

Universidad Nacional Autónoma de México

FACULTAD DE INGENIERIA.

PROCEDIMIENTO PARA EL ANÁLISIS DE SISTEMAS INFORMÁTICOS BASADO EN LA METODOLOGÍA RUP

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A N:

CABALLERO LÓPEZ ROBERTO
RESÉNDIZ GOCHE LIDIA
ROJAS SANTOYO ARTURO
ZARDANETA VÁZQUEZ NADIA MARIANA

DIRECTOR DE TESIS: ING. GLORIA MATA HERNÁNDEZ



MÉXICO D.F. 2005

ÍNDICE

PRÓLOGO

CAPÍTULO 1

Definición de sistema informático
Ciclo de desarrollo de un sistema informático
Análisis
Diseño del sistema
Desarrollo del software
Pruebas del sistema
Capacitación e implantación
Mantenimiento

CAPÍTULO 2

Definición de análisis
Importancia de la etapa de análisis en un sistema informático
Elementos que contribuyen a un buen análisis
Estrategias para el levantamiento de requerimientos
Diagramas para el modelado del análisis
La toma de decisiones en la etapa de análisis

CAPÍTULO 3

Programación Extrema (XP)
Desarrollo Enfocado a Características (FDD)
Proceso Racional Unificado (RUP)
Marco de Soluciones Microsoft
Modelo de equipos de iguales
Modelo o proceso de administración de riesgos
Modelo de proceso
Modelo de aplicaciones
Comparativo de metodologías

CAPÍTULO 4

Prácticas en el desarrollo de software con base en RUP
Desarrollo de software con el método iterativo
Manejo de requerimientos
Uso de arquitecturas basadas en componentes
Modelado de software de forma visual
Verificar continuamente la calidad del software
Control de cambios del software
El Proceso Racional Unificado (RUP)
Elementos básicos que componen a RUP
Desarrollo iterativo
Beneficios de un proceso iterativo

Flujos de trabajo
Dirección del proyecto
Requerimientos
Análisis
Diseño
Implementación
Prueba
Configuración y dirección de cambio
Ambiente
Despliegue
Planes de iteración típicos
Definiendo la visión del proyecto y el caso de uso
Construyendo un prototipo arquitectónico
Implementando el sistema

CAPÍTULO 5

Componentes principales del PAS
Proceso iterativo
Fases y puntos de transición
Flujos de trabajo
Diagramas de modelado
Propiedades del PAS
Descripción del procedimiento PAS
Fase de inicio (1.0.0.0)
Actividades del PAS (Tabla 5.9)
Fase de elaboración (2.0.0.0)
Fase de construcción (3.0.0.0)
Documento integral del análisis
Formatos propuestos

CAPÍTULO 6

Primer caso de estudio
Fase de inicio (1.0.0.0)
Formato integral de análisis
Sistema PAS – PAT versión 1.0
Segundo caso de estudio
Fase de inicio (1.0.0.0)
Formato integral de análisis
Glosario de términos

CONCLUSIONES

ANEXOS

GLOSARIO

BIBLIOGRAFÍA

PRÓLOGO

La experiencia profesional adquirida a través de diversos proyectos de desarrollo de sistemas ha permitido identificar un punto en común en aquellos que no logran ser exitosos: un análisis poco profundo de los requerimientos, lo cual provoca que surjan graves errores al momento de la liberación que pudieron ser evitados si se hubiera llevado a cabo un análisis exhaustivo para llegar a la solución óptima sin ocasionar pérdidas de tiempo, dinero y en ocasiones hasta el prestigio de una empresa.

El análisis de sistemas en cualquier disciplina, y de manera particular para los sistemas informáticos, es vital, ya que permite concebir en su totalidad las necesidades que queremos abordar. Asimismo permite acotar claramente los requerimientos del sistema a desarrollar de manera que entre mayor sea la cobertura que se alcance en esta etapa, mejores serán los resultados alcanzados. Es la plataforma sobre la que se debe basar el desarrollo, sin embargo, actualmente no se le da la importancia debida, derivando en problemas que provocan el no obtener el producto esperado como resultado de nuestro esfuerzo.

El análisis es la base del éxito de un sistema, razón por la cual se debe realizar un análisis de calidad, para ello es necesario contar con elementos y herramientas que nos permitan preparar adecuadamente las condiciones del proyecto en vistas al desarrollo de un sistema que cubra adecuadamente los requerimientos del usuario y que además de ser productivo, sea confiable y de fácil administración.

Existen diversas metodologías para llevar a cabo el análisis de sistemas como son MSF, XP, FDD, RUP, entre otras, sin embargo cada una de ellas cubre el análisis con un enfoque determinado, sin contar con la flexibilidad necesaria para facilitar la integración global y total del análisis para sistemas con características distintas.

La aplicación de un procedimiento que integre y contemple todos los elementos de abstracción del sistema así como los diagramas para su modelado, necesarios para cubrir exhaustivamente su análisis, es el inicio del éxito de un proyecto.

El análisis minucioso de un sistema proporciona ventajas al equipo de desarrollo, por mencionar algunas diremos:

- Identificar los requerimientos reales del usuario
- Prever los recursos mínimos necesarios para el desarrollo, implementación y pruebas del sistema
- Prever en tiempo y forma los posibles errores en el sistema y la manera de eliminarlos o minimizarlos
- Definir el alcance del proyecto
- Analizar un proyecto de acuerdo con las restricciones y especificaciones del usuario
- Conocer todos los inconvenientes que pudieran presentarse
- Asegurar que el usuario utilice el sistema desarrollado
- Reducir riesgos; es decir, prever problemas que pueden poner en riesgo el proyecto
- Desarrollar un buen diseño o plan del proceso
- Obtener la visión correcta del sistema

El objetivo de este proyecto de tesis es proponer un procedimiento base que permita efectuar el análisis a detalle de un proceso y modelarlo adecuadamente para obtener un sistema informático que cumpla completamente con los requerimientos del usuario, que cuente con la flexibilidad necesaria para ser

empleado en proyectos de desarrollo de sistemas que presenten características distintas y que sea susceptible de ser enriquecido con experiencias personales.

Por otro lado, el presente trabajo de tesis comprende seis capítulos:

El capítulo 1 titulado “Conceptos Generales” presenta un panorama general de los conceptos básicos que están involucrados en este trabajo de tesis, así como el típico ciclo en cascada de desarrollo de sistemas informáticos.

El capítulo 2 titulado “Análisis de Sistemas Informáticos” muestra más a fondo el proceso de análisis de sistemas, enfocándonos en su importancia como factor clave en el éxito de un proyecto de desarrollo y las prácticas más utilizadas para llevarlo a cabo.

El capítulo 3 titulado “Metodologías Diversas” presenta un panorama general de cuatro metodologías de desarrollo de sistemas informáticos, así como un análisis comparativo entre ellas con el objetivo de encontrar características útiles para el procedimiento propuesto.

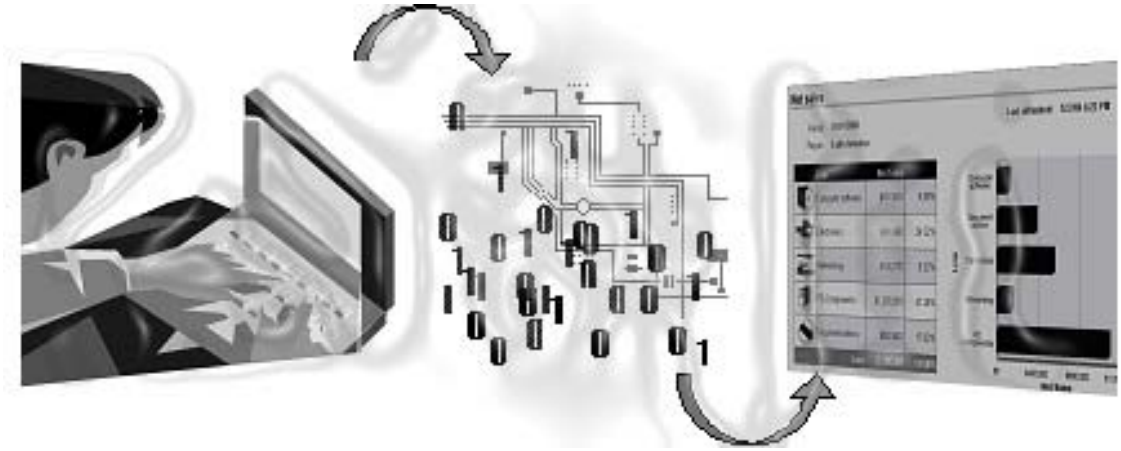
El capítulo 4 titulado “Metodología RUP”, describe a fondo las características, actividades y prácticas de la metodología RUP, la cual fue elegida como base para el procedimiento propuesto.

El capítulo 5 titulado “Procedimiento para el Análisis de Sistemas Informáticos PAS” describe a detalle las características, elementos y actividades que componen el procedimiento propuesto para el análisis de sistemas informáticos (PAS), objeto de este trabajo de tesis.

El capítulo 6 titulado “Casos de Estudio” presenta dos escenarios de aplicación del procedimiento propuesto, cada uno con características distintas, con el objeto de mostrar el amplio espectro de aplicación del procedimiento propuesto y el camino a seguir por un analista que adopte el mismo como herramienta de análisis de sistemas informáticos.

CAPÍTULO 1

Conceptos Generales



En la actualidad existe una constante necesidad de nuevos desarrollos de sistemas informáticos, y debido a diversos factores un alto porcentaje de ellos no son terminados, o su desempeño una vez implementados no es el deseado por el usuario de dicho sistema. Entre dichos factores el lugar principal lo ocupa la carencia de calidad en el análisis del sistema o su total omisión.

En preparación para exponer el “Procedimiento para el análisis de sistemas informáticos basado en la metodología RUP” que busca proporcionar una herramienta que asista en la solución del problema antes descrito, es importante abordar en este primer capítulo los conceptos generales que se utilizarán más adelante.

Se explican conceptos generales, integrando definiciones de diversas fuentes para obtener una definición de sistema en general y de sistema informático que se apliquen de forma más adecuada a este trabajo de tesis.

Asimismo, se describe el ciclo en cascada de desarrollo de sistemas informáticos, mencionando cada una de las etapas principales: investigación preliminar, determinación de los requerimientos del sistema, prototipo del sistema, ajustes y aceptación del prototipo, diseño del sistema, desarrollo del software, pruebas de los sistemas, capacitación e implantación y mantenimiento.

Al final del capítulo se comprenderán los conceptos de sistema, sistema informático y se conocerán las diversas etapas que componen el ciclo de desarrollo de sistemas informáticos en cascada.

1.1 Definición de sistema informático

La naturaleza intelectual del hombre le exige adquirir conocimientos mediante un proceso sucesivo de especialización y ampliación de la información que tiene sobre un tema determinado.

El estudio de cualquier área de conocimiento requiere entender y aprender, como un principio firme, todos los conceptos básicos involucrados, los cuales serán los cimientos para adquirir información más específica.

Es por eso que para exponer lo que es un sistema informático, como primer paso debemos comprender los términos "sistema" e "informática".

Algunos autores definen el término "Sistema" de acuerdo con diversos enfoques, veamos algunos ejemplos:

- El diccionario de la Real Academia de la Lengua Española indica que sistema es:

"Conjunto de reglas o principios sobre una materia racionalmente enlazados entre sí".

- Benjamín S. Blanchard[1] habla de un "fin u objetivo" dentro de la definición de sistema:

"Un sistema es una combinación de medios (como personas, materiales, equipos, software, instalaciones, datos...), integrados de forma tal que puedan desarrollar una determinada función en respuesta a una necesidad concreta".

- Javier Aracil [2], nos dice:

"De modo coloquial hablamos de un sistema, como de un modo o manera de hacer algo; así, decimos que tenemos un sistema para resolver un problema o para alcanzar un objetivo (. . .) Más formalmente hablamos de un sistema como de un objeto dotado de alguna complejidad, formado por partes coordinadas, de modo que el conjunto posea una cierta unidad, que es precisamente el sistema. Así, hablamos del sistema planetario, formado por los planetas unidos mediante las fuerzas gravitatorias; de un sistema económico, formado por agentes económicos, relacionados entre sí por el intercambio de bienes y servicios; de un sistema ecológico, formado por distintas poblaciones…".

Los conceptos simples son los más difíciles de explicar de forma clara, de ahí que no exista una definición universal de sistema. Lo más adecuado, en todo caso, es adoptar la definición que más se acerque a lo que, en cada contexto, se necesita comprender como sistema.

Con base en las definiciones anteriores se dirá que un sistema es, en términos generales:

"Un conjunto de elementos interrelacionados entre sí, los cuales tienen un objetivo en común y generan respuestas determinadas a entradas específicas".

La figura 1.1 muestra un sistema como una caja negra, conformada por un conjunto de elementos relacionados entre sí, los cuales generan una respuesta a ciertas entradas específicas.

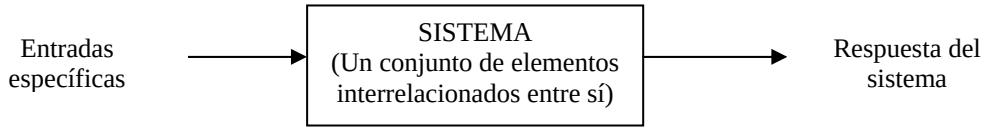


Figura 1.1 Representación general de un sistema, en el cual podemos ver que al recibir ciertas entradas (datos o estímulos de diversos tipos), éste genera una respuesta acorde con la entrada

[1] Director del Programa de Ingeniería de Sistemas del Instituto Politécnico de Virginia y Universidad Estatal
[2] Catedrático de Ingeniería de Sistemas y Automática, en la Escuela Superior de Ingenieros de la Universidad de Sevilla, España

El término “Informática” proviene de la fusión de las palabras INFORmación y autoMÁTICA, de tal manera que informática podría ser considerada como una disciplina que estudia el tratamiento automático de la información.

- La Real Academia de la Lengua Española da la siguiente definición:

“Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de computadoras electrónicas”.

Por otra parte, al hablar de un sistema informático se hará referencia a:

“La conjunción de elementos que comprenden hardware y software integrados ordenadamente, de tal forma que alcancen un objetivo determinado con el fin de apoyar las actividades del usuario”.

Los elementos que componen el ciclo completo de interacción con un sistema informático son de naturaleza diversa y normalmente incluyen:

- *Los datos o información fuente* que son introducidos en el sistema; son todas las entradas que necesita el sistema para generar como resultado la información que se desea.

- *El equipo computacional*, es decir, el hardware necesario para que el sistema de información pueda operar, lo constituyen las computadoras y el equipo periférico que puede conectarse a ellas.

- *Los programas* que son ejecutados y los cuales generan diferentes tipos de resultados, son parte del software del sistema de información que hará que los datos de entrada introducidos sean procesados correctamente y generen los resultados que se esperan.

- *El recurso humano* que interactúa con el sistema de información, el cual está formado por las personas que utilizan el sistema, alimentándolo con datos o utilizando los resultados que genere.

La figura 1.2 muestra los pasos dentro del ciclo de interacción con un sistema informático. Debe notarse que en este tipo particular de sistema, los elementos que contiene la caja negra son los programas (software) y el equipo de cómputo (hardware)

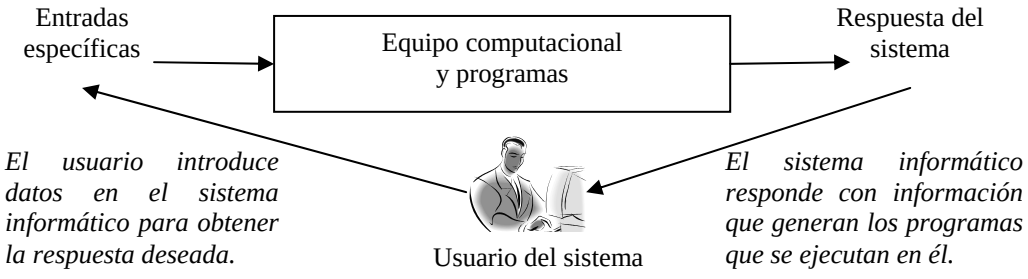


Figura 1.2 Ciclo de interacción con un sistema informático: el usuario es el elemento donde inicia y termina este ciclo continuo que comienza con la introducción de ciertos datos y termina con la obtención de otros distintos, generados y transformados por medio de los programas y el equipo computacional.

1.2 Ciclo de desarrollo de un sistema informático

Un sistema informático siempre cubre una necesidad, es creado con el fin de ejecutar una tarea determinada que realiza a través de programas y aplicaciones de software.

Cabe mencionar que el software se desarrolla, no se fabrica en el sentido clásico de la palabra que se aplica a productos materiales. Ambas actividades, el desarrollo y la fabricación, se dirigen a la obtención de un "producto" aunque los métodos y los resultados son diferentes.

Los costos del software se encuentran comprendidos en la ingeniería que se invierte en su desarrollo, esto implica que los proyectos de creación de sistemas informáticos no se pueden gestionar como si lo fueran de fabricación. A mediados de la década de 1980, se introdujo el concepto "fabrica de software", que recomienda el uso de herramientas para el desarrollo automático del software.

La formalización del proceso de desarrollo ha sido definida con un marco de referencia denominado ciclo de desarrollo del software o ciclo de vida del desarrollo, el cual se puede describir como "el periodo de tiempo que comienza con la decisión de desarrollar un sistema y que finaliza cuando se ha liberado".

Este ciclo incluye seis fases: análisis, diseño, desarrollo, pruebas, capacitación e implementación y por último mantenimiento, como se muestra en la figura 1.3.

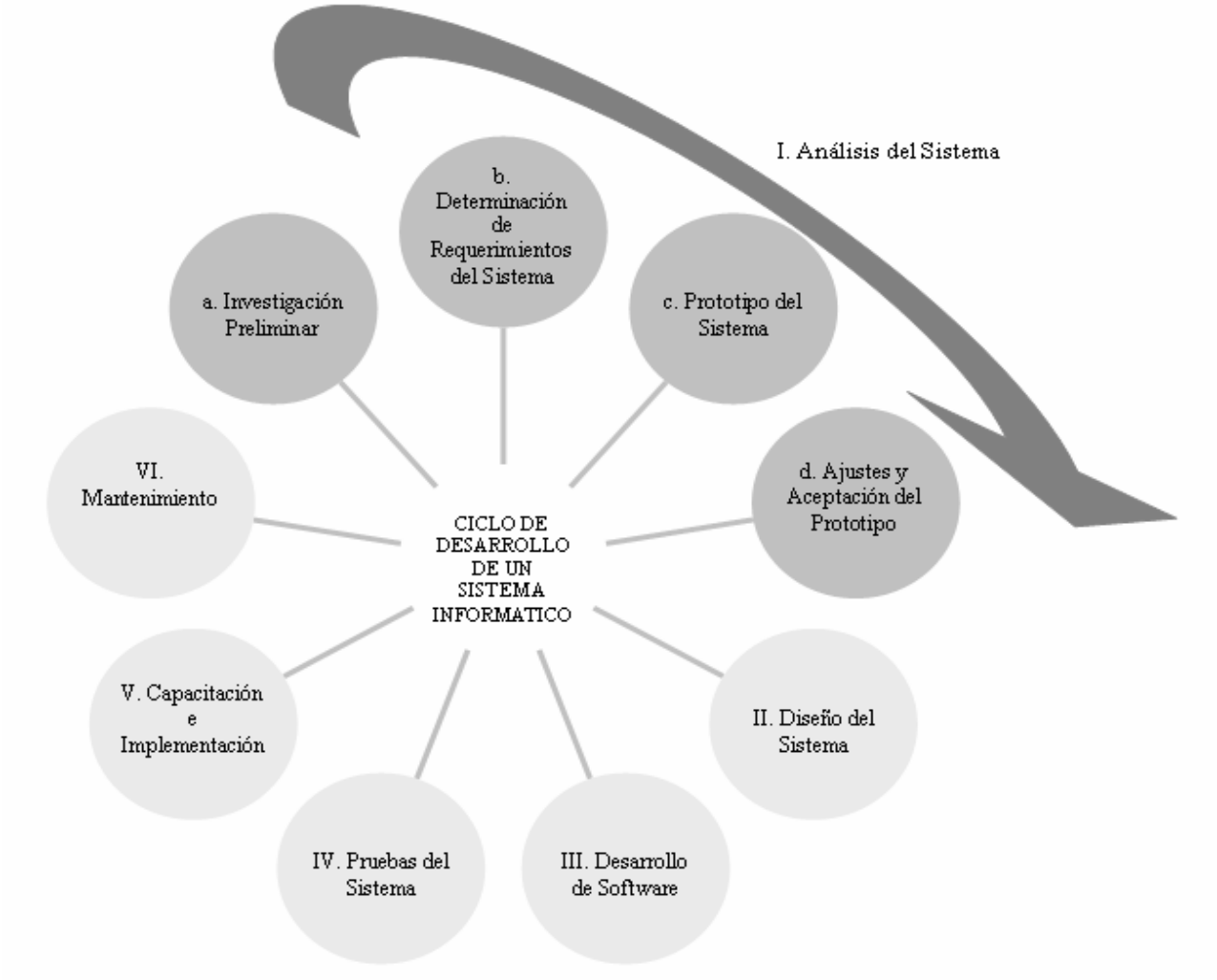


Figura 1.3 Las seis fases del ciclo de desarrollo de un sistema informático

A continuación se describirá el ciclo de desarrollo de un sistema informático

1.2.1 Análisis

Durante el análisis se realizan las siguientes actividades:

- Investigación Preliminar
- Determinación de los Requerimientos del Sistema
- Prototipo del Sistema
- Ajustes y Aceptación del Prototipo

Investigación preliminar

Se inicia siempre con la solicitud de una persona (administrador, empleado o especialista de sistemas) para que sea creado un sistema informático que realice cierta actividad o tarea. Las actividades que se realizan en esta etapa son:

- Aclaración de lo que el usuario solicita.- Antes de comenzar a realizar investigaciones propias del sistema a desarrollar, es necesario que quede completamente claro qué es lo que el sistema realizará o la necesidad que va a cubrir de forma muy general.

- Estudio de factibilidad.- Habiendo resumido y entendido la información proporcionada por el personal de la organización, se realiza un estudio de factibilidad del proyecto. Esta etapa permite saber si es posible realizar el sistema requerido, de acuerdo con la situación técnica, económica, operacional y de alcance del mismo, es decir, conocer si se cuenta con el personal y las herramientas técnicas para desarrollarlo, la capacidad económica para solventar los gastos que han de generarse obteniendo un resultado positivo en la relación costo-beneficio, y si el sistema finalmente, al ser entregado a los usuarios, éstos lo utilizarán y explotarán adecuadamente.

- Aprobación de la solicitud.- Cuando se han cubierto los puntos anteriores, se realiza una toma de decisiones en el ámbito administrativo, sobre la prioridad del desarrollo del sistema y sobre la factibilidad de su realización, para finalmente aprobarlo o rechazarlo. Todo el proceso de análisis se debe documentar adecuadamente para futuras referencias.

La figura 1.4 muestra las actividades propias de la investigación preliminar de forma esquemática.

Determinación de los requerimientos del sistema

Para conocer los procesos que ha de realizar el sistema, es necesario que el analista los estudie en conjunción con los futuros usuarios. En esta fase el analista debe comprender qué información necesitan los usuarios para realizar su trabajo. El analista debe saber los detalles de las funciones actuales del proceso:

¿Qué personas están involucradas?

¿Cuál es la actividad del negocio?

¿Dónde se lleva a cabo el trabajo?

¿En qué momento se realiza cada actividad?

¿Cómo se desarrolla ésta?

Conforme se reúnen los detalles, los analistas estudian los datos con la finalidad de identificar las características que debe tener el nuevo sistema, incluyendo la información que debe producir junto con sus características operacionales tales como controles de procesamiento, tiempos de respuesta y métodos de entrada y salida.

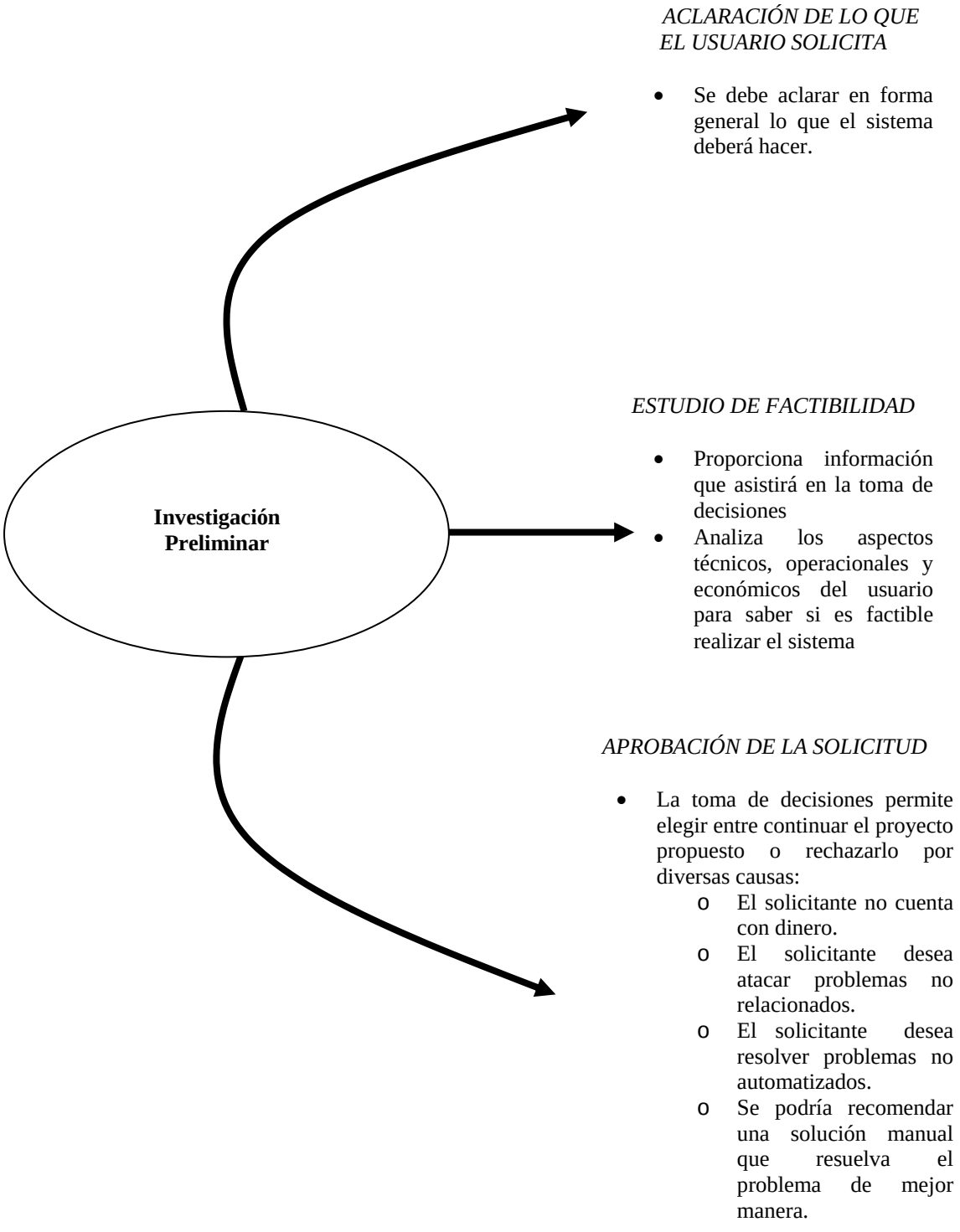


Figura 1.4 Actividades necesarias para realizar la investigación preliminar, primer paso en el desarrollo de sistemas informáticos

Prototipo del sistema

En esta etapa se construyen los formatos de pantalla que se han de presentar al usuario.

El prototipo del sistema son pantallas “tontas”; es decir, sin funcionalidad alguna o interacción con datos, lo único que hacen es anticipar la presentación preliminar del sistema para realizar una revisión con el usuario final, con lo cual podemos obtener observaciones al respecto antes de progresar más en el proceso de desarrollo.

Ajustes y aceptación del prototipo

Los ajustes y aceptación del prototipo consisten en revisar con el usuario las pantallas finales para que tenga un conocimiento más puntual del producto terminado.

Este paso nos permite realizar cambios en el funcionamiento de las pantallas sin afectar la programación, lo cual termina cuando el usuario acepta el prototipo que se le presenta.

1.2.2 Diseño del sistema

En esta fase se alcanza una solución óptima, detallada y con la mayor precisión posible para el desarrollo de la aplicación, teniendo en cuenta los recursos físicos del sistema (red, sistemas operativos de los equipos, arquitectura de las computadoras personales, periféricos, protocolos de comunicación, etc.) y los recursos lógicos (sistema operativo, programas de utilidad, compiladores, bases de datos, etc.) como se muestra en la figura 1.5.

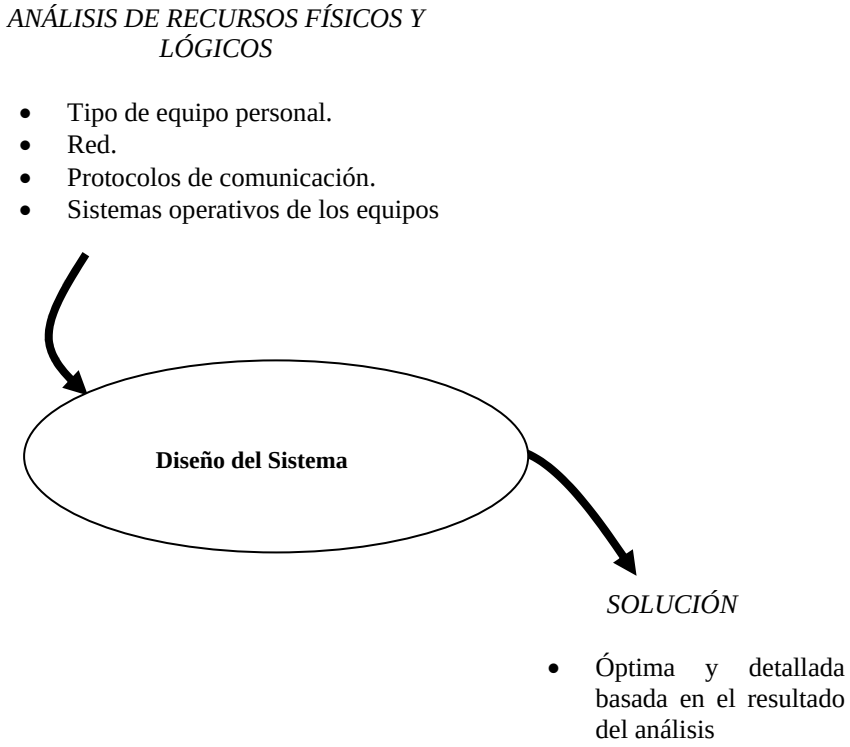


Figura 1.5 Elementos involucrados en el diseño de un Sistema Informático

1.2.3 Desarrollo del software

Consiste en la traducción de la solución obtenida en el diseño a un determinado lenguaje de programación, dando como resultado instrucciones que serán ejecutadas por la computadora.

En esta etapa del sistema se instalan y configuran las herramientas acordadas para el desarrollo del sistema informático.

El desarrollo del software, al igual que todas las etapas en la creación de un sistema informático, debe realizar su trabajo de forma óptima, pero siempre basándose en los resultados de etapas anteriores.

El desarrollador deberá seguir el camino indicado por las etapas de análisis y diseño para crear el código del programa y de ese modo ser consistente; de otra manera el código obtenido no cubrirá los requerimientos para los cuales fue creado.

1.2.4 Pruebas del sistema

En esta etapa se realizan pruebas parciales y totales del sistema informático, a fin de encontrar diversos huecos en la programación y para comprobar que el sistema cuenta con las características de funcionamiento y operación que el usuario requiere.

El cubrir huecos de programación evita que el sistema se emplee de formas no previstas, ya sean involuntarias o malintencionadas. Se realizan pruebas de estrés, de concurrencia, de históricos reales y de seguridad.

Se analizan los tiempos de respuesta del sistema, facilidad de uso, adecuación de los formatos de la información, confiabilidad general y el nivel de utilización.

1.2.5 Capacitación e implantación

Este paso requiere que el sistema informático se instale en el entorno operativo o sistema físico donde va a funcionar habitualmente, capacitar a los usuarios en el uso del sistema y realizar todo lo necesario para que el sistema pueda ser utilizado cotidianamente.

Dependiendo del tamaño de la organización que empleará la aplicación y el riesgo asociado con su uso, puede elegirse empezar la operación del sistema solo en unas cuantas áreas de la empresa o en todas. Algunas veces se deja que los dos sistemas, el viejo y el nuevo, trabajen en forma paralela con la finalidad de comparar resultados. Finalmente se generan los manuales técnico y de usuario.

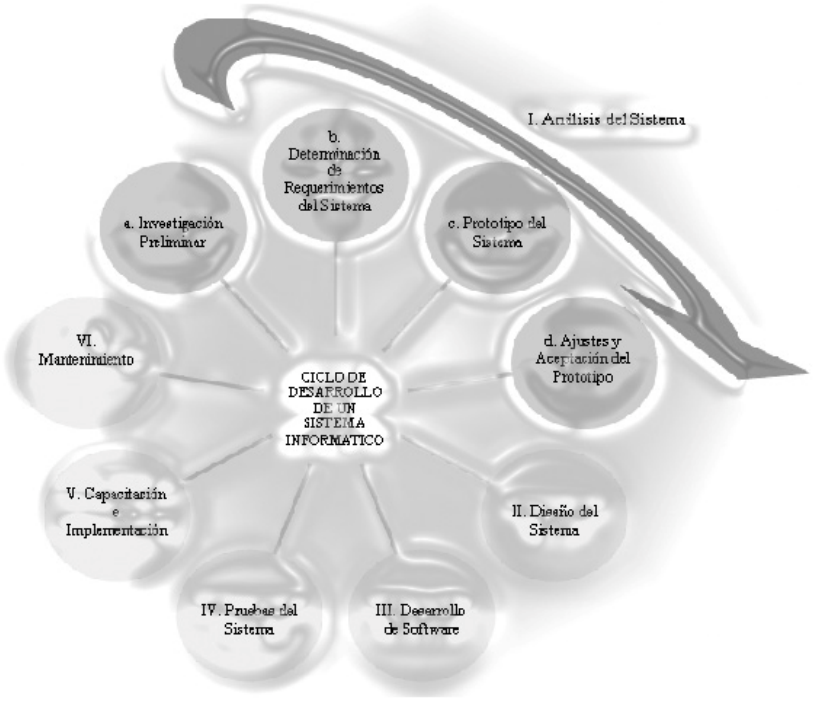
1.2.6 Mantenimiento

Esta fase completa el ciclo de desarrollo y en ella se realizan las correcciones necesarias para resolver errores y deficiencias mínimas del producto desarrollado, existiendo la posibilidad de que ciertas correcciones en esta fase por su tamaño o complejidad requieran reiniciar el ciclo.

El mantenimiento se puede ver como una garantía en la cual el desarrollo queda abierto, durante un periodo de tiempo acordado para corregir cualquier falla en el sistema que ya se ha liberado y está en operación.

CAPÍTULO 2

Análisis de Sistemas Informáticos



Como una parte del ciclo de desarrollo de sistemas, el análisis de sistemas informáticos emplea diversas herramientas para conocer los requerimientos del usuario, como cuestionarios abiertos o cerrados, entrevistas estructuradas y no estructuradas, revisión de la documentación con la que cuenta el usuario y creación de diagramas para el modelado como son diagramas de flujo y diagramas entidad-relación que asistan al analista para realizar un buen análisis del sistema.

La importancia de la etapa de análisis en el desarrollo de sistemas informáticos es determinante para el resultado de un proyecto, la calidad del sistema obtenido depende en gran parte del esfuerzo puesto en su análisis.

La toma de decisiones en la etapa de análisis es de gran importancia porque permite estudiar las alternativas que pueden llevar al éxito o fracaso un proyecto. Entre los pasos que se siguen en la toma de decisiones están el definir las metas, identificar y evaluar cada opción y sus consecuencias, considerando diversas situaciones como son la condición de certidumbre, condición de incertidumbre y las condiciones de riesgo para tomar finalmente una decisión sobre la realización del proyecto.

2.1 Definición de análisis

En términos generales el concepto de análisis se refiere a la:

"Distinción y separación de las partes de un todo hasta llegar a conocer sus principios o elementos"

Así mismo, el análisis de un sistema informático se define como *"El estudio mediante diversas técnicas de los límites, características y posibles soluciones a un problema al que se aplica un tratamiento por métodos automatizados"*, de manera más simple, "es el proceso de determinar qué se necesita hacer y cuál de las posibles soluciones es la mejor para generar un sistema informático, respetando restricciones tecnológicas y presupuestales".

Una cantidad importante de proyectos de desarrollo de sistemas informáticos fracasan debido a una inadecuada administración, ocasionada por fallas en el análisis de las necesidades del negocio y en el diseño de una solución, lo cual debe realizarse antes de comenzar su codificación.

Con "negocio" se hace referencia a tareas, en diversos niveles de detalle, a las que se dedica o que desempeña una institución, empresa, departamento, área o persona. Por ejemplo, el negocio de una mueblería son las ventas, por otra parte, el negocio del área de recursos humanos es llevar el control de nóminas y personal.

El propósito del análisis es articular completamente los requerimientos del negocio y usuario con base en la comprensión de las necesidades, encontrando de esa forma la solución que se ajuste a las restricciones de recursos y tiempo impuestas por el propio proyecto.

El analista de sistemas define la naturaleza del problema para que el diseñador de software explore las diversas soluciones y las represente a través de modelos, tomando decisiones bien informadas para llegar a un producto que dejará a los usuarios satisfechos.[1]

Lo anterior se puede reducir a un concepto simple "Encontrar lo que se requiere hacer antes de pensar como hacerlo".

Para este trabajo es posible proponer la siguiente definición de análisis de sistema informático:

"La fase del ciclo de desarrollo que consiste en recabar toda la información necesaria sobre el proceso a automatizar, por medio de la clasificación e interpretación de hechos, logrando el diagnóstico de necesidades a través del empleo de la información obtenida para desarrollar o mejorar un sistema informático".

[1] Analista y diseñador son papeles que puede desempeñar la misma persona, personas diferentes o equipos completos de personas, dependiendo del tamaño del proyecto.

2.2 Importancia de la etapa de análisis en un sistema informático

En los primeros años de la era de la computación, al final de la década de los 50's, el análisis de sistemas era muy limitado y se reducía a la descripción verbal de los requerimientos de los usuarios, los cuales eran interpretados por los desarrolladores y programados a su criterio.

En la actualidad, la sociedad y la economía exigen la generación de productos de calidad, funcionales y eficientes, y los sistemas informáticos no sólo deben cumplir con las características antes mencionadas, sino que deben ser creados en tiempos y costos cada vez más reducidos.

En numerosos casos, el tiempo y los recursos económicos destinados para el desarrollo de un sistema informático son dictados por ejecutivos de alto nivel que desconocen la importancia de tomar el tiempo necesario para realizar un adecuado análisis de sistemas.

Debido a la urgencia de entregar resultados en los tiempos establecidos muchas veces se pasa directamente a la etapa de desarrollo sin haber realizado un análisis exhaustivo del sistema a desarrollar.

Lo anterior lleva a que se presenten diversos problemas comunes:

- El equipo de trabajo labora durante todo el proyecto bajo presión por los reducidos tiempos de entrega que no se establecieron con base en un procedimiento de análisis de sistemas.

- Al haber omitido la etapa de análisis, los procesos de gran trascendencia no se conocen con la profundidad necesaria y problemas que se hubieran detectado a tiempo surgen de forma imprevista, provocando atrasos y por lo tanto ritmos de trabajo más exigentes.

- Los desarrolladores deben programar una y otra vez módulos liberados porque no existe un control de cambios o porque no tienen la funcionalidad que se requiere.

- En las últimas etapas del desarrollo aumenta la incidencia de errores, a causa del cansancio acumulado y principalmente por no haber previsto características del sistema que se hubieran tenido en cuenta gracias a un análisis exhaustivo del mismo.

- Los líderes de proyecto tratan de reducir los tiempos establecidos de entrega para tener tiempo suficiente para corregir errores que nunca se hubieran presentado si se hubiera pasado correctamente por la etapa de análisis.

- Generalmente el desarrollador entrega un producto incompleto, con diversas y a veces graves fallas que se reparan en la etapa de mantenimiento y en la cual sólo se deberían atender detalles de menor importancia y de fácil solución.

- Debido a que toda la atención se pone en dar la funcionalidad al sistema, se descuida el aspecto de la seguridad, el cual se deja en segundo plano o como un proyecto posterior o complementario.

- En algunos casos, a fin de cumplir con el contrato de desarrollo y evitar problemas legales, el líder de proyecto debe entregar el sistema en un tiempo poco realista y por ello termina invirtiendo muchas más horas hombre de las pactadas.

- En otros casos el usuario es quien finalmente absorbe los gastos de no haber terminado el proyecto en el tiempo previsto.

La siguiente estadística nos presenta la situación actual en el desarrollo de software.

La empresa estadounidense "Standish Group"[2] en su "Chaos Report" de 2004 reportó que en Estados Unidos se gastaron $255 mil millones de dólares al año en Tecnología de Información, de los cuales 15% de los proyectos se cancelaron antes de su terminación, el 34% de los proyectos costaron más del 43% de lo estimado originalmente, se gastaron $55 mil millones por los proyectos cancelados de software y solo el 51% de los proyectos se terminaron en tiempo y en presupuesto.

La estadística anterior pone en evidencia que numerosos proyectos de desarrollo se realizan sin un adecuado análisis previo, dando lugar a sistemas que no cubren los requerimientos del usuario y a proyectos que se suspenden por requerir más tiempo o recursos de los estimados.

2.3 Elementos que contribuyen a un buen análisis

Existen elementos que asisten al analista durante la etapa de análisis de sistemas informáticos, entre ellos tenemos los siguientes:

- Estrategias para el levantamiento de requerimientos
- Diagramas para el modelado del análisis
- La toma de decisiones en la etapa de análisis

2.3.1 Estrategias para el levantamiento de requerimientos

Los sistemas de información con mayor éxito se originan a partir del trabajo conjunto con el usuario, ya que tienen como objetivo resolver una necesidad de la organización que los usuarios perciben. Las contribuciones de los usuarios son muy importantes y los analistas tienen el papel esencial de estudiarlas: "extraer las mejores ideas de los usuarios para su análisis y discusión".

Un proceso de suma importancia en el análisis de sistemas es conocer los requerimientos del usuario. El usuario solicita un sistema para que le facilite realizar diversas tareas que considera pueden automatizarse. Sin embargo, para saber qué es realmente lo que el usuario necesita, debemos realizar una labor estratégica de levantamiento de requerimientos; es decir, utilizar una variedad de técnicas como entrevistas, cuestionarios, revisión de registros y observación con el fin de obtener un conocimiento profundo de las tareas que el sistema deberá ejecutar.

- Entrevista

El analista debe entender los procesos que realizará el sistema, aplicando preguntas que proporcionarán un antecedente de los datos fundamentales y de la descripción del sistema.

¿Cuál es el propósito de esta actividad?
¿Cuáles son los pasos que se realizan?

¿Dónde se realizan?

¿Quién los ejecuta?

¿Cuánto tiempo emplea?

¿Con qué frecuencia se realizan?

¿Quién utiliza la información resultante?

[2] Standish Group es el líder en tecnología de información en valor y desempeño de proyectos, dedicado a asesorar en aspectos como riesgos, costos, retorno y valor de las inversiones realizadas en tecnología de información.

Las entrevistas se deben utilizar para recabar información en forma verbal, a través de preguntas realizadas a usuarios del sistema existente o a usuarios del sistema propuesto. Pueden ser de forma individual o en forma de grupo y se llevan a cabo en forma de conversación, no de interrogación.

En las investigaciones de sistemas las formas cualitativas y cuantitativas de la información son importantes. La información cualitativa está relacionada con opiniones, políticas y descripciones narrativas de actividades o problemas. La información cuantitativa trata con números, frecuencias o cantidades.
Son valiosas las opiniones, comentarios, ideas o sugerencias con relación a como se podría hacer el trabajo. La entrevista a veces es la mejor forma para conocer la actividad de la empresa.

Nuestra habilidad como entrevistador es vital para el éxito en la búsqueda de hechos por medio de la entrevista. Las buenas entrevistas dependen del conocimiento del analista tanto de la preparación del objetivo de una entrevista específica, como de las preguntas por realizar a una persona determinada.

- Determinación del tipo de entrevista

Entrevistas no estructuradas: si el objetivo de la entrevista radica en adquirir información general, es conveniente elaborar una serie de preguntas sin estructura, con una sesión de preguntas y respuestas libres.

Entrevistas estructuradas: cuando el analista necesita adquirir datos más específicos sobre las operaciones o desean asegurar una alta confiabilidad en las respuestas a las preguntas que han propuesto a su entrevistado, las entrevistas estructuradas son mejores.

- Ventajas y desventajas de las entrevistas estructuradas y no estructuradas

El objetivo de la investigación detallada es descubrir hechos, especificar opiniones y conocer como se manejan las operaciones desempeñadas actualmente. Las entrevistas se aplican a todos los niveles gerenciales y de empleados y depende de quien pueda proporcionar la mayor parte de la información útil para el estudio. La siguiente tabla muestra las ventajas y desventajas de cada tipo de entrevista:

	Entrevista Estructurada	Entrevista no Estructurada
Ventajas	• Asegurar la elaboración uniforme de las preguntas que todos los usuarios van a responder. • Fácil de administrar y evaluar. • Evaluación más objetiva tanto de quienes responden como de las respuestas a las preguntas. • Se necesita un limitado entrenamiento del entrevistador. • Resulta en entrevistas más pequeñas.	• El entrevistador tiene mayor flexibilidad al realizar las preguntas adecuadas a quién responde. • El entrevistador puede explotar áreas que surgen espontáneamente durante la entrevista. • Puede producir información sobre áreas que se minimizaron o en las que no se pensó que fueran importantes.
	Entrevista Estructurada	**Entrevista no Estructurada**
Desventajas	• Alto costo de preparación. • Los que responden pueden no aceptar un alto nivel en la estructura y carácter mecánico de las preguntas. • Un alto nivel de la estructura puede no ser adecuado para todas las situaciones. • El alto nivel en la estructura reduce responder en forma espontánea, así como, la habilidad del entrevistador para continuar con comentarios hacia el entrevistado.	• Puede realizarse negativamente en el tiempo tanto de quien responde como del entrevistador. • Los entrevistados pueden introducir sus segos en las preguntas o al informar de los resultados. • El análisis e interpretación de los resultados pueden ser largos. • Toma tiempo recabar los hechos esenciales.

- Cuestionarios

Son un conjunto de preguntas que el analista crea de manera anticipada, pero a diferencia de las entrevistas estructuradas, la aplicación del cuestionario no requiere que el analista esté presente exponiendo las preguntas al usuario.

Cuenta con ciertas características que pueden ser apropiadas en algunas situaciones e inapropiadas en otras. Al igual que las entrevistas deben diseñarse cuidadosamente para una máxima efectividad.

Existen dos formas de cuestionario para recabar datos: cuestionarios abiertos y cuestionarios cerrados, la aplicación de uno u otro tipo depende de que el analista conozca de antemano todas las posibles respuestas a las preguntas y pueda incluirlas.

- *Cuestionarios abiertos*: Se aplican cuando se quieren conocer las opiniones y experiencias generales, son útiles cuando se quiere explorar el problema básico y proporcionan una amplia posibilidad de respuesta.

- *Cuestionarios cerrados*: Limitan las respuestas posibles del usuario, por medio de un cuidadoso estilo de preguntas con las cuales el analista puede controlar el marco de referencia. Es el mejor método para obtener información específica.

- Revisión de registros

Con frecuencia la información ya se encuentra disponible para que los analistas conozcan las actividades u operaciones de la organización con las cuales no están familiarizados.

