embargo el Proceso Unificado lo ha adoptado como lenguaje de modelado en sus diferentes flujos de trabajo, ya que los creadores de UML también crearon el Proceso Unificado.

Este lenguaje contempla diagramas de casos de uso, que permiten generar una visión genérica del sistema; diagramas de secuencia, los cuales sirven para detallar cada componente del sistema; diagramas de interacción, permiten esquematizar la comunicación o relación entre los componentes; diagramas de estado, permiten ver el comportamiento de cada componente de acuerdo a una situación o suceso; diagramas de actividades, que desglosan las responsabilidades en secuencia, de acuerdo a los roles; y diagramas de clases los que sirven para detallar cada objeto que forma parte del sistema. Todos estos pueden ser utilizados en los flujos análisis y el diseño, pues permiten modelar la arquitectura del sistema.

UML no sólo es útil durante el proceso, además cuenta con diagramas que permiten a su vez modelar el mismo proceso.

Capítulo 3. Planteamiento del proyecto.

3.1 Preámbulo.

El Proceso Unificado surge como una solución al problema de desarrollo de software, proceso que por o regular es desordenado y que muchas empresas enfrentan.

Las pequeñas empresas por falta de recursos, pueden no contar con tecnología y conocimientos, para adoptar un proceso para el desarrollo de software tan grande y complejo.

Ante esta situación, muchas empresas optan por contratar a alguien externo a la organización o empresa, cuando tienen los recursos económicos para esto, pero cuando no, es necesario que las áreas de informática trabajen en la generación de sus propios productos. Ninguna de las dos opciones es la mejor, quien gasta dinero subcontratando fuera, tarde o temprano se da cuenta de lo costoso que es, y también comienzan con la segunda alternativa, que no siempre da los mejores resultados.

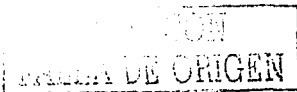
Algunas empresas con mayor organización y dinero, intentan adoptar el Proceso Unificado en sus departamentos de informática, pero en la mayoría de los casos resulta ser muy costoso tratar de seguirlo al pie de la letra con todos los artefactos y roles que están involucrados, además de complicado; lo que provoca que hagan a un lado su propósito.

En México existen grupos de investigadores en el área de ingeniería de software que al tener plena conciencia de esta problemática están tratando de llevar todo el conocimiento generado, recopilado y adquirido a las empresas, para trabajar en colaboración, de tal manera que se enriquezcan ambas partes. Estos trabajos han ido encaminados a la adopción de metodologías, herramientas, estándares, etc. a las áreas encargadas del desarrollo de software, que como se ha comentado, es bastante costoso.

Esto fue motivante en el momento de decidir el propósito de esta tesis, pues surgió una pregunta: ¿Es posible adaptar, el Proceso Unificado al departamento de Informática de una organización pequeña que no tiene cómo objetivo principal la producción de software?. Ya que no tienen la capacidad para adoptar al Proceso Unificado como tal, ¿Será posible adecuar este proceso para que pueda ser llevado a cabo por un área de informática?.

Encontrar la respuesta a esta pregunta no es sencillo, pues se requiere encontrar la organización y el equipo de trabajo adecuado, ya que es necesario contar con la opinión de más de una persona, al estar generando el modelo del proceso, para que éste abarque varios puntos de vista.

Fue así como nuestro equipo decidió escoger una organización que no sólo tuviera las características necesarias para la investigación, además debía de ser una empresa que conociéramos bien para de esa forma restar tiempo en el modelado del negocio y generar un proceso que contemplara todos los flujos de trabajo del Proceso Unificado con la suficiente profundidad y en poco tiempo.



Por esa razón esta tesis es sólo una parte del proyecto "Adaptación del Proceso Unificado a un Departamento de Informática", y está enfocada únicamente en la adecuación de tres de los flujos de trabajo del Proceso Unificado (implementación, administración del proyecto y administración de cambios y configuración) al Departamento de Informática del Instituto de Investigaciones Bibliográficas, el cual fue tomado como caso práctico para fines de esta investigación.

3.2 El caso práctico.

El Instituto de Investigaciones Bibliográficas es una dependencia de la UNAM encargada de la generación de conocimiento en el área bibliohemerográfica; además tiene a su cargo la dirección de la Biblioteca y Hemeroteca Nacionales. Para esta encomienda el Instituto cuenta con un área de informática la cual se encarga de dar servicio a cualquiera de las tres sub-dependencias que lo conforman: Biblioteca, Hemeroteca e Instituto. Este servicio contempla desde la solución de problemas cotidianos como la orientación en el uso y manejo de paquetería hasta la elaboración de sistemas de información para el manejo de bases de datos resultado de la investigación bibliográfica.

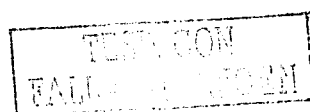
La dependencia cuenta con una red local de aproximadamente 250 terminales las cuales se encuentran conectadas a los cinco diferentes servidores que son también administrados por el personal del departamento de informática (2 Servidores NT, Servidor Unix, Servidor Novell, Servidor Linux).

El Departamento de Informática está compuesto por cinco técnicos académicos incluyendo al jefe de departamento, de ellos sólo tres además de colaborar en el resto de las actividades son quienes desarrollan software. Esta última actividad ha ido creciendo en los últimos años, ya que los investigadores comienzan a interesarse en tener su información en bases de datos. A la vez, han empezado a surgir problemas, pues por lo general no se pueden entregar resultados a tiempo, no siempre se cumple con la expectativa del solicitante, no se tienen documentados los productos generados, por lo que sólo puede darle mantenimiento a cada producto quien lo haya creado y la forma de crear software varía de acuerdo al punto de vista de cada uno de los desarrolladores.

3.3 Objetivos.

Este trabajo pretende colaborar en la adecuación del Proceso Unificado (flujos de trabajo: implementación, administración del proyecto y administración de cambios y configuración) a una organización pequeña que no tiene como actividad principal la generación de software. Lo cual se guiará por los siguientes objetivos:

- Analizar el proceso de desarrollo de software que se sigue actualmente en el Departamento de Informática.
- Analizar los flujos de trabajo, del Proceso Unificado: Implementación, Administración del proyecto y Administración de cambios y configuración.
- Detectar qué actividades del proceso actual tienen que ver con los flujos de trabajo que son el objetivo de este trabajo



- Generar el flujo de trabajo Implementación, adecuado al Departamento de Informática del IIB.
- Generar el flujo de trabajo Administración del proyecto, adecuado al Departamento de Informática del IIB.
- Generar el flujo de trabajo Administración de cambios y configuración, adecuado al Departamento de Informática del IIB.
- Integrar estos flujos de trabajo, en una herramienta que concentre todo el proceso de desarrollo de software adecuado al Departamento de Informática del IIB; la cual servirá de guía en el proceso de desarrollo.

CON
FALLA DE ORIGEN

Capítulo 4. Adaptación del flujo de trabajo implementación del Proceso Unificado al caso práctico.

Este capítulo tiene como propósito, primero mostrar cómo se plantea la especificación de este flujo de trabajo tanto en el RUP (Rational Unified Process) como en el PU (Proceso Unificado), para después continuar con la descripción y justificación de los elementos que se utilizaron y los que no se utilizaron, para la adaptación al caso real. Por esa razón, en la primer sección se menciona a los elementos o conceptos utilizados, y en la segunda parte del capítulo se estará definiendo la adaptación.

Es importante mencionar que al hacer esta adecuación del Proceso Unificado y RUP al caso real, se consideró el proceso de desarrollo actual del departamento de informática del IIB (Instituto de Investigaciones Bibliográficas), como elemento importante en la creación del nuevo proceso, con la intención de conservar los elementos propios del departamento.

Por lo anterior la forma en la que se hará la descripción del trabajo realizado, parte de una comparación entre lo que establece la base teórica y lo que se hace actualmente en el Departamento de Informática, que da como resultado la adecuación obtenida para este caso real.

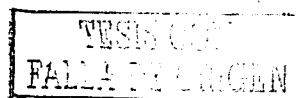
Por último debe mencionarse que para la comparación entre lo establecido por el RUP y el resultado de la adecuación, en algunos casos se utilizaron los diagramas tal cual aparecen en el RUP, con la intención de mostrar que los cambios hechos no sólo están en la estructura de cada flujo de trabajo, sino que además en el hecho de crear un proceso descrito en nuestro propio idioma, el español.

4.1 El flujo de trabajo Implementación de acuerdo al Proceso Unificado y RUP.

En el capítulo 2 comentamos que la implementación se realiza después del diseño, por lo cual en este flujo de trabajo se utilizan varios de los artefactos resultantes del flujo, anterior. Con este flujo se da fin a la fase de elaboración y comienza la fase de construcción, pues la mayor parte de sus actividades están encaminadas a la construcción del software.

El modelo de implementación “describe el cómo los elementos del modelo de diseño y el como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables.” [Jacobson, 2000]; y “describe el cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación, los lenguajes de programación utilizados, y finalmente el cómo dependen los componentes unos de otros” [Jacobson, 2000].

Todos los flujos de trabajo contenidos en el Proceso Unificado, son a su vez pequeños procesos por lo que están compuestos de elementos de proceso que son: actividades, roles y artefactos; los cuales hacen efectivo el cumplimiento de los objetivos.



Ya que el RUP es una extensión del Proceso Unificado, en este trabajo se decidió tomar al RUP como base para la adecuación, por esa razón en las siguientes secciones, se hará la unificación de lo que el RUP contempla como parte del flujo Implementación, haciendo observaciones de lo que el Proceso Unificado menciona al respecto.

4.1.1 Actividades

El Proceso Unificado de la empresa Rational(Rational Unified Process) contempla cinco actividades dentro del flujo de trabajo de Implementación: Estructuración del modelo de implementación, Creación del plan de integración, Implementación de componentes, Integración de subsistemas e Integración del sistema.

Las actividades señaladas anteriormente las podemos ver reflejadas en las tres actividades que el PU establece: Implementación de la arquitectura, Implementación de clases e Integración del sistema; de la siguiente manera:

- En la Implementación de la arquitectura.
La Estructuración del modelo de implementación y la Creación del plan de integración
- En la Implementación de clases.
La Implementación de componentes y la Integración de subsistemas.
- En la Integración del sistema
La Integración del sistema

4.1.2 Roles

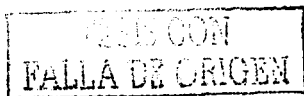
En el Proceso Unificado los roles son llamados *workers (trabajadores)* y se describen de la siguiente manera. "Un *trabajador* define el comportamiento y responsabilidades de un individuo o grupo. El comportamiento está expresado en términos de las actividades que el trabajador ejecuta y cada *trabajador* es asociado a un conjunto de actividades interrelacionadas"[Kruchten, 2000]. Los roles establecidos tanto por el PU como por el RUP son:

- Arquitecto de software o Arquitecto.
- Implementador ó Ingeniero de componentes.
- Integrador de sistemas.
- Supervisor de código

4.1.3 Artefactos

"Un artefacto es una pieza de información que es producida, modificada y usada por un proceso"[Kruchten, 2000]. Dentro del flujo de trabajo de implementación el PU y el RUP definen los siguientes artefactos:

- Modelo de implementación.
- Plan de integración
- Componente.



- Subsistema implementado
- Interfaz.

4.2 El flujo de trabajo Implementación adaptado al caso práctico

Una vez mencionados los elementos que el Proceso Unificado y el RUP consideran para el flujo de trabajo de Implementación, es importante mencionar los factores de la organización que se consideraron al hacer la adecuación:

- ⇒ Las personas disponibles para las actividades de desarrollo de software son 4.
- ⇒ La posibilidad de que las actividades seleccionadas sean ejecutadas por el personal asignado.
- ⇒ La importancia de cada actividad o artefacto dentro de la organización.
- ⇒ La importancia de cada actividad o artefacto dentro del flujo de trabajo del Proceso Unificado.
- ⇒ La necesidad de estandarizar los elementos de uso frecuente.
- ⇒ La necesidad de crear instrumentos de apoyo para la realización de las actividades.
- ⇒ La carga de trabajo para el personal de desarrollo, que viene de actividades que están fuera del Proceso de Desarrollo de software.

Una vez expuestos todos los factores determinantes para la adaptación comenzaremos con la descripción de cada elemento, además de indicar qué fue lo que se cambio, y por qué razón, y cuál es el impacto que tiene en el proceso en general.

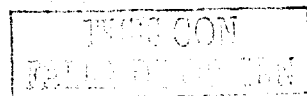
El primer paso para comenzar con la adaptación fue listar las actividades que se llevan a cabo al implementar sistemas dentro del Departamento de Informática del IIB, las cuales fueron comparadas con lo que propone nuestra base teórica, ver tabla 1.

Tabla 1. Comparación entre el proceso actual y RUP -PU.

RUP y PU	Proceso Actual
Estructuración del modelo de implementación.	
Creación del plan de integración	
Implementación de componentes	Programación de funciones ó módulos
Integración de subsistemas.	
Integración de interfaz	Integración de interfaz

Esto se hizo con dos finalidades: la primera para identificar, qué tan diferente de la base teórica se encontraba el proceso actual; y la segunda para encontrar qué elementos pudieran existir en el proceso actual y no en el Proceso Unificado.

El siguiente paso fue el crear un diagrama de actividades para el flujo de trabajo implementación que contemplara los elementos del proceso actual y los elementos del PU y del RUP que enriquecieran el proceso y no entorpecieran el trabajo rutinario del



departamento, incrementando las actividades cotidianas. Pues para decidir si una actividad, rol o artefacto debía de incluirse, era necesario evaluar qué tan factible era integrar cada elemento al proceso, sabiendo que existen pocas personas en el departamento y conociendo además de que la actividad de desarrollo de software no es la única.

Esto con el propósito de obtener un proceso que realmente pueda ser llevado a cabo durante el desarrollo del software del Departamento de Informática del IIB, y además que el modelo de desarrollo de software que se emplee dentro de esta organización tenga una base teórica firme, como lo es el Proceso Unificado, pues de esa manera podemos comprobar que el PU podrá ser adaptado.

Del análisis de actividades antes mencionado se obtuvo como resultado el diagrama de actividades para el flujo de trabajo Implementación del proceso de desarrollo de software del Departamento de Informática del IIB que se muestra en la figura 1.

Figura 1. Flujo de trabajo Implementación en la Adaptación.

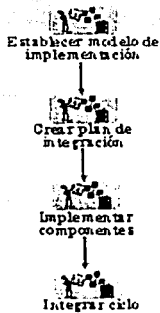
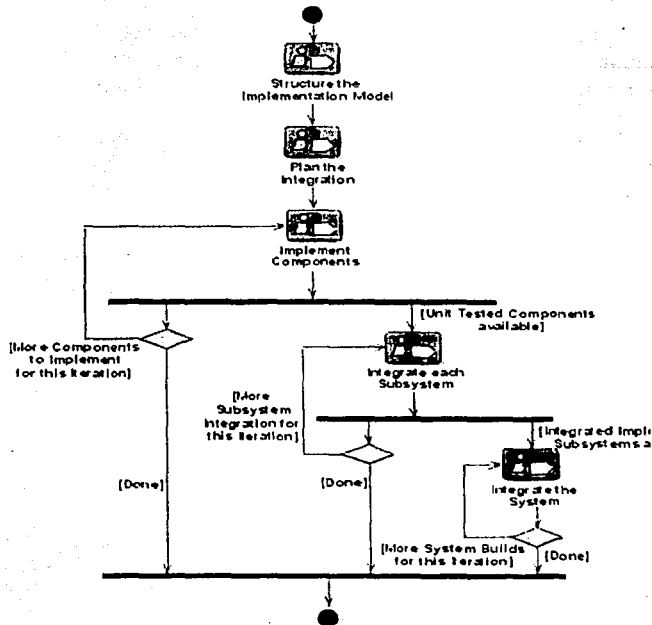


Figura 2. Diagrama tomado del flujo de trabajo Implementación en el RUP.



Si observamos el diagrama de actividades planteado por el RUP en la figura 2; a simple vista podríamos decir que existen varias diferencias, sin embargo no es así.

La primera observación que se puede hacer al respecto, es que el diagrama que se obtuvo refleja las actividades genéricas únicamente, ya que cada una de estas actividades a su vez se desglosa en una serie de pasos. Un ejemplo, es la actividad marcada como Implementar componentes, que en realidad encierra dos actividades del RUP que son implementación de componentes e integración de subsistemas, lo cual se explicará más adelante.

Por último, es conveniente recordar que uno de los propósitos de crear un modelo para el proceso de desarrollo de software es que éste, sirva como guía del mismo proceso, por lo que cada actividad fue detallada lo más posible, con la finalidad de hacer más claro el modelo. Es por eso que además de los diagramas, se crearon formatos que tienen como propósito ayudar en el seguimiento de los pasos planteados en los diagramas.

4.2.1 Actividades

Después de establecer el diagrama general del flujo de trabajo de implementación del caso real, es posible comenzar la descripción de sus actividades.

Establecer modelo de implementación.

En el RUP se llama *Estructuración del modelo de implementación* y tiene como objetivo principal crear el modelo de implementación, el cual se detalla más adelante. Con esta actividad se concluye la creación de la línea base del proyecto, ya que se refina la arquitectura para obtener todos los elementos que se requieren para iniciar con la construcción.

En la figura 3, se muestra el tipo de descripción que existe en el RUP sobre esta actividad, en donde además de estar en otro idioma, únicamente se hace mención de lo que significa estructurar el modelo de implementación y los elementos que participan, lo cual en determinado momento puede resultar escueto o insuficiente. Es por eso que para la adecuación se hizo una descripción detallada de ésta actividad, que contempla una serie de pasos, los cuales se muestran en la figura 4 y se describen a continuación:

Figura 3. Diagrama tomado de la actividad Estructuración del modelo de implementación en el RUP

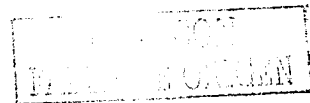
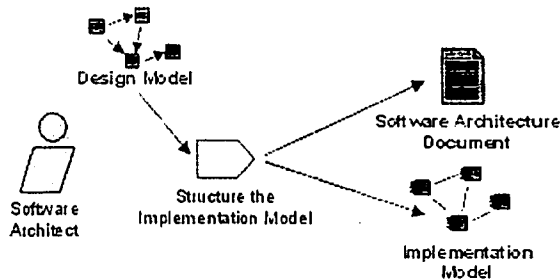
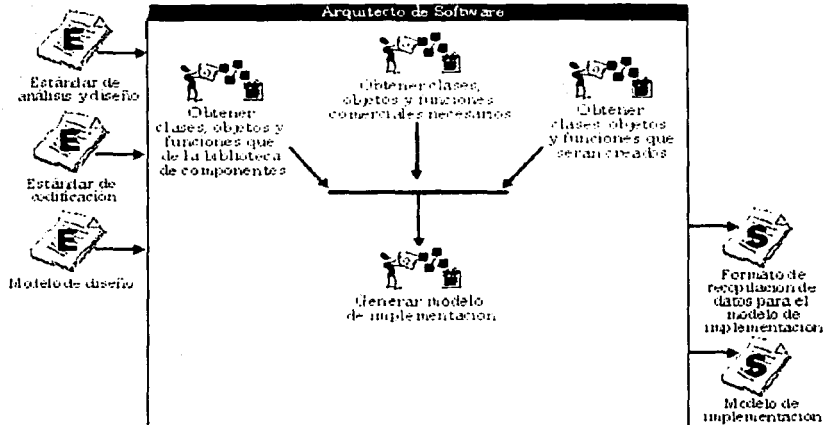


Figura 4. Diagrama de la actividad Establecer modelo de implementación en la adecuación

Establecer modelo de Implementación



- *Obtener clases, objetos y funciones de la biblioteca de componentes.* Del modelo del diseño se toman los nombres de los componentes que se crearán a partir de la reutilización de código, esto se registra dentro del *formato de recopilación de datos para el modelo de implementación*. Ésta información la utilizará después el Ingeniero de componentes para bajar el código de la Biblioteca de componentes y hacer los cambios correspondientes.
- *Obtener clases, objetos y funciones que serán creados.* Esta información también se toma del *Modelo del diseño*, y se refiere a identificar los componentes que serán construidos y detallar las características de cada uno de estos, en el lenguaje de programación seleccionado durante el flujo de trabajo Diseño[Sánchez, 2003]. La codificación de cada componente se hace, trasladando los diagramas de secuencia contenidos en el *Modelo de diseño* al lenguaje de programación, siguiendo lo establecido en el *Estándar de codificación*. Un tipo importante de componentes a codificar dentro de este paso, son los de interfaz, que ya han sido construidos sin ninguna funcionalidad, durante el flujo de trabajo Diseño[Sánchez, 2003], sin embargo en éste paso deben completarse para que funcionen. Para recopilar toda la información de éste paso se creó la tercer sección del *formato de recopilación de datos para el modelo de implementación*.

LIBRO
RECEBIDO EN

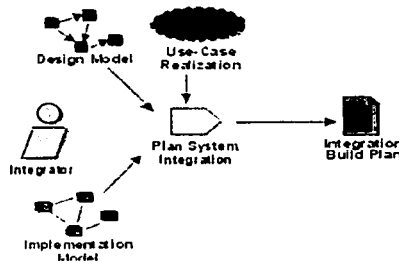
- *Obtener clases, objetos y funciones comerciales necesarios.* Al igual que la información del paso anterior, esto se toma del modelo del diseño, y se refiere a los nombres de los componentes de productos comerciales como son (Visual Basic, Java, C++, etc.) que se utilizarán en cada componente nuevo. Esto se registra en la segunda sección del *formato de recopilación de datos para el modelo de implementación*, de cada componente nuevo. Esta información le permite saber al Ingeniero de componentes, qué librerías o componentes del lenguaje de programación se deben tener configuradas en el lenguaje de programación, para comenzar la creación de cada componente.
- *Generar Modelo de Implementación.* Este modelo se genera a partir de la información obtenida en los pasos anteriores, que está contenida en el *formato de recopilación de datos para el modelo de implementación* de cada componente y con lo establecido en el *Estándar de análisis y diseño*. Para este paso se crea un ejemplo que permite visualizar la forma final del Modelo de Implementación, la primera parte esquematiza a cada componente con sus interfaces, la segunda sección muestra gráficamente cómo interactúan los componentes a través de sus interfaces y sus relaciones y finalmente la última parte del modelo de implementación muestra los nodos del sistema y los componentes que se encuentran en cada nodo.

Para esta actividad, fue creado el *formato de recopilación de datos para el modelo de implementación*, el cual se llena durante la ejecución de cada paso, tantas veces como componentes existen, con la intención de detallar cada componente para que al finalizar sólo se haga el diagrama del modelo de implementación, a partir del cual el Ingeniero de componentes construye el sistema. Este artefacto puede ser consultado al final del capítulo en la sección de anexos.

Crear plan de integración.

La siguiente actividad es *Crear plan de integración* que el RUP denomina de la misma manera y que tiene por objetivo establecer “qué subsistemas serán implementados y el orden en el que serán integrados” [Kruchten, 2000]. La descripción de esta actividad en el RUP (ver figura 5) es muy general, por lo cual fue necesario desmenuzar utilizando la experiencia y la definición planteada en el RUP, para crear un diagrama de actividades de los pasos a seguir durante esta actividad.

Figura 5 Diagrama tomado de la actividad Crear plan de Integración en el RUP.

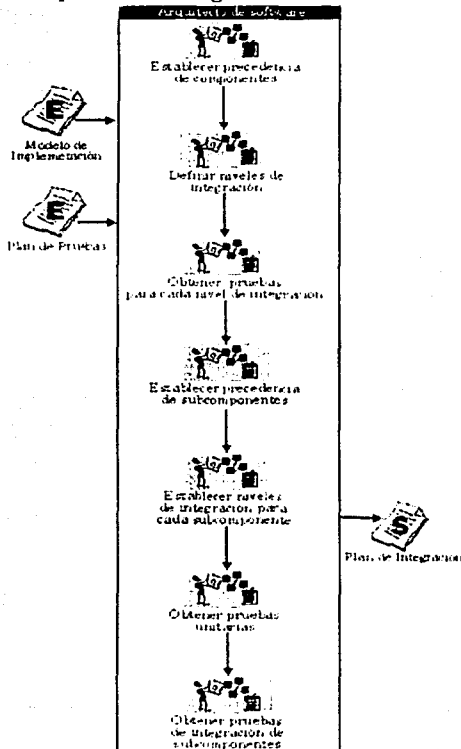


El resultado obtenido durante la adaptación de esta actividad se muestran en la figura 6 y se compone de los siguientes siete pasos:

- **Establecer dependencias entre componentes.** Esto se hace con base en el modelo de implementación. Y se refiere a establecer el orden de importancia que tiene cada componente por las dependencias que tiene con otros componentes o por las funciones que realiza, es importante considerar la creación de la base de datos como un componente importante, pues se requiere para la operación de todo proyecto. Esto significa que un componente mientras menor relación tenga con los otros componentes, menor dependencia tendrá. Esto es importante ya que con base a esta información es posible saber qué elementos deben de existir para poder integrar cada componente y así definir los niveles de integración. Esta información se registra en el *formato para el plan de integración*, en la primera sección.

Figura 6. Diagrama de la actividad Crear plan de integración en la adecuación.

Crear plan de integración



SECCION
FALLA DE ORIGEN

- *Definir niveles de integración.* Los niveles de integración se refieren a las etapas de la integración, pues el proceso de integrar se realiza en forma paulatina, ya que es necesario ir adicionando cada componente, poco a poco, de tal forma que al agregar un nuevo componente se tenga la seguridad de que lo integrado anteriormente no tiene problemas. Los niveles de integración estarán dados por el número de componentes a integrar y sus dependencias, que fueron definidas anterior. Es decir si existen componentes que dependen de otros, es muy probable que esos componentes no se integren (segundo nivel) hasta que los componentes de los que dependen estén integrados (primer nivel).
- *Obtener pruebas para cada nivel de integración.* Estas pruebas se establecen con base al Plan de pruebas y los componentes que se encuentran en el nivel, ya que de acuerdo a cada componente son las pruebas que deben aplicarse.
- *Establecer dependencias de sub-componentes.* Este paso es muy parecido al primero, sin embargo la diferencia es que ahora se hace un análisis por sub-componente. Esta información también se obtiene con el modelo de implementación.
- *Establecer niveles de integración para cada sub-componente.* Al igual que el paso anterior la información que ahora se debe detallar es la de las clases, las funciones y los procedimientos que forman el componente.
- *Obtener pruebas unitarias.* Este apartado se toma del Plan de pruebas, para establecer las pruebas que deben pasar los sub-componentes antes de ser integrados como un componente. Estas pruebas, deben efectuarse directamente por el Ingeniero de componentes en el momento de la construcción.
- *Obtener pruebas de integración de sub-componentes.* Este paso es igual al tercero, sólo que aquí se establece en base a los sub-componentes.

Como apoyo, en la ejecución de esta actividad se creó el *formato para el plan de integración*, el cual orienta en la forma en que debe recopilarse la información de cada uno de los pasos. Y que finalmente al ser llenado se convierte en el Plan de Integración.

Implementar componentes

La tercera actividad *Implementar componentes* abarca dos actividades del RUP que son: *Implementación de componentes e Integración de subsistemas*, las cuales se describen en forma separada (ver figuras 7 y 8), sin embargo, al examinar las descripciones de las dos actividades se vio la posibilidad de unir las en una sola, ya que las dos actividades se refieren a la construcción del producto y son realizadas por el programador, pues por un lado se crea cada parte independiente y por el otro se unen esas partes para formar el componente que en su momento puede ser un subsistema, módulo, clase o función.

Figura 7. Diagrama tomado de la actividad Integrar subsistemas en el RUP

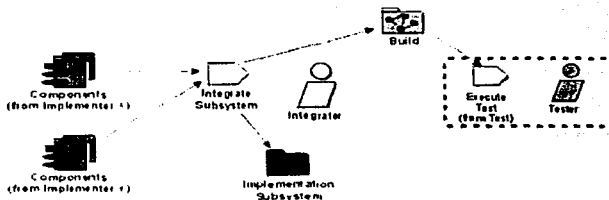
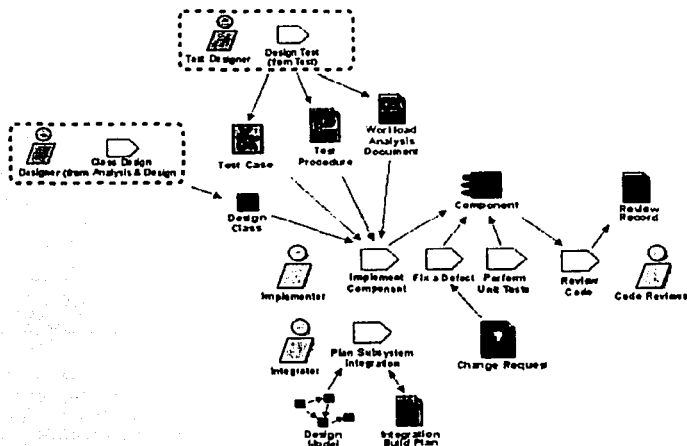


Figura 8. Diagrama tomado de la actividad Implementación de componentes en el RUP



Por otra parte los sistemas que se desarrollan dentro del departamento de Informática del IIB, no son muy complejos, ya que por lo regular, la construcción puede realizarse por una sola persona, por lo que la integración de subsistemas se realiza casi en el momento de la construcción de los sub-componentes.

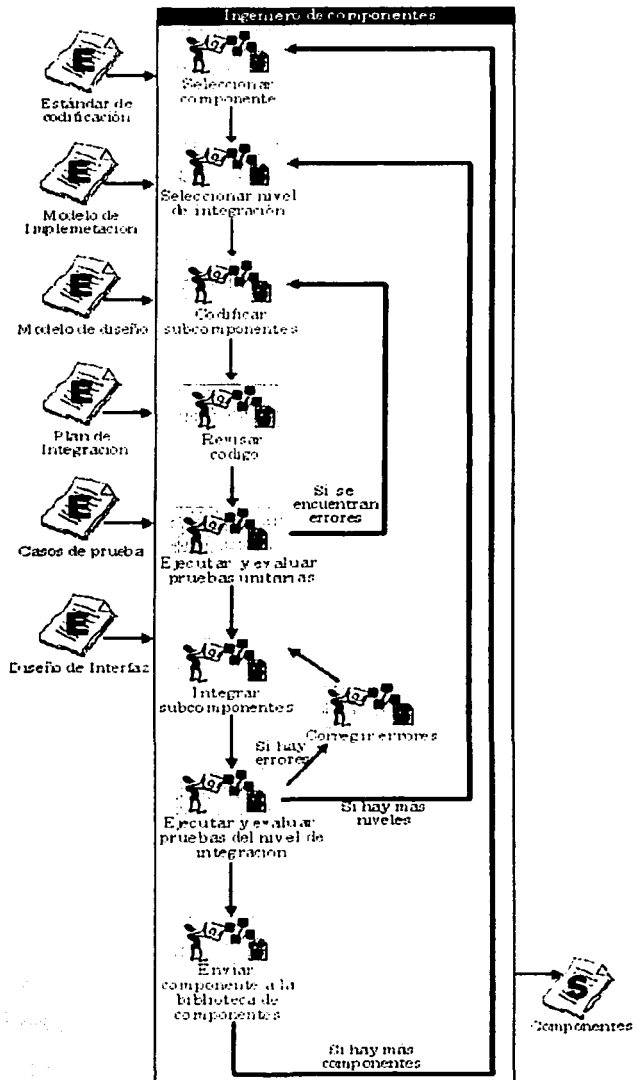
Del resultado de la unión de éstas dos actividades se obtuvo, el diagrama de la figura 9, que muestra el proceso de construcción del producto y se compone de los pasos:

- **Seleccionar componente.** Se refiere a identificar el componente con el que se comenzará a construir del Modelo de Implementación. Este paso se realiza cada que se comienza a construir un nuevo componente. Para determinar por cuál componente debe comenzarse la construcción, deben considerarse los niveles de integración del sistema, definidos en el *Plan de Integración*, pues lo más conveniente es construir primero siempre los componentes del primer nivel de integración y avanzar en la construcción conforme se realizará la integración. Esto

resulta conveniente por que por lo regular todos los componentes de un nivel de integración tienen relación entre ellos y eso hace más fácil su entendimiento.

Figura 9. Diagrama de la actividad Implementar componentes en la adecuación.

Implementar componentes



- *Identificar nivel de integración del componente.* En el Plan de Integración, se encuentran detallada la secuencia de construcción de cada componente del Modelo de Implementación, por lo cual debe consultarse el Plan de Integración para saber cuál es el nivel de integración por el cual debe comenzar la construcción y a su vez seguir.
- *Codificar sub-componente.* Una vez claro el punto de partida de la construcción, se inicia la programación de cada sub-componente, en base al estándar de codificación y el Modelo de Implementación. Lo cual significa pasar el código establecido en el Modelo de Implementación para cada sub-componente y obtener los componentes completos. Es importante recordar que dentro de los componentes están la base de datos y la interfaz, por lo que es necesario para esto, tomar el diseño de la base de datos que se encuentra dentro del *Modelo del diseño* y el *Diseño de la interfaz*, la cuál fue creada durante el flujo de trabajo Diseño[Sánchez, 2003], por lo que en éste paso sólo se completará en base al Modelo de Implementación, para que tenga funcionalidad.
- *Revisar código.* Durante el proceso de construcción de cada sub-componente, el Ingeniero de componentes debe revisar detenidamente el código fuente para identificar errores de sintaxis antes de compilarlo. Para esta revisión se creo el *formato de verificación de código*, el cual orienta al Ingeniero de componentes, en el tipo de revisión que debe hacer sobre sus programas, esta revisión ayuda a documentar el código fuente de cada componente.
- *Ejecutar y evaluar pruebas unitarias.* Las pruebas unitarias están mencionadas en el plan de integración y detalladas en los casos de prueba, por lo que, sólo es necesario consultar esos artefactos para saber qué pruebas deben efectuarse. En caso de encontrar errores es necesario volver a la codificación, para reparar el problema. La forma de ejecutar y evaluar las pruebas esta descrita en el Flujo de trabajo Pruebas, dentro de las actividades ejecutar pruebas y evaluar pruebas[Salazar, 2003], por lo que es necesario revisar esas actividades para poder ejecutar este paso.
- *Integrar sub-componentes.* Una vez terminados todos los sub-componentes del nivel de integración, deben integrarse, con base a lo establecido en el *Plan de Integración*.
- *Ejecutar pruebas del nivel de integración.* Después de unir los sub-componentes es necesario probar que funcionan correctamente, para pasar al siguiente nivel, esto esta detallado en los *casos de prueba* referidos en el *Plan de integración*. A igual que el quinto paso, la forma de ejecutar y evaluar las pruebas esta descrita en el Flujo de trabajo Pruebas, dentro de las actividades ejecutar pruebas y evaluar pruebas[Salazar, 2003], por lo que es necesario revisar esas actividades para poder ejecutar este paso.
- *Corregir errores.* Este paso sólo se ejecuta cuando al hacer la integración se encuentran errores.

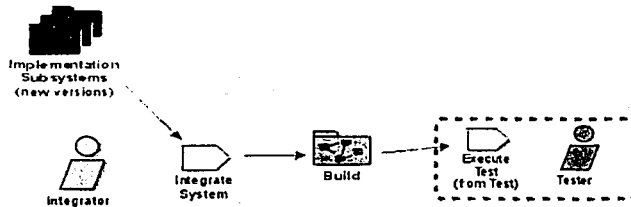
- *Enviar componente a la biblioteca de componentes.* Una vez que el componente aprobó todas las pruebas, el Ingeniero de componentes envía cada componente a la biblioteca de componentes, para su almacenamiento. El responsable de la biblioteca de componentes, toma el Modelo de Implementación para registrar los datos del componente en la biblioteca.

Finalmente esta actividad da como resultado los componentes que serán integrados en la siguiente actividad.

