



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

03063

13

POSGRADO EN CIENCIA E INGENIERIA DE LA COMPUTACION

VISION PREVENTIVA DE LAS PRUEBAS EN EL SOFTWARE

118062

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRA EN CIENCIAS

P R E S E N T A :

DALILA HERRERA CARRERA

DIRECTORA DE TESIS: DRA. HANNA OKTABA

MEXICO, D. F.

MARZO 2001



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecimientos

Agradezco infinitamente el apoyo de mi directora de tesis Dra. Hanna Oktaba, quién en todo momento estuvo a mi lado apoyándome con su amplísima experiencia.

Muchísimas gracias a la M. En C. Guadalupe Ibargüengoitia, al Dr. Fernando Gamboa y al Dr. Agustín Gutiérrez por su gran apoyo durante la revisión de esta tesis.

Un reconocimiento especial a la Dra. Mariko Nakano por el esfuerzo que significó para ella el pasearse en el mundo de las pruebas del software.

Muchas gracias a la Dra. María Garza, Lulú y Violeta por brindarme todos los recursos que requerí en la maestría... y por su amistad.

Gracias Marco por caminar conmigo hombro con hombro y complementarnos con cada logro individual y festejarlo como propio.

A mis padres y mis hermanos, quienes están presentes en cada éxito de mi vida.

Índice

Introducción	1.
Capítulo Uno: Elementos básicos de la prueba.	6.
Capítulo Dos: Planeación de la prueba.	15.
Capítulo Tres: Técnicas para la prueba de requisitos.	26.
Capítulo Cuatro: Técnicas para la prueba del diseño.	37.
Capítulo Cinco: Caso práctico. SINSAS.	46.
Conclusiones	60.
Referencias	62.
Bibliografía	63.
Anexo A	64.
Anexo B	76.
Anexo C	80.

Introducción

Esta investigación tiene como objetivo crear conciencia sobre la importancia de las pruebas en el desarrollo de un software, no solo como una actividad que forme parte del ciclo de vida del software, sino como un proceso independiente que se realice de manera paralela al desarrollo del software.

El esfuerzo principal que se realizó en esta tesis, fue lograr constituir una guía de fácil entendimiento para la planeación del proceso de prueba desde un enfoque preventivo, orientado a las organizaciones que se encargan al desarrollo de software. La aplicación de esta guía a los proyectos, permitirá que estas empresas disminuyan los recursos dirigidos a la prueba y verificación del software. El proceso de prueba no permanece estático, ya que va madurando cuando éste es aplicado a cada proyecto y de esta forma se van integrando las características únicas de cada organización.

Las actividades del proceso de prueba que aquí se presentan, forman parte de la experiencia y observaciones de expertos en pruebas. La aportación fundamental de esta tesis, es que a partir de esas experiencias y observaciones, se hayan podido constituir técnicas de prueba detalladas y explicadas de manera sencilla, generar una técnica de seguimiento durante la aplicación de éstas con ayuda de diagramas de actividad e identificar los productos – y la manera de obtenerlos –, que deberán ser generados durante las actividades de prueba.

La aplicación de esta guía a un proyecto real, permite identificar los riesgos que se pueden presentar durante el proceso del desarrollo y puesta en marcha de un software, y planear la mejor forma de evitarlos o eliminarlos. Esto ayudará a generar software de mejor calidad, así como reducir los costos de búsqueda, verificación y corrección y/o mantenimiento del mismo. Por otro lado, el cliente también se verá beneficiado al estar respaldado en la adquisición de un software confiable, que genera un proceso de prueba cada vez más fortalecido.

Este documento también tiene como propósito, mostrar las ventajas de desarrollar un proceso preventivo de la prueba en una organización, guiar teórica y esquemáticamente en sus actividades y los productos que serán obtenidos; así como recomendar a los integrantes con que el equipo de prueba debe de contar y las obligaciones que ellos deben de realizar.

La visión actual de las pruebas.

La prueba en el software se ha ido mejorando desde sus inicios – en la década de los 60's –, hasta nuestros días. Antes de los 90's, la prueba se constituía de dos etapas: la etapa de **demostración**, en donde se definían actividades para mostrar

que el software estaba listo para su uso; y la etapa de **detección**, en donde las actividades se planeaban para identificar y corregir errores, defectos y deficiencias en el software que se encontraba en proceso de liberación.

A partir de los 90's el concepto de prueba se modernizó, al desarrollarse una metodología de prueba con etapas bien identificadas. Se buscó que las actividades de estas etapas se definieran lo más formalmente posible y se agregó una tercera: la etapa de **prevención** [1].

La etapa de **prevención**, nos ayuda a definir de manera consistente los requisitos del software, así como la detección de riesgos y errores en las fases de análisis y diseño. De esta forma, los resultados obtenidos en la etapa de **prevención**, nos permiten dar seguimiento en la aplicación de la metodología de la prueba, en las etapas posteriores del desarrollo del software.

Expertos en pruebas coinciden en la importancia de planear detalladamente la etapa de **prevención**, y es que aproximadamente el 56% de los defectos que se presentan en el desarrollo del software, son introducidos por confusiones, omisiones o inconsistencias en los requisitos. El 27% es causado por defectos en el diseño, el 7% por defectos en la codificación y el 10% por problemas en otras actividades [2].

Acerca de los capítulos posteriores.

En esta investigación, se presentan las actividades de prueba que se llevan a cabo en la etapa de **prevención**, la cual se constituye de cinco capítulos y tres anexos. Los primeros cuatro capítulos son la teoría y un anexo en donde se integran los diagramas que modelan las actividades de la prueba. El capítulo Cinco presenta los resultados obtenidos al aplicar las técnicas de prueba propuestas a un proyecto real. Los dos anexos restantes contienen los documentos que fueron generados en este proceso. A continuación se detallan el contenido de los capítulos posteriores.

Capítulo Uno: Elementos básicos de la prueba. En la sección 1.1., se presentan los conceptos y definiciones de los elementos que constituyen la prueba. La sección 1.2. muestra los componentes de una metodología de la prueba, que será referida en los capítulos posteriores. La sección 1.3. presenta las tres visiones del proceso de prueba y en la sección 1.4. se define la visión de la etapa preventiva de las pruebas. Finalmente la sección 1.5., se presenta los roles de los integrantes de un equipo de prueba y sus obligaciones.

A partir del Capítulo Dos hasta el Cuatro el formato del documento cambia, ya que presentan una serie de actividades de prueba, tomando como base la

metodología de prueba definida en la sección 1.2. del Capítulo Uno. El formato que presenta consta de:

- ✓ Un texto introductorio: Es una breve introducción al capítulo y se listan las actividades que son desarrolladas en éste.
- ✓ La sección de *Actividad*: Identifica las actividades que se presentan en el capítulo, con respecto a la metodología de la prueba definida en la sección 1.2 del Capítulo Uno.
- ✓ La sección de *Objetivo*: Presenta el objetivo de las actividades que se presentan en el capítulo.
- ✓ La sección de *Descripción General*: Resume las actividades que van a ser descritas en el capítulo.
- ✓ La sección de *Definiciones de Conceptos*: Se definen los términos que son utilizados en el capítulo.
- ✓ La sección quinta: Presenta el desarrollo de las actividades.

La siguiente tabla presenta las actividades que se llevan a cabo en la etapa preventiva de las pruebas en el software, con base en la metodología de prueba propuesta:

<i>Fases de la Metodología</i>	<i>Capítulo de este documento</i>	<i>Actividades que se llevan a cabo</i>
I. Planeación de la prueba.	Capítulo Dos: Planeación de la prueba.	1. Desarrollo de una estrategia de prueba. 2. Desarrollo de un plan de prueba.
II. Prueba de los Requisitos.	Capítulo Tres: Técnicas para la prueba de requisitos.	1. "Entendiendo lo que se va a probar". 2. Apoyada en el diseño de casos de prueba basados en requisitos. 3. Apoyada en una matriz para la validación de requisitos 4. Apoyada en prototipos o modelos. 5. Apoyada en una lista de verificación.
III. Prueba del Diseño.	Capítulo Cuatro: Técnicas para la prueba de diseño.	1. Apoyada en un análisis alternativo. 2. Apoyada en el diseño de casos de prueba basados en el diseño. 3. Apoyada en modelos de prueba. 4. Apoyada en una lista de verificación.
	Capítulo Cinco: Caso práctico, SINSAS.	Se presentan los resultados de la aplicación de la prueba en un proyecto real

Capítulo Dos: Planeación de la prueba. Se explica la forma de desarrollar una estrategia de prueba, que nos permita identificar los riesgos inherentes del desarrollo del software. También se presenta la manera de crear un plan de pruebas – a partir de una estrategia de prueba –, el cual nos apoye en la administración de las actividades de la prueba en todo el proceso de desarrollo del

software. Las actividades presentes, son modeladas en un diagrama de actividad para facilitar su entendimiento – anexo A.

Capítulo Tres: Técnicas para la prueba de requisitos. Se presentan cinco técnicas para probar los requisitos, se detallan los pasos que hay que llevar a cabo y se definen los productos finales. Cada técnica se modela en un diagrama de actividad para facilitar su entendimiento – anexo A.

Capítulo Cuatro: Técnicas para la prueba del diseño. En este capítulo se presentan cuatro técnicas para la prueba del diseño, y al igual que en el capítulo anterior, se explica la forma de aplicarlas y los productos que deberán ser obtenidos. También cada una de las técnicas se modelan en un diagrama de actividad para su fácil entendimiento – anexo A.

Capítulo Cinco: Caso Práctico: SINSAS. Se presenta la experiencia adquirida en la aplicación de la fase **preventiva** de la prueba, a un proyecto real. Se anexan los documentos del plan de prueba – anexo B –, los productos que se obtuvieron en cada una de las actividades de prueba que se llevaron a cabo, integran el anexo C.

Lectura de los diagramas.

Las técnicas de prueba se encuentran modeladas en diagramas de actividad. Los diagramas muestran las obligaciones de los roles, el orden de las actividades de prueba y cada uno de los productos que son obtenidos durante la aplicación de la técnica. Los diagramas de actividad son diagramas *UML (Unified Modeling Language)* [3] que se utilizan para modelar aspectos dinámicos de los sistemas.

A continuación se presentan los elementos que constituyen los diagramas de actividad y la forma de interpretarlos.

Símbolo	Significado
●	Es el inicio del diagrama de actividad.
→	Indican el flujo de las actividades, la punta de la flecha nos indica la siguiente acción que se va a realizar y la parte posterior, cual fue la actividad anterior.
(acción)	Muestra la actividad que se va a realizar. El texto que se encuentra en el interior es la acción que se debe llevar a cabo.

Símbolo	Significado
	Representa un producto que ha sido generado por una actividad. Se coloca la etiqueta del producto en <i>etiqueta</i> y en Producto se coloca el nombre del producto. Cuando a un producto debe pasar por varias actividades antes de ser terminado, se le coloca una etiqueta entre corchetes al salir de la actividad, para saber en que estado se encuentra.
	La punta de la flecha señala el producto que ha sido generado y la parte posterior muestra, cual es la actividad que ha generado el producto.
	La punta de la flecha muestra la actividad que necesita el producto que ha sido generado por alguna otra actividad, para poder llevar a cabo sus acciones.
	Indica que se han terminado de realizar todas las actividades del diagrama.
	Cuando algunas actividades se realizan en paralelo, este símbolo indica que la siguiente actividad será llevada a cabo en serie.
	Muestra una toma de decisiones para la aplicación de una actividad u otra, dependiendo de los resultados obtenidos.

El parámetro de entrada para un diagrama, es un texto con tres entradas:

- ✓ Entrada: Son elementos de entrada que se deben de tener antes de dar inicio a las actividades.
- ✓ Metodología o Técnica: Es la actividad general que se va a modelar en el diagrama.
- ✓ Salida: Son los productos obtenidos después de llevar a cabo las actividades representadas.

Capítulo Uno: Elementos básicos de la prueba.

En la actualidad, las organizaciones que desarrollan software, dedican cada vez mas recursos a las pruebas del software. Sin embargo, en ocasiones éstas desconocen los alcances de la prueba e incluso, los elementos básicos que la constituyen.

En este capítulo se presentan los elementos básicos de la prueba; sus conceptos y definiciones, su meta, las tareas que se realizan durante la aplicación de la prueba, además de algunos mitos y realidades. También se propone una metodología de la prueba, las actividades que la constituyen, y se identifica, cuáles de éstas pertenecen a la fase preventiva de la prueba del software. Finalmente, se definen los integrantes que constituyen al equipo de prueba y sus obligaciones.

Tener conocimiento de la importancia del proceso de prueba y sus actividades, facilita la aplicación de éstas y le agrega valor a los resultados obtenidos, que finalmente se ven reflejados en un aumento en la calidad del software.

1.1. Conceptos y definiciones básicas.

En esta sección se presentan los conceptos y definiciones elementales que se utilizan en el proceso de prueba, así como términos que son usados en los capítulos posteriores. También se encuentran las actividades de prueba que deben estar contenidas en un proceso de prueba y algunas ideas de lo que es la prueba y lo que no es – mitos y realidades.

1.1.1. Definición de prueba.

La prueba es el proceso de encontrar defectos que estén relacionados con un conjunto de criterios determinados o con especificaciones previamente definidas. Las actividades que se realizan son, verificaciones y validaciones continuas, desde el momento en que se define el producto, hasta que éste se da por terminado y validado en una configuración donde el sistema ya este instalado.

1.1.2. Meta de la prueba.

El propósito de la prueba es demostrar que un sistema, software o configuración de un software no funciona, pero lo que no puede demostrar la prueba, es que el software efectivamente funcione. De esta forma, las actividades de la prueba deben de ser definidas con el objetivo de identificar de forma fácil y efectiva los problemas que se pueden presentar y planear como reducirlos o eliminarlos, anticipadamente.

1.1.3. Actividades que son utilizadas en la prueba.

Cualquier actividad que se lleva a cabo para ayudar en la evaluación o medida de algún atributo del software, se considera como una actividad de prueba. Algunas de estas son:

- ✓ Revisiones.
- ✓ *Walk-through*.
- ✓ Inspecciones apoyadas en criterios previamente establecidos.
- ✓ Diferentes análisis que se realizan durante todo el proceso de desarrollo.

La experiencia en los seminarios de prueba que ha impartido William Hetzel [5], permite constatar que el concepto de pruebas que generalmente se conoce, involucra actividades como:

- ✓ Realizar revisiones en los programas.
- ✓ Encontrar problemas en los programas.
- ✓ Buscar la aceptación del programa por el usuario.
- ✓ Asegurar que un sistema se encuentra listo para su uso.
- ✓ Mostrar que un programa funciona correctamente.
- ✓ Demostrar que no se presentan errores en la ejecución del programa.
- ✓ Entender los límites del rendimiento.
- ✓ Evaluar la capacidad de un sistema.
- ✓ Verificar la documentación.
- ✓ Convencer a sí mismo que el trabajo se ha concluido.

Hetzel asegura que todas las propiedades anteriores son correctas, sin embargo, la prueba es una actividad amplia y continua, que se lleva a cabo a través de un proceso planeado detalladamente.

1.1.4. Problemas identificados por la prueba.

A continuación se presentan los tipos de problemas que se pueden encontrar durante la aplicación de la prueba en el software [4]:

- ✓ *Error*: Es una acción humana que produce un resultado incorrecto.
- ✓ *Defecto*: Es un paso incorrecto, proceso o definición de datos en un programa. Son consecuencias de los errores.
- ✓ *Fallas*: Es un resultado incorrecto. El resultado es la manifestación de la falla.

1.1.5. Mitos y realidades de las pruebas.

Algunas actividades que forman parte del proceso de prueba son subestimadas, al ignorar el esfuerzo que se necesita para llevarlas a cabo. A continuación se presentan algunos ejemplos.

1. Mito: *es posible hacer la prueba completa de un software*: Generalmente cuando una persona se encarga de probar un programa, tiene la idea de que han sido identificados y eliminados todos los defectos encontrados. Entonces, supone que el proceso de prueba se ha finalizado. Sin embargo, el número total de posibles pruebas aplicadas a un sistema en el mundo real es esencialmente infinito, y esto se traduce en que tratar de probar todos los casos se convierta en una tarea imposible.
2. Mito: *La prueba es un proceso fácil y sencillo*: El proceso de prueba es un proceso difícil y creativo ya que es necesario entender el sistema de manera clara. En general, un sistema no es ni simple ni fácil de entender. De aquí que las pruebas no sean un proceso simple. Para realizar una prueba es necesario:
 - ✓ Creatividad.
 - ✓ Conocer la información acerca del negocio.
 - ✓ Tener experiencia en la ejecución de las pruebas.
 - ✓ Apoyarse en una metodología.
 - ✓ Planear el proceso de prueba.
3. Realidad: Un buen proceso de prueba debe de ser planeado como un proceso de mejoramiento en el diseño del tipo de prueba y en el establecimiento de los resultados que se desean obtener, para cada uno de los procesos de pruebas. Una razón importante para realizar una prueba, es precisamente, prevenir deficiencias.
4. Realidad: La prueba no es una fase o un paso a desarrollar, sino todo un proceso que se encuentra asociado en cada fase del ciclo de vida del software. Las pruebas se pueden considerar como mecanismos de prevención. Cuando estos se llevan a cabo, podemos descubrir y eliminar los problemas que se presenten durante el desarrollo del software.

1.2. Metodología de la prueba.

Las actividades de prueba se deben realizar junto con el desarrollo del software, para controlar y mejorar la calidad del mismo. Ya que la prueba se constituye de varios procesos difíciles y costosos en tiempo y dinero, la organización debe buscar administrarlos de forma acertada. Por esta razón, es importante estructurar una metodología de prueba, que nos ayude a identificar y documentar el conjunto de actividades que se llevarán a cabo y sus tiempos, para cubrir todos los aspectos del proceso de prueba y definir los responsables que se harán cargo de realizar cada una de estas actividades.

El diseño de una metodología de la prueba, nos permite identificar cuales son las actividades que se van a realizar, planear los tiempos en que éstas deberán ser llevadas a cabo y definir a los integrantes del equipo de prueba que las realizarán.

1.2.1. Criterios básicos de la prueba.

Para poder definir una metodología de prueba, es necesario tomar en cuenta algunos criterios que nos puedan dar una idea general de la complejidad del proceso de prueba que necesita ser desarrollado. Deben ser tomados en cuenta, al menos, los siguientes criterios:

- ✓ *Conocer y entender lo que debe ser probado:* Tomar en cuenta los atributos del software que se deben medir, definir y documentar en general, y planear la forma en que se va a realizar la prueba.
- ✓ *Definir el periodo de duración de la prueba:* Calcular el tiempo que se va a dedicar a la prueba de los requisitos, al análisis, al diseño y a la construcción.
- ✓ *Identificar los criterios para crear un buen proceso de prueba:* Planear como lograr un buen proceso de prueba e identificar cuando una prueba es satisfactoria y cuando no lo es.
- ✓ *Asignar las responsabilidades en las actividades de prueba:* Designar al menos a una persona que se encargue de ejecutar todas las actividades del proceso de prueba y de coordinarlo con el proceso de desarrollo del software.

1.2.2. Metodología de la prueba.

Una metodología se constituye de un conjunto de pasos, fases o tareas que son ejecutadas en un orden preestablecido y que nos permiten alcanzar un fin. La metodología para el desarrollo del software está constituida por fases y cada fase genera productos o entregables que solo pueden ser creados en cada una de las etapas en particular.

El número de etapas presentes en el proceso de desarrollo de software y la cantidad de productos obtenidos, pueden variar de una organización a otra. Sin embargo, las etapas fundamentales que se pueden identificar son:

- ✓ *Análisis:* Determina la factibilidad y la especificación de los requisitos.
- ✓ *Diseño:* Se especifica el diseño general y el diseño detallado.
- ✓ *Implementación:* Aquí se realizan las actividades relacionadas con la codificación, prueba, depuración e instalación del software.
- ✓ *Mantenimiento:* Una vez que el software se ha liberado, puede necesitar ser mejorado o modificado.

En un proceso de desarrollo de software o ciclo de vida del software tradicional, las pruebas forman parte de la etapa de implementación. Esto implica que los programas o módulos que se han terminado de construir, se les aplique la prueba solo hasta antes de ser liberados.

En la nueva visión de las pruebas en el software, las pruebas deben estar presentes en cada una de las fases del ciclo de vida del software. Esto significa,

que el proceso de prueba cuenta con sus propias fases y que las actividades que la constituyen, se llevan a cabo en paralelo con las actividades del desarrollo del software. Vea figura 1.1.

Como la prueba se debe ejecutar en paralelo al desarrollo del software, las actividades de prueba se agrupan también en fases, lo que constituyen *el ciclo de vida de la prueba*, vea la figura 1.2. Al igual que las fases en el ciclo de vida del

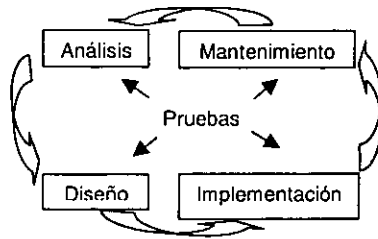


Figura 1.1. Las pruebas integradas en el ciclo de vida del software: El proceso de prueba debe presentar actividades de prueba en las fases de análisis, diseño, implementación y mantenimiento.

software, las fases del ciclo de vida de la prueba dependen de cada organización y de criterios que la organización juzgue importantes. Sin embargo las fases y actividades en el ciclo de vida de la prueba en general son:

1. **Análisis:** Se planean y se definen los objetivos de la prueba y los requisitos de la prueba.
2. **Diseño:** Se especifican las pruebas que van a ser desarrolladas.
3. **Implementación:** Se crea el procedimiento de prueba y los casos de prueba.
4. **Ejecución:** Se aplican las pruebas tantas veces como sea necesario.
5. **Mantenimiento:** Al realizarse los cambios se deben guardar y actualizar las diferentes versiones de las pruebas y sus correspondientes versiones del software.

Una vez definido el proceso de prueba, es importante que tenga mejoras continuas y de esta forma ir construyendo un proceso maduro de prueba. Se recomienda que todas las modificaciones sean documentadas y evaluadas para asegurar que va en incremento la calidad en el proceso.

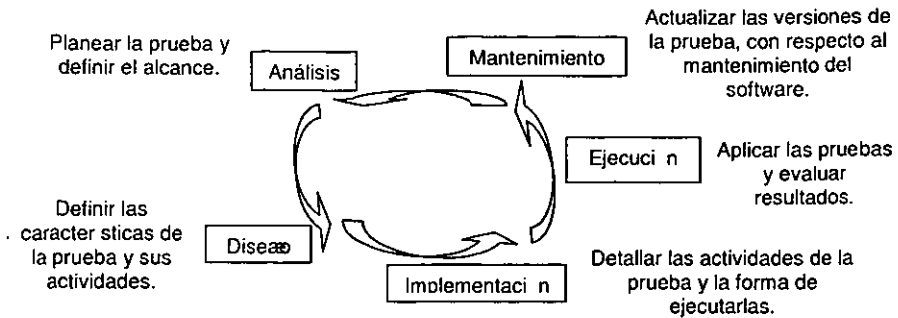


Figura 1.2 Fases del ciclo de vida de la prueba.

Es fundamental que las pruebas sean planeadas, previamente a ser ejecutadas. En las actividades que constituyen al plan de la prueba, deben estar implícitos los objetivos que se persiguen. Para lograr una buena planeación, también es necesario contar con los requisitos del software que se va a desarrollar, para poder diseñar y construir las actividades de la prueba de forma consistente. Recordemos que para lograr un buen desarrollo de software, es menester contar con una buena metodología de prueba.

Una buena metodología de prueba, revisa que el análisis del software este completo antes de que las actividades del diseño se inicien. De igual forma, la prueba asegura que el diseño este completo antes de que la codificación comience y una vez terminada la construcción de los módulos, la prueba permite que la integración del software y la ejecución del software integrado, realice las funcionalidades planeadas y que trabaje de forma correcta.

William Hetzel, que ha dedicado más de dos décadas a trabajar con las pruebas del software, propone la siguiente metodología para integrar las pruebas en cada una de las etapas del ciclo de vida del software [6], también llamado *ciclo de vida de la prueba del software*.

Metodología de la prueba en el software.

- I. **Planeación de la prueba.**
 - Desarrollar una estrategia de prueba.
 - Estimar el esfuerzo de la prueba.
 - Crear un plan de pruebas general – tentativo.
- II. **Prueba de los requisitos.**
 - Establecer la técnica de prueba para los requisitos.
 - Asignar las responsabilidades de la prueba a los diferentes roles.

Obtener el diseño preliminar de los procesos de prueba.

Probar y validar los requisitos.

III. Pruebas del diseño.

Revisar y detallar el plan de pruebas del software.

Probar y validar el diseño.

IV. Pruebas en el desarrollo.

Completar el plan de prueba del software.

Finalizar los procedimientos de prueba y algunas pruebas de código.

Completar los diseños de prueba del módulo o unidad.

Probar los programas.

Integrar y probar los subsistemas.

Llevar a cabo las pruebas del sistema.

V. Pruebas en la implementación.

Llevar a cabo la prueba de aceptación.

Probar los cambios y correcciones.

Evaluar la efectividad de la prueba.

Es importante hacer notar que lo anterior, es una metodología propuesta, que al llevarla a la práctica, será necesario considerar varias características del proyecto del software para garantizar el éxito de la prueba. Algunas consideraciones pueden ser: las características propias de la organización, los recursos del proyecto, el tipo de software que se va a desarrollar, los objetivos de la prueba y cualquier otra situación que la empresa crea importante.

1.3. Las tres visiones de la prueba.

Para asegurar la aplicación de una buena metodología de prueba, es conveniente que puedan ser identificadas dentro de ésta, tres distintas visiones o enfoques: la visión de *prevención*, *detección* y *demostración*. Cuyos objetivos son.

1. *Visión Preventiva*: Aclarar las especificaciones y el rendimiento del software. Proporcionar información que prevenga o reduzca los posibles errores que pudieran suscitarse durante el desarrollo del software, con el fin de ser identificados a tiempo. Estudiar los riesgos y problemas que se puedan presentar y la manera de evitarlos.
2. *Visión de Detección*: Descubrir defectos, errores y deficiencias del sistema. Definir las capacidades y limitaciones del sistema; ofrecer información de la calidad de los componentes, de los sistemas y de los productos obtenidos.
3. *Visión de Demostración*: Incrementar la confianza de que el sistema puede ser usado bajo un riesgo aceptable y controlado. Identificar las características y funciones bajo condiciones y situaciones inusuales. Asegurar que cada componente del software esté completo y listo para su integración.