Se debe buscar la posibilidad de consultar los manuales escritos sobre política, regulaciones y procedimientos de operaciones estándares que la mayoría de las organizaciones mantienen como guía para gerentes y empleados. Los manuales que documentan o describen las operaciones para los procesos de datos existentes, o sistemas de información que están dentro del área de investigación, también proporcionan una visión sobre la forma en la que la organización se conduce.

Sin embargo se debe tener cuidado con la documentación, ya que en la mayor parte de las empresas los manuales y estándares sobre procedimientos de operación usualmente son obsoletos, a menudo no se mantienen al corriente para describir adecuadamente los procedimientos actuales.

Los documentos y formatos que se utilizan a través del proceso (aún los que están en blanco) pueden proporcionar conocimientos al analista sobre el sistema. Al comparar los formatos en blanco con el procedimiento y los manuales de operación se muestra al analista cómo deben llenarse; por lo tanto recabar y comparar los formatos ya completos permite señalar cualquier variación entre el uso real y el prescrito en los documentos.

Puede ser muy revelador tomar muestras de algunos de los documentos o formas utilizadas en el proceso.

- Recopilación de datos mediante observación

Observar las operaciones le proporciona al analista hechos que no podría obtener de otra forma, cuando el analista necesita ver de primera mano como se manejan los documentos, como se llevan a cabo los procesos y si ocurren los pasos especificados.

Siempre se deben identificar las tareas problemáticas que llevan a los empleados a cometer errores con frecuencia, así como aquellas que tienden a retardar el procedimiento.

2.3.2 Diagramas para el modelado del análisis

El proceso de comprender un sistema a desarrollar puede ser complejo, de ahí que el analista debe asistirse en dicha labor con herramientas diversas, para documentar de forma fácil y clara lo que el sistema debe realizar.

Una de estas herramientas es el modelado del sistema por medio de diagramas. El modelo generado por el análisis es una representación abstracta de la futura combinación del equipo computacional y los programas.

El término modelo lejos de ser un término formal, representa un concepto bien conocido por todos, ya que manejamos numerosos ejemplos de ellos a lo largo de nuestra vida: un mapa es un modelo bidimensional de

una zona del planeta, los diagramas de flujo representan esquemáticamente las decisiones y secuencias de actividades para un procedimiento determinado, etc.

El modelado permite hacer énfasis en puntos críticos del sistema y resta importancia a otros menos trascendentes, permitiendo una comunicación más enfocada con los usuarios además de ayudar a detectar si nuestra visión del sistema es la correcta y, de no serlo, ajustar o desechar el modelo creado y generar uno nuevo oportunamente.

Existen numerosos diagramas para el modelado de un sistema y que son propios de cada metodología de desarrollo; sin embargo, mencionaremos algunos diagramas generales de modelado que pueden aplicar en cualquier proyecto de desarrollo:

- Diagrama de flujo de datos.- modelado de las funciones del sistema.

- Diagrama entidad - relación.- modelado de la información almacenada en una base de datos.

- Diagrama de estructuras.- modelado de las estructuras de los programas.

Diagrama de flujo de datos

Este tipo de diagrama muestra las funciones que el sistema debe realizar y responde a preguntas tales como ¿Qué transformaciones (de datos) debe realizar el sistema? y ¿Qué entradas se transforman en qué salidas?

Los diagramas de flujo de datos incluyen elementos representados como procesos, flujos, almacenadores de datos y terminadores:

- Los *procesos* se incluyen en el diagrama como círculos o 'burbujas'. Representan las funciones individuales que el sistema lleva a cabo, dichas funciones transforman entradas en salidas.

- Los *flujos* son representados por medio de flechas, muestran conexiones entre los procesos y representa la información que dicho proceso requiere como entrada o genera como salida.

- Los *almacenadores de datos* se representan por medio de dos líneas paralelas o una elipse. Son conjuntos de datos que el sistema debe conservar por un período de tiempo. Cuando los diseñadores de sistema y programadores terminen de construir el sistema, éstos se convertirán en archivos o bases de datos.

- Los *terminadores* muestran entidades externas con las que el sistema se comunica, en la mayoría de los casos son individuos o grupos de personas, organizaciones externas u otros sistemas. La figura 2.1 muestra un ejemplo de diagrama de flujo de datos.

El diagrama de flujo de datos asiste al analista de sistemas con una visión global de los componentes funcionales del sistema, sin entrar en detalles sobre éstos.

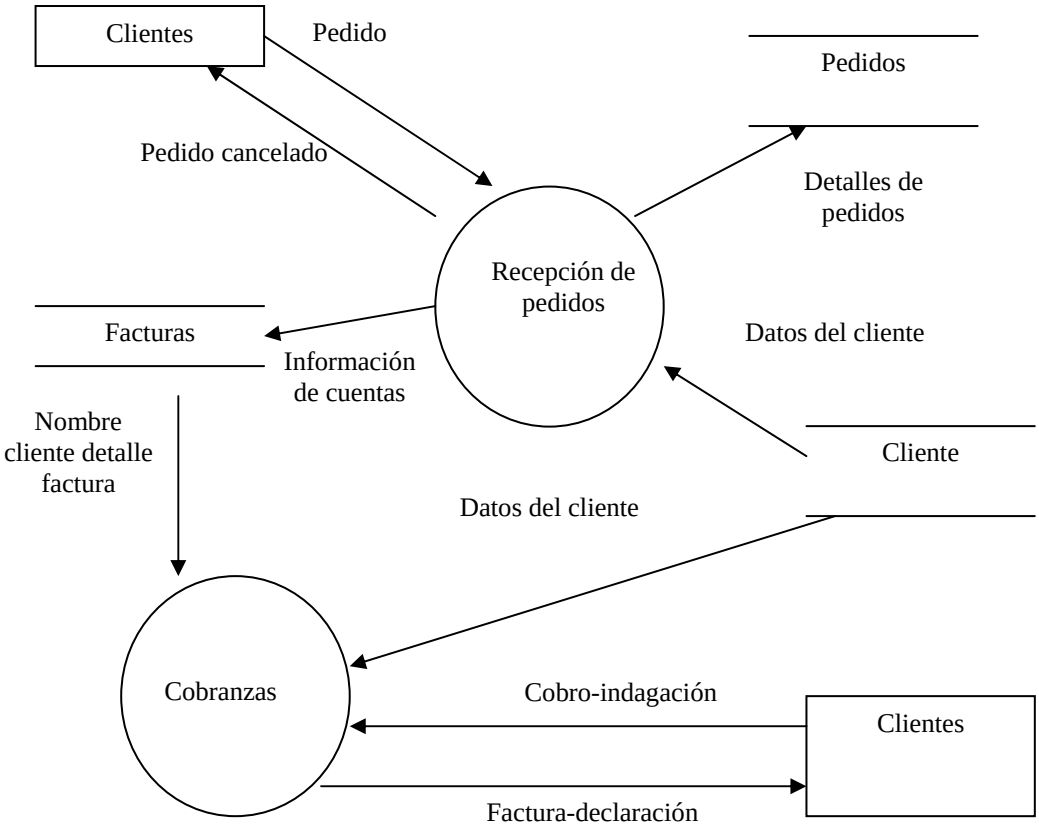


Figura 2.1 Ejemplo de un diagrama de flujo de datos. Las flechas (flujos) indican la dirección y la información que se intercambia entre procesos (círculos), almacenadores de datos (líneas paralelas) y terminadores (rectángulos). Esto permite ver de forma sencilla cómo se comporta cierto proceso.

Diagrama entidad - relación.

El diagrama Entidad-Relación hace énfasis en las relaciones entre los datos. Todos los sistemas informáticos almacenan, comparten y transforman información que la empresa requiere manejar; a veces, esta información es mínima, sin embargo en la mayoría de los sistemas es muy compleja.

Durante la etapa de análisis no sólo se debe conocer a detalle la información que hay en cada conjunto de datos sino la relación que existe entre cada conjunto.

Este diagrama consta de elementos fundamentales:

Con rectángulos se representa una entidad, una elipse representa un atributo o característica de dicha entidad y un rombo muestra la relación entre entidades diversas. La unión entre entidades se representa con flechas.

Existen diversos tipos de relación entre entidades

- Relación uno a uno

Representa la relación entre dos entidades, en las cuales un elemento de esa entidad sólo puede estar relacionado con un elemento de la otra. Un ejemplo de relación uno a uno se puede ver en la figura 2.2



Figura 2.2 En esta relación uno a uno, los atributos de la entidad "Automóvil" son el modelo, la clave de las placas y el color. Para "Tarjeta de circulación" los atributos son el propietario, el número de serie y el tipo. Ambas entidades están relacionadas por el registro vehicular y se considera una relación uno a uno porque un auto sólo tiene una tarjeta de circulación y una tarjeta de circulación sólo le pertenece a un automóvil.

- Relación uno a muchos

Representa una relación en la cual una de las entidades tiene que ver con muchos casos de otra entidad. La figura 2.3 muestra una relación uno a muchos.

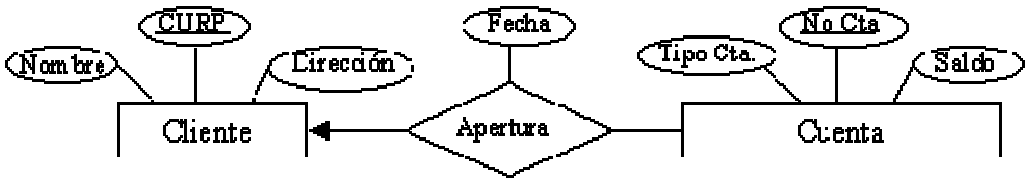


Figura 2.3 En este caso se considera una relación uno a muchos porque un cliente puede tener varias cuentas bancarias, sin embargo, una cuenta bancaria sólo tiene un cliente como titular.

Diagrama de estructura.

Esta herramienta gráfica de modelado se utiliza para representar la jerarquía de software.

Aquí cada rectángulo representa un módulo del sistema (por ejemplo un programa que a su vez se comunica con otros para formar el sistema completo) y la comunicación entre módulos es representada por medio de flechas.

En este diagrama se pueden incluir los diversos parámetros de entrada que requiere cada módulo invocado y los parámetros de salida devueltos por cada módulo cuando termina su proceso.

Aunque es una herramienta excelente para los diseñadores de sistemas, no es del tipo de modelo que normalmente se mostrará al usuario, pues modela aspectos de la implantación del sistema y no de sus requerimientos.

Estos diagramas muestran tanto jerarquía funcional como las interfaces de los datos entre los componentes.

Los principales componentes son:

- El rectángulo.
- Las flechas

El rectángulo en un diagrama de estructura no representa una declaración sino que representa un módulo, por ejemplo un procedimiento de algún lenguaje de programación.

Las flechas que conectan los módulos representan llamadas a otras partes del código; la notación implica que un programa parcial terminará o regresará al código que lo ejecutó cuando termine de realizar su función. Además, existen aquí dos tipos adicionales de flechas: la primera con un círculo vacío en uno de sus extremos que representa la transferencia de datos y la segunda, con un círculo lleno en uno de sus extremos, representa la transferencia de información de control.

La figura 2.4 ejemplifica un diagrama de estructura:

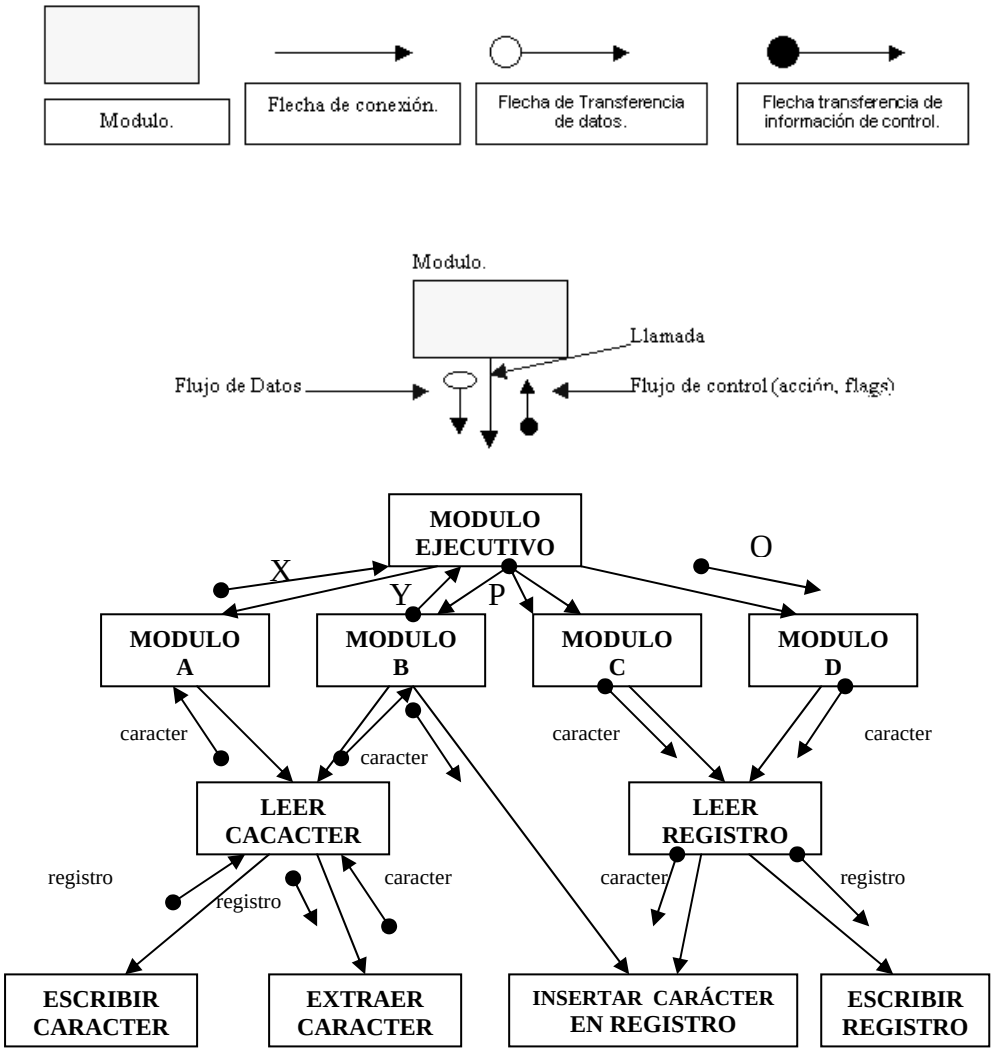


Figura 2.4 El diagrama muestra la estructura jerárquica de un conjunto de módulos en un sistema cualquiera. Los módulos superiores establecen conexiones o llamadas a módulos inferiores, para ello envía información de control y recibe datos como respuesta.

2.3.3 La toma de decisiones en la etapa de análisis

El proceso de toma de decisiones siempre estará presente en el análisis de sistemas informáticos, al momento de encontrar varias opciones o vías de solución a un requerimiento o problema.

Existen cinco características de las decisiones:

- *Efectos futuros*. Es en la medida en que los compromisos relacionados con la decisión afectarán el futuro.
- *Reversibilidad*. Es la velocidad con que una decisión puede revertirse y la dificultad que implica hacer este cambio.
- *Impacto*. Esta característica se refiere a la medida en que otras áreas o actividades se ven afectadas.
- *Calidad*. Este factor se refiere a las relaciones laborales, valores éticos, consideraciones legales, principios básicos de conducta, imagen de la compañía, etc.
- Periodicidad. Responde a la pregunta de si una decisión se toma de forma frecuente o excepcional.

Al realizar la toma de decisiones se deben considerar ciertas condiciones las cuales las podemos clasificar en:

- *Condición de certidumbre***.** No es común, especialmente cuando se trata de decisiones muy importantes. En este tipo de circunstancias las decisiones no son muy difíciles por que se conocen todos los resultados que se podrían presentar en el caso de seleccionar una u otra alternativa.
- *Condiciones de incertidumbre*. Se conocen de manera general las posibles consecuencias, pero no se cuenta con suficiente información como para asignar una probabilidad a cada situación.
- *Condiciones de riesgo***.** Se conocen las posibles consecuencias y se cuenta con suficiente información como para asignar una probabilidad a cada situación con la capacidad de poder correr el riesgo.

    - *Riesgo de requerimientos*.- Es posible que construya el sistema erróneo, un sistema que no haga lo que quiere el cliente. Durante la etapa de elaboración, el analista deberá entender bien los requerimientos y sus prioridades relativas.
    - *Riesgos tecnológicos*.- Se deben tener en cuenta las características del hardware y del software que se van a utilizar para el desarrollo y los equipos en donde se va a instalar el sistema.
    - *Riesgos de habilidades*.- El conocimiento y experiencia es una forma de evitar errores. Cometer errores consume tiempo y el tiempo cuesta.
    - *Riesgos políticos*.- Pueden existir fuerzas políticas que se interpongan en el camino y afectar seriamente el proyecto

La toma de decisiones se clasifica de acuerdo con el número de participantes en el proceso como:

- *Decisiones independientes***.** Son tomadas en forma aislada por una persona.

- *Decisiones secuenciales*. Son generadas por un grupo de personas, generalmente ejecutivos de la organización.

- *Decisión simultánea***.** Solo se toma en grupos, son producto de la interacción y negociación entre varias personas en forma simultánea.

Los pasos en el proceso de la toma de decisiones son como se muestran en la figura 2.5.

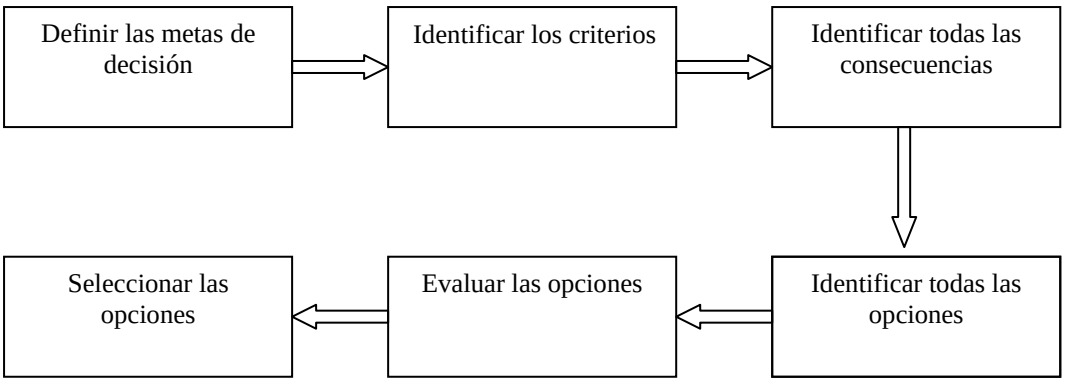


Figura 2.5 Pasos en el proceso de toma de decisiones

- Definir en forma precisa la meta de la decisión.- El proceso de toma de decisiones comienza con el reconocimiento de que se necesita. Por lo general parte de la existencia de una necesidad o una disparidad entre cierto estado deseado y la condición real del momento.

- Identificar todos los criterios de decisión relevantes.- Una vez determinada la necesidad de tomar una decisión, se deben identificar los criterios que sean importantes y cumplan con sus expectativas.

- Identificar todas las consecuencias posibles de cada alternativa.- Los criterios enumerados en el paso previo no tienen igual importancia. Es necesario ponderar cada una de las consecuencias y priorizar su importancia en la decisión.

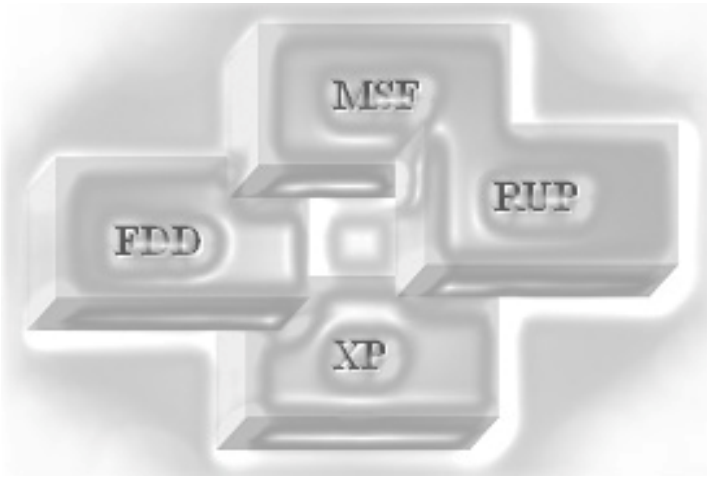
- Identificar todas las opciones de decisión relevante.- Se tiene que realizar una lista de todas las opciones posibles y que podrían utilizarse para resolver el problema.

- Evaluar las opciones respecto a los criterios para obtener un orden preferencial calificado.- Una vez identificadas las opciones, el tomador de decisiones debe evaluar de manera crítica cada una de ellas. Las ventajas y desventajas resultan evidentes cuando son comparadas. La evaluación de cada opción se hace analizándola respecto a un criterio ponderado.

- Seleccionar la opción que califique más alto la preferencia.- Una vez seleccionada la mejor opción se llegó al final del proceso de toma de decisiones. El tomador de decisiones solo tiene que escoger la opción que tuvo la calificación más alta en el paso anterior.

CAPÍTULO 3

Metodologías Diversas



El desarrollo de sistemas informáticos comenzó de manera experimental en el mundo, los desarrolladores creaban sistemas muy pequeños a partir de algunos requerimientos expuestos de forma verbal y sencilla.

Paulatinamente el desarrollador de sistemas se enfrentó con mayores problemas de complejidad, de presupuesto y de tiempo, lo cual provocaba que no todos los proyectos emprendidos terminaran exitosamente.

En consecuencia, diversos profesionales de la informática aportaron toda su experiencia en propuestas de formalización de los mecanismos que personalmente observaron les llevaban a obtener mejores resultados en el desarrollo de un sistema, lo cual se concretó en forma de metodologías.

Las metodologías de desarrollo de sistemas informáticos son una serie de pasos que se realizan bajo lineamientos o reglas probadas con anterioridad en numerosas ocasiones y que demostraron su eficiencia.

Las metodologías se fueron especializando, adaptando sus prácticas para realizar exitosamente sistemas que contaban con ciertas características de tamaño, tiempo, presupuesto, etc. dando lugar a metodologías ligeras, como XP o FDD, las cuales aplican a sistemas pequeños que deben entregarse al usuario en tiempos cortos, y a las metodologías pesadas, como MSF y RUP, que buscan realizar un sistema de grandes dimensiones y que se realiza en tiempos mayores que los ligeros.

De la misma manera que existen proyectos grandes y pequeños, urgentes o planeados, a corto, mediano o largo plazo, existen metodologías que se aplican en uno u otro tipo de proyecto de acuerdo a sus características específicas. Algunas de ellas se construyeron sólo con el desarrollo de sistemas informáticos como objetivo, sin embargo otras buscan ser tan amplias y generales que aplican a desarrollo de sistemas de cualquier tipo.

Describiremos de manera general cuatro de las metodologías más utilizadas, las cuales serán el soporte del procedimiento propuesto, objeto de este trabajo.

3.1 Programación Extrema (XP)

Kent Beck creó en 1996 la metodología de desarrollo de sistemas llamada Programación Extrema (Extreme Programming) o XP, que consiste en un conjunto de lineamientos tipo ligero que se basa en principios de simplicidad, comunicación y realimentación.

Está diseñada para ser usada por equipos pequeños que requieren desarrollar software de forma rápida en un ambiente en el que los requerimientos cambian rápidamente. Existen prácticas clave en XP, las cuales deben tomarse como guías y no como reglas:

- El proceso de planeación

Busca determinar rápidamente el alcance de la versión siguiente, combinando prioridades de negocio indicadas por el usuario con estimaciones técnicas de los programadores para elegir qué se necesita hacer y qué se necesita aplazar. Los programadores estiman el esfuerzo necesario para implementar los requerimientos del usuario y éste a su vez decide sobre el alcance y la agenda de entregas.

- Entregas pequeñas y frecuentes

XP pone un pequeño subsistema en producción rápidamente, al menos uno cada dos o tres meses, y lo actualiza frecuentemente en un ciclo muy corto. Se pueden liberar nuevas versiones incluso diariamente, pero al menos se debe liberar una nueva cada mes, cada una con pequeños cambios respecto de la anterior.

- Metáforas del sistema

Una metáfora se puede interpretar como la arquitectura simplificada del sistema.

El sistema se define a través de una o varias metáforas, una “historia” compartida por usuarios, administradores y programadores que orienta todo el sistema describiendo su funcionamiento. Los equipos de desarrollo XP usan una nomenclatura y un sistema de descripción comunes a los involucrados, lo cual guía el desarrollo y la comunicación.

- Diseño simple

Se hace énfasis en el desarrollo de la solución más simple susceptible de ser implementada en el momento. Se eliminan complejidades innecesarias y código extra, definiendo de mejor forma la menor cantidad de código posible. No existe mucho desarrollo que prevea futuras características del sistema. En lugar de ello, el objetivo es proporcionar un valor de negocio, asegurando un buen diseño por medio de la “refactorización” que se explica más adelante.

- Pruebas

El desarrollo del sistema está enfocado a las pruebas. El usuario colabora en el desarrollo de pruebas funcionales antes de que se desarrolle el código. El objetivo real del código no es cubrir un requerimiento, sino pasar las pruebas. Las pruebas y el código son escritos por el mismo programador, pero las pruebas se realizan sin intervención del desarrollador y se trabaja bajo el esquema de “todo o nada”, es decir, funciona del todo o no funciona.

Existen dos clases de prueba: la unitaria, que verifica un solo elemento o conjunto muy limitado de código y la de aceptación, que verifica el sistema entero o gran parte de él.

- Refactorización

El término refactorización (del inglés refactoring), introducido por W. F. Opdyke en su tesis doctoral, se refiere “al proceso de cambiar un sistema de software de tal manera que no se altere el comportamiento exterior del código, pero se mejore su estructura interna”.

Se refactoriza el sistema eliminando duplicación de código, mejorando la comunicación entre procesos y aumentando la flexibilidad sin cambiar la funcionalidad. El proceso consiste en una serie de pequeños cambios que alteran la estructura interna conservando el comportamiento aparente del código.

- Programación por pares

Los programadores XP escriben todo el código de producción en parejas. Dos programadores trabajan juntos en una misma máquina, el que no está escribiendo analiza la situación desde otro punto de vista y realiza lo que podría llamarse revisión de código en tiempo real; los roles pueden cambiarse varias veces al día. Esta técnica ha demostrado que genera mejor software a un costo similar o menor que con programadores trabajando solos.

- Propiedad colectiva

Todo el código pertenece a todos los programadores. Cualquier programador puede alterar cualquier parte del código, siempre que escriba antes la prueba correspondiente. Esto le permite al equipo trabajar a su máxima capacidad, ya que cuando algo requiere cambios, se pueden hacer sin retrasos.

- Integración continúa

Cada segmento de código creado, al momento de estar listo, se agrega al sistema. Primero se corre la prueba del segmento recién creado y habiéndola pasado, se integra al sistema completo.

Los equipos de desarrollo XP integran y construyen el sistema múltiples veces al día. Esto mantiene a los programadores al día y permite un progreso muy rápido. Más aún, la integración frecuente del sistema ayuda a eliminar problemas que se presentan en sistemas que lo hacen esporádicamente.

- Ritmo sostenible con un máximo de 8 horas al día

Dado que la práctica de desarrollo de software es una actividad creativa, se estima que el estar fresco y descansado ayuda a cumplirla de forma eficiente; con ello se motiva a los participantes, se evita la rotación de personal y se mejora la calidad del producto minimizando errores y eliminando “procesos neuróticos”.
Los programadores que presentan cansancio cometen más errores. Los equipos de desarrollo XP no trabajan tiempos extras de forma excesiva, manteniéndose frescos, sanos y efectivos.

- Todo el equipo en el mismo lugar

Un proyecto XP es dirigido por un usuario dedicado que debe estar siempre presente y disponible tiempo completo para el equipo de desarrollo. Debe tener las facultades para determinar requerimientos, asignar prioridades y aclarar dudas cuando los programadores las tengan. El efecto de estar ahí es que la comunicación mejora con menos documentación.

- Estándar de codificación

Se deben seguir reglas de codificación y los desarrolladores deberán comunicarse a través del código. Ya que XP es regido por el purismo en la programación, los comentarios en el código no son bien vistos, ya que si el código es lo suficientemente oscuro como para requerir comentarios, se debe reescribir o refactorizar. Para que un equipo trabaje de forma efectiva en parejas, y para compartir la propiedad de todo el código, es necesario que todos los programadores escriban código de la misma manera, con reglas que aseguren que el código se transmita claramente.

- Espacio abierto

El lugar de trabajo preferentemente será una sala grande con cubículos pequeños, o aún mejor, sin divisiones. Las parejas de programadores deben estar en el área central y en la periferia se colocan las máquinas privadas.

- Reglas justas

Cada equipo XP tiene sus propias reglas a seguir, pero se pueden cambiar en cualquier momento. XP asume que no existe un proceso universal que cubra adecuadamente todos los proyectos; lo habitual es adaptar un conjunto de prácticas simples a las características de cada proyecto.

En general XP es un proceso basado en iteraciones. En la figura 3.1 se muestra esquemáticamente el ciclo de vida de un proyecto XP:

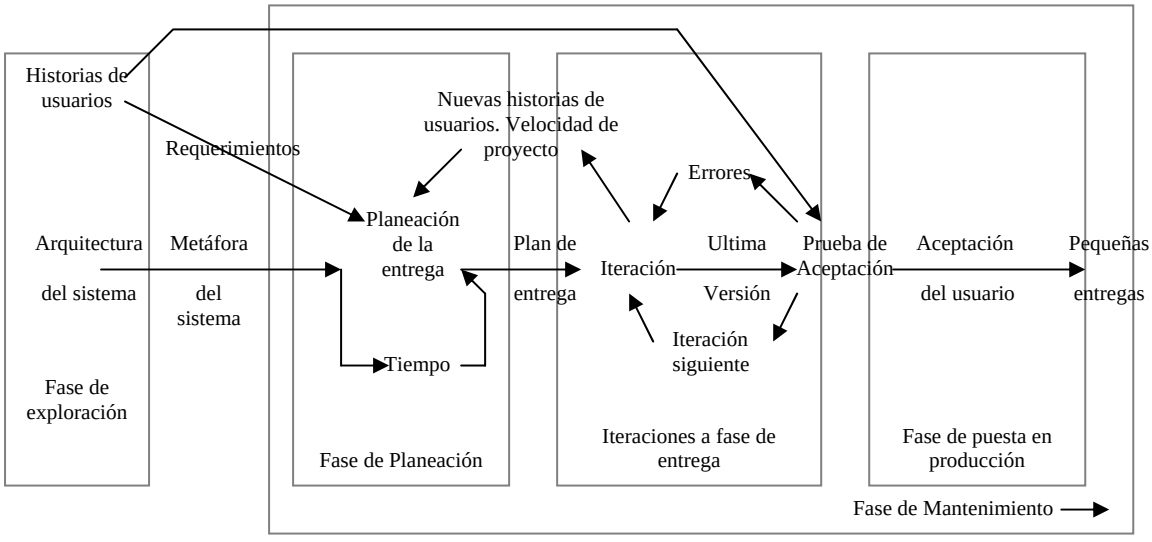


Figura 3.1 El diagrama muestra la naturaleza iterativa de XP, la cual consiste en ciclos breves que dependen principalmente de los requerimientos dinámicos del usuario y de pruebas continuas a sistemas parciales para generar pequeñas entregas de manera frecuente.

La arquitectura del sistema se genera en forma de una metáfora durante la fase de exploración.

En la fase de planeación las historias iniciales y nuevas de los usuarios (requerimientos) permiten planear la entrega (lo cual cuesta tiempo). El plan de entrega obtenido de la fase anterior se debe seguir hasta llegar a una prueba de aceptación corrigiendo errores y adaptando nuevos requerimientos del usuario. Cuando se logra ejecutar y pasar la prueba de aceptación se alcanza una nueva entrega pequeña que deberá pasar a la fase de mantenimiento.

3.2 Desarrollo Enfocado a Características (FDD)

El Desarrollo Enfocado a Características (Feature Driven Development) o FDD es un proceso ligero y está pensado para proyectos con tiempo de desarrollo relativamente corto (menos de un año). Se basa en un proceso con iteraciones cortas (aproximadamente 2 semanas) que producen un software funcional, entregando al usuario un informe constante sobre la situación del proyecto. Las iteraciones se deciden con base en características o funcionalidades que son pequeñas partes del software con significado para el usuario.

En la figura 3.2 se muestran las 5 fases de un proyecto FDD las cuales se describen a continuación:

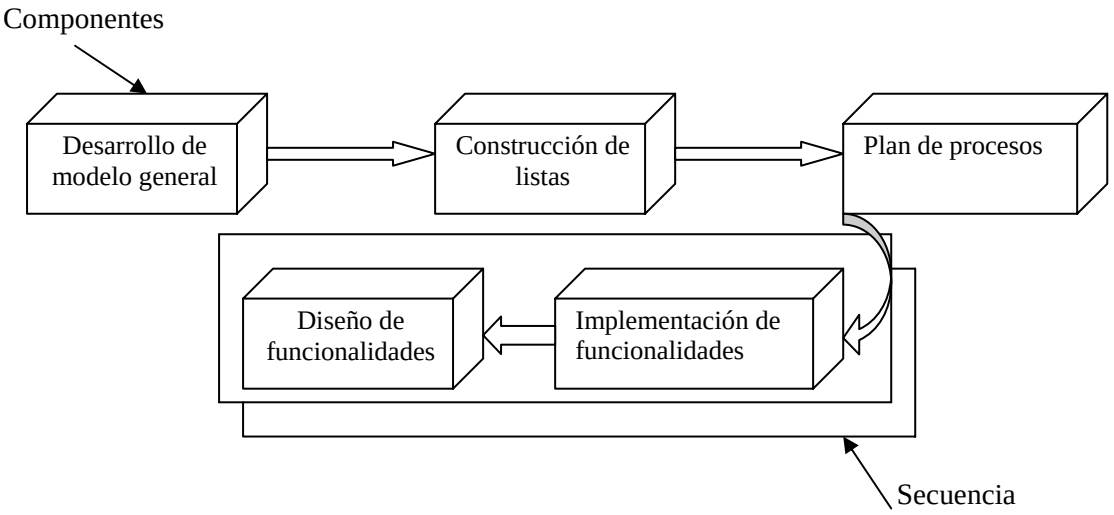


Figura 3.2 Fases generales en las que se divide un proyecto que emplea la metodología FDD

- Desarrollo de un modelo general.

Los usuarios y los desarrolladores trabajan juntos para realizar una entrada de información en la cual se especifican y definen los criterios claramente, los cuales ayudan al equipo de desarrollo a alcanzar un resultado notable.

- Construcción de la lista de funcionalidades

Se realiza una lista de las características especificando las prioridades del proceso, esto se realiza con base en los requerimientos existentes, tales como casos de uso, especificaciones, funcionalidades, entrevistas con el usuario, etc.

Como resultado de este proceso, el equipo de desarrollo debe entregar una lista detallada de características, agrupadas de forma general a particular, la cual estará sujeta a revisión y aprobación por parte del administrador de desarrollo y el arquitecto en jefe.

- Plan de procesos con base en las funcionalidades a implementar

Utilizando la lista de características jerárquica y priorizada, el administrador del proyecto, el administrador de desarrollo y los programadores en jefe establecen las bases para las iteraciones que siguen la regla "diseñar por característica, construir por característica".

Se especifican los pasos para la verificación del proceso, se genera un plan de alto nivel y es asignado a los programadores siguiendo el modelo general.

El equipo construye una lista informal de características, un trabajo inicial antes de comenzar el proceso FDD. El equipo toma notas sobre referencias específicas (documento y números de página) de documentos disponibles, conforme sea necesario.

- Diseño con base en las funcionalidades

Un programador en jefe toma alguna característica, identifica el código que este involucrado y contacta a los propietarios de dicho código. Los propietarios trabajan a su vez en un diagrama de flujo de datos detallado y conducen una inspección de diseño.

El desarrollo del sistema y la documentación necesaria se tienen que realizar en un tiempo corto para que posteriormente se seleccione un grupo pequeño de usuarios que realicen las pruebas necesarias sobre éste y verificar que sea de calidad.

- Implementar con base en las funcionalidades

Se implementa el software realizado y se empieza a capacitar al personal para el manejo del sistema. Las primeras tres fases ocupan gran parte del tiempo en las primeras iteraciones. Conforme avanza el proyecto son las dos últimas fases las que absorben la mayor parte del tiempo, limitándose las primeras a un proceso de refinamiento.

El trabajo (tanto de modelado como de desarrollo) se realiza en grupo, aunque siempre habrá un responsable. Las funcionalidades a implementar en el software se dividen entre los distintos subgrupos del equipo. La programación escrita tiene propietario; es decir, solo quien la crea puede cambiarla.

3.3 Proceso Racional Unificado (RUP)

El Proceso Racional Unificado (Rational Unified Process) o RUP es una metodología de desarrollo de software que provee un enfoque disciplinado para la asignación de tareas y responsabilidades. Esto asegura la calidad en un sistema que cumple con los requerimientos de los usuarios finales dentro del tiempo establecido y lo presupuestado. RUP es uno de los procesos más generales de los que existen actualmente, ya que está pensado para adaptarse a cualquier proyecto y no tan solo a desarrollo de software.

Un proyecto realizado siguiendo RUP se divide en cuatro fases:

- Inicio del proyecto
- Elaboración (definición, análisis, diseño)
- Construcción (implementación)
- Transición (fin del proyecto y puesta en producción)

En cada fase se ejecutarán una o varias iteraciones (de tamaño variable según el proyecto) y dentro de cada una de ellas se seguirá una versión modificada del modelo en cascada que pasa por los distintos flujos de trabajo como se muestra en la figura 3.3.

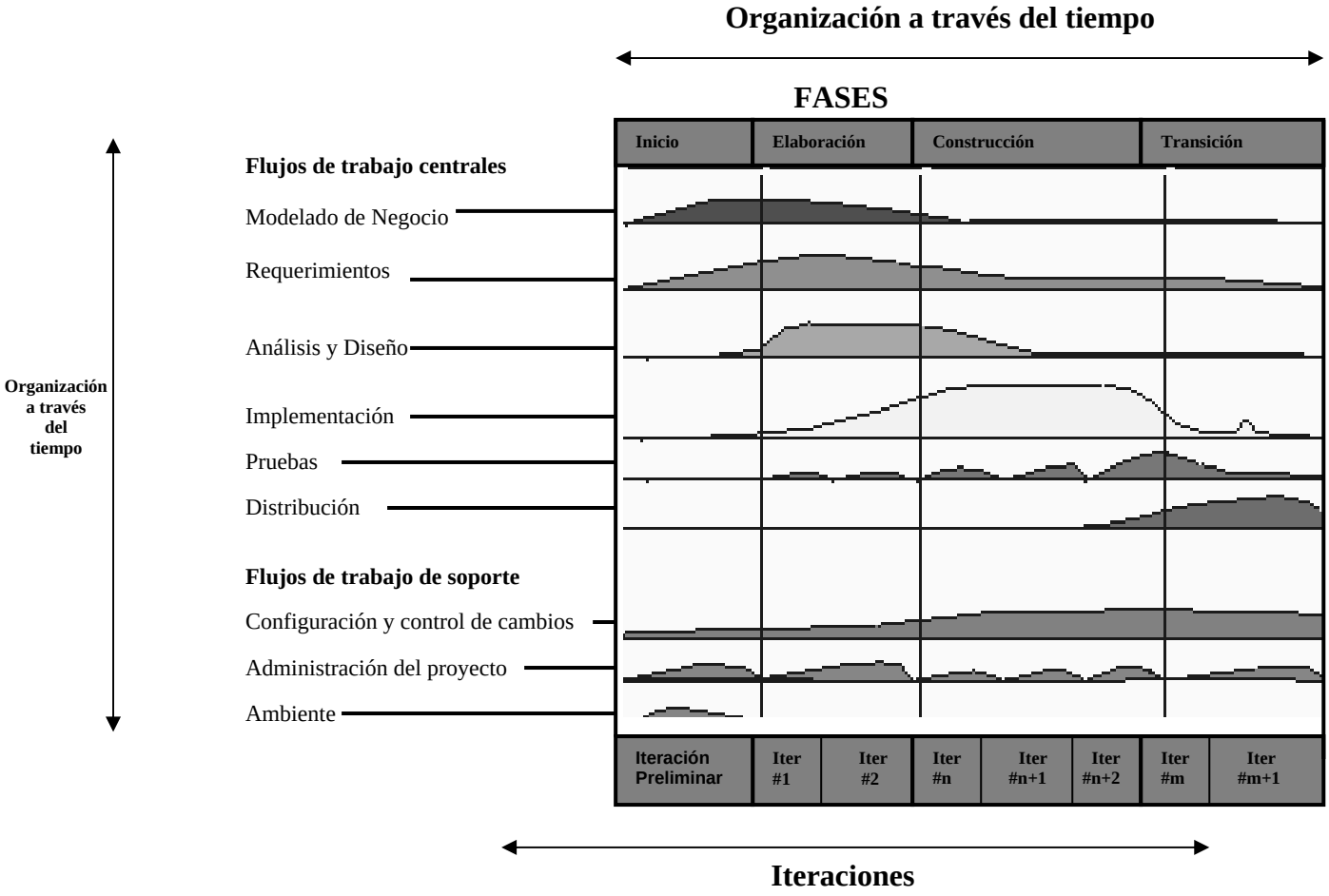


Figura 3.3 En este diagrama se muestra que las nueve actividades (modelado del negocio, requerimientos, análisis y diseño, etc) se pueden presentar en diversas fases del proyecto (Inicio, elaboración, construcción y transición) formando una matriz de trabajo.

RUP define nueve actividades a realizar en cada fase del proyecto.

- Modelado del negocio

Las metas del modelado del negocio son:

- Entender la estructura y la dinámica de la organización en la cual el sistema se empleará (organización objetivo)
- Entender los problemas actuales de la organización objetivo e identificar mejoras potenciales
- Asegurar que los clientes, usuarios y desarrolladores tengan la misma visión de la organización objetivo
- Abstraer los requerimientos del sistema necesarios para darle soporte a la organización

- Análisis de requerimientos

Los objetivos del análisis de requerimientos son:

- Establecer y mantener un acuerdo con el usuario en lo que el sistema debe hacer y por qué
- Proporcionar a los desarrolladores un mejor entendimiento de los requerimientos del sistema
- Delimitar el sistema
- Proporcionar una base para planear el contenido técnico de las iteraciones
- Proporcionar una base para estimar tiempos para el desarrollo

- Análisis y diseño

El objetivo del análisis y diseño es traducir la forma en que se entienden los requerimientos transformándolos en un diseño del sistema que seleccione la mejor estrategia de creación e implementación.

- Implementación

La implementación tiene tres propósitos

  - Definir la organización del código en términos de subsistemas organizados en capas o niveles
  - Probar los componentes desarrollados como unidades
  - Integrar en un sistema ejecutable los resultados producidos por implementaciones individuales o de equipos

- Pruebas

El propósito de las pruebas es asegurar la calidad de producto. Esto no sólo involucra el producto final, sino que empieza desde las primeras fases del proyecto con el establecimiento de la arquitectura y continúa hasta la entrega del producto final al usuario.

- Distribución

Los desarrolladores de software asumen demasiado pronto, en numerosas ocasiones, que el proyecto ha terminado. Olvidan que es el deseo del usuario final de usar el producto, más que un código limpio, lo que establece la marca de una entrega exitosa. El objetivo de la distribución es dejar el sistema en manos de los usuarios.

- Configuración y cambios

El objetivo es llevar un control y mantener la integridad de los hechos que involucra el proyecto. Se deben organizar tareas y herramientas de los programadores, lo cual implica establecer una adecuada administración de cambios al sistema.

- Administración de proyecto

Se busca balancear objetivos, administrar riesgos y superar restricciones para entregar un producto que satisfaga los requerimientos expuestos, tanto los de usuarios de alto nivel, como de usuarios operativos.

- Gestión de entorno

El propósito es dar soporte al desarrollo, tanto con procesos como con herramientas e incluye lo siguiente:

  - Selección y adquisición de herramientas de programación
  - Puesta a punto de herramientas para ajustarlas a la infraestructura de la organización
  - Configuración de procesos
  - Mejora de procesos
  - Servicios técnicos para dar soporte al proceso

La figura 3.4 muestra cómo el ciclo iterativo de RUP pasa por los flujos de trabajo de manera repetitiva.



Figura 3.4 Cada iteración realizada en un flujo de trabajo RUP comienza con el modelado del negocio y la planeación inicial, pasa por el estudio de requerimientos, análisis y diseño, implementación y pruebas, continuando este ciclo hasta que las pruebas se superan y se pasa a la distribución.

RUP se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, documentándose lo mejor posible.

Un caso de uso es un modelo de la funcionalidad del sistema según lo perciben los usuarios externos y la representación de una interacción general entre el usuario y el sistema.

3.4 Marco de Soluciones Microsoft

El Marco de Soluciones Microsoft (Microsoft Solutions Framework) o MSF, es una herramienta de desarrollo de soluciones. Surge en 1994 como respuesta al alto porcentaje de proyectos de desarrollo que terminan sin éxito.

La metodología MSF consta de 4 modelos:

3.4.1 Modelo de equipos de iguales

Se trabaja bajo el concepto de que todos ganan o todos pierden dentro del desarrollo de un sistema. Este primer modelo incluye seis roles de trabajo:

- *Administrador del Producto (Product Management)*. Se encarga de que el cliente esté satisfecho con el producto que desea.

- Administrador del Programa (Program Manager). Se encarga de que las restricciones planteadas por el Administrador del Producto se lleven a cabo (tiempo, dinero, recursos, etc.).

- *Desarrollador (Development). S*e encarga de construir el software de acuerdo con las especificaciones del análisis.

- *Probador (Testing).* Se encarga de que todos los “issues” o inconvenientes del sistema sean conocidos.

- *Administrador de Logística (Logistic Management).* Se encarga de que las implementaciones sean suaves; es decir, que toda la estructura de software y hardware donde se montará la aplicación esté disponible, también se encarga de que el sistema se pueda operar y administrar después de terminado.

- *Capacitador de Usuario (User Education)*. Se encarga de que la interacción usuario-sistema tenga un rendimiento óptimo; es decir, que el usuario realice las mismas actividades que desempeñaba, pero de

forma más eficaz. Los procesos serán mejores, lo que llevará a un mejor rendimiento. En este punto es importante considerar que una de las razones por la que no se utilizan los sistemas es debido a que los usuarios no los saben usar, de ahí la preocupación por que el usuario sea entrenado para utilizar el sistema desarrollado.

3.4.2 Modelo o proceso de administración de riesgos

Riesgo es definido en MSF como la posibilidad de un problema a la espera de ocurrir; es decir, aceptar que ciertas condiciones de riesgo son susceptibles de presentarse durante la vida del proyecto y se debe estar preparado para resolverlas en caso de que ocurran.

Proceso de Administración de Riesgos:

- Identificarlo
- Analizarlo
- Planearlo
- Dar seguimiento
- Control (verificar que el plan tenga efecto)

Las características de alta prioridad con más riesgo se programan en una fase temprana del proyecto. De este modo se amplía al máximo el tiempo disponible para reaccionar ante problemas.

Administración del tiempo adicional:

- Se agrega tiempo adicional a los plazos del proyecto para permitir al equipo acomodarse a los problemas y cambios que se produzcan de manera inesperada.

- La cantidad de tiempo adicional que debe aplicarse depende del nivel de riesgo.

- Al evaluar los riesgos en la fase inicial del proyecto, es posible calificar los riesgos más probables para conocer el impacto que éstos tendrán en los plazos y cómo se pueden compensar usando tiempo adicional.

Un modo de pensar en el tiempo adicional es como si fuesen una estimación para tareas y eventos desconocidos. No importa la experiencia que tenga el equipo, no todas las tareas de los proyectos pueden conocerse ni se puede realizar una estimación de ellas con antelación. Sin embargo, es prácticamente cierto que habrá riesgos en algunos proyectos y tendrán un impacto en los mismos, y que las acciones correctoras necesarias para responder al riesgo necesitarán más tiempo.