Integrar ciclo

La última actividad llamada *Integrar ciclo*, dentro del Proceso Unificado la llaman *Integrar sistema* (ver figura 10) y esta se refiere a juntar todos los componentes (módulos, subsistemas, etc.) que dan como resultado el producto final de cada ciclo. Y se dice de cada ciclo, por la característica incremental del mismo proceso, pues si recordamos, el Proceso Unificado divide el desarrollo de software en ciclos, de tal forma que en cada iteración se va obteniendo un grado de avance mayor del software final, de ahí el origen del nombre de esta última la actividad.

Figura 10. Diagrama de la actividad Integrar sistema tomado del RUP



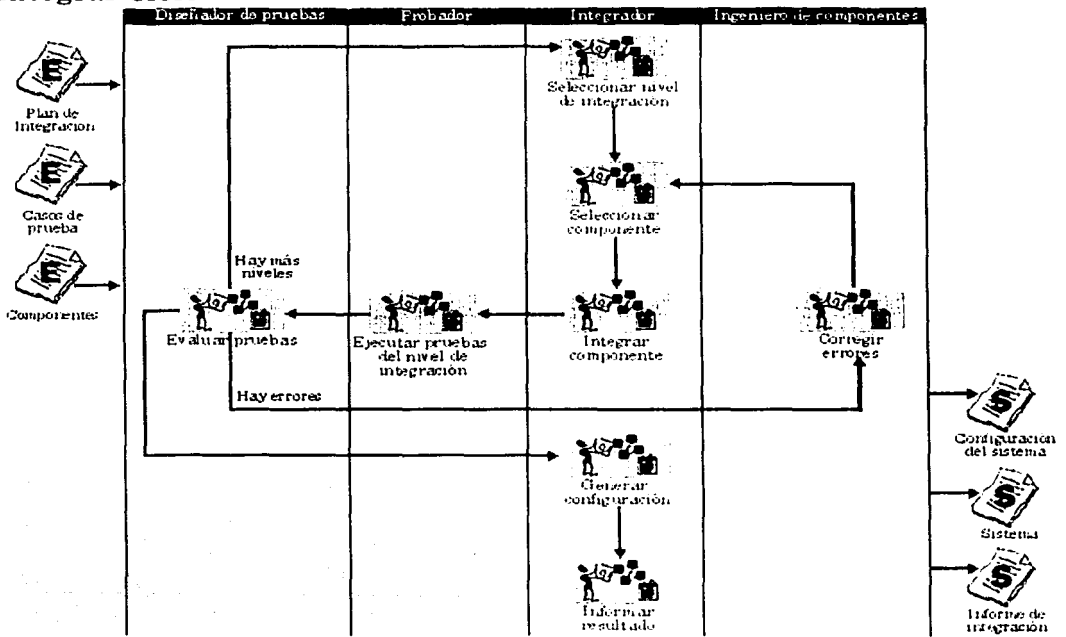
Al igual que para las actividades anteriores, se hizo un análisis para determinar los pasos a seguir en esta actividad y así obtener el diagrama que se muestra en la figura 11. Los pasos de esta actividad son descritos de la siguiente manera:

- *Seleccionar nivel de integración.* Una vez más es necesario el Plan de Integración, pues de acuerdo a lo establecido en este plan, será lo que se integrara cada vez, ya que la integración es un proceso paulatino, y los componentes se van agregando poco a poco.
- *Seleccionar componentes.* Una vez que se sabe en que etapa de la integración está, es posible identificar los componentes que participan.
- *Integrar componentes.* Se unen los componentes de acuerdo a lo establecido en el Plan de Integración.

- *Ejecutar pruebas del nivel de integración.* Cada vez que se agrega un componente es necesario hacer las pruebas establecidas, para confirmar que todo está correcto. Este paso es en realidad la actividad ejecutar pruebas del Flujo de trabajo Pruebas[Salazar, 2003], por lo que es necesario revisar esta actividad para poder realizar este paso.
- *Evaluar pruebas.* Al igual que el paso anterior éste es una actividad del Flujo de trabajo Pruebas[Salazar, 2003] y se llama Evaluar pruebas, por lo que debe consultar esta actividad para realizar éste paso.
- *Corregir errores.* Esto sólo se hace en caso de encontrar algún problema durante la integración, y es responsabilidad del Ingeniero de componentes.
- *Generar configuración.* Este paso está encaminado a la definición de los parámetros que deben establecerse en el equipo en donde se instalará el nuevo sistema, esto se hace con base a las condiciones establecidas en cada caso de prueba de los componentes integrados, llenando el formato de configuración del sistema.
- *Informar el resultado.* Finalmente se llena el formato para el informe de la integración. Este paso es importante pues a través de este documento el administrador del proyecto se entera de lo sucedido durante la integración.

Figura 11. Diagrama de la actividad Integrar ciclo en la adecuación.

Integrar ciclo



RECIBIDO CON
FECHA DE ORIGEN

4.2.2 Roles

Antes de comenzar es importante decir, que las obligaciones para cada rol están marcadas por las actividades del flujo de trabajo de las que son responsables, esto quiere decir que el detalle de responsabilidades de cada *trabajador* se ve claramente en los diagramas detallados de cada actividad del flujo Implementación adecuado. Por esa razón, en esta sección sólo se hará referencia a la actividad que le corresponde al rol.

Al igual que en las actividades, para los roles fue necesario hacer un análisis de los roles existentes en el Departamento de Informática del IIB y en el Proceso Unificado, para establecer las responsabilidades que cada *trabajador* y así determinar la importancia y la viabilidad de incluirlos o no (ver tabla 2).

Tabla 2. Comparación entre roles existentes en el proceso actual y en el RUP

RUP y PU	Proceso actual
Arquitecto de software Implementador (Ingeniero de componentes) Integrador de sistemas. Supervisor de código	Programador

De lo anterior se generaron los roles o trabajadores que se consideraron convenientes para la adecuación en este flujo de trabajo:

- Arquitecto de software
- Ingeniero de Componentes
- Integrador

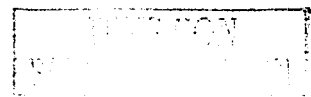
Arquitecto de software

El primer rol llamado *Arquitecto de software* tiene como principal responsabilidad la construcción de la arquitecta, que en este flujo se culmina al crear el Modelo de Implementación y el Plan de Integración.

Este rol surge desde el flujo de trabajo Requerimientos[Sánchez, 2003] y tiene las mismas responsabilidades durante todo el desarrollo del proyecto, por lo que es posible encontrar una definición más amplia, en el trabajo que se refiere a la adaptación de los flujos de requerimientos, análisis y diseño[Sánchez, 2003].

Ingeniero de componentes

El *Ingeniero de componentes* es el responsable de la actividad Implementar los componentes. Este *rol*, a diferencia de otros, se puede asumir por varias personas del equipo de trabajo al mismo tiempo, pues la cantidad de trabajo en ocasiones exige de más de un programador para completar el producto de una entrega ó ciclo.



Algo característico de ésta adecuación, es que en este caso el *Ingeniero de componentes* no sólo se encarga de codificar, además debe revisar el código como lo hace el *Supervisor de código* en el RUP, lo cual quiere decir que una vez codificado por completo un componente, éste debe ser revisado antes de compilarse, siguiendo un procedimiento más formal, con la intención de detectar errores de sintaxis. Esta decisión se tomó debido a las circunstancias de trabajo existentes en el Departamento de Informática, pues el equipo de trabajo para desarrollo, la mayoría de las veces, es una persona o a lo más dos.

Esta revisión puede no resultar tan eficaz como lo establece el Proceso Unificado, pues al hacerla la misma persona que codifica el componente, podría pensarse que no se obtienen resultados, sin embargo, se espera que al establecerla claramente como una responsabilidad del Ingeniero de componentes durante la implementación se pueda ir creando cierta costumbre, la cual con el tiempo puede acarrear buenos resultados.

Integrador.

El último rol es el *Integrador* que tiene como función principal reunir todos los componentes construidos por los programadores y ensamblarlos para conformar el producto final a entregar, esto se encuentra detallado en la actividad Integrar Ciclo.

Diseñador de pruebas y Probador

Estos dos roles aparecen durante la integración, sin embargo no están definidos dentro de éste flujo de trabajo, ya que en realidad surgen dentro del Flujo de trabajo Pruebas [Salazar, 2003], por lo que ahí, puede consultarse la definición de las responsabilidades de estos roles.

4.2.3 Artefactos

Los artefactos son tanto los instrumentos necesarios para realizar cada actividad como los que se producen a lo largo del flujo de trabajo. Sin embargo, por cuestiones de claridad sólo se hará una descripción de los artefactos creados durante el flujo de trabajo Implementación.

Al hacer el análisis de los artefactos creados durante este flujo de trabajo según el Proceso Unificado, también fue necesario compararlos con los artefactos producidos por el proceso actual del Departamento de Informática (ver tabla 3).

Tabla 3. Comparación entre los artefactos del Proceso Actual y el RUP.

RUP y PU	Proceso actual
Modelo de implementación. Descripción de arquitectura (vista del modelo de implementación). Plan de integración Componente. Subsistema implementado Interfaz.	Sistema implementado

FALLA DE ORIGEN

Al llegar a estos elementos del proceso nos enfrentamos a un problema común en el proceso de desarrollo de software, el cual se refiere a la variación que existe en el contenido y la forma de los artefactos, esto debido básicamente, a la falta de estándares tanto de desarrollo como de documentación y al desconocimiento por parte del personal de sistemas del objetivo de cada artefacto. Por esa razón al adecuar los artefactos fue necesario no sólo determinar cuáles se iban a generar, además se estableció un formato para cada uno, con el propósito de crear estándares de documentación, contenido y forma.

Los artefactos resultantes de la adecuación se dividen en dos tipos: los formatos de documentos generados, que ayudan a mostrando el contenido específico de cada artefacto y los artefactos que se convirtieron en estándares. Los artefactos se clasifican de la siguiente forma:

- Documentos
 - Formato de recopilación de datos para el modelo de implementación.
 - Modelo de implementación.
 - Plan de Integración.
 - Verificación de código.
 - Configuración del sistema
 - Informe de Integración.
- Estándares
 - Codificación

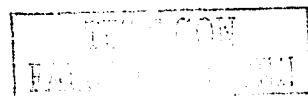
Formato de recopilación de datos para el modelo de implementación.

El primer documento *Formato de recopilación de datos para el modelo de implementación* se creó durante la adecuación con intención de facilitar la creación del *Modelo de Implementación*, pues al llenar el formato es posible darse cuenta de cada uno de los elementos que deben colocarse gráficamente en el modelo de implementación. En este formato se hace la codificación de cada componente, por lo que el Ingeniero de componentes, sólo tendrá que transcribir el código, en el momento de construir. Sin embargo, puede suceder que el Ingeniero de componentes modifique la codificación durante la construcción, por lo que al final de la construcción si existen diferencias, deberá actualizar el formato.

Modelo de Implementación

El *Modelo de Implementación* muestra la implementación de los de los elementos del diseño mediante componentes con sus interfaces y dependencia entre componentes. Este artefacto se compone de tres secciones:

- ✓ Componentes. Dibuja el diagrama de cada componente con su interfaz
- ✓ Diagrama de componentes. Muestra las dependencias entre los componentes
- ✓ Vista de despliegue. Muestra la disposición física de los recursos, tales como servidores e interconexiones
- ✓ Formato de recopilación de datos. Contiene una descripción detallada hasta el código de cada componente.



Es importante mencionar que el lenguaje de diagramación utilizado es UML, sin embargo para hacer de la elaboración de este artefacto, algo más sencillo se integró la descripción de cada símbolo de acuerdo a UML, en el estándar de análisis y diseño, que se creó en los Flujo de trabajo Análisis y Diseño[Sánchez, 2003].

Plan de Integración

Para el *Plan de Integración* se creó un formato el cual al ser llenado da como resultado el plan en sí. Esto se hizo con la intención de establecer y estandarizar el contenido del plan de integración en una forma más clara y concreta, ya que al hacer el análisis de este documento se identificó información que no tenía porque generarse cada que se realizaba un nuevo sistema. Por mencionar un ejemplo: en la descripción que hace el RUP del contenido del Plan de Integración, menciona que el documento en su inicio debe llevar una introducción y un alcance, lo cual en esta adaptación resultaba tedioso hacerlo tantas veces como sistemas se desarrollan, pues el mayor beneficio de hacer un plan de integración es mostrar la estrategia de integración, por lo que no era necesario para cada plan volver a escribir una introducción.

Registro de verificación de código

El *Registro de verificación de código* sirve únicamente para registrar cuántos errores se encontraron durante la revisión, esto de alguna manera puede motivar al programador, pues cada que este hace una nueva revisión se da cuenta de si encontró más o menos errores que la vez anterior y de alguna manera, provoca que al hacer su trabajo ponga más atención en lo que hace para evitar errores y detectar en qué parte del trabajo hay mayor propensión a generarlos y así ponerle más atención.

Configuración del sistema.

El *Formato de configuración del sistema* se agregó con la intención de que el integrador defina la configuración del sistema, una vez que esta seguro de que todo se integro correctamente, ya que es él, quien tiene toda la información referente a la configuración que debe existir para que el nuevo sistema se ejecute correctamente.

Informe de integración

El *Informe de integración* se agregó con la intención de que el administrador del proyecto sepa cual fue el resultado de la integración, además de servir como parte de la documentación del sistema o producto que se está generando.

Estándar de codificación.

El *Estándar de codificación* se agregó por que dentro del RUP existe una actividad para la definición de los estándares de codificación, sin embargo para la adaptación no era adecuado estar generando este artefacto cada que se comenzaba la implementación, ya que por lo regular siempre se usan los mismos lenguajes de programación, por lo que fue posible crear un estándar, el cual puede ser actualizado cada que sea necesario. En caso de

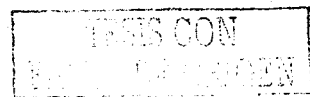


ser necesaria la actualización de éste estándar, por la utilización de un nuevo lenguaje de programación, el responsable de hacerlo es el Ingeniero de componentes, siguiendo la estructura establecida para el mismo.

Este capítulo sólo muestra la adaptación del flujo de trabajo Implementación, que finalmente sólo es uno de los tres flujos adaptados en esta tesis, por lo que es necesario leer el siguiente capítulo, para tener una idea más amplia del trabajo realizado.

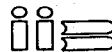
En el siguiente capítulo se habla de la adaptación del flujo de trabajo Administración del proyecto, el cual se encarga de supervisar la ejecución de todas las actividades del proceso de desarrollo de software, incluyendo las de implementación.

Artefactos de Implementación (anexos)



Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Formato de recopilación de información para el modelo de implementación.

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

(Llenar un formato por cada componente, en el lenguaje de programación)

Nombre del componente: _____

1. Proviene de la biblioteca de componentes (Llenar sólo si se toma de la biblioteca de componentes).

Nombre del componente original: _____

2. Componentes comerciales que necesita.

Nombre de función o librería	Descripción de funcionamiento:	Empresa desarrolladora o producto del que proviene

3. Descripción del componente.

Tipo de componente				
Clase	Base de Datos	Función	Procedimiento	Interfaz

Objetivo: _____

Componentes con los que se asocia para su funcionamiento	
Nombre del componente	Tipo de asociación

Datos que recibe		Datos que regresa	
Nombre	Tipo	Nombre	Tipo

Variables o atributos					
Nombre	Tipo	Alcance (global ó local)	Valor Inicial	validaciones	Rangos

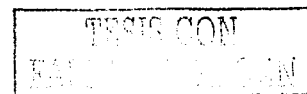
TRABAJA CON
FALLAS Y ORIGEN

Métodos (sólo para clases)				
Nombre del método	Datos que Recibe		Datos que regresa	
	Nombre	Tipo	Nombre	Tipo

Restricciones	
Validaciones	Variables globales (externas)

Código (Si se trata de un componente reutilizado, copiar aquí el código) (Si se trata de una clase duplicar esto para cada método de la clase)	
Nombre del método:	

Interfaz (en caso de no tener interfaz omita esta parte)	
Nombre de Interfaz:	



Modelo de Implementación.

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Componentes.

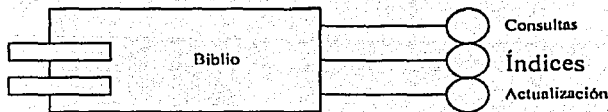
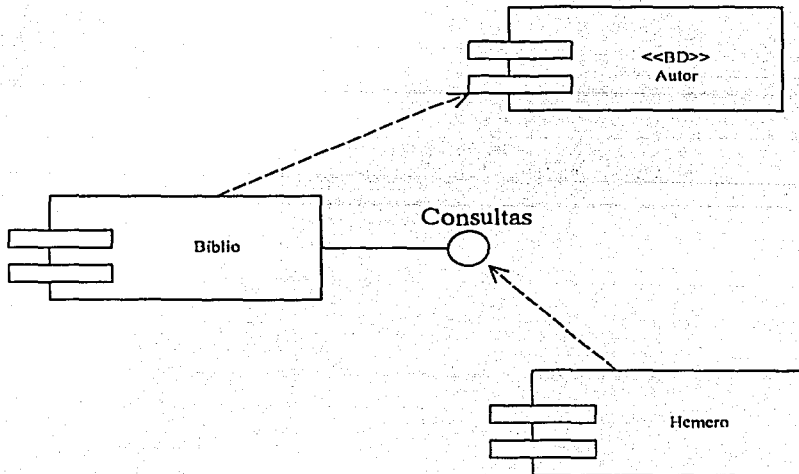
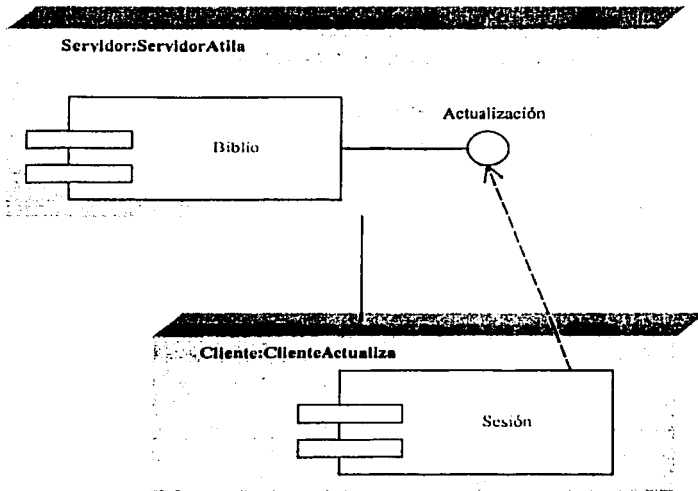


Diagrama de componentes.



Vista de despliegue.



Anexar: formato de recopilación de datos para el modelo de implementación, de cada componente.

38

TRABAJO CON
FALLAS

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



Plan de Integración.

Siglas del sistema _____ Ciclo _____
 Responsable _____ Fecha _____

1. Integración del Ciclo:

1.1. Lista de casos de uso: (establecer importancia con base a las dependencias funcionales)

Valor de importancia (1, 2, 3, etc.)	Siglas	Nombre de Módulo o Caso de Uso

1.2 Niveles de Integración (Indique más de uno por nivel, con base a su importancia):

Siglas de los casos de uso	#Nivel
	1
	2
	3
	4

1.3 Pruebas por nivel de integración (Definir con base al Plan de Pruebas).

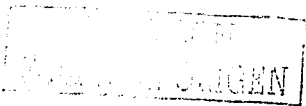
Siglas de los casos de uso	Nombre del(os) caso(s) de prueba	#Nivel
		1
		2
		3
		4

2. Integración de cada caso de uso (Definir esta sección para cada caso de uso definido anteriormente):

Nombre del caso de uso: _____

2.1 Lista de componentes del caso de uso:

Valor de importancia (1, 2, 3, etc.)	Siglas	Componente (clase, función, procedimiento)



39

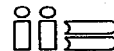
2.2 Niveles de Integración (Indique más de uno por nivel, con base a su importancia):

Siglas de los componentes	#Nivel
	1
	2
	3
	4

2.3 Pruebas por nivel de integración (Definir con base al Plan de Pruebas).

Siglas de los componentes	Nombre del(os) caso(s) de prueba	#Nivel
		1
		2
		3
		4

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



Formato de verificación de código.

Siglas del sistema _____ Ciclo _____
 Responsable _____ Fecha _____
 Siglas del caso de uso _____ Siglas del componente o sub-componente _____

(La revisión puede hacerse de dos formas, directamente sobre el código o sobre el formato. En caso de hacerse sobre el formato debe añadirse una impresión del código fuente al iniciar la verificación y únicamente marcar en forma independiente sobre el código cada objeto verificado)

1. Por cada variable realice el siguiente análisis:

Nombre de variable	Tipo (Si la verificación es sobre el código sólo marque la declaración de la variable)	Objetivo o utilidad (Si la verificación es sobre el código haga comentarios sobre el mismo)

(Si la verificación es sobre el código sólo subraye sobre el código y registre los errores sobre el formato)

Líneas en las que aparece la variable (verifique cada línea en la que aparece la variable e indique)		
No. línea (No se requiere cuando la verificación es sobre el código)	Estructura de control o instrucción (Asignación, condición, inicialización, cicloWhile, cicloFor, cicloDoWhile, incremento, decremento, comparación)	Error (Sólo si hay errores indique con una "x" para facilitar el conteo final)

Total de errores encontrados: _____

2. Por cada estructura de control realice el siguiente análisis:

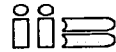
(Si la verificación es sobre el código sólo subraye sobre el código y registre los errores sobre el formato)

Tipo de estructura (condición, cicloWhile, cicloFor, cicloDoWhile)	No. de línea del inicio	No. de línea del fin	Error

Total de errores encontrados: _____

SE CON
 TALA DE ORIGEN

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



Formato de configuración del sistema.

Siglas del sistema _____ Ciclo _____
 Responsable _____ Fecha _____

Base de datos			
Servidor	Directorio	Nombre de la base de datos	Driver de conexión (ODBC, JDBC, etc)

Unidades de red		
Equipo al que debe conectarse	Directorio	Tipo de acceso (R/W)

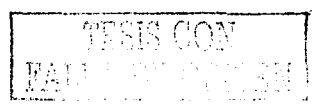
Software adicional.		
Nombre	Directorio de instalación	Configuración

Localización del sistema			
Unidad	Directorio	Archivo ejecutable	Subsistema relacionado

Estructura de directorios que componen el sistema		
Unidad	Directorio	Contenido

Archivos que componen el sistema			
Nombre del archivo	Tipo (txt, ini, dat, etc.)	Contenido	Estructura

42



Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Informe de Integración.

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Número del último nivel de integración alcanzado: _____

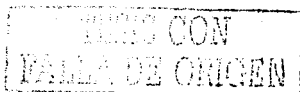
Nivel #	Se aprobaron las pruebas (Si, No)	Comentarios

Llenar sólo si no se aprobaron las pruebas.		
Nivel #	Caso de uso	Problemas

Anexar casos de prueba no superados y reporte de evaluación de pruebas.

Integrador

Firma





Estándar de Codificación.

(Este documento se irá modificando conforme sea necesario utilizar más lenguajes de programación)

Programación Visual Basic

1. Nombres de funciones y procedimientos:

Se llama procedimiento a los bloques de código que realizan una actividad específica y se agrupan bajo un nombre específico. Los nombres de procedimientos deberán comenzar con las letras "Pr" de Procedimiento seguidas por el nombre indicativo de la actividad de la función en infinitivo y el Sujeto que afecta, con la primera letra en mayúsculas y sin espacios, es decir, si la función imprime etiquetas, deberá llamarse:

PrImprimirEtiquetas

Se llama funciones a los procedimientos que al terminar su proceso, regresan un resultado o valor. Los nombres de funciones deberán comenzar con las letras "Fn" de función seguidas por el nombre indicativo de la actividad de la función en infinitivo y el Sujeto que afecta, con la primera letra en mayúsculas y sin espacios, es decir, si la función imprime etiquetas, deberá llamarse:

FnImprimirEtiquetas

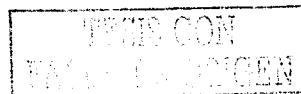
2. Nombres de variables

Las variables son aquellos espacios de memoria, que tienen asignado un nombre y permiten almacenar un tipo de datos específico. Los nombres de estas contendrán las letras "vr" seguida de la primera letra en mayúscula del tipo de dato que manejará (en inglés) y el nombre significativo del dato en el mundo real, comenzando con mayúscula cada palabra, es decir, si la variable es de tipo entero y es para guardar el precio neto de un producto, deberá llamarse:

vrIPrecioNeto

3. Nombres de parámetros

Un parámetro es un valor necesario para la ejecución de un proceso y es recibido del exterior en el momento en que se ejecuta. Estos deberán nombrarse iniciando con las letras "prm" , seguidos de la primera letra en mayúscula del tipo de dato que



manejará (en inglés) y el nombre significativo del dato en el mundo real, comenzando con mayúscula cada palabra, es decir, si el parámetro es de tipo cadena y contiene el nombre de un producto, deberá llamarse:

prmSNombreProducto

4. Nombres de objetos

Un objeto es la abstracción en la computadora de una entidad del mundo real, que cumple con las características de encapsulación, herencia y polimorfismo, que puede comunicarse con otros objetos a través de mensajes o eventos. Estos deberán nombrarse comenzando por las tres primera siglas del tipo de objeto en minúsculas, seguidas del nombre o actividad significativa en el mundo real, es decir si se tiene un objeto de tipo lista que contiene nombres de productos y un botón para guardar los nombres de estos productos, los llamaríamos:

lstNombreProductos
btnGuardaNombresProductos

5. Identación del código

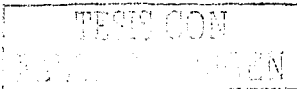
Identar es una técnica que permite organizar el código de forma que sea fácil de comprender a la vista del programador. Las reglas de indentación son las siguientes:

- Al escribir cada línea de código, en caso de pertenecer a un bloque, deberá escribirse con una sangría de tres caracteres a la izquierda, sin espacios entre renglones.
- Cada bloque deberá comenzar y terminar en la misma columna

Ejemplo:

Si tenemos una función imprime resultado cuando el valor es mayor a 5 y regresa el resultado.

```
FnImprimeResultado():Integer
    vrIResultado: Integer;
    if (vrIResultado>5) then
        print(vrIResultado);
    endif
    Return vrIResultado;
```



Capítulo 5. Adaptación del flujo de trabajo Administración del Proyecto del RUP al caso práctico.

Este capítulo a diferencia del anterior esta basado únicamente en el RUP, por lo tanto, para la adaptación sólo se hará referencia al mismo, ya que en la versión original del PU la administración del proyecto, únicamente se menciona como parte de otros flujos de trabajo.

La secuencia en la que se irá haciendo la descripción del trabajo de "adaptación" realizado, es similar a la del capítulo anterior, inicialmente se hará una pequeña descripción de lo que el RUP establece para este flujo de trabajo y a continuación comenzará la especificación de la adaptación.

La forma en que se hace la descripción del trabajo realizado para la adecuación de este flujo de trabajo es similar a la utilizada en el capítulo anterior, en donde hicimos una comparación entre lo establecido teóricamente y lo observado en el Departamento de Informática de IIB, para finalmente obtener el resultado de la adecuación.

Por último debe mencionarse que para la comparación entre lo establecido por el RUP y el resultado de la adecuación, al igual que en el capítulo anterior, se utilizaron los diagramas tal cual aparecen en el RUP, con la intención de mostrar que los cambios hechos no sólo están en la estructura de cada flujo de trabajo, sino que además en el hecho de crear un proceso descrito en nuestro propio idioma, el español.

5.1 El flujo de trabajo Administración del Proyecto de acuerdo al RUP

La Administración del proyecto como su nombre lo dice tiene como objetivo administrar la ejecución del proceso de desarrollo de software, lo cual involucra llevar un control del avance de la elaboración del producto, y que a su vez tiene que ver con la obtención de la certificación por parte del cliente, de los avances entregados.

Este flujo de trabajo tiene participación en todas las fases del ciclo de vida del desarrollo de software, pues sus actividades van desde la concepción del proyecto hasta la liberación y entrega del mismo.

El RUP marca tres propósitos para el flujo de trabajo Administración del Proyecto [Kruchten, 2000]:

- Proporcionar un marco de administración intensiva de proyectos de software.
- Proporcionar una guía práctica de planeación, ejecución, monitoreo y de los elementos humanos necesarios.
- Proporcionar una marco para la administración de riesgos.

Es importante mencionar que el tipo de Administración que establece el RUP, únicamente se enfoca en el proceso de desarrollo de software y no tiene nada que ver con la



administración de la organización desarrolladora de software. Por esa razón éste flujo de trabajo no incluye [Kruchten, 2000]:

- Administración de personal, que tiene que ver con la capacitación, contratación de personal para el equipo de trabajo.
- Administración del presupuesto, que está relacionado con la obtención de recursos para el desarrollo de proyectos, así como con la asignación de recursos económicos para cada uno de estos.
- Administración de contratos con proveedores y clientes, lo cual se refiere a la negociación que se lleva a cabo tanto con proveedores de herramientas, como con los clientes que solicitan un software, para establecer los costos de los productos.

A continuación se hará mención de la composición de cada uno de los elementos del proceso que conforman este flujo de trabajo.

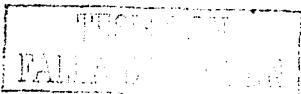
5.1.1 Actividades

Las actividades que conforman el flujo de trabajo Administración del proyecto están enfocadas en los aspectos específicos de un proceso de desarrollo iterativo, los cuales son [Kruchten, 2000]:

- La planeación iterativa del proyecto a través del ciclo de vida y la planeación particular de la iteración.
- La administración de riesgos
- El monitoreo del progreso de un proyecto iterativo.

Por lo anterior el RUP establece como actividades del flujo de trabajo Administración del Proyecto a las siguientes [Kruchten, 2000]:

- *Concebir el nuevo proyecto.* Se refiere a extraer la idea principal del proyecto de tal forma que esto sirva como punto de referencia para decidir entre continuar o abandonar el proyecto.
- *Evaluar los alcances y riesgos del proyecto.* Se intenta evaluar las capacidades y características previstas del proyecto y el riesgo asociado a cada una de estas.
- *Hacer un plan de desarrollo de software.* Este plan tiene la intención de crear la documentación que facilite y proporcione la información necesaria sobre el desarrollo del proyecto y que además sirva de base para el desarrollo del plan de iteraciones.
- *Planear la siguiente iteración.* El propósito de este plan es crear una guía detallada para la iteración que inicia. Éste nivel tiene como base el plan de desarrollo de software que se crea antes.
- *Monitorear y controlar el proyecto.* La intención de esta actividad es registrar continuamente los acontecimientos y el trabajo realizado durante el desarrollo

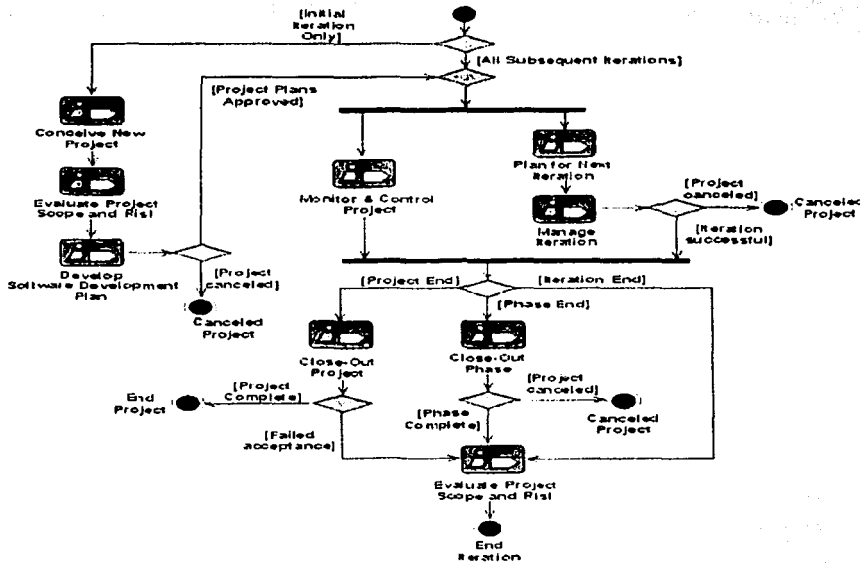


de software. Esto significa registrar los requerimientos de cambios obtenidos por la administración de cambios y programar cada uno de estos para la actual o siguiente iteración. Además del continuo registro de avances que regularmente se reporta a las autoridades correspondientes, esto también incluye el detalle de nuevas características y problemas descubiertos durante el desarrollo del proyecto.

- *Administrar la iteración.* Se refiere a la adquisición de los recursos necesarios para la iteración, y al almacenamiento continuo de los resultados de la misma, los que al final deberán poder ser consultados.
- *Cerrar la fase.* Es la actividad con la que se da por terminada una iteración y que establece qué características fueron cubiertas hasta el momento, el estado de todos los artefactos y el resultado de la distribución.
- *Cerrar el proyecto.* Marca el punto en el que el cliente finalmente recibe el producto de software y después de revisarlo lo acepta.

La secuencia de realización de las actividades establecida para este flujo de trabajo puede verse esquematizada en el diagrama de actividades (figura 1) que se muestra a continuación.

Figura 1. Diagrama tomado del flujo de trabajo Administración del proyecto en el RUP



5.1.2 Roles

El RUP establece dos roles en este flujo de trabajo: el *Administrador del proyecto* y el *Supervisor del proyecto*. En donde el primero es el responsable de ejecutar todas las actividades marcadas para este flujo de trabajo y el segundo, se encarga de hacer los monitoreos, que tienen como objetivo, revisar todos los artefactos que resultan del desarrollo del proyecto.

5.1.3 Artefactos

A continuación se hace una breve descripción de cada uno de los artefactos involucrados en el flujo de trabajo Administración del proyecto:

- *Modelo del negocio*. Establece los puntos de referencia necesarios para determinar la factibilidad de realizar un proyecto. El documento generado en esta actividad contiene una aproximación del costo económico que representa la realización de un proyecto, así como las características funcionales y no funcionales detectadas en los requerimientos.
- *Orden de trabajo*. Es el medio a través del cual el administrador del proyecto comunica a su equipo de trabajo las actividades y responsabilidades para cada uno de ellos.
- *Plan de desarrollo de software*. Esta compuesto por varios artefactos que juntos reúnen toda la información necesaria para la administración del proyecto, los cuales son: el Plan de aceptación del producto, Plan de administración de riesgos, Lista de riesgos, Plan de resolución de problemas y el Plan de métricas de evaluación.
 - *Plan de aceptación del producto*. Este plan describe cómo el cliente deberá de evaluar el producto para determinar, si lo que está recibiendo cumple con lo establecido por el mismo.
 - *Plan de administración de riesgos*. Este plan establece cómo deben manejarse los riesgos asociados al proyecto, detallando las actividades de la administración de riesgos.
 - *Lista de riesgos*. Menciona los riesgos en orden de importancia de manera decreciente, además de las actividades de contingencia específicas para cada uno.
 - *Plan de resolución de problemas*. Establece los lineamientos para reportar, analizar y resolver un problema ocurrido durante el desarrollo del proyecto.



- *Métricas de evaluación.* Define la importancia de cada uno de los alcances u objetivos del proyecto, así como las métricas empleadas para monitorear el progreso de éstos.
- *Plan de iteraciones.* Es un documento que contiene una descripción detallada de las actividades y artefactos producidos durante cada iteración, así como la dependencia entre cada uno de estos y los recursos necesarios para su realización.
- *Plan de aseguramiento de calidad.* Es un documento que contiene el cómo puede asegurarse que un artefacto cumpla con la calidad necesaria. Este plan no establece técnicas ni herramientas para la evaluación, simplemente establece cuáles son las características que deben de cumplir cada uno de los artefactos obtenidos durante el desarrollo.
- *Evaluación de iteración.* Contiene el resultado de una iteración, es decir el grado en el que fue alcanzado el objetivo de la misma, así como los cambios hechos a su contenido.
- *Evaluación del estado actual.* Obtiene periódicamente el grado de avance del proyecto en general. Esto produce la información histórica del proyecto.
- *Proyecto evaluado.* Es un repositorio que contiene la versión evaluada más reciente del proyecto.
- *Registro de revisión.* Contiene el resultado de la revisión efectuada a los artefactos del proyecto.