1.3.1. Visión preventiva de las pruebas.

Las actividades preventivas de la prueba, nos ayudan a identificar de manera anticipada los riesgos que se pueden presentar en la construcción y liberación del software.

Para realizar las acciones preventivas de la prueba es necesario llevarlas a cabo de una manera sistemática. Para ello se debe asegurar que estas actividades se encuentren contempladas dentro de la metodología de la prueba. Las actividades iniciales de una metodología de prueba son: Planear las actividades de la prueba, definir las estrategias para identificar los riesgos en el proceso de prueba y la forma de evitarlos o corregirlos.

Expertos coinciden en que más de la mitad de los errores que se presentan durante el desarrollo del software, son introducidos en la fase de requisitos. El costo de los errores es mínimo, cuando éstos son detectados en la misma fase donde fueron introducidos y un proceso efectivo de prueba previene la migración de errores desde cualquier fase de desarrollo a las subsecuentes fases.

El definir actividades útiles que nos ayuden a prevenir los problemas en el ciclo de vida del software, permitirá encontrar defectos en los requisitos y en el diseño de manera anticipada. En los siguientes capítulos se presentan diferentes técnicas de prueba que se deben llevar a cabo en la etapa preventiva de la prueba del software.

1.4. Definición de Roles.

Un equipo de pruebas en el mundo real, puede estar integrado de una sola persona que realice diferentes roles o un rol para cada persona. La cantidad de integrantes del equipo de prueba, dependerá de cada organización, de la magnitud del proyecto y las actividades de la prueba que se van a llevar a cabo. A continuación se presentan algunos roles que generalmente se encuentran en un equipo de prueba.

1.4.1. Diseñador de la prueba.

El propósito del diseñador de la prueba es planear la prueba, es decir, él decide las acciones que se deben de realizar para alcanzar la meta de la prueba y define el calendario de las actividades de la prueba. Además se encarga de seleccionar y describir los casos de prueba y sus respectivos procedimientos de prueba. El diseñador de la prueba no tiene la tarea de aplicar las pruebas, ya que solo se ocupa de preparar y evaluar las pruebas.

1.4.2. Ingeniero de prueba.

El ingeniero de prueba se encarga de identificar cuales características de los casos de uso deben ser confirmadas en el proceso de prueba, planea y construye los casos de prueba y los procedimientos de la prueba. El ingeniero de prueba también se encarga de aplicar las pruebas y evaluar los resultados. Para facilitar el desempeño en sus actividades se recomienda que él se encuentre familiarizado con el comportamiento externo del software y su funcionamiento.

1.4.3. Equipo de prueba.

El equipo de prueba se puede constituir, además de los dos roles presentados anteriormente, con personas que realicen alguna actividad dentro del proceso de prueba. El número de integrantes del equipo dependerá del tamaño de la organización, por ejemplo se pueden definir los siguientes roles:

- ✓ Administrador de la prueba: Coordina las actividades de planeación y calendarización en las actividades del equipo de prueba y es el responsable de los resultados obtenidos por el equipo de prueba.
- ✓ Probador del sistema: Se ocupa de aplicar la prueba al software para la obtención de un programa ejecutable.
- ✓ Probador de la integración: Se encarga específicamente de llevar a cabo las pruebas de integración. El se encarga de probar el software integrado, verificando la implementación del flujo de trabajo. Algunas empresas delegan esta tarea a los clientes.
- ✓ Ingeniero de componentes: Se encarga de automatizar algunas de las actividades de la prueba. Cuida la integración e interfaces de los subsistemas, para que la integración del software se lleve a cabo correctamente.

El equipo de prueba se encarga de revisar en conjunto, las actividades realizadas por cada uno de los diferentes roles del equipo de prueba. Se discuten los resultados obtenidos, se realizan las modificaciones necesarias y se crea un informe final para la prueba que se aplica a las diferentes etapas del ciclo de vida del software.

Capítulo Dos: Planeación de la prueba.

Llevar a cabo una planeación de la prueba, nos permite identificar los conflictos que se pudieran presentar durante el desarrollo del software y puesta en marcha del mismo. También nos ayuda a definir una buena estrategia para ejecutar las actividades de la prueba de manera óptima así como a partir de ésta, generar una estimación aproximada de los recursos que serán necesarios durante el ciclo de vida de las pruebas.

La planeación de las actividades de la prueba inicia una vez que se tiene una primera versión de los requisitos. Estas actividades tienen la finalidad de:

- ✓ Generar una estrategia de prueba que identifique el tipo de riesgos que presenta cada una de las etapas del desarrollo del software y la manera de contrarrestarlos.
- ✓ Estimar el esfuerzo de llevar a cabo las actividades de la prueba y en caso necesario, discutir alternativas.
- ✓ Crear una primera versión de un plan general de pruebas, en donde se especifique la manera de administrar las actividades de la prueba.

2.1. Actividad: Mostrar las actividades que se realizan durante la planeación de la prueba.

2.2. Objetivo: Construir un bosquejo general de las actividades de la prueba, del esfuerzo de la prueba y del plan de prueba, para conocer los recursos que se utilizarán en el desarrollo del software, mucho antes de que éste empiece a ser desarrollado.

2.3. Descripción General: La planeación de la prueba contiene las siguientes actividades fundamentales: Identificar los riesgos que se pueden presentar durante el desarrollo del software y los riesgos en la organización cuando éste ya se encuentra en uso. Definir las actividades del proceso de prueba, cuyas acciones están orientadas a disminuir los riesgos identificados y el esfuerzo necesario para lograrlo; y desarrollar un plan de pruebas, que ayude a administrar el trabajo de prueba. En éste se documenta la forma en que se van a definir y coordinar las actividades de la prueba, así como a los roles que estarán involucrados.

2.4. Definición de Conceptos.

2.4.1. Riesgos en el software.

Un riesgo en el software es una condición no deseada en el software que al presentarse, desencadena problemas críticos o pérdidas importantes. Por ello es necesario que los probadores identifiquen qué tipos de riesgos pueden presentarse y la manera de eliminarlos o reducirlos [7]. Los riesgos deben estar identificados y definidos dentro de la estrategia de prueba y se deben organizar en dos categorías:

1. **Factores de prueba:** Son cada uno de los riesgos identificados, con su prueba correspondiente para reducirlos o eliminarlos. También pueden ser funcionalidades importantes con su respectiva prueba detallada. Los factores de prueba son atributos del software, que si son requeridos y no se encuentran, ponen en riesgo el éxito del software y también constituyen riesgos para el negocio. A continuación se presentan algunos ejemplos de factores de riesgo:
 - ✓ *Datos correctos:* Asegurar que los datos de entrada, los datos procesados y los datos de salida generados por una aplicación, sean los correctos y completos.
 - ✓ *Integridad de los archivos:* Los procedimientos de integridad de los archivos, aseguran que sean usados los archivos correctos y que la secuencia en que los datos son almacenados y recuperados de un archivo sea de la manera correcta.
 - ✓ *Autorización:* Contar con procedimientos para que los datos que sean procesados se realicen de acuerdo a las reglas del negocio.
 - ✓ *Continuidad en el procesamiento:* Garantizar que el procesamiento continúe, aún cuando haya ocurrido algún problema. También asegura que los procedimientos necesarios y la información de respaldo este disponible en el momento que se deba recuperar la operación.
 - ✓ *Niveles de servicio:* Asegurar que los resultados deseados estén listos dentro de un período de tiempo aceptable para el usuario.
 - ✓ *Control del Acceso:* Garantizar que la información generada por la aplicación estará protegida por modificaciones intencionales o accidentales, destrucciones o uso indebido.
 - ✓ *Conformidad:* Asegurar que el desarrollo del sistema coincida con la estrategia definida por la organización, políticas, procedimientos y estándares.
 - ✓ *Facilidad de uso:* Conocer el tiempo necesitado para aprender a utilizar el software, entender su funcionalidad, la forma de operar, cuáles son sus entradas correctas y comprender la información de las salidas.
 - ✓ *Mantenimiento:* Minimizar el esfuerzo necesario para localizar y corregir un defecto en el sistema o agregar un requisito omitido.

- ✓ *Portabilidad*: Conocer el esfuerzo necesario para trasladar un programa de una configuración de hardware y/o entorno a otro diferente.
- ✓ *Desempeño*: Conocer la cantidad de recursos computacionales y código requerido por un sistema para ejecutar las funciones que lo constituyen.

2. Riesgos en la organización: Son los riesgos que se introducen en una organización cuando deciden hacer uso de un software dentro de ésta. Por ejemplo, que el software autorice alguna transacción a un usuario que la organización no lo permite; perder la integridad de algún archivo importante el cual no se pueda recuperar; dificultades en el uso del software; el procesamiento de alguna información que no cumple con alguna política de la organización o gubernamental, etcétera.

2.5. Actividades de la prueba en la fase de inicio del desarrollo del software.

En este capítulo se presentan las primeras actividades de la prueba que se deben realizar en la fase de inicio del desarrollo del software:

1. Desarrollo de una estrategia de prueba: La estrategia de prueba nos ayuda a identificar los diferentes riesgos que se pueden presentar en el desarrollo de un software y estimar el esfuerzo necesario para darles solución.

2. Desarrollo de un plan de pruebas: En el plan de pruebas se debe definir la forma en que se van a administrar las actividades de la prueba.

2.5.1. Desarrollo de una estrategia de prueba.

Objetivo

Identificar los riesgos en el proceso de desarrollo e instalación de un software, asociarlos en las etapas del proceso de prueba correspondiente y obtener una estimación del esfuerzo de la prueba.

Descripción General

La estrategia de prueba nos ayuda a identificar los riesgos que pueden surgir en el desarrollo de un software y los riesgos que se presentan en una organización al hacer uso de éste e identificar las actividades que nos permitan reducirlos. Con este análisis se obtiene una matriz de prueba donde se documenta toda esta información y a partir de los datos registrados en la matriz, se busca poder estimar el esfuerzo necesario del proceso de prueba.

Roles involucrados

- ✓ El *equipo de prueba* define y clasifica los factores de prueba.
- ✓ Los *clientes y usuarios clave*, identifican los riesgos que puede traer consigo el uso del software en su organización.
- ✓ El *grupo de desarrollo de software* identifica las fases del ciclo de vida del software donde se encuentran los factores de prueba.
- ✓ El *ingeniero de prueba* crea una matriz de prueba con las fases de prueba y los factores de prueba ya identificados.
- ✓ El *grupo de desarrollo de software* estima el esfuerzo de la prueba con el apoyo de la matriz de prueba.
- ✓ El *equipo de prueba* evalúa los resultados y genera un informe.

Entradas

- ✓ Requisitos del sistema.
- ✓ Especificación técnica para el desarrollo y puesta en marcha del sistema.
- ✓ Metodología del proceso de prueba.

Pasos

Una técnica que apoya a la definición de una estrategia de prueba, consiste en construir una matriz de prueba donde se encuentren listados y organizados los factores de prueba y los riesgos en la organización *versus* las fases del ciclo de vida del software, en donde estos se pueden presentar. A continuación se presentan los pasos para obtener una matriz de prueba:

1. Los *clientes y usuarios clave* identifican los riesgos que se pueden generar en su organización durante el desarrollo y/o uso del software.
2. El *equipo de prueba* junto con el *grupo de desarrollo de software*, discuten y registran los factores de prueba y los riesgos que se pueden presentar durante el desarrollo del software.
3. Una vez documentados los riesgos del software y los factores de prueba, el *ingeniero de prueba* los selecciona y clasifica.
4. Con el documento generado en el punto anterior, el *grupo de desarrollo de software* identifica las fases del ciclo de vida del software en donde estos se puedan presentar.
5. El *ingeniero de prueba* crea la matriz y registra los riesgos relacionados con el uso y desarrollo del software, así como los factores de prueba documentados anteriormente.
6. Con la matriz ya terminada el *equipo de prueba* termina de planear la estrategia de prueba donde se detallan algunos puntos que así se precisen.
7. Apoyándose de la estrategia de prueba, el *grupo de desarrollo de software* estima el esfuerzo de la prueba.

8. El *equipo de prueba* integra el informe con la matriz de prueba y el esfuerzo de la prueba, incluyendo los resultados de sus revisiones.

Construcción de la matriz

En la primera columna de la matriz de prueba se colocan los factores de prueba y los riesgos en la organización que fueron identificados. Las siguientes columnas son de cada una de las etapas del ciclo de vida del software en donde se aplica la prueba. El número de renglones depende de la cantidad de factores de prueba y de riesgos que se encontraron; vea figura 2.1. Los pasos para la creación de una matriz de prueba son los siguientes:

1. Los factores de prueba se colocan en la primera columna de la matriz de prueba en orden de mayor a menor importancia. Se recomienda que se listen alrededor de diez factores de prueba más importantes, los de menor importancia pueden ir incluidos dentro de estos. Vea figura 2.1.
2. Los riesgos en la organización se clasifican como altos, medios y bajos y en este orden serán integrados a la matriz de prueba. Son colocados en la misma columna en donde se encuentran los factores de prueba, separados por un título que los distinga. Vea figura 2.1.
3. Las fases del ciclo de vida del software en donde se identificaron los riesgos se colocan a partir de la segunda columna de la matriz de prueba en adelante y se registra en el primer renglón, el nombre de la fase del ciclo que se identificó. Vea figura 2.1.
4. Para registrar los riesgos en la matriz, cada uno de los factores de prueba y riesgos en la organización se relacionará con una o más etapas del ciclo de vida del software en donde será atendido. En la casilla donde coincidan se puede colocar una marca si es que se entienden las acciones que se deben de llevar a cabo, una breve explicación o el folio de un documento en donde se especifique lo que se va a realizar. Vea figura 2.1.

Ejemplo de una matriz de prueba: A continuación se presenta una matriz de prueba con algunos factores de prueba y riesgos en la organización con sus correspondientes fases del ciclo de vida del software:

✓ **Factores de prueba:**

- I. Elementos completos.
- II. Elementos correctos.
- III. Integridad de datos.
- IV. Control de acceso.
- VI. Facilidad en el mantenimiento.

✓ **Riesgos en la organización:**

- I. Permitir diferentes niveles de seguridad.
- II. Procesos que eviten la pérdida de información importante.
- III. Procesos que deben cumplir con políticas gubernamentales.
- IV. Generar bitácora del uso del software por cada usuario.

- ✓ **Fases de prueba:**
 1. Requisitos,
 2. Diseño,
 3. Implementación y
 4. Mantenimiento.

Fases del ciclo de vida del software

Factores de prueba	Requisitos	Diseño	Implementación	Mantenimiento
Elementos completos.	✓	✓	✓	
Elementos correctos.	✓	✓	✓	
Integridad de archivos.			✓	
Control de acceso.		Folio D7	✓	
Facilidad en mantenimiento.		✓	✓	✓
Riesgos en la organización				
Permitir diferentes niveles de seguridad.	Folio R15	✓	✓	
Procesos que eviten la pérdida de información importante.		✓	✓	
Procesos que deben cumplir con políticas gubernamentales.	Folio R9	✓	✓	
Generar bitácora del uso del software para cada usuario.	✓	Folio D45	✓	

Figura 2.1 Matriz de prueba.

La matriz de prueba contiene la información de los objetivos que deben cumplir las pruebas y las etapas del ciclo de vida del software donde se van a llevar a cabo. Con esta información se pueden conocer las características importantes que debe tener el software, en el momento de la recolección de requisitos. Esto permite que los requisitos del software confusos, no definidos, no considerados o difíciles de implementar, sean identificados con anticipación. La información en la matriz de prueba puede sufrir tantos cambios como sean necesarios, sin que esto afecte en el desarrollo del software.

En el caso que no sea necesario detallar las actividades de prueba que van a ser llevadas a cabo, se recomienda colocar una marca cuando éstas han sido realizadas – marca ✓. Cuando las actividades de la prueba son más complejas, se recomienda generar un documento – o varios –, en donde se explique claramente las actividades que se van a llevar a cabo y que éste se incluya en el informe que contiene a la matriz de prueba. De igual forma, solo se registra el folio en la matriz, cuando las actividades ya se han realizado – Folio D45. Vea la figura 2.1.

Para poder completar la matriz de prueba, es necesario discutir sus posibles elementos en varias sesiones. Todas las personas que participen en esta actividad, así como los acuerdos, información técnica, observaciones, etc.,

deberán ser documentadas para su revisión posterior. Una vez terminada la matriz de prueba, se cuenta con información para crear un plan en donde se detallen cada una de las actividades de prueba, así como el orden en que se llevaran a cabo, la fecha, los productos generados en cada actividad, los responsables involucrados y alguna otra información relevante.

Estimación del esfuerzo de la prueba

Una vez que la matriz de prueba ha sido creada, ya se han identificado las etapas del desarrollo del software en donde se realizarán las pruebas y una primera aproximación del proceso de prueba. Con esta información, el *ingeniero de prueba* debe desarrollar un plan en donde se detallen las actividades de la prueba y se defina la forma en que se preparará el entorno de ésta. También el *ingeniero* debe conocer la forma en que se deben evaluar los resultados de la prueba. A partir de esto, se podrá generar un documento donde se estime el esfuerzo que necesita invertir el equipo de prueba y demás roles involucrados en el desarrollo de la misma. El esfuerzo estimado dependerá además de lo anterior, de factores tales como el tipo y tamaño del software, el uso que el cliente le dé, el tipo de cliente que solicita el software, los recursos que destinará el cliente para el desarrollo del software, las políticas que deben incluirse en el software y la cantidad de recursos que dedique la empresa que desarrolla el software.

Modelo gráfico

Figura A.1 del anexo A. Diagrama de prueba: (Entrada *req*: **Requisitos**,
esp: **Especificación Técnica**,
Metodología *mpp*: **Metodología Del Proceso De Prueba**,
Salida *inf*: **Informe**).

Salidas

- ✓ Informe con la estrategia de prueba y el esfuerzo de la prueba documentados. El informe deberá tener la fecha, nombre y firma del responsable que desarrollo el plan de pruebas y de los que participaron en las actividades anteriores.

2.5.2. Desarrollo de un plan de pruebas.

Objetivo

Definir una estrategia para planear y administrar la prueba en cada una de las etapas del ciclo de vida del software y las actividades que se llevarán a cabo durante todo el proceso de la prueba.

Descripción General

El plan de pruebas es un documento – o series de documentos –, que contiene los objetivos de la prueba. En él se establece la estrategia para el proceso de la prueba y contiene la planeación en detalle de todas las etapas y las actividades de la prueba.

Roles involucrados

- ✓ El *equipo de prueba* planea, discute y evalúa diferentes versiones del plan de prueba hasta obtener el plan de prueba detallado y corregido.
- ✓ El *diseñador de la prueba*, genera un diseño del plan de prueba general y detallado.
- ✓ El *ingeniero de prueba* realiza todas las actividades que involucran la creación y desarrollo de un plan de pruebas. Realiza modificaciones, registra los acuerdos y cambios que puedan presentarse durante todo el proceso del desarrollo del plan de pruebas.

Entradas

- ✓ Requisitos del sistema.
- ✓ La estrategia de prueba obtenida en el informe del paso anterior.
- ✓ Especificación técnica para el desarrollo y puesta en marcha del sistema.

Justificación

Con el plan de prueba podrán ser dirigidas y controladas las actividades de la prueba. Al inicio del proyecto los elementos que constituyen al plan de prueba, solo son un esbozo que se irá detallando y revisando durante el desarrollo del sistema hasta constituir una guía que permita dirigir los esfuerzos de la prueba de una forma planeada. La forma en que se lleva a cabo un plan de prueba, depende de cada organización y cada proyecto en particular. Hay que tomar en cuenta que sin un plan de pruebas se corre el riesgo de generar un caos al realizar la prueba a un sistema, sin saber donde, como o cuando comenzar, continuar y terminar. Sin embargo, el hecho de planear una prueba no garantiza su éxito.

El plan de pruebas debe ser revisado por el grupo de personas que lo definió, para ser discutido detenidamente y así poder ir detallando cada una de las actividades que se han acordado. Toda la información que es generada al detallar las actividades debe ser documentada. Estos documentos pueden colocarse como anexos al plan de prueba o como un conjunto de documentos independientes. Cuando se decide tener los documentos por separado, se debe mencionar las etapas que van a ser detalladas dentro de éstos; tales como, las pruebas de integración, pruebas del sistema, los elementos para la aceptación de pruebas, etc. Una buena estrategia es separar los documentos por casos de prueba

individuales o por grupos de casos de prueba. A continuación se presentan las actividades que contiene un plan de pruebas típico.

Contenido de un plan de pruebas típico

1. Estrategia de prueba / Esquema.
 - ✓ Definición de la prueba en el ciclo de vida.
 - ✓ Planteamiento general de la prueba.
 - ✓ Principales herramientas y técnicas a ser empleadas.
 - ✓ Definición de componentes y elementos de prueba o constructores.
 - ✓ Procedimiento para revisar y expandir las revisiones.
2. Diseño de prueba y técnica de implementación.
 - ✓ Organización completa de casos de prueba.
 - ✓ Objetivos y criterios de pruebas.
 - ✓ Matriz de validación de requisitos.
 - ✓ Técnicas de diseño de pruebas.
 - ✓ Información de datos de prueba.
3. Responsabilidades.
 - ✓ Organización de pruebas.
 - ✓ Recursos y planeaciones.
 - ✓ Tareas y asignaciones.
4. Resultados de las pruebas.
 - ✓ Documentos de prueba.
 - ✓ Software que dé soporte a las actividades de la prueba.
 - ✓ Reportes de prueba.
5. Procedimientos.
 - ✓ Procedimientos de prueba.
 - ✓ Reporte de los problemas y sus ajustes.
 - ✓ Procedimientos para controlar las pruebas que se ejecutan éstas mismas, en varias ocasiones*.
6. Controles.
 - ✓ Cambios.
 - ✓ Reportes e información.

La contribución fundamental del plan de pruebas, es la definición del ciclo de vida de pruebas. Esto hace que en cada etapa, se conozcan cuales serán los productos o entregables y quiénes son los responsables de generarlos. Cada uno de éstos representa una tarea importante o una señal que nos indica el momento en que debe de concentrarse el mayor esfuerzo en la prueba.

* es la explicación del término *retesting*, utilizado en lengua inglesa.

Pasos para definir un plan de pruebas

1. Con la estrategia tentativa obtenida en la actividad de prueba anterior, el *diseñador de la prueba* realiza un bosquejo de diversas estrategias alternativas y es entregada al *equipo de prueba* para su comparación.
2. El *equipo de prueba* debe presentar y discutir abiertamente:
 - ✓ Las diferentes alternativas de la misma estrategia.
 - ✓ Las implicaciones que se presentan el ciclo de vida de la prueba.
 - ✓ Estimular los comentarios de los participantes para obtener un diálogo abierto.
3. El *equipo de prueba* escoge la mejor opción como una estrategia tentativa para definir el plan de pruebas y justificar su elección.
4. El *diseñador de la prueba* genera un análisis de las posibles exigencias que se puedan demandar para:
 - ✓ Investigar cuáles son las posibles implicaciones de cada estrategia tentativa.
 - ✓ Conocer como puede ser afectada cada fase de la prueba por la estrategia tentativa que se ha elegido.
5. El *ingeniero de prueba* revisa la estrategia tentativa basándose en lo discutido y analizado, se presentan los resultados al *equipo de prueba* para acordar si la estrategia es la correcta o se busca alguna otra opción. En caso que sea necesario la búsqueda de otra posibilidad, se deberá reiniciar desde el paso 3.
6. El *equipo de prueba* establece las responsabilidades más importantes de la prueba.
7. Una vez que la estrategia para desarrollar el plan de pruebas ha sido aceptada, *equipo de prueba* acuerda cada una de las responsabilidades involucradas:
 - ✓ Discutiendo las responsabilidades en cada fase de la prueba.
 - ✓ Definiendo las actividades de la prueba que se llevarán a cabo.
 - ✓ Identificando la forma en que nos ayudan estas actividades, en la aplicación de la prueba.
 - ✓ Identificando las actividades de la prueba que están relacionadas y la consistencia entre ellas.
8. El *ingeniero de prueba* propone los entregables del proceso de prueba, para:
 - ✓ Identificar la parte del trabajo de la prueba que se ha empezado a realizar
 - ✓ Conocer los productos finales que se van a obtener en las actividades del proceso de prueba.

9. El *ingeniero de prueba* construye y revisa el plan de prueba, discute los resultados con el equipo de prueba y publica el plan de prueba ya revisado. Las posteriores modificaciones al plan de prueba las realiza el *ingeniero de prueba*.
10. El *equipo de prueba* integra un informe con los resultados de las actividades anteriores y los acuerdos importantes.

Modelo gráfico

Figura A.2. del anexo A Diagrama de prueba: (Entrada *req*: **Requisitos**,
ep: **EstrategiaDePrueba**, *esp*: **EspecificaciónTécnica**,
Metodología *mpp*: **MetodologíaDelProcesoDePrueba**,
Salida *inf*: **Informe**, *pp*: **PlanDePrueba**).

Salidas

- ✓ Un plan de pruebas revisado.
- ✓ Un informe con la evaluación del plan de pruebas y con los acuerdos importantes. El informe deberá tener la fecha, nombre y firma del responsable que desarrollo el plan de pruebas y de los que participaron en las actividades para la definición del plan de pruebas.

Capítulo Tres: Técnicas para la prueba de requisitos.

La aplicación de la prueba a los requisitos, es una forma de prevenir los problemas que el software puede presentar en su diseño, implementación, puesta en marcha y mantenimiento. En este capítulo se presentan cinco técnicas de prueba, cuyo objetivo fundamental es obtener una definición completa y consistente de los requisitos. Estas técnicas pueden ser aplicadas en combinación, dependiendo de la complejidad del software.

Una vez que los requisitos del software han sido recopilados, es necesario verificar si son correctos y consistentes, para ello se debe aplicar una técnica de prueba que asegure que la información documentada es suficiente para poder desarrollar el software con las funcionalidades que el usuario requiere. La técnica de prueba permite verificar:

- ✓ Si las características del software se encuentran definidas en los requisitos que se han documentado.
- ✓ Si los requisitos que se tienen documentados se entienden claramente y no son redundantes o contradictorios.
- ✓ Si los requisitos representan las necesidades definidas por el usuario.

3.1. Actividad: Aplicación de la prueba a los requisitos del software.

3.2. Objetivo: Asegurar que las necesidades del usuario están entendidas y documentadas en la definición de los requisitos y si la información documentada es consistente.