3.4.3 Modelo de proceso

Pasos a seguir para llevar a cabo el desarrollo:

- Visión. Plantea cuál es el sueño del usuario, cuál es su objetivo con el proyecto.
- Planeación. Plantea una serie de pasos para llegar al sueño del usuario.
- Diseño. Se crea el sueño del usuario.
- Estabilización. Se garantiza la firmeza del desarrollo.

En todo proceso las etapas permiten puntos de revisión. Cuando se realiza un entregable se tiene una evidencia de que el equipo alcanzó una etapa.

Plantea los principios de un proceso exitoso:
- Documentos vivientes. Documentos que están cambiando constantemente (dinámicos).
- Versiones. Manipular por épocas o tiempos estimados (1.0, 2.0, etc.).

- Elementos críticos de negociación: recursos, características y calendario. Son elementos que forman un sistema, si uno de estos elementos cambia, se alteran los otros dos como se muestra en la figura 3.5.

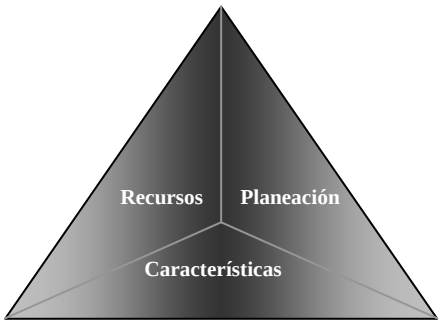


Figura 3.5 Elementos críticos de negociación

3.4.4 Modelo de aplicaciones

Una aplicación es un servicio, en informática un servicio es una unidad de código que realiza una tarea o tiene una función

Tipos de servicios:
- Servicios o aplicaciones de usuario. Los que interactúan con el usuario final, interfaz.
- Servicios o aplicaciones de negocio. Los que le dan valor a los datos, reglas de negocio.
- Servicios o aplicaciones de implementación o acceso a datos. Para este modelo se asume que datos son todo aquello a lo que se tiene acceso con un servicio de datos. Procesamiento, es la conversión que hacemos de los datos con los servicios de negocio. Información, es la suma de datos más procesamiento.

3.5 Comparativo de metodologías

Con base en las cuatro metodologías anteriores se presenta el siguiente cuadro comparativo, el cual muestra las características y diferencias de cada una de ellas.

Característica	XP	FDD	MSF	RUP
Programación en equipos por actividad	Sí	No	Sí	No
Ligero	Sí	Sí	No	No
Orientado a cualquier tipo de sistema (no sólo informático)	No	No	No	Sí
Orientado sólo a desarrollo de software	Sí	No	Sí	No
Manejo de entregas parciales	Sí	Sí	No	No
Metodología compleja que se aplica en proyectos pesados	No	No	No	Sí
Manejo de límite para el número de horas de trabajo	Sí	No	Sí	No
Desarrollo iterativo	Sí	No	Sí	Sí
Documentación formal	No	No	Sí	Sí
Administración de riesgos	No	No	Sí	Sí
Propiedad colectiva del software	Sí	No	Sí	Sí

Como podemos observar, las características propias de las diversas metodologías les permiten ser una opción a seguir en el desarrollo de sistemas en distintos niveles de complejidad.

Las metodologías que generalmente se aplican en proyectos grandes, con una operación compleja y con una cantidad considerable de procesos son RUP y MSF. Por el contrario, XP y FDD se pueden emplear en sistemas que requieren una menor cantidad de mecanismos de control para obtener un total conocimiento del proceso. Es por eso que estas dos últimas metodologías se consideran ligeras.

RUP es tan amplio que es el único que se puede aplicar a desarrollo de sistemas no informáticos, mientras que XP, FDD y MSF han sido desarrollados exclusivamente para este tipo de sistemas.

Un enfoque que favorece al equipo de desarrollo de software ante el usuario es el de entregas parciales funcionales. El usuario puede ver en poco tiempo resultados operantes y eso reduce la incertidumbre propia de desarrollos más largos. RUP realiza entregas parciales ya que cada iteración produce una nueva versión del sistema, sin embargo la labor realizada es tangible sólo hasta fases tardías como las de construcción y transición.

Como se puede observar, tres de las cuatro metodologías emplean la nueva tendencia en el desarrollo de sistemas que es el proceso iterativo, debido a las numerosas ventajas que ofrece este enfoque sobre el típico método de desarrollo en cascada.

Es de notarse que la generación de documentación depende del tamaño del proyecto. En sistemas pequeños el desarrollador y el usuario no requieren en mayor medida de la asistencia de la documentación para llevar un control efectivo del trabajo realizado o por realizar en un proyecto. Conforme los proyectos son más complejos, llevar un control de requerimientos, cambios, desarrollos, riesgos, prioridades, modelos, arquitecturas, etc. se hace más difícil y es indispensable contar con la documentación necesaria para consultar el desarrollo del proyecto.

En proyectos de menor complejidad el progreso se refleja en pequeños programas ejecutables parciales que debido a que fueron analizados en su totalidad de forma fácil y rápida, es posible desarrollarlos, probarlos e incluso liberarlos sin analizar demasiado desarrollos posteriores.

En proyectos de mayor complejidad los productos en forma de software funcional toman mucho más tiempo para ser una realidad, se debe seguir una logística mucho más compleja de desarrollo, implementación y pruebas para lograr que un módulo (formado por numerosos subsistemas más pequeños que a su vez tuvieron que ser generados en cierta secuencia, debido a las interacciones entre ellos) pueda ser liberado.

Así mismo, los riesgos no hacen necesaria una administración compleja en metodologías ligeras, ya que el costo de los riesgos aumenta con el tiempo y en proyectos pequeños los riesgos propios de cada requerimiento son detectados, eliminados o minimizados en un periodo de tiempo muy corto.

En proyectos complicados la administración de riesgos es vital para el resultado del proyecto, ya que de no llevarse un adecuado programa de detección, priorización y minimización o eliminación de riesgos el proyecto puede resultar en un fracaso o en altos costos no previstos derivados de los esfuerzos por eliminar los riesgos no detectados.

Otra práctica común que se observa en el comparativo es la propiedad colectiva del software. Este enfoque no busca eliminar aspectos como la propiedad intelectual y los derechos de autor, sino obligar a los desarrolladores a codificar de forma adecuada.

El desarrollador debe codificar de tal forma que cualquier otro programador en el equipo pueda leer cierto código como si fuera propio. Esto elimina líneas de código innecesario o ineficiente, comentarios que sólo ocupan espacio de almacenamiento y dificultan la lectura de un programa y principalmente malas costumbres en la codificación que son típicas de los desarrolladores. Con malas costumbres nos referimos a código que el desarrollador en su experiencia ha asumido como funcional, pero que es muy susceptible de ser optimado ya que es pobre en cuanto a desempeño.

En general, la diversidad de metodologías deja ver que no existe por ahora un procedimiento universal que aplique a proyectos de diversas complejidades, ya que no es posible utilizar metodologías ligeras en proyectos largos y complejos porque se perdería el control de numerosos aspectos del sistema y el resultado no plasmaría lo que el usuario requiere. Por el contrario, realizar todas las actividades de control que implica MSF o RUP en un proyecto pequeño podría significar el uso de un tiempo muy valioso en análisis redundantes y que podría utilizarse en desarrollo, pruebas o implementación.

CAPÍTULO 4

Metodología RUP



La metodología RUP se trata en mayor detalle en este capítulo, describiendo sus elementos fundamentales de desarrollo de sistemas. El método en cascada de desarrollo de sistemas es sustituido por un nuevo método de trabajo propuesto por RUP, garantizando que la calidad del software se pueda mantener de un modo predecible y repetible.

RUP basa sus actividades de desarrollo en una serie de pasos o fases principales, cada una de las cuales incluye las actividades típicas del enfoque en cascada (análisis, diseño, desarrollo, pruebas e implantación) en ciclos completos y repetitivos llamados iteraciones, pero con un marcado énfasis en ciertos aspectos de ellas. Dichas fases son inicio, elaboración, construcción y transición.

El enfoque iterativo permite detectar errores en el análisis del sistema de forma oportuna, corrigiendo problemas desde el principio y evitando que en un momento posterior sea más difícil y costoso resolverlos.

Dentro de cada fase existen actividades clasificadas en grupos con un enfoque común, las cuales cubren aspectos esenciales del desarrollo del sistema, todas ellas representadas en forma de flujos de trabajo (workflows).

4.1 Prácticas en el desarrollo de software con base en RUP

Como ya se ha mencionado, los proyectos de desarrollo de software que fracasan tienen síntomas comunes que los caracterizan, algunos de éstos son:

- Comprensión inexacta de las necesidades del usuario final
- Incapacidad para manejar requerimientos cambiantes
- Módulos que no ajustan entre sí
- Software que es difícil de mantener o ampliar
- Detección tardía de fallas serias en el proyecto
- Software carente de calidad
- Desempeño inaceptable del software
- Un proceso de construcción y entrega poco confiable

Desafortunadamente tratar estos síntomas no resuelve el verdadero problema y parece que la mayoría de los desarrollos de sistemas no funcionan debido a una combinación de las siguientes causas:

- Administración inadecuada de requerimientos
- Comunicación ambigua e imprecisa usuario-analista
- Diseños frágiles
- Demasiada complejidad
- Inconsistencias no detectadas en requerimientos, diseño e implementación
- Pruebas insuficientes
- Fallas al atacar los riesgos
- Propagación de cambios no controlados
- Automatización insuficiente

Si se abordan las causas mencionadas anteriormente, no solo se eliminan los síntomas, sino que se estará en una mejor posición para desarrollar y mantener software de calidad de un modo repetible y predecible.

Las prácticas del desarrollo de software comprenden un conjunto de técnicas probadas de desarrollo de software que cuando se usan en combinación, atacan las causas de origen de los problemas de desarrollo de sistemas.

Las prácticas son las siguientes:

- Desarrollo de software iterativo
- Manejo de requerimientos
- Uso de arquitecturas basadas en componentes
- Modelado de software de forma visual
- Verificación continua de la calidad de software
- Control de cambios de software

4.1.1 Desarrollo de software con el método iterativo

El proceso clásico de desarrollo de software sigue el ciclo en cascada, ilustrado en la figura 4.1 En este enfoque, el desarrollo procede linealmente desde el análisis de requerimientos y pasa por el diseño, codificación, pruebas de subsistemas y pruebas del sistema.

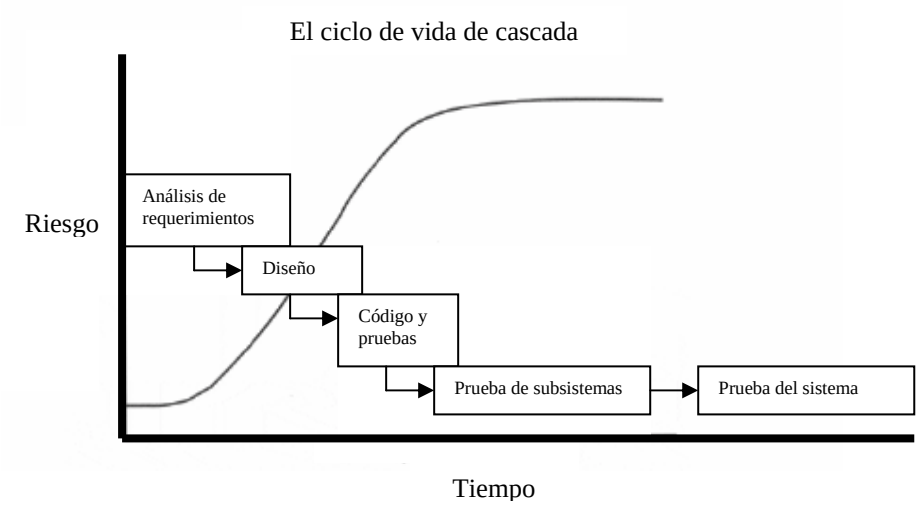


Figura 4.1 Comportamiento de los riesgos en cada punto del procesos de desarrollo en un enfoque de cascada, el cual provoca que cada vez sea más difícil resolver problemas no detectados a tiempo.

El problema fundamental de este enfoque es que aplaza los riesgos de modo que se vuelve costoso deshacer errores de fases anteriores. El diseño inicial probablemente tendrá fallas respecto a sus requerimientos principales, y posteriormente, el descubrimiento tardío de defectos de diseño tiende a resultar en costos, desarrollos extras o la cancelación de proyectos. El método de cascada no permite visualizar los riesgos reales de un proyecto hasta que es demasiado tarde para hacer algo sobre ellos.

Una alternativa al enfoque de cascada es el proceso iterativo e incremental que se muestra en la figura 4.2. En este enfoque la identificación de riesgos de un proyecto se realiza de forma temprana en el ciclo de vida, cuando es posible atacar y reaccionar a ellos de una manera eficiente y oportuna.

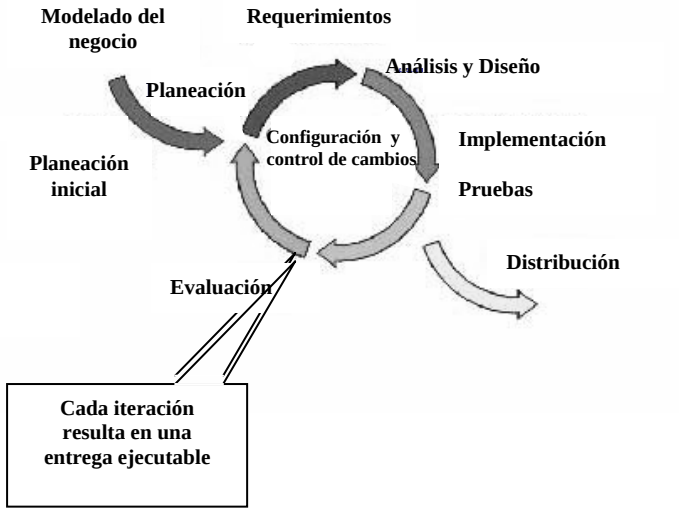


Figura 4.2 Diagrama de un proceso iterativo. Este enfoque trabaja en ciclos (iteraciones) donde se cumplen todas las actividades asignados a esa iteración. La siguiente iteración perfecciona lo logrado en la anterior. El proceso continúa hasta terminar el sistema en su totalidad.

Desarrollar software de forma iterativa ofrece varias soluciones a las causas de origen de problemas en el desarrollo de software:

- Se hacen evidentes de forma temprana serios malentendidos e inconsistencias en los requerimientos, diseños e implementaciones, cuando aún es tiempo de reaccionar adecuadamente
- Este enfoque permite e impulsa la realimentación con el usuario para visualizar los requerimientos reales del sistema
- El equipo de desarrollo se dedica a resolver los problemas clave del proyecto
- La carga de trabajo se distribuye de manera más uniforme a través del ciclo de vida del proyecto

4.1.2 Manejo de requerimientos

Un "requerimiento" es una condición, capacidad o característica con la que el sistema debe contar. El reto de administrar los requerimientos de un sistema es que son dinámicos; es decir, el equipo de desarrollo debe saber que es muy posible que cambien durante el tiempo de vida del proyecto, más aún, es un proceso continuo identificar cuáles de ellos son claves.

Una buena administración de los requerimientos del sistema ofrece las siguientes soluciones a las causas origen de problemas en el desarrollo de software:

- La comunicación entre analistas se facilita con requerimientos bien definidos
- Los requerimientos se pueden priorizar, filtrar y darles seguimiento
- Se detectan más fácilmente las fallas

4.1.3 Uso de arquitecturas basadas en componentes

Realizar las tareas de visualizar, especificar, construir y documentar un sistema requiere que éste sea visto desde diversas perspectivas. Cada uno de los participantes, usuarios finales, analistas, desarrolladores, integradores del sistema, probadores, documentadores técnicos y administradores perciben el proyecto desde un punto de vista distinto durante su ciclo de vida. La arquitectura del sistema es quizás el entregable más importante que se puede utilizar para administrar estos distintos puntos de vista y con ello controlar el desarrollo iterativo e incremental que buscamos.

La arquitectura de un sistema se define por medio de conceptos complementarios:

- La organización de un sistema de software
- La selección de elementos estructurales y sus interfaces por las cuales está compuesto el sistema
- La integración de estos elementos en subsistemas progresivamente mas grandes

La arquitectura de software no solo se ocupa de la estructura y comportamiento sino también del uso, funcionalidad, desempeño, reutilización, restricciones económicas y tecnológicas, comprensibilidad y elementos estéticos; es el borrador de las interfaces, organización y comunicación entre componentes, módulos y subsistemas de software y hardware que integran un sistema.

4.1.4 Modelado de software de forma visual

Un modelo es una simplificación de la realidad que describe completamente un sistema desde una perspectiva particular. Se construyen modelos con el fin de entender mejor el sistema que se está analizando. El modelado es importante porque ayuda al equipo de desarrollo a visualizar, especificar, construir y documentar la estructura y comportamiento de la arquitectura de un sistema.

Modelar el software ofrece soluciones a las causas de origen de problemas en el desarrollo de sistemas de manera visual.

- Los modelos capturan el diseño de sistemas de software evitando ambigüedades

- Se detectan arquitecturas no modulares e inflexibles
- Se pueden omitir detalles cuando es necesario
- Los diseños revelan inconsistencias de manera más temprana

4.1.5 Verificar continuamente la calidad del software

Los problemas de software son muchas veces más caros de encontrar y reparar después del proceso de desarrollo. Por ello, es importante asegurar continuamente la calidad de un sistema respecto a su funcionalidad, confiabilidad y desempeño.

La verificación de dicha funcionalidad involucra la creación de pruebas para cada escenario clave, cada uno de los cuales representa algún aspecto del comportamiento deseado para el sistema. Verificar la calidad del software ofrece diversas soluciones a las causas origen de problemas en el desarrollo de sistemas informáticos:

4.1.6 Control de cambios del software

Uno de los retos principales al desarrollar software es trabajar con múltiples desarrolladores organizados en equipos, posiblemente en diferentes sitios, trabajando juntos en múltiples iteraciones, versiones, productos y plataformas. En ausencia de un control disciplinado el desarrollo del sistema puede degenerar rápidamente.

Coordinar las actividades y herramientas de los desarrolladores y equipos de trabajo requiere establecer flujos de trabajo repetibles para administrar los cambios y otros elementos de desarrollo. Esta coordinación permite manejar de mejor forma los recursos con base en los riesgos y prioridades del proyecto y administrar activamente los cambios en cada iteración.

4.2 El Proceso Racional Unificado (RUP)

El Proceso Racional Unificado, es una metodología de desarrollo de software enfocado al aseguramiento de la producción de sistemas de calidad de un modo repetible y predecible.

De acuerdo con RUP un proceso de desarrollo de software tiene cuatro tareas:

- Proporcionar guías a las actividades de los equipos de trabajo
- Especificar qué partes deben ser desarrolladas y cuándo
- Dirigir las tareas de desarrolladores individuales y del equipo entero
- Ofrecer criterios para monitorear y medir los productos y actividades del proyecto

Un proceso bien definido permite y alienta las prácticas descritas anteriormente. Cuando se formalizan estas prácticas en un proceso, el equipo de desarrollo puede crecer en cuanto a experiencia colectiva de proyectos exitosos.

RUP es un proceso de ingeniería de software para:

- Asignación de tareas y responsabilidades
- Producción de software de calidad en tiempo y presupuestos predecibles
- Capacidad de adaptarse a la organización

4.2.1 Elementos básicos que componen a RUP

- Un trabajador

- Define el comportamiento y las responsabilidades de un individuo o grupo trabajando en equipo. Las responsabilidades de cada trabajador es hacer una serie de actividades que son expresadas en relación con documentos
- Son ejemplos de trabajadores: analista de sistemas, diseñadores, diseñador de prueba, entre otros

- Actividades
  - Es una unidad de trabajo que se asigna a un trabajador
  - Una actividad lleva entre un par de horas y un par de días, involucra un trabajador y un número pequeño de documentos
  - Las actividades se consideran en la planificación y evaluación del progreso del proyecto
  - Los siguientes son ejemplos de actividades (Plan del ciclo de trabajo iterativo, encontrar casos de uso y actores, revisar el diseño, ejecutar la prueba de velocidad en el sistema)

  Las actividades se divides en 3 pasos.

  - *Pensar*. El trabajador entiende la naturaleza de la tarea, examina los documentos y formula el resultado
  - *Ejecutar*. El trabajador crea o actualiza sus documentos
  - *Revisar*. El trabajador inspecciona el resultado

La Tabla 4.1 muestra la relación entre trabajador y actividad:

Recurso	Trabajador	Actividad
Nadia	Diseñador	Diseño de objetos
Pablo	Autor de casos de uso	Detallar un caso de uso
Lidia	Diseñador de casos de uso	Diseña un caso de uso
Roberto	Revisor del diseño	Revisar el diseño
Arturo	Arquitecto	Análisis de arquitectura Diseño de arquitectura

Tabla 4.1 Ejemplo de una relación Trabajador-Actividad.

- Documentos

Son piezas de información producidos, modificados o usados por el proceso. Son los productos tangibles del proyecto. Los documentos son usados por el trabajador como entrada para ejecutar una actividad y son el resultado de otras actividades.

Los documentos de RUP se organizan en 5 formas:

- *Conjunto de documentos de dirección.* Están relacionados con el desarrollo del software y alcance del proyecto. Por ejemplo el plan de desarrollo del software, los casos de negocio, análisis del proceso actual y documentos de operación.

- *Conjunto de documentos de requerimientos.* Es la definición del software que será desarrollado. Por ejemplo los requerimientos que los usuarios necesitan o el modelo de caso de uso.

- *Conjunto de documentos de diseño*. Contiene la descripción de un sistema que será desarrollado. Por ejemplo el modelo de diseño, la descripción de la arquitectura y el modelo de pruebas.

- *Conjunto de documentos de puesta en práctica*. Incluye código fuente, programas ejecutables, la asociación de datos o archivos que se necesiten.

- *Conjunto de documentos para material de apoyo.* Contiene toda la información relacionada con el material de instalación, manual de usuario, material de capacitación.

4.2.2 Desarrollo iterativo

El desarrollo iterativo es un método de construcción de sistemas cuyo ciclo de vida está compuesto por un conjunto de iteraciones, las cuales tienen como objetivo entregar versiones del software. Cada iteración se considera un subsistema que genera productos de software y no sólo documentación, permitiendo al usuario tener puntos de verificación y control más rápidos e induciendo un proceso continuo de pruebas y de integración desde las primeras iteraciones.

Para entender lo que es el desarrollo iterativo, primero tenemos que definir los términos desarrollo e iteración. Comúnmente entendemos por desarrollo una acción de crear, aumentar, acrecentar, perfeccionar o mejorar algo. Asimismo, iteración es llevar a cabo un proceso fragmentado en una serie de pasos repetitivos.

De esta manera definimos al desarrollo iterativo como: el desarrollo de un sistema mediante un proceso fragmentado en una serie de pasos llamados iteraciones, cada uno de los cuales proporciona una aproximación mejor que la anterior del sistema deseado.

El resultado de cada paso tiene que ser un sistema que se pueda ejecutar, comprobar y depurar. El desarrollo iterativo está ligado fuertemente con el concepto de desarrollo incremental. En el desarrollo iterativo e incremental, cada iteración agrega una funcionalidad adicional a la iteración anterior. El orden en que se añade la funcionalidad se selecciona para equilibrar el tamaño de las iteraciones y para atacar desde el principio las fuentes potenciales de riesgo, antes de que se incremente demasiado el costo de resolver los errores.

4.2.2.1 La relación de las etapas de modelado y las fases de desarrollo

Las etapas de modelado encajan dentro de un proceso de desarrollo iterativo que tiene las fases de inicio, elaboración, construcción y transición. Las fases son secuenciales dentro de una entrega de una aplicación, pero cada fase incluye una o más iteraciones. Dentro de una iteración, los elementos individuales del modelo se mueven a lo largo del camino desde el análisis hasta la entrega, cada uno a su propio ritmo.

4.2.2.2 Fase de inicio

También conocida como fase de concepción, tiene como propósito definir y acordar el alcance del proyecto con los usuarios, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.

A partir del modelo de casos de uso y de la lista de riesgos, se puede determinar qué casos de uso deben implementarse primero para atacar los riesgos de mayor exposición. Con base en la información previa se realiza el proceso de planificación general y un plan de trabajo detallado para la siguiente fase, así como el plan para la siguiente iteración. Se debe establecer una relación clara y directa entre los casos de uso y los casos de prueba para facilitar que el proceso de aseguramiento de la calidad del software se ejecute adecuadamente. El plan de pruebas debe planearse en esta fase, ejecutarse desde la primera iteración de la fase de elaboración y refinarse sucesivamente durante el ciclo de vida del proyecto.

Algunas consideraciones importantes son:

- Se establece la oportunidad y alcance el proyecto
- Se identifican todas las entidades externas con las que se trata (actores) y se define la interacción a un alto nivel de abstracción:
    - Identificar todos los casos de uso

    - Describir algunos en detalle
- La oportunidad del negocio incluye:
    - Criterios de éxito
    - Identificación de riesgos
    - Estimación de recursos necesarios
    - Plan de las fases incluyendo hitos

Resultado a obtener en la fase de inicio:

- Un documento de visión general:
    - Requerimientos generales del proyecto
    - Características principales
    - Restricciones
- Modelo inicial de casos de uso (10% a 20 % listos).
- Glosario de términos utilizados
- Caso de uso:
    - Contexto
    - Criterios de éxito
- Identificación inicial de riesgos
- Plan de proyecto
- Uno o más prototipos

4.2.2.3 Fase de elaboración

Los casos de uso seleccionados para desarrollarse en esta fase permiten definir la arquitectura del sistema, se realiza la especificación de los casos de uso seleccionados y el primer análisis del alcance de la iteración, se diseña la solución preliminar y comienza la ejecución del plan de manejo de riesgos, según las prioridades definidas en él. Al final de la fase se determina la viabilidad de continuar el proyecto y si se decide proseguir, dado que la mayor parte de los riesgos han sido mitigados, se escriben los planes de trabajo de las etapas de construcción y transición, se detalla el plan de trabajo de la primera iteración de la fase de construcción.

Dentro de esta fase se realiza un proceso de refinamiento de las estimaciones de tiempos y recursos para las fases de construcción y transición, llamada etapa de producción. En ella se define un plan de mantenimiento, se implementan los casos de uso pendientes y se entrega el producto al cliente, garantizando la capacitación y el soporte adecuados.

Objetivos:
- Analizar el alcance del proyecto
- Establecer una arquitectura base sólida
- Desarrollar un plan de proyecto
- Eliminar los elementos de mayor riesgo para el desarrollo exitoso del proyecto

Algunas consideraciones esenciales son:
- Las decisiones de arquitectura requieren una visión global del sistema
- Es la parte más crítica del proceso:
    - Se ha hecho gran parte del modelado
    - Se puede decidir si vale la pena seguir adelante
- A partir de aquí la arquitectura, los requerimientos y los planes de desarrollo son estables
- Ya hay menos riesgos y se puede planificar el resto del proyecto con menor incertidumbre
- Se construye una arquitectura ejecutable que contemple:
    - Los casos de uso críticos
    - Los riesgos identificados

Resultados a obtener en esta fase:

- Modelo de casos de uso (80% completo) con descripciones detalladas
- Otros requerimientos no funcionales o no asociados a casos de uso
- Descripción de la Arquitectura del Software
- Un prototipo ejecutable de la arquitectura
- Una lista revisada de riesgos y del caso de uso
- Plan de desarrollo para el resto del proyecto
- Un manual de usuario preliminar

Condiciones de éxito de la fase de elaboración:

- ¿Es estable la visión del producto?
- ¿Es estable la arquitectura?
- ¿Los riesgos han sido abordados y resueltos?
- ¿Es el plan del proyecto algo realista?
- ¿Están de acuerdo con el plan todas las personas involucradas?

4.2.2.4 Fase de construcción

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar el cambio de los elementos construidos, ejecutar el plan de administración de recursos y mejoras en el proceso de desarrollo para el proyecto.

Algunas consideraciones esenciales son:

- En esta fase todos los componentes restantes se desarrollan e incorporan al producto
- Todo es probado en profundidad
- El énfasis está en la producción eficiente y no en la creación intelectual
- Puede hacerse construcción en paralelo, pero esto exige una planificación detallada y una arquitectura muy estable

Resultado de esta fase es:

- El producto de software integrado y corriendo en la plataforma adecuada
- Manuales de usuario
- Una descripción de la versión actual
- Se obtiene una versión beta[1] que debe decidirse si puede ponerse en ejecución sin mayores riesgos
- Condiciones de éxito:
    - ¿El producto está maduro y estable para instalarlo en el ambiente del cliente?
    - ¿Están los interesados listos para recibirlo?

4.2.2.5 Fase de transición

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el sistema cumpla con las especificaciones entregadas por las personas involucradas en el proyecto al inicio del mismo.

[1] Versión beta se refiere a un sistema completo, ejecutable y estable que se pone a disposición de los usuarios para su validación integral antes de ser entregada la versión final.

- El objetivo es entregar el software desarrollado al usuario
- Una vez instalado surgirán nuevos elementos que implicarán nuevos desarrollos (ciclos)
- Incluye:
    - Pruebas a la versión beta para validar el producto con las expectativas del cliente
    - Ejecución paralela con sistemas antiguos
    - Conversión de datos
    - Capacitación de usuarios
    - Distribuir el producto

Resultado de esta fase es:

- Obtener autosuficiencia de parte de los usuarios
- Concordancia en los logros del producto de parte de las personas involucradas
- Lograr el consenso cuanto antes para liberar el producto al mercado

4.2.3 Beneficios de un proceso iterativo

Comparando el proceso iterativo contra el proceso en cascada, el primero tiene las siguientes ventajas:

- Los riesgos son minimizados de forma más temprana
- Los cambios son más manejables
- Hay un mayor nivel de reutilización
- El equipo de trabajo puede capacitarse durante el ciclo de vida del proyecto
- El sistema es de mayor calidad

El proceso secuencial o de cascada es adecuado para proyectos pequeños que implican pocos riesgos y emplean una tecnología y ambiente bien conocidos, sin embargo no puede ajustarse a proyectos largos o que implican un alto nivel de inexperiencia o riesgo.

El proceso iterativo descompone un ciclo de desarrollo en una sucesión de iteraciones, cada una de las cuales se puede ver como un mini-proceso en cascada e involucra las actividades de requerimientos, diseño, implementación y entrega del sistema.

Para controlar el proyecto y, principalmente, para darle el enfoque adecuado a cada iteración, éstas son agrupadas en una de las cuatro fases antes mencionadas. Las iteraciones hacen énfasis en actividades propias de la fase a la que pertenecen.

La figura 4.3 muestra el equilibrio del esfuerzo durante las fases e iteraciones sucesivas.

Durante el inicio, el foco de interés está principalmente en el análisis, con un esqueleto de los elementos que progresa hacia el diseño y la implementación durante la elaboración.

Durante la construcción y la transición, todos los elementos deben, eventualmente, llegar a su terminación.

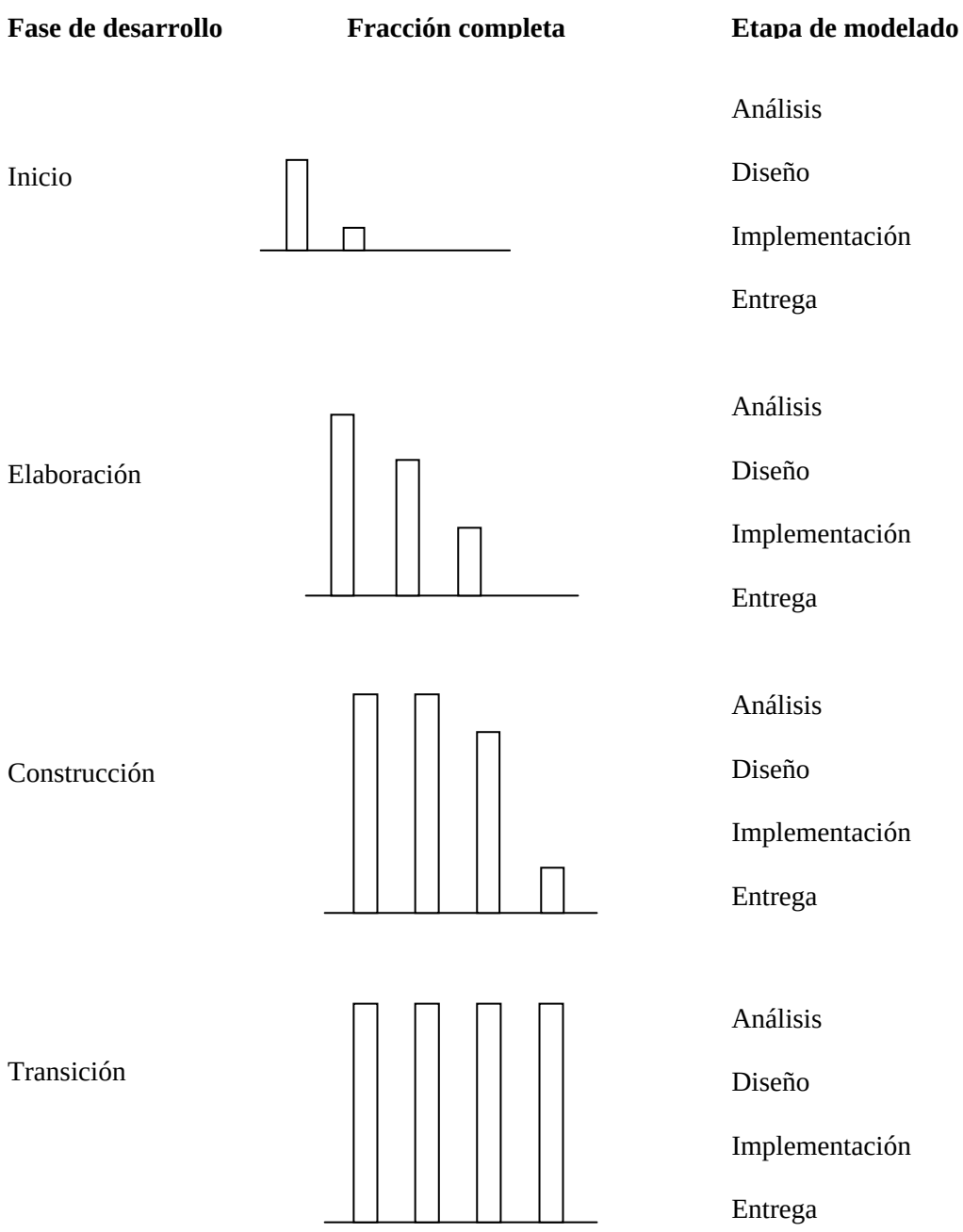


Figura 4.3 Progreso de las etapas de modelado en cada fase de desarrollo

4.3 Flujos de trabajo

Los flujos de trabajo o workflows son una secuencia de actividades divididas en bloques, cada uno de los cuales emplea los resultados del anterior. Existen 9 flujos de trabajo: dirección del proyecto, requerimientos, análisis, diseño, implementación, prueba, configuración y dirección de cambios, ambiente y despliegue.

4.3.1 Dirección del proyecto

Plantea todo lo necesario para llevar a cabo un sistema, desde el planteamiento adecuado de los objetivos, pasando por tener en cuenta los riesgos del sistema, tener bien definido los requerimientos, entender lo que el usuario realmente quiere, y finalmente empezar a realizar el planteamiento iterativo para crear un sistema exitoso.

En la figura 4.4 se muestra el flujo de trabajo de dirección del proyecto.

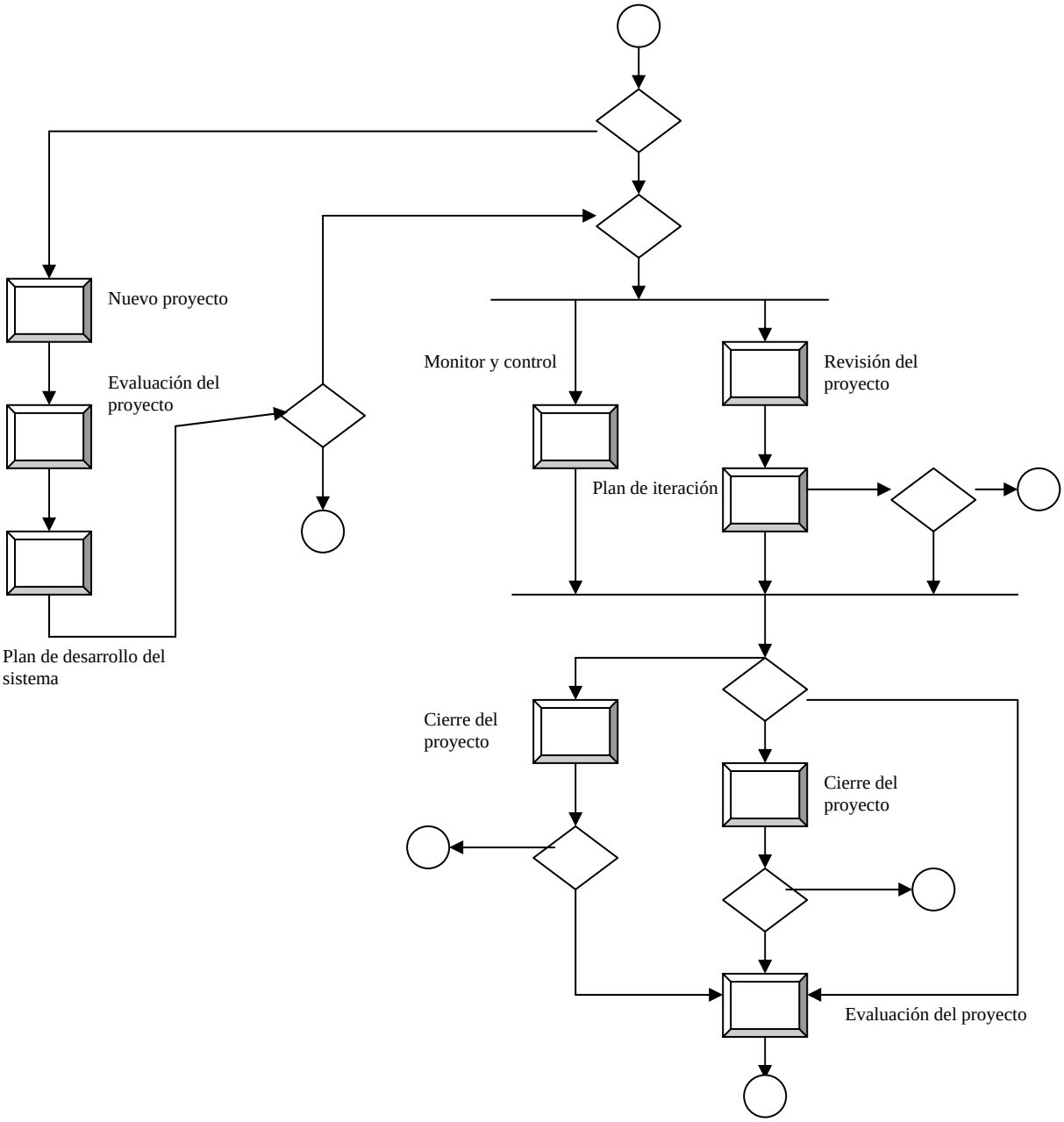


Figura 4.4 Flujo de trabajo de dirección del proyecto

- *Nuevo proyecto*.- Identificación de necesidades y riesgos del sistema.
- *Evaluación del proyecto*.- Se evalúan las necesidades y riesgos del sistema.

- *Plan de desarrollo del sistema*.- Se desarrolla el plan de medición y prevención de riesgos, aprobación del producto, resolución del sistema y se define la organización del equipo de trabajo.

- *Monitor y control del proyecto*.- Se realiza un programa de actividades y se asigna trabajo a cada integrante del proyecto.

- *Plan de iteración*.- Se prepara el término de cada fase en las actividades realizadas

- *Control de iteraciones*.- Se revisa todo el trabajo que se lleva al día para verificar la calidad del sistema.

- *Cierre de proyecto*.- Se implanta el sistema y se realizan pruebas necesarias para el funcionamiento.

- *Evaluación*.- Se instala el sistema para verificar el buen funcionamiento.

4.3.2 Requerimientos

El objetivo principal es construir el modelo de caso de uso, el cual captura los requerimientos funcionales del sistema que está siendo modelado. Este modelo permite a los usuarios estar de acuerdo sobre las capacidades y las condiciones del sistema. El modelo de caso de uso es el centro de la arquitectura del sistema.

El flujo de trabajo de requerimientos pasa por las cuatro fases del proceso unificado:

- Lo fundamental se encuentra en la etapa de inicio, el conocimiento de los requerimientos críticos del sistema.

- La mayoría de los modelos de caso de uso son construidos durante la etapa de elaboración.

- La construcción generalmente incluye elaborar el resto del modelo de caso de uso, con los requerimientos que no fueron detectados durante la etapa de inicio y elaboración.

- El modelo de caso se refina durante la etapa de transición.

En la figura 4.5 se muestra el flujo de trabajo de requerimientos.

- *Analizar el problema*.- Identificación de los límites y restricciones del sistema.

- *Necesidades del usuario*.- Tener muy claro lo que el usuario necesite.

- *Define el sistema*.- Definición de los límites del sistema

- *Dirección del alcance del sistema*.- Tener a la mano la información que será útil para el sistema.

- *Cambio de requerimientos*.- Se verifican al día los requerimientos por si hay algún cambio.

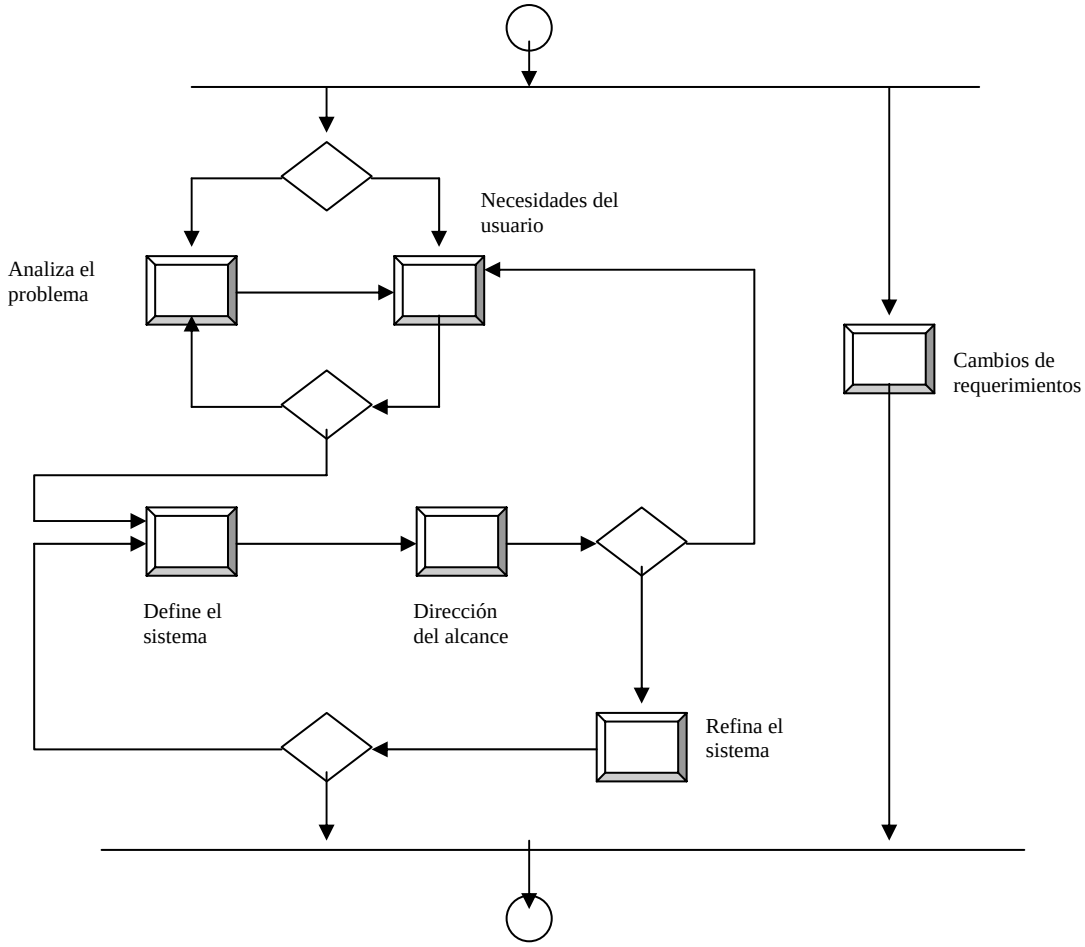


Figura 4.5 Flujo de trabajo de requerimientos

4.3.3 Análisis

Se construye un modelo de análisis que ayuda a los desarrolladores a refinar y estructurar los requerimientos definidos en el modelo de caso de uso. El modelo de análisis interviene en el diseño de pantallas y procesos en la arquitectura del sistema.

El flujo de trabajo de análisis pasa por las cuatro fases del proceso unificado:

- El modelo de análisis comienza a ser desarrollado durante la etapa de inicio
- Una gran mayoría de los modelos de análisis son construidos durante la etapa de elaboración, ya que en esa etapa los requerimientos funcionales son analizados, redefinidos y estructurados
- La etapa de construcción incluye la creación del resto del modelo de análisis, para dirigirse a los requerimientos no analizados que aparecieron después de la etapa de elaboración
- El modelo de análisis se refina durante la etapa de transición

4.3.4 Diseño

Se construye el modelo de diseño que describe las realizaciones físicas del caso de uso, y el contenido del modelo de análisis. El modelo de diseño sirve como una abstracción del modelo de implementación. El flujo de trabajo de diseño se enfoca al modelo de despliegue, el cual define la organización física del sistema. Este modelo es muy relevante para la vista de los procesos.

El flujo de trabajo de diseño pasa por las cuatro fases del proceso unificado:

- El modelo de diseño se comienza a desarrollar durante la etapa de inicio, como parte de la labor de conocer los requerimientos
- El caso de uso es significativo para la arquitectura desarrollada dentro del modelo de diseño en la etapa de elaboración
- La etapa de construcción incluye la mayor parte del modelo de diseño

El flujo de trabajo de análisis y diseño se unen en un solo diagrama como se muestra en la figura 4.6:

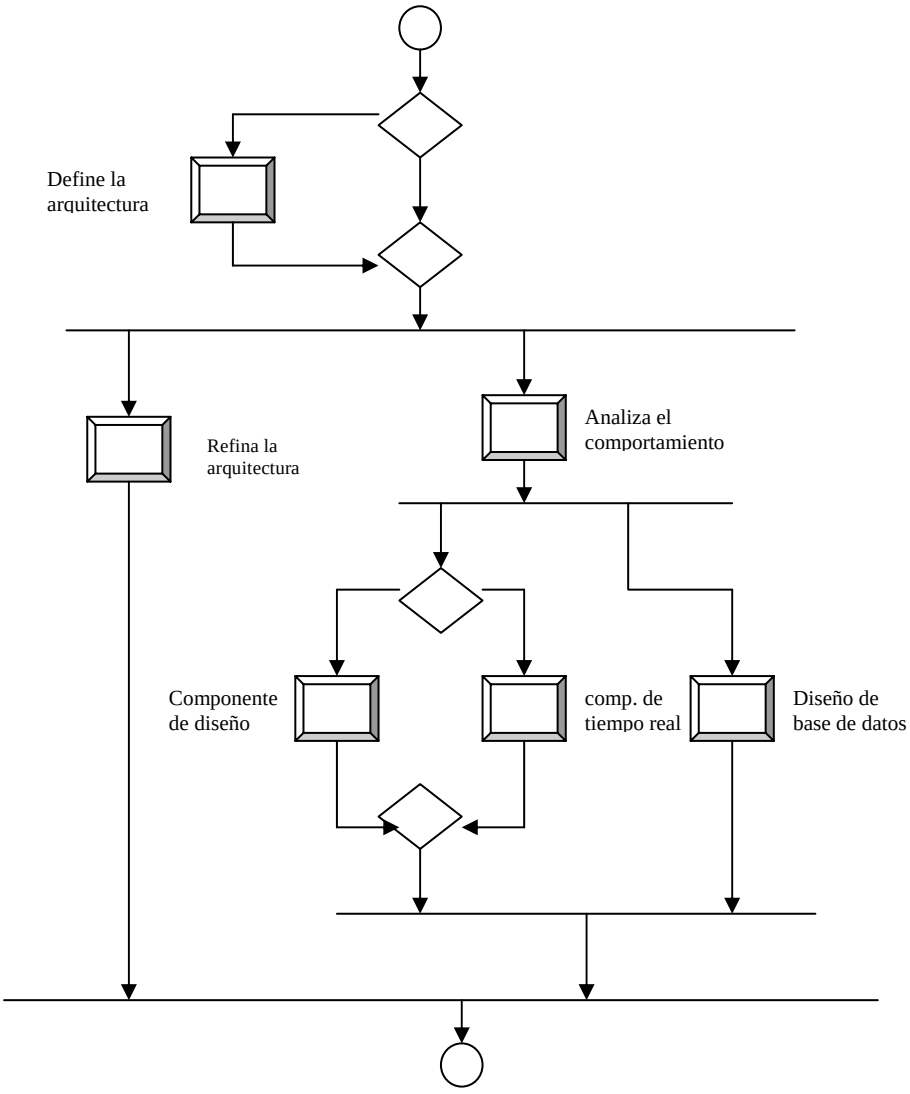


Figura 4.6 Flujo de trabajo de análisis y diseño

- *Define una arquitectura*. Se crea un bosquejo inicial sobre el sistema, identificación de los casos de uso en el análisis.

- *Refina la arquitectura*. Se detalla las actividades identificadas en el diseño del sistema e identifica los elementos como son tiempos de ejecución, distribución del proyecto etc.

- *Analiza el comportamiento*. Verificar que los requerimientos se cumplan en el sistema.

- *Componentes de diseño*. El programador examina el diseño y define los elementos.

- *Componentes de tiempo real del sistema*. Se definen los tiempos de las actividades a realizar.

- *Diseño de la base de datos*. Se diseña e implementa la base de datos a utilizar, esto solo es si el sistema involucra cantidades grandes de información.

4.3.5 Implementación

La construcción del modelo de implementación, el cual describe cómo los elementos del modelo de diseño son armados en componentes de software, tales como son los archivos de código fuente o librerías de enlace dinámicos (DLL). El modelo de implementación es el elemento clave de la arquitectura del sistema.

El flujo de trabajo de implementación pasa por las cuatro fases del proceso unificado:

- Durante la etapa de inicio, el modelo de implementación, existe en una forma muy general como un prototipo ejecutable

- En la etapa de elaboración del modelo de implementación se dirige al caso de uso

- La etapa de construcción se enfoca en la construcción de la gran mayoría de los modelos de implementación

- El modelo de implementación se refina durante la etapa de transición

En la figura 4.7 se muestran de manera general los pasos a seguir en el flujo de trabajo de implementación.

- *Estructura de implementación*.- Debe tomar en cuenta la manera y el momento en el cual la lista completa de entregables será creada, así como asegurarse de que el usuario final cuenta con la información necesaria para hacer uso del nuevo software.

- *Plan de integración*.- Se genera un plan general de integración de los diversos componentes del sistema.

- *Implementación de componentes*.- Cada subsistema es probado en el ambiente real en el que se ejecutará, de modo que se eliminen posibles errores y queden listos para la integración general del sistema.

- *Integración de subsistemas*.- El primer paso de la integración general es formar subsistemas de componentes relacionados para probar su funcionamiento como módulos independientes.

- *Integración del sistema*.- Finalmente se integran todos los subsistemas en un solo conjunto.

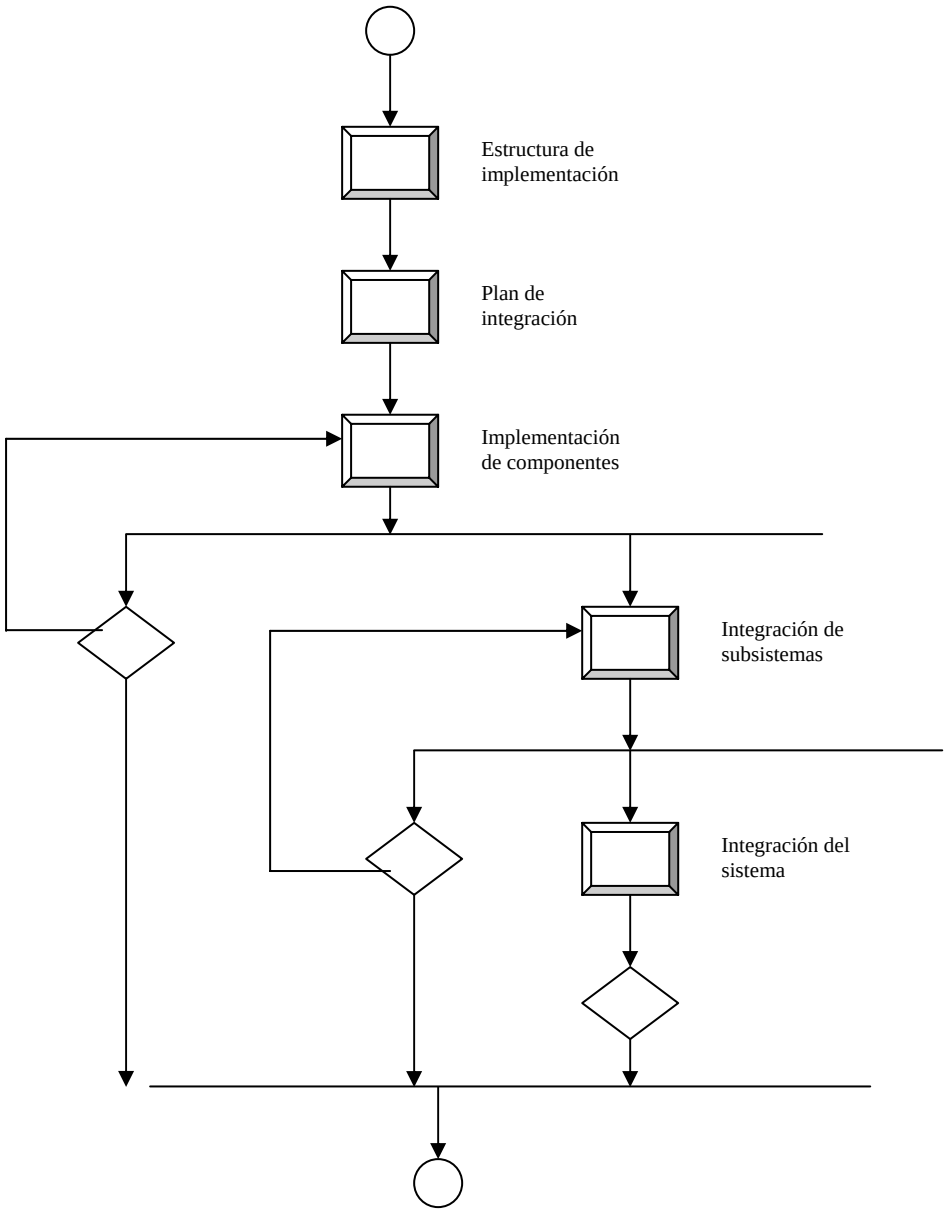


Figura 4.7 Flujo de trabajo de implementación

4.3.6 Prueba

El flujo de trabajo de prueba tiene como propósito trazar una meta de calidad del sistema, esto no solo involucra el sistema final sino también empieza con la valoración del comienzo del mismo, por medio de la verificación de componentes y la verificación de requerimientos implementados. También describe cómo la integración y los sistemas de prueba serán trabajados en componentes ejecutables desde el modelo de implementación y describe como el equipo de trabajo debería ejecutar estas pruebas.

El flujo de trabajo de prueba pasa por cuatro etapas del proceso unificado:

- Durante la etapa de inicio, el modelo de prueba se enfoca al prototipo ejecutable, si este existe
- En la etapa de elaboración el modelo de prueba se dirige hacia el caso de uso significativo

- La etapa de construcción se enfoca en el desarrollo de la gran mayoría de los modelos de prueba y en la integración y prueba del sistema

- El modelo de prueba es bien definido durante la etapa de transición, ya que realizar pruebas de modo continuo ayuda al equipo de trabajo a descubrir defectos

En la figura 4.8 se ilustra el flujo de trabajo de prueba.

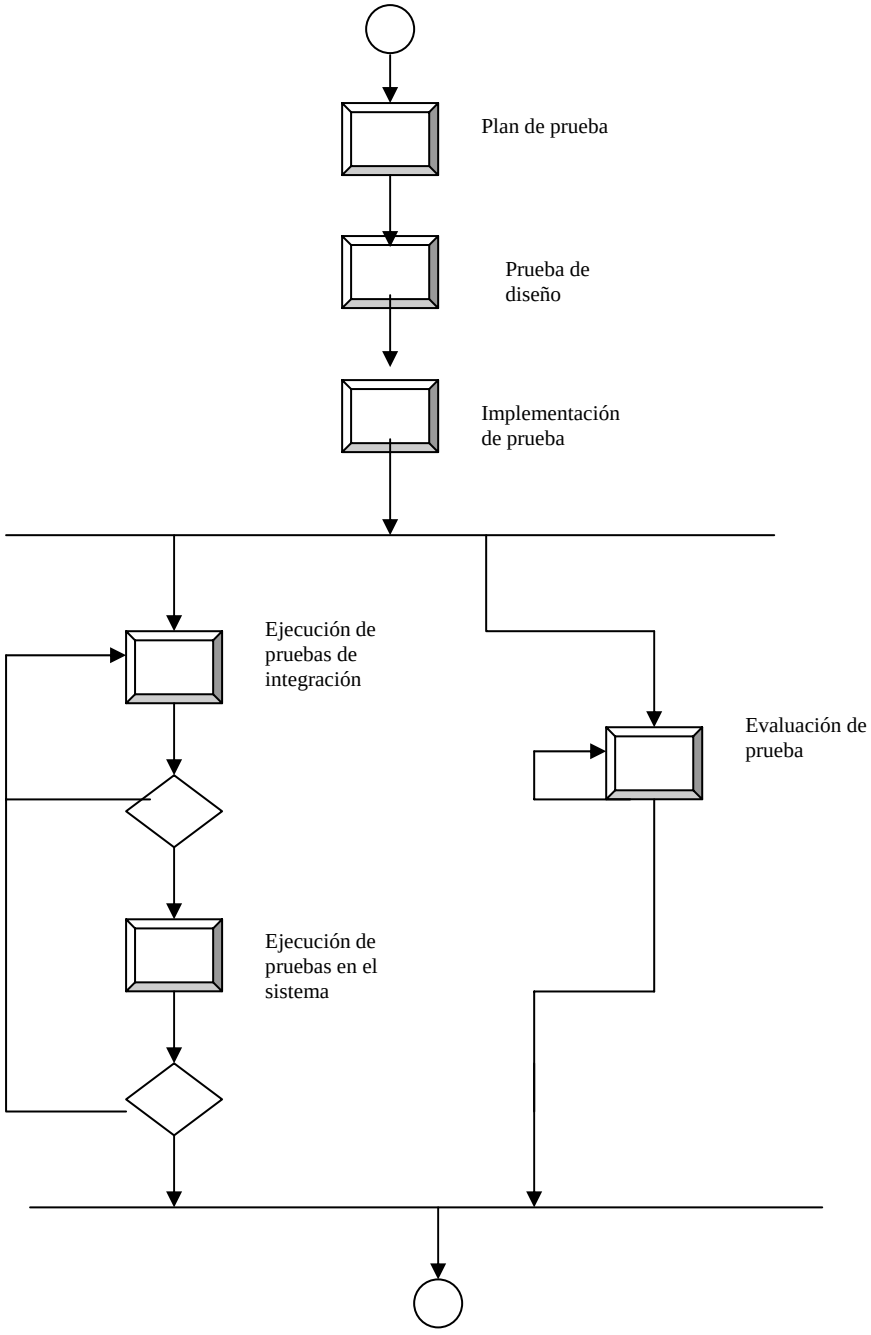


Figura 4.8 Flujo de trabajo de prueba

- *Plan de prueba*.- El propósito es identificar y describir todo tipo de pruebas que serán implementadas y ejecutadas.
- *Prueba de diseño*.- La prueba de diseño está hecha para asegurar que el software cubre con todos los requerimientos y está bien estructurado el análisis del objetivo.

- *Implementación de prueba*.- Se implementan los procedimientos de prueba que fueron definidos en pruebas de diseño.

- *Ejecución de pruebas de integración*.- Asegurarse de que durante el desarrollo el equipo de trabajo siga las tareas asignadas y que exista un archivo ejecutable.

- *Ejecución de pruebas en el sistema*.- Asegurar que el sistema funcione correctamente con lo planeado.

- *Evaluación de prueba*.- Verificar la calidad del sistema y verificar si el sistema cumple con los requerimientos.

4.3.7 Configuración y dirección de cambio

El propósito de la configuración y dirección de cambio es estar al día y que el proyecto evolucione durante el desarrollo iterativo. En este flujo de trabajo se conectan otros dos, que son la configuración y dirección de cambio.

En la figura 4.9 se muestra el flujo de trabajo.

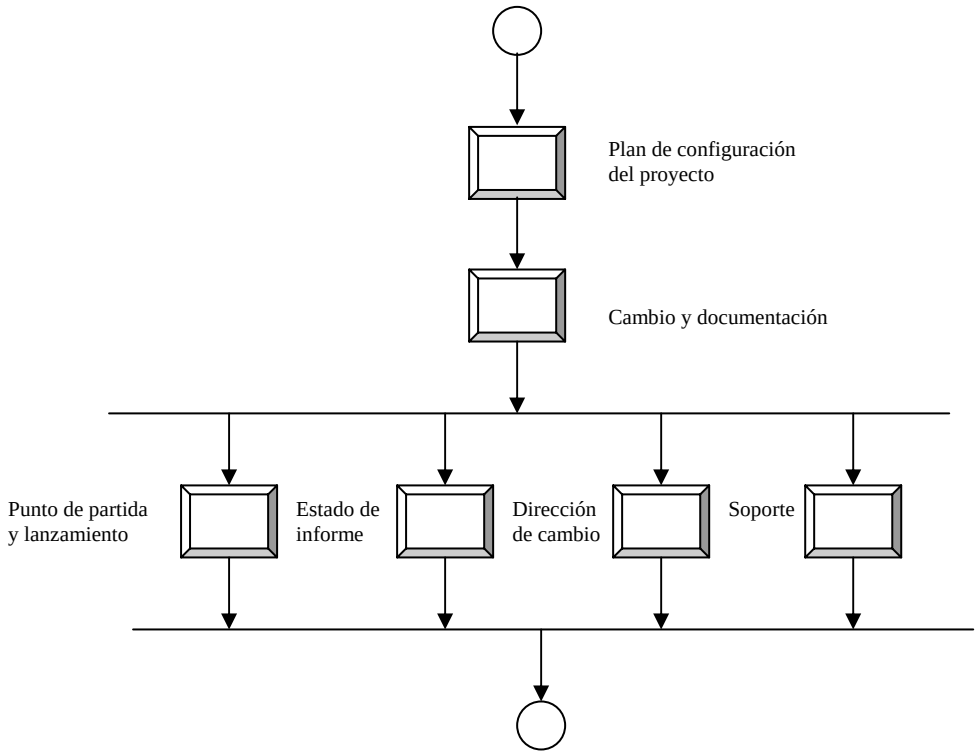


Figura 4.9 Configuración y dirección de cambio

- *Plan de configuración del proyecto y el control de cambiar*.- El plan de configuración del proyecto describe todos los procedimientos a seguir, protege la información generada durante el desarrollo y las actividades a realizar durante todo el proyecto. El control de cambio es asegurar que todos los cambios planeados sean hechos y de que se informe a los integrantes del grupo de trabajo.

- *Cambio y documentación*.- Se realizan cambios en la programación y se tienen que documentar para que posteriormente otro desarrollador pueda mejorar el sistema.

- *Punto de partida y lanzamientos*.- El punto de partida es una descripción de todas las versiones que se realizan por si hay algún defecto. El lanzamiento es el momento en el cual se entrega el sistema.

- *Estado de informe*.- Se documenta todo lo que se realiza, para que en caso de presentarse algún problema pueda ser localizado fácilmente.

- *Dirección de cambio*.- Se realiza la documentación para llevar un control de los cambios hechos y asegurarse que se hagan.

- *Soporte*.- Mantenimiento constante al sistema.

4.3.8 Ambiente

El propósito del flujo de trabajo de ambiente es suministrar soporte al desarrollo en los procesos y seleccionar y adquirir las herramientas de trabajo para respaldar el desarrollo del sistema. El soporte incluye mejoras en la infraestructura de tecnología de información, administración del proyecto y realización de copias de seguridad del sistema. En la figura 4.10 se muestra el flujo de trabajo de ambiente que permite ver la organización del desarrollo en curso.

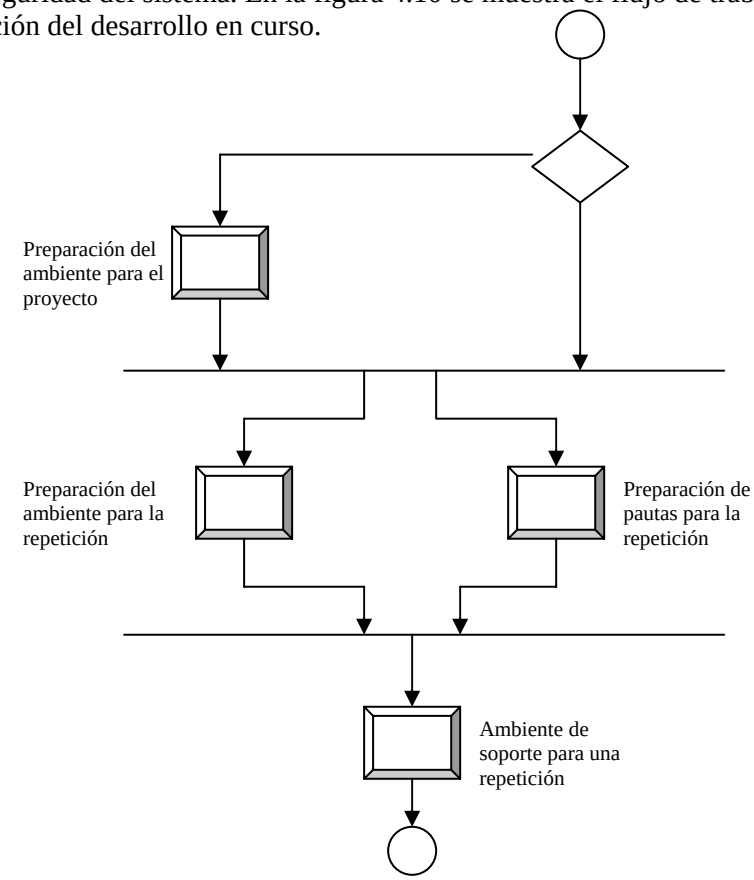


Figura 4.10 Flujo de trabajo de ambiente

- *Preparación del ambiente para el proyecto*.- La actividad principal es mantener un desarrollo organizado, producir listas de herramientas para desarrollar el sistema y especificar plantillas del proyecto.

- *Preparación del ambiente para la repetición*.- Tener bien definidas las herramientas a utilizar, realizar las pruebas necesarias al sistema para ser instalado.

- *Preparación de pautas para la repetición*.- Analizar los problemas y los defectos en la repetición previa para realizar pausas e identificar los errores del sistema y así tener un buen diseño.

- *Ambiente de soporte en la repetición*.- Los programadores necesitan trabajar en el soporte de uso de herramientas y procesos durante la repetición.

4.3.9 Despliegue

La actividad que realiza este flujo de trabajo es alcanzar la fase de transición donde se termina el desarrollo del software y que éste trabaje correctamente en los equipos, en su ambiente operacional, para finalmente entregarlo al usuario. Lo anterior dará una percepción del funcionamiento real del sistema donde se pueda realizar ajustes o dar por terminado el sistema.

En la figura 4.11 se muestran las actividades que realiza el flujo de trabajo de despliegue que es necesario para realizar el software y poderlo entregar al usuario.

- *Plan de despliegue*.- El desarrollador debe asegurarse de tener toda la información necesaria para realizar el sistema y hacer prueba necesarias antes de la entrega del software.

- *Desarrollo de material de soporte*.- Se debe realizar la documentación necesaria para realizar la instalación, operación, uso del sistema y serán entregadas al usuario.

- *Prueba del producto en el sitio*.- Se debe realizar las pruebas necesarias en equipos similares o iguales a donde se va instalar para verificar que el sistema funciona correctamente.

- *Preparación para la entrega de software*.- El propósito del flujo de trabajo es asegurar que el sistema funcione se instale y funcione correctamente.

- *Prueba beta*.- La prueba beta es esencial para asegurar que una vez instalado el sistema cumpla con las expectativas del cliente.

- *Pruebas definitivas*.- El sistema debe instalarse para verificar que funciona correctamente y que el usuario se familiarice y acepte el sistema.

- *Entrega de software*.- Entrega del archivo ejecutable al cliente.

- *Soporte*.- Se le entrega al usuario una dirección de internet para dar soporte al sistema.

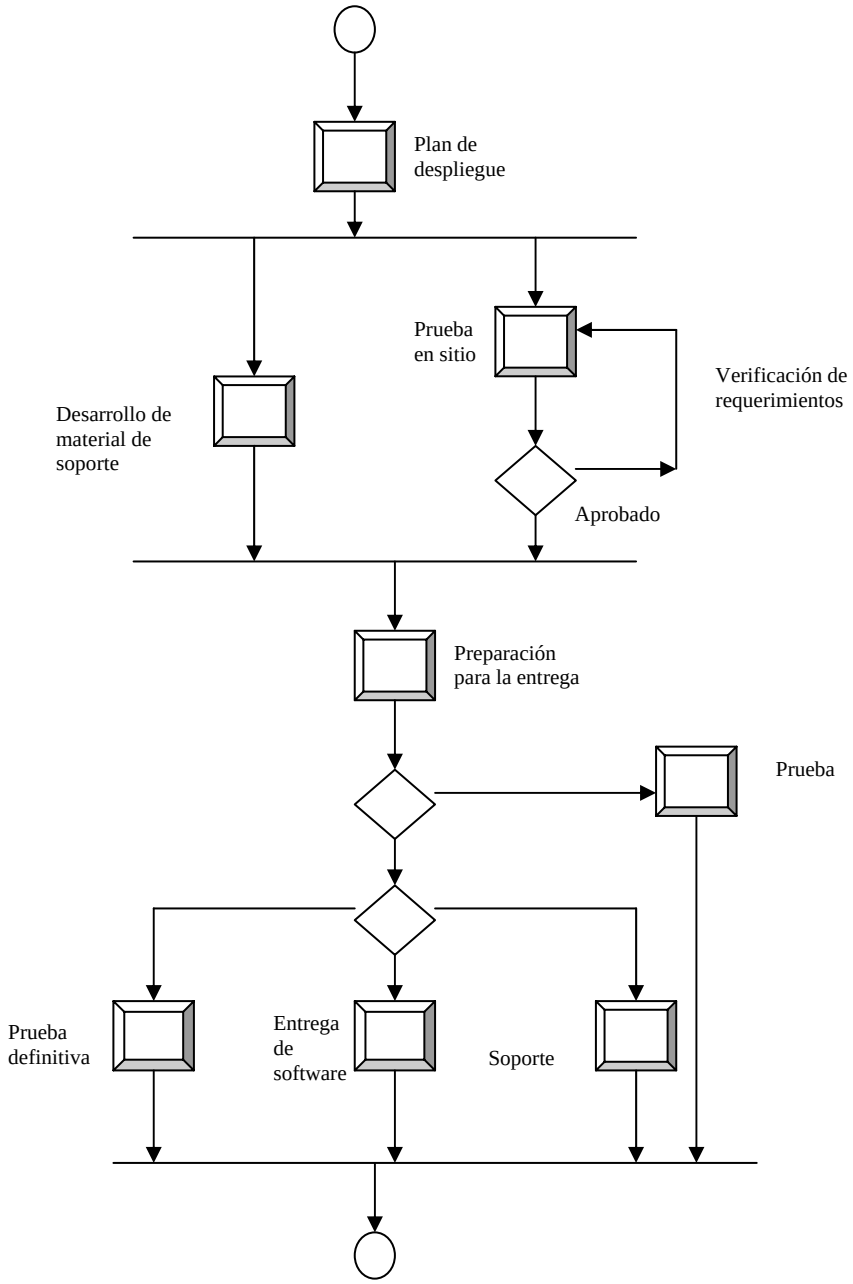


Figura 4.11 Flujo de trabajo de despliegue

4.4 Planes de iteración típicos

Los planes de iteración son una manera de presentar un proceso, describiéndolo más desde la perspectiva de lo que sucede en una iteración típica. Son lo más cercano a lo que un grupo de flujos de trabajo pudiera manejar. Se pueden ver como una selección de actividades que realmente se ejecutarán durante la iteración y se replican conforme se necesiten.

Daremos tres ejemplos de planes de iteración típicos:

- Una iteración en la fase de inicio para definir la visión del proyecto y el caso de uso
- Una iteración en la fase de elaboración para construir un prototipo de arquitectura
- Una iteración en la fase de construcción para implementar el sistema

4.4.1 Definiendo la visión del proyecto y el caso de uso

El contexto de este primer ejemplo es el ciclo inicial para un nuevo sistema, como se muestra en la figura 4.12

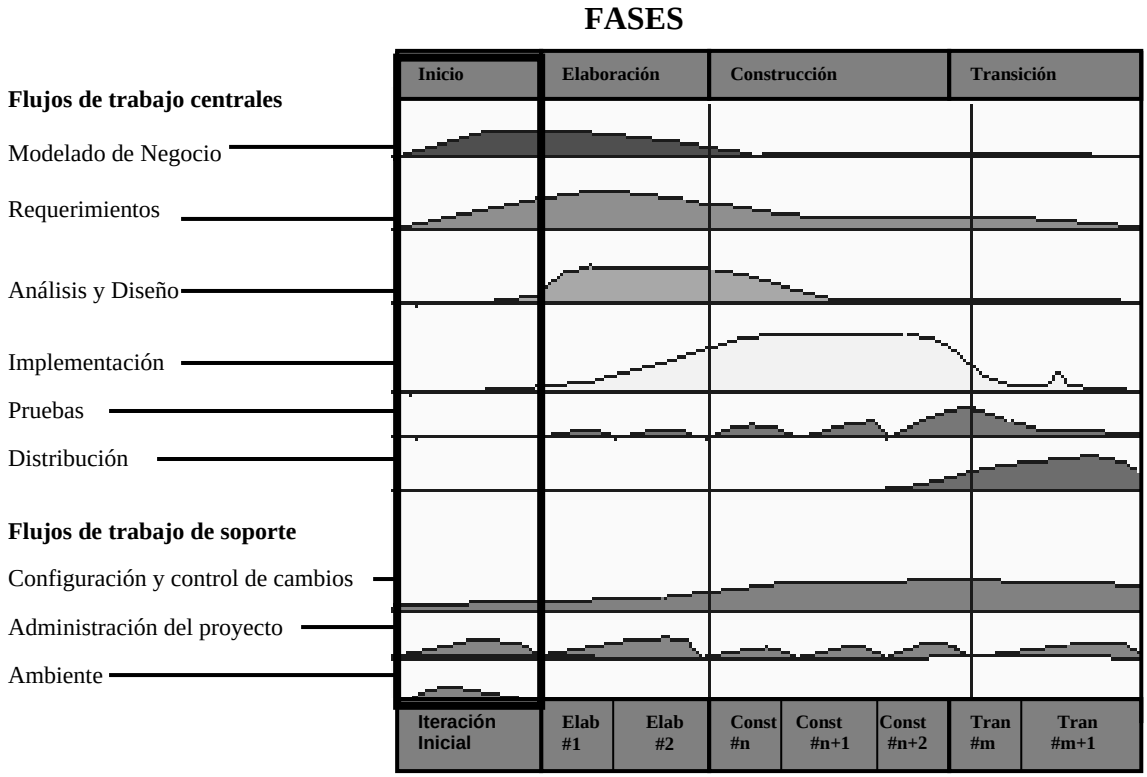


Figura 4.12 Iteración en la fase de inicio

- *Inicio*.- Se define la visión y alcance del sistema, los participantes trabajan con los analistas para definir aspectos primarios como las necesidades y expectativas de los usuarios. Se consideran las restricciones en el desarrollo del sistema, tales como plataformas a ser soportadas y las interfaces externas.

- *Requerimientos*.- Delinean y clarifican la funcionalidad del sistema, los analistas usan varias técnicas tales como elaboración de historias y lluvia de ideas conduciendo sesiones para recolectar opiniones acerca de lo que debe hacer el sistema. El resultado principal de estas sesiones es un documento de las necesidades de los usuarios y posiblemente el delineado del modelo de un caso de uso.

- *Administración del proyecto*.- Se considera la factibilidad y se establece el plan del proyecto, transformando la visión del modelado en términos económicos al considerar los costos de inversión, la estimación de recursos, el ambiente de desarrollo, etc. Se actualiza la lista de riesgos tomando en cuenta riesgos de negocio.

- *Refinación del plan del proyecto*.- En este momento los participantes deberán tener un buen entendimiento de la visión y factibilidad del proyecto. Se establece un orden de prioridad entre características y casos de uso. El administrador del proyecto puede empezar a planearlo en mayor detalle, basando el plan del mismo en los casos de uso priorizados y los riesgos asociados.

El resultado de esta iteración inicial es una primera muestra de la visión del proyecto, caso de uso y enfoque, así como el plan del proyecto.

Iteraciones subsecuentes en el inicio.

Se pueden realizar iteraciones subsecuentes para mejorar el entendimiento del enfoque del proyecto. Esta actividad puede implicar posteriores mejoras al modelo de casos de uso o en los planes, ambiente y riesgos. La necesidad de este trabajo adicional depende de variables tales como la complejidad del sistema y los riesgos asociados.

4.4.2 Construyendo un prototipo arquitectónico

La figura 4.13 ilustra una iteración en la fase de elaboración, en el contexto de que la fase de inicio se ha completado, se tienen delineados los participantes y casos de uso como una muestra inicial del plan de proyecto.

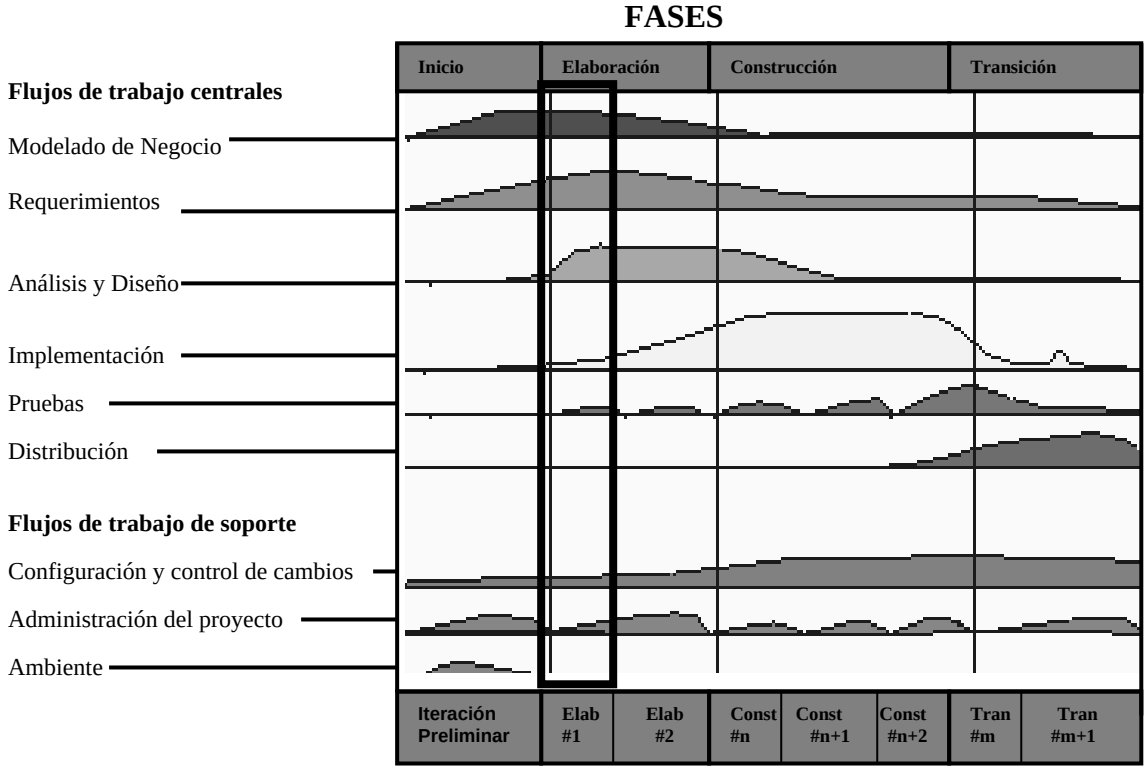


Figura 4.13 Iteración en la fase de elaboración

- *Inicio*.- Se establece el plan de iteraciones, riesgos y objetivos de la arquitectura. Una de las metas de elaboración es establecer una arquitectura robusta y ejecutable.

- *Requerimientos*.- Se decide qué casos de uso y escenarios dirigirán el desarrollo de la arquitectura, determinando en cuáles se enfocará esta iteración. Algunas especificaciones de los casos de uso los describirán a detalle: los de máxima prioridad, más críticos y más complejos serán atendidos en esta primera iteración.

- *Reconsiderar casos de uso y riesgos*.- Se revisan las visiones de los casos de uso tomando en consideración otras descripciones y posiblemente se reestructuran los modelos de casos de uso. La siguiente tarea es seleccionar el conjunto de casos de uso y escenarios que se analizarán, diseñarán e implementarán en la iteración actual. Puede ser necesario reconsiderar la forma de administración de riesgos debido a que pudieron detectarse nuevos riesgos a partir de la información adicional.

- *Requerimientos*.- Usando los casos de uso resueltos, un diseñador de interfaces empieza a plasmarlos en "historias" y construye un prototipo de interfaz de usuario para obtener una realimentación con el usuario.

- *Análisis y diseño*.- Se toma en cuenta todo lo anterior (requerimientos del sistema, el glosario, la visión de los casos de uso y en general el conocimiento del equipo de desarrollo sobre el proyecto) para hacer un borrador de la línea que seguirán los subsistemas, posiblemente en forma de capas. Se identifican mecanismos de análisis que proporcionan soluciones a problemas comunes durante el análisis y todo es documentado.

Se identifican elementos de código importantes para la arquitectura del sistema, considerando la división de subsistemas de bajo nivel y ajustando todo al ambiente de desarrollo.

Finalmente la arquitectura obtenida se revisa.

- *Implementación*.- Un integrador de sistema estudia los casos de uso que se implementarán en esta iteración y define el orden en el cual los subsistemas deberán ser implementados para posteriormente ser integrados a un prototipo.

- *Pruebas*.- Un diseñador de pruebas planea un conjunto de éstas tanto para el sistema como para su integración, seleccionando metas cuantificables para las pruebas. Dichas metas se pueden expresar como la capacidad de ejecutar un caso de uso en un cierto tiempo de respuesta y/o bajo cierta carga de datos (pruebas de estrés)

- *Implementación*.- Un conjunto de implementadores codifican y prueban ciertas unidades de código consideradas importantes. Los integradores del sistema progresivamente van agregando subsistemas al prototipo, cada integración se prueba.

- *Revisar la arquitectura ejecutable*.- Una vez que el sistema completo (que se definió como meta de esta iteración) se ha integrado, el probador de sistema ejecuta pruebas sobre éste. El diseñador de pruebas analiza los resultados para asegurar que las metas se han alcanzado.

El resultado de esta iteración inicial es una primera versión de la arquitectura. Consiste en vistas arquitectónicas bien descritas (la vista de casos de uso, la vista lógica y la vista de implementación) y un prototipo arquitectónico ejecutable.

4.4.3 Implementando el sistema

El contexto del tercer ejemplo es una fase en la cual los requerimientos son estables y gran parte de la funcionalidad se ha implementado e integrado en preparación a más pruebas, el proyecto está acercándose al lanzamiento de su primera versión beta. Como se muestra en la figura 4.14

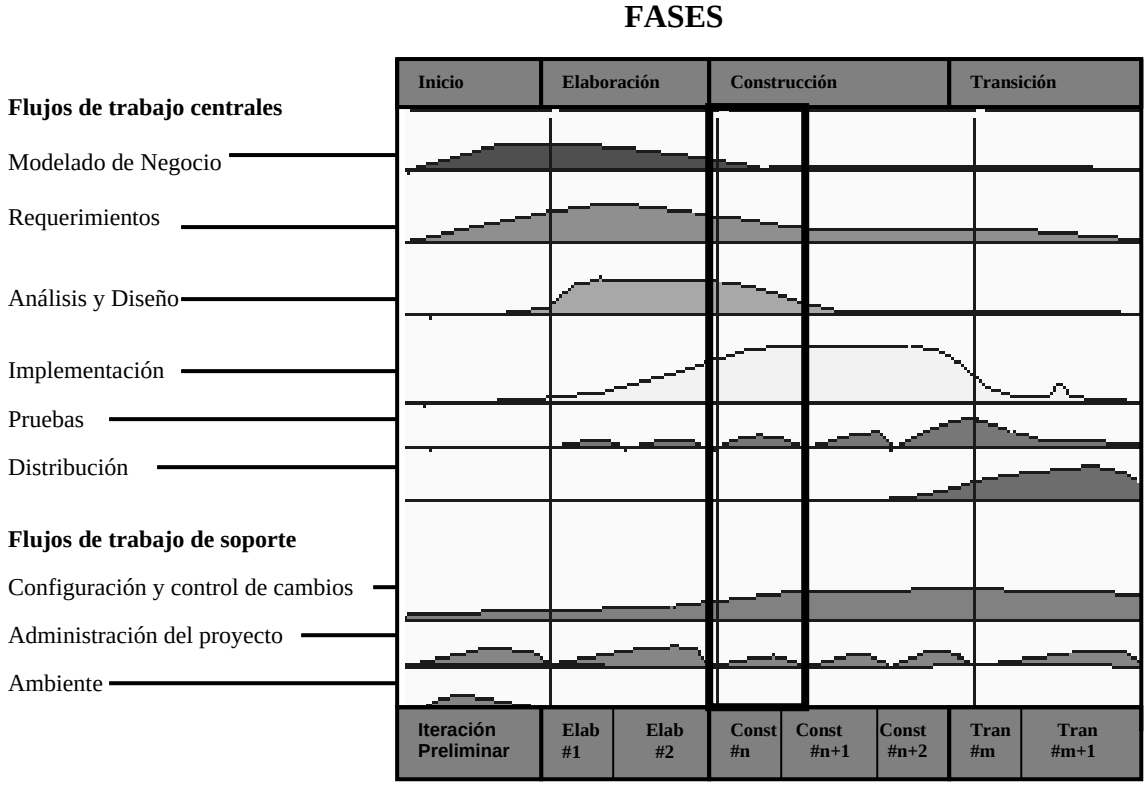


Figura 4.14 Iteración en la fase de construcción

- *Administración del proyecto*.- El administrador del proyecto actualiza el plan de iteraciones basándose en cualquier nueva funcionalidad que se vaya a agregar durante la nueva iteración.

- *Implementación*.- Se planea una integración a nivel sistema, tomando en cuenta el orden en que los subsistemas serán agregados para conformar una configuración funcional y susceptible de ser probada. Esto es realizado por el integrador del sistema y los resultados se agregan al plan de construcción, donde se define la frecuencia de construcciones y se describe cuando ciertos conjuntos construidos serán requeridos para el desarrollo, integración y pruebas subsecuentes.

- *Prueba*.- El diseñador de pruebas planea y se diseña una prueba a nivel sistema, asegurándose que habrá un número adecuado de casos de prueba para verificar los requerimientos susceptibles de prueba. Genera una lista donde cada caso de prueba tiene al menos un procedimiento de prueba asociado. Así mismo, el diseñador de pruebas debe revisar el conjunto acumulado de pruebas en las iteraciones precedentes, el cual puede ser modificado para reutilizarlo en pruebas regresivas para construcciones de iteraciones actuales y futuras.

- *Análisis y diseño*.- Los diseñadores refinan las realizaciones de los casos de uso.

- *Prueba*.- Las pruebas de integración se dirigen a detectar el grado en el que los componentes desarrollados se comunican y trabajan juntos. El diseñador de pruebas sigue el plan que describe la estrategia general de pruebas, recursos necesarios, calendario, el criterio de éxito y terminación.

- *Implementación*.- El implementador diseña pruebas unitarias que muestran lo que cada subsistema hace (prueba de caja negra) y cómo lo hace (prueba de caja blanca). En las pruebas de caja negra el implementador debe asegurarse de que el subsistema, en sus diversos estados, responde bien a las especificaciones y puede aceptar y producir un rango de datos válidos e inválidos. En la estructura de

pruebas de caja blanca el reto para el implementador es asegurar que el diseño se ha transportado a código correctamente y que el sistema puede llevarse a través de todos los árboles de decisión.

- *Implementación*.- Las pruebas de unidades se enfocan en verificar los componentes más pequeños susceptibles de ser probados. Las pruebas de unidades son diseñadas y ejecutadas por el implementador de la unidad.

- *Implementación y pruebas*.- El propósito de la integración de subsistemas es combinarlos en un conjunto ejecutable de unidades que pueden venir de muchos desarrolladores dentro del subsistema.

- *Implementación*.- Los probadores ejecutan planes de prueba desarrollados anteriormente. Si se presentan resultados inesperados los probadores documentan los defectos para decidir cuándo se arreglarán.

- *Implementación*.- Una vez que el subsistema se ha probado lo suficiente y está listo para la integración a nivel de sistema, el implementador libera la versión probada del subsistema desde el área de integración hacia un área donde sea posible verlo y utilizarlo.

- *Implementación*.- El integrador del sistema agrega subsistemas de manera incremental y el total de subsistemas agregados son enviados a los probadores de integración para la prueba general de integración.

- *Pruebas*.- Los probadores de integración ejecutan planes de prueba desarrolladas anteriormente y califican los resultados. Si se encuentran resultados inesperados los probadores documentan los resultados.

- *Pruebas*.- Una vez que el sistema entero ha sido integrado se prueba. El diseñador de pruebas entonces analiza el resultado de las pruebas para asegurar que las metas se han alcanzado.

- *Administración del proyecto*.- El administrador del proyecto compara el costo real, calendario y contenido contra el plan de la iteración, entonces determina si es necesario volver a trabajar en ello, de serlo asigna ese trabajo a futuras iteraciones, actualiza la lista de riesgos, el plan de proyecto y prepara el delineado de un plan de iteración para la siguiente iteración.

El resultado principal de una iteración en la fase de construcción es una mayor funcionalidad, lo cual lleva a un sistema más amigable. Los resultados de la iteración actual se hacen visibles a los desarrolladores para formar la base de desarrollo para iteraciones subsecuentes.

CAPÍTULO 5

Procedimiento Propuesto para el Análisis de Sistemas Informáticos (PAS)



Existen diversas metodologías disponibles para el análisis de sistemas, como se vio en el capítulo anterior, cada una de las cuales presenta un enfoque particular para llevar adecuadamente el desarrollo de un sistema informático.

Sin embargo, el analista de sistemas encuentra que están dirigidas a un limitado tipo de proyectos, de acuerdo con la complejidad que presenten. Lo anterior puede obligarle a tomar una decisión sobre las posibles alternativas: tratar de adaptar una metodología conocida para él, que puede no ser lo suficientemente flexible y no permita realizar las modificaciones necesarias, o estudiar una metodología que se ajuste a las características del proyecto pero que es desconocida para él.

Ante esta problemática, el presente capítulo expone una propuesta de procedimiento que ofrece al analista una alternativa flexible, fácilmente adaptable que le permite analizar adecuadamente proyectos de desarrollo con características diversas sin tener que realizar modificaciones a metodologías rígidas o aprender una nueva cuando no se cuenta con el tiempo necesario para hacerlo.

El procedimiento planteado toma como base principal la metodología RUP, incluyendo el proceso iterativo como unidad básica de trabajo (iteración), las fases como un conjunto de iteraciones que buscan alcanzar un objetivo parcial común y finalmente flujos de trabajo, los cuales indican qué hacer en cada punto de iteración (obtener requerimientos, analizar y diseñar).

Durante la fase de inicio el objetivo principal del procedimiento propuesto es solicitar requerimientos y hacer un análisis general del sistema detectando y reduciendo los riesgos para decidir si el sistema es viable.

En la fase de elaboración el objetivo es terminar de levantar requerimientos y realizar el análisis completo del sistema, para que en fases posteriores el levantamiento de requerimientos y el análisis sea mínimo.

En la fase de construcción el análisis que se realiza está limitado al control de cambios.

El procedimiento propuesto, en adelante PAS (Procedimiento de Análisis Simplificado) ha sido creado con el objetivo de hacer el camino más sencillo a los equipos de analistas de sistemas informáticos. Al mismo tiempo se busca facilitar el acceso al conocimiento y la práctica de actividades adecuadas de análisis a cualquier persona que no cuente con la suficiente experiencia previa.

5.1 Componentes principales del PAS

Los componentes principales del PAS son:

- Proceso iterativo
- Fases
- Flujos de trabajo
- Diagramas de modelado

5.1.1 Proceso iterativo

El método iterativo de desarrollo permite crear sistemas por medio de pasos pequeños que son más fáciles de controlar a través de la repetición continua de una versión modificada del proceso de desarrollo en cascada donde la iteración actual emplea como entrada el resultado obtenido en la iteración anterior.

El proceso iterativo es, en esencia, la repetición continua de una versión ligeramente modificada del proceso de desarrollo en cascada (ver capítulo 1) que pasa por las actividades de análisis, diseño, desarrollo, pruebas, implantación y mantenimiento, donde cada proceso en cascada emplea como entrada la salida de la iteración anterior.

Cada iteración fragmenta el trabajo en mini proyectos, aunque no son entidades completamente independientes, ya que son sólo un paso dentro del proyecto.

El análisis de un sistema usando el proceso iterativo pasa por tres flujos de trabajo: requerimientos, análisis y diseño.

Las iteraciones de las primeras fases buscan determinar el ámbito del proyecto, eliminar riesgos críticos y crear la línea de la propuesta de arquitectura. A medida que se avanza en el proyecto se reducen los riesgos gradualmente y se van implementando los componentes del sistema.

Al final de cada iteración se obtiene información, registrada en forma de documentos, que nos permite planear y ajustar los objetivos del proyecto para la siguiente iteración y después de todas las iteraciones planificadas, el análisis debe contener todo lo necesario para llegar a un sistema funcional que cubra totalmente las expectativas del usuario.

La documentación es una actividad que puede parecer improductiva, sin embargo es una herramienta de vital importancia para el desarrollo del sistema, no sólo sirve como referencia futura dentro del mismo proyecto, sino como base para el análisis, la reducción de riesgos, la toma de decisiones, la planeación, priorización, ajuste y refinamiento de los requerimientos y de la arquitectura del sistema.

5.1.2 Fases

La creación del proyecto se desarrolla en un ciclo completo, el cual se divide en fases y cada una de ellas termina en un punto de transición.

En la fase de inicio se identifican y priorizan los riesgos más importantes, se plantean las restricciones del proyecto, se hace una proyección del proyecto, se diseña el macroproceso, se generan los casos de uso, se planifica en detalle la fase de elaboración y se estima el proyecto de manera aproximada. Se propone una arquitectura candidata que posteriormente modelará el funcionamiento del sistema.

Durante la fase de elaboración se especifican en detalle la mayoría de los casos de uso del proyecto y se diseña la arquitectura del sistema. La arquitectura se puede expresar a través de todos los modelos del sistema, los cuales en su conjunto representan al sistema completo. El resultado de esta fase es una línea base de la arquitectura, momento en el cual el líder de proyecto esta en disposición de planear actividades y estimar los recursos necesarios para terminar el proyecto. Es en esta fase cuando los flujos de trabajo de requerimientos y análisis requieren de un trabajo más intenso, pues establecen en su totalidad la arquitectura del sistema que será utilizada en fases posteriores.

Durante la fase de construcción se crea el sistema, la descripción evoluciona hasta convertirse en un proyecto preparado para ser entregado a los usuarios. Al final de esta fase el sistema contiene todos los casos de uso que se acordaron desarrollar y los flujos de trabajo de requerimientos y análisis se limitan al control de cambios.

5.1.3 Flujos de trabajo

EL PAS agrupa conjuntos de actividades relacionadas entre sí en tres flujos de trabajo: requerimientos, análisis y diseño.

En el flujo de trabajo de requerimientos el analista obtiene información acerca del sistema. Cada iteración permite afinar el conocimiento que se tiene acerca de lo que el usuario necesita hacer y es en este flujo de trabajo donde las actividades están enfocadas a ese objetivo.

En el flujo de trabajo de análisis se estudia la información recopilada acerca de los requerimientos del usuario con el fin de generar modelos de los casos de uso que se emplearán en el flujo de trabajo de diseño para generar un prototipo del sistema en la fase de inicio.

En el flujo de trabajo de diseño se integran los modelos del flujo de análisis para generar un prototipo del sistema que se pueda presentar al usuario para su aprobación o ajuste.

5.1.4 Diagramas de modelado

Los diagramas de modelado permiten al analista representar gráficamente un sistema desde ciertos puntos de vista, con el objeto de hacer más fácil su interpretación y estudio. La obtención de las características o requerimientos de un aspecto del sistema y que quedan representados el diagrama son obtenidos a partir de entrevistas y cuestionarios durante las primeras fases del desarrollo del sistema.

Los diagramas que se emplearán en el PAS para modelar un sistema están representados en la figura 5.1.

- Diagrama de casos de uso
- Diagrama de estados
- Diagrama de actividades

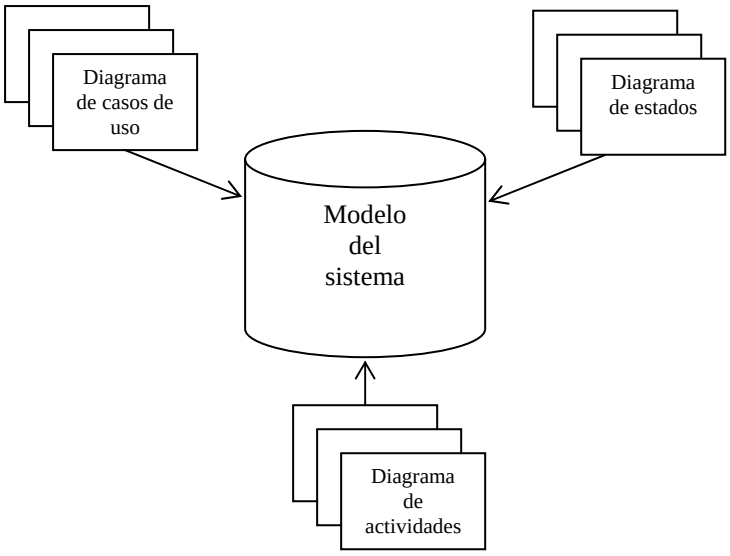


Figura 5.1 Los diagramas de casos de uso, estados, actividades y capas permitirán al analista generar un modelo adecuado del sistema

5.1.4.1 Diagrama de casos de uso

Los diagramas de caso de uso representan gráficamente el comportamiento de un sistema. Estos diagramas presentan una opinión de alto nivel de cómo es usado el sistema y de como lo ven los actores.

Un diagrama de caso de uso contiene:

- *Actores*. Entidad fuera del sistema
- *Caso de uso.* Limites que identifican que debe hacer el sistema
- *Iteraciones o relaciones entre actores y casos de uso.* incluye asociaciones, dependencias y generalizaciones.

Los diagramas de casos de uso pueden ser usados durante el análisis para extraer los requerimientos del sistema y entender como el sistema debe trabajar. Durante la fase de diseño puede usar estos diagramas para especificar el comportamiento del sistema como implementación.

El elemento actor se representa con la siguiente figura 5.2.

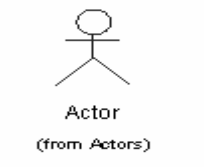


Figura 5.2 Figura representativa de un actor, que no necesariamente es una persona

El actor se encuentra al examinar preguntas como:

¿Quién usa directamente el sistema?

¿Quién es el responsable de darle mantenimiento al sistema?

¿Qué hardware externo se usa?

¿Existen otros sistemas que interactúen con el sistema?

Las necesidades de los actores son importantes puesto que son usadas para encontrar los casos de uso.

El nombre del actor se le asigna por el caso de uso en el que interviene, por ejemplo, cliente, vendedor, empleado.

El elemento caso de uso se puede ver representado en la figura 5.3.



Figura 5.3 Figura representativa de un caso de uso, generalmente un proceso, un programa o un sistema.

El caso de uso nos ayuda a capturar requerimientos, comunicación con el usuario final y líder del proyecto además de probar el sistema. Los casos de uso son el mejor descubrimiento para examinar los actores y definir que quieren hacer en el sistema.

Para asignarle un nombre al caso de uso se debe tomar en cuenta que esta asignado a un proceso y empieza con un verbo, por ejemplo transferencia bancaria, asignar pedidos, etc.

Forma de relacionar los elementos

- Relación de asociación como se muestra en la figura 5.4.

Una asociación relaciona la comunicación entre un caso de uso y un actor. Las asociaciones son las más generales de todas relaciones y significa la más débil semánticamente. Si dos objetos son considerados por separado, generalmente la relación es una asociación.

La herramienta de la asociación es unidireccional sobre un diagrama con una flecha sola a un final de la asociación. El final con la flecha demuestra quién o qué debe estar recibiendo la comunicación.

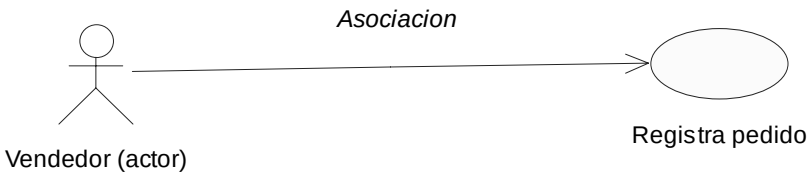


Figura 5.4 Figura representativa de la relación entre un actor y un caso de uso, la cual es llamada asociación.

Las reglas para asociar actores y casos de uso son:

- Relación de generalización

La generalización es la relación entre un caso de uso general y un caso de uso más específico. Una generalización es mostrada como una flecha sólida, y va del elemento más específico a un elemento más general. La punta de una generalización es un triángulo hueco grande que señala al elemento más general. La representación de una generalización se puede ver en la figura 5.5

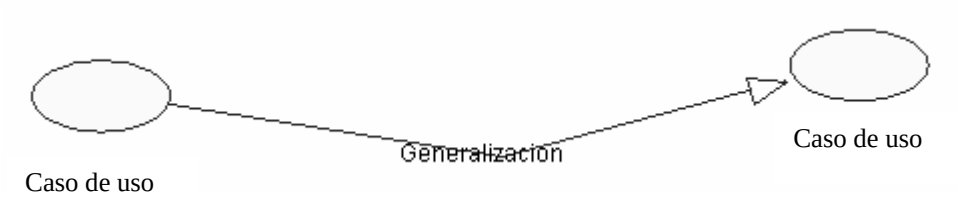


Figura 5.5 Figura representativa de la generalización de dos casos de uso.

- Relación de dependencia

Una dependencia es una relación entre dos elementos en los que un cambio para un elemento afectará al otro elemento. Se usa una relación de dependencia para conectar elementos con el mismo nivel del significado. Típicamente, una relación de dependencia indica que las operaciones del cliente invocan las operaciones del sistema. La figura 5.6 muestra dos ejemplos de dependencias.

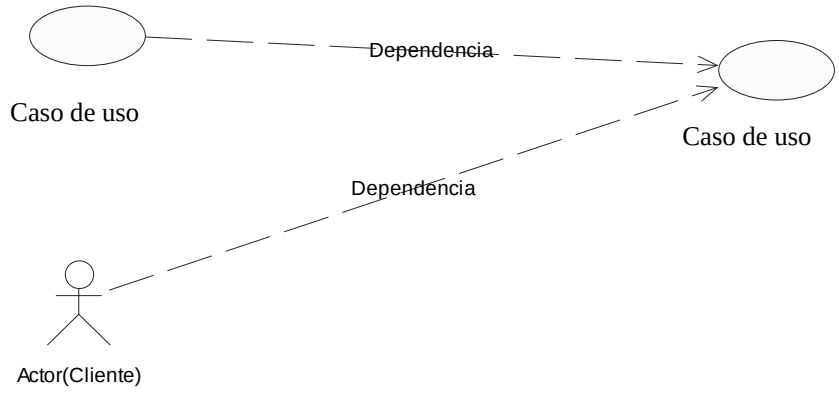


Figura 5.6 Figura representativa de la dependencia entre actores y casos de uso.

Formas de obtener casos de uso:

- Mediante la observación directa de las labores realizadas, teniendo en cuenta que el proceso que se observa es el que probablemente se quiera hacer más eficiente.

- A través de entrevistas con el usuario, cuidando realizar las suficientes a los usuarios más representativos.

- Finalmente se puede entrevistar un experto en el proceso que se está modelando, y pedirle que se simule un tarea, este experto relatará paso a paso la ejecución de la tarea y con ello se formará el caso.

Las tres opciones son complementarias y de preferencia deben usarse todas, con las dos primeras obtenemos el flujo más común y real del proceso, con la tercera aseguramos que el proceso normal esta correcto.

Tanto el actor como el caso de uso tienen una interfaz en particular, esto es que cosas se espera que hagan, con esto podemos delimitar exactamente qué hace y qué no hace y cuáles son las responsabilidades del usuario (actor) y del sistema (caso de uso). Es decir la forma mas practica de definir el alcance de un caso de uso es cuando proporciona un valor al usuario, el cual debe tener un resultado observable que le de valor a su trabajo, si el caso de uso esta tan fragmentado que no se tiene un resultado observable es mejor agrupar la operación en un solo caso.

En resumen el caso de uso debe dar un valor agregado o cumplir un objetivo por el cual el actor este dispuesto a realizar un esfuerzo y que este valor sea particular para un solo actor para evitar que el caso sea muy grande.

5.1.4.2 Diagrama de estados

Determina los estados que un objeto puede tener y cómo los eventos afectan esos estados a través del tiempo. Un diagrama de estados debe abarcar todos los elementos que tengan estados y conducta definidos claramente.

Todos los objetos tienen un estado y éste es el resultado de actividades previas ejecutadas por el objeto. Ese estado está determinado por los valores de los atributos de este objeto y sus relaciones con otros objetos.

Un objeto está representado principalmente por los siguientes elementos: estado, evento y transición.

- *Estado*: identifica un período de tiempo del objeto (no instantáneo) en el cual el objeto esta esperando alguna operación, tiene cierto estado característico o puede recibir cierto tipo de estímulos.

- *Evento:* Es una ocurrencia que puede causar la transición de un estado a otro de un objeto. Esta ocurrencia puede ser una de varias cosas*:*

    - Condición que toma el de verdadero o falso.
    - Recepción de una señal de otro objeto en el modelo.
    - Recepción de un mensaje.
    - Paso de cierto período de tiempo, después de entrar al estado o de cierta hora y fecha particular.

- *Transición.* Es una relación de tres o más estados en una transición de múltiples fuentes o múltiples destinos.

En la figura 5.7 se muestra el diagrama de estados de una compra, en la cual uno de los estados es seleccionado (el objeto asiento pasa del estado libre ha seleccionado) y la venta pasa de confirmando a vendiendo.

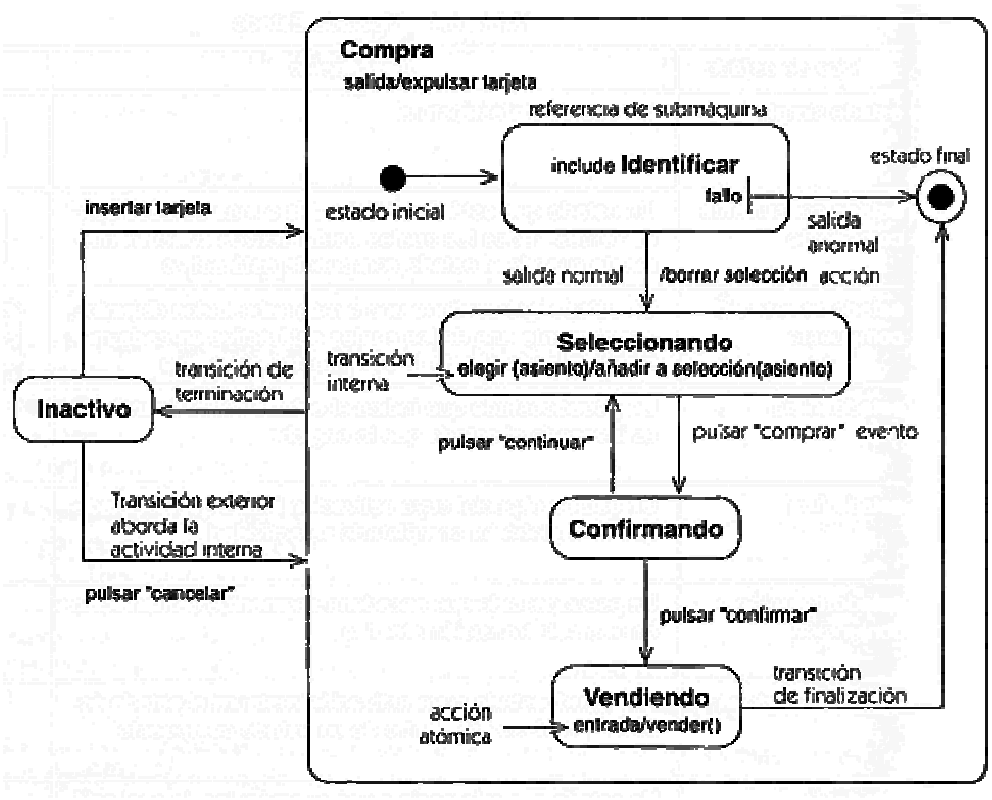


Figura 5.7 Ejemplo de diagrama de estados.

5.1.4.3 Diagrama de actividades

Un diagrama de actividades es un caso especial de un diagrama de estados en el cual casi todos los estados son estados de acción (identifican que acción se ejecuta al estar en él) y casi todas las transiciones son enviadas al terminar la acción ejecutada en el estado anterior. Puede dar detalle a un caso de uso, un objeto o un mensaje en un objeto. Sirven para representar transiciones internas, sin hacer mucho énfasis en transiciones o eventos externos.

El estado de acción representa una acción interna, por lo menos una transición que identifica la culminación de la acción (por medio de un evento implícito). No deben tener transiciones internas ni transiciones basadas en eventos (Si este es el caso, se representa en un diagrama de estados). El estado de acción permite modelar un paso dentro del algoritmo y se representa con un rectángulo.

Las flechas entre estados representan transiciones con evento implícito. Pueden tener una condición en el caso de decisiones.

Las decisiones se representan mediante una transición múltiple que sale de un estado, donde cada camino tiene un nivel distinto. Se representa mediante un diamante al cual llega la transición del estado inicial y del cual salen las múltiples transiciones de los estados finales.

El indicar de paralelismo es una barra de donde salen los flujos de actividad paralelos, y se reúnen en una barra de sincronización. Los carriles nos permiten delimitar a quien pertenecen las actividades que se desarrollan en paralelo.

El diagrama de actividades se utiliza para:

- Análisis de casos de uso. Durante el análisis de los casos de uso no estamos interesados en asociar acciones a objetos, sino en entender qué acciones se necesitan llevar a cabo y cuales son las dependencias en el comportamiento.

- Comprensión del flujo de trabajo a lo largo de diferentes casos de uso.
- Modelado de aplicaciones.

En la figura 5.8 se representa un diagrama de actividades.

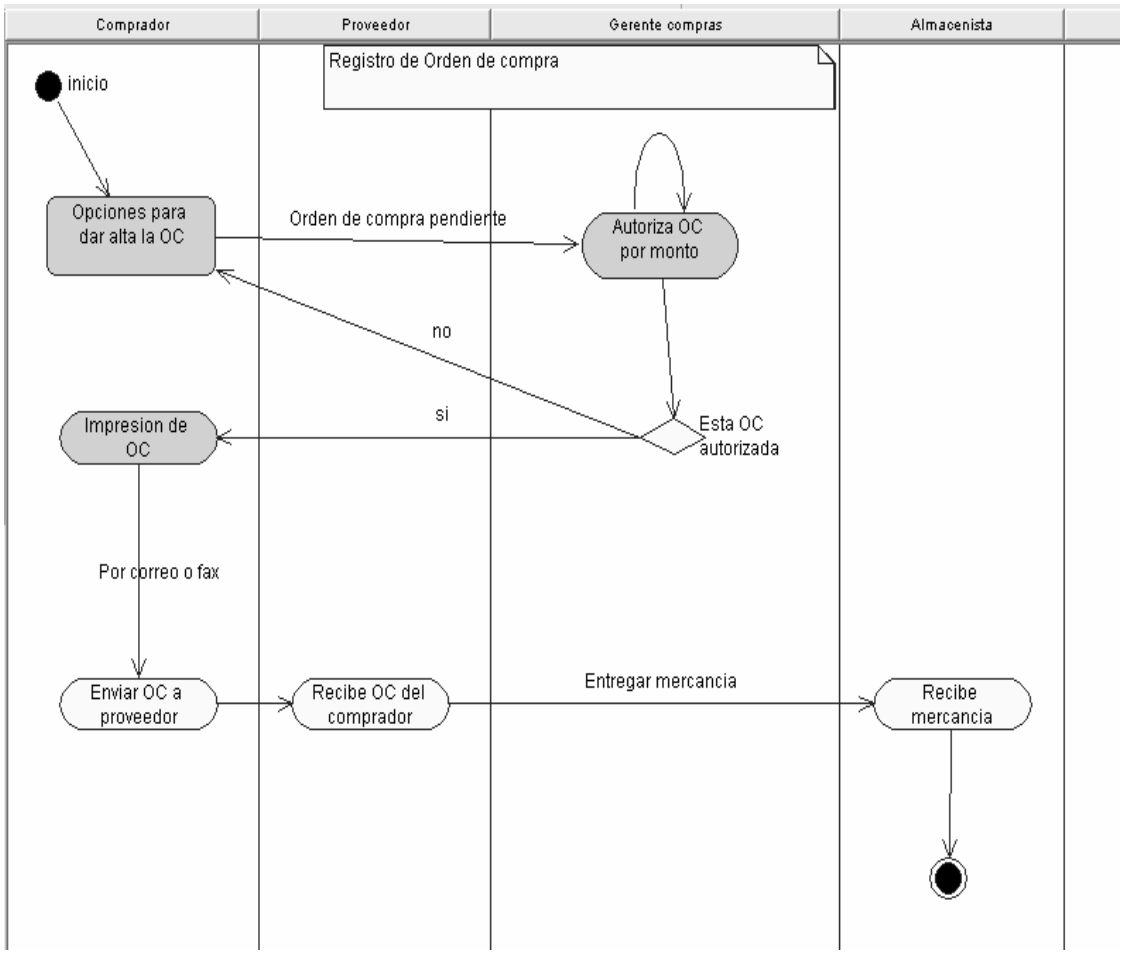


Fig 5.8 Ejemplo de diagrama de actividades

5.2 Propiedades del PAS

Las principales propiedades que presenta el PAS son la flexibilidad, la adaptabilidad, la accesibilidad y la anticipación o prevención de problemas generados por los riesgos propios de un desarrollo de sistemas.

La ventaja de emplear el método iterativo como célula básica de trabajo es que permite dividir el proyecto en partes pequeñas que son fáciles de controlar, asimismo, faculta al equipo de desarrollo para cubrir las etapas del mismo sin importar la complejidad global de éste, ya que se puede emplear un número mayor o menor de iteraciones dentro de una fase hasta cubrir sus objetivos.

En sistemas muy sencillos es posible que una sola iteración le proporcione al analista lo necesario para que se desarrolle un prototipo funcional que represente la manera en que operará el sistema.

Por el contrario, un sistema complejo puede requerir de más iteraciones para que el equipo de analistas conozca lo suficiente del sistema para representarlo en modelos y documentarlo adecuadamente. Es esencial entender que no es el número de iteraciones lo que se debe tener en cuenta, sino el alcanzar los objetivos marcados para cada iteración.

El PAS es una primera propuesta, un paso inicial en la búsqueda de una nueva cultura de análisis para el desarrollo de sistemas informáticos en nuestro país, que puede ser adaptado por el analista a partir de sus experiencias personales.

En el capítulo uno, Conceptos básicos, se explicaron los pasos típicos un desarrollo en cascada y particularmente el de análisis. El PAS contempla las cuatro actividades del análisis en cascada, sin embargo se realizan desde un enfoque distinto:

- Investigación Preliminar
  Representada por las iteraciones uno y dos de la fase de inicio, es decir, desde el acercamiento hasta el plan de actividades pasando por los flujos de trabajo de requerimientos y análisis.

- Determinación de Requerimientos del Sistema
  Esta actividad inicia en la iteración tres de la fase de inicio y continúa hasta la iteración dos de la fase de elaboración, ya que el sistema es dividido en pequeños subsistemas de los que se obtienen requerimientos poco a poco. Al llegar a la iteración uno de la fase de construcción la determinación de los requerimientos del sistema se convierte control de cambios, es decir, los requerimientos han sido recolectados por completo y el análisis se limita a llevar un control de cambios y reducir los riesgos asociados.

- Prototipo del Sistema
  Queda comprendido por la iteración tres de la fase de inicio, prototipo, en el flujo de trabajo de diseño.

- Ajustes y Aceptación del Prototipo
  La flexibilidad del procedimiento de análisis permite que se puedan generar tantas iteraciones como sean necesarias, hasta que se lleven a cabo los ajustes necesarios y se alcance la aceptación del prototipo. Al final esta fase se obtiene como resultado un plan de actividades actualizado, que servirá como guía para las siguientes fases del desarrollo del sistema.

Lo anterior nos hace notar que a diferencia del método de análisis de sistemas informáticos en cascada, el enfoque iterativo hace más accesible el cumplir estas tareas de manera eficiente, adecuada y controlada a través de procesos secuenciales más pequeños y simples.

Finalmente, el procedimiento no sólo registra los requerimientos del usuario y los modela para desarrollar el sistema, sino que emplea esta información para identificar, analizar y minimizar o eliminar de manera oportuna los riesgos asociados a dichos requerimientos.

5.3 Descripción del procedimiento PAS

A continuación se expone a detalle el PAS, indicando de manera secuencial y ordenada las actividades que se deben seguir para llevarlo a cabo adecuadamente.

El PAS ha sido sintetizado en la tabla 5.9, la cual permitirá al analista contar con una referencia rápida de las actividades que componen al procedimiento y los resultados a obtener.

Los elementos del PAS se denotarán con un identificador numérico, constituido por cuatro dígitos que representan la fase, la iteración, el flujo de trabajo y la actividad específica a realizar:

Notación (A.B.C.D)	Descripción
A Primer digito	Número de fase
B Segundo digito	Número de iteración
C Tercer digito	Número de flujo de trabajo
D Cuarto digito	Número de actividad

Esta nomenclatura permitirá llevar un mejor control de las actividades y facilitará su localización en la tabla 5.9.

La secuencia que a continuación se emplea para detallar las actividades señala la forma en que se deberá seguir el PAS y la manera de leer la tabla antes mencionada.

5.3.1 Fase de inicio (1.0.0.0)

En la fase de inicio el criterio esencial es el conocimiento de la viabilidad del proyecto, que se alcanza por medio la identificación y reducción de riesgos críticos para el sistema, la creación de una propuesta de arquitectura a partir del desarrollo de un subconjunto de requerimientos clave y del modelado de los casos de uso, la estimación inicial de esfuerzos y tiempos.

El objetivo general de la etapa de inicio es poner en marcha el análisis, es decir, desarrollar el análisis del negocio hasta el punto en el cual es posible justificar el arranque del proyecto. Para analizar el negocio es necesario que primero se delimite el alcance y ámbito del sistema propuesto, de modo que sea posible discernir que se cubre con la arquitectura del proyecto, a la cual se nombrará “arquitectura candidata”.

También se busca adelantarse a riesgos que puedan llevar al fracaso al proyecto, para mitigarlos de forma temprana, o decidir si se reduce el ámbito del proyecto o se aumentan los recursos necesarios.

La fase de inicio de este procedimiento se dividió en tres iteraciones, las cuales se detallan a continuación.

- **Iteración 1 (1.1.0.0)**

En el flujo de trabajo de requerimientos de esta iteración se busca un acercamiento con la organización que efectúa la solicitud de un sistema para conocerla, asignar un nombre al proyecto y se dan acuerdos para el manejo del proyecto. Se definen requerimientos globales del sistema y requerimientos especiales o restricciones con que debe cubrir el sistema.

En el flujo de trabajo de análisis se debe realizar un listado de riesgos priorizado y hacer un análisis del tiempo aproximado que durará el proyecto.
Hasta este momento no se pasa por los flujos de diseño, implementación y pruebas.

- **Flujo de trabajo de requerimientos (1.1.1.0)**

En la fase de inicio el énfasis reside principalmente en el primer flujo de trabajo, el de requerimientos, el cual incluye identificar y detallar los casos de uso pertinentes en esta fase lo cual incluye los siguientes aspectos o actividades:

Actividades del PAS (Tabla 5.9)

FLUJO DE TRABAJO	1 FASE DE INICIO			2 FASE DE ELABORACIÓN		3 FASE DE CONSTRUCCIÓN	
	Iteración 1	**Iteración 2**	**Iteración 3**	**Iteración 1**	**Iteración2**	**Iteración 1**	**Iteración…n**
1 Requerimiento	**1.** Acercamiento **2.** Alcance global **3.** Restricciones	**1**. Inducción**(F03)**	**1.** Levantamiento de requerimientos **(F02) (F03) (F05)**	**1.** Levantamiento de requerimientos de los casos de uso restantes**(F02)**	**1.** Refinamiento de requerimientos	**1.** Control de cambios **(F02)**	
2 Análisis	**1.** Riesgos **2.** Proyección del proyecto (en tiempo) **(F07)**	**1.** Macroproceso **(D03)** **2.** Matriz de procesos propuestos **(F06)** **3.** Selección de procesos (priorizar) **4.** Restricciones **5.** Riesgos **6.** Análisis de factibilidad **(F08)**	**1.** Procesos críticos no detectados **2.** Casos de uso críticos **(D01,D02,D03)** **3.** Relaciones entre procesos (priorizar) **4.** Riesgos	**1.** Priorizar procesos **(D01,D02,D03)** **2.** Riesgos **3.** Restricciones	**1.** Priorizar Procesos **2.** Riesgos **3.** Arquitectura **4.** Planificación de la fase de construcción	**1.** Priorizar procesos **2.** Riesgos **3.** Arquitectura **4.** Plan de trabajo	
3 Diseño			**1.** Prototipo		**1.** Detallar, priorizar y estructurar los modelos de caso de usos	**1.** Diseño de casos de uso	
4 Resultado de la Iteración	Planteamiento del proyecto **(F01)(F02)**	Plan de actividades**(F04)**	Plan de actividades actualizado**(F04)**	Arquitectura y plan de actividades actualizados	**1.** Plan de proyecto para construcción **2.** Documento integral del análisis**(F09)**	Arquitectura y plan de actividades actualizados	

Formatos útiles

(F01) Minutas
(F02) Requerimientos
(F03) Actividades
(F04) Cronograma
(F05) Puestos
(F06) Matriz de proceso
(F07) Proyección del proyecto
(F08) Factibilidad
(F09) Documento Integral

Diagramas de modelado

(D01) Diagrama de casos de uso
(D02) Diagrama de actividades
(D03) Diagrama de estados

- Acercamiento (1.1.1.1)

Esta actividad se realiza a través de una serie de reuniones entre el equipo de desarrollo y representantes de los usuarios, en las cuales se establece lo siguiente:

- Asignar un nombre al proyecto.
- Presentación del equipo de proyecto de la empresa consultora y el cliente.
  - Datos del negocio (Razón social, dirección, teléfonos)
  - Función de la empresa
  - Objetivo de la empresa
  - Misión y Visión de la empresa
  - Lista de lugares y responsables donde se realizara el proyecto
  - Para cuando el proyecto este fuera del lugar de residencia del equipo incluir datos del hotel
- Acuerdos para el manejo del proyecto
  - Toma de minutas.
  - Toma de requerimientos.
  - Horarios de trabajo
  - Reportes de actividades semanales del equipo de proyecto
  - Glosario de términos.
  - Ningún acuerdo verbales es tomado en cuenta

- Alcance global (1.1.1.2)

Incluye los requerimientos globales y características que el usuario especifica durante las primeras entrevistas con el cliente. Son requerimientos muy generales que permiten al analista comprender en forma global el sistema que se desea construir. En esta actividad del flujo de trabajo de requerimientos de la iteración uno, el usuario debe dar un panorama general de la funcionalidad que debe tener el sistema sin entrar a los detalles.

- Requerimientos especiales o restricciones (1.1.1.3)

Son características del sistema que el cliente necesita y no pueden ser modificadas, por ejemplo, características de rendimiento, interfaces, requerimientos de diseño físico así como restricciones en la arquitectura o en la implementación.

Una restricción de interfaz es la especificación del medio de interacción con un elemento externo con el cual debe comunicarse el sistema, o que establece características especiales de formatos, tiempos u otros factores de relevancia en esta interacción.

Una restricción física especifica una característica con la cual debe contar ese sistema en referencia al equipo de cómputo involucrado o el lugar donde deberá encontrarse el mismo, por ejemplo: se puede tener como requisito físico el uso de servidores SUN con sistema operativo Solaris 8.0 y como clientes computadoras personales con procesador Intel Pentium III como mínimo.

Una restricción de implementación especifica o limita la forma de codificar o construir un sistema, por ejemplo:

- Normas de codificación
- Lenguajes de programación
- Políticas de integridad de la base de datos
- Limitaciones de recursos
- Estándares requeridos
- Entornos operativos

Otras restricciones importantes son los requerimientos legales y las normatividades, por ejemplo:

- Requerimientos de seguridad en transmisiones y accesos
- Disponibilidad
- Facilidad de aprendizaje

Los requerimientos que se deben obtener durante esta iteración son:

- Entender en forma general las características principales con que debe cumplir el sistema

- Registrar las características (requerimientos especiales o restricciones) específicas indicadas por el usuario que pueden representar una limitante en cuanto a las opciones de hardware y/o software a utilizar para el desarrollo e implementación del sistema, así como limitantes de logística.

Para llevar un registro ordenado de las entrevistas y reuniones que se han realizado con los clientes o usuarios para definir los requerimientos mencionados anteriormente se deben elaborar minutas que permitan conocer el tema tratado y los acuerdos tomados durante estas reuniones.

- **Flujo de trabajo de análisis (1.1.2.0)**

Tomando en cuenta la información obtenida durante la recolección de requerimientos anterior se debe realizar un primer análisis de riesgos y hacer una propuesta del tiempo aproximado que se usara para la realización del proyecto.

- Riesgos (1.1.2.1)

El riesgo es un factor, cosa o elemento que constituye un peligro cuyo grado es incierto, se puede ver como un asunto que tiene cierto grado de probabilidad de poner en peligro el éxito del proyecto.

El desarrollo de software se enfrenta con riesgos al igual que cualquier otra actividad de ingeniería, y se afronta esta realidad identificando los riesgos tan pronto como sea posible dentro del desarrollo y tratándolos rápidamente.

Veamos algunos ejemplos de riesgos:

- Es posible que una página de Internet que vayamos a desarrollar se implemente en una plataforma que probablemente no sea capaz de soportar la carga de solicitudes remotas.

- Un sistema que recibe cierta cantidad de datos como entrada no especificados en la fase de inicio, puede que requiera realizar cálculos complejos e incluso sea necesario que envíe una respuesta en un tiempo aún no especificado.

En el proceso iterativo se identifican y se priorizan los riesgos para cubrir las necesidades de los usuarios y para establecer una arquitectura más robusta para incorporar cambios con un riesgo bajo de tener que rediseñar algo.

Muchos de los riesgos no son visibles para el analista hasta que se prueba el software, motivo por el cual se deben llevar a cabo iteraciones que examinen los riesgos desde las fases de inicio y elaboración hasta la codificación y prueba, ya que el objetivo es acabar con los riesgos en una iteración temprana.

En la fase de construcción la mayoría de los riesgos deberán ser reducidos hasta un nivel de rutina, lo cual significa que se someten a prácticas ordinarias de desarrollo.

- Proyección del proyecto (1.1.2.2)

Durante esta actividad se debe hacer un análisis de las características que debe cumplir el sistema y determinar una fecha estimada de entrega del proyecto, aclarando al cliente que la fecha es tentativa de acuerdo a los requerimientos con que se cuenta hasta el momento.

En este caso la documentación que se genera es un cronograma de actividades generales y un resumen de reuniones y de acuerdos del proyecto.

- **Resultado de la iteración (1.1.4.0)**

El resultado de esta iteración será una serie de documentos (minutas y requerimientos) que resumen el planteamiento general del proyecto donde se indican las principales características del sistema, responsabilidades, atributos y un listado inicial de riesgos.

- **Iteración 2 (1.2.0.0)**

En el flujo de requerimientos de esta iteración se obtienen características generales de cada uno de los procesos que componen el sistema, información útil para realizar un análisis de cómo están organizados los procesos que forman el macroproceso de la empresa, se hace un análisis de factibilidad y un plan de proyecto en donde se especifiquen en forma general las actividades y los tiempos de cada proceso.
Hasta este momento no es necesario por el flujo de diseño, implementación y pruebas.

- **Flujo de trabajo de requerimientos (1.2.1.0)**

Para obtener los requerimientos en esta fase se deben completar adecuadamente las siguientes actividades:

- Inducción (1.2.1.1)

El equipo de analistas hace una investigación de campo en las instalaciones del cliente para conocer en forma práctica pero general sus procesos, de ese modo se inicia la asociación procesos contra puestos y sus relaciones. De esta investigación el analista obtiene información más clara acerca de los requerimientos del usuario y sobre los posibles riesgos asociados a dichos requerimientos.

- **Flujo de trabajo de análisis (1.2.2.0)**

En este flujo de trabajo se comienza a determinar una arquitectura candidata conformada por los casos de uso críticos del sistema, además de priorizar procesos, establecer restricciones y afinar la lista de riesgos.

- Macroproceso (1.2.2.1)

Es una descripción muy general de los principales casos de uso detectados al momento el cual permite al analista empezar a familiarizarse con el sistema y que asistirá en la conformación de la arquitectura candidata.

Se hace un análisis de lo que se observó en la inducción y de ahí se define como esta formado el macroproceso. El macroproceso indica el intercambio general de información, las interacciones entre los procesos más generales que realiza el usuario y que deberá incluir el sistema o que estarán interactuando con él.

- Matriz de procesos propuestos (1.2.2.2)

Es una matriz que se crea con los conocimientos adquiridos en la inducción, incluye todos los procesos principales y generales que los componen, de tal manera, que se relacionen contra un puesto (no una persona en particular, sino un papel dentro de la organización).

El analista construye una matriz, con lo cual, puede consultar rápidamente el esquema de procesos en el sistema y quién lo realiza, esto tiene como objetivo detectar posibles riesgos, áreas de oportunidad y refinar los requerimientos.

- Selección de procesos críticos (1.2.2.3)

Analizar y seleccionar los procesos y riesgos asociados que son críticos (claves, esenciales) para el sistema. Esto permitirá al equipo de desarrollo detectar los procesos y riesgos que pueden influir de manera determinante en el desarrollo del proyecto y en el cumplimiento de los planes de trabajo hasta este momento propuestos.

Siempre que se determina una serie de riesgos se debe iniciar el trabajo de mitigarlos o eliminarlos completamente de ser posible, ya que el costo de eliminar un riesgo en etapas tardías es mucho mayor.

- Restricciones (1.2.2.4)

El analista ahora tiene una visión más amplia de lo que el sistema deberá realizar, y el equipo de desarrollo debe analizar las posibles restricciones que surjan de las actuales condiciones de operación, de ese modo se podrá revisar junto con el usuario si esas restricciones se pueden eliminar.

- Riesgos (1.2.2.4)

Los riesgos obtenidos en la iteración anterior si no han sido eliminados deberían ser minimizados para este momento. Sin embargo, la inducción abre un nuevo grupo de riesgos que surgen del nuevo conocimiento adquirido sobre el sistema. El equipo de desarrollo debe definir los nuevos riesgos del sistema y comenzar la labor de eliminarlos o minimizarlos para evitar que afecten etapas posteriores del desarrollo.

- Factibilidad (1.2.2.5)

Se hace un análisis de factibilidad usando la información obtenida en la inducción, el estudio del macroproceso y las restricciones y riesgos detectados para evaluar que tan factible es pasar por todas las siguientes fases del desarrollo del proyecto.

- **Resultado de la iteración (1.2.4.0)**

Se obtiene documentación donde se especifica una primera versión del cronograma de procesos y sus tiempos de finalización, una matriz de procesos contra puestos, una descripción de los puestos del personal clave en cada uno de los procesos y una lista de procesos críticos para el proyecto y sus riesgos asociados de forma más específica y actualizada. Para obtener este resultado, el equipo de desarrollo se puede basar en las plantillas de actividades, cronogramas y matriz de procesos propuestos.

- Plan de actividades (1.2.4.1)

Se propone un plan de actividades general estimando tiempos para cada uno de los procesos que componen el macroproceso. En este momento no pueden ser establecidos con toda precisión, sin embargo los tiempos se irán refinando conforme se vaya progresando en el proyecto. Después de todo no es sino hasta el final de la fase de elaboración cuando se hace una propuesta definitiva.

- **Iteración 3 (1.3.0.0)**

En el flujo de requerimientos de esta iteración se obtienen los requerimientos críticos del sistema tomando en cuenta el listado generado en el análisis de la iteración anterior, durante la actividad de levantamiento de requerimientos de esta iteración puede ser que se detecten requerimientos críticos que no se habían tomando en cuenta, por lo que en la etapa de análisis se debe regenerar el listado de procesos críticos incluyendo los no detectados, se definen las relaciones entre todos los procesos, se priorizan los procesos de acuerdo a su importancia y relaciones y se proponen algunos casos de uso a programar en esta iteración.

En el flujo de diseño, se propone una primera versión del proyecto que servirá como prototipo utilizando los casos de uso más representativos seleccionados en el análisis.

- **Flujo de trabajo de requerimientos (1.1.3.0)**

En este flujo de trabajo se obtienen a detalle los requerimientos necesarios para el modelado de los procesos críticos del sistema a través de entrevistas y cuestionarios realizados a los usuarios involucrados.

  - Levantamiento de requerimientos (1.3.1.1)

Se levantan los requerimientos de los procesos críticos del sistema de acuerdo a la lista generada en la actividad de análisis de la iteración anterior, para ello se puede utilizar un formato de levantamiento de requerimientos a través de una serie de entrevistas, cuestionarios y reuniones que permitan al equipo de analistas y al arquitecto conocer más a fondo los procesos críticos involucrados en el sistema.

- **Flujo de trabajo de análisis (1.3.2.0)**

En este flujo de trabajo el analista se encarga de refinar la lista de procesos críticos a partir de la nueva información obtenida y modela los casos de uso y sus relaciones. Finalmente actualiza la lista de riesgos asociados a los nuevos requerimientos y comienza a determinar soluciones para eliminarlos o minimizarlos.

  - Procesos críticos no detectados (1.3.2.1)

El equipo de desarrollo obtendrá a partir del levantamiento de requerimientos información esencial que le permitirá refinar la arquitectura candidata, agregando a la versión actual procesos que en su momento se habían considerado de menor importancia, pero que por las nuevas relaciones detectadas con otros procesos, por la logística de desarrollo o por los riesgos que están asociados a ese proceso, deberían ser considerados como críticos.

  - Casos de uso críticos (1.3.2.2)

Se genera una lista de casos de uso para los procesos determinados como críticos. Es una primera versión que deberá refinarse conforme se avanza en el ciclo de vida del proyecto.

  - Relaciones entre procesos (1.3.2.3)

Se definen las relaciones entre todos los procesos, críticos y no críticos, de modo que se conjunta una versión más completa de la arquitectura candidata. El analista en este momento obtendrá una visión más clara del sistema.

Se priorizan los procesos de acuerdo a importancia y relaciones como preparación para las fases de elaboración y construcción. La lista de riesgos debe ser tomada en cuenta como una variable dentro de los criterios de priorización.

- Riesgos (1.3.2.4)

Se agregan riesgos detectados a partir de la obtención de más detalles en los requerimientos del usuario, además de que se actualiza el estatus de los riesgos obtenidos en iteraciones anteriores. Del mismo modo se debe analizar y programar la manera de evitarlos o minimizarlos a la brevedad para evitar mayores costos en etapas posteriores del proyecto.

- **Flujo de trabajo de diseño (1.3.3.0)**

El objetivo principal del flujo de trabajo de diseño es esbozar la arquitectura candidata para ser incluida en la descripción de la arquitectura preliminar. A partir de la información obtenida en las pasadas iteraciones y con una lista de requerimientos, riesgos, prioridades y relaciones más refinados es necesario que el equipo de desarrollo genere una versión más completa de la arquitectura candidata, incluyendo en ella un estudio profundo de la información recopilada al momento.

- Prototipo (1.3.3.1)

Si es necesario elaborar un prototipo de demostración se hará empleando módulos prefabricados o técnicas de desarrollo rápido que permitan mostrar de forma simple la idea. El prototipo se muestra a usuarios representativos para asegurar que satisface sus necesidades y en caso de ser necesario hacer cambios. Es conveniente que el prototipo incluya casos de uso representativos del proyecto para que el usuario tenga una visión clara de lo que será el sistema.

- **Resultado de la iteración (1.3.4.0)**

Como resultado final se obtiene un documento el cual representa el plan de actividades actualizado y una primera versión del sistema el cual nos produce lo siguiente:

- Una lista de características del sistema a realizar.
- Una primera versión del modelo del negocio que describe el contexto del sistema
- Un esbozo de los modelos que representa una primera versión del modelo de casos de uso, análisis y diseño. También se genera una primera versión de los requerimientos adicionales.
- Un prototipo exploratorio que muestra el uso del sistema
- Una lista inicial de riesgos y una clasificación de casos de uso.
- Un primer borrador del análisis del negocio, que incluya, contexto del negocio, criterios de éxito, estimación de proyecto, y plan general de las fases del proyecto.

5.3.2 Fase de elaboración (2.0.0.0)

En esta fase el objetivo principal es alcanzar la capacidad de construir el sistema mediante:

- La identificación y reducción de riesgos importantes para la construcción del sistema
- La especificación de la mayoría de los casos de uso que representan la funcionalidad del sistema que se desarrollará
- La creación de una versión más detallada de la propuesta de arquitectura
- Generar un plan de proyecto detallado que sirva como guía para la fase de construcción
- Finalizar el análisis del negocio.

Cabe aclarar que durante la fase de elaboración sólo es necesario detectar y documentar de forma breve los casos de uso que intervienen en el sistema, es decir, no es necesario obtener todos los detalles ya que se recopilarán en el momento que sea necesario. Sólo se documentan a detalle los casos de uso que se consideren de importancia para la arquitectura del sistema.

En el enfoque típico de estudio de los casos de uso el analista busca entender la manera en la cual se realiza cierta tarea para reflejarlos en el nuevo sistema informático. En este procedimiento el enfoque cambia y el analista deberá detectar el objetivo real del usuario y pensar la mejor manera de llegar a él para que eso sea lo que se programe en el sistema.

- **Iteración 1 (2.1.0.0)**

En esta iteración equipo de analistas debe intensificar su esfuerzo en los flujos de trabajo de requerimientos y análisis, ya que se debe obtener información más detallada acerca de los procesos involucrados en el sistema. Los requerimientos deberán ser registrados, priorizados, modelados y asociados con riesgos para detallar todos los casos de uso involucrados, trabajo que permitirá al analista afinar la arquitectura del sistema

- **Flujo de trabajo de requerimientos (2.1.1.0)**

Durante el flujo de requerimientos de esta iteración se deben obtener los requerimientos a detalle de los procesos restantes del sistema. En el flujo de análisis se priorizan nuevamente los procesos, se actualiza el listado de riesgos y se analizan las restricciones pendientes por tomar en cuenta.

- Levantamiento de requerimientos de los casos de uso restantes (2.1.1.1)

Se levantan los requerimientos a detalle de todos los procesos relacionados con los casos de uso críticos. Estos procesos pueden ser componentes del macroproceso o pequeños procesos determinados como no críticos pero que intervienen en el sistema a realizar.

- **Flujo de trabajo de análisis (2.1.2.0)**

Los analistas modelan los requerimientos obtenidos en el flujo de trabajo anterior y los integran a la arquitectura del sistema.
Se realiza una priorización para asistir a la logística de desarrollo y para cumplir adecuadamente con los requerimientos del usuario. Se actualizan las listas de riesgos y restricciones derivados de la información recopilada.

- Priorizar procesos (2.1.2.1)

Se priorizan los procesos detectados en el flujo de trabajo anterior de modo que los desarrolladores hagan su trabajo en forma más ordenada y se optimicen tiempos de trabajo evitando programar tareas que deben esperar a que se terminen otras.

- Riesgos (2.1.2.2)

Cada requerimientos puede presentar riesgos asociados y la lista debe actualizarse y la labor de eliminarlos comienza.

- Restricciones (2.1.2.3)

Se actualiza la lista de restricciones y características especiales no tomadas en cuenta en etapas iteraciones anteriores y que se decubren a partir de la información detallada de los procesos del sistema.

- **Resultado de la iteración (2.1.4.0)**

A partir de la información recopilada y de los modelos de los nuevos requerimientos generados por los analistas, es posible obtener una nueva arquitectura del sistema mucho más estable y que se acerca a la

versión final de la misma. Se obtiene la documentación de plan de actividades de una forma más refinada y específica.

Se pueden realizar varias iteraciones similares a la iteración uno hasta que se tengan analizados todos los procesos que comprenden el sistema.

- **Iteración 2 (2.2.0.0)**

Durante el flujo de requerimientos de esta iteración se debe obtener más información sobre los requerimientos del sistema. En el flujo de análisis nuevamente se priorizan a detalle los procesos y se obtiene una lista más refinada sobre los riesgos para obtener un bosquejo de la arquitectura del sistema y así realizar la planificación de la fase de construcción.

En esta iteración se termina la arquitectura definitiva del sistema, obteniendo todo lo necesario para que el equipo de desarrollo trabaje sin mayor dificultad en la fase de construcción y donde este procedimiento reduce su labor al mínimo. Debe notarse que la complejidad del sistema indica el número de iteraciones iguales necesarias para alcanzar el total de procesos y casos de uso modelados en la arquitectura del sistema.

El procedimiento especifica sólo dos iteraciones, pero este número está abierto a la cantidad de trabajo que se requiere para alcanzar a detectar y modelar todos los casos de uso.

- **Flujo de trabajo de requerimientos (2.2.1.0)**

En el flujo de trabajo de requerimientos recopila información para determinar los casos de uso restantes y se refina la información obtenida en la iteración anterior.

- Refinamiento de requerimientos (2.2.1.1)

Se determina la mayoría de los requerimientos, para tratar de no pasar por alto nada que sea importante para el diseño de la arquitectura. Se debe de identificar aproximadamente el 80 por ciento de los casos de uso y se debe de describir detalladamente entre el 40 y 80 por ciento estos mismos. Todo esto es para la obtención de una descripción de la arquitectura del sistema, versiones de todos los módulos y no encontrarse con riesgos inesperados.

- **Flujo de trabajo de análisis (2.2.2.0)**

En este flujo de trabajo el analista repite las labores de la iteración anterior: refinar la arquitectura, que para este momento deberá ser la arquitectura final del sistema, actualizar la lista de riesgos y prioridades y finalmente deberá planear la primera iteración de la fase de construcción.

- Priorizar procesos (2.2.2.1)

Se vuelve a revalorar los procesos de acuerdo a su importancia y relaciones, es decir, se realizan plantillas que se pueden reutilizar para crear instancias de ella. Se vuelve a contemplar el conjunto de actividades necesarias para cubrir los cambios incrementales de los requerimientos, todo esto son factores principales que influyen en como se diferencia los procesos y nos indica si es viable o factible el sistema a realizar.

Se pueden realizar varias iteraciones similares a la iteración uno hasta que se tengan analizados todos los procesos que comprenden el sistema.

- Riesgos (2.2.2.2)

En esta etapa se realiza una lista detallada de los riesgos y se analiza el grado de probabilidad de poner en peligro el éxito de un proyecto o sistema es mínimo, ya que los riesgos son reducidos debido a que:

- Se evitan riesgos replanificando los cambios y requerimientos finales.
- Limitando los riesgos, es decir, restringirlos de modo que solo afecten a una pequeña parte del proyecto.
- Hacer que el equipo de trabajo controle los riesgos y estar verificando constantemente que no vuelvan aparecer.

- Arquitectura (2.2.2.3)

En la fase de inicio, se desarrolla el análisis de la arquitectura sólo hasta el extremo de determinar que había una arquitectura factible del sistema; ahora en esta fase de elaboración, se tiene que extender el análisis de la arquitectura hasta el extremo de que pueda servir de base a una línea de la arquitectura ejecutable. Con este propósito, el arquitecto realiza una partición inicial (de alto nivel) del sistema, trabajando sobre la vista de la arquitectura del modelo de casos de uso, requerimientos relacionados con ellos y el glosario, dando como resultado el diseño del sistema.

La descripción de la arquitectura contiene una vista de la arquitectura del modelo de diseño que muestra sus artefactos relevantes para la arquitectura.

- La descomposición del modelo de diseño en subsistemas, sus interfaces y las dependencias entre ellos. Esta descomposición es muy significativa para la arquitectura en general, debido a que los subsistemas y sus interfaces construyen la estructura fundamental del sistema.
- Realización de casos de uso-diseño que describan alguna funcionalidad importante y crítica que debe desarrollarse lo mas rápido en el ciclo de vida del software, que impliquen muchas clases de diseño y por tanto tenga una cobertura amplia.

- Planificación de la fase de construcción (2.2.2.4)

En esta etapa se comienza a planificar de forma detallada la primera iteración en la fase de construcción, y se esboza una serie de construcciones donde cada una de ellas se añadirá una pieza relativamente importante, todo esto en términos más generales en las iteraciones restantes. El número de iteraciones necesarias depende del tamaño y la complejidad del proyecto.

En la fase de elaboración se habrá completado sólo porcentajes de cada uno de los modelos donde se refleja el orden de los casos de uso, escenarios restantes y en que completar los modelos. Esto se utiliza por ejemplo en los proyectos grandes, para reducir la planificación temporal total utilizando menos personas y puedan llevar el trabajo de forma paralela.

- **Flujo de trabajo de diseño (2.2.3.0)**

En este flujo de trabajo el equipo de analistas detalla, prioriza y estructura los modelos de casos de uso que conforman la arquitectura de manera que asistan a los desarrolladores en la fase de construcción.

- Detallar, priorizar y estructurar los modelos de caso de uso (2.2.3.1)

- En la priorización se construye sobre el modelo parcial de casos de uso propuesto en la fase de inicio donde se completan los casos de uso y se trabaja en la base de la arquitectura del sistema. Al principio se invierte tiempo en encontrar mas casos de uso, y se priorizan las decisiones asociadas a los riesgos percibidos, por el orden en que se decide seguir el desarrollo, a partir de esto se genera una vista general que incluye la descripción de la arquitectura del sistema.

- El objetivo principal de detallar cada caso de uso es describir los flujos de trabajo de cómo comienza y termina e interactúa con los actores (figura 5.3.1). Con el modelo de casos de uso y los diagramas asociados como punto de comienzo, el especificador de un caso de uso individual puede ya describir cada caso de uso en detalle. Cada especificador de casos de uso debe trabajar estrechamente con los usuarios de los casos de uso para una comprensión mejor y discutir con ellos una propuesta y solicitarles que revisen cuidadosamente la descripción de los casos de uso y así perfeccionar la arquitectura.

- Estructurar el modelo de casos de uso. El analista de sistemas revisa lo que ha hecho y busca similitudes, simplificaciones y oportunidades para mejorar la estructura del modelo de casos de uso. El analista emplea mecanismos generales para lograr un modelo mejor estructurado y más fácil de entender, modificar, ampliar y mantener para reducir la redundancia.

El resultado final es la descripción, diseño y análisis detallada de un caso de uso en particular en forma de texto y diagrama. Ver figura 5.10

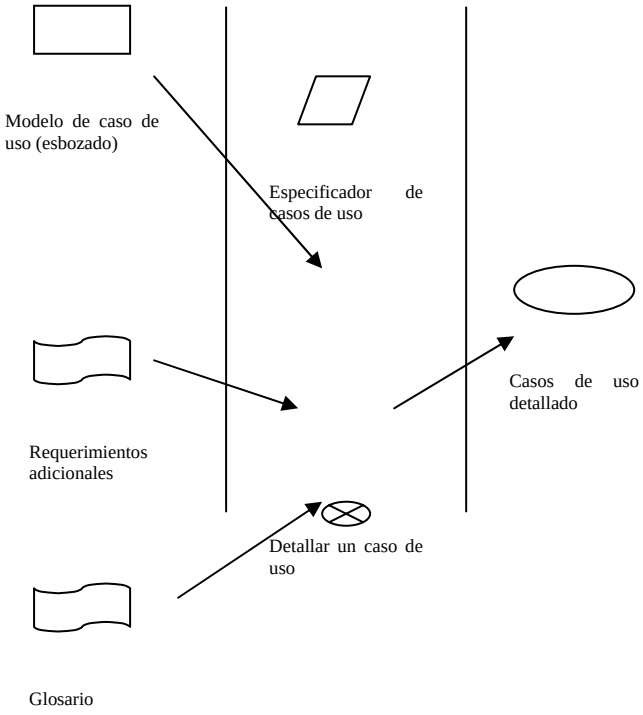


Figura 5.10 Las entradas y los resultados de detallar cada caso de uso. El especificador de casos de uso es un papel que puede ser desempeñado por cualquier elemento del equipo de analistas

- **Resultado de la iteración (2.2.4.0)**

Como resultado de la presente iteración se generan diversos documentos que formalizan el hecho de que el análisis está terminado y listo para aceptar cambios mínimos gracias a una arquitectura robusta y que se ha realizado lo necesario para arrancar la construcción de todos los casos de uso del sistema. Lo anterior permitirá exponer una propuesta económica basada en información real acerca del trabajo a realizar, lo que la hace más confiable para ambas partes.

- Plan de proyecto para construcción (2.2.4.1)

Se entrega un documento que contiene:

- Un modelo completo donde se describe el contexto del sistema.
- Una versión nueva de todos los modelos: casos de uso, análisis, diseño, implementación para que al final de la fase de elaboración están completos en cierto porcentaje, para tener la certeza de que se ha comprendido los requerimientos. La mayoría de los casos de uso han sido comprendidos para asegurar de que no se ha dejado de lado algún punto importante.
- Se propone una base del sistema (arquitectura).
- Una descripción de la arquitectura, incluyendo vistas de los modelos de casos de uso.
- Lista de riesgos actualizada
- Plan de proyecto para las fases de construcción y transición.
- Análisis de negocio completo, incluyendo la propuesta económica.

- Documento integral de análisis (2.2.4.2)

Se integra toda la documentación obtenida al momento siguiendo en formato propuesto para este efecto. El documento presenta las siguientes ventajas:

- Es una fuente común de consulta para usuarios y analistas respecto del estado del proceso de análisis
- Es un documento que registra todos los acuerdos establecidos al momento y sirve como referencia sobre requerimientos, planes de trabajo, actividades, modelos, etc.
- Permite llevar un registro centralizado de todos los documentos derivados del análisis
- No es necesario realizarlo en la iteración en la que se entrega, es un documento que se va integrando durante la labor de análisis.

5.3.3 Fase de construcción (3.0.0.0)

En esta fase el objetivo es crear un sistema con la funcionalidad inicial en el entorno del usuario, para lo cual se requiere realizar una serie de iteraciones que llevan a entregas periódicas de sistemas parciales cada vez más detallados.

- **Iteración 1 (3.1.0.0)**

Esta es la primera de un número indefinido de iteraciones y que depende del trabajo necesario para construir el sistema. El procedimiento de análisis se limita a efectuar una recopilación de requerimientos que se reducen a cambios específicos sobre la arquitectura aprobada en la fase de elaboración y que son de importancia menor.
Esto es, llevar un control de cambios y el estudio de los riesgos y restricciones asociados, actualizar la arquitectura del sistema en caso de que se vea alterada de algún modo, y la priorización necesaria para que se efectúen dichos cambios.

- **Flujo de trabajo de requerimientos (3.1.1.0)**

Este flujo de trabajo se limita a atender los posibles requerimientos de cambios del usuario, los cuales se documentan adecuadamente.

- Control de cambios (3.1.1.1)

Durante la fase de construcción, el jefe de proyecto, el arquitecto y los desarrolladores se aseguraran de establecer la prioridad de los casos de uso, que son agrupados en construcciones e iteraciones y desarrollados según un orden que evita que vuelvas atrás.

- **Flujo de trabajo de análisis (3.1.2.0)**

En el flujo de trabajo de análisis el analista se encarga de hacer un estudio de los requerimientos para determinar lo que implican dichos cambios.

Se actualiza la lista de riesgos y requerimientos, mediante su continuo refinamiento, de forma que refleje los riesgos reales. El objetivo es acabar esta fase con todos los riesgos mitigados. El arquitecto vigilará que la construcción se ajuste a la arquitectura, y si es necesario, modificar dicha arquitectura para incorporar los cambios que surgen durante el proceso de construcción.

- Priorizar procesos (3.1.2.1)

En esta fase el equipo de desarrollo está trabajando en crear el sistema, es por eso que los cambios se deben priorizar de modo que no afecten el trabajo actual de desarrollo, sin embargo se les deberá dar la importancia necesaria para no afectar desarrollos posteriores.

- Riesgos (3.1.2.2)

Se actualiza la lista de riesgos asociados a partir de los nuevos riesgos derivados de los cambios solicitados y de las posibles adaptaciones de la arquitectura a la construcción.

- Arquitectura (3.1.2.3)

La arquitectura del sistema entregada en la fase de elaboración es la necesaria para entrar a la fase de construcción adecuadamente, sin embargo los cambios solicitados por el usuario o el mismo desarrollo del sistema pueden requerir la modificación de la arquitectura. Una arquitectura robusta nos permitirá realizar estos cambios sin mayor problema y el analista sólo tendrá que realizar pequeños ajustes para actualizarla.

- Plan de trabajo (3.1.2.4)

Se actualiza el plan de trabajo con los tiempos y actividades necesarias para efectuar los cambios solicitados

- **Flujo de trabajo de diseño (3.1.3.0)**

En el flujo de trabajo de diseño se agregan los casos de uso modificados a la arquitectura del sistema de modo que siempre se tenga la última versión

- Diseño de casos de uso (3.1.3.1)

Se modelan los casos de uso modificados de acuerdo a los cambios solicitados por los usuarios o a lo que fue necesario modificar debido a la construcción del sistema

- **Resultado final (3.1.4.0)**

El resultado de la iteración es una arquitectura actualizada, adecuada a los cambios efectuados al sistema que se preparó en la fase de elaboración. Asimismo una lista de riesgos, prioridades cambios documentados para posibles aclaraciones o consultas posteriores.

Se obtiene una versión final del documento integral de análisis que sólo se actualiza con la información generada a partir del control de cambios.

5.4 Documentación integral del análisis

El formato que se muestra a continuación es una propuesta para integrar toda la documentación que se recopiló durante el proceso de análisis del sistema y que debe ser entregado al usuario al final de la fase de elaboración, momento en el cual se cuenta con la información necesaria para comenzar adecuadamente la fase de construcción.

PAT-PAS		*<<Nombre del Proyecto>>*	
Información General			
Analista:		**Fecha:** *<<dd-mm-aaaa>>*	**Versión:**
Compañía: *<<Nombre del cliente>>*		**Revisó:** *<<Nombre de quien revisó por parte del cliente / e-mail>>*	**Tel. o Ext.**

Introducción

<Objetivo del documento. Además el preámbulo: descripción de conceptos generales del negocio que tratarán en este documento>

Antecedentes

<Describir brevemente el ambiente actual>

Objetivo General

<Definir que es lo que se quiere lograr>

Necesidades

<Describir la necesidad u oportunidad detectada (Requerimientos)>

Formatos Complementarios

Formato de minutas

Formato de requerimientos

Alcance

<Delimitar el proyecto en base a las necesidades descritas, número de usuarios potenciales y tiempo de uso>

Áreas Involucradas

<Definir las personas o áreas que participarán en el proyecto>

Personas Entrevistadas

<Nombres de las personas que participaron en las entrevistas definiendo los procesos de negocio>

Formatos complementarios

Formato de actividades

.

Formato de descripción del puesto

Cronograma de actividades.

Descripción de los procesos de negocio existentes

<Identificar los procesos de negocio que se verán involucrados en el proyecto>

Descripción de los procesos de negocio propuesto

<Definir los procesos de negocio propuestos para la solución, en caso de haber hecho un análisis preliminar hacer referencia al documento>

Req. 1: detallar la propuesta para el requerimiento solicitado

Req."N": para cada requerimiento se detallara la propuesta de solución

Casos de Uso

<Para definir la secuencia de las acciones que el producto desarrolla, lo cual permite conocer el alcance de las acciones de cada una de las personas que lo utilizan>

- Diagrama General de Casos de Uso

  <Representar el diagrama general de la solución, con los casos de uso de Nivel 0>

- Diagramas Detallados de Casos de Uso

  <Detallar al mínimo nivel de desglose los casos de uso del diagrama general>

Descripción de los Casos de Uso

<En esta sección se describe como los usuarios interactúan con el módulo, esta descripción contiene el objetivo para cada caso de uso, el flujo entre los usuarios y como este caso de uso crea valor para el usuario>

Lista de Funciones

<Definir las funciones que se llevarán a cabo y las horas requeridas para completar cada actividad. El tiempo estimado es en horas y representa el tiempo requerido para codificar, probar e integrar cada una de las actividades>

Nota: pueden surgir algunas características adicionales, las cuales se integrarán al sistema conforme se desarrolla el mismo

Función	***Descripción***	***Tiempo Estimado***

Factibilidad de negocio

<Identificar los beneficios que se obtendrán con el desarrollo del proyecto >

- Beneficios cualitativos
- Beneficios cuantitativos
- Áreas beneficiadas con el sistema

Factibilidad Técnica

<Identificar las herramientas y tecnologías para soportar la solución o la Arquitectura Inicial del Proyecto>

Análisis de Riesgos

<Identificar los puntos críticos cuantificables para que el proyecto se lleve a cabo exitosamente>

Número de Riesgo	*Descripción del riesgo*	*Probabilidad de ocurrencia*	*Responsable*	*Acciones de mitigación*

Factores Clave

<Identificar los puntos críticos no cuantificables para que el proyecto se lleve a cabo exitosamente>

Plan General de Trabajo

<Describirá las etapas del proceso haciendo un estimado general del tiempo de elaboración del producto>

Exclusiones

<Lista de requerimientos que no serán incluidos en esta versión de análisis>

Conclusiones
<*Comentarios y aclaraciones sobre el análisis*>
Conformidad de los Participantes

_____

Vo.Bo. Reglas de Negocio

<<*Nombre Completo VoBo Cliente*>>

<<*Puesto*>>

_____ _____

<<*Nombre*>> <<*Nombre*>>

CONSULTORIA CONSULTORIA

5.5 Formatos propuestos

A continuación se describe el uso de los formatos sugeridos para las diversas actividades del procedimiento. Para mayor referencia acerca de la estructura de cada uno favor de referirse al Anexo A.

- *Minuta*. El formato es llenado por cualquier participante de la reunión que tenga conocimiento de la agenda a tratar. Se emplea para llevar un control de los acuerdos, compromisos y conclusiones en reuniones entre el usuario y el analista. Se debe dar seguimiento a los compromisos generados durante la reunión.

- *Requerimientos*. El formato es para el analista con usuario final ya que lleva el control de los nuevos requerimientos o cambios en ellos. Se basa en el caso de uso a través de un histórico de referencia y el macroproceso. Su objetivo es llevar un control exacto de la definición y cambios a los requerimientos.

- *Actividades*. Empleado por el analista como parte del trabajo realizado en campo. El líder y el responsable del proyecto lo revisan como parte del seguimiento a las actividades programadas o como parte de su labor de trabajo del día.

- *Cronograma.* Se usa para saber a detalle las actividades y el progreso en el tiempo del equipo de analistas y desarrolladores. Lo usa el líder de proyecto y el responsable de negocio como parte de las actividades a realizar con anticipación. Se revisa periódicamente el avance del proyecto. Si hay atraso se deben hacer los ajustes necesarios, justificando el porque y generando otra versión del proyecto

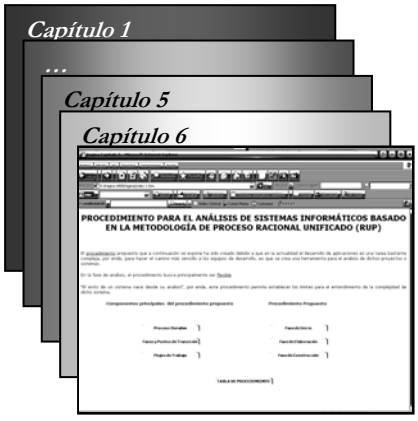
- *Puestos*. Se usa para saber a detalle lo que hace el usuario como parte de su trabajo diario. El resultado obtenido a partir de todos los usuarios entrevistados nos genera un resumen final de puestos

- *Glosario*. El formato se emplea para llevar un control de las palabras clave involucradas en el proyecto y que no son del uso común. Busca que todos los involucrados en el proyecto utilicen el mismo lenguaje evitando malentendidos, confusión o dudas.

- *Matriz de procesos.* El formato es utilizado para documentar el o los puestos que están involucrados en un proceso. Es realizado por el analista como parte de las tareas necesarias para conocer el negocio y de esa forma saber cuanta gente esta involucrada en cada proceso.

CAPÍTULO 6

Casos de Estudio



En este capítulo se llevará a cabo el análisis de dos proyectos reales, siguiendo paso a paso la tabla 5.9, que muestra las actividades que componen al PAS.

En cada caso, la primera iteración de la fase de inicio nos permitirá obtener un planteamiento del proyecto, a fin de establecer los elementos que participarán en él y la manera en que se ha de trabajar a partir de la obtención de los requerimientos globales del usuario.

La segunda iteración nos acerca más al sistema a realizar, proporcionándonos información que permita al equipo de analistas determinar si un sistema que cuente con las características solicitadas (y sus riesgos asociados) hasta ese momento es factible de ser desarrollado.

La tercera iteración busca levantar los requerimientos de los procesos principales o críticos para la arquitectura del sistema, analizarlos para determinar prioridades y eliminar o minimizar los nuevos riesgos que se detectaron.

Al final de estas tres iteraciones tendremos un modelo de la arquitectura del sistema que será empleado durante la fase de elaboración, así como un plan de actividades más definido que dará una visión más clara al usuario y al equipo de desarrollo acerca de la duración del proyecto.

Durante la fase de elaboración los flujos de trabajo de requerimientos y análisis realizan una labor mucho más intensa, ya que se deben obtener los requerimientos necesarios para modelar todos los casos de uso que compondrán el sistema.

En la primera iteración se obtiene a detalle la información (requerimientos) para establecer en su totalidad la arquitectura del sistema. Estos requerimientos se priorizan para facilitar la logística en el desarrollo del sistema.

La segunda iteración busca refinar los requerimientos obtenidos en la primera iteración, en consecuencia se deben actualizar las listas de prioridades, los riesgos y la arquitectura del sistema de modo que podamos planear la fase de construcción adecuadamente.

En la última fase del procedimiento, la de construcción, la labor de recolectar requerimientos y analizarlos se limita a llevar un control de cambios y analizar sus prioridades y riesgos asociados.

El flujo de trabajo de diseño en esta fase se ocupa de modelar los casos de uso obtenidos en los flujos de requerimientos y análisis.

Al completar el presente procedimiento se deberá contar con los elementos necesarios para que el equipo de desarrollo aplique técnicas típicas de desarrollo y se termine exitosamente el sistema.

El presente capítulo busca aplicar el procedimiento objeto de este trabajo de tesis a un proyecto real de desarrollo de sistemas informáticos, con el fin de comprobar la validez del mismo.

Como se explicó en el capítulo anterior, el procedimiento busca conocer el sistema en forma de mini proyectos, de modo que cada iteración el equipo de desarrollo detalla más su concepción del sistema.

Una de las actividades que requieren más cuidado, dedicación y por lo tanto tiempo es el análisis de los requerimientos de los usuarios, para modelarlos en forma de casos de uso tanto para procesos críticos como no críticos.

El objetivo del presente capítulo es mostrar en forma real y concisa el resultado de todas las actividades que se muestran en la tabla principal del PAS, es por eso que sólo se representarán y modelarán algunos ejemplos de casos de uso propios del sistema que se tratará en este capítulo, sin embargo se considera de poca utilidad mostrarlos todos puesto que no sería de mayor aportación al mismo.

A continuación mostramos los resultados obtenidos por el equipo de analistas, los cuales siguen fielmente el PAS.

6.1 Primer caso de estudio

En este proyecto fue requerida una página de Internet que muestre de manera interactiva el procedimiento de análisis de sistema PAS.

Dicha página debe mostrar toda la información disponible sobre el PAS de forma amigable y simple al usuario.

A continuación se muestran los resultados de las actividades realizadas para este caso de estudio siguiendo el orden indicado por el PAS. Debe notarse que debido a que el sistema es sencillo, algunas actividades se pueden cubrir totalmente en una sola iteración.

6.1.1 Fase de inicio (1.0.0.0)

El objetivo general de esta fase es poner en marcha el análisis, es decir, desarrollar el análisis del negocio hasta el punto en el cual es posible justificar el arranque del proyecto.

- **Iteración 1 (1.1.0.0)**

En esta iteración buscamos un acercamiento con la organización que efectúa la solicitud del sistema para conocerla, asignar un nombre al proyecto y se dan acuerdos para el manejo del proyecto. Asimismo, definir requerimientos globales del sistema y requerimientos especiales o restricciones con que se debe cubrir el sistema.

- **Flujo de trabajo de requerimientos (1.1.1.0)**

En este flujo de trabajo identificamos y detallamos los casos de uso pertinentes, se incluyen los siguientes aspectos o actividades:

- Acercamiento (1.1.1.1)

En esta actividad el usuario plantea por primera vez su inquietud sobre un sistema que presente de forma interactiva el capítulo 5 de la Tesis "Procedimiento para el Análisis de Sistemas Informáticos basado en la Metodología de Proceso Racional Unificado (RUP)" y de acuerdo a las pláticas sostenidas entre ambas empresas se plantearon los siguientes alcances globales:

- Proyecto PAS - PAT
  PAS = Nombre de la Consultoría
  PAT = Nombre de la Empresa usuaria

- Presentación del equipo del trabajo del cliente
  Nombre de la Empresa: Programa de Apoyo a la Titulación (PAT)

Objetivo:

Crear una página WEB en la que se visualice el Procedimiento de Análisis de Sistemas (PAS), desarrollado en la Tesis "Procedimiento para el Análisis de Sistemas Informáticos basado en la Metodología de Proceso Racional Unificado (RUP)

- Lista de responsables del proyecto:

  Responsables Cliente:
  Gloria Mata Hernández

  Responsables Consultoría:
  Roberto Caballero
  Nadia Zardaneta
  Lidia Resendiz
  Pablo Gutiérrez
  Arturo Rojas

- Acuerdos para el manejo del proyecto:

  Toma de minutas. Formato F01
  Toma de requerimientos. Formato F02
  Horarios de trabajo. Lunes a Viernes de 9 A.M. a 6 P.M.
  Reportes de actividades semanales del equipo de proyecto. Formato F03
  Glosario de términos. Formato F06
  Ningún acuerdo verbales es tomado en cuenta

- Alcance global (1.1.1.2)

En esta actividad del Flujo de Trabajo de Requerimientos de la iteración uno, el usuario proporciono un panorama general de la funcionalidad que debe tener el sistema sin entrar a los detalles.

Para efecto del sistema planteado, los módulos que se deben incluir son:

Creación de una página WEB donde se muestre en que consiste el PAS

Lugares donde debe ser instalado:

  - Debe instalarse en un sitio de Internet (por definir)

  - Es una aplicación cliente/servidor, por ende, cualquier usuario con interés sobre el tema podrá acceder a ella desde su PC en su lugar de trabajo

Tiempo estimado sugerido: 4 semanas

- Restricciones (1.1.1.3)

Debemos considerar que las restricciones son requerimientos especiales, en otras palabras, son características del sistema que el cliente necesita y que no pueden ser modificadas. Para el Sistema PAS – PAT, las restricciones planteadas son:

  - Contar con soporte de información a dudas (vía e-mail) y ayuda en sitio (botón con información de uso)

  - Que los usuarios puedan usar cualquier navegador para acceder a la página (usar HTML estándar)

  - Sin seguridad a nivel usuario, es decir, cualquier usuario con interés en el tema podrá acceder a la página a consultar información

- **Flujo de trabajo de análisis (1.1.2.0)**

Tomando en cuenta la información obtenida durante la recolección de requerimientos anterior se debe realizar un primer análisis de riesgos y hacer una propuesta del tiempo aproximado que se usará para la realización del proyecto.

- Riesgos (1.1.2.1)

El riesgo es un factor, cosa o elemento que constituye un peligro cuyo grado es incierto, se puede ver como un asunto que tiene cierto grado de probabilidad de poner en peligro el éxito del proyecto. En el Sistema PAS – PAT, tenemos lo siguiente:

  - El único riesgo que se tiene, es que el servidor de aplicaciones de Internet donde sea montado el sistema en algún momento se pueda caer

- Proyección del proyecto (1.1.2.2)

En esta actividad se hizo un análisis de las características que debe cumplir el sistema y se determino una fecha estimada de entrega del proyecto, debemos aclarar que la fecha es tentativa de acuerdo a los requerimientos con que se cuenta hasta el momento.

De acuerdo a las reuniones con el cliente y a sus prioridades de módulos se tiene la proyección donde se propone el concepto a realizar y los tiempos de cada uno de ellos. Ver anexo B opción 1 “Proyección del Proyecto”

- **Resultado de la iteración (1.1.4.0)**

El resultado de esta iteración es una serie de documentos (minutas y requerimientos) que resumen el planteamiento general del proyecto.

- Planteamiento del proyecto (1.1.4.1)

  - Como primer acercamiento las características principales del módulo que el negocio requiere están ya definidas

  - El tiempo estimado en que el negocio requiere que este terminado el proyecto esta definido en cuatro semanas

  - Los lugares donde se va a instalar el sistema están ya considerados

  - Dado el tamaño del sistema, el número de módulos a revisar esta definido en uno sólo y esta es la misma página WEB que engloba el procedimiento.

  - En esta etapa del análisis, para llevar un registro de las reuniones sostenidas, plasmamos todos los puntos tratados de la agenda, la evolución de estos puntos, los acuerdos logrados, los compromisos a cumplir, los focos de atención, etc., en un formato “Minuta de Reuniones”. El control lo llevamos con un “No Minuta”, el cual para efectos de este primer formato será el “PG032905” (PG son las iniciales del que lo llena y 032905 es la fecha de la reunión), este formato se encuentra al final de esta documentación en el anexo B, opción 2 “Minutas”.

- **Iteración 2 (1.2.0.0)**

En esta iteración se obtienen características generales de cada uno de los procesos que componen el sistema, información útil para realizar un análisis de cómo están organizados los procesos que forman el macroproceso de la empresa, se hace un análisis de factibilidad y un plan de proyecto en donde se especifiquen en forma general las actividades y los tiempos de cada proceso.

- **Flujo de trabajo de requerimientos (1.2.1.0)**

Para obtener los requerimientos en esta fase se deben completar adecuadamente las siguientes actividades:

- Inducción (1.2.1.1)

Nuestro equipo de analistas realizo una investigación de campo en las instalaciones del cliente para conocer en forma práctica pero general sus procesos, de ese modo se inicia la asociación procesos contra puestos y sus relaciones.

Las actividades de inducción realizadas se registran en un Formato de Actividades. En este formato se plasman a detalle cada una de las actividades realizadas por el usuario, de tal manera que podamos desmenuzar el proceso de negocio, paso a paso como una receta de cocina. Para su control, este formato se encuentra al final en el anexo B opción 3 “Formato de Actividades”, con número de control 001.

- **Flujo de trabajo de análisis (1.2.2.0)**

En este flujo de trabajo se comienza a determinar una arquitectura candidata conformada por los casos de uso críticos del sistema, además de priorizar procesos, establecer restricciones y afinar la lista de riesgos.

- Macroproceso (1.2.2.1)

El macroproceso indica el intercambio general de información, las interacciones entre los procesos más generales que realiza el usuario y que deberá incluir el sistema o que estarán interactuando con él.

Para el Sistema PAS – PAT, el macroproceso esta formado por tres procesos: Proceso Iterativo, Fases y Flujos de Trabajo, estos se detallan en la siguiente figura.

Macroproceso de página WEB

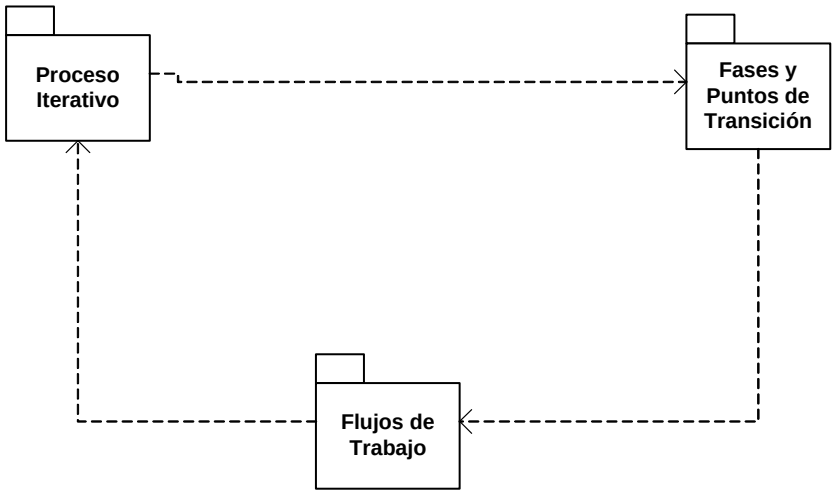


Figura 6.1 Procesos que forman en Macroproceso de la página WEB

- Matriz de procesos propuestos (1.2.2.2)

Es una matriz que se crea con los conocimientos adquiridos en la inducción, incluye todos los procesos principales y generales que los componen, de tal manera, que se relacionen contra un puesto (no una persona en particular, sino un papel dentro de la organización).

En esta fase, no aplica un Matriz de procesos, debido al tamaño del sistema que se ha proyectado. El único usuario es aquel que desea realizar una consulta del PAS y su proceso esta detallada en un formato de actividades (ver 1.2.1.1)

- **Flujo de trabajo de análisis (1.2.2.0)**

- Relaciones entre procesos (1.2.1.3)

La priorización del usuario es el módulo integral, el cual debe de mostrar el PAS en una página WEB que pueda ser visualizada vía Internet.

- Restricciones (1.2.1.4)

Las restricciones no cambian, se mantienen igual que al inicio.

- Riesgos (1.2.1.5)

No se detectaron nuevos riesgos, se mantienen los originales.

- Análisis de factibilidad (1.2.1.5)

El estudio de factibilidad consistirá en ponderar las potenciales capacidades del equipo de desarrollo para cubrir los requerimientos del usuario en los tiempos tentativos propuestos bajo los riesgos y restricciones detectados hasta este momento.

El estudio de factibilidad lo plasmamos en un "Formato de Factibilidad", el cual se encuentra al final de esta documentación en el anexo B en la opción 4 "Análisis de Factibilidad".

- **Resultado de la iteración (1.2.4.0)**

Se obtiene documentación donde se especifica una primera versión del cronograma de procesos, sus tiempos de finalización, una lista de procesos críticos para el proyecto y sus riesgos asociados de forma más específica y actualizada. Para obtener este resultado, nuestro equipo de analistas se baso en las plantillas de actividades, cronogramas y matriz de procesos propuestos.

- Plan de actividades (1.2.4.1)

Derivado de una planeación de las actividades, creamos un "Cronograma de Actividades", el cual se encuentra al final de esta documentación en el Anexo de Formatos en la sección Cronograma de Actividades, con número de revisión 001.

  - En esta etapa del análisis, para llevar un registro de las reuniones sostenidas, plasmamos todos los puntos tratados de la agenda, la evolución de estos puntos, los acuerdos logrados, los compromisos a cumplir, los focos de atención, etc., en un formato "Minuta de Reuniones". El control lo llevamos con un "No Minuta", el cual para efectos de este segundo formato será el "PG040505" (PG son las iniciales del que lo llena y 040505 es la fecha de la reunión), este formato se encuentra al final de esta documentación en el anexo B, opción 2 "Minutas".

- **Iteración 3 (1.3.0.0)**

En esta iteración se obtienen los requerimientos críticos del sistema tomando en cuenta el listado generado en el análisis de la iteración anterior.

- **Flujo de trabajo de requerimientos (1.3.1.0)**

En este flujo de trabajo se obtienen a detalle los requerimientos necesarios para el modelado de los procesos críticos del sistema a través de entrevistas y cuestionarios realizados a los usuarios involucrados.

- Levantamiento de requerimientos (1.3.1.1)

En esta etapa formalizamos el Levantamiento de los Requerimientos para la pantalla principal del sistema con un formato del mismo nombre, con folio PASPAT002, el cual para su control se encuentra al final de esta documentación en el Anexo B opción 5 "Levantamiento de Requerimientos".

- **Flujo de Trabajo de Análisis (1.3.2.0)**

En este flujo de trabajo nuestros analistas se encargaron de refinar la lista de procesos críticos a partir de la nueva información obtenida y se modelaron los casos de uso y sus relaciones. Finalmente, actualizamos la lista de riesgos asociados a los nuevos requerimientos y comienza a determinar soluciones para eliminarlos o minimizarlos.

- Casos de uso críticos (1.3.2.1)

Nuestro equipo de analistas ha obtenido a partir del levantamiento de requerimientos información esencial que nos permitió refinar la arquitectura candidata, agregando a la versión actual procesos que en su momento se habían considerado de menor importancia, pero que por las nuevas relaciones detectadas con otros procesos, por la logística de desarrollo o por los riesgos que están asociados a ese proceso, deberían ser considerados como críticos.

El sistema es muy sencillo, sólo existe un actor y éste es el usuario del procedimiento, el cual puede ser cualquier persona con el interés por ingresar a la página WEB en que se describe el procedimiento.

Con base en la abstracción global del sistema tenemos que el caso de uso que se identifico es por si solo crítico y se ataca para su correcto análisis y desarrollo.

En el diseño del modelado es a partir de un solo caso de uso: “Consultar página WEB de Procedimiento de Análisis Propuesto”



Modelado del Caso de uso

- **Resultado de la Iteración (1.3.4.0)**

Como resultado final se obtiene un documento el cual representa el plan de actividades actualizado y una primera versión del sistema.

- Plan de actividades actualizado (1.3.4.1)

Debido a que el sistema es pequeño, el resultado obtenido es que todas las actividades programadas se llevaron a cabo y las iteraciones siguientes son repetitivas, por lo tanto, no es necesario seguir iterando, ya que caeríamos en un ciclo repetitivo.

- En esta etapa del análisis continuamos con el registro de las reuniones sostenidas, plasmamos todos los puntos tratados de la agenda, la evolución de estos puntos, los acuerdos logrados, los compromisos a cumplir, los focos de atención, etc., en un formato “Minuta de Reuniones”. El control lo llevamos con un “No Minuta”, el cual para efectos de este tercer formato será el “PG041205” (PG son las iniciales del que lo llena y 041205 es la fecha de la reunión), este formato se encuentra al final de esta documentación en el anexo B, opción 2 “Minutas”.

- Dando continuidad al avance del análisis, registramos una más de las reuniones sostenidas y plasmamos todos los puntos tratados de la agenda, la evolución de estos puntos, los acuerdos logrados, los compromisos a cumplir, los focos de atención, etc., en un formato "Minuta de Reuniones". El control lo llevamos con un "No Minuta", el cual para efectos de este primer formato será el "PG041905" (PG son las iniciales del que lo llena y 041905 es la fecha de la reunión), este formato se encuentra al final de esta documentación en el anexo B, opción 2 "Minutas"..

- Finalmente, con todo lo descrito hasta este punto elaboramos el formato Integral de Análisis.

6.1.2 Formato integral de análisis

PAS-PAT

Información General

Analista:	Pablo Gutiérrez (PG)	*Fecha:* 19/04/2005	Versión: 1.0
Compañía: Programa de Apoyo a la Titulación (PAT)		Revisó: Ing. Gloria Mata Hernández	Tel. o Ext.

Antecedentes

- Se tiene un documento de tesis, el cual desea ser mostrado de forma que genere interés por su consulta y la aplicación de su contenido.

- Actualmente cualquier individuo interesado realiza una consulta del Procedimiento de Análisis de Sistemas "PAS" directamente en el documento de Tesis.

Objetivo General

- Crear una página WEB en la que se visualice el Procedimiento de Análisis de Sistemas (PAS), desarrollado en la Tesis "Procedimiento para el Análisis de Sistemas Informáticos basado en la Metodología de Proceso Racional Unificado (RUP)

Necesidades

- La prioridad del cliente (Ing. Gloria Mata H.), es ver de forma interactiva el PAS, para ello solicitó sea creada una página WEB donde pueda visualizar la información del capítulo 5 de la presente tesis pero de forma gráfica, a modo de enlaces para cada uno de los títulos que se contemplan en el procedimiento.

Formato de minutas, requerimientos:

- Ver 1.2.2.2 Fase de inicio de la iteración 2 del proceso Matriz de procesos propuestos

Alcance

- Ver 1.2.2.1 Fase de inicio de Iteración 2 del flujo Requerimientos en la opción Macro proceso

Áreas Involucradas

Responsables Cliente:

- Ing. Gloria Mata Hernández

Responsables consultoría:

- Roberto Caballero
- Nadia Zardaneta
- Lidia Resendiz
- Pablo Gutiérrez
- Arturo Rojas

Personas Entrevistadas

- Ing. Gloria Mata Hernández (Director de la Tesis)

Formato de actividades, descripción del puesto, cronograma de actividades.

- Ver 1.3.1.1) Fase de inicio de la iteración 3 de la análisis opción Levantamiento de requerimientos

Descripción de los procesos de negocio existentes

- Ver 1.3.1.2 Fase de inicio de la iteración 3 del análisis opción Casos de uso propuestas críticos

Descripción de los procesos de negocio propuesto

Se realizaron los siguientes diagramas y formatos a partir del Macroproceso:

- Formato de actividades del proceso del usuario que realiza la consulta
- Diagrama de caso de uso del usuario que realiza la consulta

Casos de Uso

- Ver descripción de caso de uso opción (1.3.1.1)

Factibilidad Técnica

Uso de tecnología de 3 capas (Interfaz, Negocio y Datos)

Análisis de Riesgos

- El único riesgo que se tiene, es que el servidor web donde se aloje el sitio deje de operar adecuadamente en algún momento

Plan General de Trabajo

PAS-PAT

Fecha: **04/04/2005**

Nombre del proyecto: **PAS - PAT**

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Elaboración de las pagínas principales	1 día	ma 05/04/05	ma 05/04/05	
2	Elaboración de páginas secundarias	5 días	mi 06/04/05	ma 12/04/05	1
3	Enlace de las paginas principales con las secundarias	7 días	mi 13/04/05	ju 21/04/05	2
4	Formato de la pagina	2 días	vi 22/04/05	lu 25/04/05	3

Conclusiones

El PAS se contempla como una técnica que puede ser ajustada y enriquecida a partir de la experiencia personal de quien lo adopte como herramienta de trabajo en sus proyectos de desarrollo.

En la medida que esta primera versión del procedimiento sea perfeccionada y documentada, irá adquiriendo solidez y aceptación.

Conformidad de los Participantes

Vo.Bo. Reglas de Negocio
<<Nombre Completo VoBo Cliente>>
<<Puesto>>

<<Nombre>>
CONSULTORIA

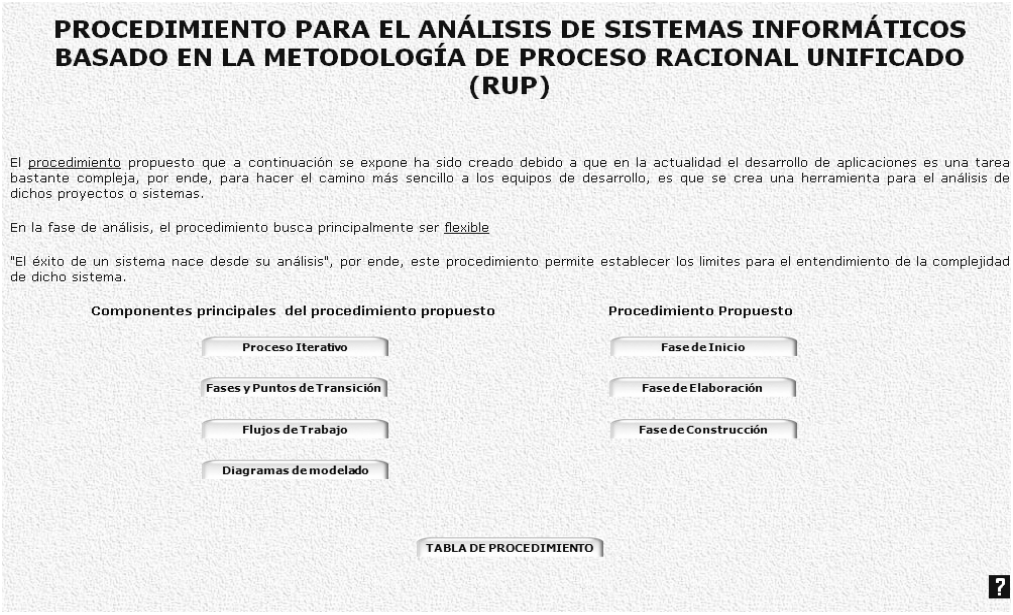
<<Nombre>>
CONSULTORIA

6.1.3 Sistema PAS – PAT Versión 1.0

Como resultado del análisis hecho por nuestro equipo de analistas, describimos el sistema web obtenido mostrando pantallas descriptivas.

En esta imagen se puede observar la pantalla de entrada al Sistema PAS – PAT y en ella encontramos:

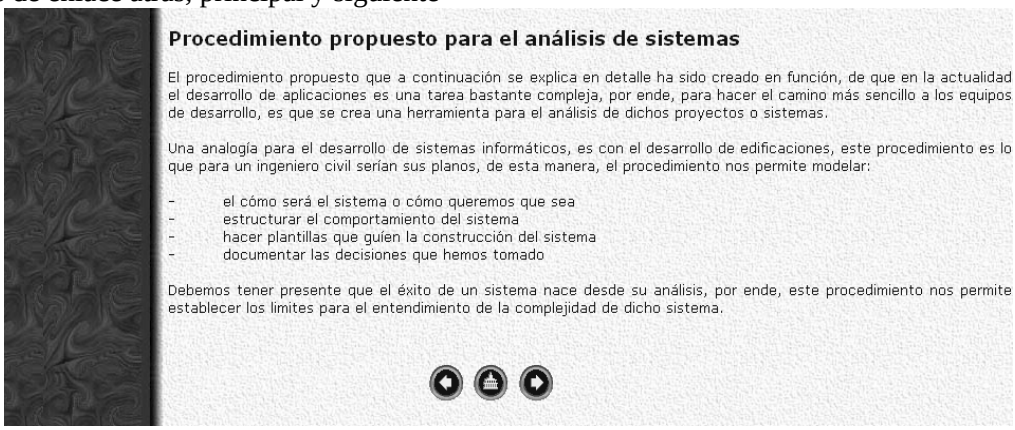
- Título referente al contenido de la página
- Texto descriptivo y objetivo de la información contenida en la página
- Botones de enlace a los componentes y procedimiento propuesto
- Botón de enlace a la tabla principal del PAS
- Botón de enlace a la ayuda del sistema



Pantalla principal de la página WEB que aloja el Sistema PAS - PAT

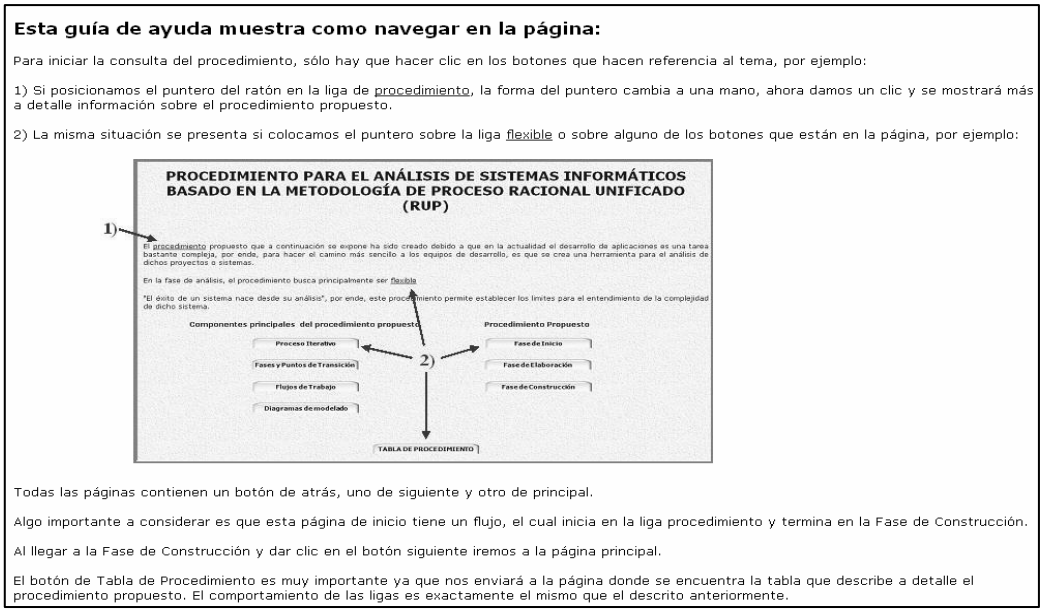
Una vez que iniciamos las llamadas al procedimiento por medio de los botones de enlace, la página muestra una serie de pantallas como la siguiente, en ellas podemos ver diversos contenidos referentes al PAS:

- Texto descriptivo de cada paso o etapa del procedimiento
- Botones de enlace atrás, principal y siguiente



Pantalla de enlace desde la página principal del Sistema PAS - PAT

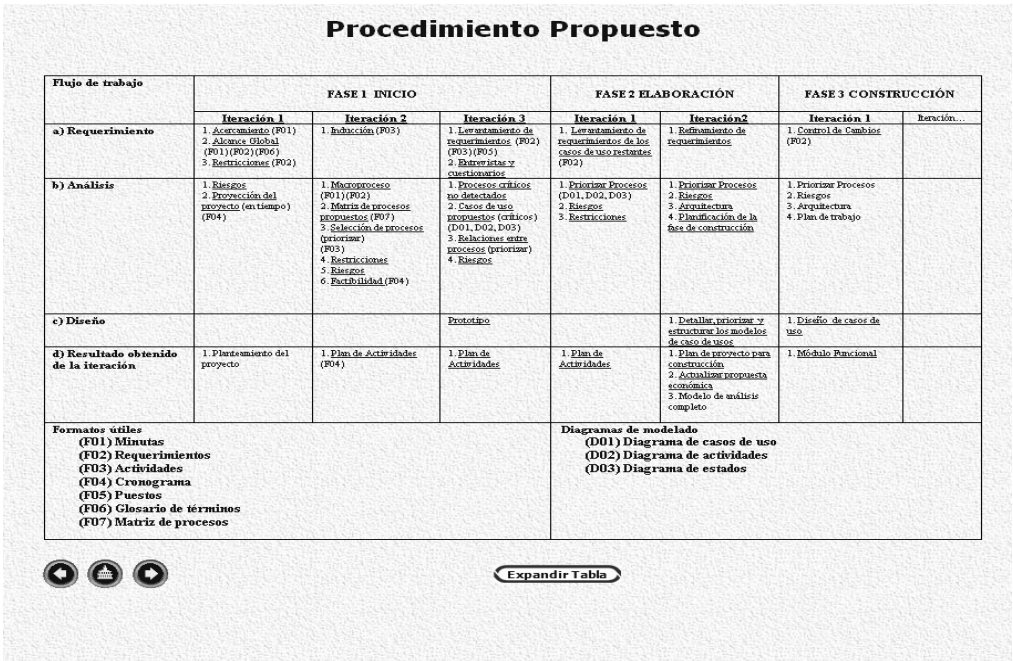
Al regresar a la pantalla principal de la página del Sistema PAS – PAT, encontramos un botón de ayuda, el cual nos enlaza a la pantalla correspondiente; una vez en la pantalla se proporciona una guía de ayuda que orienta al usuario la forma de navegar por el sistema.



Pantalla de ayuda del Sistema PAS - PAT

De igual manera, en la pantalla principal del Sistema PAS – PAT, encontramos un botón de "Tabla de Procedimiento", el cual al dar un clic sobre él nos enlaza al corazón del PAS y del sistema mismo. En esta pantalla encontramos:

- Tabla de Procedimiento diseñada estratégicamente para visualizar de forma clara y sencilla el PAS; al dar un clic sobre cada uno de los pasos o etapas el sistema nos enlaza a la descripción del elemento del procedimiento
- Enlaces a los Formatos y Diagramas utilizados por el PAS para documentar el análisis
- Botones de enlace atrás, principal y adelante
- Botón de Expandir Tabla, para visualizar de forma más a detalle la tabla



Flujo de trabajo	FASE 1 INICIO			FASE 2 ELABORACIÓN		FASE 3 CONSTRUCCIÓN	
	Iteración 1	Iteración 2	Iteración 3	Iteración 1	Iteración2	Iteración 1	Iteración...
a) Requerimiento	1. Acercamiento (F01) 2. Alcance Global (F01) (F02) (F06) 3. Restricciones (F02)	1. Inducción (F03)	1. Levantamiento de requerimientos (F02) (F03) (F05) 2. Entrevistas y cuestionarios	1. Levantamiento de requerimientos de los casos de uso restantes (F02)	1. Refinamiento de requerimientos	1. Control de Cambios (F02)	
b) Análisis	1. Riesgos 2. Proyección del proyecto (en tiempo) (F04)	1. Macroproceso (F01) (F02) 2. Matriz de procesos propuestos (F07) 3. Selección de procesos (priorizar) (F03) 4. Restricciones 5. Riesgos 6. Factibilidad (F04)	1. Procesos críticos no detectados 2. Casos de uso propuestos (críticos) (D01, D02, D03) 3. Relaciones entre procesos (priorizar) 4. Riesgos	1. Priorizar Procesos (D01, D02, D03) 2. Riesgos 3. Restricciones	1. Priorizar Procesos 2. Riesgos 3. Arquitectura 4. Planificación de la fase de construcción	1. Priorizar Procesos 2. Riesgos 3. Arquitectura 4. Plan de trabajo	
c) Diseño			Prototipo		1. Detallar, priorizar y estructurar los modelos de caso de usos	1. Diseño de casos de uso	
d) Resultado obtenido de la iteración	1. Planteamiento del proyecto	1. Plan de Actividades (F04)	1. Plan de Actividades	1. Plan de Actividades	1. Plan de proyecto para construcción 2. Actualizar propuesta económica 3. Modelo de análisis completo	1. Módulo Funcional	
Formatos útiles (F01) Minutas (F02) Requerimientos (F03) Actividades (F04) Cronograma (F05) Puestos (F06) Glosario de términos (F07) Matriz de procesos				Diagramas de modelado (D01) Diagrama de casos de uso (D02) Diagrama de actividades (D03) Diagrama de estados			

Pantalla de la Tabla de Procedimiento del Sistema PAS - PAT

Situados en la Tabla de Procedimiento, encontramos el botón “Expandir Tabla”, al dar clic sobre este botón, el sistema muestra la misma Tabla de Procedimiento, con la ventaja de que esta se ve ampliada para facilitar la navegación. La pantalla contiene:

- Tabla de Procedimiento diseñada estratégicamente para visualizar de forma clara y sencilla el PAS, al dar un clic sobre cada uno de los pasos o etapas el sistema nos enlaza a la descripción del procedimiento
- Enlaces a los Formatos y Diagramas utilizados por el PAS para documentar el análisis
- Botón en cada extremo de enlace a la página principal del Sistema PAS - PAT
- Botón en cada extremo de “Restaurar Tabla”
- Barras de desplazamiento horizontal y vertical, que facilitan la navegación


Restaurar Tabla

Procedimiento Propuesto

Flujo de trabajo	FASE 1 INICIO			FASE 2 ELABORACIÓN	
	Iteración 1	**Iteración 2**	**Iteración 3**	**Iteración 1**	**Iteración2**
a) Requerimiento	1. Acercamiento (F01) 2. Alcance Global (F01) (F02) (F06) 3. Restricciones (F02)	1. Inducción (F03)	1. Levantamiento de requerimientos (F02) (F03) (F05) 2. Entrevistas y cuestionarios	1. Levantamiento de requerimientos de los casos de uso restantes (F02)	1. Refinamiento de requerimientos
b) Análisis	1. Riesgos 2. Proyección del proyecto (en tiempo) (F04)	1. Macroproceso (F01) (F02) 2. Matriz de procesos propuestos (F07) 3. Selección de procesos (priorizar) (F03) 4. Restricciones 5. Riesgos 6. Factibilidad (F04)	1. Procesos críticos no detectados 2. Casos de uso propuestos (críticos) (D01, D02, D03) 3. Relaciones entre procesos (priorizar) 4. Riesgos	1. Priorizar Procesos (D01, D02, D03) 2. Riesgos 3. Restricciones	1. Priorizar Procesos 2. Riesgos 3. Arquitectura 4. Planificación de la fase de construcción
c) Diseño			Prototipo		1. Detallar, priorizar y estructurar los modelos de caso de usos
d) Resultado obtenido de la iteración	1. Planteamiento del proyecto	1. Plan de Actividades (F04)	1. Plan de Actividades	1. Plan de Actividades	1. Plan de proyecto para construcción 2. Actualizar propuesta económica 3. Modelo de análisis completo
Formatos útiles **(F01) Minutas** **(F02) Requerimientos** **(F03) Actividades** **(F04) Cronograma** **(F05) Puestos** **(F06) Glosario de términos** **(F07) Matriz de procesos**				**Formatos útiles** **(D01) Diagrama de casos de uso** **(D02) Diagrama de actividades** **(D03) Diagrama de estados**	

Pantalla de la Tabla de Procedimiento del Sistema PAS – PAT (expandida)

Esta es la versión 1.0 del Sistema PAS –PAT, conforme se avance en las Fases de Elaboración y Construcción, se obtendrá un mejor sistema, ya que cada iteración de estas fases, proporcionará más detalle sobre el mismo.

6.2 Segundo caso de estudio

Este segundo proyecto busca analizar la adaptación de un software existente a las necesidades de una ferretería, la cual requiere llevar un control de su empresa a través de este sistema.
Se deberán cubrir actividades de registro de pedidos, cuentas por cobrar, facturación e inventarios, entre otras.
Debido al tiempo necesario para analizar en su totalidad y de forma adecuada un sistema complejo, el equipo de trabajo enfocará sus esfuerzos al análisis de algunos casos de uso y su modelado, con el objetivo de mostrar que al seguir el PAS durante las iteraciones necesarias, se alcanzaría el objetivo de analizar y modelar por entero el sistema requerido.

A continuación se muestran los resultados de las actividades realizadas para este caso de estudio siguiendo el orden indicado por el PAS

6.2.1 Fase de inicio (1.0.0.0)

- **Iteración 1 (1.1.0.0)**

- **Flujo de trabajo de requerimientos (1.1.1.0)**

    - Acercamiento (1.1.1.1)

En esta actividad se tuvo un acercamiento con la organización con el fin de recopilar información general de la empresa obteniendo los siguientes datos:

- Proyecto: Frog-Cravioto

- Presentación del equipo del trabajo del cliente.

  Nombre de la Empresa: Ferretería Cravioto
  Dirección: Alhondiga 15, Col. Centro
  Teléfono: 51 30 27 30
  Objetivo: Ser la mejor empresa ferretera de distribución a nivel nacional incorporando innovación tecnológica para tener las mejores prácticas del negocio.
  Lista de lugares y responsables donde se realizara el proyecto
  Responsables Cliente:

  - Enrique Castañeda
  - Concepción Hernández
  - Patricio Villegas

  Responsables consultoría:

  - Roberto Caballero
  - Nadia Zardaneta
  - Lidia Resendiz
  - Pablo Gutiérrez
  - Arturo Rojas

- Acuerdos para el manejo del proyecto
  - Toma de minutas. Formato F01
  - Toma de requerimientos. Formato F02
  - Horarios de trabajo. Lunes a Viernes de 9 A.M. a 6 P.M.
  - Reportes de actividades semanales del equipo de proyecto. Formato F03
  - Se elaborará un glosario de términos que se incluirá en el Formato Integral de Análisis.
  - Ningún acuerdo verbales es tomado en cuenta.

- Alcance global (1.1.1.2)

En esta actividad se debe obtener la funcionalidad general del sistema. La información obtenida es la siguiente:

Módulo	Descripción general
1. Pedidos mostrador	• Poder capturar un pedido contado o crédito a un cliente con un vendedor de mostrador
2. Facturación	• Realizar la facturación del proceso de venta autoservicio, tele-marketing y mostrador.
3. Inventarios	• Control de entradas y salidas de sub-almacenes y almacenes.
4. Pedidos telemarketing	• Control de llamadas para atención a clientes y sus respectivos pedidos asignados
5. Compras	• Comprar eficientemente a través de algoritmo de sugerencia de compra
6. Cuentas por pagar	• Realizar proyección de pagos en tiempo
7. Tesorería	• Conciliación y transferencia bancaria así como emisión de cheques
8. Autoservicio	• Venta directa del cliente, emisión de listas de precios en código de barras
9. Contabilidad	• Pólizas automáticas de los movimientos diarios de la operación. Estados financieros, Balanzas de comprobación.
10. Sistema de ventas por Internet	• El cliente podrá realizar su pedido o el sistema hacer un sugerido con base a su historial, emitir su estado de cuenta.
11. Sistema de ventas de vendedor portafolio	• Los vendedores locales y foráneos realizaran los pedidos en su computadora portátil. Donde tendrán la información necesaria para dar mejor servicio al cliente
12. Manejo de cómputo móvil para venta autoservicio	• Los clientes realizaran su pedido a través de un cómputo móvil para dar un mejor servicio.

Módulo	Descripción general
13. Cartografía	• Los pedidos de telemarketing se visualizarán en el mapa asignando el nombre del cliente. Así como la impresión de la ruta de entrega de pedidos
14. Ingresos	• Control de los dineros producto de la venta de autoservicio, pedidos y cobro de cuentas por cobrar
15. Cuentas por cobrar	• Control de la cartera de clientes, estados de cuenta, análisis de vencimientos

Lugares donde debe ser instalado:

- Toluca centro de distribución
- Ecatepec centro de distribución
- Tienda principal en centro de la ciudad de México
- 2 Tiendas autoservicio en centro de la ciudad de México
- Tienda Cobrecentro en centro de la ciudad de México

Tiempo estimado sugerido: 1 año.

- Requerimientos especiales o restricciones (1.1.1.3)

Las restricciones y requerimientos especiales señalados y detectados al momento son:

- El usuario esta dispuesto a comprar el hardware necesario para que la aplicación funcione con base de Datos: SQL Server 2000.
- Se desea que el sistema funcione en tecnología 3 capas (capa de presentación, capa regla de negocios, capa de datos).
- Contar con soporte para la infraestructura para cómputo móvil y la red de todos los centros de distribución.
- Los clientes deben usar versiones de sistema operativo Windows 2000 y posteriores.
- Seguridad a nivel usuario y a nivel base de datos
- El sistema debe operar en un esquema 24X7

- **Flujo de trabajo de análisis (1.1.2.0)**

  - Riesgos (1.1.2.1)

Los riesgos detectados a partir de la información obtenida en el flujo de trabajo de requerimientos son:

- Si se tienen enlaces de comunicaciones poco poderosos, el desempeño del sistema puede ser inaceptable.
- El equipo elegido puede no soportar el funcionamiento de la base de datos y se sistema operará lento.
- El sistema puede no funcionar igual en las diversas versiones de sistema operativo de los clientes. Puede no contarse con las últimas versiones de navegador que ofrezcan más seguridad en la integridad de los datos.
- El sistema puede caerse en algún momento, es posible que el usuario no tenga en cuenta adquirir el equipo necesario para contar con una infraestructura de respaldo.

- Proyección del proyecto (1.1.2.2)

De acuerdo a las reuniones con el cliente y a sus prioridades de módulos se tiene la proyección donde se propone el concepto a realizar y los tiempos de cada uno de ellos.

El formato correspondiente se puede consultar en el anexo C opción 1, “Proyección del Proyecto”

- **Resultado de la iteración (1.1.4.0)**

Resumiendo la información obtenida en esta iteración

- Los equipos de trabajo están definidos por ambas partes el negocio y la empresa consultora.
- Como primer acercamiento las características principales de los módulos que el negocio requiere están ya definidas.
- El tiempo estimado en que el negocio requiere que este terminado el proyecto esta definido en un año
- Los lugares donde se va a instalar el sistema están ya considerados.
- Las restricciones y riesgos globales están considerados con respecto a la información que se tiene hasta el momento.
- El número de módulos a revisar son 10, divididos en control de ingresos y egresos

- **Iteración 2 (1.2.0.0)**
- **Flujo de trabajo de requerimientos (1.2.1.0)**
  - Inducción (1.2.1.1)

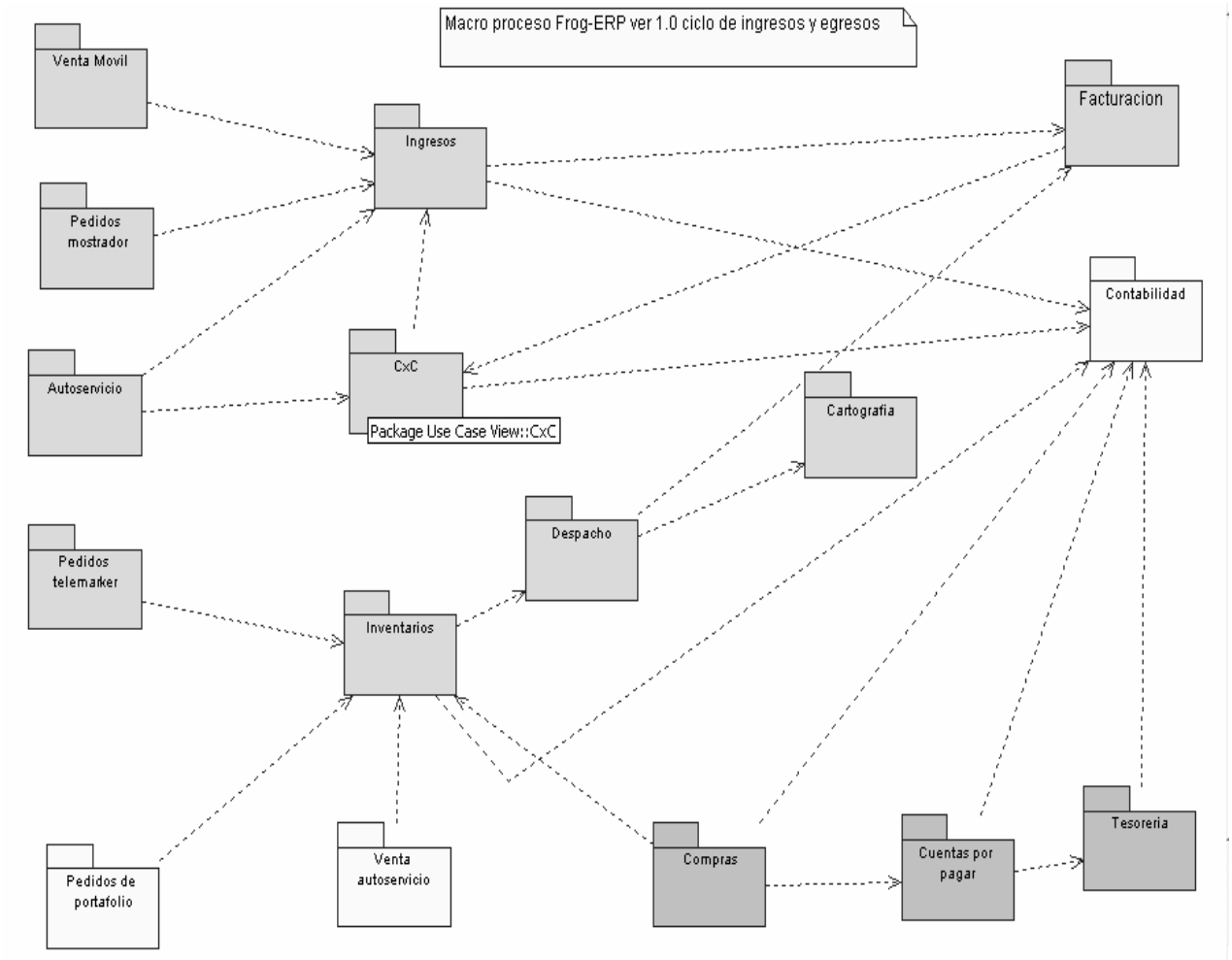
En este momento el equipo de trabajo realiza una investigación de campo para estudiar de forma más profunda y práctica las actividades del usuario.

Se registra el trabajo realizado en un Reporte de Actividades. Ver los siguientes formatos del anexo C opción 2 “Formato de Actividades”

a) Folio de formato FR/DN-ACT-001
b) Folio de formato FR/DN-ACT-002

- **Flujo de trabajo de análisis (1.2.2.0)**
  - Macroproceso (1.2.2.1)

El macroproceso siguiente muestra como están relacionados los módulos globales para los ciclos de ingresos y egresos.



- Matriz de procesos propuestos (1.2.2.2)

El objetivo de la matriz de procesos propuestos es facilitar al equipo de desarrollo el familiarizarse con las relaciones entre cada proceso y el puesto que le corresponde. De esta manera las futuras dudas o actividades a realizar sobre cada proceso se realizarán con el puesto correspondiente. Ver los siguientes formatos del anexo C opción 3 “Matriz de Procesos”

a) Compras, cuentas por pagar
b) Tesorería, Contabilidad, Pedido mostrador
c) Pedido Telemarketing

- Selección de procesos críticos (1.2.2.3)

La priorización busca establecer que procesos tienen una mayor importancia, de acuerdo a diversos criterios. En este caso de estudio, para el cliente es prioritario automatizar su proceso de ventas que incluyen las áreas de telemarketing y mostrador ya que representa el 70% de sus ingresos mensuales. Por lo tanto se harán los diagramas de caso de uso de estos procesos.

Los ingresos mensuales del cliente son de 30 millones de pesos, por eso es de gran importancia para él controlar este gran porcentaje de sus ventas.

Se realizarán los siguientes diagramas del macroproceso (Ver los formatos del anexo C opción 4 “Diagramas de modelado”:

a) Diagrama de actividades del pedido de mostrador
b) Diagrama de estados de pedido telemarketing
c) Diagrama de actividades del pedido pendiente
d) Diagrama de caso de uso de pedido telemarketing

- Restricciones (1.2.2.4)

La visión del equipo de trabajo sobre los requerimientos del usuario se va haciendo más específica con cada nueva actividad, de ahí que se detecten nuevas restricciones

- La primera restricción del cliente es que los tiempos de respuesta para levantar un pedido sean no lentos.

- Tener las herramientas de decisión para atender al cliente con más eficiencia (tener la información del cliente en línea de ventas, estado de cuenta, fecha de ultima compra, cuantos pedidos tiene en espera, cuando se entregan los pedidos, cuantas veces ha llamado, quien ha entregado sus pedidos, generar un pedido sugerido con base a su historial de ventas), es decir generar un proceso de CRM (Manejo de la relación con el cliente)

- Riesgos (1.2.2.5)

De acuerdo a la información obtenida hasta este momento no se han detectado nuevos riesgos.

- Factibilidad (1.2.2.6)

El estudio de factibilidad consistirá en ponderar las potenciales capacidades del equipo de desarrollo para cubrir los requerimientos del usuario en los tiempos tentativos propuestos bajo los riesgos y restricciones detectados hasta este momento. Para calificar la factibilidad hemos considerado los siguientes puntos:

0 No cumple nada hay que desarrollo todo
3 Requiere adecuaciones, se tiene gran parte pero hay que modificarlo
5 Cumple al 100%

Ver formato del anexo C opción 5 “Formato de Factibilidad”

- **Resultado de la iteración (1.2.4.0)**

- Plan de actividades (1.2.4.1)

El plan de actividades se realizó con los procesos que el negocio quiere que se le instale especificando días de duración, cuando empieza y termina. Ver formato del anexo C opción 6 “Plan de Actividades”

- **Iteración 3 (1.3.0.0)**

- **Flujo de trabajo de requerimientos (1.1.3.0)**

  - Levantamiento de requerimientos (1.3.1.1)

En esta actividad se obtiene información más específica acerca de lo que el usuario requiere que el sistema realice en cada proceso. Para que el equipo de trabajo tenga una idea clara de la relación proceso-puesto, se documentan los puestos que intervienen en cada proceso.

Posteriormente se trabaja en conjunto con el personal que desempeña cada puesto para determinar los requerimientos de cada proceso.

Ver formatos del anexo C opción 7 “Descripción de Puestos y Levantamiento de Requerimientos”

a) Vendedor Mostrador
b) Capturista
c) Requerimientos de la pantalla “Captura de clientes nuevos”

- **Flujo de trabajo de análisis (1.3.2.0)**

  - Casos de uso críticos (1.3.2.2)

Una vez obtenida información más específica sobre los requerimientos de los procesos críticos se modelan en forma de casos de uso y se establecen sus diagramas de estados

Se realizaron los siguientes diagramas a partir del macroproceso:

- Diagrama de actividades del pedido de mostrador
- Diagrama de caso de uso de pedido telemarketing
- Diagrama de estados de pedido telemarketing
- Diagrama de actividades del pedido pendiente

Ver los diagramas anteriores en el anexo C opción 4 “Diagramas de Modelado”

- **Resultado de la iteración (1.3.4.0)**

6.2.2 Formato integral de análisis

Aunque el presente formato se entrega al usuario al final de la fase de elaboración, debe tomarse en cuenta que es un documento que se va construyendo y actualizando a lo largo del proceso de análisis. Al Formato integral lo acompañan las minutas, cronogramas, etc. que se generan en cada actividad realizada durante el PAS. A continuación se muestra el formato resultante de este caso de estudio que es complementado con el anexo C

PAS-PAT	**Proyecto: Frog-Cravioto**		
Información General			
Analista:	Tesis	*Fecha:* 28-abr-2005	Versión: 3
Compañía: Cravioto		Revisó: Tesis / tesis_analisis@yahoo.com.mx	Tel. 51-30-27-30.

Antecedentes:

- El negocio tiene 2 sistemas de compañías diferentes para controlar la operación diaria. Un sistema que controla los ingresos y otro con controla los egresos.
- Se generan interfases al final del día para intercambio de información entre los sistemas.
- Se generan dobles procesos para controlar la información de día
- El área de sistema tiene que conocer 2 sistemas con estructuras diferentes

Objetivo General

- El negocio requiere tener el sistema en una sola estructura
- Implica optimizar las actividades de los usuarios.
- Quitar todas interfases entre los 2 sistemas

Necesidades

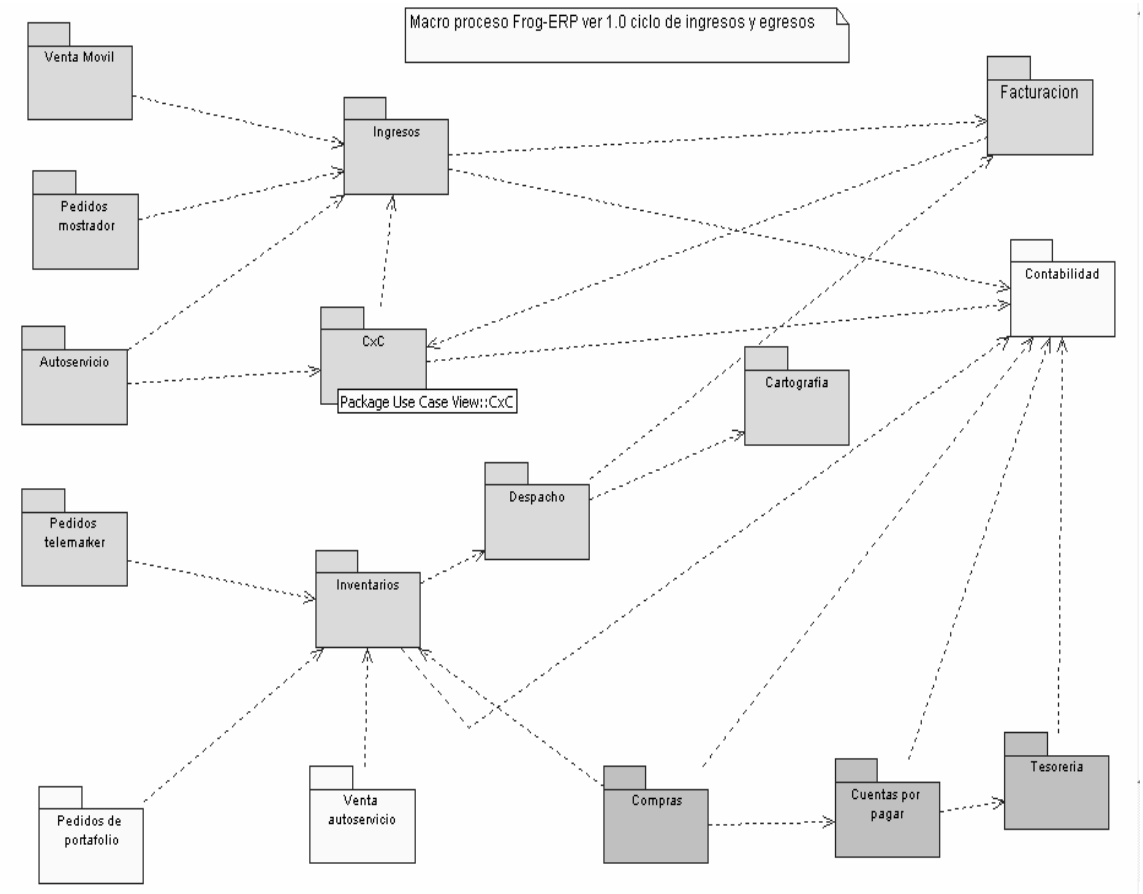
- La priorización para el cliente es automatizar su proceso de ventas que incluyen las áreas de telé marketing y mostrador ya que representa el 70% de sus ingresos mensuales.

Formato de minutas, requerimientos:

- Ver anexo C opción 3 “Matriz de Procesos”

Alcance:

- Ver actividad Macroproceso (1.2.2.1)



Áreas Involucradas

Responsables Cliente:

- Enrique Castañeda
- Concepción Hernández
- Patricio Villegas

Responsables consultoría:

- Roberto Caballero
- Nadia Zardaneta
- Lidia Resendiz
- Pablo Gutiérrez
- Arturo Rojas

Personas Entrevistadas

- Personal del área de telemarketing
- Personal del área de mostrador
- Personal del área de autoservicio

Formato de actividades, descripción del puesto, cronograma de actividades:

Ver formatos del anexo C opción 7 “Descripción de puestos y levantamiento de requerimientos”

Descripción de los procesos de negocio existentes:

Ver formatos del anexo C opción 7 “Descripción de puestos y levantamiento de requerimientos”

Descripción de los procesos de negocio propuesto:

Se realizaron los siguientes diagramas a partir del Macroproceso:

- Diagrama de actividades del pedido de mostrador
- Diagrama de caso de uso de pedido telemarketing
- Diagrama de estados de pedido telemarketing
- Diagrama de actividades del pedido pendiente

Casos de Uso:

Ver descripción de caso de uso opción (1.3.2.2)

Factibilidad de negocio:

Beneficios cualitativos

- Numero de personal administrativo se asignara a otras labores

Beneficios cuantitativos

Áreas beneficiadas con el sistema

- Compras
- Cuentas por pagar
- Tesorería
- Contabilidad

Factibilidad Técnica

- *Uso de tecnología de 3 capas*

Análisis de Riesgos

Número de Riesgo	***Descripción del riesgo***	***Probabilidad de ocurrencia***	***Responsable***	***Acciones de mitigación***
1	*Conexión remota*	*diario*	*Área soporte*	*Ampliar el ancho del enlace*
2	*Servidor de base de datos*	*diario*	*Área soporte*	*Garantizar la operación de la Base de Datos*

Plan General de Trabajo:

	Task Name	Duration	Start	Predecessors	Finish
	Sistemas de ingresos interior	**39 days**	**Mon 3/01/05**		**Thu 24/02/05**
	Pedidos mostrador	5 days	Mon 3/01/05		Fri 7/01/05
	Pedidos Telemarketing	7 days	Mon 10/01/05	2	Tue 18/01/05
	Autoservicio	4 days	Wed 19/01/05	3	Mon 24/01/05
	Ingresos	4 days	Tue 25/01/05	4	Fri 28/01/05
	Cuentas por cobrar	5 days	Mon 31/01/05	5	Fri 4/02/05
	Facturacion	3 days	Mon 7/02/05	6	Wed 9/02/05
	Inventarios	7 days	Fri 11/02/05	7	Mon 21/02/05
	Cartografia	3 days	Tue 22/02/05	8	Thu 24/02/05
	Sistemas de ingresos exterior	**17 days**	**Mon 3/01/05**		**Tue 25/01/05**
	Sistemas de ventas por Internet	5 days	Mon 3/01/05		Fri 7/01/05
	Sistema de ventas de vendedor portafoli	8 days	Mon 10/01/05	11	Wed 19/01/05
	Computo movil para venta autoservicio	4 days	Thu 20/01/05	12	Tue 25/01/05
	Sistemas de egresos	**47 days**	**Mon 3/01/05**		**Tue 8/03/05**
	Compras	7 days	Mon 3/01/05		Tue 11/01/05
	Cuentas por pagar	5 days	Wed 12/01/05	15	Tue 18/01/05
	Tesoreria	15 days	Wed 19/01/05	16	Tue 8/02/05
	Contabilidad	20 days	Wed 9/02/05	17	Tue 8/03/05

Exclusiones

- Área de recursos humanos
- Área de sistema contable

Conclusiones

- El análisis solo se hará en los lugares establecidos en el alcance global inicial y no en lugares nuevos que se agreguen como parte del crecimiento del negocio.
- Cualquier cambio a la definición inicial de los requerimientos se deberá revisar el impacto que conlleva realizarlos.

Conformidad de los Participantes

Vo.Bo. Reglas de Negocio
Roberto Gonzalez
Director general

Tesis
CONSULTORIA

Tesis
CONSULTORIA

6.2.3 Glosario de términos

Conocer y utilizar adecuadamente la terminología del usuario es de gran importancia para la comunicación y transmisión de información oral y escrita en un proyecto de desarrollo, razón por la cual es indispensable documentar un glosario de términos

El glosario obtenido a partir de este caso de estudio se puede ver en el anexo C opción 8 “Glosario de Términos”.

CONCLUSIONES

En este trabajo de tesis se ha desarrollado un procedimiento para el análisis de sistemas informáticos que busca guiar al analista en su labor de recopilar los requerimientos de un sistema y modelarlos adecuadamente, siguiendo una serie de pasos controlados, predecibles y repetibles de manera sencilla y concreta.

El procedimiento propuesto (PAS) se contempla como una técnica que puede ser ajustada y enriquecida a partir de la experiencia personal de quien lo adopte como herramienta de trabajo en sus proyectos de desarrollo. En la medida que esta primera versión del procedimiento sea perfeccionada y documentada, irá adquiriendo solidez y aceptación.

La difusión de un procedimiento sencillo y efectivo ayudará a cumplir uno de los objetivos a futuro de este trabajo: colaborar en el crecimiento de la cultura de análisis formal de sistemas informáticos en nuestro país.

Una característica importante del procedimiento es la flexibilidad, la cual permite efectuar repetidas veces las iteraciones indicadas hasta alcanzar los objetivos establecidos, y en función de la complejidad que presenta el proyecto analizarlo de manera satisfactoria.

Para lograr lo anterior se siguió un proceso secuencial de especialización en el tema, se revisó el concepto de sistema y el método típico de desarrollo en cascada. El método en cascada es el origen de todas las metodologías de desarrollo de sistemas informáticos, sin embargo presenta graves defectos los cuales invitan a buscar alternativas que los eviten. Los principales problemas que se detectaron es la carencia de un control oportuno de riesgos y el hecho de que busca desarrollar sistemas de alta complejidad en una única secuencia continua de actividades, lo cual provoca una pérdida de control en el proceso y un enorme incremento en la complejidad del trabajo.

Por esta razón se estudiaron diversas metodologías de análisis de sistemas informáticos con el fin de encontrar características o prácticas útiles y eficientes que ofrecieran resultados satisfactorios para conformar un procedimiento modificado, más sencillo y concreto pero igualmente efectivo y que no presentara las desventajas del proceso de desarrollo en cascada. Finalmente el procedimiento se estableció con una mecánica de trabajo basada en el proceso iterativo, agrupando las actividades en conjuntos llamados flujos de trabajo que conforman iteraciones completas y éstas a su vez fueron agrupadas en conjuntos más grandes llamados fases.

El proceso iterativo presenta enormes ventajas respecto del proceso en cascada y que fueron adaptadas en el PAS, entre ellas se encuentra un control y mitigación inmediato de los riesgos del sistema, la posibilidad de ir detallando y perfeccionando el conocimiento que el analista tiene del sistema, la capacidad de dividir en partes proyectos que presenten un nivel de complejidad muy alto, y en consecuencia llevar un control adecuado de lo que se hace en cada etapa.

Una técnica integrada en el procedimiento y que demuestra ser esencial para un buen análisis es sin duda el trabajar con casos de uso. Los casos de uso permiten expresar los requerimientos del usuario de un modo más amigable, es decir, con base en modelos (diagramas) podemos mostrarle a un usuario el proceso de negocio desde su propia perspectiva y la retroalimentación con éste nos permite perfeccionarlos. Con base

en casos de uso podemos identificar todas las operaciones que realiza el usuario y describirlas adecuadamente conformando la arquitectura del sistema.

Algo muy importante es determinar cuándo un caso de uso está completo, es decir, cómo saber si lo que describimos como caso de uso realmente lo es, y la respuesta es muy sencilla, *en un caso de uso el usuario debe obtener un resultado observable que le dé valor a su trabajo*.

Otra de las ventajas notables del procedimiento es el control de riesgos, el cual nos permite llevar un registro, diagnóstico y eliminación o minimización inmediata de los mismos. La atención oportuna y adecuada de los riesgos permite reducir al mínimo los costos que la tarea implica, eliminando la clara desventaja que presenta el proceso de desarrollo en cascada a este respecto. El proceso iterativo complementa esta capacidad, ya que el control de riesgos se actualiza, analiza y resuelve con cada nueva iteración.

Finalmente se buscó comprobar la efectividad del PAS a través de su aplicación a dos casos de estudio con características distintas.

El primero de ellos incluyó el análisis y diseño de una página de Internet que mostrara al usuario, en forma interactiva, en que consiste el procedimiento propuesto. Aplicando el procedimiento llegamos al resultado esperado y debido a que el sistema era sencillo, fue posible desarrollar la aplicación real.

El segundo proyecto buscó analizar la adaptación de un software existente a algunos casos de uso del módulo de pedidos en un sistema de control de una ferretería, y al aplicar de forma controlada y predecible las actividades indicadas en el procedimiento, se demostró que se puede alcanzar el objetivo de analizar adecuadamente un sistema informático independientemente de su complejidad.

Por lo anterior se considera que se ha cumplido satisfactoriamente con el objetivo de este proyecto de tesis ya que se probó la efectividad y flexibilidad del PAS en dos casos que presentaban características muy diferentes.

El procedimiento propuesto es una base, una versión “beta” abierta a la utilización y perfeccionamiento de quien lo adopte como método de análisis. La experiencia personal puede demostrar ser una fuente muy valiosa de conocimiento para ajustar las actividades y resultados que se incluyen en el procedimiento.

En la medida que se aporten experiencias y se reflejen en el procedimiento, se podrá contar con una alternativa cada vez más firme y probada para el análisis de sistemas informáticos.

El analista debe tener en mente que la actividad de recopilar requerimientos implica enfrentar condiciones que no se mencionan en ningún procedimiento formal de análisis de sistemas: el usuario es, por naturaleza, reacio al cambio o teme ser desplazado en su empleo por un sistema informático, cuando en realidad el objetivo es hacerlo más productivo. Por estas razones es posible que la recopilación de requerimientos no sea siempre una labor sencilla y transparente.

El analista debe complementar este procedimiento con técnicas adecuadas para la realización de entrevistas y cuestionarios que le preparen con una firme habilidad para descubrir, detectar y distinguir los requerimientos reales, y así encontrar la diferencia entre las tareas que el usuario solicita que el sistema haga y lo que realmente se necesita que el sistema realice.

ANEXO A

A continuación se muestran los formatos de trabajo sugeridos para el PAS

Formato F01- Minuta de reuniones

PAS-PAT *<Negocio / Consultoría>*			
Fecha de reunión:	**Lugar:**	**No Minuta:**	**Hora:**
<dd mm aaaa>	*<Oficinas/ Negocio/ Consultoría>*	*<MIN001>*	*<HH:MM>*
Asistentes Negocio:		**Asistentes Consultoría:**	
<Nombres de las personas involucradas en el proceso>:<hora de llegada de cada asistente>		*<Nombre de personas que otorgan el servicio>: <hora de llegada de cada asistente>*	
Agenda:			
<Puntos que se trataran>			
Evolución:			
<Desarrollo de cada punto agendado>			
Conclusiones:			
<Acuerdos a los que se llegaron>			
No.	**Compromiso de la Reunión**	**Fecha**	**Responsable**
1	*<Descripción del compromiso generado con fecha de solución y responsable que lo hará>*		
2	*<Descripción del compromiso generado con fecha de solución y responsable que lo hará>*		
Firmas de acuerdo de minuta de los asistentes:			
Nombre:		**Firma:**	
Focos de atención			
# Foco	**Descripción**		
	<Los focos de atención son los compromisos contraídos, pero hay que dar mayor atención a estos puntos>		
c.c.p. *<Gerente del Negocio y Gerente de la Consultoría>* *<Integrantes de la minuta>*			

Formato F02-Manejo de requerimientos

PAS-PAT

Folio Requerimiento: <*Numero interno*>

Manejo de Requerimientos

Nombre del proyecto*:<Nombre acordado para el proyecto>*
Nombre del caso de uso*:<Del diagrama de caso de uso asignar la referencia>*

Revisión Histórica

Fecha	**Folio interno de referencia**	**Concepto**	**ID. Analista**	**ID. Cliente**	**Fecha Entrega**	**Tipo de requerimiento**
<dd/mm/aaaa>	*<Numero>*	*<Descripción general>*	*<Tres letras>*	*<Tres letras>*	*<ddmmaa>*	*(Nuevo, Cambio,Error, Anexo)*

Detalle

<Hacer referencia al macroproceso al diagrama de caso que se va a usar para levantar el requerimiento>
<Ver formato de documentación de caso de uso>

Firmas Revisión:

Folio Referencia	**Analista**	**Cliente**	**Lider Proyecto**	**Lider cliente**

Formato F03- Formato de actividades

PAS-PAT	**Fecha:** *<ddmmaa>* **Folio:** *<ACT (id analista) # consecutivo>*
Nombre del proyecto:	
Formato de Actividades	
< Definir actividades programadas > <Anotar los puntos a revisar si es que existen> 1.- <Describir que se vio o se hizo> 2.- <Hora de llegada, hora de salida> 3.- <Lugar, área, departamento que se visito> 4.- <Usuarios que se visitaron>	
Firma Analista	**Firma responsable negocio**

Formato F04-Descripción de puestos

PAS-PAT	**Fecha:** <*ddmmaa*> **Nombre del proyecto:** **Nombre del puesto:** <*De acuerdo a lo definido por recursos humanos*> **Persona entrevistada:** <*Nombre de la persona a entrevistar*>
Descripción del puesto:	
Organigrama: <*Posición del puesto dentro del organigrama de la empresa*> **Actividades(Que hace):** <*Describir las funciones que realiza*> **Procedimiento(Como lo hace manual/sistematiza):** <*Redactar paso a paso el procedimiento*> **Resultados del proceso que realiza(reportes):** <*Que información resulta de sus actividades*> **Horario de trabajo:** **Preguntas clave:** <*Redactar las preguntas que le harías antes de entrevistarlo*>	
Firma Analista	**Firma Puesto**

Formato F05-Cronograma de actividades

PAS-PAT

Revisión: *<Num modificación>*

Nombre del proyecto:

Cronograma de actividades

Actividad	**Recurso(s)**	**Inicia**	**Termina**	**Avance**
<Describir lo que se va a realizar>	*<Quien lo va a hacer>*	*<Cuando empieza>*	*<Cuando termina>*	*<Porcentaje de avance>*

Firma Líder proyecto	**Firma responsable negocio**

Formato F06-Glosario de términos

PAS-PAT	**Nombre del proyecto:** **FechaRevisión:** *<Num modificación>*
Glosario de Términos	
Palabra: *<Escribir las palabras que puedan causar incertidumbre o que no es parte de lenguaje común en el proyecto>* Descripción de la palabra *<Definir donde se usa, para que sirve, quien la usa>*	
Realizado por: *<Documentador analista>*	

Formato F07-Matriz de procesos vs puestos

PAS-PAT			**Fecha:** *<ddmmaaaa>*		
			Nombre del proyecto:		
Matriz de procesos/áreas vs puestos					
Puesto / Área/proceso	**Puesto 1**	**Puesto 2**	**Puesto 3**	**Puesto 4**	**Puesto n**
Proceso 1	*<Rellenar cada cuadro que corresponda en esta matriz>*				
Área 1					
Área 2					
Área 3					
Área m					
Proceso 2					
Área 1					
Área 2					
Área 3					
Área m					
Proceso 3					
Área 1					
Área 2					
Área 3					
Área m					
Proceso 4					
Área 1					
Área 2					
Área 3					
Área m					
Proceso x					
Área 1					
Área 2					
Área 3					
Área m					
Firma Líder proyecto			**Firma responsable Cliente**		

ANEXO B

A continuación se muestran los formatos de trabajo obtenidos en el primer caso de estudio y que complementan el Formato Integral de Análisis

1 Proyección del proyecto

PAS-PAT	**Fecha:** 29/03/2005 **Nombre del proyecto:** PAS - PAT			
Concepto	**Sem1**	**Sem2**	**Sem3**	**Sem4**
Acercamiento				
Análisis prioridad				
Análisis restante				
Propuesta Construcción				
Construcción				
Revisión Funcional				
Capacitación				
Liberación				

2 Minutas

Minuta de reuniones

PAS-PAT **PAS – PAT**			
Fecha de reunión: 29/03/2005	**Lugar de reunión:** Facultad de Ingeniería	**No Minuta:** PG032905	**Hora:** 18:30 hrs.
Asistentes Negocio: Ing. Gloria Mata Hernández (GM)	**Asistentes Consultoría:** Arturo Rojas (AR) Lidia Resendiz (LR) Roberto Caballero (RC) Pablo Gutiérrez (PG) Nadia Zardaneta (NZ)		
Agenda: • Entrega de la información del capítulo 5, que contiene el PAS "Procedimiento de Análisis de Sistemas" • Toma de requerimientos de la página WEB donde se mostrará el procedimiento			
Evolución: • AR, LR, RC, PG y NZ entregaron información del capítulo 5, GM reviso la información contenida en el capítulo • PG recabo los requerimientos solicitados por GM			
Conclusiones: • GM valido la información del capítulo 5 • PG tomo los requerimientos de la página WEB			

No.	COMPROMISO DE LA REUNIÓN	FECHA	RESPONSABLE
1	Entregar el análisis completo por prioridades la siguiente semana 04/05/2005, responsable AR, LR, RC, PG y NZ.		
2	Entregar un cronograma de actividades del proyecto PAT – PAS 04/05/2005, responsable AR, LR, RC, PG y NZ.		
3	Realizar un planteamiento del Proyecto PAT – PAS 04/05/2005, responsable AR, LR, RC, PG y NZ.		
4	Realizar un plan de actividades PAT – PAS 04/05/2005, responsable AR, LR, RC, PG y NZ.		
5	AR, LR, RC, PG y NZ, propondrán la construcción		

Firmas de acuerdo de minuta de los asistentes:	
Nombre	**Firma**
Arturo Rojas (AR)	
Lidia Resendiz (LR)	
Roberto Caballero (RC)	
Nadia Zardaneta (NZ)	
Pablo Gutiérrez (PG)	
Ing. Gloria Mata Hernández (GM)	

Focos de atención	
# Foco	**Descripción**
No se generaron focos de atención	

c.c.p. Ing. Gloria Mata Hernández
c.c.p. Arturo Rojas
c.c.p. Lidia Resendiz
c.c.p. Roberto Caballero
c.c.p. Pablo Gutiérrez
c.c.p. Nadia Zardaneta

Minuta de reuniones

PAS-PAT **PAS – PAT**			
Fecha de reunión: 05/04/2005	**Lugar de reunión:** Facultad de Ingeniería	**No Minuta:** PG040505	**Hora:** 18:30 hrs.
Asistentes Negocio: Ing. Gloria Mata Hernández (GM)	**Asistentes Consultoría:** Arturo Rojas (AR) Lidia Resendiz (LR) Roberto Caballero (RC) Pablo Gutiérrez (PG) Nadia Zardaneta (NZ)		

Agenda:

- GM revisará el análisis completo por prioridades realizado por AR, LR, RC, PG y NZ
- Se discutirá el cronograma de actividades del proyecto PAT – PAS, propuesto por AR, LR, RC, PG y NZ
- AR, LR, RC, PG y NZ, discutirán con GM el planteamiento del Proyecto PAT – PAS (riesgos, restricciones, alcance y proyección del proyecto en el tiempo)
- AR, LR, RC, PG y NZ, discutirán con GM el Plan de Actividades del proyecto PAT – PAS (Macroproceso, Matriz de procesos y factibilidad)
- GM revisará la propuesta de Construcción hecha por AR, LR, RC, PG y NZ.