5.2 El flujo de trabajo Administración del Proyecto Adaptado al caso práctico

Después de hacer la revisión de lo que el RUP establece para el flujo de trabajo Administración del proyecto, fue necesario analizar las actividades que se realizan actualmente en el Departamento de Informática y que podrían tener relación con este flujo. Al hacer este análisis, se encontró que no existía ninguna actividad que tuviera la intención de administrar el proceso de desarrollo de software, ya que lo que realmente se supervisa actualmente es el cumplimiento del trabajo de cada persona, es decir que al responsable del departamento solamente le interesa conocer qué hace quién, más no cómo lo lleva a cabo, ya que no existe una forma de evaluar que lo hace de la forma correcta.

Por otra parte, el desarrollo completo de un proyecto se asigna a una sola persona, por lo que solamente esta persona conoce completamente la situación, avance, problemas, etc. de su proyecto y únicamente lo informa al jefe cuando se hace el informe anual del departamento dirigido al Director General del Instituto o en los casos en los que un problema debe ser resuelto directamente por el jefe.

Por otro lado cada persona al encargarse de un nuevo proyecto, sólo programa las entregas de avances, que en algunas ocasiones no son posibles por la falta de tiempo y la carga de trabajo. La forma en que se hace la planeación de actividades es totalmente empírica e informal, ya que no existe nada por escrito, únicamente se le informa al solicitante el día que deberá presentarse o en su caso se le llama en el momento para la entrega de avances.

Por lo anterior, en el momento de hacer la adecuación se establecieron parámetros para la selección de actividades a realizar, los cuales fueron:

- Seleccionar actividades que no entorpezcan el trabajo diario, incrementando demasiado la carga de trabajo.
- Seleccionar actividades que el RUP estableciera como de mayor importancia para la administración del proyecto.
- Tomar en cuenta que el Departamento de Informática no tiene y nunca tendrá la intención de evaluar los tiempos de desarrollo de los proyectos.
- Tomar en cuenta que los proyectos que el Departamento de Informática desarrolla no tienen mayores riesgos que aquellos que son inevitables y por tanto sin solución, como son los provocados por desastres naturales.
- Seleccionar las actividades que pueden ser generalizadas, pues el tipo de proyectos desarrollados por el Departamento de Informática tienen características muy comunes, y establecer estándares en su lugar.
- La calidad del software en el Departamento de Informática está dada por la satisfacción del solicitante.

Después de la explicación anterior es posible dar comienzo a la descripción y justificación del resultado de la adecuación.

5.2.1 Actividades

Las actividades que conforman el flujo de trabajo Administración del proyecto adecuado al caso real se ven en el diagrama de actividades de la figura 2.

Este diagrama contempla todas las actividades que el RUP establece, a excepción de la actividad *Evaluar riesgos...* que para este caso fue eliminada.

Sin embargo aunque a simple vista podría parecer que fue la única actividad eliminada durante la adecuación, lo cual resulta falso, más adelante se irán mencionando todos los elementos que fueron modificados durante la adecuación.

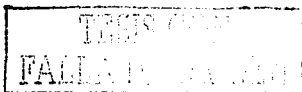
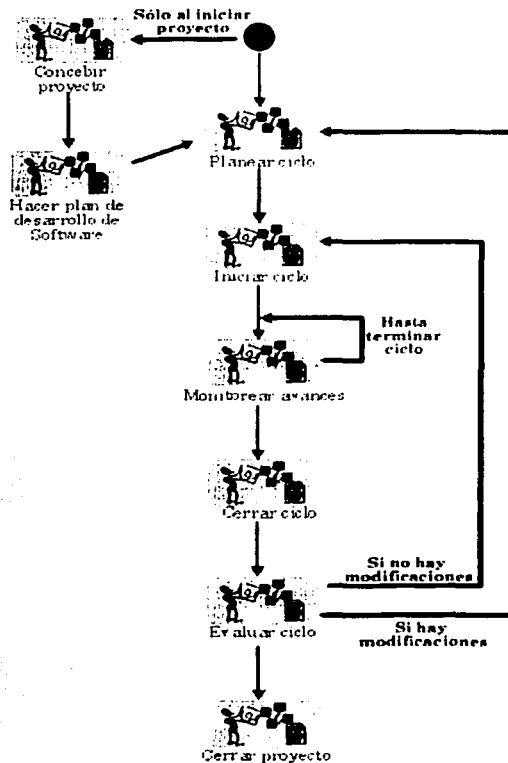


Figura 2. Diagrama del flujo de trabajo Administración del proyecto en la adecuación

Administración del proyecto



La tabla 1 muestra las diferencias que existen entre el flujo de trabajo adecuado al proceso de desarrollo de software al departamento de Informática del IIB y el flujo establecido en el RUP.

Primeramente podemos observar la ausencia de la primera actividad, que se hizo así debido a que los tipos de proyectos que realiza el Departamento de Informática no son de alto riesgo, por lo que no es necesario hacer una evaluación de este tipo.

Otro cambio importante es la actividad *Iniciar ciclo*, la cual no existe en el RUP y más adelante se justificara su creación.

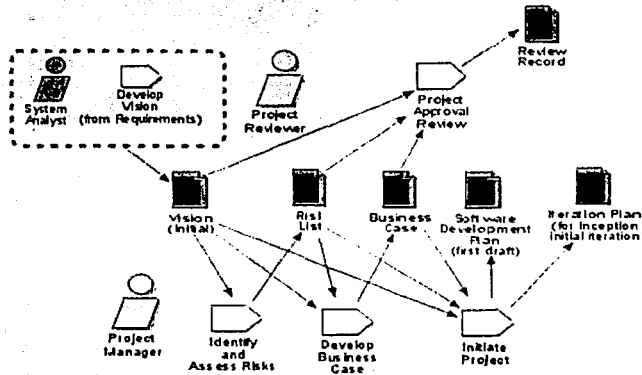
Adecuación	RUP
Concebir nuevo proyecto	Concebir el nuevo proyecto.
Hacer plan de desarrollo de software	Evaluar los alcances y riesgos del proyecto.
Planear ciclos	Hacer un plan de desarrollo de software.
Iniciar ciclo	Planear la siguiente iteración.
Monitorear avances	Administrar la iteración.
Evaluar ciclo	Monitorear y controlar el proyecto.
Cerrar ciclo	Administrar la iteración.
Cerrar proyecto	Cerrar la fase
	Cerrar el proyecto.

También son notables los cambios en los nombres de las actividades, lo cual se hizo en la mayoría de los casos con la intención de uniformar los nombres y hacer cada actividad más entendible, sin embargo se trató de conservar el nombre original siempre que fuera posible. A continuación se hará una descripción de los pasos que conforman cada una de las actividades de este flujo.

Concebir proyecto.

Al consultar el diagrama de actividades que el RUP presenta (ver figura 3), es imposible a simple vista darse una idea de lo que significa esta actividad, pues desafortunadamente no existe una descripción rápida de los pasos a seguir para concebir un nuevo proyecto.

Figura 3. Diagrama tomado de la actividad Concebir nuevo proyecto en el RUP



Por lo que fue necesario: consultar cuál era el contenido del artefacto generado en esta actividad y leer cuál era el objetivo marcado al concebir el proyecto. De ésta forma se generó el diagrama para esta actividad.

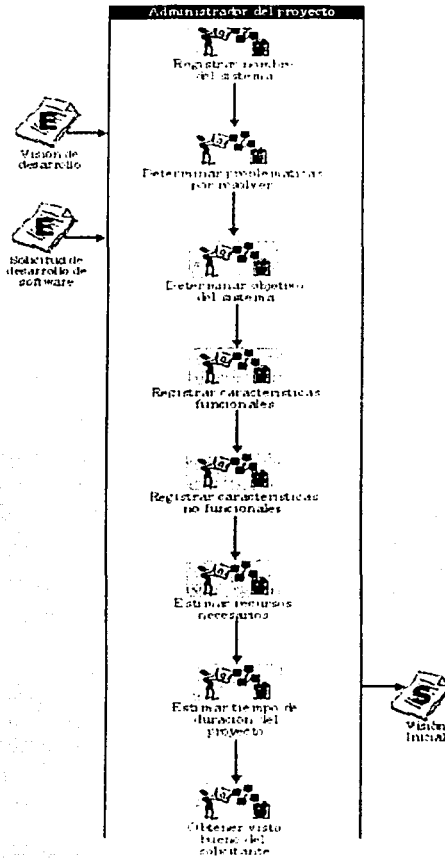
Fue así como se creó la primera actividad del flujo de trabajo Administración del proyecto (ver figura 4). Esta actividad sólo se realiza una vez al iniciar el proyecto y cumple en su mayoría con lo que establece al respecto el RUP. Sin embargo, en la

adecuación, la intención principal de esta actividad, no es determinar si el proyecto se realizará o no, como lo dice el RUP.

Esto se estableció así porque la mayor parte del trabajo de desarrollo de software, que se solicita al Departamento de Informática, siempre se realiza, a menos que exceda la capacidad del departamento, lo cual siempre es poco probable. Fue por eso que el principal objetivo de *Concebir proyecto* es crear un documento que sirva únicamente para formalizar el inicio de un proyecto, el cual se da, cuando los requerimientos del solicitante han sido recopilados por completo.

Figura 4. Diagrama de la actividad Concebir nuevo proyecto en la adecuación.

Concebir proyecto



Los pasos que se realizan al concebir un proyecto son:

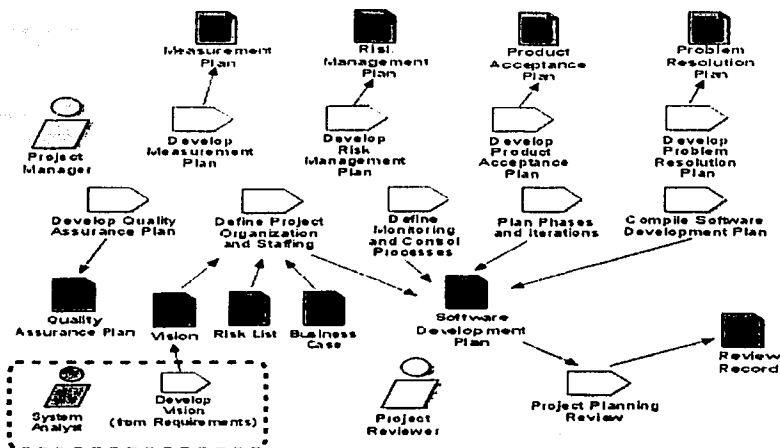
- *Registrar nombre del sistema.* Esta información se toma de la visión de desarrollo que se realiza en el flujo de trabajo de Requerimientos[Sánchez, 2003], está determinada por la funcionalidad del proyecto y la decisión del solicitante. Para el registro de la información de todos estos pasos se creó el *Formato de Visión Inicial*.
- *Determinar problemática por resolver.* Esta información se deduce a partir de la solicitud de desarrollo de software que el solicitante entrega al Departamento de Informática y la visión de desarrollo.
- *Determinar objetivo del sistema.* El objetivo se establece a partir de la problemática planteada anteriormente y la *Visión de Desarrollo*.
- *Registrar características funcionales.* Estas son tomadas de la *Visión de Desarrollo*.
- *Registrar características no funcionales.* Al igual que las características funcionales, deben tomarse de la *Visión de Desarrollo*.
- *Estimar recursos necesarios.* Se refiere a establecer los recursos que el departamento de Informática proporciona para el proyecto y a los recursos que el solicitante deberá de facilitar. Esta información se obtiene del análisis de la *Visión de Desarrollo, el Estándar de Herramientas y el Estándar de Distribución*.
- *Estimar tiempo de duración del proyecto.* Esta estimación se hace con base a la dimensión del proyecto, la experiencia del equipo de trabajo y la carga de trabajo del Departamento de Informática, en el momento de la solicitud. Se recomienda hacer una estimación por fases, lo que significa establecer un tiempo determinado para cada fase del ciclo de desarrollo del proyecto, así como para cada ciclo. Esta estimación se hace con dos intenciones, por un lado ir adquiriendo experiencia en la estimación de tiempos; que seguramente en las primeras veces no se hará apropiadamente; y por el otro, para dar una idea al solicitante, de la complejidad del trabajo. Para esto se creó una sección en el formato de *Visión Inicial*, que permite calendarizar cada actividad del proceso.
- *Obtener visto bueno del solicitante.* Se refiere a llevar el documento obtenido en esta actividad para que el solicitante lo firme, y se quede con una copia del mismo.

Hacer plan de desarrollo de software.

La segunda actividad es *Hacer plan de desarrollo de software*, y se refiere a la creación del artefacto del mismo nombre, que está compuesto a su vez de otros artefactos, es por eso que los pasos de esta actividad se refieren básicamente a la creación de los artefactos contenidos en el Plan de desarrollo de software. El RUP esquematiza esta actividad con un diagrama de actividades (ver figura 5) en el se muestra la realización de cada uno de los artefactos que componen el Plan de Desarrollo de Software.



Figura 5. Diagrama tomado de la actividad Hacer Plan de desarrollo de software en el RUP



Hacer el plan de desarrollo de Software, al igual que la actividad anterior se realiza sólo una vez al inicio del proyecto, sin embargo a diferencia de la primera actividad, ésta fue generalizada para todos los proyectos, pues se crearon estándares para cada uno de los artefactos que componen éste plan, de tal forma que el plan de desarrollo de un nuevo proyecto, está conformado por los estándares y la Visión Inicial, la cual incluye las cosas consideradas para cada caso particular.

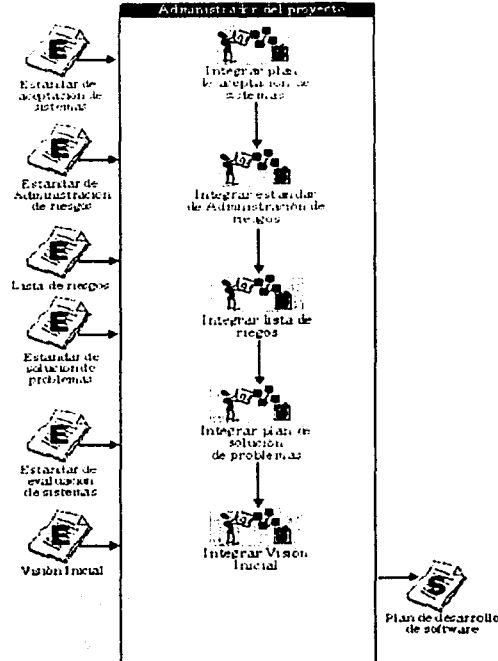
El diagrama de actividades (ver figura 6) de ésta actividad muestra una secuencia en la ejecución de los pasos, sin embargo, ésta secuencia es sólo un ejemplo del orden en el que se pueden ir integrando cada elemento al Plan de desarrollo de software, por lo que el orden es poco relevante, ya que lo importante es que todo esté en un solo documento, sin importar que fue lo que se hizo primero.

Cada actividad marcada como *Integrar...* se refiere a tomar una copia del estándar respectivo y ponerla dentro del Plan de desarrollo de software, con las adecuaciones que pudieran surgir, por el tipo de proyecto que se trate en el momento.

Es importante mencionar, que en ésta segunda actividad, el Plan de desarrollo de Software aún no está completo, pues le falta el Plan de ciclos, y éste se crea en la siguiente actividad.

Figura 6. Diagrama de la actividad Hacer plan de desarrollo de software en la adecuación

Hacer plan de desarrollo de software



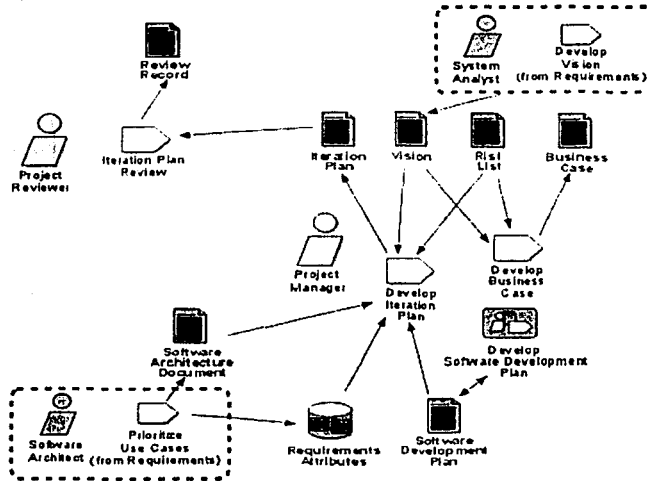
Planear ciclo.

La actividad *Planear ciclo* tiene características muy especiales, pues se realiza tantas veces cuantos ciclos haya. Siempre que inicia un nuevo ciclo se va modificando el Plan de ciclos, que es el objetivo de esta actividad y va almacenando la historia del desarrollo del proyecto, por ciclos.

Esta actividad se llama en el RUP *Planear la siguiente Iteración* y se refiere a ir detallando lo que se realizará en la siguiente etapa del proyecto, cada que inicia una nueva etapa (ver figura 7). Sin embargo, el diagrama de actividades que se muestra en el RUP, no muestra a diferencia de la actividad anterior, que se debe de hacer específicamente. Por eso fue necesario hacer un análisis del contenido del Plan de Iteraciones que el RUP maneja, y de los documentos que son necesarios para esta actividad, pues de esa manera fue posible crear un diagrama de actividades que detallara los pasos a seguir para realizar el Plan de ciclos, que como se mencionó, es el objetivo de esta actividad.



Figura 7. Diagrama tomado de la actividad Planear ciclo en el RUP.

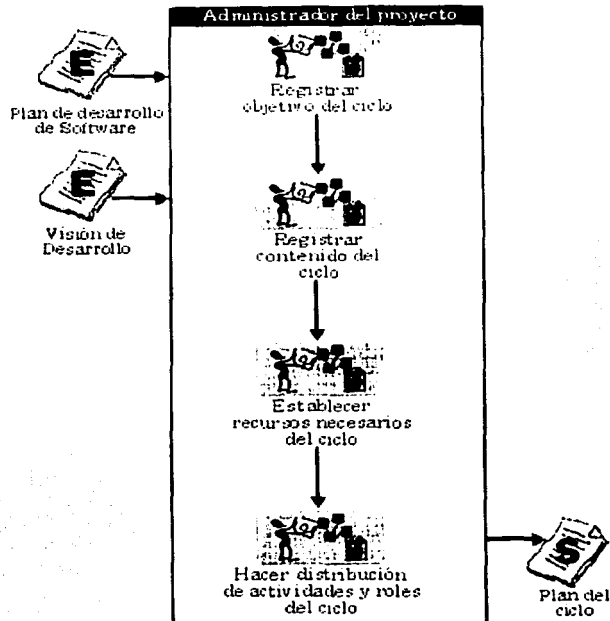


Una vez realizado éste análisis se obtuvo el diagrama de actividades (ver figura 8) de *Planear Ciclo* del flujo de trabajo Administración del Proyecto adecuado al departamento de Informática del IIB que se compone de los pasos:

- *Registrar objetivo del ciclo.* En este paso se toma el Informe del ciclo anterior, en caso de que exista, para ver si no hubo alguna modificación por el ciclo anterior, (que esta contenido en el Plan de Ciclos del Plan de Desarrollo de Software) y la *Visión de Desarrollo*, donde está determinado el número de ciclos. Toda la información de estos pasos se registra en el *Formato del Plan del ciclo*, que al ser llenado se convierte en el *Plan del ciclo*.
- *Registrar contenido del ciclo.* Esta información se toma de la *Visión de Desarrollo*, con base a la información registrada en el paso anterior. En este paso también se marcan las fechas de entrega, de cada caso de uso, por lo que es necesario consultar la *Visión Inicial*.
- *Establecer recursos necesarios del ciclo.* Esta información se desglosa a partir de la *Visión Inicial* del proyecto, pues ahí están definidos los recursos que se requieren para el software.
- *Hacer la distribución de actividades y roles del ciclo.* Con base al equipo de trabajo que se contempló dentro de la *Visión Inicial*, se llenan sólo las dos primeras secciones del *Formato de asignación de trabajo*, pues después al iniciar el ciclo se completa el formato que se entregará a cada persona.

Figura 8. Diagrama de la actividad Planear ciclo en la adecuación.

Planear ciclo

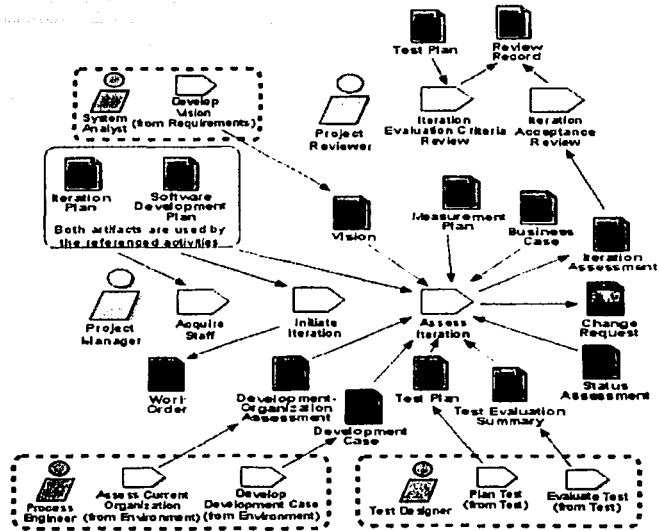


Iniciar ciclo y Evaluar ciclo

Las actividades *Iniciar ciclo* y *Evaluar ciclo* surgieron de la actividad que el RUP llama *Administrar iteración*. En dónde la intención es llevar un control del proyecto en desarrollo, esto puede verse en el diagrama de actividades (ver figura 9) que el RUP presenta para la Administración de la iteración. En este diagrama se observan más actividades además del inicio y la evaluación del ciclo, las cuales se refieren al trabajo diario del Administrador del proyecto, en donde continuamente se le da seguimiento. Sin embargo, esto no es posible hacerlo en el Departamento de Informática del IIB, pues es tan poco el personal que trabaja ahí, que el administrador en la mayoría de los casos es la misma persona que el Ingeniero de componentes.

Fue por la falta de recursos humanos que la Administración del proyecto se redujo al Inicio y evaluación de cada ciclo, sin embargo no se paso por alto por completo, ya que existen monitoreos.

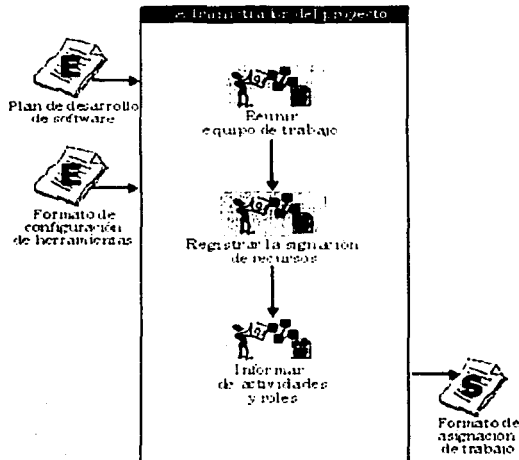
Figura 9. Diagrama tomado de la actividad administrar iteración en el RUP



El inicio de ciclo por tanto tiene como objetivo autorizar los recursos al equipo de trabajo para el inicio de ciclo, así como, informar de las responsabilidades que cada miembro del equipo tiene para este ciclo. Esto es más claro en diagrama de actividades de *Iniciar ciclo* que resultó de la adecuación (ver figura 10), en donde se ven los siguientes pasos:

Figura 10. Diagrama de la actividad Iniciar ciclo en la adecuación.

Iniciar ciclo

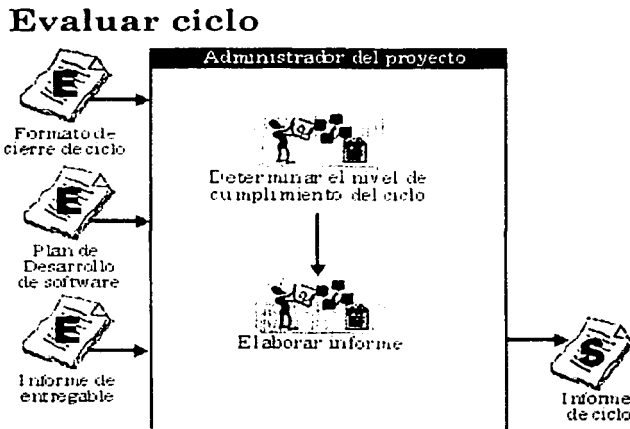


- *Reunir equipo de trabajo.* Se refiere a convocar a una reunión, a los seleccionados para participar en el ciclo.
- *Registrar la asignación de recursos.* El Administrador deberá completar el formato de asignación de trabajo, asignando a cada integrante, los permisos registrados en el formato de configuración de herramientas y firmar el formato, antes de entregarlo a cada participante del proyecto.
- *Informar de actividades y roles.* Durante la reunión el Administrador del proyecto deberá entregar el formato de asignación de trabajo para informar de las actividades a cada uno de los integrantes del equipo.

La otra actividad que surge de la administración del ciclo es *Evaluar ciclo*, la cual se realiza después de cerrar el ciclo, para el caso de esta adaptación. Existe un diagrama (ver figura 11) para la descripción de sus pasos:

- *Determinar el nivel de cumplimiento del ciclo.* Se refiere a comparar el contenido del *formato de cierre de ciclo* y lo planeado en *Plan de ciclos* que se encuentra dentro del *Plan de desarrollo de software*, y el resultado del *Informe del entregable*, registrando el resultado de la comparación en el *Formato para el Informe de ciclo*, el cual al estar lleno se convierte en el *Informe de ciclo*.
- *Elaborar informe.* Una vez completado el Informe de ciclo, se anexa al Plan del ciclo.

Figura 11. Diagrama de la actividad Evaluar ciclo en la adecuación.



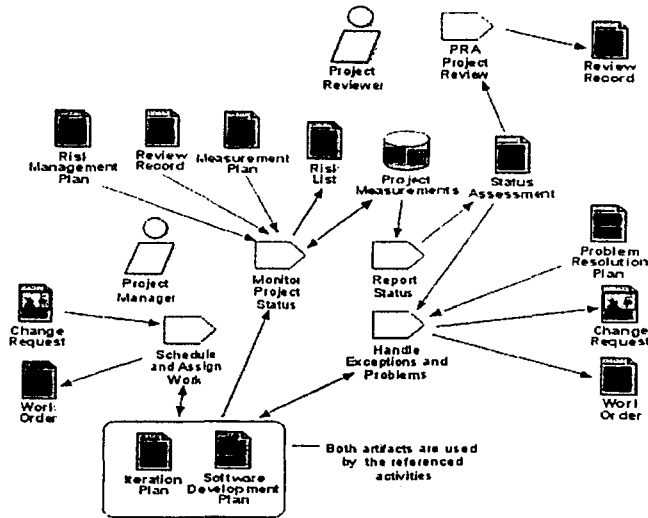
Por último cabe mencionar que el resultado de esta actividad se ve registrado en el Informe de ciclo, el cual forma parte del Plan del ciclo.



Monitorear avances

Monitorear avances es una actividad muy importante, y es por eso que el RUP lo maneja fuera de la Administración del ciclo, pues involucra varios pasos, que permiten llevar un control del grado de avance de la fase actual del proyecto, durante el proceso de desarrollo.

Figura 12. Diagrama tomado de la actividad Monitorear avances en el RUP.

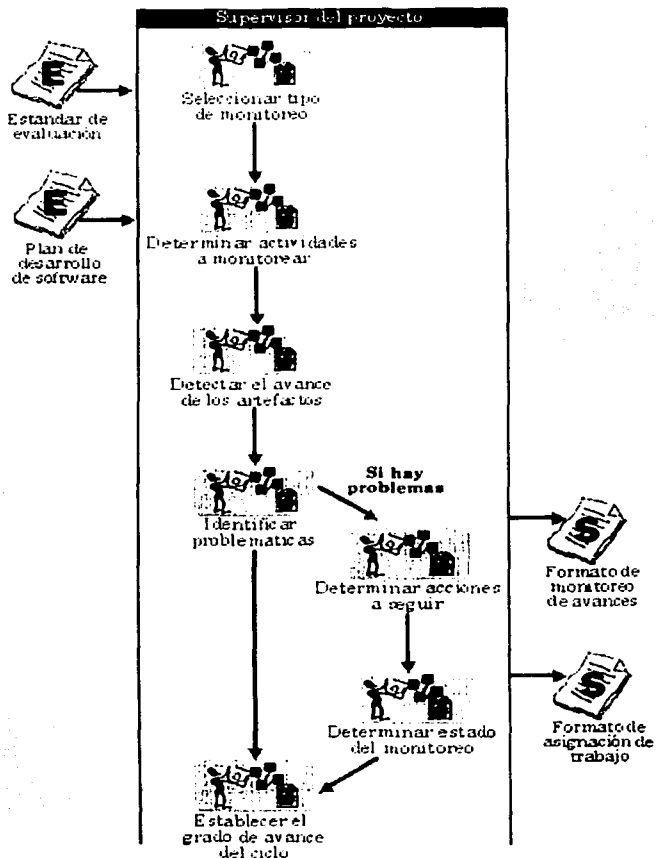


Los pasos establecidos por el RUP para esta actividad pueden verse en el diagrama de actividades de la figura 12, en donde no sólo se muestra la importancia sobre el control del avance del proyecto, además se puede ver la relación tan estrecha que existe entre la administración del proyecto y la administración de cambios a los requerimientos, de la cual se hablará ampliamente en el siguiente capítulo.

La actividad de monitoreo es especial, pues su ejecución requiere de una calendarización estricta, la cual para el caso de esta adaptación, se encuentra programada en el Estándar de Evaluación, del cual se hablará más adelante. Por lo anterior, en ésta sección únicamente se hablará de los pasos a seguir al realizar el monitoreo, los cuales se ven en el diagrama de actividades de la figura 13 y son:

Figura 13. Diagrama de la actividad Monitoreo de avances en la adecuación.

Monitorear avances



- **Seleccionar tipo de monitoreo.** Este se selecciona en el *Estándar de Evaluación* que se encuentra en el Plan de Desarrollo de Software, pues dependiendo del grado de avance del proceso, es el tipo o nivel de monitoreo que será necesario.
- **Determinar actividades a monitorear.** Esta información, al igual que la anterior, se toma del mismo documento y el mismo estándar, y se registra en la sección uno del *Formato de monitoreo de avances*.

TESIS CON
FOLIO DE ORIGEN

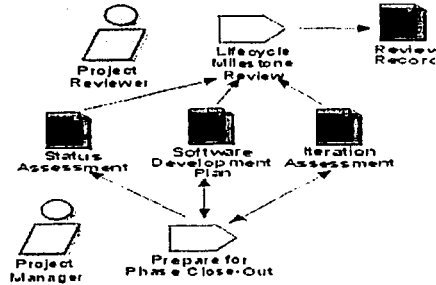
- *Detectar el avance de los artefactos.* Esto se refiere a revisar físicamente cada artefacto y determinar cual es su grado de avance, esto se hace con base al contenido de cada artefacto, ya que para cada uno de éstos se ha establecido puntualmente qué información deben de contener, por lo que es posible determinar el avance, comparando los elementos terminados entre los pendientes, de cada documento. Para esto se diseñó la sección dos del *Formato de monitoreo de avances*.
- *Identificar problemáticas.* Esto se hace por lo regular, cuando el avance del proyecto se ve retrasado, y es con la intención de conocer y reportar o registrar, cuáles son las razones por las que no se ha alcanzado el progreso esperado del proyecto. Las problemáticas se especifican tanto al monitorear el avance de las actividades, como el avance de los artefactos.
- *Determinar acciones a seguir.* Este paso se refiere a tomar las decisiones correspondientes para la solución de los problemas que causan retraso, para esto puede ser útil consultar el Estándar de Solución de problemas, que está en el Plan de Desarrollo de software y en caso de ser un problema especial, convocar a una reunión con el equipo de trabajo para establecer una solución. Toda determinación deberá comunicarse a la persona indicada, a través del formato de asignación de trabajo del cual se hará una copia para el formato de monitoreo de avances.
- *Determinar estado del monitoreo.* Esto es para dejar por escrito cualquier comentario al respecto del monitoreo realizado, sólo cuando se han encontrado problemas y determinados soluciones. Un ejemplo de lo que puede escribirse es: "*queda por confirmar que las acciones se hicieron y los problemas se resolvieron, por lo que será necesario ejecutarse después, otro tipo de monitoreo para confirmar que se hayan solucionado los problemas*". O en el caso de ser un monitoreo de tipo *Rectificación de problemas* (ver Estándar de Evaluación), entonces debe indicarse cual es el resultado de la rectificación. Esta información se escribe dentro de la sección de observaciones del *Formato de monitoreo de avances*.
- *Establecer el grado de avance del ciclo.* Una vez realizado el análisis de cada actividad y artefacto que se están evaluando, debe establecer el grado de avance de todo el proyecto logrado hasta el momento de la evaluación, el cual debe determinarse tomando en cuenta lo planeado versus lo obtenido hasta el momento del monitoreo. Este paso deberá realizarse independientemente de que la solución a los problemas haya dado resultado positivo, pues el resultado de las acciones tomadas se examina con otro tipo de monitoreo.

Por último es importante mencionar que los tipos de monitoreo se explican en el Estándar de evaluación, el cual puede ser consultado al final del capítulo. Por lo que sólo queda por comentar, que toda esta información debe quedar registrada en el *Formato de monitoreo de avances*, del que se hablará más adelante.

Cerrar ciclo

La penúltima actividad de éste flujo de trabajo es *Cerrar ciclo* la cual es en el Proceso Unificado *Cerrar fase* y se hace con la finalidad de marcar el fin de una etapa del desarrollo, que en este caso, cada etapa es un ciclo. El diagrama que el RUP establece para esta actividad se ve en la figura 14, sin embargo, no muestra qué pasos deben realizarse. Por lo tanto fue necesario hacer un análisis de los objetivos y los artefactos que intervienen al cerrar el ciclo.

Figura 14. Diagrama de la actividad Cerrar fase en el RUP.



El cierre de ciclo es especial, ya que el momento en el que sucede esta actividad, puede estar marcado por el tiempo determinado para el ciclo o por el cumplimiento de los objetivos del ciclo, que se establecen en el Plan de ciclos. Sin embargo, para el caso de ésta adecuación, se ha establecido que el cierre de ciclo se hará en el momento en que se cumpla con lo planeado en el plan del ciclo, de tal forma que los tiempos estimados, únicamente sirvan como parámetro, que permita mejorar la estimación de tiempos para los consecutivos ciclos ó proyectos. Esto con la intención de adquirir experiencia en la estimación de tiempos para el tipo de proyectos que se realizan en el Departamento de Informática y así, en un futuro hacer el cierre de ciclo en el tiempo estimado y entonces, hacer evaluaciones del desarrollo de proyectos, de acuerdo al cumplimiento de los tiempos establecidos.

Por lo anterior, los pasos que conforman la actividad Cerrar ciclo(ver figura 15), van encaminados únicamente a marcar el fin de las etapas o ciclos del desarrollo de un producto, en el momento en que se han cumplido los objetivos planteados, sin importar, por el momento, el tiempo empleado para el ciclo. Los pasos de esta actividad para la adecuación son:

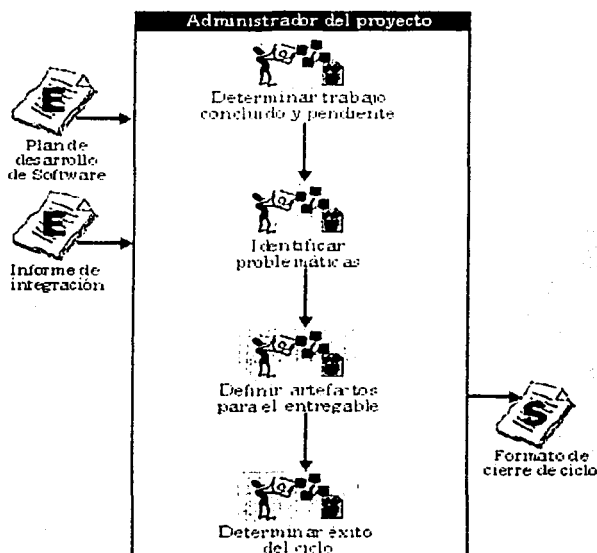
- *Determinar trabajo concluido y pendiente.* Se refiere a listar los artefactos que se debían realizar durante el ciclo (ver Plan de ciclos), e ir determinando el grado de avance obtenido de cada uno, Toda la información generada en este paso y los pasos siguientes, se registra en el *Formato de cierre de ciclo*.