3.3. Descripción General: La prueba aplicada a los requisitos se apoya en técnicas que nos aseguran obtener la definición completa y correcta del software y documentar las características acerca de su rendimiento y calidad.

3.4. Definición de Conceptos.

3.4.1. Revisión.

Una revisión es un conjunto de actividades que involucran evaluaciones de aspectos técnicos, que son ejecutados por un grupo de personas que trabajan en conjunto. El objetivo es obtener información confiable de un estado y/o de la calidad de un trabajo. Las revisiones llevadas a cabo con cierto rigor se pueden considerar como una técnica de prueba. Cada revisión debe ser planeada tomando en cuenta:

- ✓ Los criterios que serán probados o revisados.
- ✓ Los resultados esperados y en qué momento se debe detener la prueba.
- ✓ Las personas responsables de llevar a cabo las actividades relacionadas con las revisiones.

El producto generado en cada revisión es un reporte escrito con los resultados obtenidos. Cada uno de los participantes del grupo de trabajo debe ser responsable del impacto que puede causar su informe en el desarrollo de la prueba.

3.4.2. Casos de prueba.

Un caso de prueba es un conjunto de datos de prueba que requiere la prueba como entrada, condiciones de ejecución y datos de salida que se deberán obtener como resultado de la prueba. Es necesario definir el objetivo de la prueba y la forma en que ésta se va a realizar. Los datos de prueba, son datos de entrada que representan las condiciones asociados a un caso de prueba en particular.

3.4.3. Procedimiento de prueba.

Un procedimiento de prueba, es un documento en donde se especifica como se va a preparar el ambiente de la prueba, la forma en que va a ser ejecutada y de que manera serán evaluados los resultados para un caso de prueba en particular. Un caso de prueba puede ser utilizado en más de un procedimiento de prueba, esto depende de la relación que se tenga entre éstos.

3.5. Técnicas para la prueba de requisitos.

Se presentan cinco técnicas para la prueba de requisitos:

- 1. “Entendiendo lo que se va a probar”:** Se revisan los requisitos para verificar si se entienden y a partir de esto planear su prueba posterior.
- 2. Apoyada en el diseño de casos de prueba basados en requisitos:** Se diseñan algunos casos de prueba con información de los requisitos para entender como probar los requisitos.
- 3. Apoyada en una matriz para la validación de requisitos:** Nos ayuda a tener registrados los requisitos, la forma en que van a ser probados y los avances de la prueba.
- 4. Apoyada en prototipos o modelos:** Esta técnica es usada cuando se requiere probar una funcionalidad del software claramente delimitada.
- 5. Apoyada en una lista de verificación:** Es una lista con elementos que ayudan a verificar que los requisitos presenten ciertas características que sean importantes.

3.5.1. Técnica “Entendiendo lo que se va a probar”.

Objetivo

Entender los requisitos del software, verificar si se han definido correctamente y planear como pueden ser probados.

Roles involucrados

- ✓ El *diseñador de la prueba* revisa cada requisito para definir de que forma debe ser probado y lo documenta.
- ✓ El *equipo de prueba* revisa el documento y realiza modificaciones en caso de ser necesario y crea un informe.

Entrada

- ✓ Requisitos del software.

Justificación

Para asegurar que los requisitos se definieron o se entendieron correctamente, es necesario que al leerlos se tenga una idea de cómo se pueden probar. Si existe una duda o confusión sobre la definición de algún requisito, ésta desaparece cuando se trata de definir la forma en que debe ser probado.

Pasos

1. El *diseñador de la prueba* lee cada uno de los requisitos y verifica si se entienden.
2. El *diseñador de la prueba* diseña de qué forma se puede probar el requisito y lo documenta en una lista de *aprobados*.
3. El *diseñador de la prueba* también documenta cada requisito que no se entienda o que no se pueda definir la forma en que se debe probar y explica las observaciones. Todo esto deberá ser documentado en una lista de *no aprobados*.
4. El *equipo de prueba*, revisa la lista de *aprobados* y *no aprobados* y genera un informe de los resultados obtenidos.

Modelo gráfico

Figura A.3. del anexo A. Diagrama de prueba: (Entrada *req*: **Requisitos**, Técnica *nlec*: “Entendiendo lo que se va a probar”, Salida *inf*: **Informe**).

Salida

- ✓ Un informe integrado con dos listas, una lista de los requisitos y la forma en que serán probados y otra de los requisitos confusos o con dudas acerca de su prueba. El informe deberá tener la fecha, nombre y firma del responsable que aplicó la prueba y firma de quien recibió el documento.

3.5.2. Apoyada en el diseño de casos de prueba basados en requisitos.

Objetivo

Entender los requisitos del software, identificar cuáles fallas pueden generarse y expresarlas en el diseño del caso de prueba.

Roles involucrados

- ✓ El *diseñador de la prueba* planea como se prueban los requisitos.
- ✓ El *ingeniero de prueba* construye los casos de prueba, define el procedimiento de la prueba y documenta los resultados obtenidos.
- ✓ El *equipo de prueba* evalúa la prueba y genera un informe con los resultados obtenidos.

Entrada

- ✓ Requisitos del software.

Justificación

Realizar el diseño de algunos casos de prueba cuando se están documentando los requisitos, nos ayuda a definirlos de una forma clara, reconocer las fallas que puedan presentarse y entender como probarlos. La especificación del diseño de la prueba basada en requisitos puede quedar incompleta, al no contar con toda la información del diseño del software y las estructuras de datos. Sin embargo, el propósito es entender los requisitos y planear la forma de validarlos.

Pasos

1. El *diseñador de la prueba*, identifica cada uno de los requisitos y documenta su prueba de manera general.
2. El *ingeniero de prueba* diseña los casos de prueba apoyándose en la información obtenida en la actividad anterior y los documenta.
3. Cuando los requisitos estén confusos o incompletos, el *ingeniero de prueba* debe construir un caso de prueba que identifique los elementos que faltan por definir y que puedan ser verificados con la información que se conoce.

4. El *equipo de prueba*, genera un informe con los documentos obtenidos por el *diseñador de la prueba* y por el *ingeniero de prueba*, junto con las observaciones importantes.

Modelo gráfico

Figura A.4. del anexo A. Diagrama de prueba (Entrada *req.*: **Requisitos**, Técnica *ntec.*: **Apoyada en el diseño de casos de prueba basados en requisitos**, Salida *inf.*: **Informe**).

Salida

- ✓ Un informe con las actividades en general que se deben realizar para probar los requisitos. Los casos de prueba de los requisitos que fueron claros. La descripción que nos permite entender y validar los requisitos que no se encuentran definidos completamente o que no se entendieron. Algunos comentarios, aclaraciones o acuerdos importantes. El informe deberá tener la fecha, nombre y firma del responsable que aplicó la prueba y firma de quien recibió el documento.

3.5.3. Apoyada en una matriz para la validación de requisitos.

Objetivo

Registrar y organizar los requisitos del software, así como llevar el registro de los avances de la prueba para cada requisito.

Roles involucrados

- ✓ El *diseñador de la prueba* define el nombre de las columnas de la matriz y la forma en que se van a organizar los requisitos en los renglones de la misma.
- ✓ El *ingeniero de prueba* construye los casos de prueba y sus procedimientos de la prueba correspondientes. Construye la matriz y documenta la información.
- ✓ El *equipo de prueba* evalúa la prueba y resultados finales.

Entrada

- ✓ Requisitos del software.

Justificación

La matriz para la validación de requisitos nos permite organizar los requisitos y asegurar que se tienen contempladas las pruebas para cada uno de ellos. Esta

estrategia nos ayuda a diseñar los casos de prueba tomando en cuenta cada uno de los requisitos para asegurar la calidad en su definición. Una vez que los requisitos están organizados en la matriz, se facilita la tarea de revisarlos, de registrar el progreso de los requisitos revisados y el progreso en la construcción de los casos de prueba.

Pasos

- 1.. El *diseñador de la prueba* define el nombre de las columnas y el número de renglones de la matriz. Para la definición de los renglones, revisa los requisitos del software y planea la forma en que se van a agrupar, con el fin de no tener una gran cantidad de renglones.
2. El *ingeniero de prueba* construye los casos de prueba de los requisitos, sus procedimientos de la prueba correspondientes y la matriz.
3. El *ingeniero de prueba* documenta los requisitos, registra las referencias de los casos de prueba y cuantos de estos ya han sido construidos.
4. El *equipo de prueba* evalúa la matriz, las actividades de la prueba y genera un informe de los resultados obtenidos.

Construcción de la matriz

La matriz para la validación de requisitos se constituye de cuatro columnas, *Número, Descripción del requisito, Casos de prueba y Terminado*.

- ✓ En la columna *Número*, se enumeran los requisitos.
- ✓ En la columna *Descripción del requisito*, se describe el requisito del software, de manera corta y clara.
- ✓ En la columna *Casos de prueba*, se hace la referencia a los casos de prueba o a las actividades que se han definido para facilitar la prueba.
- ✓ En la columna *Terminado*, sólo se registra el caso de prueba que ya haya sido concluido.

En la columna de *Casos de prueba*, la referencia de los casos de prueba puede ser definida como:

- ✓ Un conjunto de datos que describan completamente la prueba.
- ✓ Un folio de alguna carpeta o documento que contiene la definición completa del caso de prueba.

Modelo gráfico

Figura A.5. del anexo A. Diagrama de prueba: (Entrada *req*: **Requisitos**, Técnica *nbc*: **Apoyada en una matriz para la validación de requisitos**, Salida *inf*: **Informe**, *mat*: **Matriz**).

Salidas

- ✓ La matriz de requisitos verificada por el equipo de pruebas.
- ✓ Un informe con los casos de prueba y los procedimientos de prueba que se han terminado. También debe tener los resultados obtenidos al aplicarse la prueba, la fecha, nombre y firma del responsable que aplicó la prueba y firma de quien recibió el documento.

Ejemplo de una matriz de requisitos.

En la matriz observamos que los casos de prueba 12, 23 y 40, deben terminarse completamente para que la prueba del requisito número 1 esté terminada. Nótese que para el requisito número 2, se encuentra incluidos los casos de prueba 23 y 40. Los casos de prueba, generalmente se utilizan repetidamente para probar la claridad y concordancia de un requisito, además de que estos nos ayudan a probar otros requisitos que se encuentren relacionados entre sí. En el caso del requisito 5, la ausencia del número en los casos de prueba, significa que las pruebas aún no se han desarrollado. La columna de *Terminado* puede ayudarnos a registrar el avance en el desarrollo de los casos de prueba de cada requisito. Vea figura 3.1.

Organización de las pruebas a los requisitos con una matriz de validación de requisitos					
Número	Descripción del requisito	Casos de prueba		Terminado	
1	Proporcionar la capacidad de obtener el historial académico de un alumno en particular.	CP-12 CP-40	CP-23	CP-12 CP-40	CP-23 CP-23
2	Proporcionar la capacidad de registrar los alumnos egresados.	CP-15 al CP-40	CP-23	CP-17 al CP-40	CP-23 CP-23
3	Generar automáticamente una lista de alumnos por grupo al final del ciclo escolar, incluyendo los egresados.	CP-29 CP-38	CP-31	CP-29 CP-38	CP-31 CP-31
4	Generar un informe del promedio de cada materia en un grupo en particular.	CP-38	CP-45	CP-38	
5	Proporcionar la capacidad de administrar a los profesores, relacionándolos con las materias que imparten.				

Figura 3.1. Matriz de requisitos.

La creación de una matriz útil en cada organización, dependerá de la forma en que se apegue al proceso de prueba y a las necesidades del proyecto. Se debe crear un plan de revisión de los requisitos, donde se especifique como se usa la matriz y quien es el responsable de darle el seguimiento. También a las columnas se les puede asignar un diferente uso, por ejemplo:

- ✓ En la primera columna se pueden registrar las actividades terminadas del equipo que se ocupa de los requisitos.
- ✓ La siguiente, se puede utilizar para una segunda revisión de un equipo diferente.
- ✓ En la tercera, un tercer equipo podría colocar una marca que indique que el requisito se ha terminado y liberado, etc.

3.5.4. Apoyada en prototipos o modelos.

Objetivo

Crear un prototipo de una parte del software que se va a desarrollar, con la finalidad de asegurar que los requisitos se entienden y están completos.

Roles involucrados

- ✓ El *diseñador de prueba* planea la prueba que se va a aplicar a los requisitos del prototipo.
- ✓ El *ingeniero de prueba* planea y define el procedimiento de la prueba para el prototipo del software. Construye los casos de prueba, aplica el procedimiento de la prueba y documenta los resultados obtenidos.
- ✓ El *grupo de análisis* realiza el análisis del prototipo.
- ✓ El *grupo de diseño* realiza el diseño del prototipo.
- ✓ Los *constructores* implementan el prototipo.
- ✓ El *equipo de prueba* evalúa la prueba, los resultados obtenidos y genera un informe.

Entradas

- ✓ Documento donde se argumenta el objetivo del prototipo.
- ✓ Requisitos del prototipo.

Justificación

Se recomienda aplicar la técnica *Apoyada en prototipos o modelos* para probar un software cuyos requisitos son difíciles de conocer o que cambian continuamente. Esta técnica prueba partes o piezas de un software que se ha dividido para facilitar su implementación y así realizar la prueba al software en forma incremental. Cada una de las partes probadas nos provee de experiencia para probar la siguiente pieza – a esta pieza se le llama prototipo o modelo, facilitando en gran medida el trabajo total de la prueba y al final lograr un software totalmente integrado y probado. En esta técnica no es necesario conocer claramente todos los requisitos, ni invertir demasiado tiempo en la especificación de los prototipos, ya que al ir

integrando cada pieza del software, se generan cambios en los prototipos ya terminados.

Pasos

1. El *diseñador de prueba* debe conocer y entender el objetivo del prototipo, para poder planear la prueba que se va a aplicar a sus requisitos.
2. El *ingeniero de prueba* debe tener los requisitos del prototipo. Construye los casos de prueba y planea y define el procedimiento de la prueba. Si surge alguna duda sobre las características que debe cumplir el prototipo, el *ingeniero de prueba* se puede apoyar con el documento donde se argumenta el objetivo del prototipo.
3. El *grupo de análisis* también debe contar con los requisitos del prototipo y realiza su análisis.
4. El *grupo de diseño* se encarga de obtener el diseño del prototipo a partir del análisis entregado.
5. El *grupo de construcción*, implementa el prototipo.
6. Una vez terminado el prototipo, se entrega al *ingeniero de prueba* y aplica el procedimiento de la prueba al prototipo y documenta los resultados obtenidos.
7. El *equipo de prueba* evalúa el trabajo de prueba, los resultados obtenidos y genera un informe.

Modelo gráfico

Figura A.6. del anexo A. Diagrama de prueba: (Entrada *req.*: **Requisitos**,
Técnica *nec.*: **Apoyada en prototipos o modelos**, Salida *inf.*: **Informe**).

Salida

- ✓ Un informe con los casos de prueba y el procedimiento de prueba del prototipo. El Informe también debe tener de los resultados obtenidos al aplicarse la prueba final, con fecha, nombre y firma del responsable que aplicó la prueba y firma de quien recibió el documento.

3.5.5. Apoyada en una lista de verificación.

Objetivo

Analizar y evaluar los requisitos en conjunto, apoyándose en una lista de verificación.

Roles involucrados

- ✓ El *diseñador de prueba* define los elementos de la lista de verificación.
- ✓ Los *clientes y usuarios clave* pueden contribuir con algunos elementos para la lista, que son importantes desde su punto de vista.
- ✓ El *probador del sistema* aplica la lista de verificación y documenta los resultados. Este rol también puede realizarlo el *ingeniero de prueba* o algún integrante del *equipo de prueba*.
- ✓ El *equipo de prueba* evalúa la prueba y resultados finales y genera un informe.

Entrada

- ✓ Requisitos del software.

Justificación

La lista de verificación se usa para evitar omisiones en los requisitos y/o recordar las áreas de requisitos comunes, también nos ayuda a encontrar redundancias, contradicciones y verificar la consistencia en los requisitos. Estas listas pueden ser tan generales o tan específicas como se requiera. Cuando una lista de verificación se aplica a los requisitos, es importante realizar un informe en donde se explique los resultados obtenidos y algunas sugerencias para corregir los problemas encontrados.

Pasos

1. El *diseñador de prueba* define los elementos de la lista de verificación. En esta actividad pueden intervenir los clientes o usuarios clave.
2. El *probador del sistema* aplica la lista de verificación en presencia de quien recopiló los requisitos o alguna persona responsable. Registra los resultados en la lista y los discute con el responsable, para despejar cualquier duda que pudiera surgir en ese momento. Integra su informe y lo entrega al *equipo de prueba*.
3. El *equipo de prueba* evalúa la prueba y resultados finales y crea un informe final con observaciones o acuerdos importantes.

Modelo gráfico

Figura A.7. del anexo A. Diagrama de prueba: (Entrada *req*: **Requisitos**, Técnica *nlac*: **Apoyada en una lista de verificación**, Salida *inf*: **Informe**).

Salida

- ✓ Un informe con la lista de verificación, incluyendo los resultados obtenidos y sugerencias para poder corregir los problemas que se hayan presentado. El informe debe tener fecha, nombre y firma del responsable que aplicó la prueba y firma de quien recibió el documento.

Capítulo Cuatro: Técnicas para la prueba del diseño.

Las pruebas del diseño, verifican que el modelado del software contenga las funcionalidades establecidas previamente en la fase de análisis. En este capítulo se presentan cuatro técnicas para la prueba del diseño de software.

Las técnicas para la prueba del diseño, se apoyan en una serie de revisiones para verificar que el diseño:

- ✓ Incluya todos los requisitos del software – previamente probados.
- ✓ Represente las funcionalidades definidas en los requisitos.
- ✓ Sea la forma más fácil de modelar la información de los requisitos.
- ✓ Facilite la implementación.

Es importante que dentro de las actividades de la prueba del diseño, también se incluyan acciones que permitan identificar las posibles fallas que se puedan presentar, durante la implementación del software, al liberarlo y en el mantenimiento.

4.1. Actividad: Aplicación de la prueba al diseño del software.

4.2. Objetivo: Asegurar que el diseño propuesto es una solución correcta y que cumple con los requisitos del software.

4.3. Descripción General: Las actividades de la prueba del diseño, permiten identificar si el diseño ofrece una buena representación de los requisitos del software. Verifica además, que el diseño pueda ser implementado con facilidad, investiga las posibles fallas que se pueden presentar en la implementación y uso del software y como podrían ser eliminadas o controladas.

4.4. Definición de Conceptos.

4.4.1. Métricas en la prueba de diseño.

Las métricas son mediciones que ayudan a predecir la calidad del software. En la prueba del software se debe identificar los elementos importantes en el proceso de prueba y buscar asociarle una métrica. Esto nos ayudará en la toma de decisiones para la asignación de recursos, en el seguimiento del desempeño y en la eficacia de la prueba. En las revisiones del diseño se usan métricas para cuantificar y definir los resultados de la prueba.

Para realizar una medición, podemos apoyarnos en cuestionarios o listas de verificación, identificando antes de aplicarlos cual es la información que se necesita conocer.

Ejemplo: Un grupo de preguntas puede relacionar la integridad del diseño y la seguridad del software. Las preguntas comunes pueden ser como las siguientes:

- ✓ ¿Se controlan en módulos independientes las características de seguridad?
- ✓ ¿Existe alguna huella de auditoría en los accesos de mantenimiento para revisión o investigación?
- ✓ ¿Se encuentran ya definidas las claves y permisos de acceso?
- ✓ ¿Es necesario realizar cambios a diferentes programas para violar la seguridad del acceso?

Cada revisor debe investigar las respuestas de las preguntas anteriores, documentar los resultados y crear un informe donde puedan ser registradas o contabilizadas. Se debe establecer una cota mínima como criterio de paso o falla, para poder definir una métrica de integridad. Los diseños que se encuentran debajo de la cota mínima, se trabajan nuevamente y se les aplica una prueba de revisión adicional aún después de ser aceptados.

4.4.2. La prueba para el diseño.

Las actividades para la prueba del diseño son fundamentalmente, revisar y probar el diseño. Se debe planear el diseño tomando en cuenta las actividades de prueba para facilitarla y simplificarla. Aplicar un buen trabajo de ingeniería se traduce en encontrar posibilidades para realizar una construcción simplificada y que la tarea de verificación pueda ser realizada adecuadamente.

Ejemplo: Cuando se desarrolla un software complejo se recomienda realizar el diseño dividiéndolo en componentes con funcionalidad, que permita observar a los probadores como operan éstos individualmente y verificar si su funcionamiento es correcto. A estos componentes se les llaman “ventanas”. La creación de estas “ventanas” y las revisiones a su diseño, aseguran que se le ha aplicado la prueba a la porción del diseño que constituye la “ventana”. Finalmente cuando cada una de las ventanas son revisadas, se obtiene la prueba del diseño del software por completo

Aplicar la prueba a un diseño complejo, puede no resultar muy efectivo, incluso cuando el desarrollo del software ha contemplado las actividades del proceso de prueba. Es por ello, que este tipo de situaciones deben ser consideradas por los integrantes del equipo de prueba, ya que ellos podrán decidir la cantidad de pruebas óptimas que será necesario aplicar una vez que los requisitos se encuentren ya definidos o en una primera versión.

4.5. Técnicas para la prueba del diseño.

En este capítulo se presentan cuatro técnicas de prueba para el diseño:

1. **Apoyada en un análisis alternativo:** Se identifica la mejor versión del diseño de entre varias y se verifica si el diseño representa correctamente a los requisitos.
2. **Apoyada en el diseño de casos de prueba basados en el diseño:** Se construyen casos de prueba con la información del diseño y se evalúan para verificar que los requisitos se incluyeron y que están representados correctamente.
3. **Apoyada en modelos de prueba:** Se diseña una funcionalidad importante del software y se prueba por separado para asegurar que la prueba cumple con su cometido.
4. **Apoyada en una lista de verificación:** Agrupa partes importantes del diseño para ser verificadas y evitar contradicciones, omisiones o redundancias.

4.5.1. Apoyada en un análisis alternativo.

Objetivo

Revisar diferentes versiones del diseño, para escoger la mejor opción que represente adecuadamente a los requisitos.

Roles involucrados

- ✓ El *grupo de diseño* hace entrega de los diferentes diseños del software y el diseño que ellos escogieron como el mejor.
- ✓ El *ingeniero de prueba*, revisa el diseño y documenta los resultados.
- ✓ El *equipo de prueba* convoca a una serie de reuniones para discutir los resultados de la prueba con el *grupo de diseño* y cuando el diseño está evaluado, genera un informe.

Entradas

- ✓ Requisitos del software.
- ✓ Diseño del software.

Justificación

Cuando se va desarrollando la prueba del diseño, es necesario confirmar que el diseño que se ha escogido es el mejor. Para poder lograrlo, es necesario tener diferentes versiones del diseño y realizar una crítica seria y comparativa entre

estos. Si alguna de las alternativas propuestas, se considera superior a otras, ésta debe ser posible implementarla con los recursos que se tienen.

Sin embargo, la experiencia señala que dentro del ambiente del desarrollo del software, este tipo de análisis raramente se realiza. Las revisiones en el diseño generalmente buscan justificar que lo propuesto va a funcionar, y todas las actividades giran alrededor de esta decisión.

Pasos

1. Para evitar que el *grupo de diseño* trabaje sobre un solo diseño sin buscar más alternativas, el *ingeniero de prueba* revisa el diseño tomando en cuenta los siguientes criterios:
 - ✓ La revisión debe hacerse a un alto nivel y en un periodo no mayor de dos semanas después de que se inició el diseño, y así tener suficiente tiempo para corregirlo o proponer nuevas versiones de diseños.
 - ✓ La revisión puede ser dirigida a encontrar alternativas y tipos distintos de evaluación que nos permitan identificar diferentes formas en que el diseño puede ser mejorado.
 - ✓ Generar un documento con las observaciones obtenidas.
2. El *grupo de diseño* deberá discutir las diferentes alternativas del diseño que se consideraron, pero que fueron rechazadas por alguna circunstancia. Dada la importancia de esta actividad, se propone que se realice esta revisión en dos etapas:

Primera etapa: El *equipo de prueba* debe discutir y entender todas las alternativas consideradas por el *diseñador* y encontrar las posibles ventajas y desventajas de cada una de ellas.

Segunda etapa: La sesión será suspendida para que el *ingeniero de prueba* proporcione algunas alternativas que no se habían considerado hasta ese momento y que él identificó en la etapa anterior. En una reunión posterior, el *equipo de prueba* deberá forzar a que las alternativas sean nuevamente evaluadas, así las decisiones que integran el diseño se volverán a considerar.

3. Los problemas que vayan surgiendo se discutirán en cada sesión y se les podrá dar solución en el momento que aparezcan, así se evita que estos sean corregidos cuando el diseño ya esté muy avanzado.
4. El *equipo de prueba* genera un informe completo con las observaciones obtenidas, discusiones y acuerdos tomados

Modelo gráfico

Figura A.8. del anexo A. Diagrama de prueba: (Entrada *req*: **Requisitos**,

dis: Diseño, Técnica *ntec:* Apoyada en un análisis alternativo,
Salida *inf:* Informe).

Salida

- ✓ Un informe con el resultado de las evaluaciones del diseño del software y los acuerdos importante que fueron tomados. El informe deberá tener la fecha, nombre y firma del responsable que aplicó la prueba y firma de quien recibió el documento.

4.5.2. Apoyada en el diseño de casos de prueba basados en el diseño.

Objetivo

Evaluar el diseño del software, desarrollando casos de prueba con la información que se encuentra en el diseño.

Roles involucrados

- ✓ El *diseñador de la prueba* identifica los casos de prueba más importantes que se deben desarrollar, para que permitan llevar a cabo la prueba del diseño en forma completa y sencilla.
- ✓ El *ingeniero de prueba* revisa el diseño, construye los casos de prueba y documenta los problemas que surjan.
- ✓ El *equipo de prueba* discute los resultados de la prueba con el grupo de diseño, verifica los resultados documentados y genera un informe.

Entradas

- ✓ Requisitos del software.
- ✓ Diseño del software.

Justificación

El diseño de casos de prueba basados en el diseño, es una prueba que se encarga de validar el diseño del software e identificar sus errores y fallas. Los casos se llaman “basados en el diseño”, debido a que la información que los constituye se obtiene del diseño del software. Estos casos de prueba expresan como el diseño debe de manipularse y como debe ser entendido apropiadamente. Cuando los casos de prueba ya se encuentran desarrollados, se pueden utilizar para revisiones del diseño o como *walkthroughs*.

Pasos

1. El *diseñador de la prueba*, identifica cada uno de los datos y rutas de procesos dentro de las estructuras del software y documenta en forma general como pueden ser probados. Esta información es la base de los casos de prueba.
2. El *ingeniero de prueba* analiza como se representan las propiedades del diseño dentro de los casos de uso y lo documenta.
3. El *ingeniero de prueba* construye los casos de prueba con la información de los requisitos y con los datos que se identificaron en el paso 1 y 2. Los casos de prueba se deben centrar en interfaces internas, rutas de datos complejos, procesos complejos, riesgos en el diseño y áreas débiles - entre otras.
4. El *equipo de prueba*, genera un informe con los documentos obtenidos por el *diseñador de la prueba* y por el *ingeniero de prueba*, junto con observaciones importantes.

Modelo gráfico

Figura A.9. del anexo A. Diagrama de prueba: (Entrada *req*:Requisitos, *dis*: Diseño, Técnica *ntec*: Apoyada en el diseño de casos de prueba basados en el diseño, Salida *inf*:Informe).

Salidas

- ✓ Un informe con los casos de prueba basados en el diseño del software y también debe contener los resultados obtenidos y acuerdos importantes. El informe deberá tener la fecha, nombre y firma del responsable que aplicó la prueba y firma de quien recibió el documento.

4.5.3. Apoyada en modelos de prueba.

Objetivo

Construir una representación o modelo simplificado de una propiedad en particular del diseño y aplicar la prueba como método de exploración.

Roles involucrados

- ✓ El *diseñador de prueba* diseña el modelo de prueba.
- ✓ El *ingeniero de prueba* construye el modelo de prueba y lo verifica
- ✓ El *equipo de prueba* evalúa los resultados obtenidos y genera un informe.

Entradas

- ✓ Requisitos del modelo de prueba.
- ✓ Diseño del software.

Justificación

El modelo de prueba generalmente es utilizado para probar configuraciones del diseño de bases de datos, tiempos de respuesta, interfaces de usuario y secuencia de transacciones. La prueba permite que el diseño sea mejorado o modificado a tiempo y optimizar los tiempos de respuesta requeridos, mucho antes que se inicie la programación. Por ejemplo, en casos en que los tiempos de respuesta fallan, esta prueba identifica si el tiempo de entrega de los datos de respuesta es muy lento y las consideraciones que se deben tomar en cuenta para mejorar su rendimiento.

Pasos

1. El *diseñador de prueba* debe conocer y entender qué parte del diseño se va a representar en el modelo de prueba. Una vez determinado el modelo de prueba debe ser estudiado por el *diseñador*, para identificar el objetivo del modelo y así poder planear las actividades de la prueba que le serán aplicadas.
2. El *ingeniero de prueba* construye el modelo de prueba. Las actividades deben ser orientadas hacia la exploración o prueba de la porción del diseño, según sea el objetivo del modelo.
3. El *ingeniero de prueba* debe lograr que la construcción del modelo de prueba, permita introducir por ejemplo, información básica sobre la naturaleza de las bases de datos y accesos que cada transacción necesita. El modelo de prueba debe considerar las limitaciones del factor que involucra el tiempo de E/S, de este modo, el modelo puede "calcular" el tiempo de respuesta esperado para las diferentes transacciones bajo varias estructuras de bases de datos hipotéticas.
4. Cuando el modelo de prueba se ha terminado el *equipo de prueba* evalúa el trabajo de prueba, los resultados obtenidos y genera un informe.

Modelo gráfico

Figura A.10 del anexo A. Diagrama de prueba:

(Entrada *req*: **Requisitos del Modelo**, *dis*: **Diseño**,
Técnica *nlac*: **Apoyada en modelos de prueba**, Salida *inf*: **Informe**).

Salida

- ✓ Un informe con el modelo de prueba y las observaciones importantes sobre la parte del diseño que fue modelado. El informe deberá tener la fecha, nombre y firma del responsable que aplicó la prueba y firma de quien recibió el documento.

4.5.4. Apoyada en una lista de verificación.

Objetivo

Agrupar las observaciones importantes que se deben verificar en el diseño, para que puedan ser analizadas y evaluadas conjuntamente.

Roles involucrados

- ✓ El *diseñador de prueba* define los elementos de la lista de verificación.
- ✓ Los *clientes y usuarios clave* pueden contribuir con algunos puntos importantes desde su punto de vista.
- ✓ El *probador del sistema* aplica la lista de verificación y documenta los resultados. Este rol también puede ocuparlo el *ingeniero de prueba* o algún integrante del *equipo de prueba*.
- ✓ El *equipo de prueba* evalúa la prueba y resultados finales y genera un informe.

Entradas

- ✓ Requisitos del software.
- ✓ Diseño del software.

Justificación

La lista de verificación también puede ser usada para probar el diseño, nos permite recordar las áreas comunes y nos ayuda a agrupar partes del diseño para verificar si existen contradicciones, omisiones o redundancias, como se explicó en la técnica 3.5.5 del Capítulo Tres.

Pasos

1. El *diseñador de prueba* define los elementos de la lista de verificación. En esta actividad pueden intervenir los clientes o usuarios clave, para asegurar que la revisión apoyada con la lista cumpla su propósito.
2. El *probador del sistema* aplica la lista de verificación en presencia de algún integrante del *grupo de diseño* o alguien responsable. Registra los resultados en la lista y los discute con el responsable, para despejar cualquier duda que

podiera surgir en ese momento. Integra su informe y lo entrega al *equipo de prueba*.

3. El *equipo de prueba* evalúa la prueba y resultados finales y crea un informe final con observaciones o acuerdos importantes.

Modelo gráfico

Figura A.11 del anexo A. Diagrama de prueba: (Entrada *req*: **Requisitos**,
dis: **Diseño**, Técnica *ntec*: **Apoiada en una lista de verificación**,
Salida *inf*: **Informe**).

Salida

- ✓ Un informe con la lista de verificación, incluyendo los resultados obtenidos y sugerencias para poder corregir los problemas que se hayan presentado. El informe debe tener fecha, nombre y firma del responsable que aplicó la prueba y firma de quien recibió el documento.

Capítulo Cinco: Caso práctico. SINSAS.

En este capítulo se presenta la experiencia que fue adquirida al aplicarse las actividades que contiene la fase preventiva de la prueba en el desarrollo del software en un proyecto real. Las técnicas utilizadas para estas actividades son las que han sido explicadas en los capítulos anteriores. Los documentos que se muestran, fueron previamente verificados y se encuentran ordenados de la forma en que estos se crearon durante el proceso de la prueba.

5.1. Definición del proyecto SINSAS.

SINSAS es el Sistema de Inscripciones de Aspirantes del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) de la Universidad Nacional Autónoma de México (UNAM), cuyo objetivo es automatizar el sistema de inscripciones de aspirantes al examen de admisión en la maestría y el doctorado.

El proyecto se inició en julio de 1999 y tuvo una duración de cuatro meses, las actividades que se realizaron fueron desde la recolección de los requisitos del software, hasta la liberación y puesta en marcha del mismo.

5.1.1. Grupo de Trabajo

El grupo de trabajo encargado de desarrollar SINSAS, se constituyó de nueve integrantes, la definición de actividades fue decidida en consenso. En la figura 5.1 se presentan los integrantes, cargos y sus obligaciones.

<i>Nombre</i>	<i>Iniciales</i>	<i>Cargo</i>	<i>Actividades</i>
Carmen D. Álvarez Sánchez	CAS	Diseñador de Sistemas.	Actividades de diseño.
Eddie Basto Aguilar	EBA	Administrador de la Configuración.	Administración de la configuración, apoyo en la construcción.
Cristina P. Dávila Vega	CDV	Diseñador de Sistemas.	Actividades de diseño.
Victor González Castro	VGC	Auditor Interno.	Auditorías y constructor.
Dalila Herrera Carrera	DHC	Administrador de Pruebas.	Coordinar y realizar todas las actividades de la Prueba.
Mario E. Isauro Martínez	MIM	Líder del Proyecto.	Coordinador y constructor.
Elio Vega Munguía	EVM	Constructor.	Constructor y desarrollador de interfaces.
Gabriel Ramírez López	GRL	Administrador del Proyecto.	Administrar el proyecto y apoyo en construcción.
Jacqueline Téllez Cruz	JTC	Analista de Sistemas.	Análisis y apoyo en construcción.

Figura 5.1. Integrantes del equipo de SINSAS.

5.2. Planeación de la prueba.

De acuerdo a los elementos del capítulo Dos de esta tesis, se realizaron las actividades relacionadas a la planeación de la prueba para el proyecto SINSAS. A continuación se presenta una versión condensada de los requisitos de SINSAS y de los documentos que fueron generados durante la planeación de la prueba.

5.2.1. Requisitos de SINSAS.

La Coordinación de la Maestría en Ciencia e Ingeniería de la Computación de la UNAM, con sede en el IIMAS, requiere un sistema de inscripción de aspirantes al examen de admisión a Maestría y Doctorado a través de Internet. Este sistema contará con dos tipos de acceso; en el primero tendrán acceso los aspirantes para ingresar sus datos personales y en el segundo tendrá acceso la Coordinadora del Posgrado, para consultar y actualizar información relacionada con el examen de admisión.

En esta página electrónica los aspirantes deberán ingresar sus datos personales que se les soliciten, universidad de procedencia, si él es titulado o pasante, si desea ingresar a la maestría o al doctorado. Para tener derecho a examen, es necesario que el aspirante realice un pago en el banco anticipadamente, ya que deberá ingresar el número de ficha de pago y la fecha de pago; la fecha de inscripción al examen de admisión la asignará el sistema en forma automática

La Coordinadora será la encargada de marcar en SINSAS, a los aspirantes que hayan cumplido con todos los requisitos para presentar el examen de admisión. Una vez que se tienen identificados a estos aspirantes, el sistema asignará de manera automática a cada uno de ellos un número de examen.

El coordinador podrá realizar consultas en pantalla para los aspirantes que se hayan inscrito sistema. Los criterios son, por fecha de pago que tiene la ficha del banco, por nombre, por estado del aspirante – admitido, admitido con condición y no admitido –, por apellido, por número de examen, por los aspirantes que ya hayan realizado el pago correspondiente al examen de admisión. SINSAS también permite la impresión de los resultados de las consultas.

SINSAS presentará una funcionalidad para permitir que la Coordinadora edite mensajes para ser enviados por el correo electrónico. Los criterios del envío de mensajes son: enviar a todos los aspirantes, a un aspirante en particular, a todos aquellos que estén admitidos, a los aspirantes que no fueron admitidos, a aquellos que fueron admitidos con algún tipo de condición y a los aspirantes que no fueron admitidos.

Una vez que la Coordinadora de la Maestría tenga los resultados de los exámenes de admisión de los aspirantes ingresará en el sistema el estado de cada uno de ellos – admitido, no admitido, admitido con condición.

SINSAS permitirá que los aspirantes a la Maestría o Doctorado del IIMAS, verifiquen que tipo de estado se les ha asignado tomando en cuenta los trámites y el resultado del examen de admisión. El sistema mostrará un listado con el número de examen solamente y no presentará ningún otro tipo de información del aspirante.

Finalmente, SINSAS generará un archivo con todos los aspirantes al Posgrado que ya han sido admitidos. Este archivo se almacenará en el disco duro y tendrá posibilidad de ser impreso.

5.2.2. Desarrollo de una estrategia de prueba.

La estrategia de prueba para SINSAS, se obtuvo después de estudiar los requisitos que fueron obtenidos en la primera entrevista del cliente con algunos integrantes del equipo de desarrollo. En la junta estuvieron presentes el administrador del proyecto, el líder del proyecto y un analista de sistema. En esta reunión se obtuvo la primera versión de los requisitos, se discutieron las características que debía tener el software y se llegaron a algunos acuerdos importantes, para la definición del contrato. A los requisitos se les aplicó una primera revisión para asegurar que se contaba con toda la información y que ésta era clara y consistente. A partir de esta información se obtuvo una estrategia de prueba y su correspondiente esfuerzo de la prueba.

La primera versión de los requisitos fue aprobada, para permitir que se continuara con el desarrollo de SINSAS. Se aplicó una revisión apoyada con una lista de verificación y se entregó un informe por escrito al encargado del análisis con las observaciones obtenidas (Interpretación del resultado de la prueba). Vea la lista de verificación y la interpretación del resultado de la prueba, en el *Folio LV-ANA-VOR1* y *Folio IR-ANA-VOR1* respectivamente, en el anexo C.

5.2.2.1. Matriz de prueba.

Antes de la realización de la matriz de prueba, el administrador de la prueba DHC, convocó a una reunión con el responsable del equipo de análisis JTC, para discutir los resultados de la primera prueba aplicada a los requisitos. Las dudas que fueron surgiendo en la revisión eran resueltas, y en caso contrario, eran registradas en una lista para ser posteriormente, respondidas por el cliente. Una vez discutidos los requisitos, se comenzó a trabajar en la obtención de la matriz de prueba. La figura 5.2. muestra la matriz de prueba obtenida con esta información.

Factores de prueba	Requisitos	Diseño	Implementación
1. <i>Datos correctos:</i> entrada, salida y procesados.	✓	✓	✓
2. <i>Integridad de Archivos:</i> Uso de archivos correctos, así como el almacenamiento y recuperación de los mismos.		✓	✓
3. Datos procesados de acuerdo a las reglas del negocio.	✓	✓	✓
4. <i>Control de Acceso:</i> La información generada debe estar protegida.	✓		✓
5. <i>Conformidad:</i> El desarrollo del sistema respeta las políticas, procedimientos y estándares definidos.	✓	✓	✓
6. El sistema debe soportar la ejecución en Internet.			✓
Riesgos en la organización			
1. Modificación de datos ya almacenados			✓
2. Envío de correos electrónicos.	✓	✓	✓

Figura 5.2. Matriz de prueba de SINSAS.

5.2.2.2. Detalle de la estrategia de prueba.

Una vez analizada y verificada la información de la matriz de prueba de SINSAS, se tuvo la información necesaria para determinar los pasos de la estrategia de la prueba y así definir el esfuerzo de la prueba.

SINSAS fue desarrollado tomando en cuenta al paradigma orientado a objetos. Para poder realizar un análisis previo a la aplicación de la estrategia de prueba que se plantea en esta tesis, se aplicó la estrategia de la *Prueba de Validación de un Contexto Orientado a Objetos* [8]. Esto nos permitió dirigir las actividades de prueba y definir las bases para la obtención de los casos de prueba y del procedimiento de la prueba. Esta estrategia se centra en las acciones visibles y las salidas reconocidas por el usuario. Para ayudar en la determinación de las pruebas de validación, el ejecutor de la prueba se basó en los casos de uso que forman parte del modelo de análisis. Ya que un caso de uso, ofrece un escenario que posee una alta probabilidad de encubrir errores en los requisitos de interacción con el cliente.

Para llevar a cabo las actividades de prueba para SINSAS, solo se contó con un integrante del equipo de desarrollo, DHC que fue el Administrador de la Pruebas. Se contempló la posibilidad de pedir apoyo de algún integrante del equipo de desarrollo, para la aplicación de las pruebas en la construcción y en la integración del sistema. Sin embargo, no fue necesario ya que el tiempo planeado fue suficiente para la aplicación de las pruebas. El administrador de pruebas evaluó los resultados que fueron obtenidos durante el proceso de la prueba.

Para las prueba de requisitos, diseño y la revisión de los manuales, se realizó en escritorio. En el caso de la prueba de la implementación, integración del sistema y pruebas del sistema, se realizaron en los espacios donde se tuvo instalado el

servidor y el cliente así como el hardware y software requerido. En todos los casos anteriores estuvo presente al menos un integrante del equipo en el momento de realizar la prueba, cuando se entregaron los informes de la prueba y en la discusión de los resultados. Toda la documentación fue firmada por al menos un responsable al momento de recibir los resultados de la prueba.

5.2.2.3. Esfuerzo de prueba.

El esfuerzo de la prueba se definió tomando en cuenta a un solo integrante del equipo de prueba y al periodo de tiempo en que se planeó que debería estar terminado el software. El periodo fue acordado con el usuario, y constó de tres meses, como se encuentra registrado en el contrato de SINSAS.

El esfuerzo de la prueba utilizado se presenta en el documento con *Folio DC-PP- NUM1* del anexo C. En éste informe se describe en forma general, las actividades que fueron probadas y los días que se planearon utilizar. En el mismo anexo, el documento con *Folio DC-PP- NUM2*, presenta el número de revisiones necesarias para cada actividad y las horas que se planearon en invertir en cada actividad de la prueba.

En la realidad esto puede variar, sin embargo es una aproximación para cuestiones de planeación. El detalle de las fechas en cada etapa de prueba en particular se especificó en el plan del proyecto general, presentado por el administrador del proyecto.

5.2.3. Desarrollo de un plan de prueba.

A continuación se presenta el plan de pruebas que se desarrolló tomando en cuenta la estrategia de prueba y el esfuerzo de la prueba, así como los requisitos de SINSAS.

1. Definición de una Estrategia de prueba detallada: Se explica a que partes y de que forma se aplicará la prueba a SINSAS, así como las técnicas que serán empleadas y la forma de darles seguimiento.

1.1 Definición de la prueba en el ciclo de vida: Las pruebas se llevarán a cabo desde la definición de los requisitos hasta la liberación del software. Las actividades de la prueba se encuentran detalladas en el anexo C, *Folio DC-PP- NUM1*. Quedan excluidas la etapa de mantenimiento. Se presenta un documento con el tiempo necesario para la aplicación de las pruebas a SINSAS, expresadas en horas. Vea el anexo C, *Folio DC-PP- NUM2*.

1.2 Planteamiento general de la prueba: La primera fase de la prueba para SINSAS, se realizará incluyendo la etapa preventiva de las pruebas – análisis de requisitos y el diseño. La prueba en la fase preventiva, nos

permitirá asegurar que se cuentan con los requisitos que el usuario definió, y que los requisitos se encuentran claros, completos y consistentes. La prueba al diseño nos permitirá verificar que todos los requisitos y funcionalidades se encuentren en el diseño, correctamente modelados. La segunda fase de la prueba, incluirá las pruebas a la codificación, integración y del sistema, así como la prueba de las interfaces y manuales.

1.3 Técnicas de prueba a ser empleadas: Se aplicarán las listas de verificación como una técnica de prueba en cada una de las etapas del ciclo de vida del software. Se realizarán revisiones centradas en descubrir problemas específicos, como una técnica de prueba auxiliar.

1.4 Definición de componentes y elementos de prueba o constructores: El equipo de prueba ha definido formatos para llevar a cabo las pruebas y comunicar los resultados o hacer recomendaciones. Para el caso específico de la codificación se ha definido un formato para los archivos y la forma de nombrar las variables, el cual deberá ser respetado en todo código realizado. Los formatos se definieron por el equipo de pruebas y fueron discutidos con el líder de proyecto (MIM) y el auditor interno (VGC).

1.5 Procedimiento para revisar y expandir las revisiones: Las pruebas serán aplicadas en un plazo no mayor de 24 horas, de cuando sea entregado el documento para su evaluación. Los resultados serán entregados en un plazo no mayor de 48 horas, para posteriormente ser discutidos. En caso de no ser así, se generará un informe en donde se explique la razón del atraso. Las pruebas en la etapa preventiva del software, se realizarán en escritorio y no será necesario que se encuentre presente algún integrante del equipo que respalde el trabajo a probar. El informe se entregará a un responsable y se discutirán los resultados, de tal forma que en caso que surgiera alguna duda, sea aclarada en el momento. Una vez que se han discutido los resultados obtenidos de la prueba, se deberá firmar de recibido por el responsable del equipo o por todos los integrantes, según sea el caso. En caso de no estar de acuerdo con los resultados de la prueba – incluso cuando se ha firmado de conformidad –, el equipo inconforme tendrá 24 horas para discutir la situación con el equipo de pruebas y poder evaluar los resultados nuevamente.

2. Diseño de prueba y técnica de implementación: Se explica la forma general de llevar a cabo el proceso de prueba, así como la manipulación y obtención de los datos involucrados en este proceso.

2.1 Organización de casos de prueba: Los casos de prueba deben asegurar que las etapas de la prueba estén identificadas con las clases – obtenidas en el diseño –, que se van a probar. Para ello, es necesario establecer el propósito de la prueba y desarrollar una lista de pasos en donde se detalle

la forma de llevarla a cabo. Esta lista debe contener los estados especificados del objeto a probar, mensajes y operaciones que se vayan a realizar. La lista de mensajes y operaciones, deberá ser de utilidad cuando se esté ejercitando la prueba. También será necesario identificar algunas excepciones que puedan ocurrir y detallarlas, de modo que ayuden en la comprensión o implementación de la prueba.

2.2 Matriz de validación de requisitos: La matriz de validación de requisitos forma parte de la estrategia de prueba. Se presentó una técnica para la obtención de esta matriz en la sección 2.1. del Capítulo Dos de esta tesis. La figura 5.2, muestra la matriz de prueba de SINSAS, que fue obtenida tomando en cuenta esta técnica.

2.3 Información de datos de prueba: Los datos de prueba relacionado con los aspirantes, serán datos reales que se han obtenido de los estudiantes que han hecho la solicitud de ingreso al examen de la Maestría o Doctorado en Ciencias de la Computación de la UNAM. Dado que los resultados obtenidos en los exámenes de ingreso no son números complicados, éstos serán inventados, así como el contenido de los mensajes que serán enviados para la prueba de los correos electrónicos. Los datos de entrada y salida de SINSAS, son:

- ✓ Datos personales del aspirante para su registro al examen de admisión: Serán datos reales de los aspirantes que han hecho la solicitud al examen con anterioridad.
- ✓ Calificación del resultado del examen de los aspirantes: Es un dato numérico que se encuentra en el rango de 0.00 a 10.00.
- ✓ Fichas de pago para verificar que el estudiante pagó previamente en el banco: La información necesaria es el nombre del aspirante, la cantidad del pago y la fecha en que realizó el pago.
- ✓ Texto de los correos electrónicos para enviar información al aspirante: Puede ser cualquier tipo de mensaje alfanumérico.
- ✓ Estado del resultado del examen del aspirante, los cuales son admitido, no admitido y admitido con condición.
- ✓ Criterio de búsqueda de los aspirantes: Una vez que se encuentran almacenados los datos de los aspirantes, será necesario realizar búsqueda por los criterios de nombre, apellido, fecha de pago, calificación y por el estado del resultado del examen del aspirante.

3. Responsabilidades: Se definen los integrantes del equipo de prueba para SINSAS, los roles que necesitarán y sus obligaciones correspondientes.

3.1 Organización de pruebas: Las pruebas se llevarán a cabo simultáneamente con el desarrollo del software. Las actividades de la prueba se presentarán

en el análisis, diseño, implementación y liberación del software, así como revisión de los manuales de usuario y técnico.

3.2 Recursos y planeaciones: Para llevar a cabo las actividades de prueba para SINSAS, será necesario contar con los siguientes roles: Administrador de prueba, diseñador de prueba, ingeniero de prueba, probador del sistema y probador de la integración. Dado que solo se tiene un solo integrante del equipo de prueba, éste realizará todos los roles necesarios, en caso contrario se expresará explícitamente en un escrito.

3.3 Tareas y asignaciones: A continuación se definen las actividades que deberá llevar a cabo cada uno de los roles.

- ✓ **Administrador de prueba:** Es el responsable de los resultados de la aplicación de la prueba en SINSAS. El administrador definirá la calendarización, el plan y las políticas de prueba. Éste también dará seguimiento al plan de prueba, hasta que incluya todas las actividades de prueba que fueron planeadas.
- ✓ **Ingeniero de prueba:** Creará los casos de prueba y los procedimientos de la prueba, también se encargará de aplicar las pruebas, registrar los resultados, generar el informe, entregarlo y discutirlo con el equipo involucrado. Este rol se encargará de probar los manuales técnico y de usuario.
- ✓ **Probador de la integración:** Deberá llevar a cabo las pruebas del código, las pruebas a nivel módulo y finalmente las pruebas de integración de los módulos. También realizará las pruebas de la interfaz.
- ✓ **Probador del sistema:** Deberá llevar a cabo las pruebas cuando todos los módulos se encuentren integrados. Se encargará de preparar los datos de prueba que serán usados para probar el software con datos reales, antes de ser liberado.
- ✓ **Revisión por pares:** Debido a que las actividades de prueba serán realizadas – en principio, solamente por DHC, serán revisados todos los documentos de prueba antes de ser aprobados por VGC, para realizar observaciones y correcciones en caso necesario.

4. Resultados de las pruebas: Se explican los formatos que serán ocupados por el equipo de prueba, la forma de completarlos y el significado de su uso dentro de las actividades de prueba.

4.1 Documentos de prueba: Se definieron cuatro formatos para la aplicación de pruebas y la comunicación de los resultados. Vea los siguientes formatos en el anexo B.

- ✓ El formato *FP-LV-01* o **Lista de verificación**; *FP* significa formato de prueba, *LV* lista de verificación y a continuación el número de

documento. Contiene las listas de verificación con las que se van a probar cada una de las etapas del software que se han planeado. Los primeros datos que se requieren, son la etapa a la cual se le aplicará la lista de verificación. Posteriormente el nombre del equipo que desarrolló esa parte del software y el nombre del responsable del equipo. En el recuadro se define la lista de verificación, la cual debe ser revisada por el administrador de la prueba y discutida con el equipo de prueba. En el pie de página del formato *FP-LV-01*, se escribirá el nombre de quien desarrolló la lista de verificación y las siglas de quién aplicó la prueba. También es necesario escribir las siglas y firma de quien estuvo presente en la discusión de los resultados de la prueba, así como las siglas y la firma de quien recibió el informe, ya que será entregada una copia de éste. Finalmente se escribe la fecha de cuando este informe se entregó.

- ✓ El formato *FP-IR-02* o **Interpretación del resultado de la prueba**; *FP* significa formato de prueba, *IR* interpretación del resultado y a continuación el número de documento. Contiene el resultado que se obtuvo después que se aplicó la prueba. Es necesario registrar la etapa a la cual se le aplicó la prueba, el equipo que desarrolló la parte del software y el nombre del responsable del equipo. En el recuadro, en la columna *Criterio de prueba* se describe brevemente el lugar donde se encontraron los problemas; y en la columna *Resultados/Recomendaciones*, se describe que tipo de problema y algunas ideas o recomendaciones para resolverlos. En el pie de página del formato *FP-IR-02*, se escribirán las siglas y la firma de quien realizó y entregó el informe. También es necesario escribir las siglas y firma de quien estuvo presente en la discusión de los resultados de la prueba, así como las siglas y la firma de quien recibió el informe – ya que será entregada una copia del mismo, y la fecha de cuando éste se entregó.
- ✓ El formato *FP-IC-03* o **Informe de cambio**; *FP* significa formato de prueba, *IC* informe de cambio y a continuación el número de documento. Se utiliza para llevar el control de los cambios que vaya presentando el software. Cuando surge un cambio, es necesario identificar el impacto que presenta el proceso de prueba y los componentes que serán afectados, así como la forma de adecuar la prueba nuevamente. Es necesario registrar el nombre de la etapa donde se presentó el cambio, el nombre y el responsable del equipo. El primer recuadro debe ser llenado por el responsable del equipo que presentó el cambio y contiene toda la información que se refiere al componente del software que sufrió el cambio. También el nombre y la firma de quien entrega esta información, así como la fecha en que fue entregado este documento al equipo de pruebas. El segundo recuadro es llenado por un integrante del equipo de pruebas, y aquí se define el impacto que causa el cambio informado. Debe ser detallado claramente el impacto que causa en la prueba ya diseñada y las nuevas consideraciones que deben ser

tomadas en cuenta, incluso pueden ser anexados documentos. No es necesario dar una copia al equipo que presentó el cambio, ya que ellos tienen otro formato definido por el administrador del proyecto. Finalmente, se registra el nombre y firma de quien elaboró el formato, así como la fecha en que fue terminado el documento.

- ✓ El formato *FP-DC-04* o **Documentación complementaria**; *FP* significa formato de prueba, *DC* documentación complementaria y a continuación el número de documento. Se utiliza para cualquier información que necesite ser comunicada por el equipo de prueba y que no esté contemplada en los tres formatos definidos anteriormente. Se debe registrar el nombre de la etapa a la cual le va a ser entregado el informe, el nombre del equipo y el nombre del responsable del ese equipo. En el inciso **a**, se explica brevemente el objetivo del documento, es decir, la razón que éste se originó. En el inciso **b**, se explica brevemente de que forma se presentará la información y ésta se redacta, de forma clara y sin perder de vista el objetivo. En el pie de página este formato, se escribirán las siglas y la firma de quien realizó y entregó el documento. También se registrarán las siglas y la firma de quien lo recibió – ya que será entregada una copia del documento –, y la fecha de cuando éste se entregó.

4.2 *Reportes de prueba*: Los reportes de prueba deben ser realizados por el administrador de pruebas; ya que solo existe un responsable de las actividades de prueba, aún en el caso que alguien mas haya aplicado la prueba. Todos los reportes de prueba, deben ser llenado debidamente en los formatos ya definidos. Todo documento de prueba debe ser firmado por el responsable de la prueba, por el responsable del equipo que lo recibió y en caso de ser necesario, por todos los integrantes del equipo al cual se le ha aplicado la prueba.

4.3 *Folio de los reportes*: El folio de los documentos se constituye de cuatro datos por ejemplo *Folio: LV-ANA-VOR1*:

- ✓ *Folio*: La palabra folio.
- ✓ *LV*: El tipo de documento que puede ser *LV* lista de verificación, *IR* interpretación del resultado, *IC* informe de cambio o *DC* documentación complementaria.
- ✓ *ANA*: La etapa a la que pertenece el documento: *PP* planeación de la prueba, *ANA* análisis o *DIS* diseño.
- ✓ *VOR1*: *V0* es la versión del documento, la numeración empieza en cero y aumenta de uno en uno, cuando ya ha sido aceptada la versión. *R1* es el número de revisiones que lleva el documento, éste puede tener cualquier número de versiones antes de pasar a la siguiente versión que ya haya sido aceptada.

5. Procedimientos: Se presentan los procedimientos para solucionar situaciones que pudieran surgir durante el proceso de prueba, conflictos, repeticiones de pruebas, etc.

5.1 Procedimientos de prueba: Deberán ser entregados las versiones más recientes de los documentos, para la aplicación de la prueba. En el caso de la prueba a la etapa de análisis y diseño, el probador registrará anotaciones en los informes que le fueron entregados, así como en las listas de verificación. De tal forma que cuando se entreguen los resultados de la prueba, serán devueltos los documentos probados, con las anotaciones hechas por el equipo de prueba. Se realizarán informes, para respaldar los resultados obtenidos en las listas de verificaciones, en donde se detallará el resultado de la prueba. En caso necesario se harán recomendaciones específicas para poder dar solución al problema que la prueba ha identificado. Para las pruebas en la implementación de SINSAS, nos apoyaremos en los documentos de los Componentes de Prueba, en los Casos de Prueba – generados por el equipo de prueba –, y los diagramas de interacción obtenidos en la etapa de diseño. El equipo de prueba creará los documentos Proceso de Prueba y Casos de Prueba, para asegurar que se ha verificado el funcionamiento correcto de SINSAS.

5.2 Reporte de los problemas y sus ajustes: En caso de que surja algún desacuerdo con el procedimiento de la prueba, los resultados de la misma o cualquier otra situación, podrán dárseles solución en tres formas distintas. Primeramente, serán discutidas con el equipo de pruebas o con el administrador de prueba. En caso de que no se haya solucionado el conflicto, éste será discutido con el administrador del proyecto o el auditor interno; el administrador de la prueba deberá estar presente. Si aún no se ha encontrado solución alguna al problema, entonces éste será escalado a las coordinadoras del proyecto, Dra. Hanna Oktaba o la M. en C. Guadalupe Ibargüengoitia González; en este caso deberán estar presentes todos los que han sido involucrados en la búsqueda de la solución del problema.

5.3 Procedimientos para controlar las pruebas que se ejecutan éstas mismas, en varias ocasiones*: En el documento con *Folio DC-PP-NUM2* del anexo C, se ha definido la cantidad de pruebas que se han de aplicar en cada etapa del software. En caso de necesitar agregar alguna prueba extra o en caso de no llevar a cabo la prueba ya definida, será necesario anexar un documento para que quede comunicado y deberá ser firmado por el equipo involucrado, además de ser informado al auditor interno VGC y al administrador del proyecto GRL.

* es la explicación del término *retesting*, utilizado en lengua inglesa

6. Controles: Se explica la forma en que se va a manipular la información generada en el proceso de prueba.

6.1 Cambios: Para controlar los cambios que vayan surgiendo en el plan de pruebas y los documentos generados por la prueba, éstos serán registrados en el *Formato FP-IC-03* del anexo B, cuya forma de llenarlo ya se ha explicado en el punto 4.1 de este mismo documento. Este documento, está relacionado con el informe de solicitud de cambio, definido por el administrador del proyecto. En caso que este documento no exista, no será aceptado el cambio.

6.2 Reportes e información: Toda la información que sea generada por el equipo de prueba, deberá ser registrada en los formatos que el equipo de prueba definió – vea el punto 4: **Resultados de las pruebas**. El administrador de la prueba deberá validar de que los informes, tengan la información necesaria. Toda información generada por el equipo de prueba, deberá ser archivada en papel y electrónicamente. Los documentos probados, deberán ser almacenados solamente en papel, para respaldar los resultados obtenidos por las pruebas. Ningún informe de prueba será válido sin las firmas requeridas. Los nombres de los archivos serán etiquetados igual que el folio del informe, para evitar incompatibilidad de éste y el nombre del archivo electrónico.

5.3. Pruebas a los Requisitos.

A continuación se presentan los documentos que fueron obtenidos al aplicar la prueba los requisitos de SINSAS. Las actividades de la prueba se llevaron a cabo tomando en cuenta el plan de prueba que fue presentado en la sección 5.2.3. de este capítulo

5.3.1. Objetivo.

Ayudar a integrar de manera consistente, todos los requisitos de SINSAS en un corto periodo de tiempo y buscar que el equipo de análisis adquiera la responsabilidad que sus productos forman parte fundamental de las actividades del diseño.

5.3.2. Actividades.

Se llevaron a cabo reuniones con JTC, la encargada del equipo de análisis. En la primera reunión se discutieron las actividades del plan de prueba que se aplicaron a esta etapa de desarrollo. Las siguientes reuniones fueron para discutir la interpretación de los resultados obtenidos de la lista de verificación, la cual evaluó a los requisitos y a las especificaciones de SINSAS.

5.3.3. Documentos generados.

En la prueba a los requisitos se generaron los siguientes documentos, los cuales se encuentran en el anexo C.

- ✓ *Folio LV-ANA-V0R1*: Es la primera prueba que se le aplicó a los requisitos, con el fin de obtener una primera versión de los requisitos para iniciar el plan de prueba.
- ✓ *Folio IR-ANA-V0R1*: Es la interpretación de la prueba correspondiente a la lista de verificación del folio anterior.
- ✓ *Folio DC-PP-NUM1* y *Folio DC-PP-NUM2*: Estos documentos forman parte de los fundamentos del proceso de prueba. Fueron evaluados por VGC y su información se encuentra integrada al plan de prueba.
- ✓ *Folio DC-ANA-NUM1*: Es un documento informativo, en donde se explican la forma de llevar a cabo las pruebas en la fase de análisis.
- ✓ *Folio LV-ANA-V0R2*: Es la segunda lista de verificación aplicada a los requisitos y la especificación, en una versión preliminar.
- ✓ *Folio: IR-ANA-V0R2*: Es la interpretación del resultado correspondiente a la segunda lista de verificación.
- ✓ *Folio: LV-ANA-V1R1*: Es la tercer lista de verificación que fue aplicada a la primera versión aceptada de los requisitos y a la especificación.
- ✓ *Folio: IR-ANA-V1R1*: Interpretación del resultado correspondiente a la aplicación de la tercer lista de verificación.

5.3.4. Resultados.

Se aplicó la prueba a los requisitos en tres ocasiones, mejorando notoriamente en cada nueva versión. Al principio se presentó cierta resistencia en la realización de los cambios que se proponían en los resultados de la prueba. Sin embargo, al final fueron llevados a cabo casi en su totalidad. Ya que en la aplicación de la última prueba a los documentos obtenidos en el análisis, se encontraron ciertos puntos que corregir, pero debido a la premura de tiempo, se acordó entre la encargada del equipo de análisis JTC, el líder del proyecto MIM, el administrador del proyecto GRL y el administrador de prueba DHC, que JTC realizará los cambios y no habría necesidad de aplicar la prueba en una cuarta ocasión.

5.4. Pruebas del Diseño.

En esta sección se presentan los documentos generados al aplicarse las pruebas en la etapa de diseño. La prueba significó una gran ayuda en la integración de los documentos del diseño, identificando desde la primer prueba los elementos incompletos o faltantes con respecto al mismo diseño y al análisis.

5.4.1. Objetivo.

El objetivo principal fue evidenciar al equipo de diseño la importancia de crear el diseño de SINSAS, de una forma consistente y de fácil implementación, así como satisfacer completamente los elementos definidos en la etapa del análisis.

5.4.2. Actividades.

Se realizaron reuniones con el equipo de diseño – CAS y CDV –, para discutir las actividades del plan de pruebas contempladas para esta etapa de desarrollo. Las siguientes reuniones fueron, de igual forma que en el análisis, para discutir la interpretación de los resultados obtenidos de la lista de verificación, durante la prueba del diseño de SINSAS.

5.4.3. Documentos generados.

En la prueba del diseño se generaron los siguientes documentos, los cuales se encuentran en el anexo C.

- ✓ *Folio DC-DIS-NUM1*: Es un documento informativo en donde se explican la forma de llevar a cabo las pruebas en la fase de diseño.
- ✓ *Folio LV-DIS-V0R1*: Es la primer lista de verificación aplicada al diseño en una versión preliminar.
- ✓ *Folio IR-DIS-V0R1*: Es la interpretación del resultado correspondiente a la primera lista de verificación.
- ✓ *Folio LV-DIS-V1R1*: Es la segunda lista de verificación que fue aplicada a la primera versión ya aceptada del diseño.
- ✓ *Folio IR-DIS-V1R1*: Interpretación del resultado correspondiente a la aplicación de la segunda lista de verificación.

5.4.4. Resultados.

Se aplicó la prueba al diseño en dos ocasiones, ya que desde la primera revisión del documento, se presentaba un diseño sin tantos problemas. Una vez obtenida la primera versión del diseño, se aplicó una prueba más, para verificar que se realizaron las modificaciones recomendadas. Los detalles que quedaron pendientes, se acordaron – con el equipo de diseño CAS y CDV –, que se realizarían antes de generar la primera versión formal del diseño. Por esta razón no fue necesario aplicar una prueba más a los documentos del diseño.

Conclusiones

La definición de una metodología de prueba y su aplicación en el ciclo de vida del software, permite identificar los riesgos que se pueden presentar durante el desarrollo y puesta en marcha del software.

La mejor metodología de prueba se obtiene mediante la integración de las diferentes versiones corregidas de ésta, para lograr constituir una metodología de prueba madura.

En la búsqueda de una metodología de prueba madura, es fundamental conocer y planear todas las actividades que la constituyen, y es precisamente el objetivo de la visión preventiva de las pruebas.

Las principales actividades preventivas de la prueba, son la planeación de la misma, tomando en cuenta las características y recursos del software, así como la realización de pruebas al análisis y diseño del mismo. Por esta razón, en esta tesis se explican a detalle las técnicas para desarrollar y aplicar las actividades preventivas de la prueba de una manera sencilla, que es aplicable a proyectos reales de desarrollo de software.

Dentro de las actividades preventivas de la prueba, se incluyen la prueba al análisis y diseño del software. Debido a que la complejidad de estas etapas varía considerablemente de un software a otro, aquí se presentan cinco distintas técnicas de prueba para el análisis y cuatro técnicas para el caso del diseño.

Las técnicas de prueba para el análisis y diseño, se planearon para facilitar su aplicación en la práctica. Cada técnica muestra una descripción clara y concisa, los roles del equipo de prueba que realizaron las actividades de la técnica y los elementos de entrada que éste necesita. Posterior a esto, se presentan listados los pasos que constituyen a la técnica y las salidas que deber ser obtenidas cuando ésta es finalizada.

Dada la importancia de las actividades de prueba aquí desarrolladas, éstas son modeladas en diagramas de actividad. Estos diagramas contienen toda la información que se encuentra expresada en forma de texto, y permite identificar cada componente de la actividad y el grado de avance de la técnica, de manera inmediata.

Para comprobar que las actividades preventivas de la prueba forman parte de los elementos que respaldan el éxito en el desarrollo de un software, se presentan los resultados que se obtuvieron durante la aplicación de estas actividades en el desarrollo de una proyecto real, el sistema SINSAS.

El resultado fue excelente, ya que las pruebas fueron vistas como una gran ayuda para la identificación y corrección de todos los conflictos que se presentaron durante la etapa de análisis y diseño de SINSAS. Las pruebas ayudaron en la recolección de los requisitos, resolviendo inconsistencias, omisiones y confusiones. Una vez integrados los requisitos, las especificaciones fueron creadas con una disminución considerable de los recursos. En el caso del diseño, las pruebas solucionaron nuevamente, situaciones de inconsistencias, omisiones y confusiones, pero ahora con respecto al análisis. Las correcciones se realizan de una manera fácil, gracias a las actividades definidas en el plan de prueba y de los resultados que se obtuvieron en las actividades de prueba ya aplicadas.

Los resultados de las pruebas fueron discutidos con el equipo de prueba y con los integrantes del equipo de desarrollo pertinentes. Una investigación futura que pude realizarse tomando en cuenta esta tesis, es buscar técnicas de investigación y recolección de información para integrar dentro del plan de prueba, actividades que involucren verificaciones del cliente, con el fin de que él evalúe si el software que se está desarrollando esta cumpliendo sus expectativas.

Integrando las ideas anteriores, el proceso de prueba esta formado de tres fases, la fase preventiva, la fase demostrativa y la fase de detección. La visión preventiva de las pruebas en el software, contiene los elementos de la primera fase que constituye al proceso de prueba – fase preventiva. Las actividades que constituyen a la visión preventiva de la prueba nos permiten:

- ✓ Identificar los elementos de la prueba y sus riesgos inherentes que pudieran causar problemas en el desarrollo y puesta en marcha del software.
- ✓ Definir y planear las actividades de la prueba en las etapas del ciclo de vida del software, con el objetivo de disminuir los riesgos identificados.
- ✓ Acordar el contenido y el formato de los entregables, así como las actividades en donde éstos van a ser generados.
- ✓ Identificar e integrar un equipo de prueba con sus respectivos roles y obligaciones, dependiendo de la complejidad del software a desarrollar.
- ✓ Realizar mejoras en esta fase preventiva, a partir de los resultados obtenidos en la aplicación de estas actividades.

El logro obtenido al integrar las actividades de prueba anteriores en un conjunto de técnicas preventivas, constituye las bases para continuar esta investigación sobre las demás fases de la prueba. Si además se respetan los formatos que se proponen en esta tesis, se logrará integrar una guía completa para las pruebas del software que facilite las actividades de prueba durante el desarrollo del software en situaciones reales.

Referencias

- [1] Computers & Concepts Associates., Little Book of Testing. Volume I. Software Program Managers Network. USA 1998. Pág. 4-5.
http://www.iceinc.to/products_guidebooks.html
- [2] KIT, Edward. Software Testing in the Real World, Addison-Wesley, USA 1995. Pág. 19.
- [3] BOOCH, G., RUMBAUGH J., JACOBSON I. The Unified Modeling Language - User Guide. Addison-Wesley. USA 1999. Capítulo 19.
- [4] KIT, Edward. *Obra Citada*. Pág. 18.
- [5] HETZEL, William C., The Complete Guide to Software Testing. John Wiley & Sons, Inc. USA 1988. Pág. 6.
- [6] HETZEL, William C., *Obra Citada*. Pág. 44.
- [7] PERRY, William, Effective Methods for Software Testing. John Wiley & Sons, Inc. USA 1995. Págs. 25-27.
- [8] PRESSMAN, S. Roger, Ingeniería del Software.Un enfoque práctico. McGraw-Hill. España 1998. Pág. 428.

Bibliografía

1. BOOCH, G., RUMBAUGH J., JACOBSON I. *The Unified Modeling Language - User Guide*. Addison-Wesley. USA 1999.
2. Computers & Concepts Associates., *Little Book of Testing. Volume 1 y Little Book of Testing. Volume 2*. Software Program Managers Network. USA 1998.
http://www.iceinc.to/products_guidebooks.html
3. HETZEL, William C., *The Complete Guide to Software Testing*. John Wiley & Sons, Inc. USA 1988.
4. KIT, Edward. *Software Testing in the Real World*, Addison-Wesley, USA 1995.
5. KOOMEN, Tim. POL, Martin., *Test Process Improvement. A practical step-by-step guide to structured testing*, Addison-Wesley. England 1999.
<http://www.awl-he.com/computing/>
6. PERRY, William, *Effective Methods for Software Testing*. John Wiley & Sons, Inc. USA 1995.
7. PRESSMAN, S. Roger, *Ingeniería del Software.Un enfoque práctico*. McGraw-Hill. España 1998.
8. TAI, K. C., *What to Do Beyond Branch Testing*. ACM Software Engineering. Notes, vol. 14 No. 2, abril, 1989, pp. 58-61.
<http://www.acm.org>
9. BINDER, R. V., *Object-Oriented Software Tesing*. Communications of the ACM, vol. 37, No. 9, septiembre 1994, pp. 29.
<http://www.acm.org>
10. MCGREGOR, J. D. KORSON T. D., *Integrated Object-Oriented Testing and Development Process*. Communications of the ACM, vol. 37, No. 9, septiembre 1994, pp. 59-77.
<http://www.acm.org>
11. GOLDSMITH, Robin, *Plan your testing. Process*. Software Development Magazine, SD Magazine's, No. 28, abril 1999.
<http://www.sdmagazine.com>
12. Sitios WEB relacionados con pruebas de software.
<http://www.sqe.com/>, <http://www.ondaweb.com/sti/>, <http://www.stlabs.com>,
<http://www.csst-technologies.com/>.

Anexo A

Figura A.1. Desarrollo de una estrategia de prueba y el esfuerzo estimado de la prueba.

Diagrama de prueba: (Entrada *req*:Requisitos, *esp*:EspecificaciónTécnica, Metodología *mpp*:MetodologíaDelProcesoDePrueba, Salida *inf*:Informe).

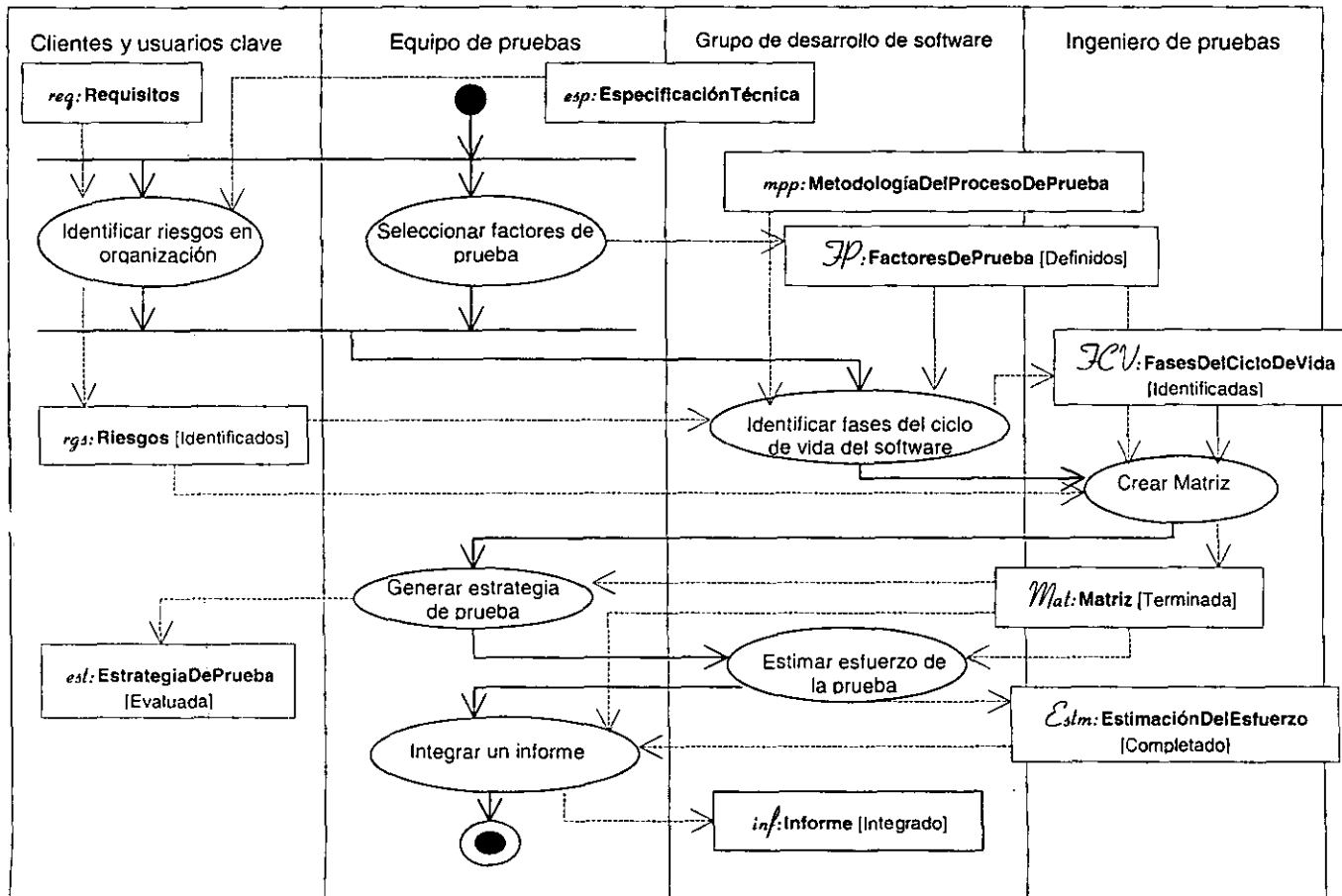


Figura A.2. Desarrollo de un plan de pruebas.

Diagrama de prueba: (Entrada *req*:Requisitos, *esp*:EspecificaciónTécnica,
 Metodología *mpp*:MetodologíaDelProcesoDePrueba, Salida *inf*:Informe, *pp*:PlanDePrueba).

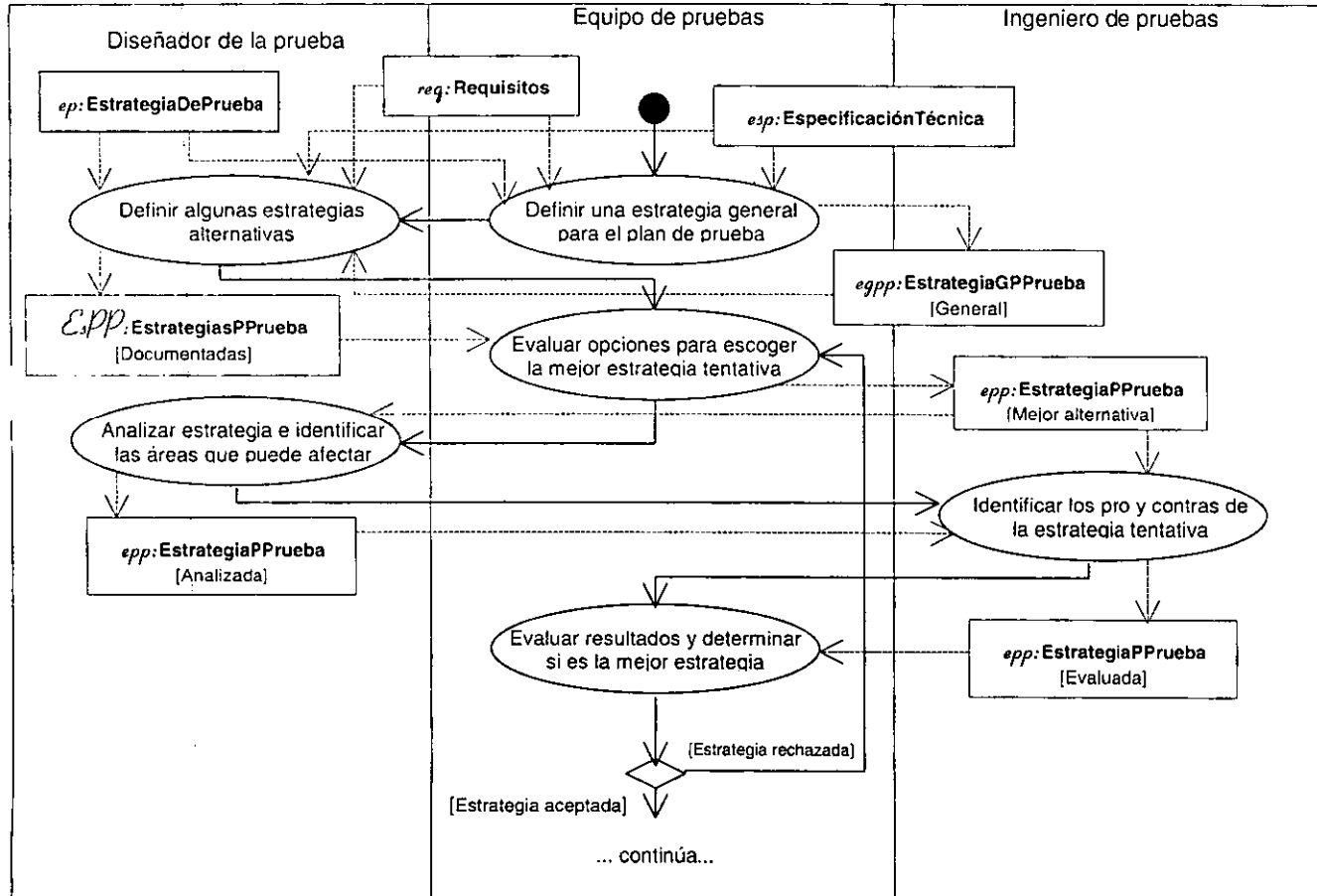


Figura A.2. Desarrollo de un plan de pruebas. (...Continuación...)

Diagrama de prueba: (Entrada *req*:Requisitos, *esp*:EspecificaciónTécnica,

Metodología *mpp*:MetodologíaDelProcesoDePrueba, Salida *inf*:Informe, *pp*:PlanDePrueba).

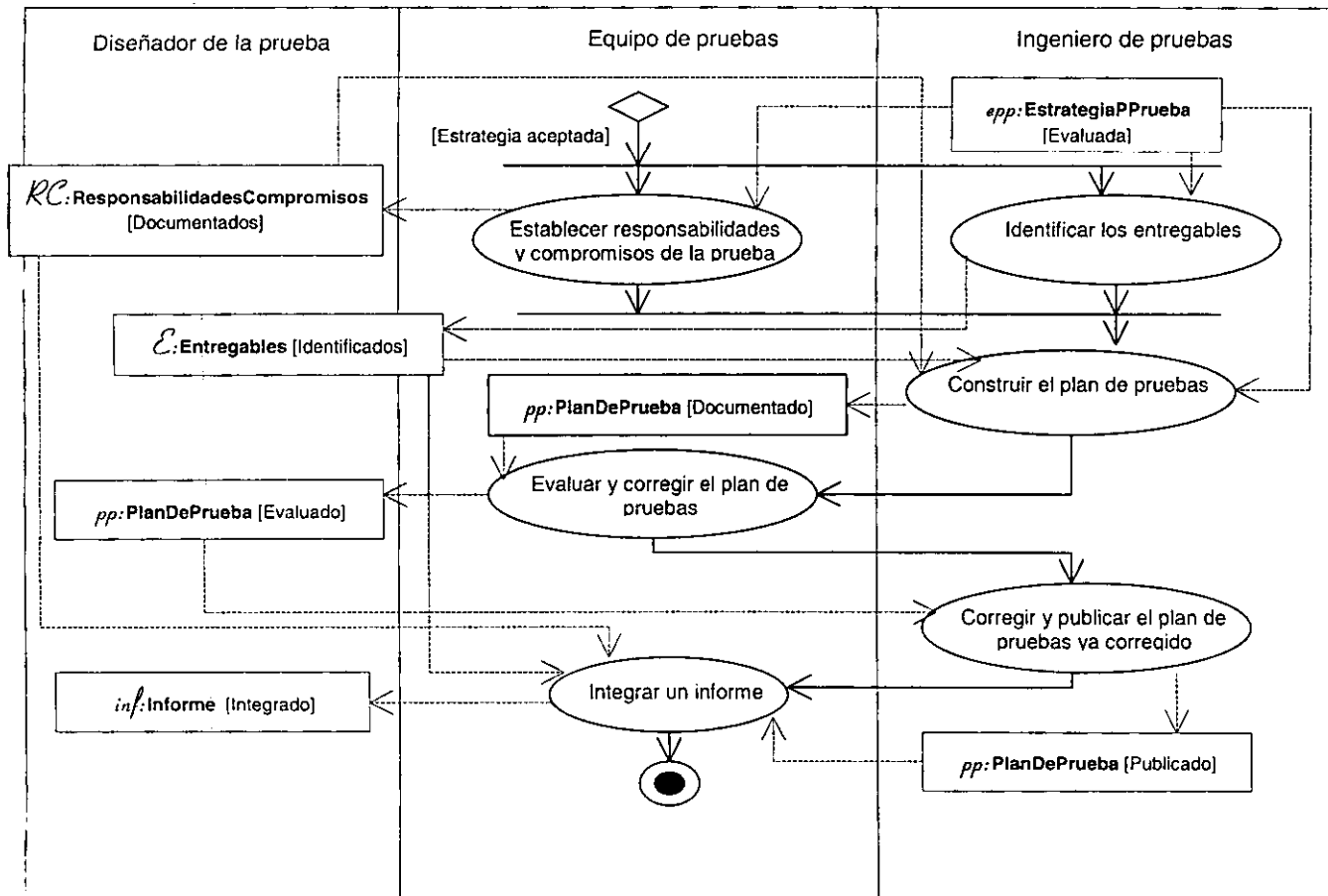


Figura A.3. Prueba de Requisitos con técnica: "Entendiendo que se va a probar".

Diagrama de prueba: (Entrada *req: Requisitos*, Técnica *ntec: "Entendiendo que se va a probar"*, Salida *inf: Informe*).

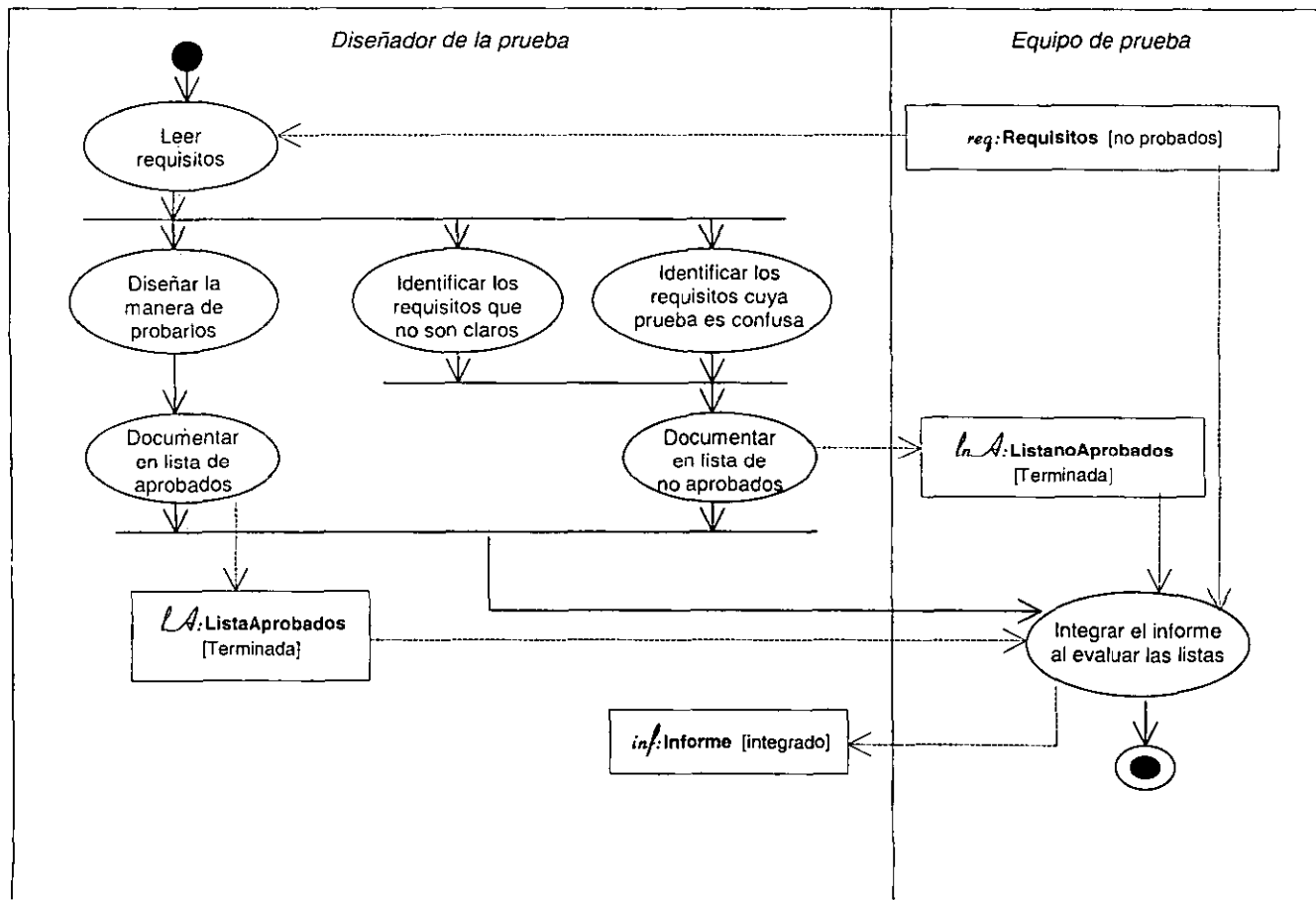


Figura A.4; Prueba de Requisitos con técnica: Apoyada en el diseño de casos de prueba basados en requisitos.
 Diagrama de prueba: (Entrada *reg*: Requisitos, Técnica *nlec*: Apoyada en “el diseño de casos de prueba basados en requisitos”, Salida *inf*: Informe).

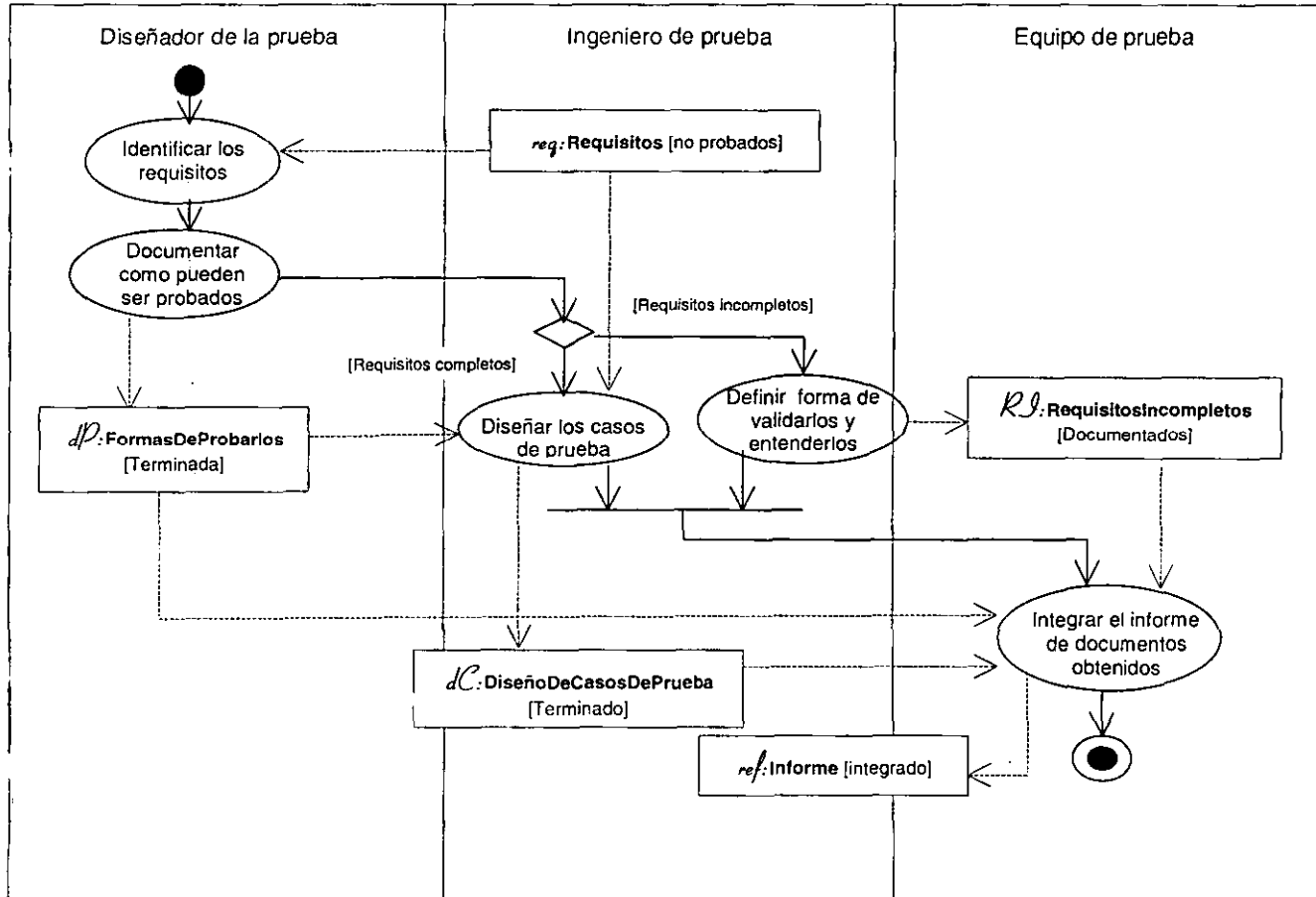


Figura A.5. Prueba de Requisitos con técnica: Apoyada en una matriz para la validación de requisitos.
 Diagrama de prueba: (Entrada *req:Requisitos*, Técnica *ntec: Apoyada en una matriz para la validación de requisitos*,
 Salida *inf:Informe*, *mat:Matriz*).

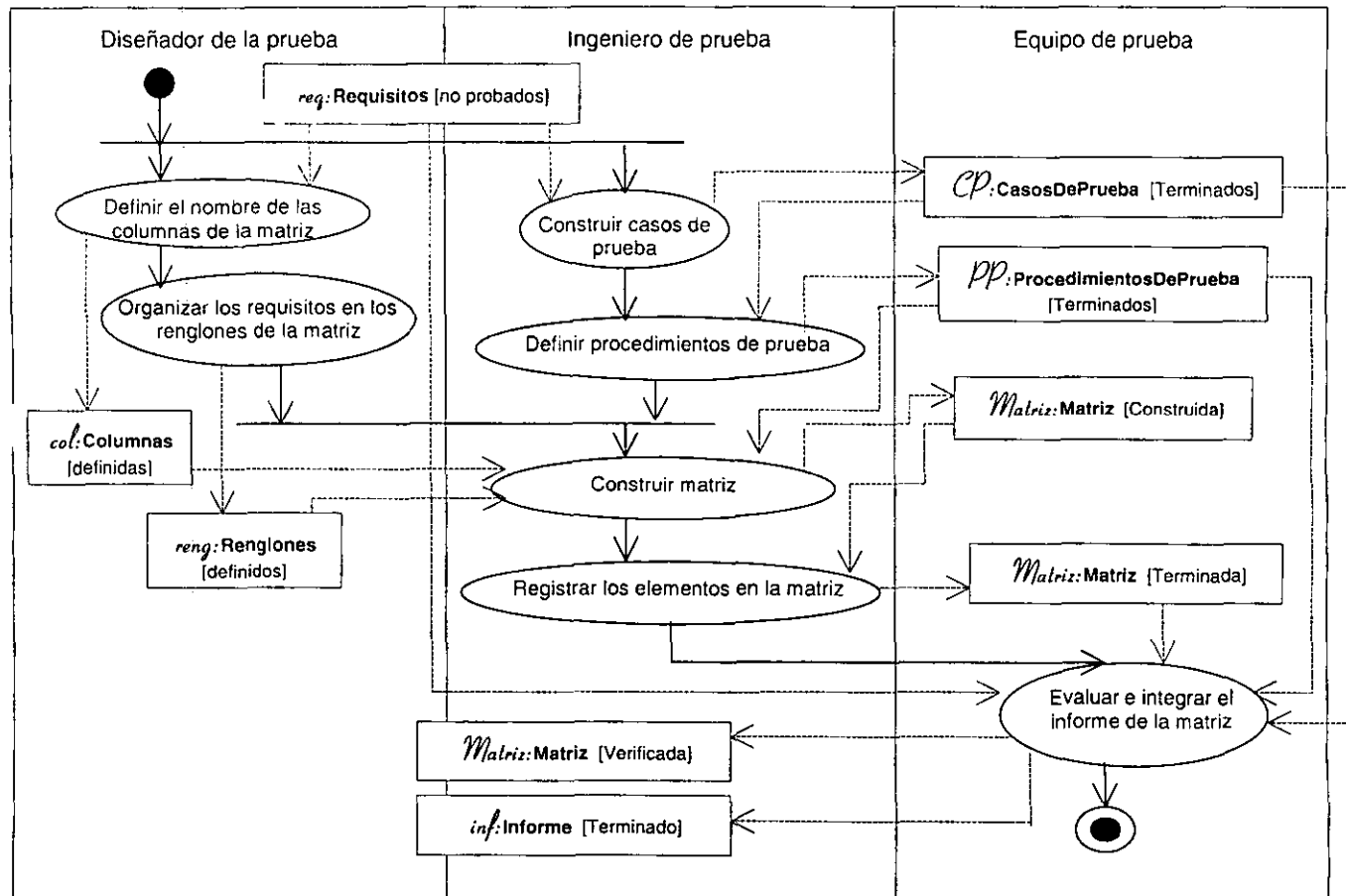


Figura A.6. Prueba de Requisitos con técnica: Apoyada en prototipos o modelos.

Diagrama de prueba: (Entrada *req*:Requisitos, Técnica *nlec*: Apoyada en prototipos o modelos, Salida *inf*:Informe).

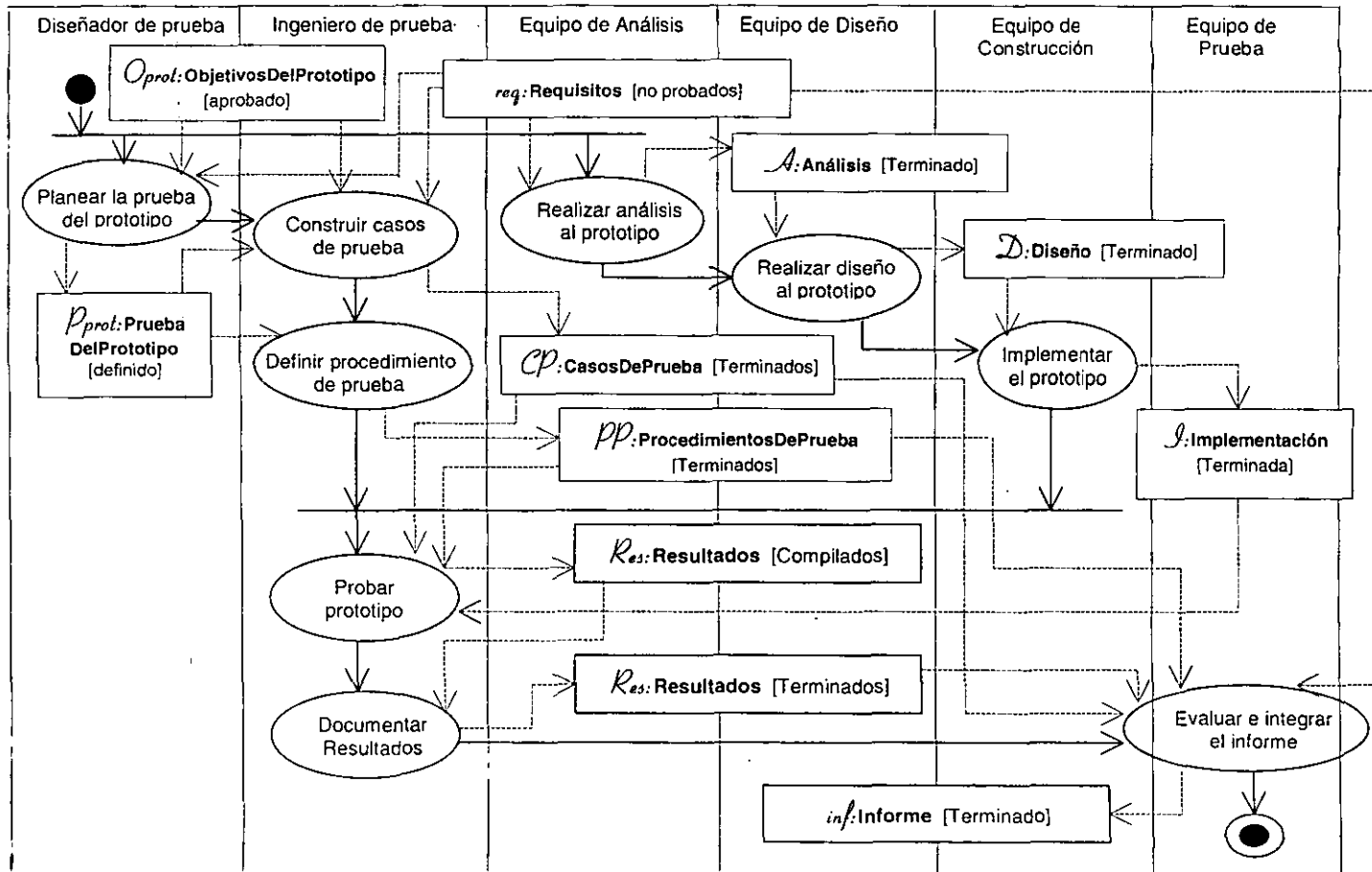


Figura A.7. Prueba de Requisitos con técnica: Apoyada en una lista de verificación.

Diagrama de prueba: (Entrada *req:Requisitos*, Técnica *nlec: Apoyada en una lista de verificación*, Salida *inf:Informe*).

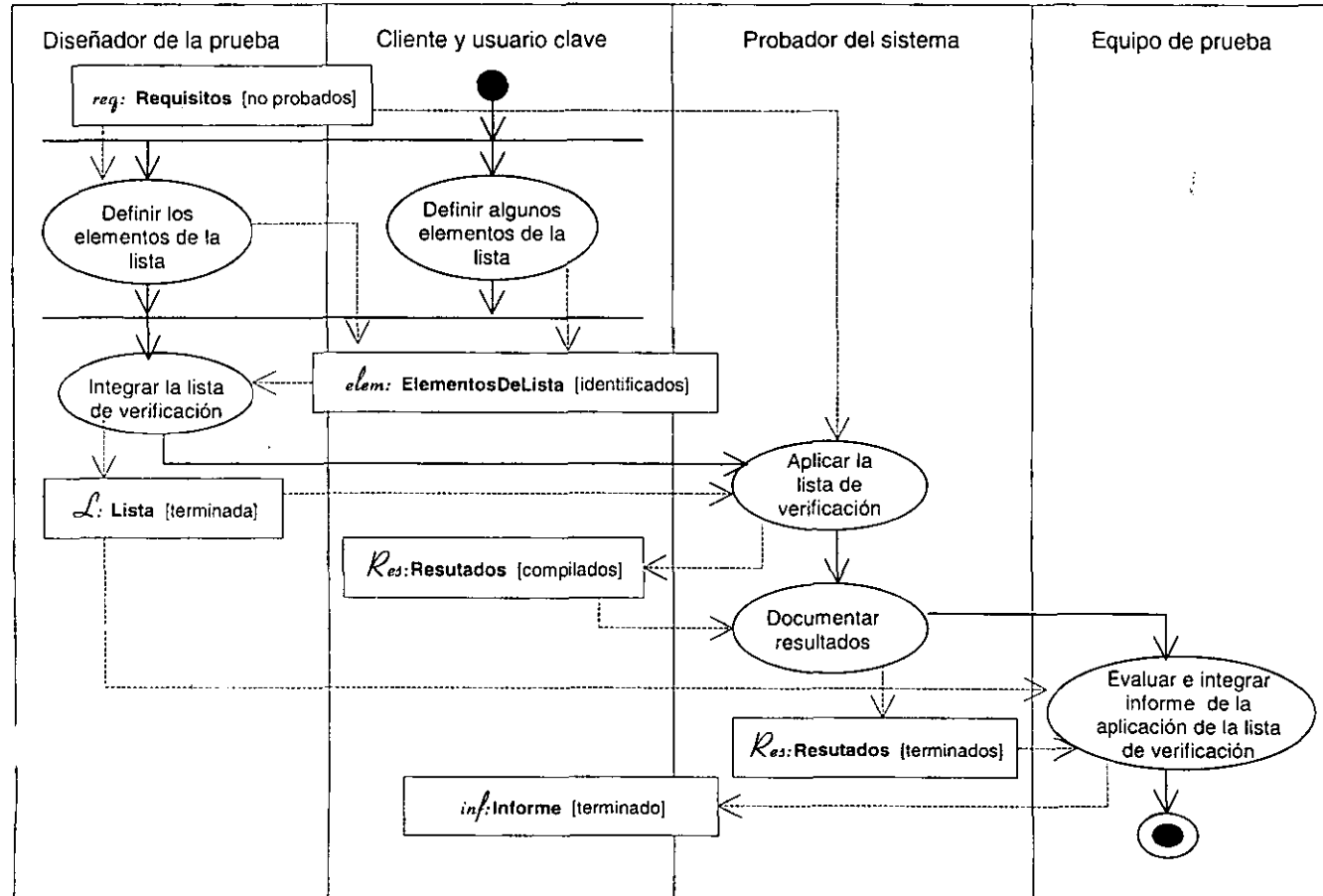


Figura A.9 Prueba del Diseño con técnica: Apoyada en el diseño de casos de prueba basados en el diseño.
 Diagrama de prueba: (Entrada *req*:Requisitos, *dis*:Diseño, Técnica *nlec*: Apoyada en el diseño de casos de prueba basados en el diseño, Salida *inf*:Informe).

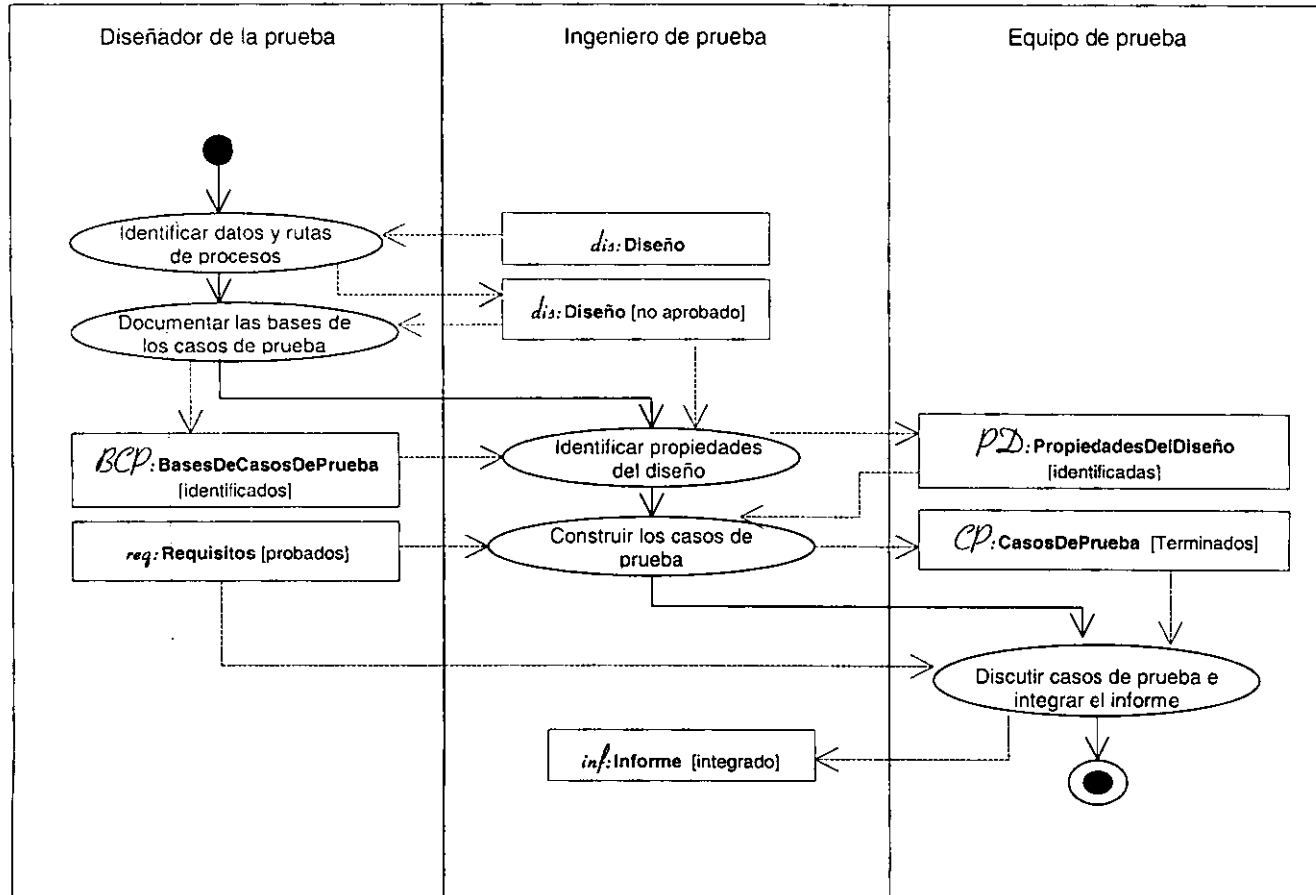


Figura A.10. Prueba del Diseño con técnica: Apoyada en modelos de prueba.

Diagrama de prueba: (Entrada *req*: **Requisitos del Modelo**, *dis*: **Diseño**, Técnica *tec*: **Apoyada en modelos de prueba**, Salida *inf*: **Informe**).

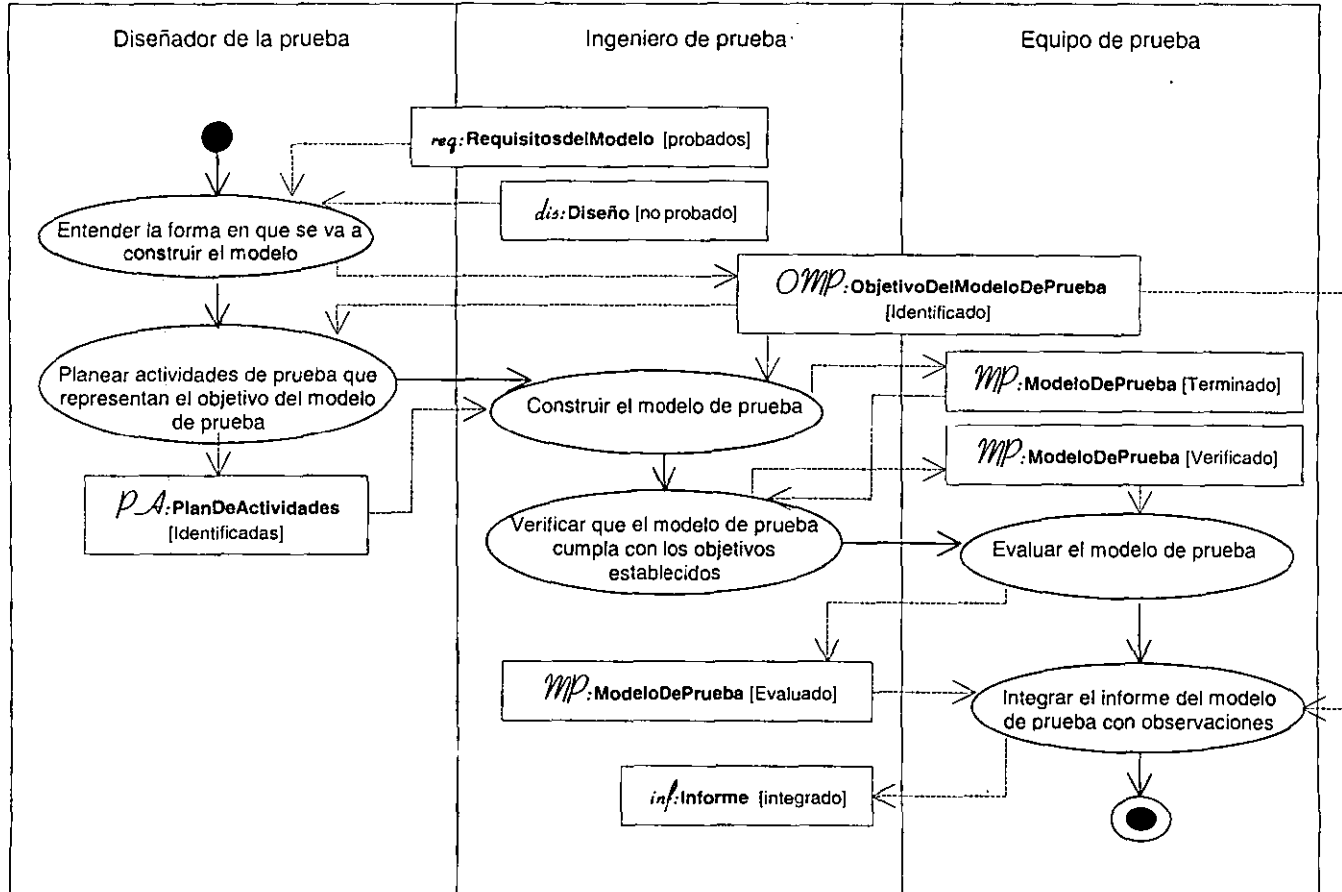
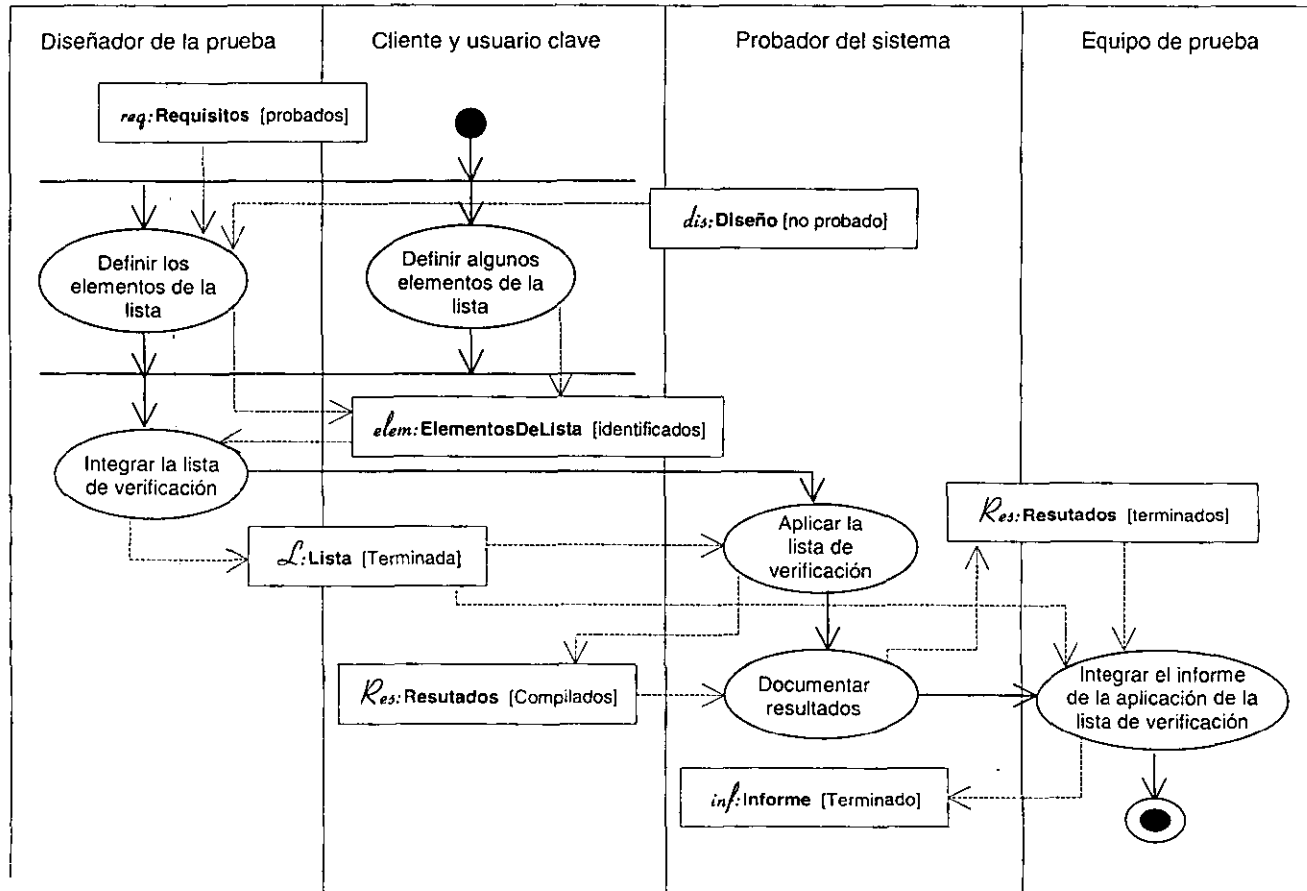


Figura A.11. Prueba del Diseño con técnica: Apoyada en una lista de verificación.

Diagrama de prueba: (Entrada *req*:Requisitos, *dis*:Diseño, Técnica *ntec*: Apoyada en una lista de verificación, Salida *inf*:Informe).



Anexo B



Lista de verificación

Aplicada a la etapa de: _____

Equipo: _____ Responsable del equipo: _____

...lista de verificación...

Desarrolló la prueba: _____ Aplicó la prueba: _____

Responsable(s) presente(s) en el momento de la prueba: _____
(siglas y firma)

Recibió el documento: _____ Fecha de entrega: _____
(siglas y firma)



Interpretación del resultado de la prueba

Aplicada a la etapa de: _____

Equipo: _____ Responsable del equipo: _____

Criterio de Prueba	Resultados/Recomendaciones

Realizó el informe: _____ Entregó el informe: _____

Responsable(s) presente(s) en el momento de la prueba: _____
(siglas y firma)

Recibió el documento: _____ Fecha de entrega: _____
(siglas y firma)



Informe de cambio

Aplicada a la etapa de: _____

Equipo: _____ Responsable del equipo: _____

Llenado por el responsable del equipo al que se le aplica la prueba.

1. Componente que sufrió cambio:
2. Breve descripción del cambio:
3. Versión del archivo anterior:
4. Versión del nuevo archivo:
5. Folio de la solicitud de cambio:
6. Fecha de aceptación de cambio:
7. Nombre y firma de quien brindó esta información: _____
(siglas y firma)
8. Fecha en que llenó éste informe:

Llenado por un integrante del equipo de pruebas.

1. Evaluación del impacto de la prueba: _____
(Descripción o anexar documento)
2. Nuevas consideraciones para el procedimiento de prueba:

(Descripción o anexar documento)
3. Archivo de prueba versión anterior:
4. Archivo de prueba nueva versión:
5. Nombre y firma de quien elaboró el formato: _____
(siglas y firma)
6. Fecha cuando se terminó el documento:



Documentación complementaria

Referente a la etapa de: _____

Equipo: _____ Responsable del equipo: _____

a) Objetivo del documento:

- ✓ Explicar el objetivo del documento en forma breve.

b) Contenido del documento:

- ✓ Explicar brevemente el formato del documento.
- ✓ El contenido del documento deberá ser claro y de acuerdo al objetivo propuesto

Realizó el documento: _____ Entregó el documento: _____

Recibió el documento: _____ Fecha de entrega: _____
(siglas y firma)

**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

Anexo C



Lista de verificación

Aplicada a la etapa de: Análisis. (Recolección de requisitos) (Primer prueba)

Equipo: Análisis. Responsable del equipo: JTC.

Esta forma debe ser llenada por un integrante del equipo de prueba. Cada pregunta se evalúa en una escala del 0 al 5, del peor resultado al mejor, respectivamente.

I. Requisitos.

A. Definición.

1. ¿Se encuentran debidamente documentados los requisitos?
2. ¿Se encuentra incluida la definición de éxito?
3. ¿Se han identificado condiciones de falla?
4. ¿Están los requisitos en un nivel suficientemente alto?
5. ¿Están los requisitos en un nivel suficientemente bajo?
6. ¿Están claramente estructurados los requisitos?
7. ¿Se encontraron debidamente ordenados los requisitos?
8. ¿Se encuentra considerada la calidad en los requisitos?
9. ¿Está claramente establecida la factibilidad?

2
0
0
2
2
1
1
0
0

Total

8

B. Elementos completos y consistentes.

1. ¿Fueron revisados los requisitos para verificar que se encuentran completos y consistentes?
2. ¿Los requisitos son los necesarios para especificar la solución del problema?
3. ¿Cada requisito no presenta conflictos con algún otro?
4. ¿Fueron revisados los requisitos por la persona correcta?
5. ¿Se hicieron los cambios necesarios para que se completaran los requisitos?

2
2
3
5
7

Total

12

C. Simplicidad y entendimiento.

1. ¿Se pueden entender la organización de los requisitos?
2. ¿Verifican las revisiones la simplicidad en los requisitos?
3. ¿Cada requisito es preciso, no confuso y claro?
4. ¿Se puede determinar si los requisitos satisfacen lo pedido por el cliente?
5. ¿Cuándo es cambiado algún requisito, no afecta a otros considerablemente?

2
3
2
3
2

6. ¿Se tomaron en cuenta los criterios de prueba, en la recolección de los requisitos? 2

7. ¿Se apoyaron los requisitos en la especificación del diseño? 1

Total 14

Desarrolló la prueba: DHC. Aplicó la prueba: DHC.

Responsable(s) presente(s) en el momento de la prueba: JTC.
(siglas y firma)

Recibió el documento: JTC. Fecha de entrega: 24/09/99.
(siglas y firma)



Interpretación del resultado de la prueba

Aplicada a la etapa de: Análisis. (Recolección de requisitos) (Primer prueba)

Equipo: Análisis Responsable del equipo: JTC.

Criterio de Prueba	Resultados / Recomendaciones
<p>A. Definición</p>	<p>Los requisitos no se encuentran debidamente documentados, están mezclados e incompletos. Nos presentan una idea general de las funcionalidades que deberá tener el sistema. Presenta faltas de ortografía y problemas de puntuación.</p> <p>Es necesario complementar la definición de los requisitos, ya que todavía se encuentran confusos y en algunos casos, es necesario completarlo con información que el cliente aún no ha proporcionado. Se deberán planear algunas reuniones con el cliente, con el fin de aclarar las dudas que se tienen e integrarlas a la información que ya se tienen identificados. Falta que se defina bajo qué condiciones el software se considera exitoso, y en cuales condiciones se considera una falla.</p> <p>Es necesario revisar la forma en que se redactaron los requisitos, para evitar repetición e inconsistencia de la formación, el cual se está presentando en algunas partes del documento.</p>
<p>B. Elementos completos y consistentes</p>	<p>Falta completar información que deben de contener los requisitos, y verificar la consistencia entre ellos. Se recomienda la creación de una lista detallada con la información que está confusa o incompleta para asegurar que se le pregunte al cliente.</p> <p>La confusión de algunos requisitos, provoca que sea difícil de entender la definición completa del software. Es necesario aplicar la prueba al menos un par de veces más, para asegurar que los cambios se han entendido y que éstos sean llevados a cabo.</p>

Criterio de Prueba	Resultados / Recomendaciones
<p>C. Simplicidad y entendimiento.</p>	<p>Se recomienda que los requisitos se ordenen por funcionalidades. Separando las que pertenecen al usuario y por separado, las que pertenecen al administrador y finalmente las que utilizan ambos. Dado que JTC, tiene un gran conocimiento sobre los términos que se manejan y en el proceso que se sigue en forma manual, en la <i>Inscripción de Aspirantes</i> y demas situaciones, ha pasado por alto algunas definiciones detalladas de términos y explicación de procesos. Es recomendable que JTC discuta los requisitos con alguna(s) persona(s) ajenas al equipo de análisis. Esto permitirá verificar si éstos se encuentran definidos correcta y claramente. Una vez terminada la siguiente versión de los requisitos, se recomienda una entrevista con el cliente para presentarle el documento y así verificar si éstos realmente expresan sus necesidades. De esta forma se evitan cambios que puedan impactar considerablemente en el proceso de desarrollo del software.</p> <p>Es necesario tomar en cuenta estas consideraciones, ya que permitirán la recolección de requisitos de forma óptima y las pruebas serán más fáciles de aplicar y se presentarán mejores resultados.</p>

Realizó el informe: DHC Entregó el informe: DHC

Responsable(s) presente(s) en el momento de la prueba: JTC
(siglas y firma)

Recibió el documento: JTC Fecha de entrega: 25/09/99
(siglas y firma)



Documentación complementaria

Referente a la etapa de: Planeación de la prueba.

Equipo: Pruebas Responsable del equipo: DHC.

a) **Objetivo del documento:** El siguiente documento presenta la planeación del esfuerzo de la prueba para SINSAS.

b) **Contenido del documento.**

Planeación del esfuerzo de la prueba: El esfuerzo se representa en días y cada día consta de 8 horas de actividad.

No	Actividades	Días empleados para cada actividad(8h)
1	Requisitos	4
2	Análisis	4
3	Diseño	6
4	Codificación	8
5	Integración	6
6	Pruebas del sistema	8
7	Interfaz	4
8	Manual técnico	3
9	Manual del usuario	2

Realizó el documento: DHC. Entregó el documento: DHC.

Recibió el documento: DHC. Fecha de entrega: 26/09/99.
 (siglas y firma)



Documentación complementaria

Referente a la etapa de: Planeación de la prueba.

Equipo: Pruebas Responsable del equipo: DHC.

c) **Objetivo del documento:** El presente documento muestra la planeación de las actividades de la prueba para SINSAS.

d) **Contenido del documento.**

Planeación de las actividades de prueba: El esfuerzo de la prueba, se representa en horas.

<i>Actividades</i>					
Requisitos	1er revisión (h)	2da revisión (h)	3er revisión (h)	4ta revisión (h)	Total
Revisión	7	7	4		
Informe	7	5	2		
Discusión	3	2	2		39
Análisis					
Revisión	9	8	5		
Informe	7	5	2		
Discusión	3	2	2		43
Diseño					
Revisión	10	10	7	5	
Informe	7	5	2	4	
Discusión	3	3	2	2	60
Codificación	1er versión (h)	2da versión (h)	3er versión (h)	4ta versión (h)	
Código	7	7	4	4	
Informe	7	7	3	3	
Ejecución	15	15	10	10	
Informe	7	7	5	5	
Discusión	3	3	2	2	126
Integración					
Revisión	15	20	20	15	
Informe	7	7	5	5	
Discusión	3	3	2	2	104

Pruebas del sistema						
Revisión	18	20	20	20		
Informe	12	9	9	9		
Discusión	4	4	3	3		131
Interfaz						
	1er revisión	2da revisión	3er revisión	4ta revisión		
Revisión	7	7	5	5		
Informe	7	5	4	3		
Discusión	3	2	2	2		52
Manual técnico						
Revisión	7	7	4			
Informe	4	2	2			
Discusión	3	2	2			33
Manual del usuario						
Revisión	7	6	4			
Informe	5	2	2			
Discusión	3	2	2			33
Total de horas en las actividades de prueba (esfuerzo de la prueba):						621

Realizó el documento: DHC Entregó el documento: DHC

Recibió el documento: DHC Fecha de entrega: 26/09/99
 (siglas y firma)



Documentación complementaria

Referente a la etapa de: Análisis

Equipo: Análisis. Responsable del equipo: JTC.

- e) **Objetivo del documento:** El siguiente documento tiene como finalidad explicar en forma breve la manera en que se llevarán a cabo las pruebas en la etapa de análisis.
- f) **Contenido del documento.**
- ✓ **Pruebas en los requisitos:** Cuando los requisitos se encuentran documentados, es necesario asegurar que la información que se compiló es suficiente para realizar el desarrollo de un software. Precisamente las pruebas ayudan a asegurar esto. Existen varias técnicas de prueba para llevar a cabo las actividades de prueba, y cada una de ellas depende del tipo de software que va a ser desarrollado. En el caso de SINSAS, se aplicará una lista de verificación para ratificar que los requisitos son claros y que no presentan redundancia o contradicciones.
 - ✓ **Forma de llevar a cabo las prueba:** Se acordarán al menos tres juntas con JTC para realizar las pruebas en la fase de análisis. En cada reunión será necesario contar la(s) versión(es) de los documentos que serán probados, en los formatos previamente establecidos. La prueba podrá ser aplicada en presencia de JTC, sin embargo no es obligatorio. Los resultados serán entregados en un plazo no mayor de 48 horas. Una vez entregado el resultado de la prueba, podrá ser discutido hasta antes que hayan concluido 24 horas. Esto evita dedicar demasiado tiempo en la realización y discusión de una sola actividad. Cada vez que sea entregado cualquier documento de prueba, será necesario firmar de recibido en original y copia, la copia deberá ser entregada al encargado de la fase en cuestión. Cualquier situación que de la prueba deberá ser discutida con DHC, en caso de no llegar a un acuerdo será necesario convocar a una reunión donde deberán estar presentes JTC, DHC y MIM.
 - ✓ **Realización de cambios:** Cada vez que se realice un cambio en una documentación ya probada y aprobada por el equipo de prueba, deberá llenarse el formato *FP-IC-03*, para informar de la decisión tomada al equipo de prueba. Cabe recordar que en la prueba se cuenta con un documento que contiene el modelado de SINSAS y éste debe estar siempre actualizado. En caso de no registrar los cambios se verá afectado el resultado de la prueba en la fase de análisis. Además será informado al líder MIM y al administrador del proyecto GRL.

✓ **Bibliografía adicional:** Si desea alguna bibliografía relacionada con las actividades de prueba en la fase de análisis, puede revisar la siguiente bibliografía:

1. HETZEL, Bill. *The Complete Guide to Software Testing*. John Wiley & Sons, Inc. USA1988. pp 57-64.
2. KIT, Edward. *Software Testing in the Real World*. Addison-Wesley. USA 1995. pp 130-141, 194-195.

Realizó el documento: DHC Entregó el documento: DHC

Recibió el documento: JTC Fecha de entrega: 26/09/99.
(siglas y firma)



Lista de verificación

Aplicada a la etapa de: Análisis. (Segunda prueba)

Equipo: Análisis. Responsable del equipo: JTC.

Esta forma debe ser llenada por un integrante del equipo de prueba. Cada pregunta se evalúa en una escala del 0 al 5, del peor resultado al mejor, respectivamente.

I. Requisitos.

A. Definición.

1. ¿Se encuentran debidamente documentados los requisitos?
2. ¿Se encuentra incluida la definición de éxito?
3. ¿Se han identificado condiciones de falla?
4. ¿Están los requisitos en un nivel suficientemente alto?
5. ¿Están los requisitos en un nivel suficientemente bajo?
6. ¿Están claramente estructurados los requisitos?
7. ¿Se encontraron debidamente ordenados los requisitos?
8. ¿Se encuentra considerada la calidad en los requisitos?
9. ¿Está claramente establecida la factibilidad?

Total 12

3
0
0
2
2
2
2
2
1
0

B. Elementos completos y consistentes.

1. ¿Fueron revisados los requisitos para verificar que se encuentran completos y consistentes?
2. ¿Los requisitos son los necesarios para especificar la solución del problema?
3. ¿Cada requisito no presenta conflictos con algún otro?
4. ¿Fueron revisados los requisitos por la persona correcta?
5. ¿Se hicieron los cambios necesarios para que se completaran los requisitos?

Total 16

3
3
3
5
2

C. Simplicidad y entendimiento.

1. ¿Se pueden entender la organización de los requisitos?
2. ¿Verifican las revisiones la simplicidad en los requisitos?
3. ¿Cada requisito es preciso, no confuso y claro?
4. ¿Se puede determinar si los requisitos satisfacen lo pedido por el cliente?
5. ¿Cuándo es cambiado algún requisito, no afecta a otros considerablemente?

2
2
2
2
2

6. ¿Se tomaron en cuenta los criterios de prueba, en la recolección de los requisitos? 3
 7. ¿Se apoyaron los requisitos en la especificación del diseño? 3

Total 18

II. Especificaciones

A. Definición.

1. ¿Fueron expresadas claramente las especificaciones? 3
 2. ¿Se revisó la estructura de las especificaciones? 3
 3. ¿Están las especificaciones en un nivel suficientemente alto? 3
 4. ¿Están las especificaciones en un nivel suficientemente bajo? 2
 5. ¿Estuvieron ordenadas las especificaciones? 3
 6. ¿Se encuentra incluida la calidad en las especificaciones? 2

Total 16

B. Elementos completos y consistentes.

1. ¿Se verificó que se encuentre definida completamente la especificación? 3
 2. ¿Fueron revisados por la persona correcta? 5
 3. ¿Se hicieron los cambios necesarios para completar la revisión? 7
 4. ¿Se revisó que las especificaciones fueran consistentes? 3
 5. ¿Se hicieron los cambios necesarios para que fueran consistentes? 7
 6. ¿Se aprobaron las especificaciones asegurando que se encuentra contenida toda su información dentro de los requisitos? 3
 7. ¿El desempeño fue especificado adecuadamente? 3

Total 17

C. Simplicidad y entendimiento.

1. ¿Se entienden claramente las especificaciones? 3
 2. ¿Se incluyeron las pruebas en las especificaciones? 2
 3. ¿Fueron identificados los elementos no claros en las especificaciones? 3
 4. ¿Se hizo énfasis en que las especificaciones fueran claras y simples? 4
 5. ¿Fueron claramente identificadas las salidas y las entradas? 2
 6. ¿Se incluyeron las razones de cada especificación? 2

Total 16

Desarrolló la prueba: DHC Aplicó la prueba: DHC

Responsable(s) presente(s) en el momento de la prueba: JTC
(siglas y firma)

Recibió el documento: JTC Fecha de entrega: 30/09/99
(siglas y firma)



Interpretación del resultado de la prueba

Aplicada a la etapa de: Análisis (Relacionado a la segunda prueba)

Equipo: Análisis Responsable del equipo: JTC.

Criterio de Prueba	Resultados / Recomendaciones
<p>I. Requisitos.</p> <p>A. Definición</p> <p>B. Elementos completos y consistentes</p> <p>C. Simplicidad y entendimiento.</p>	<p>Los requisitos aún no se encuentran corregidos en su totalidad. Algunas de los errores que fueron mencionados no fueron corregidos. Algunos requisitos todavía se encuentran incompletos o confusos. Presenta faltas de ortografía y problemas de puntuación. Es importante realizar las entrevistas con el cliente para aclarar las dudas que todavía se tienen. Todavía falta que se defina bajo qué condiciones el software se considera exitoso, y en cuales se considera una condición de falla. Se puede encontrar información repetida, será necesario revisarla con mas detalle para evitar esta situación.</p> <p>A pesar que se ha incrementado el resultado de la prueba, todavía aparecen requisitos incompletos, con problemas de definición y de inconsistencia. Es importante realizar la lista con el detalle de toda la información que aún se desconoce o se tienen dudas en su definición y presentarla al cliente para que tome las decisiones necesarias para su aclaración. Será necesario aplicar la prueba al menos una vez más, para asegurar que los cambios se realicen o, en caso necesario que DHC esté presente en las entrevistas con el cliente para aportar ideas o soluciones a los problemas que aún no se han resuelto.</p> <p>Se han identificado más requisitos, sin embargo es necesario que las funcionalidades sean agrupadas, como ya fue propuesto anteriormente. Todavía hace</p>

Criterio de Prueba	Resultados / Recomendaciones
<p data-bbox="156 517 338 560">II. Especificaciones. A. Definición.</p> <p data-bbox="201 876 424 920">B. Elementos completos y consistentes.</p>	<p data-bbox="500 230 1020 465">hace falta que se expliquen a mas detalle algunos procesos ya que se da por hecho que tenemos conocimiento del proceso de inscripción. Se recomienda que sea discutido el documento de los requisitos con miembros ajenos del equipo de diseño con el fin de identificar cual es la información que no se tiene completa y si la información que se tiene, se encuentra suficientemente clara. Se recomienda entrevistarse con el cliente para que sean discutidos esta versión de los requisitos.</p> <p data-bbox="500 517 1012 847">En el modelado de las especificaciones presentaron algunas omisiones. Es necesario tomar en cuenta los flujos normales y excepcionales. Hace falta agregar información para obtener las especificaciones en un nivel bajo. Para ello se necesita discutirlo con el equipo de análisis y llegar a un acuerdo hasta que grado será necesario detallar mas. Se recomienda separar las funcionalidades que pertenecen al aspirante y al administrador, esto permitirá que se entienda mejor el modelado. Es importante recordar que se esta trabajando con normas de calidad ya establecidas, en caso de desconocerlas, discutir esto y todo lo relacionado con Calidad, con VGC, el auditor interno.</p> <p data-bbox="500 876 1009 1333">El objetivo fundamental de esta prueba, es verificar que se tienen especificados todos los requisitos y que se encuentran especificados correctamente. Se encontró mas información en las especificaciones, que en los mismos requisitos. En ocasiones se detalla información que se encuentra solo en un documento, por ello no se sabe si es una inconsistencia o un error. Se recomienda revisar los dos documentos al mismo tiempo – requisitos y especificaciones, y resolver este conflicto, ya que si se espera mas tiempo, habrá serios problemas en el diseño. Las operaciones de Consultas y de EnvíodeCorreos electrónicos expresados en los requisitos, no fueron completamente expresados en las especificaciones. Algunos de ellos se encuentran modelados a medias y otros fueron omitidos. El administrador de la prueba, revisó en los tiempos y de la forma ya acordada con el responsable del equipo de Análisis. Solo falta discutir los resultados.</p>

Criterio de Prueba	Resultados / Recomendaciones
<p>C. Simplicidad y entendimiento.</p>	<p>Se presentan algunos nombres demasiado largos que pueden provocar problemas en la implementación. Se recomienda organizar de una forma mas sencilla la presentación de la información en los diagramas, quizá el pensar que cada escenario es un módulo del sistema (o funcionalidad), puede ayudar en la definición clara y simple de las especificaciones. Es importante que se pueda diferenciar entre una entrada y una salida, esto brindará un gran ayuda en la tarea del diseño y en la implementación.</p> <p>Se entregó al equipo de análisis las anotaciones que se deben de corregir, en el mismo documento que le fue entregado al probador por el equipo de análisis para su prueba.</p> <p>Esto permitirá señalar los problemas, precisamente en el lugar en donde fueron identificados.</p> <p>En caso de que JTC lo considere, podrá añadir mas detalles a los escenarios, para que estos no se puedan prestar a confusiones. Como por ejemplo, cuando se está leyendo un escenario, es necesario conocer cual fue el escenario que le precedió y cual será el siguiente. Se encuentran algunas faltas de ortografía y algunos problemas de puntuación.</p> <p>Se recomienda revisar el documento de las especificaciones primero y luego revisar los requisitos, de tal forma que se verifique, cuales son los requisitos que aún no se han definido completamente.</p> <p>Se recomienda realizar un glosario de términos, esto ayudará a entender conceptos y a mantenerlos en un solo documento, evitando repetir la explicación una y otra vez.</p>

Realizó el informe: DHC Entregó el informe: DHC

Responsable(s) presente(s) en el momento de la prueba: JTC
(siglas y firma)

Recibió el documento: JTC Fecha de entrega: 02/10/99
(siglas y firma)



Lista de verificación

Aplicada a la etapa de: Análisis. (Tercer prueba)

Equipo: Análisis Responsable del equipo: JTC

Esta forma debe ser llenada por un integrante del equipo de prueba. Cada pregunta se evalúa en una escala del 0 al 5, del peor resultado al mejor, respectivamente.

I. Requisitos.

A. Definición.

- | | |
|--|---|
| 1. ¿Se encuentran debidamente documentados los requisitos? | 4 |
| 2. ¿Se encuentra incluida la definición de éxito? | 4 |
| 3. ¿Se han identificado condiciones de falla? | 4 |
| 4. ¿Están los requisitos en un nivel suficientemente alto? | 4 |
| 5. ¿Están los requisitos en un nivel suficientemente bajo? | 4 |
| 6. ¿Están claramente estructurados los requisitos? | 5 |
| 7. ¿Se encontraron debidamente ordenados los requisitos? | 4 |
| 8. ¿Se encuentra considerada la calidad en los requisitos? | 4 |
| 9. ¿Está claramente establecida la factibilidad? | 4 |

Total 37

B. Elementos completos y consistentes.

- | | |
|--|---|
| 1. ¿Fueron revisados los requisitos para verificar que se encuentran completos y consistentes? | 5 |
| 2. ¿Los requisitos son los necesarios para especificar la solución del problema? | 4 |
| 3. ¿Cada requisito no presenta conflictos con algún otro? | 4 |
| 4. ¿Fueron revisados los requisitos por la persona correcta? | 5 |
| 5. ¿Se hicieron los cambios necesarios para que se completaran los requisitos? | 4 |

Total 22

C. Simplicidad y entendimiento.

- | | |
|--|---|
| 1. ¿Se pueden entender la organización de los requisitos? | 4 |
| 2. ¿Verifican las revisiones la simplicidad en los requisitos? | 5 |
| 3. ¿Cada requisito es preciso, no confuso y claro? | 4 |
| 4. ¿Se puede determinar si los requisitos satisfacen lo pedido por el cliente? | 4 |
| 5. ¿Cuándo es cambiado algún requisito, no afecta a otros considerablemente? | 4 |

6. ¿Se tomaron en cuenta los criterios de prueba, en la recolección de los requisitos? 4
7. ¿Se apoyaron los requisitos en la especificación del diseño? 4

Total 29

II. Especificaciones

A. Definición.

1. ¿Fueron expresadas claramente las especificaciones? 4
2. ¿Se revisó la estructura de las especificaciones? 5
3. ¿Están las especificaciones en un nivel suficientemente alto? 4
4. ¿Están las especificaciones en un nivel suficientemente bajo? 4
5. ¿Estuvieron ordenadas las especificaciones? 4
6. ¿Se encuentra incluida la calidad en las especificaciones? 4

Total 25

B. Elementos completos y consistentes.

1. ¿Se verificó que se encuentre definida completamente la especificación? 5
2. ¿Fueron revisados por la persona correcta? 5
3. ¿Se hicieron los cambios necesarios para completar la revisión? 4
4. ¿Se revisó que las especificaciones fueran consistentes? 5
5. ¿Se hicieron los cambios necesarios para que fueran consistentes? 4
6. ¿Se aprobaron las especificaciones asegurando que se encuentra contenida toda su información dentro de los requisitos? 5
7. ¿El desempeño fue especificado adecuadamente? 4

Total 32

C. Simplicidad y entendimiento.

1. ¿Se entienden claramente las especificaciones? 4
2. ¿Se incluyeron las pruebas en las especificaciones? 5
3. ¿Fueron identificados los elementos no claros en las especificaciones? 5
4. ¿Se hizo énfasis que las especificaciones fueran claras y simples? 4
5. ¿Fueron claramente identificadas las salidas y las entradas? 5
6. ¿Se incluyeron las razones de cada especificación? 4

Total 27

Desarrolló la prueba: DHC Aplicó la prueba: DHC

Responsable(s) presente(s) en el momento de la prueba: JTC
(siglas y

firma)

Recibió el documento: JTC Fecha de entrega: 10/10/99
(siglas y firma)



Documentación complementaria

Referente a la etapa de: Diseño

Equipo: Diseño Responsable del equipo: CAS

g) **Objetivo del documento:** El siguiente documento tiene como finalidad explicar en forma breve la manera en que se llevarán a cabo las pruebas en la etapa de diseño.

h) **Contenido del documento.**

- ✓ **Pruebas en los requisitos:** Se ha entregado una versión de los requisitos y demás documentos del análisis ya probados y aprobados. Es de mi conocimiento que ya se ha terminado la primera versión del diseño haciendo uso de los requisitos ya probados. En la etapa de diseño, la prueba ayudarán a ratificar que el modelado de SINSAS, se realizó tomando en cuenta lo expresado en los documentos del análisis. Para realizar la prueba en el diseño, se aplicará una lista de verificación conteniendo una serie de preguntas que nos ayudarán a asegurar que los requisitos estén incluidos, que el diseño sea claro y que no presenta redundancia o contradicciones.
- ✓ **Forma de llevar a cabo las prueba:** Se acordarán al menos tres juntas con CAS y/o CDV, para realizar las pruebas en la fase de diseño. En cada reunión será necesario contar la(s) versión(es) de los documentos que serán probados, en los formatos previamente establecidos. La prueba podrá ser aplicada en presencia del encargado de la fase de diseño, sin embargo no es obligatorio. Los resultados serán entregados en un plazo no mayor de 48 horas. Una vez entregado el resultado de la prueba, podrá ser discutido hasta antes que hayan concluido 24 horas. Esto evita dedicar demasiado tiempo en la realización y discusión de una sola actividad. Cada vez que sea entregado cualquier documento de prueba, será necesario firmar de recibido en original y copia, la copia deberá ser entregada a CAS. Cualquier situación que de la prueba deberá ser discutida con DHC, en caso de no llegar a un acuerdo será necesario convocar a una reunión donde deberán estar presentes JTC, DHC y MIM.
- ✓ **Realización de cambios:** Cada vez que se realice un cambio en una documentación ya probada por el equipo de prueba, deberá llenarse el formato *FP-IC-03*, para informar de la decisión tomada al equipo de prueba. Cabe recordar que en la prueba se cuenta con un documento que contiene el modelado de SINSAS y éste debe estar siempre actualizado. En caso de no registrar los cambios se verá afectado el resultado de la prueba en la fase de análisis. Además será informado al líder MIM y al administrador del proyecto GRL.

✓ **Bibliografía adicional:** Si desea alguna bibliografía relacionada con las actividades de prueba en la fase de análisis, puede revisar la siguiente bibliografía:

3. HETZEL, Bill. *The Complete Guide to Software Testing*. John Wiley & Sons, Inc. USA1988. pp 57-64.
4. KIT, Edward. *Software Testing in the Real World*. Addison-Wesley. USA 1995. pp 130-141, 194-195.

Realizó el documento: DHC Entregó el documento: DHC
Recibió el documento: CAS Fecha de entrega: 26/09/99
(siglas y firma)



Lista de verificación

Aplicada a la etapa de: Diseño. (Primer prueba)

Equipo: Diseño Responsable del equipo: CAS

Esta forma debe ser llenada por un integrante del equipo de prueba. Cada pregunta se evalúa en una escala del 0 al 5, del peor resultado al mejor, respectivamente.

I. Perspectiva general

A. Definición.

- | | |
|---|---|
| 1. ¿Se produjo un documento en la etapa de diseño? | 5 |
| 2. ¿Presentó opciones y alternativas clave el documento del diseño? | 5 |
| 3. ¿Estas opciones fueron debatidas y evaluadas? | 5 |
| 4. ¿Fomentaron competencia entre ideas? | 3 |
| 5. ¿Se asignó una arquitectura principal? | 5 |
| 6. ¿Fue claro el diseño de los objetivos? | 4 |
| 7. ¿Fue claramente establecida la factibilidad? | 4 |
| 8. ¿Presentó un mecanismo de cambio efectivo? | 4 |
| 9. ¿Fue usado de forma efectiva el mecanismo de cambio? | 4 |

Total 40

B. Elementos completos y consistentes.

- | | |
|---|---|
| 1. ¿Se pudo revisar el diseño en forma efectiva? | 4 |
| 2. ¿Fue revisado por la persona correcta? | 5 |
| 3. ¿Se encuentra el diseño completo? | 3 |
| 4. ¿El diseño fue consistente de acuerdo al análisis? | 5 |
| 5. ¿Al probarse el diseño, asegura que la especificación se conoce? | 4 |
| 6. ¿Fueron identificadas las áreas de riesgos fácilmente? | 2 |
| 7. ¿Fue llevada a cabo la revisión y la prueba fácilmente? | 5 |

Total 28

C. Simplicidad y entendimiento.

- | | |
|--|---|
| 1. ¿Es entendible el diseño? | 5 |
| 2. ¿Se estructuró el diseño claramente en distintos niveles? | 3 |
| 3. ¿Fue efectivo el énfasis a la simplicidad en el diseño? | 4 |
| 4. ¿Son claras las entradas y salidas de cada nivel? | 3 |
| 5. ¿Cada módulo presenta un propósito diferente? | 5 |
| 6. ¿Son claras las entradas y salidas de cada módulo? | 2 |

- | | |
|---|---|
| 7. ¿El diseño del sistema presenta una vista total además de tener incluido un flujo esquemático? | 3 |
| 8. ¿Están documentadas las razones de cambios de cada módulo principal? | 3 |
| 9. ¿Son descritos claramente los datos principales? | 4 |
| 10. ¿Están documentadas las razones para los datos principales? | 4 |
| 11. ¿Entiende el diseño la gente apropiada? | 5 |
| 12. ¿Se entienden las implicaciones del diseño? | 4 |
| 13. ¿Se entienden las alternativas de diseño? | 3 |
| 14. ¿Se desarrolló y se revisó el plan de prueba durante la etapa de diseño? | 3 |

Total 52

D. Alcance técnico

- | | |
|---|---|
| 1. ¿La metodología del diseño sigue los principios sobre el diseño y el análisis? | 4 |
| 2. ¿Están en el diccionario de datos la definición y descripción de los datos? | 4 |
| 3. ¿Se utilizó en el diseño la tecnología apropiada? | 5 |
| 4. ¿Utilizan los datos y los archivos estructurados los principios de los sistemas de bases de datos? | 4 |
| 5. ¿Fue tomada en cuenta la consideración de reusabilidad? | 3 |
| 6. ¿El diseño construye un procedimiento (de diseño) para aplicaciones similares? | 3 |

Total 23

Desarrolló la prueba: DHC Aplicó la prueba: DHC

Responsable(s) presente(s) en el momento de la prueba: CAS CDV

(siglas y firma)

Recibió el documento: CAS CDV Fecha de entrega: 08/10/99

(siglas y firma)



Interpretación del resultado de la prueba

Aplicada a la etapa de: Diseño. (Relacionado a la primer prueba)

Equipo: Diseño. Responsable del equipo: CAS.

Criterio de Prueba	Resultados / Recomendaciones
<p>I. Perspectiva General.</p> <p>A. Definición</p> <p>B. Elementos completos y consistentes.</p> <p>C. Simplicidad y Entendimiento.</p>	<p>El documento del diseño superó las expectativas, no presenta demasiados problemas debido a que le fue entregada al equipo, una versión del análisis ya probada en dos ocasiones. También cabe mencionar que el trabajo realizado por CAS y CDV, es claro y bien detallado.</p> <p>En la parte de Definición, se recomienda definir un mecanismo de cambio que sea efectivo. Esto ahorrará problemas cuando sea necesario realizar cambios en el diseño, también ayudará a conocer el impacto de cada cambio y tener argumentos para respaldar la decisión de realizarlos..</p> <p>El diseño no se ha completado debido a que aún se han realizado las adecuaciones en algunas partes del análisis. Sin embargo CAS asegura que se tiene identificada la parte que está haciendo falta. El equipo de Diseño no ha identificado completamente las áreas de riesgo, CAS se compromete tenerlas definidas en la siguiente versión del diseño.</p> <p>Solo se han identificado cuatro niveles en la estructura del diseño, CDV y CAS comentaron que serán revisados para asegurarse que está correcto y detallarlos un poco más. En caso de no definirlos, la visión global del sistema no podrá quedar expresado. Falta documentar las entradas y salidas a nivel módulo. Algunos diagramas se encuentran sobreentendidos, deberán detallarse y de ser necesario corregirse, ya que en caso contrario provocará conflictos en la implementación.</p>

Criterio de Prueba	Resultados / Recomendaciones
D. Alcance técnico.	<p>El flujo del sistema se presenta en forma general, falta detallarlo y documentarlo. En caso contrario, quedarían incompletos los diagramas de interacción y por tanto también el diseño. En la primera etapa del diseño –modelado de los diagramas de interacción, fueron realizadas las pruebas cuando estos fueron terminados por completo. Sin embargo se acordó con CAS, que en las siguientes etapas serán discutidas las posibles dudas que surjan en el modelado, en una junta con DHC, antes que se dé por finalizado el diseño completo.</p> <p>Debido a la complejidad del modelado, se han presentado algunas complicaciones en la obtención de algunos diagramas. Por esto se ha tardado en terminar el diseño. Se han tenido algunos conflictos con la longitud de algunos nombres de las clases y de sus atributos, se decidió discutir esta situación con la encargada del análisis JTC y con el líder de proyecto MIM, para definir si se modifican los nombres que son demasiado largos, para no acarrear esta situación hasta la construcción. En cuestión de reusabilidad apenas se ha tomado en cuenta este concepto, por ello se discutió con CAS y CDV para su consideración y aplicarlo en el modelado en caso de ser necesario.</p>

Realizó el informe: DHC Entregó el informe: DHC

Responsable(s) presente(s) en el momento de la prueba: CAS CDV
(siglas y firma)

Recibió el documento: CAS CDV Fecha de entrega: 10/10/99
(siglas y firma)



Lista de verificación

Aplicada a la etapa de: Diseño. (Segunda prueba)

Equipo: Diseño. Responsable del equipo: CAS.

Esta forma debe ser llenada por un integrante del equipo de prueba. Cada pregunta se evalúa en una escala del 0 al 5, del peor resultado al mejor, respectivamente.

I. Perspectiva general

A. Definición.

1. ¿Se produjo un documento en la etapa de diseño?
2. ¿Presentó opciones y alternativas clave el documento del diseño?
3. ¿Estas opciones fueron debatidas y evaluadas?
4. ¿Fomentaron competencia entre ideas?
5. ¿Se asignó una arquitectura principal?
6. ¿Fue claro el diseño de los objetivos?
7. ¿Fue claramente establecida la factibilidad?
8. ¿Presentó un mecanismo de cambio efectivo?
9. ¿Fue usado de forma efectiva el mecanismo de cambio?

Total 42

5
5
5
5
5
5
4
4
4

B. Elementos completos y consistentes.

1. ¿Se pudo revisar el diseño en forma efectiva?
2. ¿Fue revisado por la persona correcta?
3. ¿Se encuentra el diseño completo?
4. ¿El diseño fue consistente de acuerdo al análisis?
5. ¿Al probarse el diseño, asegura que la especificación se conoce?
6. ¿Fueron identificadas las áreas de riesgos fácilmente?
7. ¿Fue llevada a cabo la revisión y la prueba fácilmente?

Total 34

5
5
5
5
5
5
4
5

C. Simplicidad y entendimiento.

1. ¿Es entendible el diseño?
2. ¿Se estructuró el diseño claramente en distintos niveles?
3. ¿Fue efectivo el énfasis a la simplicidad en el diseño?
4. ¿Son claras las entradas y salidas de cada nivel?
5. ¿Cada módulo presenta un propósito diferente?
6. ¿Son claras las entradas y salidas de cada módulo?
7. ¿El diseño del sistema presenta una vista total además de tener incluido un flujo esquemático?

5
5
5
5
5
5
5

1. ¿Están documentadas las razones de cambios para cada módulo principal?
2. ¿Son descritos claramente los datos principales?
3. ¿Están documentadas las razones para los datos principales?
4. ¿Entiende el diseño la gente apropiada?
5. ¿Se entienden las implicaciones del diseño?
6. ¿Se entienden las alternativas de diseño?
7. ¿Se desarrolló y se revisó el plan de prueba durante la etapa de diseño?

4
5
5
5
5
5
5

Total 69

D. Alcance técnico

1. ¿La metodología del diseño sigue los principios sobre el diseño y el análisis?
2. ¿Están en el diccionario de datos la definición y descripción de los datos?
3. ¿Se utilizó en el diseño la tecnología apropiada?
4. ¿Utilizan los datos y los archivos estructurados los principios de los sistemas de bases de datos?
5. ¿Fue tomada en cuenta la consideración de reusabilidad?
6. ¿El diseño construye un procedimiento (de diseño) para aplicaciones similares?

5
5
5
4
4
4

Total 27

Desarrolló la prueba: DHC Aplicó la prueba: DHC

Responsable(s) presente(s) en el momento de la prueba: CAS CDV
(siglas y firma)

Recibió el documento: CAS CDV Fecha de entrega: 15/10/99
(siglas y firma)



Interpretación del resultado de la prueba

Aplicada a la etapa de: Diseño. (Relacionado a la segunda prueba)

Equipo: Diseño, Responsable del equipo: CAS.

Criterio de Prueba	Resultados / Recomendaciones
<p>I. Perspectiva General.</p> <p>A. Definición</p> <p>B. Elementos completos y consistentes.</p> <p>C. Simplicidad y Entendimiento.</p>	<p>En la segunda versión del diseño se ha corregido en un 95% las observaciones realizadas por DHC. CAS y CDV han realizado una excelente labor en la obtención de los documentos del diseño. A continuación se describen algunos detalles que faltan.</p> <p>Quedan algunas dudas sobre las actividades que hay que realizar en caso que se presenten cambios al diseño. Todavía se tienen dudas sobre el impacto que le puede causar a las fases posteriores y al diseño. Sin embargo MIM y CAS, se han comprometido a resolver esta situación e incluirla en el diseño.</p> <p>Se han identificado las áreas de riesgo, sin embargo al discutir la segunda revisión del diseño, se presentaron algunas situaciones que no se había previsto. CAS asegura que será incluida en la próxima versión del diseño.</p> <p>Se tienen algunas dudas sobre los cambios que puede presentar el módulo, principal. Por ello se discutió esta situación con MIM, quien aseguró comunicar esta información al equipo de diseño, para que quede expresado en la siguiente versión del modelo.</p>

Criterio de Prueba	Resultados / Recomendaciones
D. Alcance técnico.	<p>Todavía persisten los problemas con algunos nombres de las clases y sus métodos. Sin embargo JTC, CAS y MIM, han decidido dejarlos de este modo. DHC ha comentado que pueden generar problemas en el momento de la implementación. Con respecto a la reusabilidad, no se le ha dedicado el suficiente tiempo, CAS argumenta no será necesario ya que el tamaño del software no lo amerita, sin embargo comentó que será estudiado con detenimiento.</p> <p>MIM, GRL y VGC, discutieron con DHC para acordar que no es necesario aplicar una prueba más en esta fase argumentando que se realizarán los cambios. Por tanto se da por terminada las actividades de prueba aplicadas a la fase de diseño.</p>

Realizó el informe: DHC Entregó el informe: DHC

Responsable(s) presente(s) en el momento de la prueba: CAS CDV
(siglas y firma)

Recibió el documento: CAS CDV Fecha de entrega: 16/10/99
(siglas y firma)