Evolución:

- GM revisó el análisis
- Todos los asistentes revisaron el Planteamiento del Proyecto
- Todos los asistentes revisaron el Plan de Actividades
- Todos los asistentes revisaron la propuesta de Construcción

Conclusiones:

- GM valido el análisis
- Todos los asistentes validaron el Planteamiento del Proyecto
- Todos los asistentes validaron el Plan de actividades
- Todos los asistentes validaron la propuesta de Construcción

No.	COMPROMISO DE LA REUNIÓN	FECHA	RESPONSABLE
1	Entregar el avance de la construcción del sistema PAT – PAS en base al plan 12/04/2005, responsable AR, LR, RC, PG y NZ.		

Firmas de acuerdo de minuta de los asistentes:	
Nombre	**Firma**
Arturo Rojas (AR)	
Lidia Resendiz (LR)	
Roberto Caballero (RC)	
Nadia Zardaneta (NZ)	
Pablo Gutiérrez (PG)	
Ing. Gloria Mata Hernández (GM)	

Focos de atención	
# Foco	**Descripción**
No se generaron focos de atención	

c.c.p. Ing. Gloria Mata Hernández
c.c.p. Arturo Rojas
c.c.p. Lidia Resendiz
c.c.p. Roberto Caballero
c.c.p. Pablo Gutiérrez
c.c.p. Nadia Zardaneta

Minuta de reuniones

PAS-PAT **PAS – PAT**			
Fecha de reunión: 12/04/2005	**Lugar de reunión:** Facultad de Ingeniería	**No Minuta:** PG041205	**Hora:** 18:30 hrs.
Asistentes Negocio: Ing. Gloria Mata Hernández (GM)	**Asistentes Consultoría:** Arturo Rojas (AR) Lidia Resendiz (LR) Roberto Caballero (RC) Pablo Gutiérrez (PG) Nadia Zardaneta (NZ)		
Agenda: • Revisar el avance de la construcción del sistema "Proyecto PAT – PAS"			
Evolución: • GM revisó el avance del sistema PAT - PAS			
Conclusiones: • GM valido el avance del sistema PAT – PAS y no se registraron contratiempos, todo fue validado conforme al plan			

No.	COMPROMISO DE LA REUNIÓN	FECHA	RESPONSABLE
1	Revisión funcional del sistema PAT – PAS con base en el plan 19/04/2005, responsable AR, LR, RC, PG, NZ y GM.		
2	Validar la Liberación del sistema PAT – PAS con base en el plan 19/04/2005, responsable AR, LR, RC, PG, NZ y GM.		

Firmas de acuerdo de minuta de los asistentes:	
Nombre	**Firma**
Arturo Rojas (AR)	
Lidia Resendiz (LR)	
Roberto Caballero (RC)	
Nadia Zardaneta (NZ)	
Pablo Gutiérrez (PG)	
Ing. Gloria Mata Hernández (GM)	

Focos de atención	
# Foco	**Descripción**
No se generaron focos de atención	

c.c.p. Ing. Gloria Mata Hernández
c.c.p. Arturo Rojas
c.c.p. Lidia Resendiz
c.c.p. Roberto Caballero
c.c.p. Pablo Gutiérrez
c.c.p. Nadia Zardaneta

Minuta de reuniones

PAS-PAT **PAT – PAS**			
Fecha de reunión: 19/04/2005	**Lugar de reunión:** Facultad de Ingeniería	**No Minuta:** PG041905	**Hora:** 18:30 hrs.
Asistentes Negocio: Ing. Gloria Mata Hernández (GM)	**Asistentes Consultoría:** Arturo Rojas (AR) Lidia Resendiz (LR) Roberto Caballero (RC) Pablo Gutiérrez (PG) Nadia Zardaneta (NZ)		
Agenda: • GM revisaran la funcionalidad del sistema PAT – PAS • AR, LR, RC, PG, NZ y GM, formalizaran la liberación del sistema PAT – PAS • AR, LR, RC, PG, NZ y GM, acordaran la capacitación			
Evolución: • GM revisó la funcionalidad del sistema PAT – PAS • AR, LR, RC, PG, NZ y GM formalizaron la liberación del sistema PAT – PAS • AR, LR, RC, PG, NZ y GM programaron una sesión de capacitación del sistema PAT - PAS			
Conclusiones: • GM valido la funcionalidad del sistema con base en los requerimientos • Todos los asistentes formalizaron la liberación • Se programa una sesión de 30 minutos de capacitación del sistema PAT - PAS			

No.	COMPROMISO DE LA REUNIÓN	FECHA	RESPONSABLE
1	Se programo una sesión de 30 minutos para la capacitación de uso del sistema para el día 26/04/2005, responsable de impartirla PG		

Firmas de acuerdo de minuta de los asistentes:	
Nombre	**Firma**
Arturo Rojas (AR)	
Lidia Resendiz (LR)	
Roberto Caballero (RC)	
Nadia Zardaneta (NZ)	
Pablo Gutiérrez (PG)	
Ing. Gloria Mata Hernández (GM)	

Focos de atención	
# Foco	**Descripción**
No se generaron focos de atención	

c.c.p. Ing. Gloria Mata Hernández
c.c.p. Arturo Rojas
c.c.p. Lidia Resendiz
c.c.p. Roberto Caballero
c.c.p. Pablo Gutiérrez
c.c.p. Nadia Zardaneta

3 Formato de actividades

PAS-PAT	**Fecha:** 29/03/2005 **Folio:** 001

Nombre del proyecto: PAS -PAT

Formato de Actividades

1. El usuario (Ing. Gloria Mata), realiza una consulta del procedimiento propuesto "PAS"

2. Consulta la primera parte del procedimiento que contempla una introducción

3. Consulta los componentes principales del PAS

4. Consulta el Proceso Iterativo

5. Consulta las Fases y los Puntos de Transición

6. Consulta los Flujos de Trabajo

7. Revisa los Diagramas de Modelado

8. Consulta el corazón de la Tesis "Tabla de Procedimiento"

9. Realiza una consulta en forma de Matriz de cada uno de los puntos que integran la Tabla de Procedimiento

10. La Ing. finaliza la consulta

Firma Analista	**Firma responsable negocio**
PAS	Ing. Gloria Mata

4 Análisis de factibilidad

Para calificar la factibilidad hemos considerado los siguientes puntos:

Para un desarrollo completamente nuevo

0 No es posible realizarlo
3 Se puede realizar sólo en parte
5 Cumple al 100%

PAS-PAT	**Fecha:** 05/04/2005 **Nombre del proyecto:** PAS - PAT		
Concepto	**Calificación previa**	**Valor de Importancia (0-100)**	**Puntos**
Se puede programar de modo que sea visible por cualquier navegador de Internet	5	5	25
Puede ser visible en cualquier plataforma	5	5	25
Se sigue la secuencia del procedimiento	5	5	25
Se puede explicar el procedimiento a detalle	5	5	25
Se puede analizar un sistema a partir de la vista en WEB del procedimiento	5	5	25
Puntos a cubrir (125 total)			125
Resultados: Total de conceptos a calificar: Numero de conceptos que cumple 5: Numero de conceptos que cumple 3: Numero de conceptos que cumple 0:	 5 5	100 100 % (5/5)	
Conclusiones: • De operación se cubre un (5/5) 100 = 100%			

5 Levantamiento de requerimientos

PAS-PAT

Folio Requerimiento: PASPAT002

Nombre del proyecto: PAS - PAT

Manejo de Requerimientos

Nombre del proyecto: PAS-PAT
Nombre del proceso/área: Página WEB

Revisión Histórica

Fecha	Folio referencia	Concepto	ID. Analista	ID. Cliente	Fecha Entrega	Tipo de Requerimiento (Nuevo, cambio, error, anexo)
1-abr-05	1	Página Principal	Tesis	PAT		Nuevo

Detalle

Los datos que la página principal debe incluir son:

- Liga a la explicación de Proceso Iterativo
- Liga a la explicación de fases y puntos de transición
- Liga a la explicación de flujos de trabajo
- Ligas a las fases
- Liga a la tabla principal del procedimiento

Firmas Revisión

Folio Referencia	Analista	Cliente	Líder Proyecto	Líder cliente
1	Tesis	PAT		

6 Cronograma de actividades

PAS-PAT

Revisión: 001

Nombre del proyecto: PAS - PAT

Cronograma de actividades

Actividad	Recurso(s)	Inicia	Termina	Avance

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Elaboración de las pagínas principales	1 día	ma 05/04/05	ma 05/04/05	
2	Elaboración de páginas secundarias	5 días	mi 06/04/05	ma 12/04/05	1
3	Enlace de las paginas principales con las secundarias	7 días	mi 13/04/05	ju 21/04/05	2
4	Formato de la pagina	2 días	vi 22/04/05	lu 25/04/05	3

Firma Líder proyecto	Firma responsable negocio

ANEXO C

A continuación se muestran los formatos de trabajo obtenidos en el segundo caso de estudio y que complementan el formato integral de análisis

1 Proyección del proyecto

PAS-PAT													
	Fecha: 25/03/2005 **Nombre del proyecto:** Proyecto Frog-Cravioto												
Concepto	**Sem 1,2**	**Mes 01**	**Mes 02**	**Mes 03**	**Mes 04**	**Mes 05**	**Mes 06**	**Mes 07**	**Mes 08**	**Mes 09**	**Mes 10**	**Mes 11**	**Mes 12**
Acercamiento													
Análisis prioridad													
Análisis restante													
Propuesta Construcción													
Construcción													
Revisión Funcional													
Capacitación													
Liberación													

2 Formato de actividades

a) Folio de formato FR/DN-ACT-001

PAS-PAT	**Fecha:** 05/03/2005 **Folio:** FR/DN-ACT-001 **Nombre del proyecto:** Proyecto Frog-Cravioto **Modulo: Pedidos mostrador**

Formato de Actividades

1.- Se visito al tienda principal del centro de la ciudad de México

2.- Se tienen 15 vendedores de mostrador

3.- Se generan en promedio al día 800 pedidos de los 15 vendedores

4.- Cada pedido tiene entre 5 y 80 partidas

5.- Los sábados es un día donde hay mas pedidos pero con menos partidas

6. Existe un área de entrega de mercancía

7. Hay 2 cajeros

8. Tienen un área de devoluciones

9. Una vez que se cobra el pedido se genera un ticket al sub almacén de manera automática. La impresión se genera cada 2 minutos.

10. Los pedidos pueden ser de contado, crédito o el cliente lo pase a recoger posteriormente.

11. La hora pico para tomar pedidos son desde las 12:00 horas hasta las 15:00

Firma Usuario	**Firma Cliente**
_____ Tesis	_____ Cravioto

b) Folio de formato FR/DN-ACT-001

PAS-PAT	**Fecha:** 03/03/2005 **Folio:** FR/DN-ACT-002 **Nombre del proyecto:** Proyecto Frog-Cravioto **Modulo: Pedidos telemarketing** **Visita a centro de distribución**

Formato de Actividades

1.- Se visito al centro de distribución de Ecatepec

2.- Se capturan pedidos vía teléfono o los pedidos los envían por fax los vendedores de Portafolio

3. Se tienen 10 capturistas de pedidos

4. Se tiene 8 vendedores de portafolio

5.- Se generan en promedio al día 100 pedidos de los 18 vendedores

6.- Cada pedido tiene entre 50 y 300 partidas

7. El pedido registrado pasa a cuentas por cobrar para autorizarse

8. Una vez autorizado en cuentas por cobrar pasa a administración de almacén

9. El administrador de almacén decide por antigüedad y monto del pedido la autorización para generar el surtido

10. Los subalmacenes reciben la parte del pedido que corresponda y los envían al área de consolidado

11. El área de consolidado recibe los pedidos de los subalmacenes y vuelve a empacar.

12. Consolidado los entrega a el área de despacho y despacho los asigna a una ruta de envío

13. Una vez enrutados los pedidos se envían a facturar

Firma Usuario	**Firma Cliente**
_____ Tesis	_____ Cravioto

3 Matriz de procesos

a) Compras, cuentas por pagar

PAS-PAT	**Fecha:** 22/03/2005 **Nombre del proyecto:** Proyecto Frog-Cravioto **Compras, Cuentas por Pagar**

MACROPROCESO	**PROCESO**	**Almacenista**	**Comprador**	**Proveedor**	**Gerente de Compras**	**Recepcionista**	**Auxiliar CXP**
Compras							
	Alta requisición						
	Autorización requisición						
	Cotización proveedor						
	Orden de compra						
	Autorización Orden compra						
	Entrega OC inventarios						
Cuentas por pagar							
	Tomar a revisión						
	Conciliación documentos						
	Anticipo a proveedores						
	Programación de pagos						
	Envío de pagos a tesorería						

b) Tesorería, contabilidad, pedido mostrador

PAS-PAT		**Fecha:** 22/03/2005 **Nombre del proyecto:** Proyecto Frog-Cravioto **Tesorería, Contabilidad, Pedido mostrador**				
MACRO-PROCESO	**PROCESO**	**Auxiliar Contable**	**Contador**	**Auxiliar Tesorería**	**Vendedor mostrador**	**Cliente**
Tesorería						
	Generación de cheques					
	Generación de transferencias					
	Conciliación bancaria					
Contabilidad						
	Póliza de ingresos contado					
	Póliza de ingresos CxC					
	Póliza de costos					
	Póliza de compras					
	Estado de resultados					
Pedidos mostrador						
	Pedido mostrador					

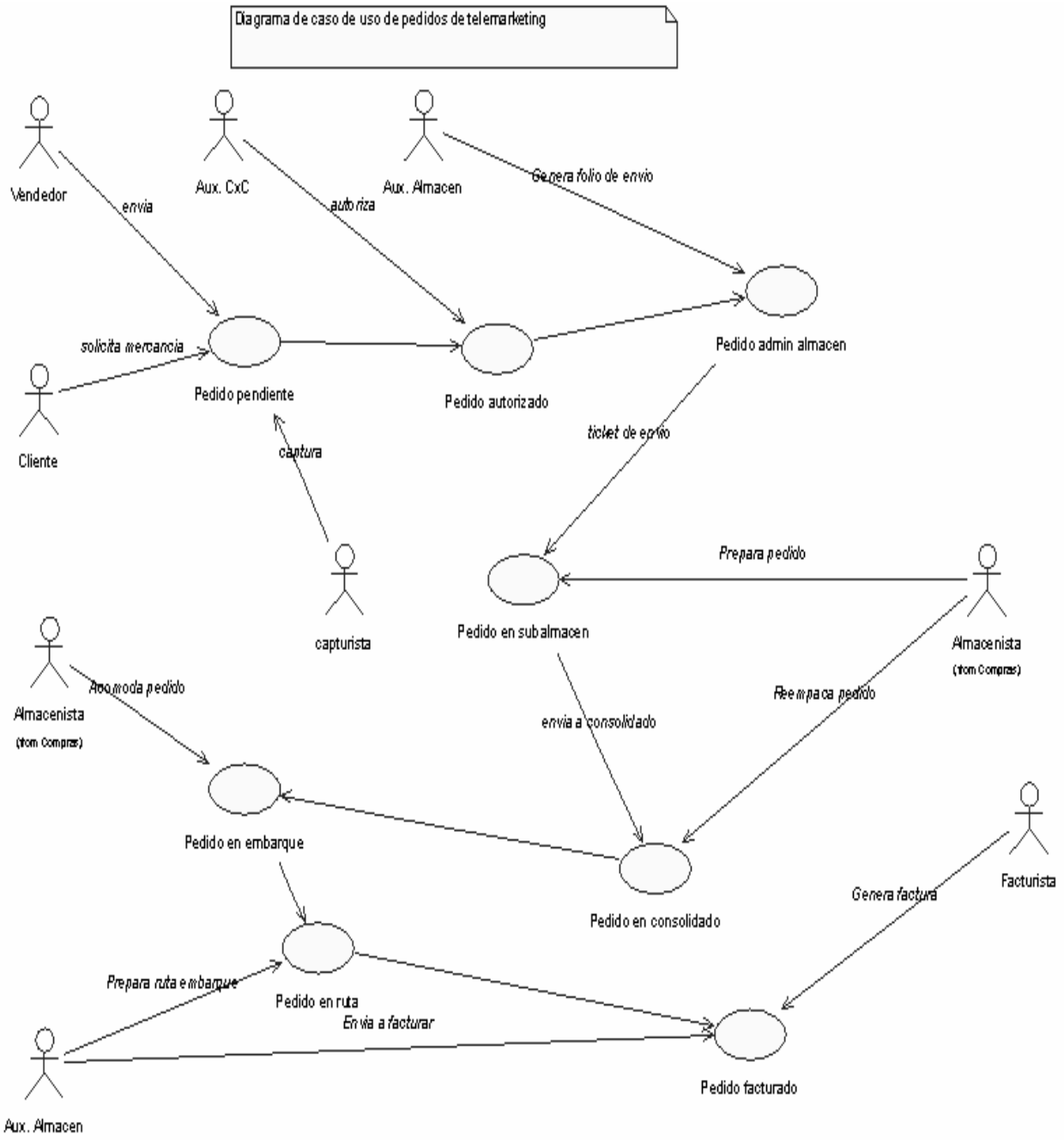
c) Pedido telemarketing

PAS-PAT	**Fecha:** 22/03/2005 **Nombre del proyecto:** Proyecto Frog-Cravioto **Pedido Telemarketing**

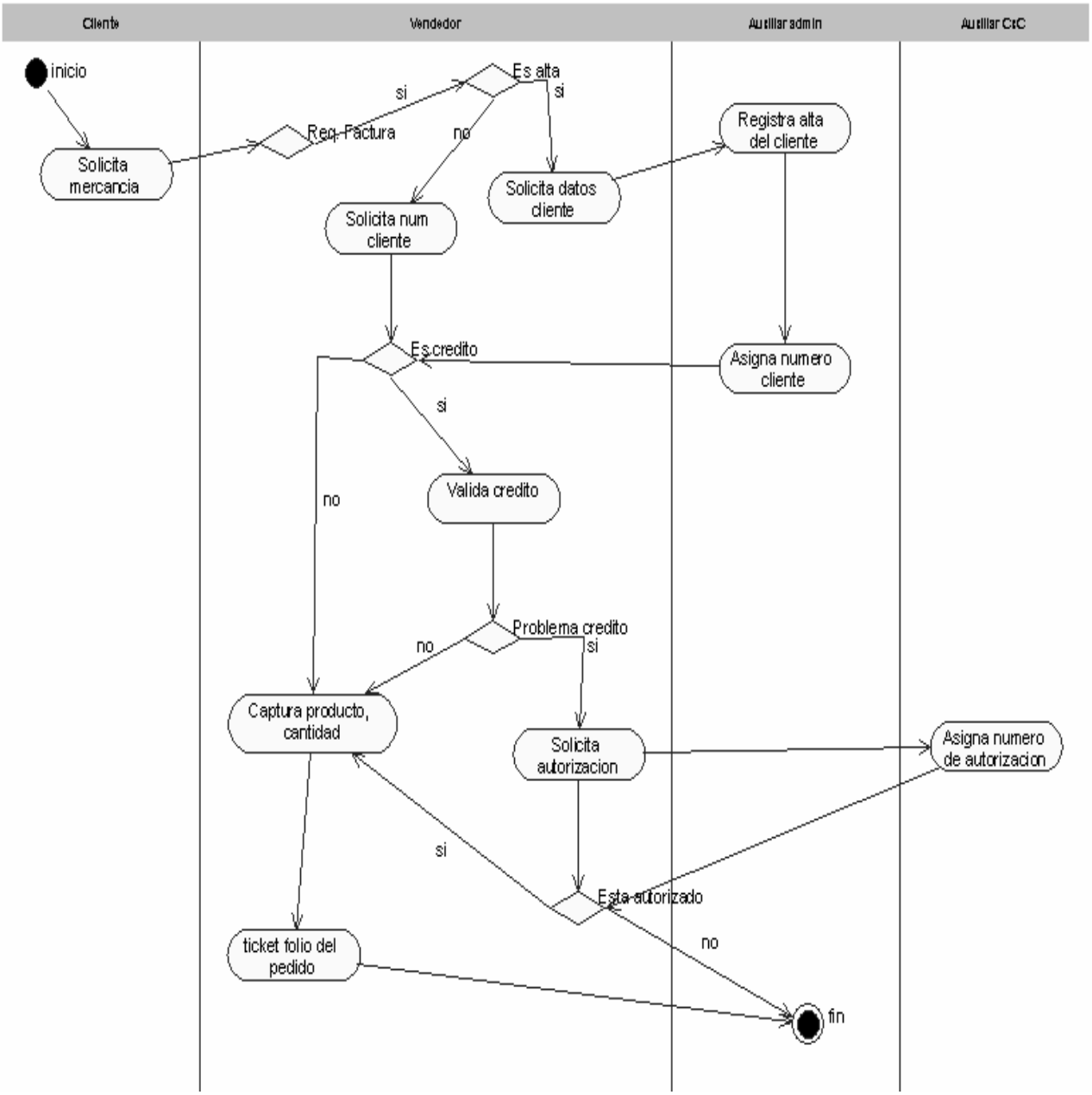
MACRO PROCESO	PROCESO	Capturista	Vendedor Portafolio	Cliente	Consolidado	Cuentas por Cobrar	Facturista	Embarque
Pedido telemarketing								
	Pedido pendiente							
	Pedido cancelado							
	Pedido suspendido							
	Pedido autorizado							
	Pedido admin almacén							
	Pedido subalmacen							
	Pedido consolidado							
	Pedido en despacho							
	Pedido enrutado							
	Pedido reprogramado							
	Pedido facturado							
	Pedido entregado							

4 Diagramas de modelado

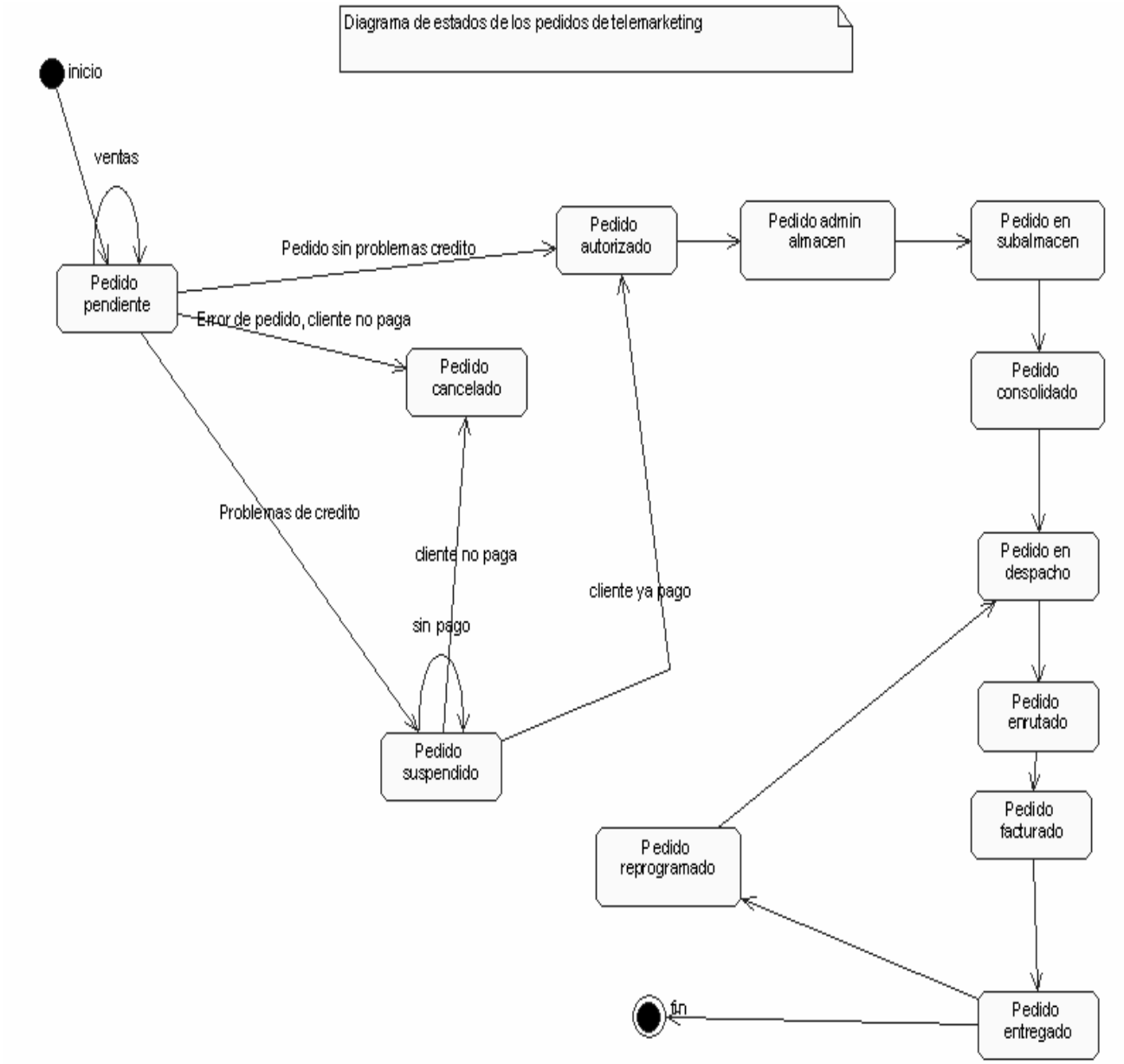
a) Diagrama de caso de uso de pedido de telemarketing



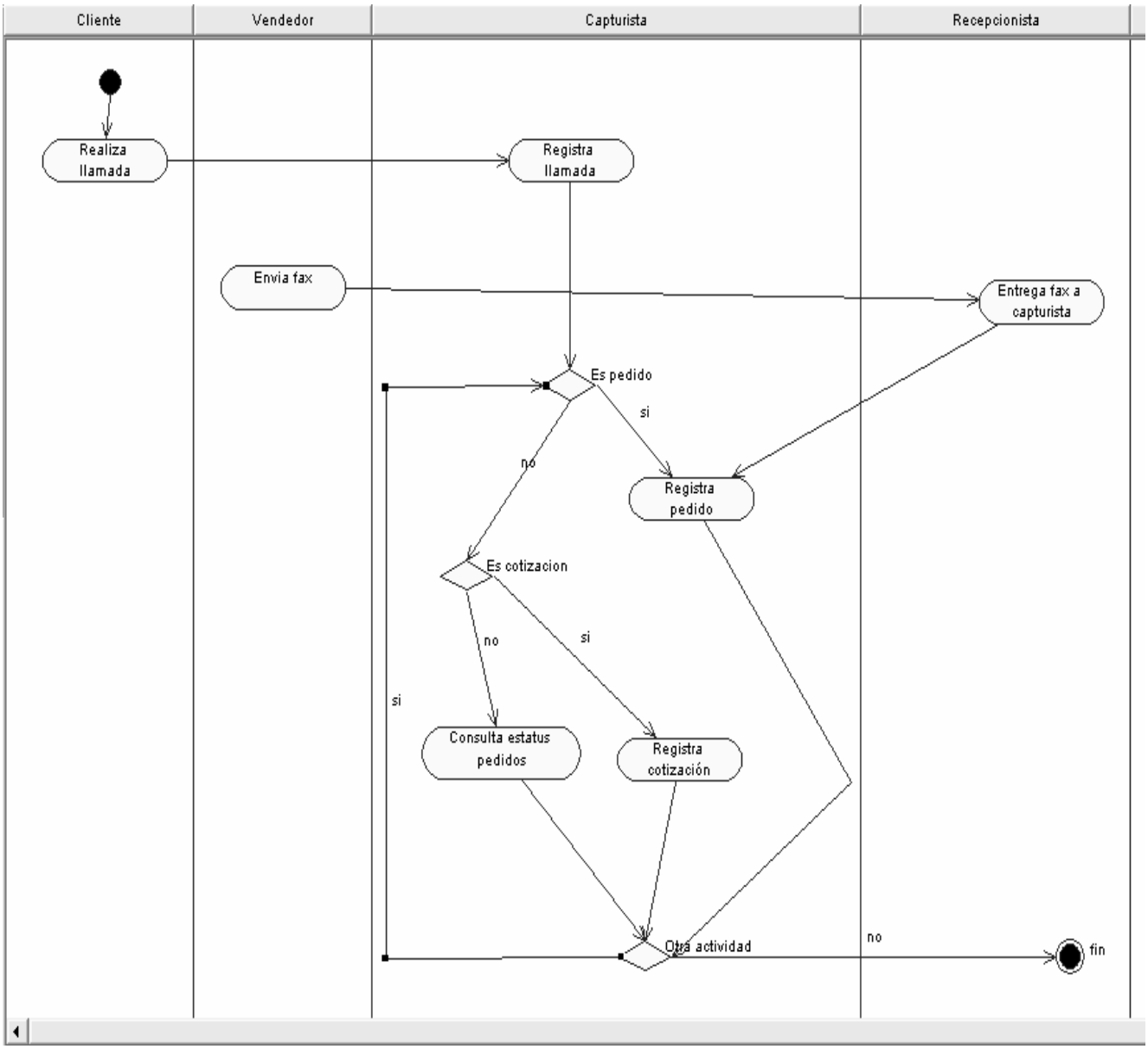
b) Diagrama de actividades del pedido de mostrador



c) Diagrama de estados de pedido telemarketing



d) Diagrama de actividades del pedido pendiente



5 Formato de factibilidad

PAS-PAT	**Fecha:** 22/03/2005 **Nombre del proyecto:** Proyecto Frog-Cravioto		
Concepto	Calificación previa	Valor de Importancia (0-100)	Puntos
Otros: (35%) a) Tiempos esperados del cliente para entregar el proyecto b) Hardware adecuado c) Funcionamiento remoto d) Base de datos e) Tiempos de respuesta	 5 3 3 5 3	 5 5 5 10 10	 25 15 15 50 30
Puntos a cubrir (175 total)			135
Operación de módulos (65%) a) Autoservicio b) Pedidos mostrador c) Facturación d) Cuentas pro cobrar e) Inventarios f) Pedidos telemarketing g) Contabilidad h) Compras i) Cuentas por Pagar j) Tesorería k) Manejo de cómputo móvil para venta autoservicio l) Sistema de ventas por Internet m) Sistema de ventas de vendedor portafolio n) Cartografía	 5 5 3 0 3 3 0 3 5 3 0 3 3 0	 5 5 5 5 5 5 5 5 5 5 5 2 5 3	 25 25 15 0 15 15 0 15 25 15 0 6 15 0
Puntos a cubrir 325 total			171
Resultados: Total de conceptos a calificar: Numero de conceptos que cumple 5: Numero de conceptos que cumple 3: Numero de conceptos que cumple 0:	 19 4 10 5	100 21 % (4/19) 53 % (10/19) 26 % (5/19)	
Conclusiones: • De operación se cubre un (171/325) = 53% • De otros se cubre un (135/175) = 77%			

6 Plan de actividades

PAS-PAT

Fecha: 28/03/2005

Nombre del proyecto: Proyecto Frog-Cravioto

Plan de Actividades

Task Name	Duration	Start	Predecessors	Finish
Sistemas de ingresos interior	**39 days**	**Mon 3/01/05**		**Thu 24/02/05**
Pedidos mostrador	5 days	Mon 3/01/05		Fri 7/01/05
Pedidos Telemarketing	7 days	Mon 10/01/05	2	Tue 18/01/05
Autoservicio	4 days	Wed 19/01/05	3	Mon 24/01/05
Ingresos	4 days	Tue 25/01/05	4	Fri 28/01/05
Cuentas por cobrar	5 days	Mon 31/01/05	5	Fri 4/02/05
Facturacion	3 days	Mon 7/02/05	6	Wed 9/02/05
Inventarios	7 days	Fri 11/02/05	7	Mon 21/02/05
Cartografia	3 days	Tue 22/02/05	8	Thu 24/02/05
Sistemas de ingresos exterior	**17 days**	**Mon 3/01/05**		**Tue 25/01/05**
Sistemas de ventas por Internet	5 days	Mon 3/01/05		Fri 7/01/05
Sistema de ventas de vendedor portafoli	8 days	Mon 10/01/05	11	Wed 19/01/05
Computo movil para venta autoservicio	4 days	Thu 20/01/05	12	Tue 25/01/05
Sistemas de egresos	**47 days**	**Mon 3/01/05**		**Tue 8/03/05**
Compras	7 days	Mon 3/01/05		Tue 11/01/05
Cuentas por pagar	5 days	Wed 12/01/05	15	Tue 18/01/05
Tesoreria	15 days	Wed 19/01/05	16	Tue 8/02/05
Contabilidad	20 days	Wed 9/02/05	17	Tue 8/03/05

Firma Líder proyecto	**Firma responsable negocio**

7 Descripción de puestos y levantamiento de requerimientos

a) Vendedor mostrador

PAS-PAT	**Fecha:** 03/04/2005 **Nombre del proyecto:** Proyecto Frog-Cravioto **Puesto: Vendedor mostrador**

Descripción del puesto

Organigrama:

GERENTE DE TIENDA

VENDEDOR

Actividades(Que hace):

- Saluda al cliente
- Captura pedidos del cliente
- Genera ticket del pedido
- Ofrece promociones de artículos
- Ofrece artículos fuera de línea
- Consulta existencias

Procedimiento(Como lo hace manual/sistematizado):

- Llega el cliente y lo saluda
- Le pregunta al cliente que desea
- El cliente le pregunta código del producto y si tiene existencias el precio de mayoreo y menudeo
- Si hay existencias entonces solicita le levanten el pedido.
- El vendedor pregunta si es nota o factura
- Si es factura solicita numero de cliente en caso contrario procede a dar de alta como cliente nuevo.
- Una vez que tenga el numero de cliente entonces pregunta si el pedido es de contado o crédito. Y hora de entrega.
- Comienza a capturar código de producto, cantidad, validando el sistema si tiene existencias.
- El cliente puede interrumpir cuanto lleva pedido en dinero.
- Una vez terminado de capturar el vendedor genera el ticket y lo entrega al cliente.

Resultados del proceso que realiza(reportes)

- Ticket del pedido
- Reporte de comisiones por vendedor

Horario de trabajo:

- Lunes a sábado
- De 9:00 a 18:00 hrs primer horario
- De 11:00 a 20:00 hrs segundo horario

Sueldo:

- 700 semanales
- 10% en vales
- Seguro social
- Comisión por volumen de ventas por articulo comisionable

Preguntas clave:

- Numero de pedidos en promedio por día?
- Horas picos
- Numero de partidas por pedido
- Volumen de ventas para precios de mayoreo
- Datos de cliente para dar de alta
- Características principales del articulo

Firma Analista	**Firma Puesto**
_____ Tesis	_____

b) Capturista

PAS-PAT	**Fecha:** 03/04/2005 **Nombre del proyecto:** Proyecto Frog-Cravioto **Puesto: Capturista**

Descripción del puesto

Organigrama:

GERENTE DE TIENDA

CAPTURISTA

Actividades(Que hace):

- Saluda al cliente
- Captura pedidos del cliente y pedidos del vendedor
- Genera ticket del pedido
- Ofrece promociones de artículos
- Ofrece artículos fuera de línea
- Consulta existencias
- Registra tipo de llamadas
- Registra causas de no compra al realizar la llamada del cliente

Procedimiento(Como lo hace manual/sistematizado):

- Llama al cliente o recibe una llamada o recibe un fax del pedido
- Le pregunta al cliente que desea si es llamada
- Registra causa de venta o no venta
- El cliente le pregunta código del producto y si tiene existencias el precio de mayoreo
- Si hay existencias entonces solicita le levanten el pedido.
- El vendedor pregunta si es nota o factura
- Si es factura solicita numero de cliente en caso contrario procede a dar de alta como cliente nuevo y termina el proceso.
- Una vez que tenga el numero de cliente entonces pregunta si el pedido es de contado o crédito. Y hora de entrega.
- Comienza a capturar código de producto, cantidad, validando el sistema si tiene existencias.
- El cliente puede interrumpir cuanto lleva pedido en dinero.
- Una vez terminado de capturar el vendedor genera el ticket y lo guarda
- El pedido se queda en estado pendiente

Resultados del proceso que realiza(reportes)

- Reporte de comisiones por vendedor
- Reporte de actividades del día por vendedor
- Reporte de pedidos pendientes por autorizar

Horario de trabajo:

- Lunes a sábado
- De 9:00 a 18:30 hrs

Sueldo:

- 700 semanales
- 10% en vales
- Seguro social
- Comisión por volumen de ventas por articulo comisionable

Preguntas clave:

- Numero de pedidos en promedio por día?
- Horas picos
- Numero de partidas por pedido
- Datos de cliente para dar de alta
- Características principales del articulo

Firma Analista	**Firma Puesto**
_____ Tesis	_____

c) Levantamiento de requerimientos para la pantalla de captura de clientes nuevos

PAS-PAT

Folio Requerimiento: F02CRA001

Manejo de Requerimientos

Nombre del proyecto: Frog-Cravioto
Nombre del proceso/área: Alta de clientes nuevo

Revisión Histórica

Fecha	Folio referencia	Concepto	ID. Analista	ID. Cliente	Fecha Entrega	Tipo de Requerimiento (Nuevo, cambio, error, anexo)
1-abr-05	1	Clientes nuevo	Tesis	Crav		Nuevo

Detalle

1. Los datos de cliente que se deben incluir son:

a) Código de cliente asignado en automático con la clave de la tienda que lo dio de alta. Debe ser alfanumérico de 8.
b) Razón social de cliente. Alfanumerico de 60
c) RFC obligatorio es el cliente desea factura. Validación de RFC de acuerdo a formato del SAT.
d) Dirección fiscal (calle y numero Alfanumerico de 60, colonia Alfanumerico de 60, entre calles Alfanumerico de 60, delegación (catalogo de delegaciones), Estado (catalogo de estados), Codigo postal)
e) Numero de vendedor a quien pertenecería el cliente
f) Horario de entrega
g) Si el cliente es de crédito día que desea le hagan la llamada para hacer el pedido

Firmas Revisión

Folio Referencia	Analista	Cliente	Líder Proyecto	Líder cliente
1	Tesis	Cravioto		

8 Glosario de términos

PAS-PAT

Fecha: 25/04/2005

Nombre del proyecto: Proyecto Frog-Cravioto

Glosario de Términos

Revisión histórica

Fecha	Versión	Descripción	Autor
9-abri-2005	1	Glosario inicial	Tesis

AGEB:
Ubicación física en la cartografía que se llama Área geográfica básica

Almacén:
Lugar físico donde se guardan artículos o productos. Generalmente dividido en área, sección, ubicación, pasillo.

Almacenista:
Personal encargado de controlar las entradas y salidas

Autoservicio:
Tienda para venta directa al público en general

Cajero:
Personal que se encarga de recibir los dineros por producto de la venta realizada a través de pedidos, venta directa o liquidación del vendedor.

Cartografía:
Sistema grafico de calles, colonias, estados, municipios, AGEB

Consolidado:
Área de recepción para recibir los pedidos parciales de los subalmacenes

Pedido autorizado:
Pedido que el área de cuentas por cobrar autoriza para ser atendido por el administrador del almacén

Pedido cancelado: Documento registrado por cuentas por cobrar, cancelado por problemas de crédito.
Pedido consolidado: El pedido se encuentra en proceso de reempacamiento, una vez que los subalmacenes ya le entregaron
Pedido de Crédito: Documento otorgado al cliente por los artículos que requiera y no haya excedido su límite y no tenga documentos vencidos.
Pedido de Internet: Son los pedidos que el cliente los registra a través de un usuario y password en la pagina de Internet. Donde además puede ver su estado de cuenta, historial de ventas y sugerido de compra.
Pedido de portafolio: Son todos los pedidos que se realizan directamente en el negocio del cliente y son registrados por el vendedor.
Pedido de telemarketing: Son los pedidos que se realizan a través de que se llame o se reciba la llamada al cliente.
Pedido en administración almacén: El pedido autorizado le aparece a la administración de almacén para su despacho por almacén
Pedido en despacho El pedido lo enviaron de consolidado a despacho. Para su enrutamiento
Pedido enrutado Asignación de los pedidos a un chofer de entrega. Una vez que se los entrego el área de consolidado completos
Pedido en subalmacén: El pedido se encuentra en proceso de preparación por cada subalmacen. Dependiendo del articulo al que pertenezca el subalmacen.
Pedido entregado: Los pedidos que lleva el chofer de ruta, los entrega al cliente. En caso contrario se pregunta la causa del rechazo del pedido.
Pedido facturado: Los pedidos asignados a un chofer de entrega se envían al área de cuentas por cobrar para ser facturados
Pedido pendiente: Documento registrado por ventas y esta en espera de autorizar, cancelar o suspender.
Pedido reprogramado: Los pedidos que no se alcanzaron a entregar se regresan al área de despacho para volverse a asignar a una ruta de entrega.
Pedido suspendido: Documento registrado por cuentas por cobrar y esta en espera por problemas de crédito. Además se registra causa de suspensión.

Subalmacén: División física del almacén, asignando un encargado. Por ejemplo subalmacen de herramientas eléctricas, tubos de cobre
Ticket del pedido: Documento impreso de los artículos y cantidad que requiere el cliente de acuerdo a las existencias disponibles.
Realizado por: Tesis

GLOSARIO

Analista: Es la persona o personas que tratan de determinar la problemática concreta que de solucionar una aplicación y las líneas generales de cómo debe desarrollarse dicha aplicación para resolver la problemática.

Automatizar: Hacer algo automático, es decir sustituir algún proceso por otro y hacerlo más flexible.

Calidad: Índole o manera de que algún producto es bueno. Condición que se pone en algún contrato

Ciclo de desarrollo: También llamado ciclo de vida es el periodo de tiempo que comienza con la concepción del producto del software y termina cuando el producto esta disponible para su uso.
Normalmente, el ciclo de vida del software incluye las fases de análisis, requerimientos, diseño, implementación, prueba instalación, verificación, validación, operación y mantenimiento.

Código: Proceso de descripción de un programa de ordenador en un lenguaje de programación. Transformación del diseño lógico y además especificaciones de diseño en un lenguaje de programación.

Compilador: Es el programa que se encarga de trasladar la fuente a lenguaje entendible por el ordenador, conocido como lenguaje máquina.

Desarrollador: Persona que realiza las líneas de un código llamado programa o sistema.

Diseñador: Persona que diseña la arquitectura y estructura del sistema.

Diseño: Proceso de definición de la arquitectura, componentes, interfaces y otras características de un sistema o de un componente.
El resultado de un proceso.

Eficaz: Adj. Activo, poderoso para producir el efecto deseado.

Eficiente: Acción que se logre el efecto de poder, facultad y aptitud especial.

Factibilidad: Calidad o condición que se pueda hacer.

FDD: (Feature Driven Development.) Metodología ágil de desarrollo. No requiere de un modelo específico de proceso y se complementa con otras metodologías. Enfatiza cuestiones de calidad y define claramente entregas tangibles y formas de evaluación del progreso. FDD consiste en cinco procesos secuenciales durante lo que se diseña y construye en el sistema.

Flexibilidad: Facilidad con la que un sistema o un componente puede modificarse para ser empleado con aplicaciones o en entornos distintos para los que fue construido.

Hardware: Es toda parte física que conforma una computadora: componentes electrónicos, tarjetas, dispositivos, periféricos, etc.

Inducción: Acción de inducir. Método que va de lo particular a lo universal, para buscar establecer leyes, acuerdos o principios generales con base de la observación.
Extraer a partir de determinadas experiencias.

Ingeniería: Conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización de la materia y de las fuentes de energía, mediante inversiones o construcciones útiles para el hombre.

Interfaz: Característica común en la información enviada.. Componente de hardware o software que conecta dos o más componentes con el propósito de trasmitir información entre ellos.

Iterativo: Que tiene la condición de repetirse o reiterarse.

Metodología: Conjunto de métodos que siguen en una investigación, es decir modo ordenado de proceder o hacer una cosa.

MSF: (Microsoft Solutions Framework) Marco para desarrollo de sistemas de software basado en principios, modelos, disciplinas, conceptos, prácticas y recomendaciones propias, derivadas de la experiencia de Microsoft. Sé auto define como "marco" y no como metodología, porque considera que no hay una única estructura de procesos válida para todos los proyectos. El marco MSF se adapta de forma flexible a las características de cada proyecto.

Negocio: Todo lo que es objeto o materia de una ocupación lucrativa o de interés.

Optimo: Cuando algo es muy bueno y no puede ser mejorado.

Periférico: Cualquier componente físico externo al ordenador o ala CPU. Con carácter general suelen diferenciarse n periféricos o dispositivos de entrada, es decir que le envía los datos al ordenador, bien a través de la acción del usuario.

Prioridad: Dìcese de lo que tiene mas importancia respecto a algo.

Procesos: Término empleado para describir los pasos que se llevan a cabo en el sistema. Cuando el ordenador se encuentra en funcionamiento se ejecutan diferentes pasos, la mayoría de las beses en segundo plano, y de manera encubierta para el usuario.

Productivo: Que es capaz y tiene la virtud de producir.

Programadores: Personas que diseñan lógicamente y escribe las líneas de código de un programa de computadora.

Programa: Instrucciones que varían según el lenguaje que se utiliza, pero cuyo fin es el de controlar las acciones que tiene que llevar a cabo el ordenador y sus periféricos.

Prototipo: Versión preliminar de un sistema que sirve de modelo para fases posteriores.

Pseudocódigo: Proceso similar al de la programación pero independiente del lenguaje que se utilice. Es típico en el diseño de algoritmos para su posterior traducción.

Requerimientos: (1) Condición o facultad que necesita un usuario para resolver un problema. (2) Condición o facultad que debe poseer un sistema o un componente de un sistema para satisfacer una especificación, estándar, condición de contrato u otra formalidad impuesta documentalmente.

RUP: (Racional Unified Process) Proceso de Ingeniería del Software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades en las organizaciones de desarrollo de software. Se trata

de un proceso integrado en un producto, desarrollado y mantenido por Racional Software, e integrado en su conjunto de herramientas de desarrollo.

Sistema operativo: Es un conjunto de programas que se podría decir, sirve de enlace entre el ordenador y el programador usuario. Son los responsables de gestionar los recursos del ordenador, disco duro, memorias, control de periféricos, como pantallas etc. De alguna forma unifican y estandarizan el funcionamiento de los ordenadores.

Software: Programa o programas de computadoras.

Subsistemas: Sistema subordinado a otro mayor

Tangible: Algo que se puede tocar, percibir de manera precisa.

Usuario: Cualquier individuo que interactúa con la computadora en el ámbito de aplicación. Los programadores, operadores u otro personal técnico no son considerado usuarios cuando trabajan con la computadora en el ámbito profesional.

XP: (Extreme Programming) Metodología heterodoxa de programación. Es la más popular de las denominadas *metodologías ágiles*. Surgida a partir de la metodología de trabajo empleada Kent Beck, Wark Cunningham y Martín Fowler en el desarrollo del proyecto.

BIBLIOGRAFÍA

Ruble, David A.(1997), Análisis y diseño práctico de sistemas, Ed. Prentice-Hall, Pág 2-10.

Kendall, Análisis y diseño de sistemas, Ed. Prentice-Hall, Cap 11.

Piattini Mario, Villalba José, Mantenimiento de software, Ed. Alfaomega, Pag 64-70.

Fowler Martín, UML gota a gota, Ed. Pearson, Pág. 18-29.

Jacobson, Booch, Rumbaugh, (2000) El proceso unificado de desarrollo de software, Ed. Addison-Wesley.

Referencias en Internet

http://malaadministracion.webspace4free.biz/seleccion_personal/p7_toma_decisiones_hechos.php

http://www.ilustrados.com/secciones/Recursos_Humanos-11.html

http://www.tuobra.unam.mx/publicadas/040920185320.html

http://www.creangel.com/uml/actividad.php#

http://www-gris.det.uvigo.es/~avilas/UML/node46.html

http://members.tripod.com/~gepsea/sistema.htm

http://www.um.es/docencia/barzana/IAGP/Iagp1.html

http://www.ilustrados.com/publicaciones/EpykZkulZZsYzwABxh.php

http://grupo-uml.frcu.utn.edu.ar/proyectos.htm

http://www.geocities.com/chuidiang/metodologia/metodologia.html

http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html

http://www.xprogramming.com/what_is_xp.htm

http://www.nebulon.com/fdd/