- *Identificar problemáticas.* Esto se hace en caso de no haber cubierto del todo algún artefacto, y se refiere a justificar o determinar las razones por las que no se obtuvo, tal artefacto.
- *Definir artefactos para el entregable.* Se refiere a establecer qué elementos conformarán la entrega al cliente, esto con base en lo establecido en el *Plan del ciclo* y el *Informe de Integración*.
- *Determinar éxito del ciclo.* Es necesario escribir en forma resumida cuáles fueron los logros obtenidos en el ciclo y qué cosas quedaron pendientes. De tal forma que los pendientes se tomen en cuenta para la planeación del siguiente ciclo.

Figura 15. Diagrama de la actividad Cerrar ciclo en la adecuación.

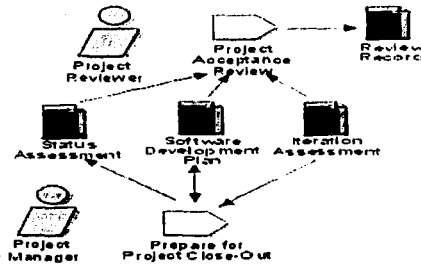
Cerrar ciclo



Cerrar proyecto

La última actividad de este flujo es *Cerrar proyecto*, y al igual que la primera actividad, ésta sólo se realiza una vez durante todo el desarrollo del proyecto, pues marca el momento en que se da por terminado todo el producto. La forma en que el RUP la muestra es, al igual que la actividad anterior, poco clara, ya que el diagrama (ver figura 16) no proporciona mayor información de lo que debe hacerse en el cierre del proyecto. El cierre del proyecto es muy parecido al cierre de ciclo, sólo que a diferencia del cierre de ciclo, este sólo se hace una vez, por lo que el producto final debe cumplir con todo lo establecido en un principio, pues aquí no es posible decir que lo que quede pendiente se hace en el siguiente ciclo.

Figura 16. Diagrama tomada de la actividad Cerrar proyecto en el RUP.



Por lo anterior esta actividad se realiza después de cerrar el último ciclo planeado, en donde se realizaron los últimos requerimientos solicitados. Con esta actividad se da por terminado el desarrollo del proyecto, por lo que al llegar aquí, no hay marcha atrás, pues lo que sigue es la liberación, en donde se hacen todos los preparativos para la entrega formal al cliente, del producto obtenido que ya no sufrirá ningún cambio.

Por lo anterior, el cierre del proyecto en la adecuación, al igual que en el RUP, tiene como finalidad marcar el término del proyecto para dar pauta a la entrega formal del desarrollo, pues actualmente esto es un gran problema en el Departamento de Informática, ya que ninguno de los desarrollos que se llevan a cabo se han liberado formalmente, pues desde el inicio de los mismos, no existe ningún plan que marque los alcances o en su caso el momento en el que se da por terminado un trabajo.

Los pasos que componen el Cierre del proyecto se ven claramente en el diagrama de actividades (ver figura 17) para la adecuación y son tres:

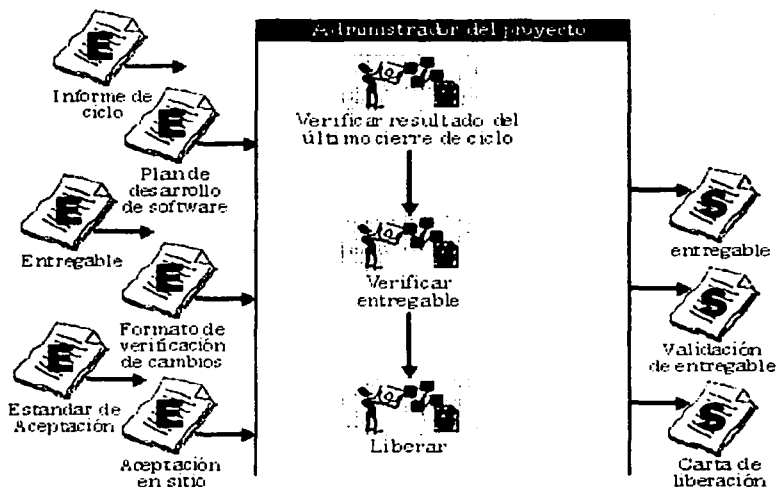
- *Verificar resultado del último cierre de ciclo.* Este paso se realiza con el *último cierre de ciclo que se encuentra dentro del Plan de Desarrollo de Software y el último monitoreo de cambios*, por lo que debe verificarse que todos los pendientes se hayan resuelto, y hacer el cierre del proyecto hasta resolver el último pendiente.
- *Verificar entregable.* Esto se refiere a tomar el *Estándar de aceptación del Plan de Desarrollo de Software y el Informe del último ciclo* para verificar que lo que se va a entregar esta completo y cumple con lo establecido en los diversos estándares para la elaboración de artefactos. Esta revisión se hace sobre el formato electrónico, por lo que sólo se genera una copia impresa, hasta después de haber validado el entregable.



- **Liberar.** Para esto el Administrador del proyecto autoriza la impresión de todos los manuales, una vez impresos se hace la entrega, y el solicitante firma la carta de liberación. Esto sólo se hace para que el Departamento de Informática pueda comprobar que el trabajo se ha realizado.

Figura 17. Diagrama de la actividad Cerrar proyecto en la adecuación.

Cerrar proyecto



5.2.2 Roles

Para la adecuación se decidió conservar los dos roles que se establecen en el RUP para el flujo de trabajo Administración del Proyecto: el *Administrador del proyecto* y el *Supervisor del proyecto*. En dónde, como se mencionó al principio, el *Administrador del proyecto* ejecuta casi todas las actividades marcadas para este flujo de trabajo, a excepción de los monitoreos, que son realizados por el *Supervisor del proyecto*. Sin embargo, como cualquier otro rol, éstos pueden ser ejecutados por una misma persona.

5.2.3 Artefactos

Visión Inicial.

Este artefacto contiene el resultado inicial del análisis de los requerimientos, el cual detalla las características funcionales y no funcionales del sistema. Este análisis es descrito por el Administrador del Proyecto en forma entendible para el cliente, de tal manera que ésta es la primera vista de lo que será el nuevo sistema y por lo tanto deberá contener el visto bueno del cliente, pues de esta forma se indica que el proyecto puede iniciarse. Este

documento fue creado como un formato, pues intenta dar una idea clara de su contenido, además de establecer un estándar para todos los proyectos.

Estándar de aceptación.

El *Estándar de aceptación* de sistemas define cuáles serán los parámetros para evaluar un producto, de tal forma que este resulte aceptable. Este artefacto es el resultado de la adecuación de lo que el RUP define como Plan de aceptación del producto. Este documento se convirtió en un estándar, por que al hacer su análisis se encontró que la mayoría de los proyectos realizados por el Departamento de Informática del IIB, tenían características en común, por lo que era posible establecer un conjunto de parámetros estandarizado para la evaluación de los productos generados por el departamento.

Estándar de solución de problemas.

El *Estándar de solución de problemas* deriva del Plan de resolución de problemas, pues al igual que el documento anterior, al hacer su estudio se encontró que en la mayoría de los desarrollos de sistemas, de los problemas que surgen son similares. Este estándar esta clasificado por tipo de problema y contiene una lista de procedimientos que deben seguirse para la solución de los mismos. Este documento debe anexarse al Plan de Desarrollo de Software de cada proyecto, sin embargo no debe duplicarse cada vez que se hace un nuevo software, únicamente debe hacerse referencia a él, y en caso de que algún problema no se considere dentro del estándar, deberá anexarse siguiendo el formato establecido para el mismo.

Estándar de evaluación.

Este documento es muy importante, pues establece los diferentes momentos y elementos en que deben ser evaluados los sistemas, esto incluye los monitoreos, pues dentro de este estándar se establece el momento en el que debe hacerse cada monitoreo, de acuerdo a su tipo, así como los elementos que se deben evaluar en cada uno y, las herramientas que deben utilizarse para estas revisiones. La programación de los monitoreos se hizo identificando puntos críticos en el Proceso de Desarrollo de Software, los cuales se determinaron con base a dependencias entre actividades, es decir, fue necesario evaluar cuál era el impacto de un error en un punto del proceso, hacia las actividades subsecuentes para determinar si era necesario incorporar un monitoreo en ese punto. Existen varios tipos de monitoreos, los que se hacen durante el Proceso de Desarrollo de Software que pueden ser de revisión o de seguimiento, los que se hacen al terminar cada ciclo y los que se hacen al terminar el proyecto. Este estándar al igual que los dos anteriores, fue resultado de la adecuación ya que el RUP realmente lo maneja como Plan de evaluación.

Estándar de Administración de riesgos.

Este artefacto establece un conjunto de procedimientos que deben seguirse para manejar los riesgos que existen al Desarrollar Sistemas en el Departamento de Informática del IIB, que en realidad son pocos y se encuentran mencionados en la Lista de riesgos.

Lista de riesgos.

La *Lista de riesgos* es un documento en el cual se encuentran mencionados y clasificados los riesgos de acuerdo a su tipo.

Plan de desarrollo de software.

Para el *Plan de desarrollo de software* no existe un formato pues este documento es el resultado de la unión de todos los artefactos mencionados antes, además del plan de ciclos y el formato de cierre de proyecto que se detallan más adelante.

Plan de ciclos.

Este artefacto también tiene un formato que define los puntos que deben detallarse y que son necesarios para dar inicio al ciclo. Este Plan se va conformando conforme da inicio cada ciclo, pues al iniciar la última iteración es posible decir que se tiene el plan de ciclos completo. Este plan contienen el Formato de asignación de trabajo, el Formato de cierre de ciclo, el Formato de monitoreo de avances y el Informe de cierre de ciclo.

Formato de asignación de trabajo.

Este formato permite asignar claramente las tareas o responsabilidades a cada persona, para cada ciclo, es por eso que debe incluirse la copia de cada asignación de trabajo durante el ciclo, en el Plan del ciclo.

Formato de cierre de ciclo.

El *Formato de cierre de ciclo* es un documento que contienen el detalle de lo que se obtuvo durante el ciclo, en donde se especifican artefactos, problemas o comentarios al respecto. Este artefacto únicamente lista lo que se esperaba y lo que se obtuvo y en caso de existir algún problema, detalla cual es el problema, pues este formato sirve para realizar el Informe final del ciclo.

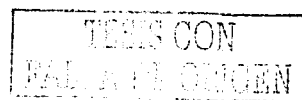
Informe de ciclo.

Este documento es el resultado de la evaluación de ciclo y contiene las alternativas para la solución de problemas en caso de que haya problemas o de lo contrario una oración en la que se resume el resultado del ciclo.

Formato de monitoreo de avances.

El contenido de este documento varía de acuerdo al tipo de monitoreo que se realiza, pues no siempre se revisan los mismos elementos, por lo que al diseñar el formato, se especificó qué elementos deben ser llenados de acuerdo a lo que se revisa.

Formato de validación del entregable.



Contienen los puntos a validar de un entregable, este formato forma parte del estándar de aceptación y esta descrito como una herramienta en el estándar de evaluación.

Carta de liberación.

Este es el último documento que conforma el Plan de Desarrollo del proyecto, pues contiene el resultado final del proyecto, así como la fiema de conformidad del cliente, es importante mencionar que este documento se hace después de cerrar el último ciclo, por lo que este documento únicamente sirve como respaldo del fin del proyecto.

La administración del proyecto es una actividad muy importante, por el control que ejerce sobre el proceso de desarrollo de software, sin embargo existe otro tipo de control que también es muy importante, y debe ejercerse sobre el resultado final del software que se entrega, y para eso esta la Administración de la configuración y el control de cambios, que es el flujo de trabajo a tratar en el siguiente capítulo.

TESIS CON
FALLA DE ORIGEN

Artefactos de Administración del proyecto (anexos)

TESIS CON
FALLA DE LA UCA

Visión Inicial.

Nombre de Software (tomar de la visión de desarrollo): _____

Problemática a Resolver (tomar de la visión de desarrollo): _____

Objetivo del Sistema (tomar de la visión de desarrollo): _____

Módulos de los que se compondrá el sistema (tomar de la visión de desarrollo):

Nombre	Descripción	Información que maneja	Usuarios	Verbo

Interacción con otros sistemas (tomar de la visión de desarrollo):

Nombre del Sistema	Detalle

Características no funcionales:

1. Plataforma.

UNIX MSDOS WINDOWS MAC

2. Ambiente.

REDLOCAL MONOUSARIO INTRANET INTERNET

Recursos necesarios.

1. Personal del departamento de Informática y de fuera que participa.

Nombre	Rol
	Administrador de cambios



	Administrador de configuración
	Administrador de distribución
	Administrador del proyecto
	Analista de sistemas
	Arquitecto de software
	Diseñador de bases de datos
	Diseñador de pruebas
	Escritor técnico
	Especialista en herramientas
	Especificador de casos de uso
	Ingeniero de casos de uso
	Ingeniero de componentes
	Integrador
	Probador
	Usuario responsable
	Supervisor del proyecto

2. Software y Hardware necesario:

Nombre	Version	Actividad en la que se usará	Responsable de su adquisición

3. Capacitación:

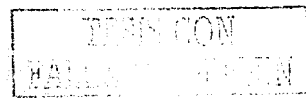
Nombre	Version	Lugar	Duración

4. Estimación de tiempos (ver formato anexo "gannt.xls").

Después de leer cuidadosamente y aclarar cualquier duda que pudiera surgir, referente al software solicitado al Departamento de Informática, acepto todo lo establecido anteriormente como parte de mi solicitud, además de aceptar todas las responsabilidades que esto conlleva. Quedando conforme respecto a los tiempos de entrega y en el entendido de que el Departamento de Informática se compromete a cumplir con lo establecido asignando el apoyo y los recursos que por su parte son necesarios.

Nombre y firma Coordinador
Nombre y firma del solicitante
Nombre y firma del administrador del proyecto

74



Estándar de aceptación.

Introducción.

Este documento determina los lineamientos a seguir, del departamento de Informática del Instituto de Investigaciones Bibliográficas, para evaluar y determinar si el producto de software final, llamado "entregable", es aceptable y por lo tanto, puede entregarse al solicitante dando lugar a la liberación del proyecto.

Los estándares de documentación, elaboración de entregable, diseño y codificación, forman parte importante en el proceso de evaluación establecido en éste estándar, ya que ayudan a determinar el grado de aceptación del entregable.

Alcance.

El tipo de evaluación planteado para los sistemas desarrollados dentro del departamento de Informática, se enfocan principalmente en la parte funcional de los mismos, ya que el objetivo primordial al hacer este tipo de evaluación es confirmar que al solicitante se le hará entrega del producto que solicitó y de los elementos que le permitan hacer uso del mismo por lo cual existen algunos puntos de evaluación que toman en cuenta otros aspectos.

Responsabilidades.

⇒ Solicitante.

Es responsabilidad del solicitante cumplir con todas las actividades y necesidades planteadas por el departamento de informática para el desarrollo de su aplicación, ya que en el momento de hacer una evaluación debe contarse con toda la información proporcionada por el mismo, ya que de lo contrario no se tendrían puntos de referencia para saber si el producto obtenido cumple con lo planteado por el solicitante, y en ese caso no es posible realizar ningún tipo de evaluación, por lo que el producto se vuelve automáticamente aceptable para el Departamento de Informática.

⇒ Departamento de informática.

Es responsabilidad del Departamento de Informática especificar por escrito y aclarar todas las dudas y requerimientos que el solicitante recibirá en su producto final, ya que al hacer la evaluación del software, éste debe cumplir con todos los requerimientos planteados por el solicitante, de lo contrario el software automáticamente se vuelve inaceptable.

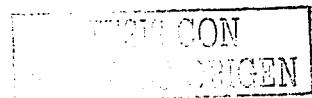


Criterios de Evaluación.

1. El manual de usuario de un sistema se considera aceptable, si permite que un usuario involucrado en los procesos que abarca el software y con los mínimos conocimientos en computación que se requieren, utilice el sistema sin la ayuda de alguien más. Además de cumplir con lo establecido en el Estándar de Distribución.
2. El manual técnico de un sistema se considera aceptable, si cumple con lo establecido en el estándar para la elaboración del entregable y contempla todos artefactos finales de los flujos de trabajo: Requerimientos, Análisis, Diseño, Implementación y Pruebas; creados bajo los estándares de documentación, diseño y codificación, establecidos para el Departamento de Informática. Por lo tanto, la información contenida en éste manual debe ser idéntica a la información de los artefactos recabados por el Administrador de versiones en la última versión del sistema.
3. El manual de instalación de un sistema se considera aceptable, si contiene toda la información referente a la instalación del software, de tal forma que el usuario con los conocimientos de computación básicos necesarios, pueda realizar la instalación, sin la ayuda de alguien más.
4. El manual de configuración de un sistema se considera aceptable, si contiene toda la información referente a la configuración, de tal forma que el usuario con los conocimientos de computación básicos necesarios, pueda realizar la configuración, sin la ayuda de alguien más.
5. Todos los documentos que tienen que ver con requerimientos y cambios del sistema deberán contener la firma de aprobación del solicitante del mismo, así como la del responsable del departamento de informática.
6. Todas las pruebas planeadas como parte del plan de pruebas del sistema deberán realizarse nuevamente en presencia del solicitante y resultar aprobatorias.
7. Debe existir un paquete de instalación en un medio de almacenamiento conveniente para el solicitante.
8. Un sistema se considera aceptable si aprueba todos los puntos establecidos en este documento.

Proceso de evaluación.

Este proceso se registra a través del formato de verificación de entregable, por lo que el proceso de evaluación esta basado en el llenado del mismo formato.



Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Formato de validación de entregable.

Siglas del sistema _____

Ciclo _____

Responsable _____

Fecha _____

1. Manual de usuario (Marque con una "x", los puntos que se cumplen).

- Cumple con los márgenes establecidos
- Cumple con los tipos y tamaños de letra establecidos.
- Al seleccionar uno de los procesos descrito dentro del manual de usuario, fue posible entenderlo y ejecutarlo, sin la ayuda de nadie.

2. Manual técnico (Marque con una "x", los puntos que se cumplen).

- Cumple con los márgenes establecidos
- Cumple con los tipos y tamaños de letra establecidos.
- Al seleccionar uno de los documentos que componen el manual y comparar el documento correspondiente de la última versión almacenada en el repositorio de versiones, eran idénticos.

3. Manual de Instalación (Marque con una "x", los puntos que se cumplen).

- Cumple con los márgenes establecidos
- Cumple con los tipos y tamaños de letra establecidos.
- Al seguir al pie de la letra lo indicado en el manual para la instalación, fue posible entenderlo e instalar el software, sin la ayuda de nadie.

4. Manual de Configuración (Marque con una "x", los puntos que se cumplen).

- Cumple con los márgenes establecidos
- Cumple con los tipos y tamaños de letra establecidos.
- Al seguir al pie de la letra lo indicado en el manual para la configuración, fue posible entenderlo y configurar el software, sin la ayuda de nadie.



5. Documentación de Requerimientos y cambios al sistema (Marque con una "x", los puntos que se cumplen).

- Al revisar todos los documentos de requerimientos que necesitan de la autorización del solicitante, no se encontró ninguna omisión de firma.
- Al revisar las verificaciones de cambios aceptados, no se encontró ninguna observación pendiente.
- Al revisar los resultados de todos los monitoreos, no se encontró ninguna observación pendiente.

6. Funcionalidad del software (Marque con una "x", los puntos que se cumplen)..

- Al hacer la revisión de las pruebas del sistema en desarrollo y en producción, no se encontró ningún problema pendiente.
- Al seleccionar uno de los procesos descrito dentro del manual de usuario, fue posible ejecutarlo, sin ningún problema.

7. Empaquetado (Marque con una "x", los puntos que se cumplen)..

- Al instalar el sistema en el equipo del solicitante, con las características necesarias para la ejecución del sistema, no hubo problemas, por el medio de almacenamiento en el que se encontraba el paquete de instalación.

8. Resultado (Marque con una "x", los puntos que se cumplen)..

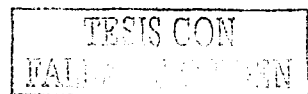
Existe alguna opción de las anteriores que no se haya cumplido:

SI	NO
<input type="checkbox"/>	<input type="checkbox"/>

SI la respuesta de la pregunta 8 es SI, el producto no es aceptable.

**Firma del
Administrador del
proyecto**

1.78



Estándar de solución de problemas.

Introducción.

Este documento determina los lineamientos a seguir en el departamento de Informática del Instituto de Investigaciones Bibliográficas, para reportar, analizar y resolver los problemas durante el proceso de desarrollo de software.

¿Qué es un problema?

Un problema es una situación que detiene la ejecución normal de un proceso, que por lo regular siempre tiene una solución que permite terminar la ejecución del proceso aunque no sea de manera normal.

Tipos de problemas de acuerdo a su importancia.

- ✓ Problemas con el solicitante.
- ✓ Problemas de recursos.
- ✓ Problemas de conocimientos.
- ✓ Problemas personales.

Proceso para reportar un problema de acuerdo al tipo de problema.

La notificación de todos los problemas deberá hacerse con el formato para registro de problemas.

- ✓ Problemas con el solicitante.
 - El responsable del proyecto deberá hablar con el solicitante para informarle que hay un problema con alguna de las personas o en su caso con él mismo, explicando claramente cuál es la situación problemática.
 - El responsable del proyecto deberá hablar con el Jefe del departamento de Informática para dar a conocer la problemática y a su vez solicitar una reunión entre el solicitante, el jefe del departamento y el responsable del proyecto.
- ✓ Problemas de recursos.
 - Informar al Jefe del departamento de Informática y al solicitante, del recurso que tiene problemas.

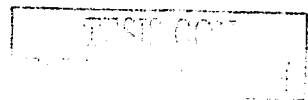


- ✓ Problemas de conocimientos.
 - Informar al jefe del departamento del tipo de conocimiento que falta en el equipo de trabajo del proyecto. Este tipo de problemas *sólo pueden reportarse durante la fase de iniciación*, pues es en ese momento, en el cual pueden tomarse decisiones al respecto.
- ✓ Problemas personales.
 - Si los problemas personales (salud, familiares, etc.) tienen repercusión en los tiempos de entrega prometidos al solicitante, el responsable del proyecto deberá reportar al Jefe del departamento, el problema.

Solución de problemas de acuerdo a su tipo.

El registro de la solución de un problema se hace en el mismo formato para registro de problemas, con la intención de tener un documento por escrito de las decisiones tomadas, en donde las tres partes involucradas firmen de conformidad (jefe del departamento, responsable de informática y solicitante). Todo problema debe ser reportado en el momento en el que sucede, de lo contrario toda la responsabilidad recae sobre el responsable del proyecto.

- ✓ Problemas con el solicitante.
 - Para la reunión establecida para dar solución al problema, el responsable del proyecto deberá llevar un conjunto de soluciones posibles que previamente fueron discutidas con el jefe del departamento.
 - El Jefe del departamento dará a conocer la problemática al solicitante en donde al solicitante, se le pedirá una solución al problema, en caso de no haber respuesta por parte del solicitante, el jefe del departamento expondrá las propuestas de solución.
 - Deberán llegar a un acuerdo entre las tres partes involucradas, el cual debe escribirse en el formato mencionado, y enseguida, firmar de conformidad la solución acordada.
 - En caso de no haber respuesta por parte del solicitante, entre el Jefe del departamento y el responsable del proyecto podrían decidir la suspensión del proyecto e informar al secretario académico de la solución tomada.
- ✓ Problemas de recursos.
 - En caso de ser un recurso que depende del Departamento de Informática el jefe del departamento determinará la solución. Si se trata de un recurso que no es posible conseguir, el responsable del proyecto deberá hacer una evaluación de la posibilidad de sustituir el recurso por otro, y de los cambios implicados. Finalmente el usuario deberá ser informado en un plazo no mayor a 5 días hábiles.
 - En caso de ser un recurso que depende del solicitante, el solicitante deberá dar respuesta en un plazo no mayor a 2 días hábiles, ya que en caso de no conseguir el recurso el responsable del proyecto deberá evaluar el impacto y dar la solución en un plazo no mayor a 3 días hábiles.



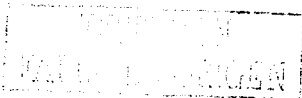
✓ Problemas de conocimientos.

- Con base a las deficiencias encontradas es importante considerar tiempo de capacitación dentro de los tiempos estimados para todo el proyecto, por lo que deberá incluirse dentro de la visión inicial en el rubro de capacitación.

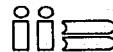
✓ Problemas personales.

- Si es necesario deberá sustituirse al elemento de equipo en forma temporal para evitar el retraso, en caso de no ser posible la sustitución, el jefe del departamento deberá informar al solicitante del posible retraso, mínimo con un día de anticipación. En caso de ser algo repentino el jefe del departamento o en su caso el responsable del proyecto tiene la obligación de localizar el mismo día al solicitante para informar de problema.

Recae la responsabilidad sobre el responsable del proyecto de cualquier problema no reportado a tiempo.



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



Formato para el registro de problemas.

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Tipo de problema:

- Problemas con el solicitante. Problemas de recursos. Problemas de conocimiento. Problemas personales.

Descripción del problema: _____

Firma del jefe del Departamento: _____

Fecha de recepción: _____

Fecha de reunión con el solicitante: _____

Solución: _____

**Firma del
Administrador del
proyecto**

**Firma del
Coordinador**



Estándar de evaluación.

Introducción.

Este documento determina los lineamientos a seguir, del Departamento de Informática del Instituto de Investigaciones Bibliográficas, para evaluar. Además de establecer el momento en el que se debe ejecutar una evaluación(monitoreo), de acuerdo a las actividades del proceso de desarrollo de software.

Los estándares de documentación, elaboración de entregable, diseño y codificación, forman parte importante en el proceso de monitoreo establecido en éste estándar, ya que ayudan a determinar el grado de aceptación de los artefactos.

Alcance.

El Departamento de Informática del Instituto de Investigaciones Bibliográficas estableció como objetivo de los monitoreos, la verificación del cumplimiento de las normas establecidas para la elaboración de cada artefacto. Ya que por el momento no se tienen los recursos para hacer una verificación de contenido. Sin embargo hay monitoreos que no sólo ponen su atención en la estructura, también verifican el avance del proceso, lo cual permite saber, si se cumple con los tiempos establecidos.

Tipos de verificaciones.

Existen básicamente dos tipos de verificación: el monitoreo y la evaluación, las cuales se diferencian por su naturaleza.

Tipos de monitoreo			
Nombre	Objetivo	Descripción	Rol
Tipo1	Monitorear avances	Determinar el grado de avance de los artefactos, actividades o ciclos; y verificar el cumplimiento del estándar de elaboración.	Administrador del proyecto
Tipo2	Monitorear configuración (rutina)	Verificar que haya cambios estructurales en el proyecto, que no se vean reflejados en la documentación.	Administrador de configuración
Tipo3	Validación entregable	Verificar que el entregable contenga todos sus elementos y que éstos cumplan con el estándar de elaboración.	Administrador del proyecto
Tipo4	Monitoreo de configuración (petición de cambio aceptada)	Este tipo de evaluación se realiza para determinar los flujos que son impactados por un cambio, por lo que sólo se realiza en el momento que una solicitud de cambio es aceptada.	Administrador de configuración
Tipo5	Rectificación de problemas	Se refiere a ejecutar nuevamente una verificación para confirmar que los problemas encontrados en el último monitoreo, fueron resueltos.	Administrador del proyecto
Tipo6	Monitoreo de cambios	Este tipo de monitoreo surge por que se tiene una reunión con el usuario, por lo que puede surgir una solicitud de cambios.	Administrador de cambios

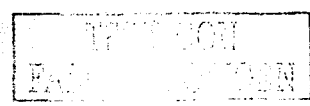
Tipos de evaluación			
Nombre	Objetivo	Descripción	Rol
Tipo1	Evaluar ciclo	Se refiere a determinar el resultado de un ciclo después del cierre.	Administrador del proyecto
Tipo2	Evaluar resultado del último cierre de ciclo	Se refiere a determinar el resultado final del proyecto, ya que se realiza pues de evaluar el último ciclo, como parte del cierre del proyecto.	Administrador del proyecto

Elementos a verificar.

Tipo	Fase	Actividad en la que se realiza	Monitoreo Descripción
Tipo1	Cualquiera	Cualquiera	Revisar el ciclo hasta el momento del monitoreo
Tipo1	Elaboración	Elaborar modelo de análisis	Revisar el avance del ciclo, es decir que en la fecha marcada como fin de la actividad, verificar que se haya terminado.
Tipo1	Elaboración	Elaborar modelo de análisis	Revisar el avance de los artefactos generados, para la construcción del Modelo de análisis.
Tipo2	Elaboración	Elaborar modelo de análisis	Comparar Visión de desarrollo con Modelo de análisis, y guardar una copia de éste.
Tipo5	Elaboración	Elaborar modelo de análisis	Verificar que el cambio se vea reflejado en el Modelo de análisis.
Tipo1	Elaboración	Elaborar modelo de diseño	Revisar el avance del ciclo, es decir que en la fecha marcada como fin de la actividad, verificar que se haya terminado.
Tipo1	Elaboración	Elaborar modelo de diseño	Revisar avance de los artefactos que componen el Modelo de diseño.
Tipo2	Elaboración	Elaborar modelo de diseño	Comparar Modelo de análisis con Modelo de diseño y guardar una copia de éste.
Tipo5	Elaboración	Elaborar modelo de diseño	Verificar que el cambio se vea reflejado en el Modelo de diseño.
Tipo1	Construcción	Diseñar Interfaz	Revisar el avance del ciclo, es decir que en la fecha marcada como fin de la actividad, verificar que se haya terminado.
Tipo6	Construcción	Diseñar Interfaz	Verificar con el usuario que la interfaz cumple con sus necesidades.
Tipo5	Construcción	Diseñar Interfaz	Verificar que el cambio se vea reflejado en la interfaz
Tipo2	Construcción	Diseñar Interfaz	Comparar Diseño de interfaz con Modelo de diseño.
Tipo2	Construcción	Hacer plan de distribución, Crear plan de integración	Recolectar versiones de todos los artefactos generados hasta el momento.
Tipo1	Construcción	Hacer plan de distribución, Crear plan de integración	Revisar el avance del ciclo, es decir que en la fecha marcada como fin de la actividad, verificar que se haya terminado.
Tipo1	Construcción	Integrar ciclo	Revisar el avance del ciclo, es decir que en la fecha marcada como fin de la actividad, verificar que se haya terminado.
Tipo5	Transición	Evaluar pruebas	Verificar que el cambio se vea reflejado en los artefactos impactados.
Tipo2	Transición	Cerrar ciclo	Recolectar versiones de todos los artefactos generados hasta el momento.
Tipo1	Transición	Cerrar ciclo	Revisar el avance del ciclo, es decir que en la fecha marcada como fin de la actividad, verificar que se haya terminado.
Tipo3	Transición	Realizar pruebas de aceptación en desarrollo	Verificar que el sistema cumpla con todas las pruebas planeadas
Tipo3	Transición	Realizar pruebas de aceptación en sitio	Verificar con el usuario que el sistema cumple con todo lo establecido.
Tipo3	Transición	Producir entregable	Verificar entregable de acuerdo al estándar de aceptación.
Tipo5	Transición	Evaluar ciclo	Verificar que el cambio se vea reflejado en los artefactos impactados.
Tipo2	Transición	Cerrar proyecto	Recolectar carta de liberación y los últimos artefactos generados.

La mayoría de las verificaciones están programadas dentro del proceso de desarrollo de software por lo que los artefactos a verificar dependen de la actividad en la que se realiza el monitoreo o validación, sin embargo puede darse la necesidad de hacer una verificación durante la ejecución de una actividades no programada, y es por eso que en este documento fueron incluidos todos los artefactos que deben ser revisados de acuerdo a la actividad y la programación.

CF 84



La programación de una verificación puede ser de tres tipos:

- "SI" programada. Se encuentra marcada dentro del proceso de desarrollo de software.
- "NO" programada. Se realiza en forma aleatoria sólo para reforzar la verificación.
- "EMERGENTE". Se realiza a causa de una solicitud de cambio aceptada.

VERIFICACION	ACTIVIDAD	ARTIFACTO
SI	Elaborar modelo de análisis	Modelo de análisis
SI	Elaborar modelo de diseño	Modelo de diseño
SI	Integrar ciclo	Componentes
SI	Hacer pruebas del sistema.	Casos de prueba
SI	Cerrar ciclo	Cierre del ciclo
SI	Realizar pruebas de aceptación en desarrollo.	Casos de prueba, Informe de las pruebas
SI	Realizar pruebas de aceptación en sitio.	Casos de prueba, Informe de las pruebas
SI	Cerrar ciclo.	Informe de ciclo
SI	Cerrar proyecto.	Carta de liberación
NO	CUALQUIER ACTIVIDAD*	Se verifican la estructura sólo de la lista de los artefactos generados en la actividad con respecto al estándar de elaboración correspondiente al artefacto.
RECTIFICACION	CUALQUIER ACTIVIDAD	Se verifican nuevamente los mismos artefactos del monitoreo que resulto con fallas.
EMERGENTE	CUALQUIER ACTIVIDAD	Se utilizan todos los artefactos generados hasta la actividad en la que surgió el cambio

* Para determinar los artefactos consultar lista de actividades y artefactos.

Instrumentos para la verificación.

- Estándar de elaboración.** Se le llama así el conjunto de estándares que tienen como objetivo fijar los parámetros para la elaboración de artefactos, que en este caso son: estándar de documentación, estándar de elaboración de entregable, estándar de diseño y estándar de codificación.
- Formato de monitoreo.** Existen seis formatos diferentes, contruidos de acuerdo a los tipos de monitoreo y evaluación que existen.

Nombre de formato	Tipo de monitoreo y evaluación
Formato de monitoreo de avances	Monitoreo Tipo1
Formato de monitoreo de configuración	Monitoreo Tipo2
Formato de validación del entregable	Monitoreo Tipo3
SE UTILIZA NUEVAMENTE EL FORMATO DEL MONITOREO A RECTIFICAR	Monitoreo Tipo4
Formato de Informe de ciclo	Evaluación tipo1
Formato de Cierre del proyecto	Evaluación tipo2
Formato de Verificación de cambios.	Evaluación tipo3

- Última versión almacenada. Se refiere al conjunto de artefactos de un proyecto, que contienen los cambios más recientes. Que se encuentran recopilados por el Administrador de versiones en el repositorio de versiones.
- Lista de actividades y artefactos. Es una relación de todos los artefactos generados por actividad, ordenado por fases.

Anexo Lista de actividades, artefactos y roles del proceso de desarrollo de software.
Anexo Formatos de monitoreo.

8 86



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



Lista de actividades, artefactos y roles.

1. Artefactos.

NOMBRE	ASIGLAS
Arquitectura	Arqui
Carta de liberación	Cacep
Casos de prueba	Caspr
Cierre del proyecto	Ciepr
Clases del análisis	Claan
Clases del diseño	Clads
Componentes	Compn
Configuración de herramientas	Conhe
Descripción del patrón de arquitectura	DptrA
Descripción de casos de uso	Descu
Diagramas de casos de uso	Diacu
Diseño de arquitectura	DisAr
Diagrama Entidad Relación	DiaER
Entregable	Entre
Estándar de administración de riesgos	EstAR
Estándar de Aceptación	EstAp
Estándar de administración de cambios	EstAC
Estándar de análisis y diseño	EstAD
Estándar de codificación	EstCd
Estándar de configuración	EstCo
Estándar de documentación	EstDo
Estándar de evaluación	EstEv
Estándar de pruebas	EstPr
Estándar de base de datos	EstBD
Estándar de requerimientos	EstRe
Estándar de herramientas	EstHe
Estándar de resolución de problemas	EstRP
Formato de asignación de trabajo	EstAT
Formato de diseño de base de datos	FrmBD
Formato de cierre de ciclo	FrmCP
Formato de distribución	FrmDs
Formato de monitoreo	FrmMn
Formato de preguntas	FrmPr
Formato de prioridad de casos de uso	FrmPC
Formato de requerimientos	FrmRq
Formato de validación de entregable	FrmVE
Herramientas	Herra
Informe de ciclo	InfCL
Informe de configuración	InfCo
Informe de entregable	InfEn
Informe de integración	InfIn
Interfaz	Intfz
Invitación de pruebas formales	InvPF

RECIBIDO
 1998
 10/11/98

87

Lista de herramientas del ciclo	ListH
Manual de configuración	MnlCo
Manual de instalación	MnlIn
Manual de usuario	MnlUs
Manual técnico	MnlTe
Modelo de análisis	ModAn
Modelo de casos de uso	ModCU
Modelo de diseño	ModDs
Modelo de Implementación	ModIm
Notificación de configuración	NotCo
Paquetes del análisis	PaqAn
Plan de ciclos	PlaCl
Plan de Desarrollo de Software	PlaDS
Plan de integración	PlaIn
Plan de pruebas	PlaPr
Realización de casos de uso del diseño	ReaCU
Realización de casos de uso del análisis	ReaCA
Solicitud de cambio	SolcC
Solicitud de desarrollo de software	SolcD
Subsistemas de diseño	SubDs
Subsistemas o sistema	SubSs
Visión de desarrollo	VisDs
Visión Inicial	VisIn

1788

TRABAJO CON
FALLA DE ORIGEN

2. Actividades.

NOMBRE	SIGLAS
Abrir registro de cambios	ARC
Abrir registro de versiones	ARV
Analizar casos de uso	ACU
Analizar clases	ACL
Analizar paquetes	APQ
Cerrar ciclo	CRC
Cerrar proyecto	CRP
Cambiar y actualizar configuración	CBC
Concebir proyecto	CPY
Conocer el ambiente para el proyecto	CAP
Crear ambiente de configuración	CAC
Crear plan de integración	CPI
Detallar casos de uso	DTC
Diseñar arquitectura	DAQ
Diseñar base de datos	DBD
Diseñar casos de uso	DCU
Diseñar clases	DCL
Diseñar interfaz	DIT
Diseñar subsistemas	DSB
Distribuir informe de configuración	DIC
Ejecutar pruebas del sistema	EPS
Elaborar el diagrama entidad relación	EDR
Elaborar modelo de análisis	EMA
Elaborar modelo de diseño	EMD
Establecer arquitectura	EAR
Establecer modelo de implementación	EMI
Estructurar modelo de casos de uso	EMC
Evaluar las pruebas	EVP
Evaluar ciclo	EVC
Generar visión de desarrollo	GVD
Hacer el diseño de pruebas	HDP
Hacer manuales	HMN
Hacer plan de desarrollo de software	PDS
Hacer plan de distribución	HPD
Hacer el plan de pruebas	HPP
Identificar actores y casos de uso	IAC
Implementar componentes	IMC
Iniciar ciclo	INC
Integrar ciclo	ITC
Monitoreos	MNT
Planear ciclo	PLC
Preparar ambiente para un ciclo	PAC
Priorizar casos de uso	PCU
Producir entregable	PEN
Realizar pruebas de aceptación en desarrollo	PAD
Realizar pruebas de aceptación en sitio	PAS
Solicitar sistema	SLS



3. Roles.

Rol	Flujo de trabajo	Actividades
Administrador de cambios	Administración de configuración y control de cambios	Abrir registro de cambios Monitorear cambios
Administrador de configuración	Administración de configuración y control de cambios	Crear ambiente de configuración Distribuir informe de configuración Abrir registro de versiones Monitorear configuración Cambiar y actualizar configuración
Administrador de distribución	Distribución	Realizar pruebas de aceptación en desarrollo Producir entregable Realizar pruebas de aceptación en sitio Conocer el ambiente para el proyecto Hacer plan de distribución
Administrador del proyecto	Administración del proyecto	Concebir proyecto Hacer plan de desarrollo de software Planear ciclo Iniciar ciclo Cerrar ciclo Evaluar ciclo Cerrar proyecto
Analista de sistemas	Requerimientos, Análisis	Identificar actores y casos de uso Estructurar modelo de casos de uso Generar visión de desarrollo
Arquitecto de software	Análisis, Diseño, Implementación	Establecer arquitectura Elaborar modelo de análisis Diseñar arquitectura Elaborar modelo de diseño Establecer modelo de implementación Crear plan de integración
Diseñador de bases de datos	Diseño	Elabora el diagrama entidad relación Diseñar base de datos
Diseñador de pruebas	Pruebas	Hacer el plan de pruebas Hacer el diseño de pruebas Evaluar las pruebas Integrar ciclo
Escritor técnico	Distribución	Hacer manuales
Especialista en herramientas	Ambiente	Preparar el ambiente para un ciclo Conocer el ambiente para el proyecto
Especificador de casos de uso	Requerimientos	Detallar casos de uso Priorizar casos de uso
Ingeniero de casos de uso	Análisis, Diseño	Analizar casos de uso Diseñar casos de uso Diseñar interfaz
Ingeniero de componentes	Análisis, Diseño, Implementación	Analizar clases Analizar paquetes Diseñar clases Diseñar subsistemas Implementar componentes Integrar ciclo
Integrador	Implementación	Integrar ciclo

10 90

RES. CON
FALLA DE ORIGEN

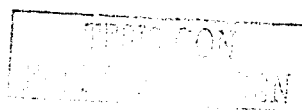
Probador	Pruebas, Distribución	Realizar pruebas de aceptación en desarrollo Ejecutar pruebas Realizar pruebas de aceptación en sitio Integrar ciclo
Solicitante	Requerimientos	Solicitar sistema
Supervisor del proyecto	Administración del proyecto	Monitorear avances

91
CON
FALLA DE ORIGEN

4. Relación entre artefactos y actividades.

Artefacto	Actividades en las que es entrada	Actividad en las que es salida
Arquitectura	ACU, PQ, MA, AQ, CU, SB, MD	EAR, DAQ
Carta de liberación	PAS	PAD
Casos de prueba	PAD, PAS	HDP, PAD
Cierre del proyecto		CRP
Clases del análisis	ACL, EMA	ACU, ACL
Clases del diseño	DCL, DBD, DIT, EMD	DCU, DLC
Componentes	ITC	IMC
Configuración de herramientas		PAC
Descripción de casos de uso	EMC	DTC
Diagrama de casos de uso	PCU, DTC, EMC, GVD, ACU	IAC, PCU
Diagrama Entidad Relación	IMC	DBD
Entregable	PAS, CRP	PEN, PAS
Estándar de administración de riesgos	PDS	
Estándar de Aceptación	PDS, HPD	
Estándar de administración de cambios	ARC	
Estándar de análisis y diseño	IAC, PCU, DTC, EMC, EAR, DAQ, DCU, EMI	
Estándar de codificación	IMC	
Estándar de configuración	CAC, ARV, PAC	
Estándar de documentación	IAC, PCU, DTC, EMC, EAR	
Estándar de evaluación	PDS	
Estándar de pruebas	HPP	
Estándar de base de datos	DBD	
Estándar de resolución de problemas	PDS	
Formato de asignación de trabajo		INC
Formato de Bases de Datos	DIT	DBD
Formato de cierre de ciclo	EVC	CRC
Formato de distribución	CRP	HPD
Formato de monitoreo	MNT	MNT
Formato de preguntas	IAC	IAC
Formato de priorización de casos de uso	PLC	PCU
Formato de requerimientos	IAC	SLS
Formato de validación de entregable	CRP	CRP
Herramientas	PAC	PAC
Informe de ciclo	CRP	EVC
Informe de configuración	DIC	CAC
Informe de entregable	PAS, CRP	PEN, PAS
Informe de integración	CRC	ITC
Interfaz	IMC	DIT
Invitación de pruebas formales	PAD, PAS	PAD, PAS
Lista de herramientas del ciclo	PAC	PAC
Manual de configuración	PEN	HMN
Manual de instalación	PEN	HMN
Manual de usuario	PEN	HMN
Manual técnico	PEN	HMN
Modelo de análisis	DAQ, DCU, HMN	EMA
Modelo de casos de uso	GVD, EAR	EMC
Modelo de diseño	EMI, HDP, HMN	EMD

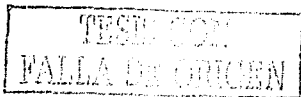
'92



Modelo de Implementación	CPI, IMC, HMN	EMI
Notificación de configuración		DIC, MNT
Paquetes del análisis	APO, EMA	EAR, APO
Plan de ciclos	PAC	PLC
Plan de Desarrollo de Software	CAC, ARV, PLC, HPD, INC, MNT, CRC, EVC, CRP	PDS
Plan de integración	IMC, ITC	CPI
Plan de pruebas	HDP, CPI, PAD, PAS	HPP
Realización de casos de uso del diseño	DCL, EMD	DCU
Realización de los casos de uso del análisis	ACL, EMA, DCV	ACU
Solicitud de cambio	MNT	PAD, MNT
Solicitud de desarrollo	CPY	SLS
Subsistemas de diseño	DSB, EMD	DCU, DSB
Subsistemas o sistema	EPS, HMN, PAD, PEN, PAS	ITC, EPS
Visión de desarrollo	CPY, HPP	GVD
Visión Inicial	PLC, HPP	CPY

5. Artefactos que componen el entregable.

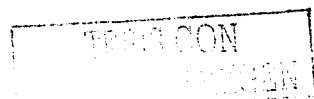
Artefacto	Formato en el que se genera
Carta de aceptación	
Manual de configuración	
Manual de instalación	
Manual de usuario	
Manual técnico Casos de prueba Modelo de análisis Modelo de casos de uso Modelo de diseño Modelo de Implementación Diagrama Entidad Relación	
Subsistemas o sistema	



6. Clasificación por Actividad.

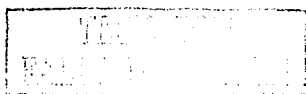
Actividad	Artefactos de entrada	Artefactos de salida
Abrir registro de cambios		
Abrir registro de versiones		
Analizar casos de uso		
Analizar clases		
Analizar paquetes		
Cerrar ciclo		
Cerrar proyecto		
Cambiar y actualizar configuración		
Concebir proyecto		
Conocer el ambiente para el proyecto		
Crear ambiente de configuración		
Crear plan de integración		
Detallar casos de uso		
Diseñar arquitectura		
Diseñar base de datos		
Diseñar casos de uso		
Diseñar clases		
Diseñar interfaz		
Diseñar subsistemas		
Distribuir informe de configuración		
Ejecutar pruebas del sistema		
Elaborar el diagrama entidad relación		
Elaborar modelo de análisis		
Elaborar modelo de diseño		
Establecer arquitectura		
Establecer modelo de implementación		
Estructurar modelo de casos de uso		
Evaluar las pruebas		
Evaluar ciclo		
Generar visión de desarrollo		
Hacer el diseño de pruebas		
Hacer manuales		
Hacer plan de desarrollo de software		
Hacer plan de distribución		
Hacer el plan de pruebas		
Identificar actores y casos de uso		
Implementar componentes		
Iniciar ciclo		
Integrar ciclo		
Monitoreos		
Planear ciclo		
Preparar ambiente para un ciclo		
Priorizar casos de uso		
Producir entregable		
Realizar pruebas de aceptación en desarrollo		
Realizar pruebas de aceptación en sitio		
Solicitar sistema		

94



7. Fases, artefactos y actividades

SIGLAS	FASE	Actividad	Artefactos acumulados al final de la actividad o fase
IN	INICIO	Conocer ambiente para el proyecto	
		Solicitar sistema	
		Identificar actores y casos de uso	
		Priorizar casos de uso	
		Detallar casos de uso	
		Estructurar modelo de casos de uso	
		Generar visión de desarrollo	
		Concebir proyecto	
		Hacer plan de desarrollo de software	
		Crear ambiente de configuración	
		Distribuir informe de configuración	
		Abrir registro de cambios	
		Abrir registro de versiones	
		Planear ciclo	
		Preparar ambiente del ciclo	
	Iniciar ciclo		
	ELABORACIÓN	Establecer arquitectura	
		Analizar casos de uso	
		Analizar clases	
		Analizar paquetes	
		Elaborar modelo de análisis	
		Monitoreo	
		Diseñar arquitectura	
		Diseñar casos de uso	
		Diseñar clases	
		Elaborar modelo entidad relación	
		Diseñar subsistemas	
		Diseñar base de datos	
		Elaborar modelo de diseño	
		Monitoreo	
		Hacer plan de pruebas	
	CONSTRUCCIÓN	Diseñar interfaz	
		Monitoreo	
		Preparar ambiente del ciclo	
		Establecer modelo de implementación	
Hacer el diseño de las pruebas			
Crear plan de integración			
Hacer plan de distribución			
Monitoreo			
Implementar componentes			
Integrar ciclo			
Monitoreo			
Ejecutar pruebas del sistema			
Evaluar pruebas			
Monitoreo			
TRANSICIÓN		Cerrar ciclo	
	Monitoreo		
	Hacer manuales		



		Realizar pruebas de aceptación en desarrollo	
		Monitoreo	
		Producir entregable	
		Monitoreo	
		Realizar pruebas de aceptación en sitio	
		Monitoreo	
		Evaluar ciclo	
		Monitoreo	
		Cerrar proyecto	

YES CON
 FALLA EN OMCEN

96

Estándar de Administración de riesgos.

Introducción.

Este documento establece las políticas a seguir para el manejo de los principales riesgos que existen al Desarrollar Sistemas en el departamento de Informática.

¿Que es un riesgo?

Según el diccionario de la real academia española un riesgo es una contingencia o proximidad de un daño. Lo cual en el área de Ingeniería de Software se puede interpretar como la probabilidad que existe de que un problema de gran magnitud suceda y afecte el proceso de desarrollo de software, de tal forma, que no pueda concluirse.

Alcance

En realidad los riesgos que se corren durante el proceso de desarrollo de software del Departamento de Informática del IIB no son muchos y por lo regular son riesgos que provienen de una catástrofe natural o social o en su defecto son circunstancias que retrasan por un tiempo indeterminado el proyecto, sin que este pueda cancelarse; ya que todos los proyectos que el Departamento de Informática recibe son atendidos.

Lista de Riesgos.

Riesgo	Acciones	Responsable
Huelga	1- Estar al tanto de los problemas en la UNAM. 2- Generar respaldos cada mes del proyecto. 3- En caso de ser muy alta la probabilidad de huelga, generar diariamente respaldos y llevarlos diariamente a un lugar accesible en caso de huelga.	Administrador del proyecto
Infección de virus severa	1- Actualizar continuamente el antivirus. 2- Hacer revisión por lo menos una vez al mes del equipo. 3- Hacer respaldos por lo menos una vez al mes del proyecto	Administrador de cambios
Daño físico del hardware	1- Tener un espejo de la información en otro equipo. 2- Hacer respaldos por lo menos una vez al mes del proyecto en un medio físico.	Administrador de cambios

ESTE DOCUMENTO
 ORIGEN

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



Plan de Desarrollo de Software.

Siglas del sistema _____

Responsable _____

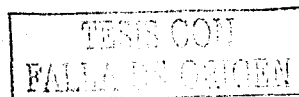
Ciclo _____

Fecha _____

Formado por:

- Estándar de administración de riesgos.
- Estándar de resolución de problemas.
- Estándar de evaluación.
- Estándar de aceptación.
- Plan de Iteraciones.
- Visión Inicial.

7 98





Plan del Ciclo.

Siglas del sistema _____ Ciclo _____
 Responsable _____ Fecha _____

Responsable para la entrega de avances: _____

Objetivo del ciclo: _____

Establecer composición del ciclo (tomar de la Visión de desarrollo y Visión Inicial):

Ciclo#	Casos de Uso (módulos)	Prioridad del Caso o Módulo	Porcentaje de Entrega

Recursos necesarios (tomar de la Visión de Inicial):

Recursos/Hombres	
Nombre	Actividad

Tipo de recurso (material, técnico, etc)	
Nombre	

- Estimación de tiempos (ver formato anexo "gant.xls").
- Anexo Formato de Asignación de trabajo
- Anexo Formato de Cierre de ciclo.
- Anexo Formato de Monitoreo de Avances.
- Anexo Informe del Cierre de ciclo.
- Anexo Informe de ciclo

**Firma del
 Coordinador**

**Firma del
 Administrador del
 proyecto**



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



Formato de asignación de trabajo.

Siglas del sistema _____ Ciclo _____
 Responsable _____ Fecha _____

Nombre del trabajador: _____
 Siglas asignadas: _____
 Rol(es) asignado(s): _____

1. Descripción de actividades.

Nombre	Artefactos

2. Fechas de entrega.

Artefacto	Primeros avances	Primer versión final	Versión final

3. Recursos Asignados.

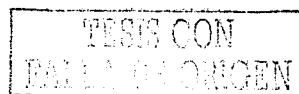
Nombre	Localización	Clave, permiso

 Firma del
 Trabajador

 Firma del
 Coordinador

 Firma del
 Administrador del
 proyecto

201/00



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



Formato de cierre de ciclo.

Siglas del sistema _____ Ciclo _____
 Responsable _____ Fecha _____

1. Nivel de integración alcanzado(# nivel): _____

2. Resultado de actividades.

# Actividad	Grupo de trabajo / actividad	Resultado entregado / propuesta	Comentarios / Problemas (señalar los problemas que se presentaron durante el ciclo)	Problemas / Comentarios

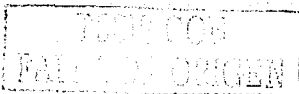
3. Artefactos Obtenidos.

# Actividad	Artefactos

4. Actividades para siguiente Ciclo:

# Actividad	Ciclo	Problemas

**Firma del
 Administrador del
 proyecto**



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



Informe de Ciclo.

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

1. Total de actividades terminadas al 100% _____ de 47
2. Total de artefactos terminados al 100% _____ de 66

3. Problemáticas.

Actividad o artefacto	Problemática	Solución

4. Lista de Entregables.

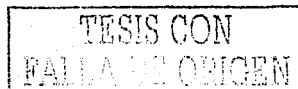
Nombre	Formato

5. Observaciones generales: _____

**Firma del
Coordinador**

**Firma del
Administrador del
proyecto**

102



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



Formato de monitoreo.

Siglas del sistema _____ Ciclo _____
 Responsable _____ Fecha _____

1. Avance del ciclo (Llenar sólo si se trata de un monitoreo de avance del ciclo).

Actividades a monitorear.

Actividad o módulo	Fecha de Entrega propuesta	Nº de Artefactos terminados	Estado de la actividad (terminado, en proceso, pendiente)

Especificar cuales son las actividades que no se han ejecutado al 100% y por qué razón.

Actividad	Comentario

Resumen de actividades (Pendientes).

Actividad	Nº de Artefactos terminados	Nº de Artefactos pendientes	Problemas	Estado de avance	Fecha próxima de terminación

Total de actividades terminadas (esperadas) para la fecha de monitoreo: _____
 Total de actividades terminadas (real): _____

2. Avance de Artefactos (Llenar sólo para monitoreo de avance de artefactos).

Liste todos los artefactos a monitorear y marque las opciones.

Artefacto	Cumple con tipo y tamaño de letra	Cumple con márgenes e interlineado	Estado (fulcrado, terminado, no fulcrado)

Se han obtenido al 100% _____ de un total de _____ artefactos.

Especificar cuales son los artefactos que no se han ejecutado al 100% y por qué razón.

Artefacto	Comentarios

Resumen de artefactos (Pendientes).

Artefacto	Problemáticas	Estado de avance	Fecha próxima de terminación

Total de artefactos terminados (esperados) para la fecha de monitoreo: _____

Total de artefactos terminados (real): _____

3. Resultado.

Observaciones:

Firma del
Supervisor del
proyecto

Firma del
Administrador del
proyecto

104



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



Carta de liberación.

Ciudad Universitaria a

de

Por medio de la presente se hace la entrega del sistema: _____
el cual esta conformado por: _____

Nombre	Formato

Que se atendió con la solicitud _____ entregada al Departamento de Informática
en la fecha _____ por _____

Firma del
Solicitante

Firma del
Coordinador

Firma del
Administrador del
proyecto



Capítulo 6. Adaptación del flujo de trabajo Administración de la Configuración y Control de Cambios del RUP al caso práctico.

Este flujo de trabajo al igual que el anterior se tomó del RUP, puesto que dentro de la primera versión del Proceso Unificado no existe el flujo de trabajo Administración de cambios y configuración. Este flujo de trabajo tiene que ver con el proceso de administrar el proyecto, solo que no se enfoca en la administración del proceso de desarrollo de software únicamente, ya que lo hace además en el desarrollo del producto.

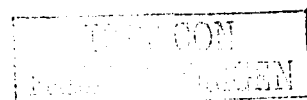
El proceso de descripción del trabajo realizado durante la adecuación de este flujo de trabajo es el mismo que el de los dos flujos anteriores, por lo que primero se menciona lo que el RUP establece para cada elemento del flujo y después se hace la descripción detallada del resultado de la adecuación.

El objetivo principal del flujo de trabajo Administración de la Configuración y control de cambios, es mantener la integridad de los artefactos del proyecto durante todo el desarrollo, lo cual normalmente resulta complicado, pues tener control sobre los cambios que sufren los artefactos de un proyecto durante su desarrollo, no es fácil, ya que la mayoría de las veces la atención se encuentra sobre el producto final, provocando que los cambios que van surgiendo, sólo se vean reflejados en el producto del desarrollo y no en toda la evolución del mismo, que es la documentación.

Esto es muy importante, ya que al final la documentación debería de servir como la *historia clínica* del sistema, o como una memoria técnica, pues en los momentos en que surgen problemas, y es necesario dar mantenimiento o hacer cambios al sistema, la documentación debería contener todas las decisiones y cambios hechos a cada uno de los elementos que componen el sistema, lo cual en la mayoría de las ocasiones no sucede, provocando muchos problemas.

Figura 1. Descripción tomada de los niveles de CMM/SEI

Nivel	Características Proceso	Áreas Clave del Proceso
Optimizado	Se institucionaliza la mejora del proceso	Gestión del Cambio del Proceso Gestión del Cambio de Tecnología Prevención de Defectos
Gestionado	El producto y el proceso se controlan cuantitativamente	Gestión de Calidad Gestión Cuantitativa del Proceso
Definido	Las prácticas técnicas se integran con las prácticas de gestión y se institucionalizan	Enfoque en el Proceso - Definición del Proceso Programa de Entrenamiento - Ingeniería del Producto de Software - Revisiones por Igualares Coordinación entre Grupos - Gestión Integrada del Software
Repetible	Se institucionalizan las prácticas de Gestión del proyecto	Gestión de Requisitos - Planificación del Proyecto - Gestión de Configuración - Garantía de Calidad - Seguimiento y Control del Proyecto - Gestión de Subcontratación
Inicial	El proceso es informal y ad hoc	



El Software Engineering Institute (SEI) de Carnegie Mellon University desarrolló el modelo Capability Maturity Model (CMM) el cual tienen como objetivo proporcionar una guía para las organizaciones desarrolladoras de software, que les permite obtener control de sus procesos de desarrollo y mantenimiento de software, de tal forma que esto les permita evolucionar hacia una cultura de calidad en el software. De tal forma que CMM/SEI sirve como herramienta para la clasificación de las organizaciones de acuerdo a la madurez de sus procesos y como guía para la mejora continua de las mismas organizaciones (ver figura 1).

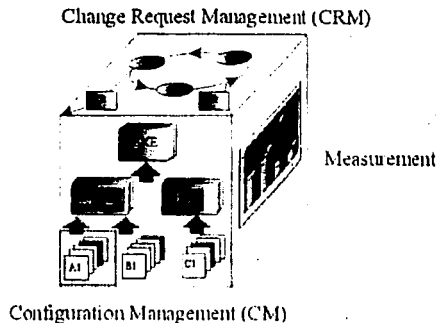
Lo anterior es importante mencionarlo por que dos de los tres flujos de trabajo que se tocan en esta tesis, tienen como objetivo el establecer medios para la administración y control de lo que se está produciendo, y de la forma en la que se está produciendo, lo cual tiene mucho que ver con lo que se habla en el nivel 2 de CMM (Repeatible), que básicamente se enfoca en el seguimiento del proceso. Pues en CMM se establece, que no es suficiente crear un proceso para el desarrollo de software, además debe ser posible ejecutarlo mas de una vez y llevar un control de cada uno de los proyectos desarrollados, con el proceso establecido.

Por último debe mencionarse que para la comparación entre lo establecido por el RUP y el resultado de la adecuación, al igual que en los capítulos anteriores, se utilizaron los diagramas tal cual aparecen en el RUP, con la intención de mostrar que los cambios hechos no sólo están en la estructura de cada flujo de trabajo, sino que además en el hecho de crear un proceso descrito en nuestro propio idioma, el español.

6.1 El flujo de trabajo Administración de la Configuración y control de Cambios de acuerdo al RUP

El RUP al hacer una definición del flujo de trabajo Administración de la configuración y control de cambios, lo esquematiza con un "cubo", en el cual se ilustran tres funciones interdependientes, cada lado examina aspectos distintos del problema de la administración de cambios y configuración (ver figura 2). La cara Administración de la configuración (CM) está relacionada con la estructura del producto, pues se refiere a la identificación de las versiones de los artefactos y las dependencias entre ellos, de tal forma que puedan crearse conjuntos de configuraciones de acuerdo a su versión.

Figura 2. Cubo de la Administración de la configuración y control de cambios tomado del RUP.



La cara Administración de cambios a requerimientos (CRM) está enfocada a la estructura del proceso, pues recaba y administra las solicitudes de cambio generados interna o externamente. Esto incluye el análisis del impacto de las solicitudes de cambio sobre el proyecto.

La última cara, se refiere a las métricas, por lo que se enfoca en la estructura de control del proyecto, esto es básicamente el conjunto de herramientas que permiten explotar la información recabada durante la administración de cambios y configuración.

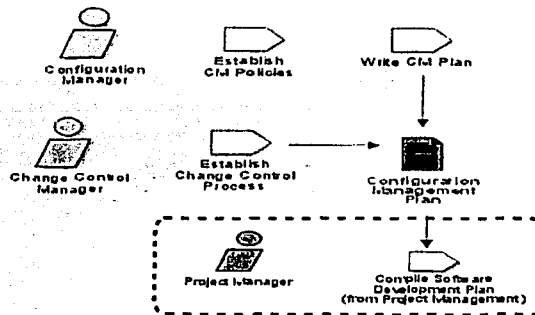
Estos tres aspectos de la administración de cambios y administración de la configuración, se ven envueltos dentro de las actividades que componen el flujo de trabajo Administración de cambios y configuración que a continuación se explican.

6.1.1 Actividades

La Administración de la configuración y control de cambios se enfoca en cuatro actividades importantes: Identificar los elementos que componen la configuración, restringir y auditar los cambios de cada elemento y definir y administrar la configuración de cada elemento. Estos puntos son cubiertos por el flujo en la ejecución de las siguientes actividades:

- *Creación del Plan de Configuración del Proyecto y Control de Cambios.* Tiene como propósito crear un documento en el cual se establezcan las políticas para la administración de cambios y configuración, procedimientos para la solicitud de cambios y la organización de versiones; y por último los nombres de los artefactos que forman parte de la configuración (ver Figura 3).

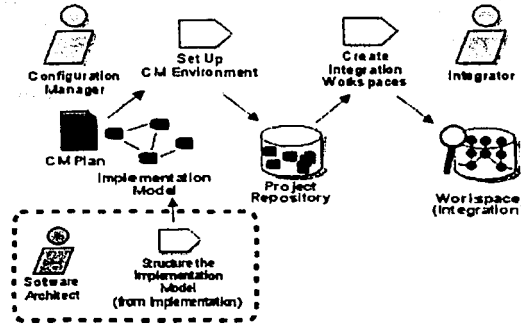
Figura 3. Diagrama tomado de la Actividad Creación del plan de configuración del proyecto en el RUP.



- *Creación del ambiente de administración de configuración del proyecto.* El propósito de esta actividad es crear un depósito, al cual los trabajadores puedan

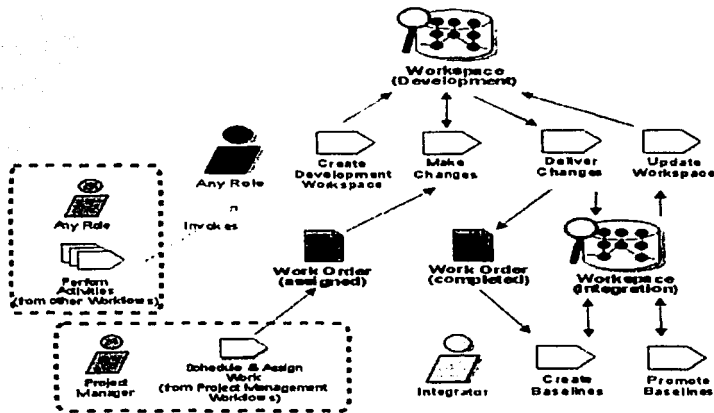
acceder para el uso de los artefactos de la versión en proceso, que a la vez esté controlada por el administrador de la configuración (ver Figura 4).

Figura 4. Diagrama tomado de la actividad Creación del ambiente de administración de la configuración en el RUP.



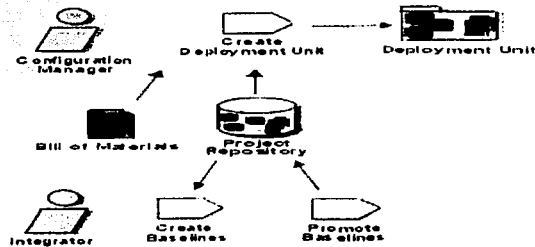
- *Control de Cambios y entrega de elementos de la configuración.* Esta actividad se refiere a la actualización de la configuración, en caso de haber un cambio este debe verse reflejado en la configuración actual, provocando la creación de una nueva versión de trabajo y otra versión anterior, la cual sirve para tener un respaldo de lo anterior en caso de ser necesario regresar a la versión anterior (ver Figura 5).

Figura 5. Diagrama tomado de la actividad Control de cambios y entrega de elementos de la configuración en el RUP.



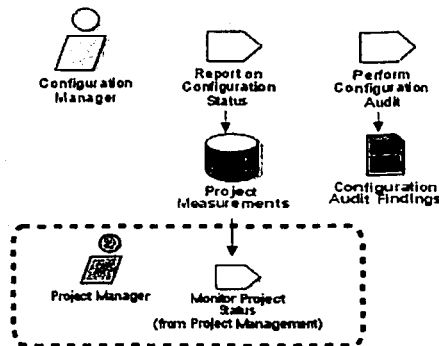
- **Administración de versiones.** Se refiere a la continua administración y almacenamiento de las versiones que se van generando, de acuerdo a los cambios efectuados durante el proceso de desarrollo del producto(ver Figura 6).

Figura 6. Diagrama tomado de la actividad Administración de versiones en el RUP.



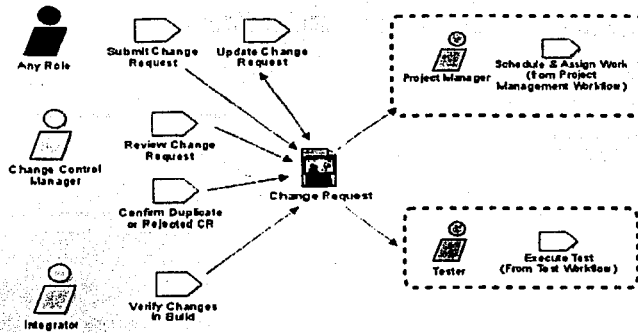
- **Monitorear y reportar estado de configuración.** Se refiere a la revisión continua del estado de la configuración, ya que es necesario mantener la configuración del ambiente de trabajo actualizada(ver Figura 7).

Figura 7. Diagrama tomado de la actividad Monitorear y reportar estado de la configuración en el RUP.



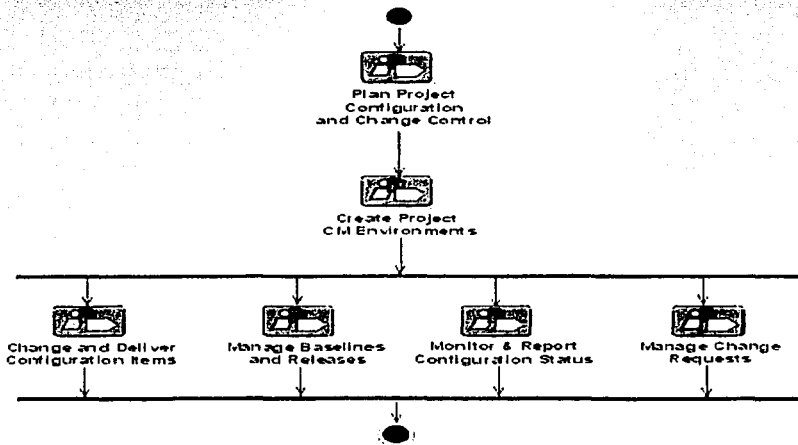
- **Administración de cambios a requerimientos.** Esta actividad se relaciona directamente con las solicitudes de cambio, pues una vez recibida la solicitud es necesario analizar el impacto del cambio para determinar las alternativas a seguir, informando a todo el equipo de trabajo, las resoluciones tomadas (ver Figura 8).

Figura 8. Diagrama tomado de la actividad Administración de requerimientos de cambio en el RUP.



La secuencia de ejecución de estas actividades se ve claramente en el diagrama establecido por el RUP para este flujo (ver figura 9).

Figura 9. Diagrama tomado del Flujo de trabajo Administración de la configuración y control de cambios en el RUP.



6.1.2 Roles

Administrador de configuración. Es el responsable de establecer la estructura para la administración de la configuración, así como otorgar los espacios para el desarrollo e integración del producto. Este rol también genera la información necesaria para el administrador del proyecto.

Administrador del control de cambios. Es responsable de definir el proceso de administración de cambios, además de ser quien se encarga de la recepción, análisis y seguimiento de las solicitudes de cambios y en caso de ser necesario es quien organiza y coordinar la mesa de control de cambios.

Arquitecto. Es quien efectúa los cambios en la arquitectura.

Implementador. Es quien efectúa los cambios en los componentes.

Integrador. Es quien acepta e integra los nuevos cambios al proyecto, en caso de suceder durante la integración.

6.1.3 Artefactos

Plan de administración de configuración. Describe las políticas y prácticas usadas en el proyecto para la Administración de la configuración. Este documento forma parte del Plan de desarrollo de Software.

Requerimiento de cambio. Es la solicitud en la que se expresa la necesidad de efectuar un cambio en el proyecto.

Depósito del proyecto. Es el lugar en donde se van almacenando las versiones del proyecto.

Resultado de la auditoria de la configuración. Es un reporte en donde se establecen los resultados de los monitoreos hechos a la configuración.

6.2 El flujo de trabajo Administración de cambios y configuración, adaptado al caso práctico.

Al hacer el análisis para la adecuación del Flujo de trabajo Administración de cambios y configuración, al igual que en los flujos anteriores, fue necesario identificar las actividades que actualmente se realizan en el Departamento de Informática y que tienen que ver con este flujo.

Sin embargo al analizar se encontró que sólo se hacen dos actividades. La primera actividad es la recepción de solicitudes de cambio, las cuales al ser recibidas por el responsable o encargado del proyecto, simplemente son atendidas, sin hacer ningún tipo de evaluación al respecto. Esta situación ha provocado muchos problemas, ya que en más de un proyecto y más de una ocasión, los cambios son retrocesos, es decir que en un momento solicitan un cambio y después solicitan otro cambio en el que se deshace el cambio anterior. En otras ocasiones resulta más grave pues una vez hechos los cambios, los usuarios se retractan de su decisión y en ocasiones suelen molestarse pues ya no recuerdan que ellos solicitaron esos cambios.

La otra actividad que se realiza en el Departamento de Informática del IIB, es el respaldo de los proyectos, que se puede relacionar con lo que el RUP llama creación del ambiente de configuración, que por supuesto actualmente no es como lo establece el RUP, ya que sólo se hacen los respaldos de cada proyecto, sin clasificar en versiones, pues en el mejor de los casos lo único que se tiene de cada sistema es el código fuente.

Estas dos actividades dieron pauta para dividir al flujo de trabajo en dos procesos, en el Departamento de Informática, , por un lado la Administración de cambios y por otro la Administración de la configuración, de tal forma que a diferencia del RUP, el diagrama general del flujo de trabajo Administración de cambios y configuración adaptado al caso práctico sólo contempla estos dos procesos (ver figura 10), que sucesivamente se componen por actividades las cuales se detallarán en los diagramas correspondientes (ver figura 11 y 12).

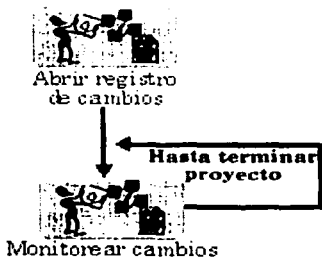
Figura 10. Diagrama general del flujo de trabajo

Administración de cambios y configuración

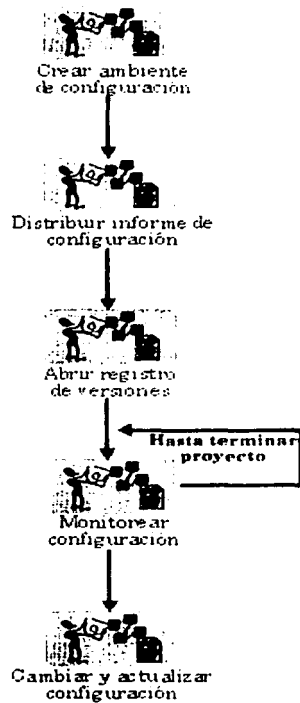


Figura 12. Administración de Cambios

Administración de cambios



Administración de la configuración



Esta división no sólo diferencia claramente a los dos procesos, de tal forma que es posible identificar cuáles son las actividades que se deben seguir para cada uno de estos, además hace más fácil su entendimiento y ejecución, ya que al comparar, el flujo resultado de la adecuación, con el flujo establecido por el RUP (ver tabla 1) es posible identificar rápidamente a qué se refiere cada proceso de este flujo.

Tabla 1. Comparación entre el resultado de la adecuación y el RUP

Administración de Cambios	Administración de la configuración	
Establecer estándar de Administración de cambios	Establecer estándar de administración de la configuración	Creación del Plan de Configuración del Proyecto y Control de Cambios.
Abrir registro de cambios	Crear ambiente de configuración	Creación del ambiente de administración de configuración del proyecto
	Distribuir informe de configuración	
	Abrir registro de versiones	Administración de versiones.
	Monitorear configuración	Monitorear y reportar estado de configuración.
	Cambiar y actualizar configuración	Cambio y entregar elementos de configuración.
Monitorear cambios		Administración de requerimientos de cambio.

6.2.1 Actividades

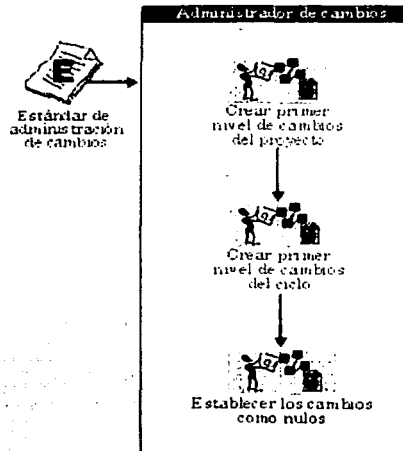
Tanto para la administración de cambios como para la administración de la configuración el RUP menciona una actividad denominada Creación del plan de configuración del proyecto y control de cambios. Sin embargo, para el caso práctico éste plan se estableció como un estándar, ya que se consideró que al igual que en casos anteriores, la mayoría de los proyectos del Departamento de Informática podían regirse por el mismo plan, por lo que fue posible convertirlo en un estándar.

Administración de cambios.

- **Abrir registro de cambios.** La primera actividad de la administración de cambios es Abrir el registro de cambios, la cual fue extraída de la actividad que el RUP denomina Creación del ambiente de administración de configuración del proyecto. Los pasos se describen a continuación(ver Figura 13):

Figura 13. Diagrama de la actividad Abrir registro de cambios en la adecuación.

Abrir registro de cambios



- o *Crear primer nivel de cambios del proyecto.* Se refiere a crea un fólder para el registro de los cambios. Este fólder contendrá una impresión de las solicitudes de cambio, de tal forma que sea posible consultar rápidamente cuáles son los cambios hasta el momento. Para consultar la forma de archivar los cambios de cada proyecto deberá consultarse el *Estándar de administración de cambios*.



- *Crear primer nivel de cambios del ciclo.* Este fólder deberá estar dividido en ciclos, para poder separar los cambios por ciclo y de esa forma identificar cómo se fueron dando los cambios.
- *Establecer los cambios como nulos.* Esto se refiere a dar por iniciada la administración de cambios.
- **Monitorear cambios.** La segunda actividad de la administración de cambios es la más importante, ya que se ejecuta durante todo el ciclo de vida del proyecto. Se tomó de lo que el RUP marca como Administración de cambios a requerimientos. A pesar de que la actividad tenga el nombre de monitoreo, en realidad no se está haciendo una búsqueda continua de cambios, si no que simplemente en el momento que surge la necesidad de un cambio, esta actividad dice cómo debe procederse para dar una respuesta a la solicitud de cambio.

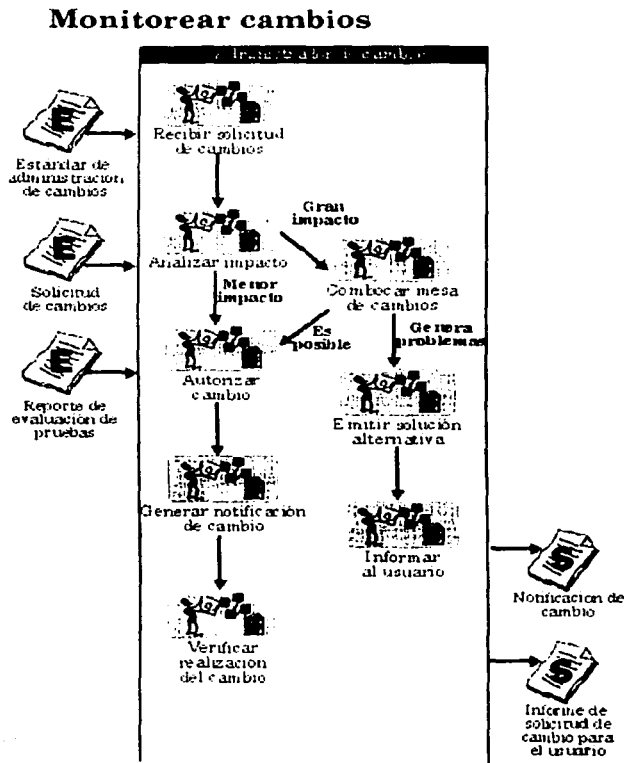
Los pasos que componen el monitoreo de cambios (ver figura 14) son:

- *Recibir solicitud de cambios.* Se refiere a archivar la solicitud de cambios, en el fólder correspondiente del proyecto y ciclo. Esta solicitud de cambios puede venir del usuario, del resultado de la evaluación de pruebas [Salazar, 2003], o de una verificación del entregable. Cuando la solicitud de cambios proviene de la evaluación de una prueba, se toma como solicitud de cambios *el reporté de evaluación de pruebas.*
- *Analizar impacto.* Una vez registrada la solicitud, el Administrador de cambios hace el análisis del impacto del cambio y así determina cuáles son las repercusiones al proyecto, generadas por ese cambio.
- *Convocar mesa de cambios.* Esto se hace sólo en el caso de que el cambio recibido, tenga muchas repercusiones para el proyecto, el proceso para hacer la convocatoria y las personas a las que debe convocarse está descrito dentro del estándar de Administración de cambios.
- *Autorizar cambio.* Esto se refiere a que una vez hecho el análisis o en su caso tener el resultado aprobatorio de la mesa de cambios, el Administrador de cambios informa al Administrador del proyecto para dar comienzo a la notificación del cambio, esto se hace a través del *Formato de notificación de cambio*, con la intención de que la distribución de trabajo sea autorizada por el Administrador del Proyecto.
- *Emitir solución alternativa.* Esto sucede cuando el impacto del cambio es muy grande, es decir que cambia por completo alguno de los requerimientos iniciales hasta convertirlo en algo completamente diferente o contrario, lo cual provoca que al realizarlo se den problemas en el proyecto, desde el incumplimiento en los tiempos planeados hasta la obsolescencia del trabajo realizado hasta el momento. Es entonces que el administrador del proyecto toma una decisión alternativa. En estos casos dentro del *Estándar de*

administración de cambios, existen propuestas de solución para problemas poco frecuentes.

- **Informar Usuario.** Esto se hace sólo en los casos en el que el cambio no se realizará por el impacto que tiene sobre el proyecto, lo que se genera es un documento, para el cual se hizo el *formato de Informe de solicitud de cambio para el usuario*, en el que se detalla la solución alternativa a la petición hecha por el solicitante.
- **Generar notificación de cambio.** El Administrador de cambios informa a las personas responsables (Integrador, Ingeniero de componentes o Arquitecto) para que efectúen los cambios necesarios y al Administrador de la Configuración se le deberá informar, en caso de que el cambio sea grande, que las versiones van a cambiar. Esta asignación se hace con la autorización del Administrador del Proyecto.

Figura 14. Diagrama de la actividad Monitorear cambios en la adecuación.



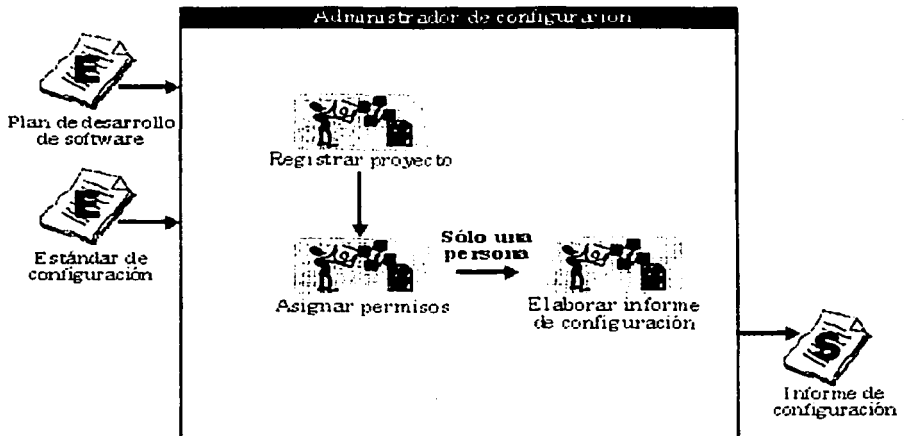
- *Verificar realización de cambio.* Después de haber informado a los responsables de realizar los cambios, el Administrador de cambios deberá rectificar que los cambios se hayan efectuado y que el administrador de configuración ya tenga las versiones completas, esto se hace con *el Formato de verificación de cambios.*

Administración de la configuración.

- **Crear ambiente de configuración.** Esta actividad deriva de la actividad que el RUP establece con el mismo nombre y se refiere a preparar todo para dar inicio a la administración de la configuración. Esta actividad (ver figura 15) esta compuesta por los siguientes pasos:
 - *Registrar proyecto.* Se refiere a abrir un fólder, para almacenar una copia de los originales de cada artefacto generado del proyecto. Por tratarse de la apertura se inicia con el almacenamiento de los artefactos generados hasta el momento de la apertura de este registro (Visión de desarrollo y Visión Inicial). La forma de almacenar los artefactos de un proyecto esta detallada en el *Estándar de administración de la configuración* y debe registrarse en el *Formato de informe de configuración.* Cuando el equipo de trabajo para el desarrollo del proyecto es de una persona, esta persona debe hacer dos copias, una para el y la otra debe quedar almacenada como lo marca el *Estándar de administración de la configuración*, en la sección de almacenamiento de versiones.

Figura 15. Diagrama de la actividad Crear ambiente de configuración en la adecuación.

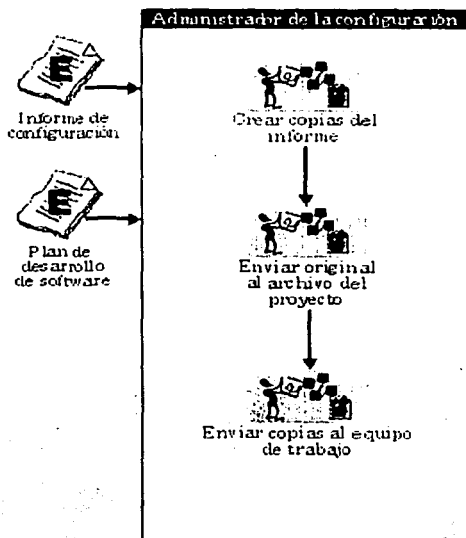
Crear ambiente de configuración



- *Asignar permisos.* Se refiere a asignar las contraseñas a los artefactos, esto se hace completando el *Formato de Informe de configuración*. Esta asignación se hace con base a lo establecido en el *Estándar de Configuración*.
- *Elaborar informe de configuración.* Este paso se realiza sólo cuando el equipo de trabajo para el desarrollo del proyecto es de una persona y se refiere a imprimir el formato de informe de configuración, para anexarlo al *Plan de Desarrollo de Software*.
- **Distribuir informe de configuración.** Esta actividad se realiza sólo cuando el equipo de trabajo para el desarrollo del proyecto es de más de una persona. Se refiere a proporcionar una copia de los artefactos obtenidos hasta el momento, a cada participante del equipo de trabajo con la intención de que trabajen con la misma versión todos. Los pasos (ver figura 16) para esta actividad son los siguientes:
 - *Crear copias del informe.* Se generan copias impresas del informe de configuración, de acuerdo al número de integrantes del equipo, para hacer la repartición correspondiente.

Figura 16. Diagrama de la actividad Distribuir informe de configuración en la adecuación.

Distribuir informe de configuración

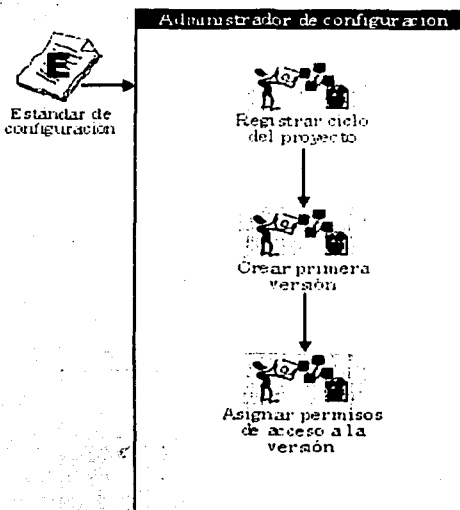


- *Enviar original al archivo del proyecto.* Se guarda el original en el archivo de la configuración y se agrega al *Plan de Desarrollo del Proyecto*.

- *Enviar copias al equipo de trabajo.* Se entrega a cada persona una copia electrónica, de todos los artefactos que se tienen hasta el momento y una copia impresa del Informe de configuración.
- **Abrir registro de versiones.** Cuando el equipo de trabajo para el desarrollo del proyecto es de una persona esta actividad también se omite, pues con la realización de la primera actividad, se tiene ya la primera versión almacenada en forma individual. Esta actividad es similar a la primer actividad, que es crear el ambiente de configuración, pues se hace una copia de lo que se creó como ambiente de configuración, esto constituye la primera versión del proyecto y de esta forma se da comienzo a la administración de versiones que se establece en el RUP. Los pasos para esta actividad (ver figura 17) son los siguientes:
 - *Registra ciclo del proyecto.* Esto es crear un fólder, en papel y electrónico, para almacenar la primera versión del ciclo en cuestión. La especificación para la creación de este fólder se detalla en el *Estándar de configuración*.

Figura 17. Diagrama de la actividad Abrir registro de versiones en la adecuación.

Abrir registro de versiones



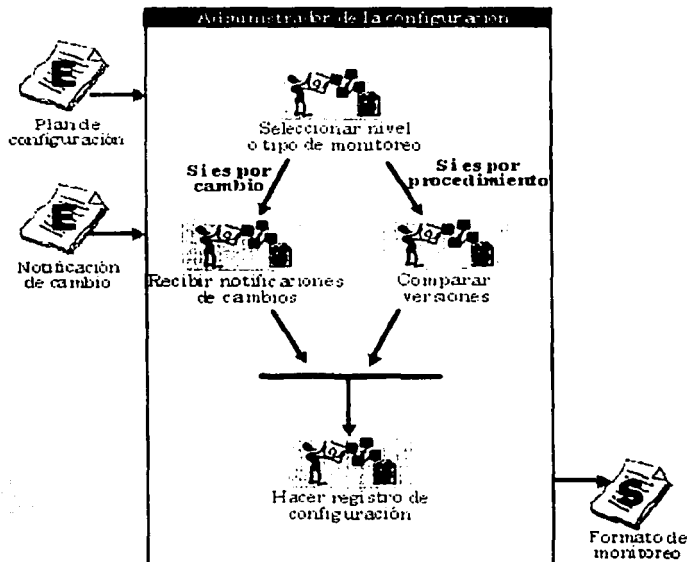
- *Crear primera versión.* Se hace una copia de lo que compone la primera versión del proyecto (Visión de desarrollo y Visión Inicial).
- *Asignar permisos de acceso a la versión.* Se establece qué personas pueden tener acceso a la versiones completas de cada proyecto. Esto esta definido en

el *Estándar de configuración*, por lo que sólo debe informarse a las personas correspondientes.

- **Monitorear configuración.** Esta actividad se realiza con la intención de ver que la configuración no haya sufrido cambios y que si se han efectuado se vean reflejados para todo el equipo de trabajo, sin embargo también puede ejecutarse cuando se haya autorizado un cambio y debe actualizarse la configuración. Esta actividad tiene el mismo nombre en el RUP, sin embargo para la actual adecuación se hizo una especificación más detallada de los pasos que involucra esta actividad, (ver figura 18) los cuales son:
 - *Seleccionar nivel o tipo de monitoreo.* Esto se hace así por que, la configuración puede haber cambiado por una autorización, o simplemente por el mismo proceso de desarrollo de software, por lo tanto el proceso de monitoreo es diferente en cada caso, lo cual esta bien definido en el *Estándar de evaluación*.

Figura 18. Diagrama de la actividad Monitorear configuración en la adecuación.

Monitorear configuración

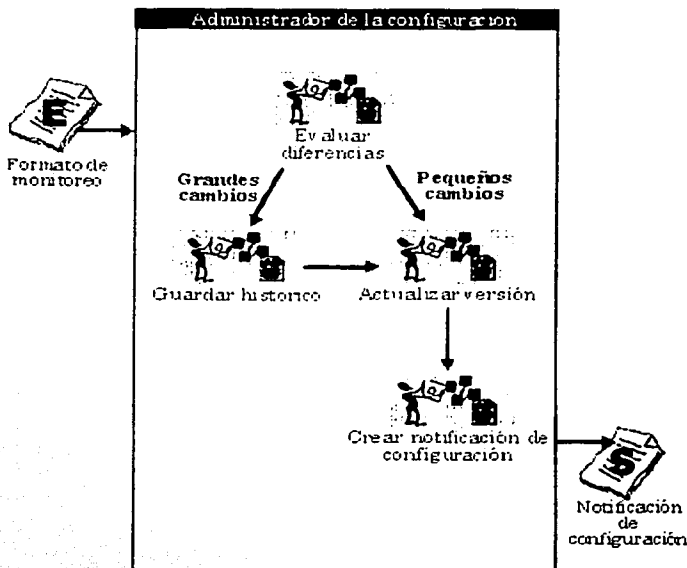


- *Recibir notificaciones de cambios.* Esta actividad se realiza sólo cuando el monitoreo se activa por una solicitud de cambio. Es entonces necesario saber que fue lo que se cambió.

- *Comparar versiones.* Esto sólo se realiza cuando el monitoreo es por rutina y lo que se intenta es detectar algún cambio en la configuración originado por el mismo proceso de desarrollo.
- *Hacer registro de configuración.* Se refiere a registrar en el *formato de monitoreo de configuración* el resultado de los pasos anteriores.
- **Cambiar y actualizar configuración.** Esto se realiza sólo en los casos de que el informe del monitoreo reporte cambios. Esta última actividad que establece el RUP para administración de la configuración, no solo sirve para mantener actualizada la configuración, si no que además genera las versiones del proyecto. Los pasos de esta actividad (ver figura 19) son:
 - *Evaluar diferencias.* Se refiere a determinar en base al *Estándar de Administración de la configuración*, si los cambios encontrados en la configuración ameritan la creación de una nueva versión.

Figura 19. Diagrama de la actividad Cambiar y actualizar configuración en la adecuación.

Cambiar y actualizar configuración



- *Guardar histórico.* En caso de que los cambios sean determinantes, entonces se hace el registro de una nueva versión, en donde se almacenan todos los artefactos, sin el cambio, que formarán la versión histórica.

- *Actualizar versión.* Se refiere a actualizar la última versión con los cambios detectados durante el monitoreo, esto involucra agregar los artefactos que no existen en la última versión almacenada.
- *Crear notificación de configuración.* Una vez hecho el movimiento de versiones, es necesario notificar y entregar la nueva configuración, al Administrador del Proyecto y a todos los trabajadores del equipo, para que trabajen todos bajo la misma configuración, esto se hace con el formato de notificación de configuración y sólo se realiza cuando el equipo de trabajo es de más de una persona. Esta actualización de configuración se describe en el *Estándar de configuración.*

6.2.2 Roles

Los roles que tienen que ver exclusivamente con este flujo son dos: el Administrador de la configuración y el Administrador de cambios, los cuales tienen las mismas responsabilidades que el RUP establece para cada uno. Los roles que interfieren en este flujo pero que no tienen que ver directamente son: el integrador, el programador y el arquitecto, que en realidad solo intervienen cuando hay cambios, ya que ellos son los responsables de ejecutarlos. Sin embargo no son roles que se hayan creado para este flujo ya que forman parte de otros flujos.

6.2.3 Artefactos

Todos los artefactos generados durante la adaptación de este flujo de trabajo pueden ser consultados en el anexo correspondiente al capítulo.

Estándar de Administración de configuración.

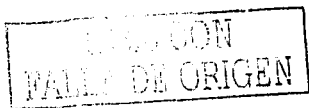
Este estándar establece qué artefactos conforman la configuración, cuáles son los formatos para el registro y control de la configuración, cada cuando o en qué momento deben efectuarse los monitoreos de la configuración y los procedimientos para la notificación de cambios.

Estándar de Administración de cambios.

Este documento establece cuáles son las políticas a seguir para la realización de un cambio, ahí también se encuentran los formatos necesarios para este proceso, además de las políticas para el establecimiento de la mesa de cambio, así como parámetros que permiten determinar cuando debe convocarse una mesa de cambios y propuesta de solución a problemas comunes.

Informe de configuración.

Es un formato en el que se plasma el estado actual de la configuración y los permisos para cada artefacto que compone la configuración.



Formato de monitoreo de configuración.

Es un formato que contiene los nombres de los artefactos que deben revisarse, en caso de tratarse de un monitoreo de rutina. Además de contener una sección en la que se listan los elementos que fueron cambiados, así como los cambios que se encontraron.

Formato de Notificación de configuración.

Este es un formato en el cual se le informa a cada individuo de los cambios que se efectuaron en la configuración.

Solicitud de cambios.

Es el formato empleado para solicitar un cambio.

Notificación de cambio.

Este formato se utiliza para informar a todos los involucrados la aceptación de un cambio. Por lo regular esta va acompañada de un formato de asignación de trabajo, para quienes deben realizar los cambios.

Formato de verificación de cambios.

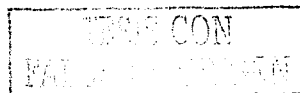
Este formato contiene el resultado de la verificación de los cambios realizados después de la aceptación de una solicitud de cambio.

Informe de solicitud de cambio para el usuario.

Es un formato en el cual se le informa al usuario de la determinación de la mesa de cambios. Es este informe se establecen las razones por las que no se realiza el cambio, además de la alternativa que se tomó.

**Artefactos de
Administración de la
Configuración y Control de
Cambios.
(Anexos)**

125



Estándar de administración de la configuración.

Introducción.

Este estándar establece qué artefactos conforman la configuración, cuáles son los formatos para el registro y control de la configuración, cada cuando o en qué momento deben efectuarse los monitoreos de la configuración y los procedimientos para la notificación de cambios en la configuración.

Elementos que conforman la configuración.

Todos los artefactos contenidos en la lista de artefactos por actividad, son los que conforman la configuración. Además de las herramientas para el trabajo diario durante el proceso de desarrollo de software, las bibliotecas de funciones y la configuración de permisos de acceso a la información de acuerdo al rol asignado.

Monitoreos de configuración.

El objetivo de estos monitoreos es asegurar que la estructura del producto se mantiene durante todo el proceso de desarrollo de software, por lo que siempre el administrador de la configuración conserva una versión correcta del proyecto.

Los monitoreos de la configuración están establecidos en el estándar de evaluación como un monitoreo de Tipo2, así como la descripción del mismo.

Procedimiento de notificación de configuración.

- La primer notificación de la configuración es la que se hace al abrir el registro del proyecto, y se hace a través del formato de informe de configuración.
- La notificación sucede cuando después de un monitoreo se detectaron cambios, de tipo estructural, lo cual genera la creación de una nueva versión.
- La forma de notificar es a través del formato de notificación, a todos los miembros del equipo de trabajo.

Políticas para la creación de una nueva versión.

- Se encontraron cambios en la secuencia de los procesos descritos.
- Se encontraron cambios en el número de componentes, clases, módulos o casos de uso.
- Se encontraron cambios en la arquitectura o patrón seleccionado.
- Se encontraron cambios en la plataforma de la aplicación.

- Se encontraron cambios en el lenguaje de programación.
- Se encontraron cambios en la descripción de los componentes, clases, módulos o casos de uso.
- Se encontraron cambios en la descripción de los algoritmos.

Políticas para almacenamiento de versiones

Creación del fólдер.

- Si es la primera versión crear una carpeta dentro del equipo personal del Administrador de configuración, en la unidad que el designe para el almacenamiento de versiones, bajo la carpeta proyectos. El nombre del fólдер es el resultado de combinar las siglas del proyecto, fecha de almacenamiento y número de versión. Por ejemplo si el proyecto se llama SISCLIN, se esta creando el día 5 de febrero de 2002 y es la primera versión, la carpeta deberá nombrarse:

C:\PROYECTOS\SISCLIN05022002V1

- Si ya existe una versión y se debe crear otra, deberá seguirse el mismo patrón, marcado anteriormente. Por ejemplo si se crea la segunda versión de SISCLIN, el día 8 de abril de 2002, la carpeta deberá nombrarse:

C:\PROYECTOS\SISCLIN08042002V2

Estructura de directorios dentro del fólдер

- El fólдер deberá contener separada la información de acuerdo a la siguiente estructura:

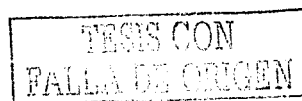
FASE	DIRECTORIO
INICIO	INICIO
ELABORACIÓN	ELABOR
CONSTRUCCIÓN	CONSTR
TRANSICIÓN	TRANSC

- Los artefactos contenidos en cada directorio serán las salidas de las actividades contenidas en cada fase. Para más información consultar *Lista de actividades, artefactos y roles*.

Asignación de contraseñas.

- Las contraseñas se asignan a nivel de archivo (para acceso a la configuración) y a nivel de fólдер (para acceso a las versiones). Se recomienda que la contraseña involucre dos caracteres especiales (*?\${}&%#&|!) tres alfabéticos y tres números, sin importar el orden. Por ejemplo una contraseña valida es:

E9R#TS67



- Las contraseñas de versiones sólo deben proporcionarse a las personas que tienen los roles de: Administrador de proyecto, Administrador de configuración y Coordinador.

Anexo formato de monitoreo de configuración
Anexo formato de informe de configuración
Anexo formato de notificación de configuración.



1281

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Estándar de administración de cambios.

Introducción.

Este documento establece las políticas para la realización de un cambio, los formatos necesarios para este proceso, las políticas para el establecimiento de la mesa de cambio, así como parámetros que determinan cuándo debe convocarse una mesa de cambios y propuesta de solución a problemas comunes, aun que poco frecuentes.

¿Qué es un cambio?

Para el Departamento de Informática, un cambio es un requerimiento no planteado al inicio del proyecto, que provoca una transformación evidente del proyecto.

Política para la realización de un cambio.

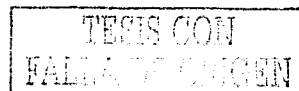
- Todo cambio requiere ser expresado por escrito a través del formato de solicitud de cambio, avalado por el solicitante del sistema.
- Cualquier solicitud de cambio recibida deberá ser sometida a evaluación antes de realizarse.
- Todo cambio aceptado deberá informarse por escrito a cada una de las personas involucradas.
- Todo cambio aceptado deberá de ser verificado y reflejado en la última versión del proyecto.

Conformación de la mesa de cambios.

- Administrador del proyecto
- Arquitecto.
- Jefe del departamento.

Políticas para convocar a la mesa de cambios.

- Si el cambio afecta a más de una fase.
- Si el cambio se solicita en la fase final del proceso de desarrollo de software.
- Si el cambio retrasa con más de dos semanas el plan de actividades.



Procedimiento para realizar un cambio.

- Notificar del cambio a los responsables de las actividades impactadas
- Al cumplimiento del tiempo estimado para el cambio, hacer la verificación de que los cambios se han realizado satisfactoriamente.
- Informar al Administrador de versiones que se ha generado una nueva versión del proyecto, para que sea actualizada.

Soluciones alternativas para cambios

- Dejar para el siguiente ciclo.
- Tomar como un nuevo proyecto.

Anexo Formato de solicitud de cambios.

Anexo Notificación de cambios.

Anexo Informe para el usuario

Anexo Formato de verificación de cambios.



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



Formato de Informe de configuración.

Siglas del sistema _____ Ciclo _____
 Responsable _____ Fecha _____

1. Asignación de contraseñas a los artefactos.

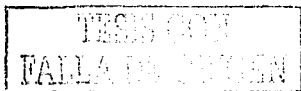
Direccionaliz	Contraseña(L)	Contraseña(E)
Arquitectura		
Carta de liberación		
Casos de prueba		
Cierre del proyecto		
Clases del análisis		
Clases del diseño		
Componentes		
Configuración de herramientas		
Descripción del patrón de arquitectura		
Descripción de casos de uso		
Diagramas de casos de uso		
Diseño de arquitectura		
Diagrama Entidad Relación		
Entregable		
Estándar de administración de riesgos		
Estándar de Aceptación		
Estándar de administración de cambios		
Estándar de análisis y diseño		
Estándar de codificación		
Estándar de configuración		
Estándar de documentación		
Estándar de evaluación		
Estándar de pruebas		
Estándar de base de datos		
Estándar de requerimientos		
Estándar de herramientas		
Estándar de resolución de problemas		
Formato de asignación de trabajo		
Formato de diseño de base de datos		
Formato de cierre de ciclo		

173-1

TESIS CON
FALLA DE ENTREGA

Formato de distribución		
Formato de monitoreo		
Formato de preguntas		
Formato de prioridad de casos de uso		
Formato de requerimientos		
Formato de validación de entregable		
Herramientas		
Informe de ciclo		
Informe de configuración		
Informe de entregable		
Informe de integración		
Interfaz		
Invitación de pruebas formales		
Lista de herramientas del ciclo		
Manual de configuración		
Manual de instalación		
Manual de usuario		
Manual técnico		
Modelo de análisis		
Modelo de casos de uso		
Modelo de diseño		
Modelo de Implementación		
Notificación de configuración		
Paquetes del análisis		
Plan de ciclos		
Plan de Desarrollo de Software		
Plan de integración		
Plan de pruebas		
Realización de casos de uso del diseño		
Realización de casos de uso del análisis		
Solicitud de cambio		
Solicitud de desarrollo de software		
Subsistemas de diseño		
Subsistemas o sistema		
Visión de desarrollo		
Visión Inicial		

**Firma del
Administrador de
configuración**



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



Formato de Monitoreo de configuración .

Siglas del sistema _____ Ciclo _____
 Responsable _____ Fecha _____

1. Actividad: _____

2. Verificación de artefactos.

Artefacto	Descripción	Verificación	Cambios encontrados (SI/NO)	Cambios en estructura (SI/NO)

Total de cambios encontrados: _____

3. Cambios.

Artefacto	Descripción del cambio

3.1. Verificar artefactos derivados.

Artefactos derivados	Cambios reflejados (SI/NO)

Total de artefactos derivados no actualizados: _____

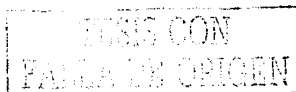
3.2 Lista de artefactos a cambiar

Nombre

Observaciones

**Firma del
 Administrador de
 configuración**

133



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



Formato de notificación de configuración .

Siglas del sistema _____
 Responsable _____

Ciclo _____
 Fecha _____

No. Cambio en Configuración: _____

1. Configuración Anterior

Artículo	Com. Asena (D)	Com. Asena (E)
Arquitectura		
Carta de liberación		
Casos de prueba		
Cierre del proyecto		
Clases del análisis		
Clases del diseño		
Componentes		
Configuración de herramientas		
Descripción del patrón de arquitectura		
Descripción de casos de uso		
Diagramas de casos de uso		
Diseño de arquitectura		
Diagrama Entidad Relación		
Entregable		
Estándar de administración de riesgos		
Estándar de Aceptación		
Estándar de administración de cambios		
Estándar de análisis y diseño		
Estándar de codificación		
Estándar de configuración		
Estándar de documentación		
Estándar de evaluación		
Estándar de pruebas		
Estándar de base de datos		
Estándar de requerimientos		
Estándar de herramientas		
Estándar de resolución de problemas		
Formato de asignación de trabajo		

Formato de diseño de base de datos		
Formato de cierre de ciclo		
Formato de distribución		
Formato de monitoreo		
Formato de preguntas		
Formato de prioridad de casos de uso		
Formato de requerimientos		
Formato de validación de entregable		
Herramientas		
Informe de ciclo		
Informe de configuración		
Informe de entregable		
Informe de integración		
Interfaz		
Invitación de pruebas formales		
Lista de herramientas del ciclo		
Manual de configuración		
Manual de instalación		
Manual de usuario		
Manual técnico		
Modelo de análisis		
Modelo de casos de uso		
Modelo de diseño		
Modelo de implementación		
Notificación de configuración		
Paquetes del análisis		
Plan de ciclos		
Plan de Desarrollo de Software		
Plan de integración		
Plan de pruebas		
Realización de casos de uso del diseño		
Realización de casos de uso del análisis		
Solicitud de cambio		
Solicitud de desarrollo de software		
Subsistemas de diseño		
Subsistemas o sistema		
Visión de desarrollo		
Visión Inicial		

TESIS CON
FALLA DE ORIGEN

2. Configuración Actual

Directorio	Contraseña (U)	Contraseña (E)
Arquitectura		
Carta de liberación		
Casos de prueba		
Cierre del proyecto		
Clases del análisis		
Clases del diseño		
Componentes		
Configuración de herramientas		
Descripción del patrón de arquitectura		
Descripción de casos de uso		
Diagramas de casos de uso		
Diseño de arquitectura		
Diagrama Entidad Relación		
Entregable		
Estándar de administración de riesgos		
Estándar de Aceptación		
Estándar de administración de cambios		
Estándar de análisis y diseño		
Estándar de codificación		
Estándar de configuración		
Estándar de documentación		
Estándar de evaluación		
Estándar de pruebas		
Estándar de base de datos		
Estándar de requerimientos		
Estándar de herramientas		
Estándar de resolución de problemas		
Formato de asignación de trabajo		
Formato de diseño de base de datos		
Formato de cierre de ciclo		
Formato de distribución		
Formato de monitoreo		
Formato de preguntas		
Formato de prioridad de casos de uso		
Formato de requerimientos		
Formato de validación de entregable		
Herramientas		
Informe de ciclo		



Informe de configuración		
Informe de entregable		
Informe de integración		
Interfaz		
Invitación de pruebas formales		
Lista de herramientas del ciclo		
Manual de configuración		
Manual de instalación		
Manual de usuario		
Manual técnico		
Modelo de análisis		
Modelo de casos de uso		
Modelo de diseño		
Modelo de implementación		
Notificación de configuración		
Paquetes del análisis		
Plan de ciclos		
Plan de Desarrollo de Software		
Plan de integración		
Plan de pruebas		
Realización de casos de uso del diseño		
Realización de casos de uso del análisis		
Solicitud de cambio		
Solicitud de desarrollo de software		
Subsistemas de diseño		
Subsistemas o sistema		
Visión de desarrollo		
Visión Inicial		

3. Total de cambios: _____

**Firma del
Administrador de
configuración**

100% CON
FALLA DE ORIGEN

137

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



Solicitud de cambio .

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

No. Cambio: _____

Solicitante: _____
Descripción del cambio: _____

Solo llenar por personal de informática

Módulos impactados: _____

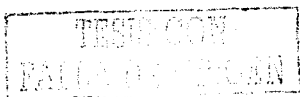
Tiempo estimado del cambio: _____

Resolución tomada: _____

**Firma del
Solicitante**

**Firma del
Coordinador**

**Firma del
Administrador del
proyecto**



1138

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



Formato de notificación de cambio .

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

No. Cambio: _____

Solicitante: _____

Descripción del cambio: _____

Impacto

Valor inicial	Cambio	Impacto estimado	Responsable

**Firma del
Solicitante**

**Firma del
Coordinador**

**Firma del
Administrador del
proyecto**

**Nombre y firma de representante de la mesa de
cambios**

**Firma
Administrador de
cambios**

TRABAJO CON
FALLA DE ORIGEN

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



Formato de verificación de cambios .

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

No. Cambio: _____

Solicitante: _____
Descripción del cambio: _____

Verificación.

Actividad	Tiempo de realización	Cambio realizado(SI/NO)	Comentarios

Observaciones generales: _____

**Firma del
Administrador de
configuración**



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



Formato de Informe de solicitud de cambio para el usuario.

Siglas del sistema _____ Ciclo _____
Responsable _____ Fecha _____

Solicitante: _____

Por medio del presente, el Departamento de Informática le informa de la resolución tomada sobre la petición de cambio hecha el _____ recibida con número de cambio _____.

Módulos impactados: _____ _____
Tiempo estimado del cambio: _____
Resolución tomada: _____ _____ _____

Sin más por el momento y en espera de cualquier duda o aclaración.

**Representante de
mesa de cambios**

**Firma del
Coordinador**

**Firma del
Administrador del
proyecto**



37/141

**TESIS CON
FALLA DE ORIGEN**

Capítulo 7. Creación de la herramienta de consulta para el seguimiento del proceso de desarrollo de software.

Después de haber hecho la adecuación de cada uno de los flujos de trabajo Implementación, Administración del proyecto y Administración de cambios y configuración; fue necesario integrarlos con los demás flujos que también fueron obtenidos [Salazar, 2003] [Sánchez, 2003], en un sólo medio que permitiera su consulta. Pues para seguir cualquier tipo de proceso, se requiere de una herramienta que sirva de guía en la ejecución. Es por eso que se decidió integrar los flujos de trabajo en un producto de software que facilitara el seguimiento del proceso, además de proporcionar la información y ayuda necesaria de cada uno de los elementos del proceso que son actividades, trabajadores y artefactos.

La creación de ésta *herramienta*, también fue diseñada, estructurada y creada por el equipo de trabajo formado para la adecuación del proceso, por lo que nuevamente fue necesario crear estándares de desarrollo, ya que la herramienta fue construida por separado para cada flujo de trabajo e integrada en una sola, dando como resultado el Proceso de Desarrollo de Software del Instituto de Investigaciones Bibliográficas (PDSIIB) que en realidad es la herramienta.

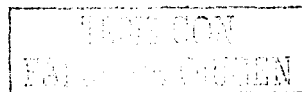
7.1 Objetivo

Uno de los primeros acuerdos que debió tomarse por el grupo de trabajo, para el desarrollo de la herramienta (PDSIIB), fue la determinación de su objetivo. En donde el factor más importante a considerar, fue el tiempo de desarrollo del cual se disponía, ya que se debía hacer algo que no llevara mucho tiempo de implementación para que pudiera estar listo lo más pronto posible y de esa manera comenzar a usarlo. Sin olvidar que por muy sencilla que fuese la herramienta debía servir como guía para la ejecución del proceso.

Para el equipo de trabajo era claro que se necesitaba, por el momento, una forma sencilla que permitiera darle seguimiento al proceso de desarrollo de software, ya que sólo así sería posible probar la efectividad del proceso obtenido durante la adaptación.

Fue así como, el equipo estableció un conjunto de características que debían de contemplarse para el desarrollo del PDSIIB:

- Debe permitir el acceso desde cualquier computadora, para su consulta, por lo que se decidió crear el PDSIIB en una página web, pues de esa forma no importa en que equipo o sistema operativo se consulte, siempre habrá la posibilidad de visualizar su contenido.
- Debe facilitar el manejo y seguimiento del proceso, recordando que las personas que consultarán este proceso, por lo regular son gente del equipo de desarrollo.
- El usuario de la herramienta debe poder identificar claramente, el lugar dentro del proceso en el cual se encuentra en ese momento.



- La herramienta únicamente servirá como consulta o guía del proceso, ya que no permitirá, por el momento, llevar control sobre los proyectos desarrollados.
- Proporcionará en el momento en el que se está consultando una actividad, todos los artefactos, así como la descripción clara y detallada de cada elemento del proceso (rol o artefacto) que interviene en la actividad en cuestión.

Tomando en cuenta este conjunto de características se determinó como objetivo general de la herramienta al siguiente:

La herramienta PSDIIB deberá ser una guía para el seguimiento del proceso de Desarrollo de Software del Departamento de Informática del IIB, proporcionando en cada momento, los artefactos necesarios para su ejecución, así como una explicación clara de cada elemento de este proceso.

7.2 Estructura del PSDIIB

El PSDIIB es el Proceso de Desarrollo de Software del IIB basado en el Proceso Unificado, por lo que es necesario conservar su estructura básica en donde el proceso se divide en fases, que son ejecutadas un número finito de veces.

Fue entonces cuando se identificó el primer problema, ¿Cómo definir el Proceso de Desarrollo de Software del IIB?. Ya que sólo se tenían los flujos de trabajo adecuados en forma independiente y el PU sólo mencionaba las fases, su descripción y que se ejecutaban en forma iterativa hasta obtener el producto final.

Se trataron de integrar los flujos tal cual estaban y en el orden en que se mencionaban en el PU, sin embargo no había cohesión ya que las actividades perdían secuencia. Por dar un ejemplo, al decir que primero era el flujo implementación y después pruebas, nos dimos cuenta que no era posible que después de crear los componentes se hiciera el plan de pruebas unitarias.

Finalmente nos dimos cuenta de que era necesario desmenuzar todos los flujos para formar el PSDIIB en base a las fases que el PU definía, ya que con base al objetivo de cada fase, sería posible determinar y seleccionar qué actividades pertenecían a la fase. Fue así como al terminar la adecuación de todos los flujos, se desmenuzaron para formar un diagrama de secuencia que se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición. Las cuales marcan claramente el avance del proceso de desarrollo de software ya que cada fase tiene un objetivo específico, de tal forma que mientras no se cumple el objetivo de la fase no es posible seguir avanzando.

Este trabajo fue muy interesante, ya que fue necesario nuevamente ir revisando actividad por actividad para determinar su secuencia dentro de la fase y en consecuencia dentro del proceso. Personalmente, esta experiencia me ayudó a comprender más al PU, ya que finalmente, pude darme cuenta de que el PU es una secuencia de actividades divididas en fases, que pueden ser clasificadas por flujo de trabajo para comprender con más facilidad cual es su objetivo, sin embargo la clasificación de estas actividades sólo tienen un fin didáctico, ya que en la práctica están mezcladas formando el proceso de desarrollo de software.

El resultado del trabajo antes mencionado dió como resultado los diagramas de actividades de cada fase, las cuales se pueden ver en las figuras 1, 2, 3 y 4.

Figura 1. Fase de Iniciación del PDSIIB.

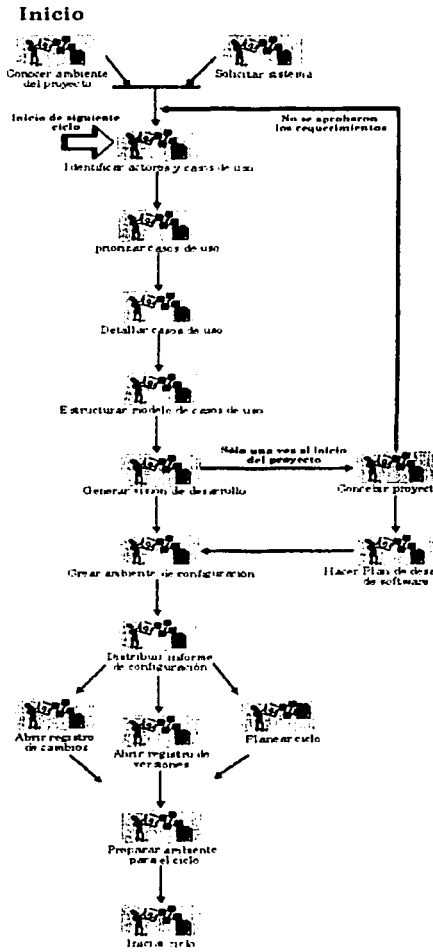


Figura 2. Fase de elaboración del PDSIIB.

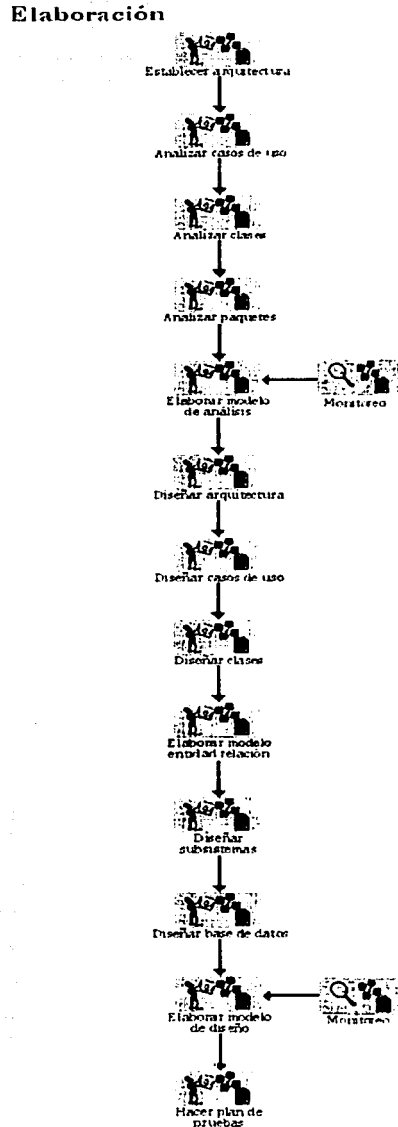


Figura 3. Fase de Construcción del PDSIIB.

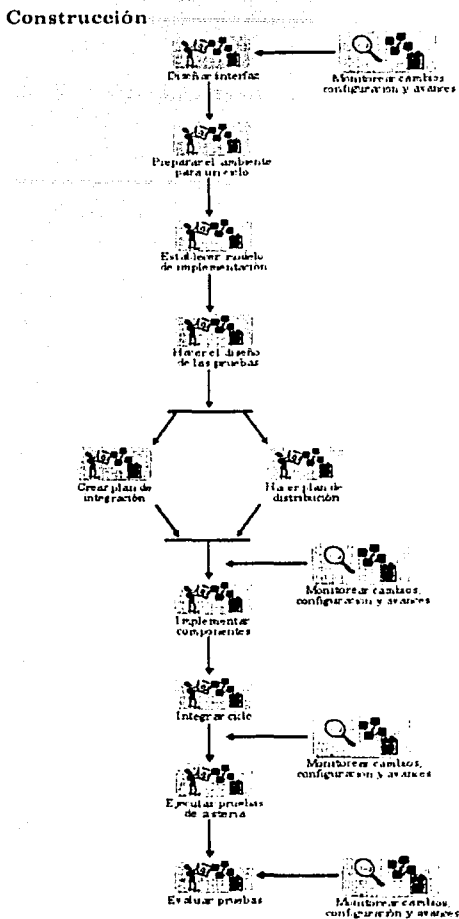
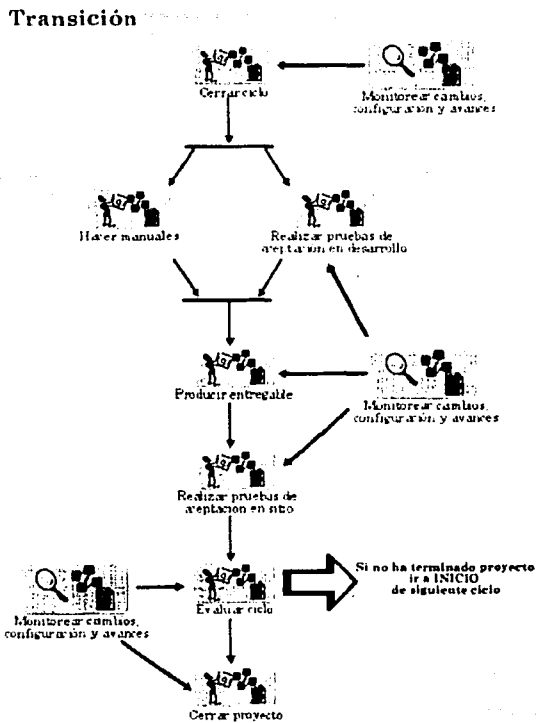
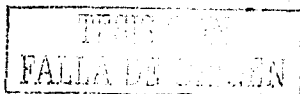


Figura 4. Fase de transición del PDSIIB.



Después de crear la estructura básica del PDSIIB, fue necesario establecer los parámetros técnicos de construcción, por lo cual se llegó a los siguientes acuerdos:

- La herramienta se hizo en HTML, de tal forma que la consulta puede hacerse desde cualquier navegador y en consecuencia desde cualquier equipo o sistema operativo.



- El PDSIIB se dividió en flujos y fases, con la intención de conservar el objetivo didáctico de los flujos y además establecer la secuencia clara del Proceso de desarrollo de software del Departamento de Informática. Esta herramienta, a diferencia de la herramienta RUP no sólo muestra el contenido de cada flujo, además muestra gráficamente cómo es el proceso a partir de las fases que el mismo Proceso Unificado establece (ver figura 5).

Figura 5. Las fases del PDSIIB.



- Se estableció un estándar de diagramación, para la construcción de diagrama general. Se diseñaron los iconos para cada uno de los elementos del proceso de tal forma que cada miembro del equipo de trabajo usara la misma simbología para crear sus diagramas (ver figuras 6, 7,8 y 9):

Figura 6. Elemento simbólico que representa una actividad



Figura 7. Elemento simbólico que representa un artefacto de entrada



Figura 8. Elemento simbólico que representa un artefacto de salida.



Figura 9. Elemento simbólico que representa una actividad de monitoreo.



- El responsable de cada flujo, creó los diagramas, artefactos y descripciones necesarias, de tal forma que al crear el diagrama de cada fase, sólo se hace referencia a la información establecida por las actividades involucradas en la fase.

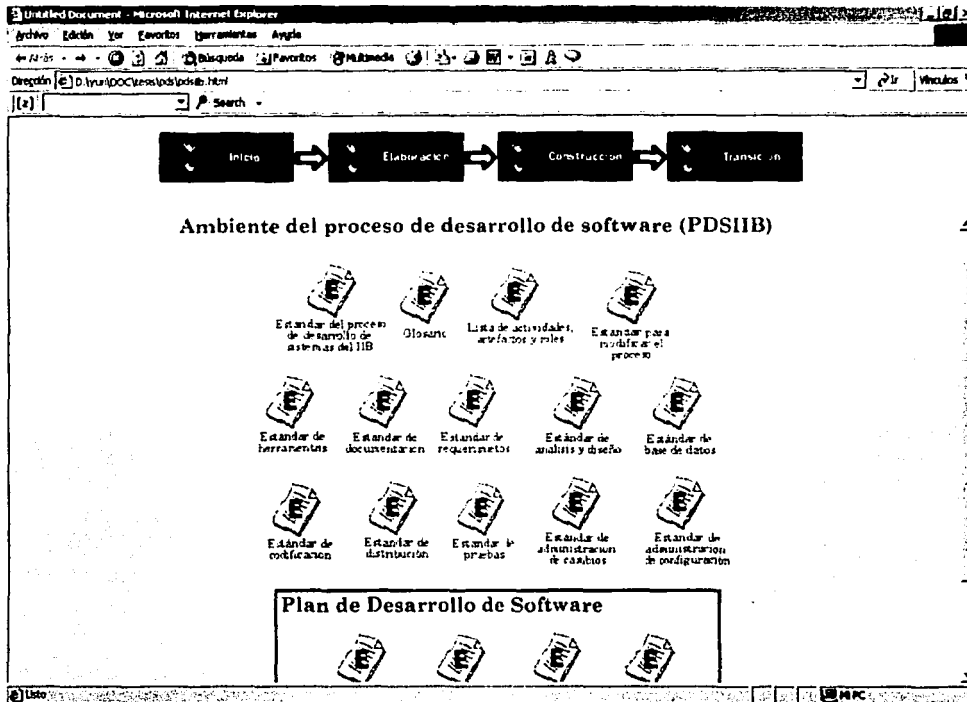
El establecimiento de todos estos estándares permitió al equipo trabajar en forma individual e integrar en forma transparente la herramienta al final.

7.3 Funcionamiento.

Como se mencionó la herramienta para el Proceso de desarrollo de Software del IIB, es una página electrónica a la cual se puede acceder en la dirección <http://132.248.77.248/psdiib> únicamente estando dentro del intranet del instituto por ser sólo para uso del personal del Departamento de Informática.

Al entrar a la herramienta (PDSIIB), la primer página muestra cómo se compone el ambiente del proceso, además de las fases en las que se divide el mismo (ver figura 10).

Figura 10. Primer pantalla de la herramienta del PDSIIB.

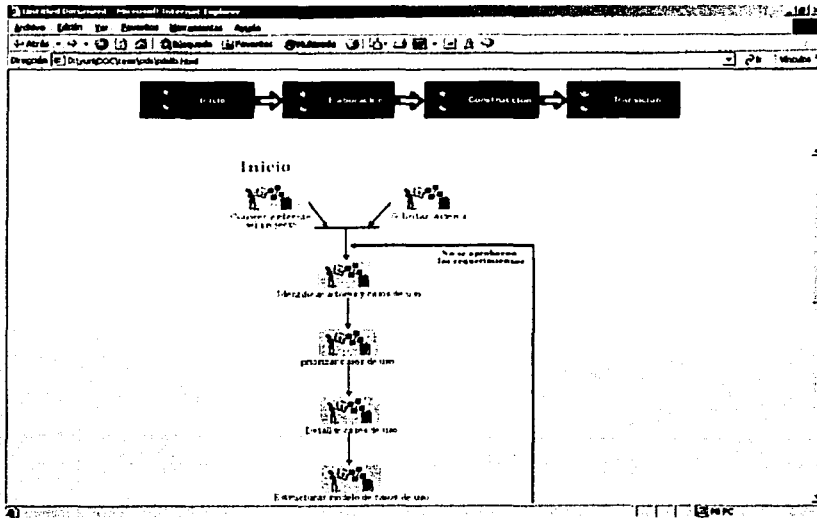


Debido a que la herramienta esta hecha para guiar el proceso de desarrollo de software del Departamento de Informática, para dar inicio en el seguimiento del proceso debe seleccionarse la fase de Inicio, cuando ese sea el caso, o si ya se encuentra en ejecución el proceso la fase en la que se esté en ese momento. Al dividir la elaboración de esta herramienta, se me asignaron las fases de Inicio y Construcción, es por eso que la ejemplificación del funcionamiento del PDSIIB se harán con estas dos fases.

Seleccionada la fase en la que se está ejecutando el proceso, es posible ver las actividades que deben ejecutarse para cumplir el objetivo de ésta (ver figura 11).

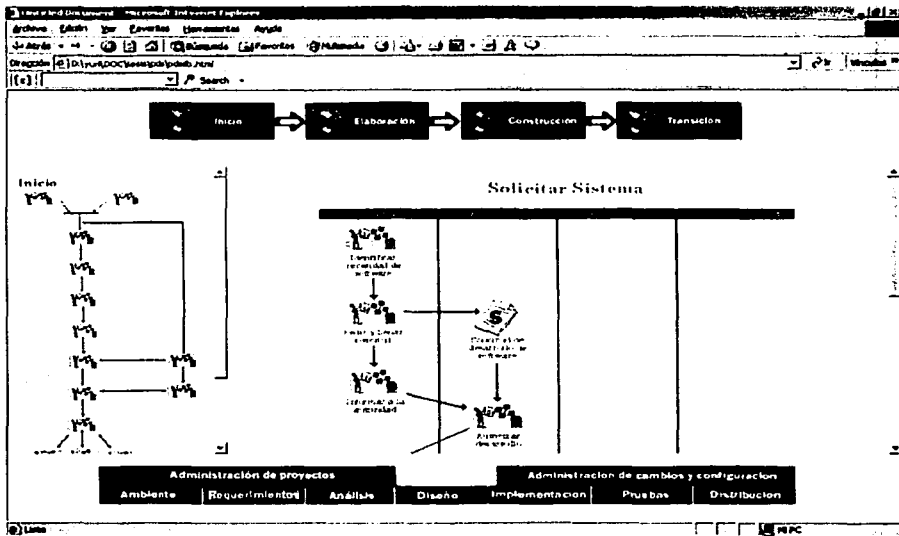
TELECOM
FALLA DE COMUNICACION

Figura 11. Fase de Inicio en el PDSiB.



Dentro de alguna fase el usuario sólo debe seleccionar la actividad que desea consultar, para conocer los pasos de la misma (ver figura 12).

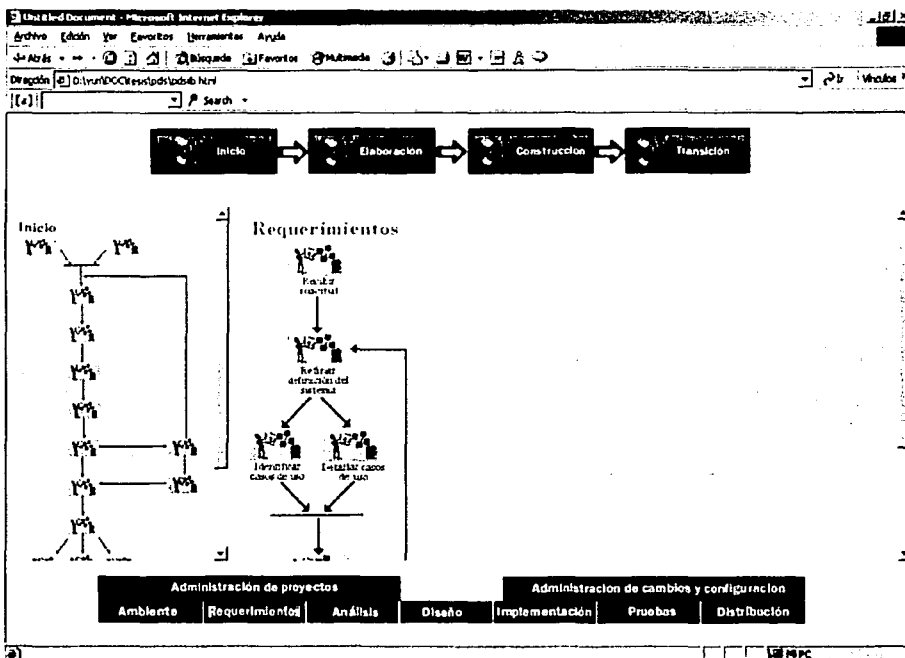
Figura 12. Primer actividad de la fase Inicialización del PDSiB.



Dentro de alguna actividad la pantalla se divide en cuatro zonas: la parte superior muestra las fases de las cuales se compone el proceso, marcando de un color diferente la fase actual; en la parte central a la izquierda se muestra una reducción del diagrama de las actividades de la fase actual, en donde se marca con un círculo de color, la actividad actual; la parte central derecha, se muestran los pasos a seguir y los artefactos, además de marcar con carriles los roles que intervienen; y en la parte inferior se muestran los flujos de trabajo, marcando con un color diferente el flujo al que pertenece la actividad actual.

Una vez dentro de la descripción de una actividad, puede ser consultado el flujo de trabajo al que pertenece, seleccionando el que se encuentra marcado con otro tono (ver figura 13).

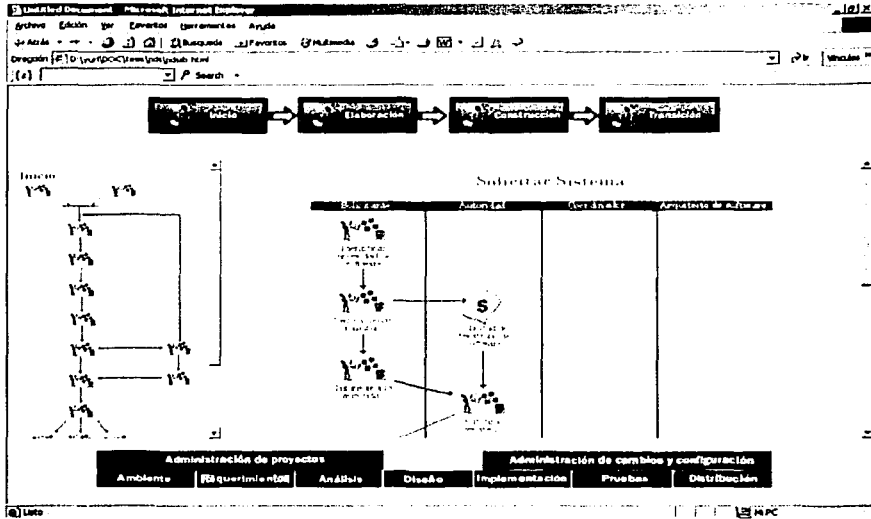
Figura 13. Flujo de trabajo Requerimientos, el cual contiene a la actividad Recibir solicitud.



Dentro de la descripción del flujo de trabajo, se puede consultar la misma actividad o alguna otra del flujo seleccionándola, y entonces se mostrará nuevamente la información de la actividad. Esto permite que el usuario sepa cual es la secuencia de cada actividad dentro del PDSIIB y a su vez conozca el flujo de trabajo al que pertenece la actividad (ver figura 14).



Figura 14. Primer actividad del flujo de trabajo Requerimientos dentro del PDSIIB .



En la herramienta es posible consultar la descripción detallada de cada paso, artefacto o actor, de tal forma que toda la información que se necesita saber se encuentra explicada y en su caso ejemplificada dentro de cada actividad (ver figura 15).

Figura 15. Formato de requerimientos dentro del PDSIIB.

Finalmente sólo resta decir que a pesar de que la herramienta tiene sus limitaciones, pues sólo permite consultar cada elemento del proceso y no interviene en la administración

del mismo. Eso no significa que no pueda utilizarse como parte de la administración, pues establece una línea a seguir y eso ayuda también al administrador, aun que no le facilite el trabajo, si le permite entender cuáles son sus funciones.

Además esta herramienta puede ser utilizada para capacitar a los ingenieros de software en el entendimiento del PU, y a su vez en el aprendizaje del Proceso de Desarrollo de Software del Departamento de Informática del IIB.

TESIS CON
FALLA DE ORIGEN

Conclusiones.

Es importante recalcar que Adoptar el Proceso Unificado como proceso de desarrollo de Software en una empresa u organización, no es trivial

Este trabajo forma parte de un proyecto de Adaptación del Proceso Unificado al Proceso de desarrollo de Software de una organización de tal forma que el proceso obtenido, conserva los elementos que conforman la estructura fundamental del PU y sólo consume los recursos con los que cuenta la organización seleccionada.

La posibilidad de Adaptar el Proceso Unificado para ser utilizado en una organización en donde su objetivo principal no es el desarrollar software, es factible bajo las siguientes condiciones:

- El grupo de trabajo debe conocer el Proceso Unificado.
- El grupo de trabajo debe conocer el funcionamiento de la organización para la cual se hará la adaptación, con la mayor profundidad posible. Este es un punto muy importante ya que el trabajo requiere la modificación y ajuste del Proceso Unificado a un ambiente que debe conocerse perfectamente para poder aproximarse a su realidad.
- La división de trabajo es conveniente para realizar la adecuación, sin embargo será necesario hacer una revisión grupal de cada flujo adecuado, para evitar que los resultados individuales sólo contemplen un punto de vista.
- Una vez hecha la adecuación de cada flujo será necesario revisar las intersecciones que existen entre los flujos, para evitar que haya pérdida de relaciones, lo cual podría evitar la integración de los flujos en un solo proceso.
- Durante la adecuación de cada flujo de trabajo es indispensable tomar en cuenta dos parámetros, por un lado no perder de vista la situación real que se vive dentro de la organización, durante el desarrollo de software ya que eso da las restricciones que se tienen y por otro lado, el objetivo planteado por el Proceso Unificado para el flujo en cuestión, ya que eso nos da los requerimientos mínimos que debe cumplir la adaptación.
- Una vez hecha la adaptación de cada flujo será necesario generar el esquema general del proceso de Desarrollo de Software.
- Con base en el proceso armado y dividido por fases, debe generarse un instrumento de consulta que permita, re-usar la secuencia de ejecución del proceso obtenido.



- Por último, lo que resta es ejecutar el proceso para probar su eficiencia y así identificar los puntos en los que se deberán efectuar cambios, siguiendo las mismas condiciones de adecuación.

Por otro lado, se puede decir que los flujos correspondientes a la administración son los que requieren mayor atención y cuidado, ya que por lo regular en organizaciones pequeñas, muchas de las actividades marcadas por éstos flujos no se realizan. Por lo que es difícil determinar hasta qué grado puede aplicarse la administración, sin caer en algo demasiado tedioso y difícil de llevar a cabo, o por otro lado resulte tan trivial, que no cumpla con su objetivo.

Otro punto importante para la adaptación de los flujos de administración, es la selección y uso de herramientas, ya que para muchas de las actividades de estos flujos ya existen, es por eso que la elección de las herramientas a utilizar debe hacerse durante el análisis de la adaptación para determinar que tan útil, factible y necesaria es su adquisición.

El resultado de la adaptación de cada uno de los tres flujos de trabajo, que trata esta tesis, tiene las siguientes diferencias con respecto al RUP:

- En el flujo de trabajo Implementación se realizaron las siguientes variantes:
 - ✓ Se creó un diagrama de actividades para el flujo de trabajo, así como para cada actividad.
 - ✓ Se eliminó la actividad Integrar subsistema ya que ésta se integró como un paso dentro de la actividad Integrar componentes.
 - ✓ Se adecuaron los nombres con base a los estándares establecidos para el proyecto.
 - ✓ Se definieron claramente los pasos para cada actividad, con base al objetivo de la misma planteado en el PU.
 - ✓ Se establecieron claramente los artefactos generados en cada actividad.
 - ✓ Se creó una descripción detallada de cada artefacto, además de formatos necesarios para recolectar sólo la información necesaria.
 - ✓ Se eliminó el rol Supervisor de código ya que sus responsabilidades se integraron al rol programador.
 - ✓ Se crearon tres artefactos nuevos: Formato de Análisis de componentes, Registro de verificación y el Informe de integración
- En el flujo de trabajo Administración del proyecto las variantes fueron:
 - ✓ Fue eliminada la actividad Evaluar riesgos.
 - ✓ Se adecuaron los nombres de acuerdo a los estándares establecidos.
 - ✓ La información de los artefactos obtenidos es concreta, ya que únicamente se guardan los datos importantes para la organización.
 - ✓ Muchos de los artefactos que componen el Plan de Desarrollo de Software se convirtieron en estándares, debido a que fue posible generalizar el tipo de aplicaciones que se construyen dentro del DIIB. Los cuales son: Estándar de aceptación de sistemas, Estándar de resolución de problemas, Estándar de evaluación, Estándar de Administración de riesgos, Lista de riesgos.



- ✓ Lo anterior hace muy fácil la creación del Plan de desarrollo de Software de cada proyecto.
- ✓ La actividad Administrar iteración se dividió en dos Iniciar ciclo y evaluar ciclo, debido a que no es posible realizar todas las actividades de Administración de iteración que el PU establece.
- ✓ Se crearon formatos que auxilian en la creación de los artefactos y en algunos casos son los artefactos mismos.
- ✓ El cierre de ciclo se hará en el momento en que se cumpla con lo planeado en el Plan del ciclo, de tal forma que los tiempos estimados, únicamente servirán como un parámetro que permita mejorar la estimación de tiempos par los consecutivos ciclos ó proyectos.
- En el flujo de trabajo Administración de la configuración y control de cambios las variantes aplicadas fueron:
 - ✓ El flujo se divide en dos proceso, la Administración de cambios y la Administración de la configuración.
 - ✓ El plan de Administración de cambios y configuración se convirtió en dos estándares: Estándar de administración configuración y Estándar de administración de cambios.
 - ✓ La administración de versiones, por el momento es "rudimentaria", ya que se hace a mano, sin embargo establece un comienzo, lo más formal y menos complicado posible, para la administración de versiones y el control de cambios.
 - ✓ El reporte de auditoria se convirtió en tres artefactos: Informe de configuración, notificación de configuración y monitoreo de configuración y cambios.
 - ✓ Se creó un artefacto llamado Informe para el usuario en el cual se comunica de la resolución de un cambio mayor.
 - ✓ Finalmente se estableció un formato para la solicitud de cambios.

De todo lo anterior pueden obtenerse las siguientes conclusiones:

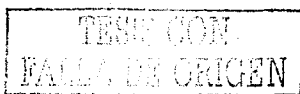
- ✓ Los elementos que se consideraron como factores primordiales, para determinar los elementos de la adecuación, pueden considerarse como generales para muchas organizaciones pequeñas en México.
- ✓ La colaboración de la gente que trabaja en la organización en la que se crea el proceso es fundamental, pues son ellos los que definen el contenido del proceso.
- ✓ La adecuación de cualquier flujo de trabajo jamás se puede hacer en forma aislada a los demás elementos de Proceso de Desarrollo de software, por lo que es necesario hacer una revisión grupal de cada flujo adecuado, para no perder la cohesión del proceso.
- ✓ La incorporación de herramientas automáticas para la administración es importante, sin embargo es más importante primero lograr que la gente esté dispuesta a administrar, por lo que en algunos casos es necesario comenzar la administración de la manera más rudimentaria.

- ✓ Durante la adaptación uno de los trabajos más pesados fue la definición de los formatos para cada artefacto, ya que se hizo con la intención de que todos fuesen lo más concretos y adecuados a las necesidades del departamento.
- ✓ Todo proceso requiere de un instrumento que guíe su ejecución por lo que ningún trabajo de creación de procesos está completo sin la creación de su instrumento o herramienta de apoyo.
- ✓ El trabajo para la creación de la herramienta, no sólo incluye un esfuerzo de programación, además es necesario armar un proceso con base a los flujos adaptados, despedazando cada flujo para crear un proceso dividido por fases que incluyera todas las actividades de cada flujo.

Trabajos futuros.

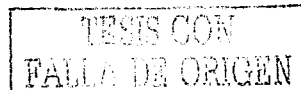
Este trabajo deja algunos puntos por refinar como son:

- ✓ Las pruebas del proceso obtenido para el Departamento de Informática del IIB, no se han concluido, por lo que queda pendiente para formar parte de los trabajos futuros del proyecto de adecuación del PU a un departamento.
- ✓ Con base a lo anterior, será necesario determinar si es conveniente usar una herramienta más sofisticada para la administración o en su caso si es posible modificar la herramienta que se creó para el seguimiento del proceso (PDSIIB), de tal forma que también permita cierto grado de administración del mismo.



Bibliografía.

- [Booch, 1999]. Booch, G. El lenguaje unificado de modelado. Madrid : Addison Wesley Iberoamericana, 1999. 438 p.
- [Curtis, 1992]. Curtis M. Bill. Kellner, I. and Over, J. Process Modeling. Communications of the ACM, vol. XXXV, núm. 9, 1992, pp. 75-90.
- [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.
- [Huff, 1996]. Huff, K. E. Software Process Modeling. Trends in Software, 1996, pp. 1-24.
- [Jacobson, 1992] Jacobson, I., Cristerson, M., Jonson, P., and Övergaard, G., *Object Oriented Software Engineering: A Use-Case Driven Approach*, Reading, MA: Addison Wesley, 1992.
- [Jacobson, 2000]. Jacobson, I., Booch G. y Rumbaugh J. El proceso unificado de desarrollo de software. Pearson Educación, S.A., 2000.
- [James, 1986] James E. Archer Jr. y Michael T. Devlin, *Rational's Experience Using Ada for Very Large Systems, Proceedings of the First International Conference on Ada Programming Language Applications for the NASA Space Station*, (Junio 1986).
- [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.
- [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.
- [Sánchez, 2003]. Sánchez Mendoza, Ma. Guadalupe. Adaptación del proceso unificado en los flujos de Requerimientos, Análisis y Diseño: Caso práctico México : UNAM, 2003, pp. 176.
- [UCM, 2003] . Sitio de la Universidad Carlos tercero de Madrid. <http://www.ic.inf.uc3m.es/Tesis/Documentos/ProcesosSoftware/Procesos1/Procesos1.html>



Ejemplo del llenado de artefactos (Implementación)

TESIS CON
FALLA DE ORIGEN

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



Formato de recopilación de información para el modelo de implementación.

Siglas del sistema PESIMEX Ciclo 1
 Responsable JASC Fecha _____

(Llenar un formato por cada componente, en el lenguaje de programación)

Nombre del componente: BibliografiaDP

1. Proviene de la biblioteca de componentes (Llenar sólo si se toma de la biblioteca de componentes).

Nombre del componente original: _____

2. Componentes comerciales que necesita.

Nombre de función o librería	Descripción de funcionamiento:	Empresa desarrolladora o producto del que proviene
Common Dialog Control CMDLG32.OCX	OCX para la creación de botones, cajas de dialogo.	Microsoft Visual Basic.
Rich Textbox Control RICHTX32.OCX	OCX para el manejo de cajas de texto para archivos .rtf	Microsoft Visual Basic
Windows Common Control COMCTL32.OCX	OCX para el manejo de ventanas con funcionamientos especificos, como guardar archivos, cambiar colores.	Microsoft Visual Basic

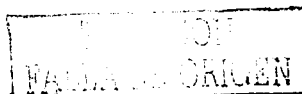
3. Descripción del componente.

Tipo de componente				
Clase		Función	Procedimiento	Interfaz

Objetivo: Permite almacenar en tiempo de ejecución la información de una ficha bibliográfica

Componentes con los que se asocia para su funcionamiento	
Nombre del componente	Tipo de asociación
BIBLIOGRAFIAMD	envia la información para la base de datos, o recibe lo que hay en la base de datos
VENTANACAPTURA	Medio por el cual ingresan los datos

Datos que Recibe		Datos que regresa	
Nombre	Tipo	Nombre	Tipo
vrSAutor	string	vrSAutor	string
vrSTitulo	string	vrSTitulo	string



159

Variables o atributos					
Nombre	Tipo	Alcance (global ó local)	Valor inicial	validaciones	Rangos

Métodos (sólo para clases)				
Nombre del método	Datos que Recibe		Datos que regresa	
	Nombre	Tipo	Nombre	Tipo

Restricciones	
Validaciones	Variables globales (externas)

Código (Si se trata de un componente reutilizado, copiar aquí el código) (Si se trata de una clase duplicar esto para cada método de la clase)	
Nombre del método:	
Type BibliografiaDP vrSAutor As String vrSTitulo As String vrSTema As String vrSPielmp As String End Type	

Interfaz (en caso de no tener interfaz omite esta parte)	
Nombre de Interfaz:	VentanaCaptura

TESIS CON
FOLIO DE ORIGEN

12/160

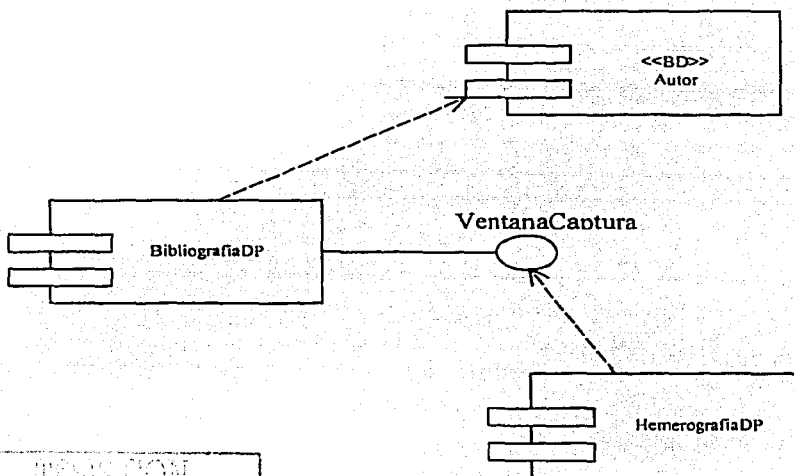
Modelo de Implementación.

Siglas del sistema PESIMEX Ciclo 1
Responsable JASC Fecha

Componentes.

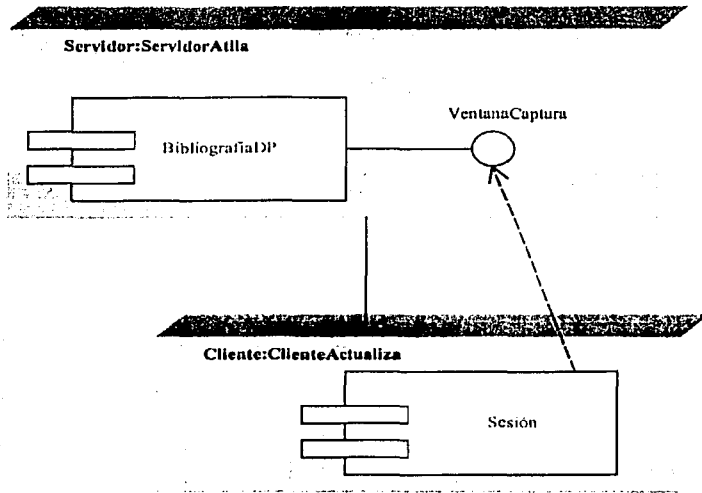


Diagrama de componentes.



TIENE CON
FALLA DE ORIGEN

Vista de despliegue.



Anexar: formato de recopilación de datos para el modelo de implementación, de cada componente.

TESIS CON
FALLA DE ORIGEN

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



Plan de Integración.

Siglas del sistema PESIMEX Ciclo 1
 Responsable JASC Fecha _____

1. Integración del caso de uso

1.1. Lista de casos de uso: (establecer importancia con base a las dependencias funcionales)

Valor de importancia (1, 2, 3, etc.)	Siglas	Nombre de Módulo o Caso de Uso
1	ENT	Entrar
1	SAL	Salir
2	CAP	Captura

1.2 Niveles de Integración (Indique más de uno por nivel, con base a su importancia):

Siglas de los casos de uso	#Nivel
ENT, SAL	1
CAP	2
	3
	4

1.3 Pruebas por nivel de integración (Definir con base al Plan de Pruebas).

Siglas de los casos de uso	Nombre del(os) caso(s) de prueba	#Nivel
ENT, SAL		1
CAP		2
		3
		4

2. Integración de cada caso de uso

Nombre del caso de uso: ENTRAR (ENT)

2.1 Lista de componentes del caso de uso:

Valor de importancia (1, 2, 3, etc.)	Siglas	Componente (clase, función, procedimiento)
5	VTNP	VENTANAPRINCIPAL
4	VTNPS	VENTANAPASSWORD
1	LPS	LEEPASSWORD
2	VPS	VERIFICAPASSWORD
3	VTNE	VENTANAERROR

2.2 Niveles de Integración (Indique más de uno por nivel, con base a su importancia):

Siglas de los componentes	#Nivel
LPS, VPS	1
VTNP,VTNPS,VTNE	2
	3
	4

2.3 Pruebas por nivel de integración (Definir con base al Plan de Pruebas).

Siglas de los componentes	Nombre del(os) caso(s) de prueba	#Nivel
LPS, VPS		1
VTNP,VTNPS,VTNE		2
		3
		4

TESIS CON
FALLA DE CONTEN

211 164

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Formato de verificación de código.

Siglas del sistema PESIMEX Ciclo 1
 Responsable JASC Fecha _____
 Siglas del caso de uso ENT Siglas del componente o sub-componente LPS

(La revisión puede hacerse de dos formas, directamente sobre el código o sobre el formato. En caso de hacerse sobre el formato debe añadirse una impresión del código fuente al iniciar la verificación y únicamente marcar en forma independiente sobre el código cada objeto verificado)

Para cada variable realice el siguiente análisis:

Nombre de variable	vrSPassword	Objetivo o utilidad
	Tipo (Si la verificación es sobre el código sólo marque la declaración de la variable)	(Si la verificación es sobre el código haga comentarios sobre el mismo)
String		Almacena la cadena que se recibe como password

(Si la verificación es sobre el código sólo subraye sobre el código y registre los errores sobre el formato)

Líneas en las que aparece la variable (verifique cada línea en la que aparece la variable e indique)		
No. línea (No se requiere cuando la verificación es sobre el código)	Estructura de control o instrucción (Asignación, condición, inicialización, cicloWhile, cicloFor, cicloDoWhile, incremento, decremento, comparación)	Error (Sólo si hay errores indique con una "x" para facilitar el conteo final)
6	If vrSPassword=""&W95" then	x

Total de errores encontrados: 1

Para cada estructura de control realice el siguiente análisis:

(Si la verificación es sobre el código sólo subraye sobre el código y registre los errores sobre el formato)

Tipo de estructura (condición, cicloWhile, cicloFor, cicloDoWhile)	No. de línea del inicio	No. de línea del fin	Error
condición	6	10	

Total de errores encontrados: 0

TRAM CON
FALLA DE ORIGEN

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



Formato de configuración del sistema.

Siglas del sistema PESIMEX Ciclo 1
 Responsable JASC Fecha _____

Base de datos			
Servidor	Directorio	Nombre de la base de datos	Driver de conexión (ODBC, JDBC, etc)
PC LOCAL	C:\PESIMEX	PESIMEX.MST	

Unidades de red		
Equipo al que debe conectarse	Directorio	Tipo de acceso (R/W)

Software adicional.		
Nombre	Directorio de instalación	Configuración

Localización del sistema			
Unidad	Directorio	Archivo ejecutable	Subsistema relacionado
C:\	PESIMEX	PESIMEX.EXE	CAPTURA

Estructura de directorios que componen el sistema		
Unidad	Directorio	Contenido
C:\PESIMEX	BASE	BASE DE DATOS
	INIS	ARCHIVOS INIS

Archivos que componen el sistema			
Nombre del archivo	Tipo (txt, ini, dat, etc.)	Contenido	Estructura
REGISTRO.INI	INI	REGISTRO ACTUAL DE CAPTURA	<ETIQUETA>=<CONTENIDO>
PESIMEX.FST	FST	PARÁMETROS DE INDEXACIÓN	<#> <TIPO DE INDX> <TIPO EXTRACCION> <ETIQUETA>

20166



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



Informe de Integración.

Siglas del sistema PESIMEX Ciclo 1
Responsable JASC Fecha _____

Número del último nivel de integración alcanzado: 2

Nivel #	Se aprobaron las pruebas (SI, No)	Comentarios
1	SI	
2	SI	

Llenar sólo si no se aprobaron las pruebas.		
Nivel #	Caso de uso	Problemas

Anexar casos de prueba no superados y reporte de evaluación de pruebas.

Integrador

Firma

TESIS CON
FALLA DE ORIGEN

Ejemplo del llenado de artefactos (Administración del proyecto)

TESIS CON
FALLA DE ORIGEN

17/68

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



Visión Inicial.

Nombre del software: Sistema de Administración de Información para las publicaciones sobre la independencia de México

Problemática a resolver: Contar con una base de datos y los mecanismos que permitan almacenar y controlar la información recabada sobre la Independencia.

Objetivo del sistema: Almacenar y controlar la información recabada sobre la Independencia

Módulos de los que se compondrá el sistema:

Nombre	Descripción	Algoritmo	Usuario	Orden
Administración bibliográfica	Proporcionar una interfaz que permita, Guardar, modificar e imprimir los registros bibliográficos	Bibliografía	Bibliotecario	1
Administración Hemerográfica	Proporcionar una interfaz que permita, Guardar, modificar e imprimir los registros hemerográficos	Hemerografía	Bibliotecario	2
Administración de folletos	Proporcionar una interfaz que permita, Guardar, modificar e imprimir los registros folletos	Folletos	Bibliotecario	3

Interacción con otros sistemas:

Nombre del sistema	Aplicación

Características no funcionales:

1. Plataforma.

UNIX

MSDOS

WINDOWS

MAC

2. Ambiente.

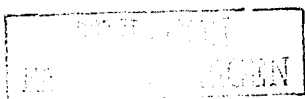
REDLOCAL

MONOUSUARIO

INTRANET

INTERNET

Recursos necesarios.



1. Personal del departamento de Informática y de fuera que participa.

Nombre	Rol
José Antonio Salazar Carmona	Administrador de cambios
José Antonio Salazar Carmona	Administrador de configuración
José Antonio Salazar Carmona	Administrador de distribución
José Antonio Salazar Carmona	Administrador del proyecto
José Antonio Salazar Carmona	Analista de sistemas
José Antonio Salazar Carmona	Arquitecto de software
José Antonio Salazar Carmona	Diseñador de bases de datos
José Antonio Salazar Carmona	Diseñador de pruebas
Guadalupe Sánchez	Escritor técnico
José Antonio Salazar Carmona	Especialista en herramientas
José Antonio Salazar Carmona	Especificador de casos de uso
José Antonio Salazar Carmona	Ingeniero de casos de uso
José Antonio Salazar Carmona	Ingeniero de componentes
José Antonio Salazar Carmona	Integrador
Ana Yuri Ramírez Molina	Probador
Paulo Cesar	Usuario responsable
José Antonio Salazar Carmona	Supervisor del proyecto

2. Software y Hardware necesario:

Nombre	Versión	Actividad en la que se usará	Responsable de su adquisición
Windows NT	4.0	Pruebas	Informática
Nodo de Red (en sitio)	Rj45	Pruebas en sitio	Usuario responsable
Windows	XP	Pruebas	Usuario responsable, Informática

3. Capacitación:

Nombre	Versión	Lugar	Duración

4. Estimación de tiempos (ver formato anexo "gannt.xls").

Después de leer cuidadosamente y aclarar cualquier duda que pudiera surgir, referente al software solicitado al Departamento de Informática, acepto todo lo establecido anteriormente como parte de mi solicitud, además de aceptar todas las responsabilidades que esto conlleva. Quedando conforme respecto a los tiempos de entrega y en el entendido de que el Departamento de Informática se compromete a cumplir con lo establecido asignando el apoyo y los recursos que por su parte son necesarios.

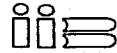
Nombre y firma
Coordinador

Nombre y firma del
solicitante

Nombre y firma del
administrador del proyecto



101 170



Formato de validación de entregable.

Siglas del sistema PESIMEX Ciclo 1
Responsable JASC Fecha

1. Manual de usuario (Marque con una "x", los puntos que se cumplen).

- Cumple con los márgenes establecidos
- Cumple con los tipos y tamaños de letra establecidos.
- Al seleccionar uno de los procesos descrito dentro del manual de usuario, fue posible entenderlo y ejecutarlo, sin la ayuda de nadie.

2. Manual técnico (Marque con una "x", los puntos que se cumplen).

- Cumple con los márgenes establecidos
- Cumple con los tipos y tamaños de letra establecidos.
- Al seleccionar uno de los documentos que componen el manual y comparar el documento correspondiente de la última versión almacenada en el repositorio de versiones, eran idénticos.

3. Manual de Instalación (Marque con una "x", los puntos que se cumplen).

- Cumple con los márgenes establecidos
- Cumple con los tipos y tamaños de letra establecidos.
- Al seguir al pie de la letra lo indicado en el manual para la instalación, fue posible entenderlo e instalar el software, sin la ayuda de nadie.

4. Manual de Configuración (Marque con una "x", los puntos que se cumplen).

- Cumple con los márgenes establecidos
- Cumple con los tipos y tamaños de letra establecidos.
- Al seguir al pie de la letra lo indicado en el manual para la configuración, fue posible entenderlo y configurar el software, sin la ayuda de nadie.



5. Documentación de Requerimientos y cambios al sistema (Marque con una "x", los puntos que se cumplen).

- Al revisar todos los documentos de requerimientos que necesitan de la autorización del solicitante, no se encontró ninguna omisión de firma.
- Al revisar las verificaciones de cambios aceptados, no se encontró ninguna observación pendiente.
- Al revisar los resultados de todos los monitoreos, no se encontró ninguna observación pendiente.

6. Funcionalidad del software (Marque con una "x", los puntos que se cumplen)..

- Al hacer la revisión de las pruebas del sistema en desarrollo y en producción, no se encontró ningún problema pendiente.
- Al seleccionar uno de los procesos descrito dentro del manual de usuario, fue posible ejecutarlo, sin ningún problema.

7. Empaquetado (Marque con una "x", los puntos que se cumplen)..

- Al instalar el sistema en el equipo del solicitante, con las características necesarias para la ejecución del sistema, no hubo problemas, por el medio de almacenamiento en el que se encontraba el paquete de instalación.

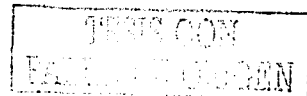
8. Resultado (Marque con una "x", los puntos que se cumplen)..

Existe alguna opción de las anteriores que no se haya cumplido:

SI	NO
	x

SI la respuesta de la pregunta 8 es SI, el producto no es aceptable.

**Firma del
Administrador del
proyecto**



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



Formato para el registro de problemas.

Siglas del sistema PESIMEX Ciclo I
Responsable JASC Fecha _____

Tipo de problema:

- Problemas con el solicitante. Problemas de recursos. Problemas de conocimiento. Problemas personales.

Descripción del problema: No se cuenta con una máquina XP para las pruebas en desarrollo

Firma del jefe del Departamento: _____

Fecha de recepción: 10/05/2002

Fecha de reunión con el solicitante: 10/05/2002

Solución: Se ocupará uno de los equipos que llegan por nueva adquisición de principios de junio

Firma del
Administrador del
proyecto

Firma del
Coordinador

TESIS CON
FALLA DE ORIGEN

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



Plan del Ciclo.

Siglas del sistema PESIMEX Ciclo 1
 Responsable JASC Fecha _____

Responsable para la entrega de avances: Paulo cesar

Objetivo del ciclo: Administración Bibliográfica

Establecer composición del ciclo (tomar de la Visión de desarrollo y Visión inicial):

Ciclo#	Casos de Uso (módulos)	Prioridad del Caso/Módulo	Fecha de entrega
1	Entrar	1	15/07/2002
	Capturar	2	15/07/2002
	Salir	1	15/07/2002

Recursos necesarios (tomar de la Visión de inicial):

Recursos humanos	
Nombre	Actividad
JASC	DISEÑO, ANALISIS
JASC	ANALISIS, DISEÑO

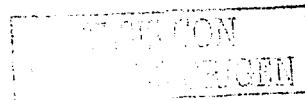
Nombre
OBJETEERING
WORD

- Estimación de tiempos (ver formato anexo "gannt.xls").
- Anexo Formato de Asignación de trabajo
- Anexo Formato de Cierre de ciclo.
- Anexo Formato de Monitoreo de Avances.
- Anexo Informe del Cierre de ciclo.
- Anexo Informe de ciclo

 Firma del
 Coordinador

 Firma del
 Administrador del
 proyecto

174



Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Formato de asignación de trabajo.

Siglas del sistema PESIMEX Ciclo 1
Responsable JASC Fecha _____

Nombre del trabajador: ANAYURI RAMÍREZ
Siglas asignadas: YURI
Rol(es) asignado(s): PROBADOR

1. Descripción de actividades.

Nombre	Recursos
Ejecutar pruebas de integración	Casos de prueba
Realizar pruebas de aceptación en desarrollo	Casos de prueba
Realizar pruebas de aceptación en sitio	Casos de prueba

2. Fechas de entrega.

Actividad	Fecha de entrega	Fecha de entrega	Fecha de entrega
Resultados de casos de prueba	05/07/2002	05/07/2002	05/07/2002

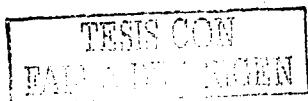
3. Recursos Asignados.

Nombre	Ubicación	Observaciones
Equipo con Windows XP	Segundo piso	Pruebas

Firma del
Trabajador

Firma del
Coordinador

Firma del
Administrador del
proyecto



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



Formato de cierre de ciclo.

Siglas del sistema PESIMEX Ciclo 1
 Responsable JASC Fecha _____

1. Nivel de integración alcanzado(# nivel): 2

2. Resultado de actividades.

Nivel	Actividad	Fecha	Estado
1	ENTRAR	15/07/2002	TERMINADO
2	SALIR	15/07/2002	TERMINADO
3	CAPTURA	15/07/2002	TERMINADO

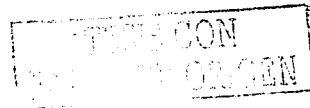
3. Artefactos Obtenidos.

1	MODELO DE ANALISIS, MODELO DE DISEÑO, MODELO DE IMPLEMENTACIÓN, SISTEMA
2	MODELO DE ANALISIS, MODELO DE DISEÑO, MODELO DE IMPLEMENTACIÓN, SISTEMA
3	MODELO DE ANALISIS, MODELO DE DISEÑO, MODELO DE IMPLEMENTACIÓN, SISTEMA

4. Actividades para siguiente Ciclo:

**Firma del
 Administrador del
 proyecto**

176



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



Informe de Ciclo.

Siglas del sistema PESIMEX Ciclo 1
Responsable JASC Fecha _____

1. Total de actividades terminadas al 100% 47 de 47
2. Total de artefactos terminados al 100% 66 de 66

3. Problemáticas.

Actividad/artefacto	Problemática	Solución

4. Lista de Entregables.

Nombre	Formato
SISTEMA	INSTALABLES
MANUAL DE USUARIO	IMPRESO
MANUAL DE CONFIGURACIÓN	IMPRESO

5. Observaciones generales: LA ENTREGA SE HACE EN LOS TIEMPOS ESTIMADOS

Firma del
Coordinador

Firma del
Administrador del
proyecto

TESIS CON
FALLA DE ORIGEN

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



Formato de monitoreo.

Siglas del sistema PESIMEX Ciclo 1
 Responsable JASC Fecha 26/06/2002

1. Avance del ciclo (Llenar sólo si se trata de un monitoreo de avance del ciclo).

Actividades a monitorear.

Actividad	Fecha	Completado	Proceso o pendiente
Diseño de Interfaz	28/06/2002	0	En proceso

Especificar cuales son las actividades que no se han ejecutado al 100% y por qué razón.

Actividad	Comentarios
Diseño de Interfaz	No se han diseñado toda la interfaz

Resumen de actividades (Pendientes).

Actividad	Completado	Pendientes	Porcentaje	Fecha
Diseño de Interfaz	0	1	50%	27/06/2002

Total de actividades terminadas (esperadas) para la fecha de monitoreo: 0
Total de actividades terminadas (real): 0

2. Avance de Artefactos (Llenar sólo para monitoreo de avance de artefactos).

Liste todos los artefactos a monitorear y marque las opciones.

Artefacto	Completado	Pendientes	Estado
Diseño de interfaz			Iniciado

Se han obtenido al 100% _____ de un total de _____ artefactos.

Especificar cuales son los artefactos que no se han ejecutado al 100% y por qué razón.



Artefacto	Comentarios
Diseño de interfaz	Se tiene diseñado el 50% de interfaz, pues se está en los tiempos estipulados

Resumen de artefactos (Pendientes).

Artefacto	Problemáticas	Estado de Avance	Fecha próxima de terminación
Diseño de interfaz	ninguna	50%	27/06/2002

Total de artefactos terminados (esperados) para la fecha de monitoreo: 0

Total de artefactos terminados (real): 0

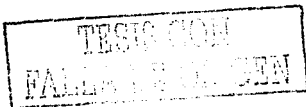
3. Resultado.

Observaciones:

El monitoreo se hizo antes de la fecha de entrega y se encontró un avance aceptable, pues es probable alcanzar los tiempos establecidos.

**Firma del
Supervisor del
proyecto**

**Firma del
Administrador del
proyecto**



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



Carta de liberación.

Ciudad Universitaria a 15 julio de 2002

**Por medio de la presente se hace la entrega del sistema: PESIMEX -MÓDULO
ADMNISTRACIÓN BIBLIOGRÁFICA el cual esta conformado por:**

Nombre	Formato
SISTEMA	INSTALABLES
MANUAL USUARIO	IMPRESO
MANUAL DE CONFIGURACIÓN	IMPRESO

**Que se atendió con la solicitud 1 entregada al Departamento de Informática
en la fecha 12/06/2002 por Seminario de Independencia**

**Firma del
Solicitante**

**Firma del
Coordinador**

**Firma del
Administrador del
proyecto**

TESIS CON
FALLA DE ORIGEN

180

**Ejemplo del llenado de
artefactos
(Administración de la
Configuración y Control de
Cambios)**

TEMA CON
FALLA EN EL PROCESO

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



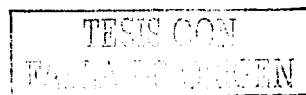
Formato de Informe de configuración.

Siglas del sistema PESIMEX Ciclo 1
 Responsable JASC Fecha _____

1. Asignación de contraseñas a los artefactos.

Directorio/raíz	Contraseña (E)	Contraseña (E)
Artefacto	Contraseña (E)	Contraseña (E)
Arquitectura		
Carta de liberación		
Casos de prueba		
Cierre del proyecto		
Clases del análisis		
Clases del diseño		
Componentes		
Configuración de herramientas		
Descripción del patrón de arquitectura		
Descripción de casos de uso		
Diagramas de casos de uso		
Diseño de arquitectura		
Diagrama Entidad Relación		
Entregable		
Estándar de administración de riesgos		
Estándar de Aceptación		
Estándar de administración de cambios		
Estándar de análisis y diseño		
Estándar de codificación		
Estándar de configuración		
Estándar de documentación		
Estándar de evaluación		
Estándar de pruebas		
Estándar de base de datos		
Estándar de requerimientos		
Estándar de herramientas		
Estándar de resolución de problemas		
Formato de asignación de trabajo		
Formato de diseño de base de datos		
Formato de cierre de ciclo		

182



Formato de distribución		
Formato de monitoreo		
Formato de preguntas		
Formato de prioridad de casos de uso		
Formato de requerimientos		
Formato de validación de entregable		
Herramientas		
Informe de ciclo		
Informe de configuración		
Informe de entregable		
Informe de integración		
Interfaz		
Invitación de pruebas formales		
Lista de herramientas del ciclo		
Manual de configuración		
Manual de instalación		
Manual de usuario		
Manual técnico		
Modelo de análisis		
Modelo de casos de uso		
Modelo de diseño		
Modelo de Implementación		
Notificación de configuración		
Paquetes del análisis		
Plan de ciclos		
Plan de Desarrollo de Software		
Plan de integración		
Plan de pruebas		
Realización de casos de uso del diseño		
Realización de casos de uso del análisis		
Solicitud de cambio		
Solicitud de desarrollo de software		
Subsistemas de diseño		
Subsistemas o sistema		
Visión de desarrollo	RT&5%A78	RT&5%A78
Visión Inicial	PST56\$8&	PST56\$8&

**Firma del
Administrador de
configuración**



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



Formato de Monitoreo de configuración .

Siglas del sistema PESIMEX Ciclo 1
 Responsable JASC Fecha _____

1. Actividad: Elaborar modelo de Análisis

2. Verificación de artefactos.

Modelo de Análisis	12	13	NO	SI

Total de cambios encontrados: _____

3. Cambios.

Modelo de Análisis	Faltaba carátula
--------------------	------------------

3.1. Verificar artefactos derivados.

Modelo de diseño	NO
------------------	----

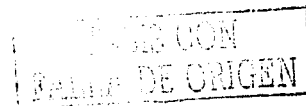
Total de artefactos derivados no actualizados: 1

3.2 Lista de artefactos a cambiar

--

El cambio no influye en el contenido del modelo de análisis por lo que no hay que cambiar el modelo de diseño.

**Firma del
 Administrador de
 configuración**



184

Formato de notificación de configuración .

Siglas del sistema PESIMEX Ciclo 1
 Responsable JASC Fecha _____

No. Cambio en Configuración: _____

1. Configuración Anterior

Directorio raíz	Contraseña(D)	Contraseña(E)
Arquitectura		
Carta de liberación		
Casos de prueba		
Cierre del proyecto		
Clases del análisis		
Clases del diseño		
Componentes		
Configuración de herramientas		
Descripción del patrón de arquitectura		
Descripción de casos de uso		
Diagramas de casos de uso		
Diseño de arquitectura		
Diagrama Entidad Relación		
Entregable		
Estándar de administración de riesgos		
Estándar de Aceptación		
Estándar de administración de cambios		
Estándar de análisis y diseño		
Estándar de codificación		
Estándar de configuración		
Estándar de documentación		
Estándar de evaluación		
Estándar de pruebas		
Estándar de base de datos		
Estándar de requerimientos		
Estándar de herramientas		
Estándar de resolución de problemas		
Formato de asignación de trabajo		



185

Formato de diseño de base de datos		
Formato de cierre de ciclo		
Formato de distribución		
Formato de monitoreo		
Formato de preguntas		
Formato de prioridad de casos de uso		
Formato de requerimientos		
Formato de validación de entregable		
Herramientas		
Informe de ciclo		
Informe de configuración		
Informe de entregable		
Informe de integración		
Interfaz		
Invitación de pruebas formales		
Lista de herramientas del ciclo		
Manual de configuración		
Manual de instalación		
Manual de usuario		
Manual técnico		
Modelo de análisis		
Modelo de casos de uso		
Modelo de diseño		
Modelo de Implementación		
Notificación de configuración		
Paquetes del análisis		
Plan de ciclos		
Plan de Desarrollo de Software		
Plan de integración		
Plan de pruebas		
Realización de casos de uso del diseño		
Realización de casos de uso del análisis		
Solicitud de cambio		
Solicitud de desarrollo de software		
Subsistemas de diseño		
Subsistemas o sistema		
Visión de desarrollo	RT&5%A78	RT&5%A78
Visión Inicial	PST56\$8&	PST56\$8&

TESIS CON
FALLA DE ORIGEN

186

2. Configuración Actual

Artículo	Contribución (L)	Contribución (E)
Arquitectura		
Carta de liberación		
Casos de prueba		
Cierre del proyecto		
Clases del análisis		
Clases del diseño		
Componentes		
Configuración de herramientas		
Descripción del patrón de arquitectura		
Descripción de casos de uso		
Diagramas de casos de uso		
Diseño de arquitectura		
Diagrama Entidad Relación		
Entregable		
Estándar de administración de riesgos		
Estándar de Aceptación		
Estándar de administración de cambios		
Estándar de análisis y diseño		
Estándar de codificación		
Estándar de configuración		
Estándar de documentación		
Estándar de evaluación		
Estándar de pruebas		
Estándar de base de datos		
Estándar de requerimientos		
Estándar de herramientas		
Estándar de resolución de problemas		
Formato de asignación de trabajo		
Formato de diseño de base de datos		
Formato de cierre de ciclo		
Formato de distribución		
Formato de monitoreo		
Formato de preguntas		
Formato de prioridad de casos de uso		
Formato de requerimientos		
Formato de validación de entregable		
Herramientas		
Informe de ciclo		

1187

Informe de configuración		
Informe de entregable		
Informe de integración		
Interfaz		
Invitación de pruebas formales		
Lista de herramientas del ciclo		
Manual de configuración		
Manual de instalación		
Manual de usuario		
Manual técnico		
Modelo de análisis	Y67UN&9#	Y67UN&9#
Modelo de casos de uso		
Modelo de diseño		
Modelo de Implementación		
Notificación de configuración		
Paquetes del análisis		
Plan de ciclos		
Plan de Desarrollo de Software		
Plan de integración		
Plan de pruebas		
Realización de casos de uso del diseño		
Realización de casos de uso del análisis		
Solicitud de cambio		
Solicitud de desarrollo de software		
Subsistemas de diseño		
Subsistemas o sistema		
Visión de desarrollo	RT&5%A78	RT&5%A78
Visión Inicial	PST56\$8&	PST56\$8&

3. Total de cambios: 1

**Firma del
Administrador de
configuración**

TESIS CON
FALLA DE ORIGEN

1188

Instituto de Investigaciones Bibliográficas

Departamento de Informática
Sección de desarrollo de Software



Solicitud de cambio .

Siglas del sistema PESIMEX Ciclo 1
Responsable JASC Fecha _____

No. Cambio: 1

Solicitante: USUARIO RESPONSABLE (PAULO CESAR)
Descripción del cambio: Cambiar color de pantalla de entrada de rojo a azul marino

Solo llenar por personal de informática

Módulos impactados: ENTRAR

Tiempo estimado del cambio: 1 DIA

Resolución tomada: SE REALIZARA EL CAMBIO

Firma del
Solicitante

Firma del
Coordinador

Firma del
Administrador del
proyecto

TEXT CON
FALLA DE ORIGEN

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



Formato de notificación de cambio .

Siglas del sistema PESIMEX Ciclo 1
 Responsable JASC Fecha _____

No. Cambio: _____

Solicitante: USUARIO RESPONSABLE (PAULO CESAR)

Descripción del cambio: Cambiar color de pantalla de entrada de rojo a azul marino

Impacto

Actividad	Cambios	Tiempo estimado	Responsable
Diseñar interfaz	1	1 día	jasc

Firma del Solicitante

Firma del Coordinador

Firma del Administrador del proyecto

Nombre y firma de representante de la mesa de cambios

Firma Administrador de cambios

TESIS CON
 FALLA DE ORIGEN

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



Formato de verificación de cambios .

Siglas del sistema PESIMEX Ciclo 1
Responsable JASC Fecha _____

No. Cambio: 1

Solicitante: USUARIO RESPONSABLE (PAULO CESAR)
Descripción del cambio: Cambiar color de pantalla de entrada de rojo a azul marino

Verificación.

Actividad	Plazo	Estado	Comentarios
Diseño de interfaz	1 día	SI	

Observaciones generales: _____

**Firma del
Administrador de
configuración**

TESIS CONT
FALLA DE ORIGEN

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



Formato de Informe de solicitud de cambio para el usuario.

Siglas del sistema PESIMEX Ciclo 1
Responsable JASC Fecha _____

Solicitante: Dr. Tarcisio

Por medio del presente, el Departamento de Informática le informa de la resolución tomada sobre la petición de cambio hecha el 5 de Julio del 2002 recibida con número de cambio 3.

Módulos
impactados: Captura

Tiempo estimado del cambio: 1 mes

Resolución tomada: Se decidió tomar su cambio como un nuevo requerimiento que será tratado en el siguiente ciclo

Sin más por el momento y en espera de cualquier duda o aclaración.

**Representante de
mesa de cambios**

**Firma del
Coordinador**

**Firma del
Administrador del
proyecto**

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN