



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Instituto de Ciencias Matemáticas

**“DISEÑO DE UN SISTEMA DE CALIDAD DE
PRUEBAS DE SOFTWARE PARA UNA EMPRESA
DE SERVICIOS BASADO EN EL ESTÁNDAR ISO
9126”**

PROYECTO DE GRADUACIÓN

Previa a la obtención del Título de:

AUDITOR EN CONTROL DE GESTIÓN

Presentada por:

Dorothy Katusca Hurtado Sáenz

Guayaquil-Ecuador

2012

AGRADECIMIENTO

Mi agradecimiento en primer lugar es a Dios por la vida y por darme la oportunidad de formarme como profesional en la Universidad Politécnica del Litoral, a mi Directora de tesis, Msc. Alice Naranjo.

A mi familia, quienes me guiaron en la formación académica y humana y a todos aquellos que me apoyaron de una u otra manera.

DEDICATORIA

A mis padres y hermanos, por ser las personas que me inculcaron el espíritu de responsabilidad y superación desde muy niña, quienes con sacrificios y esmeros han logrado formarme para poder ser quien soy en estos momentos; **a mi hijo Andy**, fruto de un inmenso amor y cariño, quien me da fuerzas para seguir superándome y preparándome para los nuevos retos que se presenten en mi vida.

TRIBUNAL DE GRADUACION

**Ing. Dalton Noboa
DELEGADO**

**Msc. Alice Naranjo
DIRECTORA DEL PROYECTO DE
GRADUACIÓN**

**Ing. Miguel Chang
VOCAL**

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta tesis de grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de graduación de la ESPOL)

Dorothy Katusca Hurtado Sáenz

RESUMEN

El presente estudio se enfocó en el desarrollo de las pruebas de calidad del software que han venido evolucionando desde que era una tarea de unos pocos expertos para satisfacer necesidades de grupos reducidos de gran poder económico a ser una tarea de algunas personas que satisfacen necesidades para pequeñas y medianas empresas, lo que si se ha mantenido es la enorme presión porque estos sistemas sean desarrollados y puestos en producción con el menor costo posible y en el menor tiempo. De allí que la labor de desarrollar software se vaya tornando más compleja cada vez y los productos resultantes dejen mucho que desear.

Pero no sólo esto afecta a la labor del personal de sistemas sino también la rapidez de cambio, en necesidades, en formas de uso, en lenguajes, herramientas, sistemas operativos, esquemas de redes y en equipos donde se ejecutará el software. La tecnología tan cambiante y en constante evolución es otra cosa a la que se deben enfrentar.

El desarrollo de un sistema desde cero, es decir usando una metodología como es la del ciclo de vida, está sujeto a muchas equivocaciones que generan defectos que más tarde se manifestarán como fallas del sistema, lo que degenerará la operatividad de la empresa.

Las pruebas mejoran la integridad de un sistema, al detectar las desviaciones del diseño y los errores en el mismo. Las pruebas también incrementan el valor del producto, al adaptarlo a las necesidades del usuario

INDICE GENERAL

	Pág.
RESUMEN.....	I
INDICE GENERAL.....	II
ABREVIATURAS.....	V
INDICE DE FIGURAS.....	VII
INDICE DE TABLAS.....	IX
INTRODUCCIÓN.....	X
CAPITULO 1	
I. ANTECEDENTES	
1.1 Evolución de los Sistemas de Información.....	1
1.2 Ciclo de Vida de un Sistema de Información.....	.9
1.3 Ciclo de Vida Clásico del Desarrollo del Sistema.....	10
1.4 Método de Desarrollo por análisis Estructurado.....	21
1.5 Método del Prototipo de Sistemas.....	27
CAPITULO 2	
II. MARCO TEORICO	
2.1. Metodologías para desarrollo de Pruebas de sistemas.....	30
2.1.1. Introducción.....	30
2.1.2. Tipos de pruebas.....	31
2.1.3. Concepto de Metodología.....	39
2.1.3.1. Tipos de Metodología.....	40
2.1.3.2. Metodologías de desarrollo de pruebas más comunes.....	41
2.1.3.3. Metodología de Pruebas de Sistema según estándar IEEE.....	43
2.2. Calidad de Software.....	54
2.2.1. Concepto de calidad de Software.....	55
2.2.2. Sistema de calidad.....	57

2.2.3. Tipos de Errores.....	61
2.2.3.1. Manejo de Proceso para garantizar la calidad.....	62
2.2.3.2. Niveles de Seguridad.....	62
2.2.4. Métricas.....	64
2.2.4.1. Concepto de métricas.....	64
2.2.4.2. Clasificación de métricas.....	66
2.2.4.3. Ejemplo de métricas.....	68
2.3. Definiciones Conceptuales.....	70

CAPITULO 3

III. NORMAS Y/O ESTÁNDARES INTERNACIONES

3.1 Estándares de calidad de Software.....	76
3.1.1. ISO 9000.....	77
3.1.2. ISO 9001.....	80
3.1.3. ISO 9126.....	87
3.1.4. ISO /IEC 29119.....	92

CAPITULO 4

IV. Metodología de resultados

4.1 Antecedente.....	106
4.2 Introducción.....	106
4.3 Objetivos de trabajo Diseño de un Sistema de Calidad de pruebas de software para una empresa de servicio basado en el estándar ISO 9126.....	107
4.4 Metodología de investigación.....	108
4.4.1. Técnicas e Instrumentos de Investigación de resultados..	109
4.4.2. Análisis e Interpretación de Investigación.....	133
4.5 Presentación del sistema de calidad de pruebas.....	134
4.5.1. Metodología de pruebas diseñadas.....	134
4.5.2. Fases de la metodología de prueba.....	135
4.5.3. Aplicación de la metodología establecida en el caso práctico.....	158

CONCLUSIONES Y RECOMENDACIONES.....	175
--	------------

BIBLIOGRAFÍA.....	177
--------------------------	------------

ANEXOS.....	183
--------------------	------------

ABREVIATURAS

Las abreviaturas presentadas en la siguiente tesis son las siguientes:

- TI:** Tecnologías de la Información.
- SPICE:** Software Process Improvement and Capability Determination.
- CASE:** Computer Aided Software Engineering (Ingeniería de Software Asistida por Ordenador)
- CMM:** Capability Maturity Model (Modelo de Capacidad y Madurez).
- CMMI:** Capability Maturity Model Integration (Modelo de Capacidad y Madurez Integrado).
- GQM:** Goals Questions Metrics (Metas Preguntas Métricas)
- ISO:** International Organization for Standardization (Organización Internacional para la Estandarización).
- IEC:** International Electrotechnical Commission (Comisión Electrotécnica Internacional).
- MSF:** Microsoft Solution Framework.
- SEI:** Software Engineering Institute.

PYMES: Pequeñas y Medianas empresas.

ERP: Enterprise Resource Planning (Planeación de Recursos de la Empresa).

TPS: Transactional Processing Systems (Sistemas de Procesamiento Transaccional).

CRM: Customer Relationship Management (Administración de la Relación con los Clientes).

MIS: Management Information Systems (Sistemas de Información Gerencial).

DSS: Decisions Support Systems (Sistemas de Soporte a Decisiones).

INDICE DE FIGURAS

FIGURA 1.1 Sistema de información de una organización

FIGURA 2.1 Sistema de seguridad

FIGURA 2.2 Tipos de seguridad

FIGURA 2.4 Niveles de seguridad

FIGURA 2.5 Perfil del usuario

FIGURA 2.6 Función del perfil del usuario.

FIGURA 2.7 Método de datos de prueba

FIGURA 2.8 Evaluación del sistema de caso base

FIGURA 2.9 Operación paralela

FIGURA 2.10 Facilidad de prueba Integrada A

FIGURA 2.11 Facilidad de prueba Integrada B

FIGURA 2.12 Selección de transacciones de entrada

FIGURA 2.13 Archivo de revisión de auditoría (SCARF)

FIGURA 2.14 Archivo de revisión de auditoría por muestreo (SARF)

FIGURA 2.15 Registros extendidos A

FIGURA 2.16 Aplicación de software de auditoría

FIGURA 2.17 Vaciado de archivos

FIGURA 2.18 Esquema básico de una metodología en riesgo

FIGURA 2.19 Fases de la prueba del Sistema

FIGURA 2.20 Documentación del diseño de pruebas

FIGURA 2.21 Proceso de la ejecución de pruebas

FIGURA 2.22 Proceso de la documentación de la ejecución de pruebas

FIGURA 2.23 Ciclo de vida de las pruebas

INDICE DE TABLAS

TABLA I Categoría de los sistemas de información

TABLA II Informe del resumen de las pruebas

INDICE DE CUADROS

CUADRO N° 2.1 Frecuencia de la documentación en el proceso del desarrollo de software.

CUADRO N° 2.2 Importancia de la documentación en el proceso del desarrollo de software.

CUADRO N°2.3 Incorporación de la documentación en el proceso de desarrollo de software.

CUADRO N° 3.3 Roles Definidos

CUADRO N° 3.4 Roles para la equipo de organización del proyecto.

CUADRO N° 3.5 Roles para el rol del equipo de desarrollo

CUADRO N° 4.1 Esfuerzos sobre la calidad de software.

CUADRO N° 4.2 Aplicación de la metodología en la calidad del producto final

CUADRO N° 4.3 Metodologías y técnicas de calidad de software

CUADRO N° 4.4 Ventajas de las metodologías de calidad de software

CUADRO N° 4.5 Conoce organizaciones de estándares de calidad

CUADRO N° 4.6 Organizaciones de estándares de calidad

CUADRO N° 4.8 Estándar de Calidad ISO 9126

CUADRO N° 4.9 Atributos ISO 9126

CUADRO N° 5.1 Pruebas en el desarrollo

CUADRO N° 5.2 Control de los errores

CUADRO N° 5.3 Tipo de Pruebas

CUADRO N° 5.4 Porcentaje de defectos en las pruebas

GRAFICOS

GRÁFICO N° 2.1 Frecuencia de la documentación en el proceso del desarrollo de software.

GRÁFICO N° 2.2 Importancia de la documentación en el proceso del desarrollo de software.

GRÁFICO N°2.3 Incorporación de la documentación en el proceso de desarrollo de software.

GRÁFICO N° 3.3 Roles Definidos

GRÁFICO N° 3.4 Roles para la equipo de organización del proyecto.

GRÁFICO N° 3.5 Roles para el rol del equipo de desarrollo

GRÁFICO N° 4.1 Esfuerzos sobre la calidad de software.

GRÁFICO N° 4.2 Aplicación de la metodología en la calidad del producto final

GRÁFICO N° 4.3 Metodologías y técnicas de calidad de software

GRÁFICO N° 4.4 Ventajas de las metodologías de calidad de software

GRÁFICO N° 4.5 Conoce organizaciones de estándares de calidad

GRÁFICO N° 4.6 Organizaciones de estándares de calidad

GRÁFICO N° 4.8 Estándar de Calidad ISO 9126

GRÁFICO N° 4.9 Atributos ISO 9126

GRÁFICO N° 5.1 Pruebas en el desarrollo

GRÁFICO N° 5.2 Control de los errores

GRÁFICO N° 5.3 Tipo de Pruebas

GRÁFICO N° 5.4 Porcentaje de defectos en las pruebas

INTRODUCCIÓN

En la actualidad, el software de empresa se ha tornado cada vez más complejo, debiendo resistir cambios frecuentes en su plataforma de hardware y aún en el software, adaptándose a ambientes diversos e incluso hasta estar disponibles en entornos difíciles e inseguros como internet.

Surgiendo por consiguiente la necesidad de efectuar pruebas de todo tipo que garanticen la calidad de los sistemas.

Independientemente del método de desarrollo de software, se hace necesario el uso de pruebas. La necesidad surge de la propensión humana a cometer equivocaciones, que, en el caso del software, originan defectos que pueden provocar fallas al momento de ejecución e incluso en constituirse en problemas legales para los desarrolladores.

Además de equivocaciones humanas existen fallas del hardware donde el software se ejecuta, fallas en el sistema operativo y errores de interacción con equipos, y con los seres humanos que complican el panorama de las pruebas.

Conforme el software se torna más complejo, las pruebas son más necesarias para asegurar el cumplimiento de ciertos criterios mínimos de conformidad de operatividad y ayudan a garantizar la calidad del software. A pesar de su importancia, las pruebas resultan un tema mal visto por

algunos y desconocido o menospreciado por gran parte de los desarrolladores.

Este trabajo pretende resaltar la importancia de las pruebas con un enfoque de calidad de software describiendo una metodología que conlleve a la aprobación de sistemas.

En el primer capítulo se da a conocer la importancia de cómo han ido evolucionando los sistemas de información.

El segundo capítulo, se enfoca en el marco teórico, es decir la teoría que respalda el estudio y que es necesaria para la comprensión del mismo.

En el tercer capítulo, se detallan los fundamentos teóricos relacionados al Diseño de un sistema de calidad de pruebas de software para una empresa de servicios basado en el estándar ISO 9126.

En el cuarto capítulo se desarrolla el caso práctico del proyecto. Finalmente se dan a conocer las conclusiones y recomendaciones del presente trabajo.

CAPÍTULO 1

1. ANTECEDENTES

En el presente capítulo se dará a conocer la importancia de los sistemas de información para alcanzar los resultados esperados en su funcionalidad, además se presentará la categoría de los sistemas de información y la metodología del ciclo de vida desarrollo del sistema en los cuales se detallará las fases que se deben tener presente, en la mayoría de las situaciones del negocio. Estas fases están íntimamente relacionadas y son inseparables.

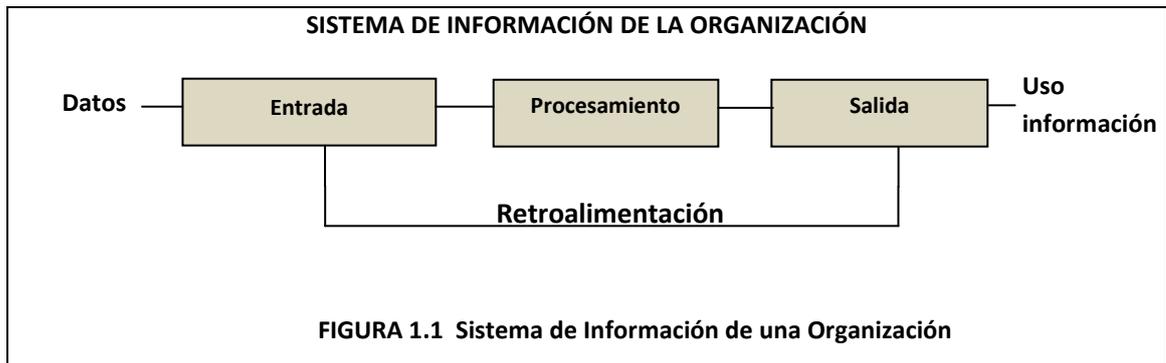
1.1. EVOLUCIÓN DE LOS SISTEMAS DE INFORMACIÓN

Durante los últimos años los sistemas de información constituyen uno de los principales ámbitos de estudio. El entorno donde las compañías desarrollan sus actividades se vuelve cada vez más complejo. La creciente globalización, el proceso de internacionalización de la empresa, el incremento de la competencia en los mercados de bienes y servicios, la rapidez en el desarrollo de las tecnologías de información, el aumento de la incertidumbre en el entorno y la reducción de los ciclos de vida de los producto originan que la información se convierta

en un elemento clave para la gestión, así como para la supervivencia y crecimiento de la organización empresarial. Si los recursos básicos analizados hasta ahora eran tierra, trabajo y capital, ahora la información aparece como otro insumo fundamental a valorar en las empresas.

A la hora de definir un sistema de información existe un amplio abanico de definiciones. Tal vez la más precisa sea la propuesta por Andreu, Ricart y Valor (1991), en la cual un sistema de información queda definido como; “conjunto formal de procesos que, operando sobre una colección de datos estructurada de acuerdo a las necesidades de la empresa, recopila, elabora y distribuyen selectivamente la información necesaria para la operación de dicha empresa y para las actividades de dirección y control correspondientes, apoyando, al menos en parte, los procesos en la toma de decisiones necesarias para desempeñar funciones de negocio de la empresa de acuerdo con su estrategia”.

Todo sistema de información utiliza como materia prima los datos, los cuales almacena, procesa y transforma para obtener como resultado final información, la cual será suministrada a los diferentes usuarios del sistema, existiendo además un proceso de retroalimentación o “feedback”, en la cual se ha de valorar si la información obtenida se adecua a lo esperado **(Ver figura 1.1)**.



En la actualidad existen muchos sistemas información para la toma de decisiones, pero ¿funcionan bien estos sistemas en realidad?

Siempre debemos preguntarnos, si van a dar los resultados esperados, pero ¿por qué no funcionan? En mi punto de vista, esto sucede por la desinformación, cuando una persona requiere un sistema, va con su programador y le informa lo que quiere obtener del sistema, sin saber cuáles son las limitaciones del software.

Un sistema de información es el depósito de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

Para que un sistema de información computarizado funcione lo mejor posible, deben existir 3 criterios importantes:

Primero es necesario que se combinen el Hardware y Software adecuados para el desempeño de la tarea requerida, es muy importante que se realicen estudios sobre los diferentes programas (Software) y los dispositivos (Hardware) que existen para elegir el que mejor convenga, esto es el, hardware necesario para que el sistema de información pueda operar. Segundo debe de haber una buena comunicación entre el usuario y el programador para especificar claramente lo que se requiere y tercero debe de existir toda la documentación del mismo, los procedimientos realizarán cada función del sistema, ya que el mal uso de éste puede ocasionar que no sea óptimo en su funcionamiento.

Los elementos de un Sistema de Información son:

1. Entrada,
2. Almacenamiento,
3. Procesamiento y
4. Salida de información.

1. Entrada de Información: Es el proceso mediante el cual, el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas.

Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Estos últimos se denominan interfaces automáticas.

Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de disco, los puertos usb, los códigos de barras, los escáner, la voz, los monitores sensibles al tacto, el teclado y el mouse, entre otras.

- 2. Almacenamiento de información:** El almacenamiento es una de las actividades o capacidades más importantes que posee un hardware, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos, los discos externos y los dispositivos usb.

- 3. Procesamiento de Información:** Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse

con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

- 4. Salida de Información:** La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, discos, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo. En este caso, también existe una interfaz automática de salida. Por ejemplo, el Sistema de Control de Clientes tiene una interfaz automática de salida con el Sistema de Contabilidad, ya que genera las pólizas contables de los movimientos procesales de los clientes.

A continuación se muestran las diferentes actividades que puede realizar un Sistema de Información de Control de Clientes:

Actividades que realiza un Sistema de Información:

Entradas:

- Datos generales del cliente: nombre, dirección, tipo de cliente, etc.
- Políticas de créditos: límite de crédito, plazo de pago, etc.
- Facturas (interface automático).
- Pagos, depuraciones, etc.

Proceso:

- Cálculo de antigüedad de saldos.
- Cálculo de intereses moratorios.
- Cálculo del saldo de un cliente.

Almacenamiento:

- Movimientos del mes (pagos, depuraciones).
- Catálogo de clientes.
- Facturas.

Salidas:

- Reporte de pagos.
- Estados de cuenta.
- Pólizas contables (interface automática)
- Consultas de saldos en pantalla de una terminal.

Existen diversas categorías de los Sistemas de Información acorde a su funcionalidad, operatividad, tipos de usuarios, entre otros PS criterios.

A continuación se muestran las categorías de los sistemas de información
(ver TABLA I)

CATEGORÍA DE LOS SISTEMAS DE INFORMACIÓN	
	CARACTERÍSTICAS
SISTEMAS PARA EL PROCESAMIENTO DE TRANSACCIONES	Sustituyen los procesamientos manuales por otros basados en computadora. Trata con procesos de rutina bien estructurados. Incluye aplicaciones para el mantenimiento de registros.
SISTEMAS DE INFORMACIÓN ADMINISTRATIVOS	Proporcionan la información que será empleada en los procesos de decisión administrativa. Trata con el soporte de situaciones de decisión bien estructuradas. Es posible anticipar los requerimientos de información más comunes.
SISTEMAS PARA EL SOPORTE DE DECISIONES	Proporcionan información a los directivos que deben tomar decisiones sobre situaciones particulares. Apoyan la toma de decisiones en circunstancias que no están bien estructuradas.

TABLA I Categoría de los Sistemas de Información

Los sistemas para ser desarrollados requieren de una metodología que permita definir los pasos necesarios para su completa ejecución.

1.2. CICLO DE VIDA DE UN SISTEMA DE INFORMACIÓN

Según James Senn, existen tres estrategias para el desarrollo de sistemas: el método clásico del ciclo de vida de desarrollo de sistemas, el método de desarrollo por análisis estructurado y el método de construcción de prototipos de sistemas. Cada una de estas estrategias tiene un uso amplio en cada una de los diversos tipos de empresas que existen, y resultan efectivas si son aplicadas de manera adecuada.

El ciclo de un sistema de información es un enfoque por fases del análisis y diseño que sostiene que los sistemas son desarrollados de mejor manera mediante el uso de un ciclo específico de actividades del analista y del usuario.

El ciclo de vida del desarrollo de sistemas consiste en las siguientes actividades:

1. Investigación preliminar
2. Determinación de requerimientos del sistema
3. Diseño de sistema
4. Desarrollo de software
5. Prueba de los sistemas
6. Implantación y Evaluación

1.3. CICLO DE VIDA CLÁSICO DEL DESARROLLO DE SISTEMAS

En esta sección se describirá la metodología del ciclo de vida, metodología vigente en todo desarrollo de sistemas.

El desarrollo de sistemas es un proceso que consiste de seis etapas, dentro de las cuales hay dos etapas principales, el análisis y el diseño de sistemas. El ciclo de vida también llamado CVDS (Ciclo de vida de desarrollo de sistemas) comienza cuando la gerencia o en algunas ocasiones el área de usuarios, se da cuenta de la necesidad de un sistema de negocio.

El ciclo de vida del desarrollo de sistemas es el conjunto de actividades de los analistas, programadores y usuarios, que necesitan llevarse a cabo para desarrollar y poner en marcha un sistema de información. Se debe tener presente que en la mayoría de las situaciones del negocio, las actividades están íntimamente relacionadas y son inseparables.

El método de ciclo de vida para el desarrollo de sistemas es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información. El método del ciclo de vida para el desarrollo de sistemas consta de 6 fases:

1). Investigación Preliminar:

¿Cuántas veces se está en situaciones en donde se pregunta si no existe una mejor manera de hacer algo? Por ejemplo, abrir una tienda departamental adicional que creará una necesidad para nuevos procedimientos de facturación, cuando un alto porcentaje de clientes utiliza la cuenta de crédito de esta compañía para compra en todas las tiendas.

Duplicar el número de clientes para agrandar las instalaciones y la introducción de muchos nuevos productos, puede traer nuevos requerimientos de pago e cuentas. Un cambio en las áreas de los gerentes departamentales puede guiarlos hacia nuevas formas para registrar las ventas, con implicaciones para el sistema de entrada de pedidos basado en computadora. Una compañía en crecimiento, puede contemplar los sistemas de información computarizados como una forma para hacer posible el crecimiento continuo, sin tener dificultades en el proceso de los pedidos de los clientes.

Aclaración de la solicitud.

Muchas solicitudes que provienen de usuarios, no se encuentran formuladas de manera clara. Es decir la solicitud del proyecto debe examinarse para determinar con precisión lo que el usuario desea.

Si el solicitante pide ayuda sin saber qué es lo que está mal o dónde se encuentra el problema, la aclaración del mismo se vuelve más difícil. En cualquier caso, antes de seguir adelante, la solicitud del sistema debe estar claramente planteada.

Estudio de Factibilidad

Un resultado importante de la investigación preliminar es la determinación de que el sistema requerido es factible. Existen tres aspectos en el estudio de factibilidad de la investigación preliminar:

- **Factibilidad técnica.** ¿Puede realizarse el trabajo para el proyecto con el equipo actual, tecnología de software y el personal disponible? Si se requiere nueva tecnología ¿qué probabilidades hay de que pueda desarrollarse?
- **Factibilidad económica.** ¿Existen suficientes beneficios en la creación del sistema para hacer que los costos sean aceptables?

O, en forma inversa, ¿son tan altos los costos como para que el proyecto no deba llevarse a cabo?

- **Factibilidad operativa.** ¿Se utilizará el sistema si se desarrolla y pone en marcha? Habrá resistencia de los usuarios.

El estudio de factibilidad se lleva a cabo con un pequeño grupo de gente, familiarizada con las técnicas de los sistemas de información, que entienden la parte de la empresa que será afectada por el proyecto y tienen los conocimientos suficientes del proceso de análisis y diseño de sistemas.

Aprobación de la solicitud.

La solicitud de creación de un sistema de información puede originarse por varias razones: sin importar cuales sean estas, el proceso se inicia siempre con la petición de una persona.

No todos los proyectos requeridos son deseables o factibles. Sin embargo, aquellos que son tanto factibles como deseables deben anotarse para tomarlos en cuenta. En algunos casos, el desarrollo puede comenzar inmediatamente, pero en la mayor parte, los miembros del departamento de sistemas están ocupados en otros proyectos que se encuentran en marcha.

Cuando esto sucede, la gerencia decide qué proyectos son más importantes y entonces los programa. Después de que se aprueba la solicitud de un proyecto, se estima su costo, la prioridad, el tiempo de terminación y los requerimientos del personal que se utilizan, para determinar de la lista existente de qué proyectos se incluirán.

Se puede iniciar una petición por muchas razones, pero la clave es que alguien, ya sea gerente, un empleado o un especialista de sistemas, inicie un requerimiento de sistema de información. Cuando se formula la solicitud comienza la primera actividad de sistemas: la investigación preliminar. Esa actividad tiene tres partes: aclaración de la solicitud, estudio de factibilidad y aprobación de la solicitud.

2). Determinación de los requerimientos del sistema.

El aspecto fundamental del análisis de sistemas es comprender todas las facetas importantes de la parte de la empresa que se encuentra bajo estudio. Los analistas, al trabajar con los empleados y administradores, deben estudiar los procesos de una empresa para dar respuesta a las siguientes preguntas clave:

¿Qué es lo que hace?

¿Cómo se hace?

¿Con que frecuencia se presenta?

¿Qué tan grande es el volumen de transacciones o decisiones?

¿Cuál es el grado de eficiencia con el que se efectúan las tareas?

¿Existe algún problema? ¿Qué tan serio es? ¿Cuál es la causa que lo origina?

3). Diseño del sistema:

El diseño de un sistema de información produce los elementos que establecen cómo el sistema cumplirá los requerimientos indicados durante el análisis de sistemas. A menudo los especialistas de sistemas se refieren a esta etapa como el diseño lógico, en contraste con el desarrollo del software de programas, que se conoce como diseño físico.

Los analistas de sistemas comienzan por identificar los informes y otras salidas que el sistema producirá. A continuación los datos específicos con éstos se señalan, incluyendo su localización exacta sobre el papel, la pantalla de despliegue u otro medio. Usualmente, los diseñadores dibujan la forma o la visualización como la esperan cuando el sistema está terminado.

El diseño del sistema también describe los datos calculados o almacenados que se introducirán. Los grupos de datos individuales y los procedimientos de cálculo se describen con detalle. Los diseñadores seleccionan las estructuras de los archivos y los dispositivos de almacenamiento, como son discos magnéticos, cintas magnéticas o incluso archivos en papel. Los

procedimientos que ellos escriben muestran cómo se van a procesar los datos y a producir la salida.

Los documentos que contienen las especificaciones de diseño utilizan diversas formas para representar los diseños, diagramas, tablas y símbolos especiales, algunos de los cuales el lector puede haber utilizado ya y otros que pudieran ser totalmente nuevos. La información del diseño detallado se pasa al grupo de programación para que pueda comenzar el desarrollo del software.

Los diseñadores son responsables de proporcionar a los programadores las especificaciones completas y escritas con claridad, que establezcan lo que debe hacer el software. Conforme comienza la programación, los diseñadores están pendientes para contestar preguntas, esclarecer ideas confusas y manejar los problemas que confronten los programadores cuando utilicen las especificaciones de diseño.

El diseño de un sistema de información produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis. Los especialistas en sistemas se refieren, con frecuencia, a esta etapa

como diseño lógico en contraste con la del desarrollo del software, a la que denominan diseño físico.

4). Desarrollo del software:

Los desarrollares del software pueden instalar o modificar; por ejemplo software comercial que se haya comprado, o pueden escribir programas nuevos diseñados a la medida. La decisión de qué se va a hacer depende del costo de cada una de las opciones, el tiempo disponible para describir el software y la disponibilidad de programadores. En forma usual, en las grandes empresas los programadores de computadoras (o la combinación de analistas-programadores) son parte del grupo profesional permanente. Las compañías más pequeñas en donde los programadores permanentes no se han contratado, pueden obtener servicios externos de programación en base a un contrato.

Los programadores también son responsables de documentar el programa e incluir los comentarios que expliquen tanto cómo y por qué se utilizó cierto procedimiento conforme se codificó de cierta forma. La documentación es esencial para probar el programa y darle mantenimiento una vez que la aplicación se ha puesto en marcha.

Los encargados de desarrollar software pueden instalar software comprobado por terceros o escribir programas diseñados a la medida del solicitante. La elección depende del costo de cada alternativa, del tiempo disponible para escribir el software y de la disponibilidad de los programadores.

Por lo general, los programadores que trabajan en las grandes organizaciones pertenecen a un grupo permanente de profesionales disponibles para diversas actividades de sistemas.

5). Pruebas de sistemas:

Durante la prueba, el sistema se utiliza en forma experimental para asegurar que el software no falle; es decir, que funcione de acuerdo a sus especificaciones y a la manera que los usuarios esperan que lo haga. Se examinan datos especiales de prueba en la entrada del procesamiento y los resultados para localizar algunos problemas inesperados. Puede permitirse también a un grupo limitado de usuarios que utilicen el sistema por un tiempo determinado, de manera que los analistas puedan captar si lo utilizan de forma no planeada. Es preferible detectar cualquier anomalía antes de que la empresa ponga en marcha el sistema y dependa de él.

En muchas compañías la prueba de programas se lleva a cabo por personas diferentes a aquellos que los escriben; es decir si se utilizan personas que no conocen cómo se diseñaron ciertas partes de los programas, se asegura una mayor y más completa prueba, además de ser imparcial, lo que da a un software más confiabilidad.

Durante la prueba de sistemas, el sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir, que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga.

Se alimentan como entradas conjunto de datos de prueba para su procesamiento y después se examinan los resultados.

Existen diversos tipos o clases de pruebas, éstos se tratarán en este documento más adelante.

6). Implantación y evaluación:

La implantación es el proceso de verificar e instalar nuevo equipo, entrenar a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizarla. Una vez instaladas, las aplicaciones se emplean durante muchos años. Sin embargo, las organizaciones y los usuarios cambian con el paso del tiempo,

incluso el ambiente es diferente con el paso de las semanas y los meses.

Por consiguiente, es indudable que debe darse mantenimiento a las aplicaciones. La evaluación de un sistema se lleva a cabo para identificar puntos débiles y fuertes. La evaluación ocurre a lo largo de cualquiera de las siguientes dimensiones:

Evaluación operacional: Valoración de la forma en que funciona el sistema, incluyendo su facilidad de uso, tiempo de respuesta, lo adecuado de los formatos de información, confiabilidad global y nivel de utilización.

Impacto organizacional: Identificación y medición de los beneficios para la organización en áreas tales como finanzas, eficiencia operacional e impacto competitivo. También se incluye el impacto sobre el flujo de información externo e interno.

Opinión de los administradores: evaluación de las actividades de directivos y administradores dentro de la organización así como de los usuarios finales.

Desempeño del desarrollo: La evaluación de proceso de desarrollo de acuerdo con criterios tales como tiempo y esfuerzo de desarrollo, concuerdan con presupuestos y estándares, y otros criterios de administración de proyectos. También se incluye la

valoración de los métodos y herramientas utilizados en el desarrollo.

1.4. MÉTODO DE DESARROLLO POR ANÁLISIS ESTRUCTURADO

Muchos especialistas en sistemas de información reconocen la dificultad de comprender de manera completa sistemas grandes y complejos. El método de desarrollo del análisis estructurado tiene como finalidad superar esta dificultad por medio de:

- 1). La división del sistema en componentes
- 2). La construcción de un modelo del sistema.

El análisis estructurado se concentra en especificar lo que se requiere que haga el sistema o la aplicación. Permite que las personas observen los elementos lógicos (lo que hará el sistema) separados de los componentes físicos (computadora, terminales, sistemas de almacenamiento, etc.). Después de esto se puede desarrollar un diseño físico eficiente para la situación donde será utilizado.

El análisis estructurado es un método para el análisis de sistemas manuales o automatizados, que conduce al desarrollo de especificaciones para sistemas nuevos o para efectuar modificaciones a los ya existentes. Éste análisis permite al

analista conocer un sistema o proceso en una forma lógica y manejable al mismo tiempo que proporciona la base para asegurar que no se omite ningún detalle pertinente.

Componentes:

- **Símbolos gráficos:** Iconos y convenciones para identificar y describir los componentes de un sistema junto con las relaciones entre estos componentes.
- **Diccionario de datos:** Descripción de todos los datos usados en el sistema. Puede ser manual o automatizado.
- **Descripciones de procesos y procedimientos:** Declaraciones formales que usan técnicas y lenguajes que permiten a los analistas describir actividades importantes que forman parte del sistema.
- **Reglas:** Estándares para describir y documentar el sistema en forma correcta y completa.

Diseño Estructurado.

El diseño Estructurado es otro elemento del Método de Desarrollo por Análisis Estructurado que emplea la descripción gráfica, se enfoca en el desarrollo de especificaciones del software.

El objetivo del Diseño Estructurado es desarrollar programas formados por módulos independientes unos .de otros desde el punto de vista funcional.

La herramienta fundamental del Diseño Estructurado es el diagrama estructurado que es de naturaleza gráfica y evitan cualquier referencia relacionada con el hardware o detalles físicos. Su finalidad no es mostrar la lógica de los programas (que es la tarea de los diagramas de flujo).

Los Diagramas Estructurados describen la interacción entre módulos independientes junto con los datos que un módulo pasa a otro cuando interacciona con él.

Análisis de flujo de datos.

Estudia el empleo de los datos para llevar a cabo procesos específicos de la empresa dentro del ámbito de una investigación de sistemas, usa los diagrama de flujos de datos y los diccionarios de datos.

Herramientas

Las herramientas muestran todas las características esenciales del sistema y la forma en que se ajustan entre sí, como es muy

difícil entender todo un proceso de la empresa en forma verbal, las herramientas ayudan a ilustrar los componentes esenciales de un sistema, junto con sus acciones.

Diagrama de flujo de datos

Es el modelo del sistema. Es la herramienta más importante y la base sobre la cual se desarrollan otros componentes.

El modelo original se detalla en diagramas de bajo nivel que muestran características adicionales del sistema. Cada proceso puede desglosarse en diagramas de flujos de datos cada vez más detallados. Repitiéndose esta secuencia hasta que se obtienen suficientes detalles para que el analista comprenda la parte del sistema que se encuentra bajo investigación.

El diagrama físico de datos da un panorama del sistema en uso, dependiente de la implantación, mostrando que tareas se hacen y cómo son hechas. Incluyen nombres de personas, nombres o números de formato y documento, nombres de departamentos, archivos maestro y de transacciones, equipo y dispositivos utilizados, ubicaciones, nombres de procedimientos.

El diagrama lógico de datos da un panorama del sistema, pero a diferencia del físico es independiente de la implantación, que se centra en el flujo de datos entre los procesos, sin considerar los

dispositivos específicos y la localización de los almacenes de datos o personas en el sistema. Sin indicarse las características físicas.

Notaciones: Son cuatro los símbolos, que fueron desarrollados y promovidos al mismo tiempo por dos enfoques: Diagrama de Yourdon y Diagrama de Gane&Sarson.

Gane-Sarson es una de las metodologías para implementar diagramas de flujo de datos, que son representaciones gráficas que muestra información acerca del funcionamiento de un sistema, y un programa que se puede utilizar es VISIO de Microsoft.

Esta metodología al igual que otras, no se sabe si pertenece a los campos de las tecnologías o las herramientas, y su mayor apogeo fue en la década de los 80. En particular Gane&Sarson al comenzar el desarrollo de un diagrama de flujo de datos, crea una lista de entidades que expliquen las entradas que llegan a cada una y las salidas que hacia ellas fluyen. Las diferencias más notorias en comparación a la metodología de Yourdon son la política de refinamiento, modelamiento del sistema actual, y relación del DFD con el modelo de datos.

Diferencias entre Metodologías de Yourdon y Game&Sarson

MÉTODO DE YOURDON	MÉTODO DE GAME % SARSON
<ul style="list-style-type: none"> • Construir el mod. Físico actual DFD físico actual • Construir mod. Lógico actual DFD lógico actual • Crear un conjunto de modelos físicos alternativos • Estimar Costos y tiempo de cada opción • Seleccionar un modelo • Empaquetar la especificación 	<ul style="list-style-type: none"> • Construir el Mod. Lógico actual DFD lógico actual • Contruir el mod. Lógico del nuevo sistema: datos en 3FN • Seleccionar un Mod. Lógico • Crear nuevo mod. Físico del sistema • Empaquetar la especificación

TABLAII Diferencias entre Metodologías de Yourdon y Game & Sarson

1.5. Método del prototipo de sistemas

La construcción de prototipos representa una estrategia de desarrollo, cuando no es posible determinar todos los requerimientos del usuario. Es por ello que incluye el desarrollo interactivo o en continua evolución, donde el usuario participa de forma directa en el proceso.

Este método contiene condiciones únicas de aplicación, en donde los encargados del desarrollo tienen poca experiencia o información, o donde los costos y riesgos de que se cometa un error pueden ser altos.

Así mismo este método resulta útil para probar la facilidad del sistema e identificar los requerimientos del usuario, evaluar el diseño de un sistema o examinar el uso de una aplicación. El método del prototipo de sistemas consta de 5 etapas:

1). Identificación de requerimientos conocidos: La determinación de los requerimientos de una aplicación es tan importante para el método de desarrollo de prototipos como lo es para el ciclo de desarrollo de sistemas o análisis estructurado. Por consiguiente, antes de crear un prototipo, los analistas y usuarios deben de trabajar juntos para identificar los requerimientos conocidos que tienen que satisfacer.

2). Desarrollo de un modelo de trabajo: Es fácil comenzar los procesos de construcción del prototipo con el desarrollo de un plan general que permita a los usuarios conocer lo que se espera de ellas y del proceso de desarrollo. Un cronograma para el inicio y el fin de la primera interacción es de gran ayuda. En el desarrollo del prototipo se preparan los siguientes componentes:

a). El lenguaje para el diálogo o conversación entre el usuario y el sistema.

b). Pantallas y formatos para la entrada de datos.

c). Módulos esenciales de procesamiento.

d). Salida del sistema

3). Utilización del prototipo: Es responsabilidad del usuario trabajar con el prototipo y evaluar sus características y operación. La experiencia del sistema bajo condiciones reales permite obtener la familiaridad indispensable para determinar los cambios o mejoras que sean necesarios, así como las características inadecuadas

4). Revisión del prototipo: Durante la evaluación los analistas de sistemas desean capturar información sobre los que les gusta y lo que les desagrada a los usuarios.

Los cambios al prototipo son planificados con los usuarios antes de llevarlos a cabo, sin embargo es el analista el responsable de tales modificaciones.

5). Repetición del proceso las veces que sea necesarias: El proceso antes descrito se repite varias veces, el proceso finaliza cuando los usuarios y analistas están de acuerdo en que el sistema ha evolucionado lo suficiente como para incluir todas las características necesarias.

Un proyecto de desarrollo de un Sistema de Información comprende varios componentes o pasos llevados a cabo durante la etapa del análisis, el cual ayuda a traducir las necesidades del

cliente en un modelo de Sistema que utiliza uno más de los componentes: Software, hardware, personas, base de datos, documentación y procedimientos.

Es por eso que existen varios modelos o métodos para la realización del análisis y diseño de un sistema, lo primero del trabajo fue revisar que es el Análisis y el diseño y posteriormente el autor Kendall, presenta varios modelos que podemos utilizar para la realización y elaboración de un proceso y trabajo exhaustivo y dar solución o respuesta al problema que se ha generado desde la perspectiva del programador y analista.

CAPITULO 2

MARCO TEÓRICO

2.1. Metodologías para desarrollo de Pruebas de sistemas

2.1.1. Introducción

A lo largo del proceso surgen una serie de dificultades que es preciso superar. En las primeras fases es necesario adquirir una serie de conocimientos y dominar determinadas técnicas relacionadas fundamentalmente con la electrónica y la informática. Además el diseñador necesita conocer exactamente el problema y especificar al máximo su funcionalidad, su alcance y las restricciones que puedan existir. Por otro lado el proyecto será realizado por un conjunto de personas, lo que origina problemas de comunicación, administración de recursos, etc. El resultado final dependerá de la conjunción de estos factores. La gestión y el desarrollo se hacen basándose en la experiencia del director de proyecto y de los componentes del mismo aplicando alguna metodología.

Las metodologías son métodos formales que permiten pasar eficazmente del problema a la definición de la solución. Estas tratan de garantizar la obtención del resultado, determinar la factibilidad de

la aplicación (tanto en coste económico y temporal como técnicamente), aumentar la productividad de los diseñadores, conseguir una mayor calidad del producto y permitir una gestión y organización eficiente del conjunto del proyecto. [20]

2.1.2. Tipos de pruebas

Hace algunos años solamente unos pocos autores hacían notar la necesidad de realizar las pruebas de sistemas, teniendo en cuenta que como todo desarrollo humano los sistemas están sujetos a errores.

Formalmente, Perry y Kaiser en 1990 [Perry y Kaiser, 1990] hicieron notar que en realidad se requerían más pruebas y no menos, como se pensaba. Después de ellos, [Berard, 1993], [Binder, 1994], [Jorgensen y Erickson, 1994], [Firesmith, tal como lo demuestran sus escritos referente al tema.

Así mismo se ha establecido, tanto teórica como práctica, que los módulos o componentes, según sea el caso, aún probados individualmente, pueden causar problemas al interactuar en una aplicación particular ([Voas et al, 1997 b], [Weyuker, 1998]). Esto puede ocasionar problemas al desarrollador, inclusive de tipo legal, ya que será éste quien presente el producto final.

Barbey, 1997) y [Orso y Silva, 1998], entre otros, han insistido en las necesidades particulares de que los sistemas sean sujetos a más pruebas.

Aun cuando falta mucho por hacer, se ha dado un buen impulso al tema como lo hace notar el propio Binder: *“Hoy, existen algunas herramientas, se han publicado muchas estrategias de prueba y se tiene mucha experiencia colectiva, pero aún quedan algunos grandes retos.”* [Binder, 1997].

La prueba del software se realiza para satisfacer una o ambas de las metas siguientes:

- a) Validar directamente el producto contra sus requerimientos establecidos;
- b) Encontrar defectos en el producto y, de esta manera indirecta, contribuir a la validación y ganar confianza en el producto.

Pruebas Alfa.

Cuando los sistemas comerciales se desarrollan con la intención explícita de distribuirlos a través de terceros para su venta, o comercialización por medio de oficinas de la propia compañía, primero deben pasar por la verificación. La retroalimentación de la fase de validación generalmente produce cambios en el software para resolver los errores y fallas que se descubren.

Pruebas Beta.

Se elige un conjunto de instalaciones usuarias que ponen a trabajar el sistema en un ambiente real. Estas instalaciones usan el sistema en las actividades cotidianas; procesan transacciones en directo y producen salidas normales del sistema.

Pruebas de Seguridad.

Cualquier sistema basado en la computadora que manipule información sensible o efectúe acciones que pueden perjudicar o beneficiar a los individuos es objeto de penetraciones impropias o ilegales.

La prueba de seguridad verifica la aplicación de los mecanismos de protección incorporados en el sistema. Durante la prueba el encargado de la prueba desempeña el papel de intruso tratando de

violar la seguridad del sistema, intentando obtener las claves de acceso por cualquier medio externo; debe bloquear el sistema, negando así el servicio a otras personas además de producir errores a propósito en el sistema; o debe curiosear los datos públicos intentando encontrar una clave de acceso al sistema.

Una prueba de seguridad debe verificar el cumplimiento de por lo menos tres principios de seguridad que son: confiabilidad, integridad y disponibilidad del sistema.

Pruebas de salida.

La mayoría de estas pruebas son un subproducto de la prueba de otros componentes estructurales.

Una prueba de salida no es otra cosa que la generación de un reporte, su entrega al usuario y determinar si satisface las necesidades de información. Una buena forma de llevar a cabo la prueba de salida consiste en entregar un reporte a una persona no familiarizada con el sistema, si esta persona puede explicar el reporte entonces probablemente el formato puede ser entendido por los usuarios.

Pruebas de la base de datos.

La prueba de la base de datos satisface las necesidades de los usuarios, en cuanto a las consultas realizadas por los usuarios y los resultados de dichas consultas o búsquedas dentro de las respectivas tablas de una Base de Datos. Se pueden llevar a cabo pruebas adicionales para asegurarse que las bases de datos contengan la información que satisfaga todas las demandas que se le plantean. Incluyendo pruebas tales como la creación de un nuevo registro antes del primer registro en el archivo maestro, la creación de un nuevo registro después del último, la creación de un registro de un departamento que no existe, intentar leer o escribir en un archivo.

Pruebas de caja negra.

El sistema de pruebas de caja negra no considera la codificación dentro de sus parámetros a evaluar, es decir, que no están basadas en el conocimiento del diseño interno del programa.

Estas pruebas se enfocan en los requerimientos establecidos y en la funcionalidad del sistema.

Pruebas de caja blanca.

Al contrario de las pruebas de caja negra, éstas se basan en el conocimiento de la lógica interna del código del sistema. Las pruebas contemplan los distintos caminos que se pueden generar gracias a las estructuras condicionales, a los distintos estados del mismo, entre otros aspectos.

Pruebas de smoke (sistemas)

Las pruebas de smoke son realizadas cuando el código de un nuevo componente o transacción está completo. Las pruebas de smoke son un conjunto de pruebas cortas, que son ejecutadas para asegurar el funcionamiento básico del producto.

Pruebas de unidad

Durante la etapa de planeación, el producto es dividido en varios componentes bien definidos que funcionen como una entidad. Las pruebas de unidad son utilizadas durante el proceso de desarrollo del producto para asegurar la calidad de cada unidad y los resultados de los mismos.

Pruebas de sistema

En estas pruebas integra el sistema en su totalidad de componentes. Es el más común en las empresas y existen variaciones previas a su liberación pruebas en el ambiente del usuario.

En cuanto a las pruebas de sistema, por su naturaleza, son similares en cualquier tipo de software, pues se prueban elementos visibles al usuario final, nivel donde no importa tanto la forma de implementación.

Pruebas de integración

Las pruebas de integración buscan probar la combinación de las distintas partes de la aplicación para determinar si funcionan correctamente en conjunto.

Las pruebas de integración toman varios componentes o transacciones y los prueba juntos para ver que tan bien actúan como unidad cohesiva. Las pruebas de integración están hechas para probar la existencia de inconsistencias en un ciclo.

Las pruebas de integración es un tipo especializado de pruebas aplicado a interfaces. El propósito de las pruebas de integración es demostrar consistencia y compatibilidad entre elementos. Estos pueden producir basura, no dar un servicio útil conocido, o ser completamente erróneos.

Pruebas de funcionalidad

Este tipo de pruebas examina si el sistema cubre sus necesidades de funcionamiento, acorde a las especificaciones de diseño. En ellas se debe verificar si el sistema lleva a cabo correctamente todas las funciones requeridas, se debe verificar la validación de los datos y se deben realizar pruebas de comportamiento ante distintos escenarios.

Estas pruebas deben estar enfocadas a tareas, a límites del sistema, a condiciones planeadas de error y de explotación. Para estas pruebas usamos los esquemas de pruebas de caja negra ya que nos interesa saber si funciona o no, independientemente de la forma en que lo haga.

Pruebas de stress (sistemas)

Las pruebas de stress tienen como propósito verificar límites bajos los cuales puede operar con normalidad el producto desarrollado. Los límites son esbozados en la documentación de usuario, especificaciones funcionales. Ciertas condiciones de stress también serán probadas (algunas con la ayuda de herramientas) por: volumen de datos a operarse y transmitirse, baja memoria, bajo espacio en disco, archivos perdidos archivos corruptos, sistemas operativos diferentes, instalaciones e redes.

Pruebas por terceros.

Una forma de lograr confiabilidad para los componentes a reutilizar es su prueba por un tercero, con características que lo hagan merecedor de confianza, por sus métodos de prueba bien definidos y el uso de estándares **(Ver figura 2.1)**.

Relación entre productos de desarrollo y niveles de prueba

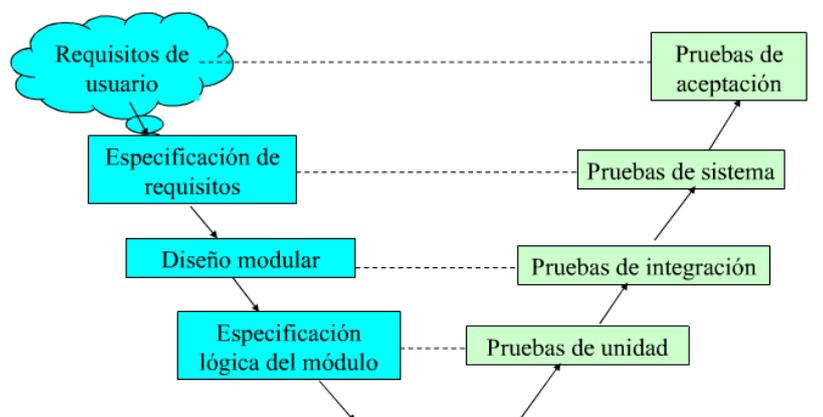


Figura 2.1. Relación entre productos de desarrollo y niveles de prueba
[30]

2.1.3. Concepto de Metodología

Según el Diccionario de la *Lengua de la Real Academia Española*, MÉTODO es el “modo de decir o hacer con orden una cosa”.

Asimismo define el diccionario la palabra METODOLOGÍA como “conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal”. Esto significa que cualquier proceso científico debe estar sujeto a una disciplina de proceso definida con anterioridad que llamaremos METODOLOGÍA.

Las Metodologías son necesarias para desarrollar cualquier proyecto que nos propongamos de manera ordenada y eficaz.

Las metodologías usadas por un profesional dicen mucho de su forma de entender su trabajo, y están directamente relacionadas con su experiencia profesional acumulada como parte del comportamiento humano de “acierto /error”.

Asimismo una metodología es necesaria para que un equipo de profesionales alcance un resultado homogéneo tal como si lo hiciera uno solo, por lo que resulta habitual el uso de metodologías en las empresas auditoras/consultoras profesionales, desarrolladas por los más expertos, para conseguir resultados homogéneos en equipos de trabajo heterogéneos.

2.1.3.1. Tipos de Metodología

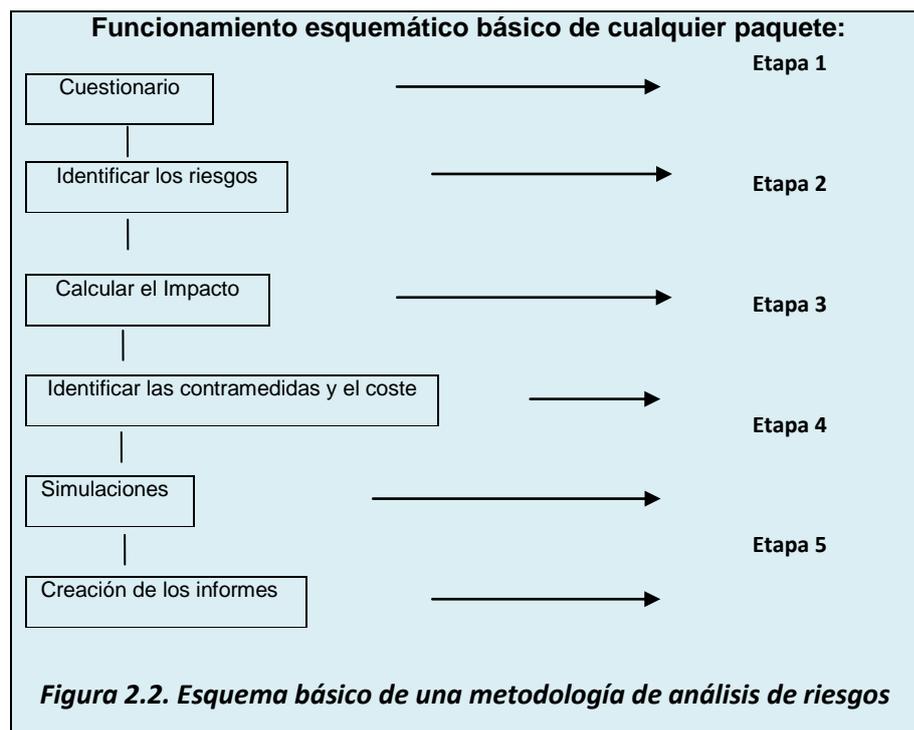
Todas las metodologías existentes desarrolladas y utilizadas en la auditoria y el control informáticos, se pueden agrupar en dos grandes familias. Éstas son:

- **Cuantitativas:** Basadas en un modelo matemático numérico que ayuda a la realización del trabajo.

- **Cualitativas:** Basadas en el criterio y raciocinio humano capaz de definir un proceso de trabajo, para seleccionar en base a la experiencia acumulada.

2.1.3.2. Metodologías de desarrollo de pruebas más comunes.

Las metodologías más comunes de evaluación de sistemas que podemos encontrar son de análisis de riesgos (**Ver figura 2.2**) o de diagnósticos de seguridad, las de plan de contingencias, y las de auditoría de controles generales.



[30]

En base a unos cuestionarios se identifican vulnerabilidades y riesgos y se evalúa el impacto para más tarde identificar las contramedidas y el coste. La siguiente etapa es la más importante, pues mediante un juego de simulación (que llamaremos “¿QUÉ. PASA SI...?”) Analizamos el efecto de las distintas contramedidas en la disminución de los riesgos analizados, eligiendo de esta manera un plan de contramedidas (plan de seguridad) que compondrá el informe final de la evaluación.

De forma genérica las metodologías existentes se diferencian en:

- Si son cuantitativas o cualitativas, o sea si utilizan un modelo matemático o algún sistema cercano a la elección subjetiva. Aunque, bien pensado, al aproximar las probabilidades por esperanzas matemáticas subjetivamente, las metodologías cuantitativas, aunque utilicen aparatos matemáticos en sus simulaciones, tienen un gran componente y además se diferencian en el propio sistema de simulación.

2.1.3.3. Metodología de Pruebas de Sistema según estándar IEEEstd. 829-1998.

Es fundamental que exista un adecuado plan de acción para evitar los posibles problemas que puedan generarse en los sistemas informáticos, telemáticos y otros relacionados con la operación del negocio.

El IEEE ha desarrollado una metodología de prueba, a la cual hacemos referencia en este trabajo, porque ha sido la base para generar el nuevo diseño.

Según esta figura, la metodología constaría de planear las pruebas para poder así diseñarla, ejecutar esperando de esa manera resultados para poder de en la siguiente fase evaluar las pruebas, si muestra error en un sistema requerirá de una depuración para así poder evaluar de nuevo el sistema y poder desarrollarlo con las acciones preventivas del sistema. **(Ver figura 2.3).**

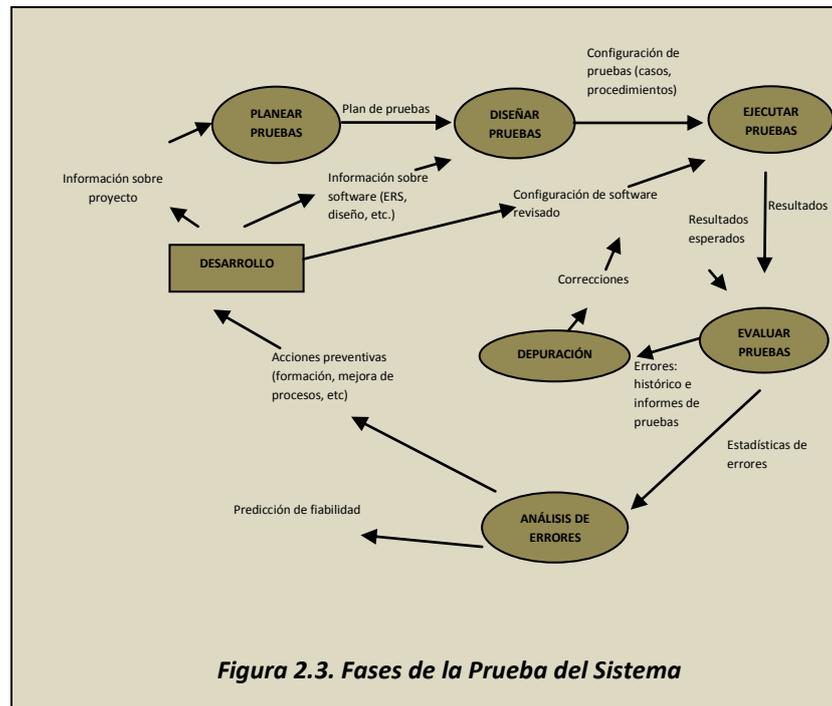


Figura 2.3. Fases de la Prueba del Sistema

[30]

Enfoque práctico para generar los casos de prueba.

Una vez definidas las fases de pruebas podemos enfocarla a un caso práctico con el objetivo de tener más claro el proceso.

— Si la especificación contiene combinaciones de condiciones de entrada, comenzar formando sus grafos de causa-efecto (ayudan a la comprensión de dichas combinaciones).

— En todos los casos, usar el análisis de valores-límites para añadir casos de prueba: elegir límites para dar valores a las causas en los casos generados asumiendo que cada causa es una clase de equivalencia.

_Utilizar la técnica de conjetura de errores para añadir nuevos casos, referidos a valores especiales.

— Ejecutar los casos generados hasta el momento (de caja negra) y analizar la cobertura obtenida (usar herramientas de análisis de cobertura)

— Examinar la lógica del programa para añadir los casos precisos (de caja blanca) para cumplir el criterio de cobertura elegido.

DOCUMENTACIÓN DEL DISEÑO DE PRUEBAS

- Documentos relacionados con el diseño de pruebas según IEEE std. 829 -1998 **(Ver figura 2.4).**

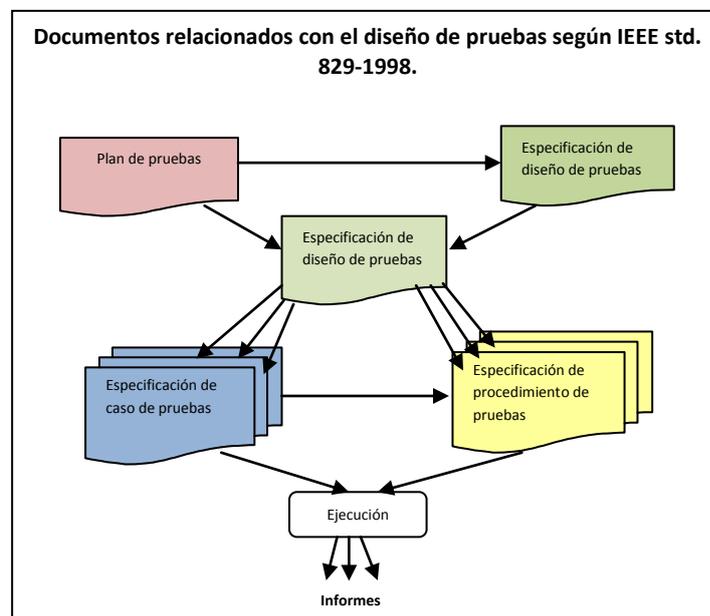


Figura 2.4. Documentación del diseño de Pruebas
[30]

PLAN DE PRUEBAS

El propósito del plan de pruebas es explicitar los siguientes puntos:

- **Señalar.**
 - El enfoque
 - Los recursos
 - El esquema de actividades de prueba
 - Los elementos a probar
 - Las características
 - Las actividades de prueba
 - El personal responsable
 - Los riesgos asociados

- **Estructura Fijada en el estándar**
 - Identificador único del documento (para la gestión de configuración).
 - Introducción y resumen de elementos y características a probar.
 - Elementos software que se van a probar (p. ej., programas o módulos).
 - Características que se van a probar.

- Características que no se prueban.
- Enfoque general de la prueba (actividades, técnicas, herramientas, etc.).
- Criterios de paso/fallo para cada elemento.
- Criterios de suspensión y requisitos de reanudación.
- Documentos a entregar (como mínimo, los descritos en el estándar).
- Actividades de preparación y ejecución de pruebas.
- Necesidades de entorno.
- Responsabilidades en la organización y realización de las pruebas.
- Necesidades de personal y de formación.
- Esquema de tiempos (con tiempos estimados, hitos, etc.).
- Riesgos asumidos por el plan y planes de contingencias para cada riesgo.
- Aprobaciones y firmas con nombre y puesto desempeñado.

Note que puede haber un plan global que explicita el énfasis a realizar sobre los distintos tipos de pruebas (verificación, e integración)

Especificación del Diseño de Pruebas.

Objetivo: Especificar los refinamientos necesarios sobre El enfoque general reflejado en el plan e identificar las características que se deben probar con este diseño de pruebas

Estructura Fijada por el estándar:

- Identificador (único) para la especificación. Proporcionar también una referencia del plan asociado (si existe).
- Características a probar de los elementos software (y combinaciones de características).
- Detalles sobre el plan de pruebas del que surge este diseño, incluyendo las técnicas de prueba específica y los métodos de análisis de resultados.
- Identificación de cada prueba:
 - Identificador.
 - Casos que se van a utilizar.
 - Procedimientos que se van a seguir.
 - Criterios de paso/fallo de la prueba (criterios para determinar si una característica o combinación de características ha pasado con éxito la prueba o no).

Especificación de un Caso de Prueba

— Objetivo; definir uno de los casos de prueba identificado por una especificación del diseño de las pruebas.

Estructura Fijada por el estándar

Identificador único de la especificación.

Elementos software (p. ej., módulos) que se van probar; definir dichos elementos y las características que ejercitará este caso.

- Especificaciones de cada entrada requerida para ejecutar el caso (incluyendo las relaciones entre las diversas en radas; p. ej., la sincronización de las mismas).

Especificaciones de todas las salidas y las características requeridas (p. ej., el tiempo de respuesta) para los elementos que se van a probar.

- Necesidades de entorno (hardware, software y otros como, por ejemplo, el personal).
- Requisitos especiales de procedimiento (o restricciones especiales en los procedimientos para ejecutar este caso).
- Dependencias entre casos (p. ej., listar los identificadores de los casos que se van a ejecutar antes de este caso de prueba).

Especificación de un Procedimiento de Prueba

— Objetivo: especificar los pasos para la ejecución de un conjunto de casos de prueba o, más generalmente, los pasos utilizados para analizar un elemento software con el propósito de evaluar un conjunto de características del mismo.

Estructura Fijada por el estándar

- Identificador único de la especificación y referencia a la correspondiente especificación de diseño de prueba.
- Objetivo del procedimiento y lista de casos que se ejecutan con él.
- Requisitos especiales para la ejecución (p. ej., entorno especial o personal especial).
- Pasos en el procedimiento. Además de la manera de registrar los resultados y los incidentes de la ejecución, se debe especificar:
 - La secuencia necesaria de acciones para preparar la ejecución.
 - Acciones necesarias para empezar la ejecución
 - Acciones necesarias durante la ejecución.
 - Cómo se realizarán las medidas (p. ej., el tiempo de respuesta).

- Acciones necesarias para suspender a prueba (cuando los acontecimientos no previstos lo obliguen).
- Puntos para reinicio de la ejecución y acciones necesarias para el reinicio en estos puntos.
- Acciones necesarias para detener ordenadamente la ejecución.
- Acciones necesarias para restaurar el entorno y dejarlo en la situación existente antes de las pruebas.

Acciones necesarias para tratar los acontecimientos anómalos

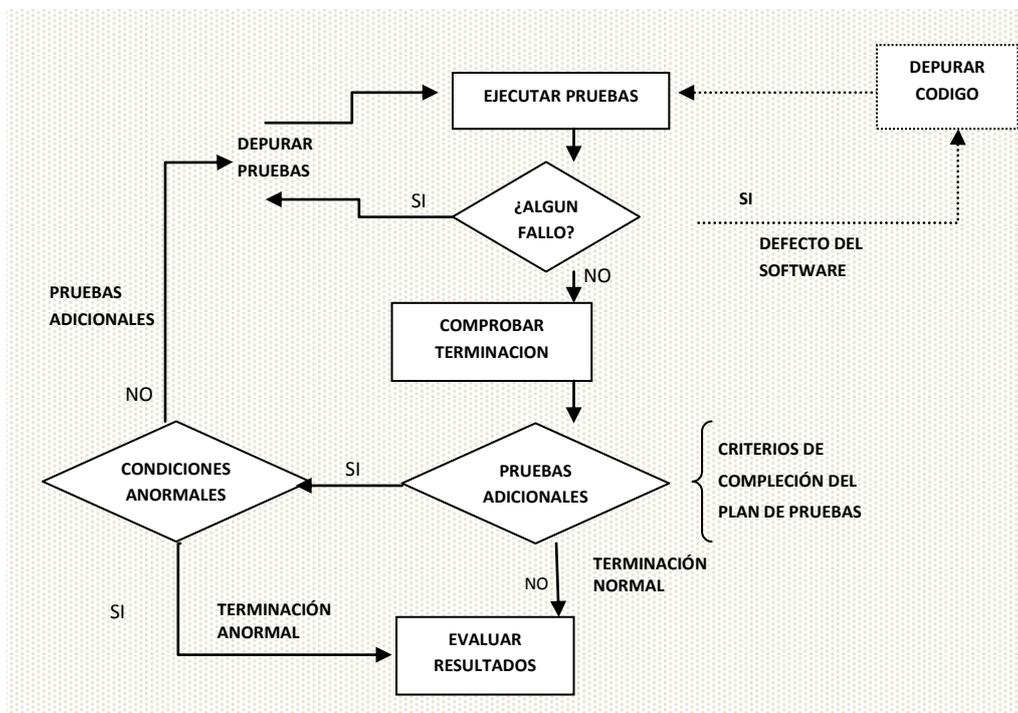


Figura 2.5. Proceso de la Ejecución de Pruebas

Documenta todos los hechos relevantes ocurridos durante la ejecución de las pruebas.

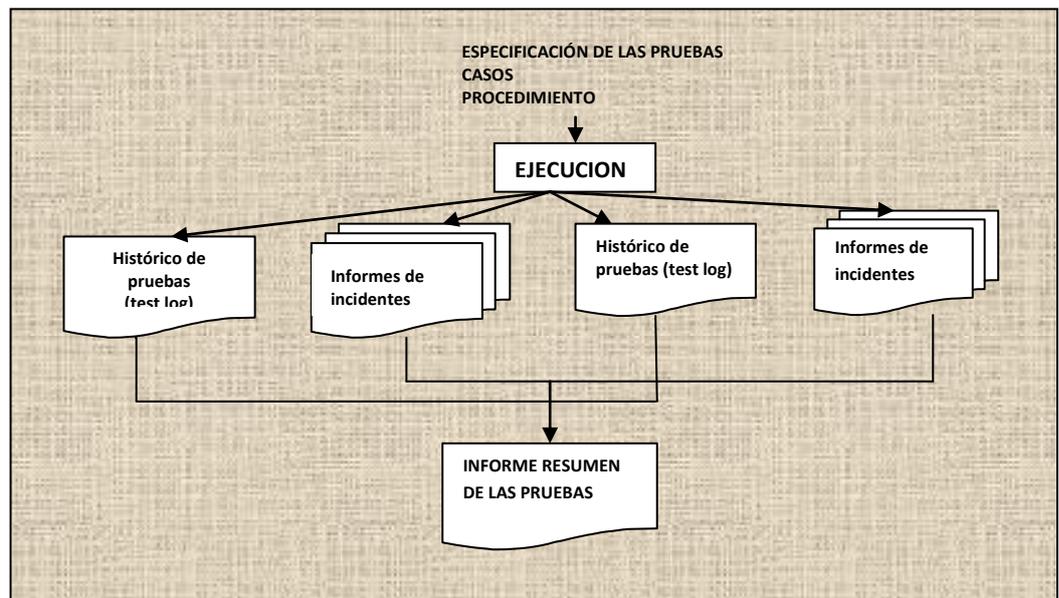
Informe de Incidente:

Documenta cada incidente (p. ej. una interrupción en las pruebas debido a un corte de electricidad, bloqueo del teclado, etc.) ocurrido en la prueba y que requiera una posterior investigación.

Resumen de Informe

Resume los resultados de las actividades de prueba (las reseñadas en el propio informe) y aporta una evaluación del software, basada en dichos resultados.

DOCUMENTACIÓN DE LA EJECUCIÓN DE LAS PRUEBAS



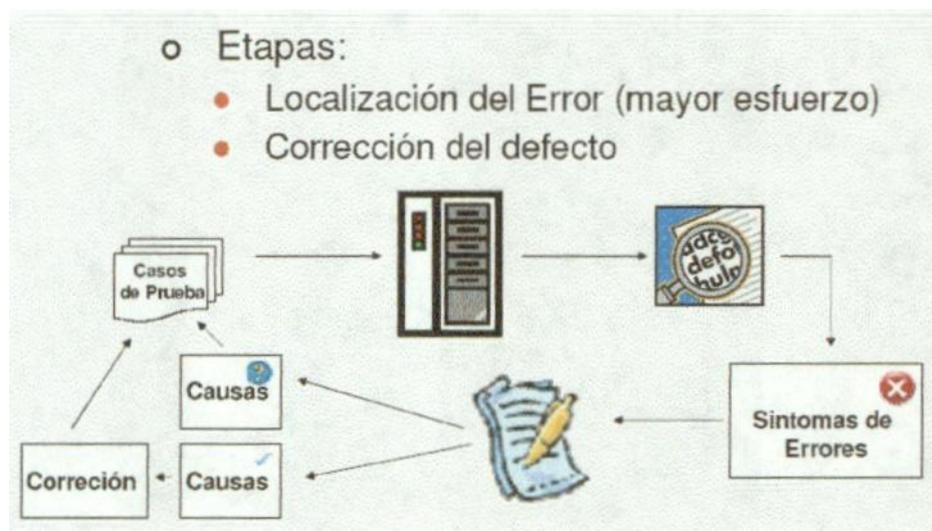
DEPURACION

— Es el proceso de localizar, analizar y corregir los defectos que se sospecha que contiene el software. Suele ser la consecuencia de una prueba con éxito.

— Consecuencias

Encontrar la causa del *error analizarla y corregirla*

No encontrar la causa. (*Casos de prueba para depuración*).



EJECUCION

ANÁLISIS DE ERRORES O ANÁLISIS CAUSAL

— ¿Cuándo se cometió?

— ¿Quién lo hizo

— ¿Qué se hizo mal?

— ¿Cómo se podría haber prevenido?

— ¿Por qué no se detectó antes?

— ¿Cómo se podría haber detectado antes?

2.2. Calidad de Software

Primero hay que aclarar que el software es inmaterial y por lo tanto la calidad de software es intangible, pero a pesar de esto se tienen ciertas pautas para determinar la calidad de un determinado software; entre estas pautas tenemos:

- Que el software en cuestión, se acerque a tener cero defectos.
- Que se haya cumplido con todos los requisitos intrínsecos y expresos.
- Que se logre alcanzar la satisfacción del cliente.

Aunque la calidad es un objetivo importante para cualquier producto, no debemos olvidar que los productos, y también los productos software, se construyen para ser utilizados. Por tanto, el principal objetivo de un producto es satisfacer una necesidad (o varias) de un usuario y, por consiguiente, ofrecer al usuario algún beneficio por su utilización. Es decir, la calidad no es el objetivo último del producto, sino satisfacer las necesidades de un cliente. También es importante señalar que esto implica que la calidad de un producto software no se puede referir únicamente a obtener un producto sin errores.

A la hora de abordar la calidad en productos de software, hay que tener en cuenta un conjunto de características que hace que el software sea un producto específico:

- El software se desarrolla, no se fabrica en el sentido clásico del mismo.
- Se trata de un producto lógico, sin existencia física.
- No se degrada con el uso.
- Por la complejidad del software y la ausencia de controles adecuados, se suele entregar el software conscientemente con defectos (incluso públicamente declarados).
- Un gran porcentaje de la producción se hace aún a medida en vez de emplear componentes existentes y ensamblar.
- Es muy flexible. Se puede cambiar con facilidad e incluso reutilizar fragmentos.

2.2.1. Concepto de calidad de Software

Basándonos en lo antes mencionado se puede decir que, la calidad de software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

La calidad del software es medible y varía de un sistema a otro o de un programa a otro. Un software elaborado para el control de naves espaciales debe ser confiable al nivel de "cero fallas"; un software

hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad; mientras que un producto de software a ser utilizado durante un largo período (10 años o más), necesita ser confiable, mantenerle y flexible para disminuir los costos de mantenimiento y perfeccionamiento durante el tiempo de explotación.

La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software, esto es la calidad se debe realizar antes, durante y después de elaborar el producto.

Otros autores definen a la calidad de software como la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente.

A partir de esta definición vale la pena señalar que la calidad no es solo afectada por el incumplimiento de los requisitos del cliente y los explícitamente definidos por la ingeniería de software, sino que los

requisitos implícitos también deben ser considerados. Téngase en cuenta que pocas veces el cliente está en condiciones reales de explicitar todo lo que se puede esperar del producto, muchas veces por desconocimiento y otras por asumir de forma tácita muchas funcionalidades.

Hay que aclarar que los requisitos establecidos explícitamente se reflejan en el documento de especificación de requisitos del sistema, mientras que los requisitos implícitos no aparecen en el documento de especificación de requisitos del sistema. Si se cumplen los explícitos y no los implícitos, la calidad del software queda entre dicho.

2.2.2. Sistema de calidad

El trabajo para la mejora de la calidad tiene distintos ámbitos de actuación:

- Nivel individual.
- Nivel de empresa u organización.
- Nivel de proyecto. **[4]**

La gestión de la calidad a nivel de empresa u organización consiste en la creación de una estructura organizativa apropiada para fomentar el trabajo por la calidad de todas las personas y

departamentos de la empresa. Se suele recurrir al concepto de sistema de calidad. Para la implementación de una infraestructura de calidad es necesario el apoyo de un sistema de calidad que se adecúe a los objetivos de calidad de la empresa. Hay que tener en cuenta que para ser útil, un sistema de calidad debe:

- Ser eficaz, comprendido por todos.
- Ofrecer confianza en satisfacer las necesidades de los clientes.
- Poner énfasis en prevenir en lugar de detectar.

Un sistema de calidad consta de dos partes:

1. Documentación: en la que se describe el sistema, procedimientos, etc. ajustándose a una norma:

a. Manual de calidad: Descripción del sistema que sirve de referencia permanente en la aplicación del sistema. Los elementos, requisitos y los medios que adopte la empresa para su sistema de calidad se deben establecer por escrito, ordenadamente, en forma de políticas y procedimientos. También se debe describir el sistema de gestión de calidad para servir como referencia al implantar el sistema.

b. Procedimientos de calidad: Instrucciones específicas para ciertas actividades o procesos. Para que el manual sea más manejable, puede completarse con procedimientos o

instrucciones específicas para ciertas actividades o procesos.

Cada empresa puede tener sus propios procedimientos.

- c. Registros de datos sobre calidad:** Pretenden almacenar datos sobre las actividades relacionadas con la calidad o sobre la evaluación de los productos. **[4]**

2. Parte práctica: Que tiene dos vertientes:

- a.** Aspectos físicos (locales, herramientas, ordenadores,...)
- b.** Aspectos humanos: formación del personal a todos los niveles y creación y coordinación de equipos de trabajo. **[4]**

El desarrollo del software se suele organizar en proyectos. En cada proyecto de desarrollo se deben aplicar las directrices de calidad fijadas a nivel de la organización. Para ello es imprescindible la adaptación de las mismas a las condiciones de cada proyecto. Las directrices contenidas en el sistema de calidad deben adecuarse a cada uno de los proyectos. Para adaptar las directrices marcadas por los sistemas de calidad a cada proyecto particular, hay que generar un plan específico de calidad: Plan de aseguramiento de la calidad. El plan de aseguramiento debe contener:

- Objetivos de calidad del proyecto y enfoque para su consecución.
- Documentación referenciada en el plan.
- Gestión del aseguramiento de la calidad.
- Documentación de desarrollo y de control o gestión.
- Estándares, normas y prácticas que hay que cumplir.
- Actividades de revisión y auditorías.
- Gestión de la configuración del software.
- Informes de problemas.
- Herramientas, técnicas y métodos de apoyo.
- Control del código, de los equipos y de los suministradores.
- Recogida, mantenimiento y almacenamiento de datos sobre la documentación. de las actividades de aseguramiento de la calidad realizadas. [4]

Ahora que sabemos todo lo que involucra la calidad, hay que conocer que el aseguramiento de la calidad aborda principalmente tres áreas o técnicas:

- a. Métricas del software:** para el control del proyecto.
- b. Verificación y validación:** a lo largo del ciclo de vida del software, incluyendo pruebas y procesos de revisión.
- c. Gestión de la configuración del software.** [4]

La especificación de la calidad del software debe ser más detallada y exacta, y el camino para ello es la formalización de la calidad mediante un modelo de calidad que, detallaremos más adelante y define las características de un producto que influyen a la hora de medir su calidad.

2.2.3 Tipos de Errores

Los errores de software son resultados de la imprecisión o inhabilidad de comunicar perfectamente en las especificaciones lo que se piensa o se desea de la aplicación.

Existen errores privados y públicos: Los errores privados son aquellos que se muestran cuando la aplicación no ha sido liberada, y los públicos son aquellos que se presentan cuando la aplicación está en producción.

2.2.3.1. Manejo del Proceso para Garantizar la Calidad.

El aseguramiento de la calidad es la revisión de los productos y documentación relacionada con el software para verificar su cobertura, corrección, confiabilidad y facilidad de mantenimiento. Y, por supuesto, incluye la garantía de que un sistema cumple las

especificaciones y los requerimientos para su uso y desempeño deseados. **[30]**

2.2.3.2. Niveles de seguridad.

Los analistas usan cuatro niveles de aseguramiento de la calidad: prueba, verificación, validación y certificación.

La prueba del sistema es un proceso caro pero crítico que puede llevarse hasta el 50% del presupuesto para el desarrollo del programa. El punto de vista común respecto a las pruebas compartido por los usuarios, es que se lleva a cabo para demostrar que no hay errores en un programa. Sin embargo, esto es prácticamente imposible, puesto que los analistas no pueden demostrar que el software está limpio de errores.

Por lo tanto, el enfoque más útil y práctico es en el entendimiento de que la prueba es el proceso de ejecutar un programa con la intención explícita de hallar errores, es decir, hacer que el programa falle.

El examinador, que puede ser un analista, programador, o especialista entrenado en la prueba de software, está tratando realmente de hacer que el programa falle. Así, una prueba exitosa es aquella que encuentra un error.

Verificación y Validación.

Como la prueba, la *verificación* tiene la intención de hallar defectos y errores. Se lleva a cabo ejecutando un programa en un ambiente simulado. La *validación* se refiere al proceso del uso del software en un ambiente no simulado para hallar sus errores.

Un defecto de software, es el resultado de un fallo o deficiencia durante el proceso de creación de programas de ordenador o computadora (software). Dicho fallo puede presentarse en cualquiera de las etapas del ciclo de vida del software aunque los más evidentes se dan en la etapa de desarrollo y programación. Los errores pueden suceder en cualquier etapa de la creación de software.

2.2.4 Métricas

2.2.4.1 Concepto de métricas

Para controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios involucrados dentro de un proceso de medición. Es en esto donde son tan importantes las métricas, porque son éstas las que proporcionan una medida en términos de cantidades, o en otras palabras una medida

cuantitativa acerca de un sistema, o de algún componente o proceso de dicho sistema.

En términos generales, para la evaluación de la calidad, es más habitual centrarse en medidas del producto que en medidas del proceso. Una métrica es una asignación de un valor a un atributo (tiempo, complejidad, etc.) de una entidad software, ya sea un producto (código) o un proceso (pruebas).[9]

Las cualidades para medir la calidad del software son definidas por innumerables autores, los cuales las denominan y agrupan de formas diferentes. Otros autores identifican la calidad con el nivel de complejidad del software y definen dos categorías de métricas: de complejidad de programa o código, y de complejidad de sistema o estructura.

Todos los autores coinciden en que el software posee determinados índices medibles que son las bases para la calidad, el control y el perfeccionamiento de la productividad. Una vez seleccionados los índices de calidad, se debe establecer el proceso de control, que requiere los siguientes pasos:

- Definir el software que va a ser controlado: Clasificación por tipo, esfera de aplicación, complejidad, etc., de acuerdo con los estándares establecidos para el desarrollo del software.
- Seleccionar una medida que pueda ser aplicada al objeto de control. Para cada clase de software es necesario definir los indicadores y sus magnitudes.
- Crear o determinar los métodos de valoración de los indicadores: métodos manuales como cuestionarios o encuestas estándares para la medición de criterios periciales y herramientas automatizadas para medir los criterios de cálculo.
- Definir las regulaciones organizativas para realizar el control: quiénes participan en el control de la calidad, cuándo se realiza, qué documentos deben ser revisados y elaborados, etc.

Basado en lo antes mencionado podemos concluir que si se quiere obtener calidad, entonces no se puede dejar de lado la medición, es por este motivo que en el campo de la calidad de software, la definición de métricas es esencial si se desea entregar un producto de calidad acorde a la necesidades de los clientes y sobretodo un producto con alta grado de fiabilidad. Además también se puede establecer que no existe una métrica estándar, es decir que cada

métrica provendrá de un punto de vista diferente, es por esta razón que existen diferentes tipos de métricas relacionados con todos los aspectos de la calidad de software.

2.2.4.2 Clasificación de métricas.

Las métricas pueden clasificarse de la siguiente manera:

- **Métricas de producto.**

Este tipo de métricas se enfocan a los aspectos presentes en el producto final y son aplicables una vez que se ha finalizado el proceso de desarrollo del producto en cuestión.

- **Métricas de proceso.**

Son aquellas que van encaminadas a la medición de los procesos, y nos proporcionan resultados que mejoraran la calidad de los procesos para que nos conduzcan a un mejor resultado.

- **Métricas basadas en atributos internos del producto:**

- Medidas de estructuración de un programa.
- Métricas de complejidad.
- Métricas de cobertura de pruebas.
- Métricas de calidad del diseño.

- **Métricas basadas en atributos externos del producto:**

- Métricas de portabilidad.

- Métricas de defectos.
- Métricas de usabilidad.
- Métricas de mantenibilidad.
- Métricas de fiabilidad.
- **Métricas basadas en código fuente:**
 - Número de líneas de código.
 - Número de líneas de comentario.
 - Número de instrucciones.
 - Densidad de documentación.
- **Métricas basadas en estructura de diseño:**
 - Relacionadas con el control intramodular.
 - Relacionadas con el acoplamiento entre clases.
- **Métricas para sistemas orientados a objetos:**
 - Acoplamiento.
 - Herencia.
 - Cohesión.[9]

2.2.4.3. Ejemplo de métricas

En este punto hemos querido presentar un ejemplo concreto de métricas, que sirva para tener una referencia de hacia dónde deben ir orientadas las métricas, o dicho de otra manera hacia qué aspectos se debe enfocar la medición.

Un ejemplo de métricas son las que McCall definió en su modelo: McCall propone para las métricas asociadas al software un nivel de evaluación entre cero (0) y diez (10) como medidas. Además, aclara que las métricas y la evaluación son procesos subjetivos. Los elementos que se pueden tener en cuenta para la evaluación son:

- **Auto documentación**– Que el archivo ejecutable entregue documentación significativa.
- **Complejidad** – Se han implementado las funciones requeridas.
- **Concisión** – Compacto en líneas de código.
- **Consistencia** – Uso de métodos de diseño, técnicas de documentación a través del desarrollo.
- **Eficiencia en la ejecución** – Medida del tiempo de ejecución.
- **Estandarización de los datos** – Manejar tipos abstractos de datos (TAD) a través del programa.
- **Exactitud** – Preciso en cálculos y control.
- **Facilidad de auditoría** – Comprobar la conformidad con los estándares.
- **Facilidad de expansión** – Facilidad de ampliar diseños arquitectónicos, de datos, o procedimiento.

- **Facilidad de operación** – Facilitar la explotación real de la aplicación, dedicando especial atención durante el diseño, codificación y pruebas del sistema.
- **Facilidad de traza** – Realizar ingeniería en reversa. Que tan fácil es devolverme a los requerimientos.
- **Formación** – Debe poseer un buen sistema de ayudas para que los nuevos usuarios apliquen el sistema.
- **Generalidad** – Amplitud de aplicación potencial de los componentes del programa. Es decir, los módulos creados pueden ser útiles en otras aplicaciones del mismo tipo, o aplicaciones que manejen tipos de datos semejantes.
- **Independencia del hardware** – Que los diseños sean independientes de la máquina o máquinas que se tienen destinadas para el software. A calidad, pero no a implantación
- **Independencia del sistema software** – Hasta dónde el programa es independiente de la plataforma de desarrollo.
- **Instrumentación** – En qué grado el programa muestra funcionamiento e identifica errores.
- **Modularidad** – División del programa en componentes funcionales. Acoplamiento, cohesión.

- **Normalización de las comunicaciones** – Que tanto se usan estándares, interfaces, protocolos, entre otros elementos que pueden ser de importancia.
- **Seguridad** – Seguridad de la información comúnmente conocida, la cual incluye temas como: la protección de los sistemas de información en contra del acceso no autorizado y la modificación de información.
- **Simplicidad** – El sistema de información debe ser fácil de entender.
- **Tolerancia de errores** – Que tanto se pierde al ocurrir un daño grave. [10]

2.3. Definiciones Conceptuales

Arquitectura de información. Conjunto de conceptos que definen cuál será el estilo que una organización va a dar a sus sistemas informáticos, afectando a temas como la arquitectura de red, el modo de proceso, y la distribución de capacidad de almacenamiento.

Base de datos. Conjunto de datos no redundantes, almacenados en un soporte informático, organizados de forma independiente de su utilización y accesibles simultáneamente por distintos usuarios y aplicaciones.

Diccionario de datos. Descripción lógica de los dato para el usuario. Reúne la información sobre los datos almacenados en la BD(descripciones, significado, estructuras, consideraciones de seguridad, edición y uso de las aplicaciones, etc.).

Diccionario de seguridad. Base de datos que guarda los usuarios de un sistema, los recursos disponibles y los derechos de acceso de cada usuario a esos recursos.

Diseño del sistema. Fase de la metodología de desarrollo de sistemas, inmediatamente posterior a la de análisis, en la que se definen todas las estructuras de programación y sus interrelaciones junto con los detalles técnicos de construcción que sean necesarios.

Dispositivo de entrada. Elemento físico que permite al usuario introducir datos en el sistema.

Dispositivo de salida. Elemento físico que permite al usuario recibirlos datos que el sistema ha procesado.

Formulario. Soporte externo (físico) de la estructura de la información a grabar. Normalmente los formularios existen como documentos pre impresos para facilitar la recogida de datos.

Integridad. Condición de seguridad que garantiza que la información es modificada, incluyendo su creación y borrado, sólo por el personal autorizado.

Modelo de datos. Un conjunto de reglas que especifican a alto nivel la estructura de la base de datos. Según el modelo utilizado, los mismos datos se almacenarán de forma muy distinta.

Planeación: Consiste en definir los objetivos, trazar políticas y determinar la naturaleza, el alcance, la extensión y el calendario de aplicación de los procedimientos y de las comprobaciones necesarias para el logro de los objetivos.

Plan de Sistemas. Planificación de los sistemas de información de una organización o empresa cuya finalidad es asegurar la adecuación entre los objetivos estratégicos de la misma y la información necesaria para soportar dichos objetivos. Conjunto de planes que especifican a largo plazo la actividad informática de una organización.

Política de seguridad. Conjunto de principios y reglas, propias del organismo, que declaran cómo se especificará y gestionará la

protección de los activos de información de una manera consistente y segura.

Procedimientos de auditoría: Comprobaciones, instrucciones y detalles incluidos en el programa de auditoría que han de llevarse a cabo sistemática y razonablemente.

Programas de Seguridad. Estudios técnicos utilizados para apoyar los planes de seguridad: programa de clasificación de datos, programa de asignación de riesgos.

Protocolos de pruebas. Conjunto de valores de prueba, condiciones y normas para efectuar las pruebas de un programa.

Prueba. Comprobación de cada uno de los módulos construidos en la fase de desarrollo e integración de los módulos en una única estructura de programa, que luego se prueba como un todo para garantizar que se comporta de la forma que ha sido diseñada.

Pruebas de Auditoría: Información que constituye la base en que se sustentan las opiniones, conclusiones o informes del auditor.

Pruebas Adecuadas: Información que, cuantitativamente es suficiente y apropiada para lograr los resultados de la auditoría; y que, cualitativamente, tiene la imparcialidad necesaria para inspirar confianza y fiabilidad.

Pruebas Pertinentes: Información que es pertinente en relación con los objetivos de la auditoría.

Pruebas Razonables: Información que es económica en sentido de que el costo de reunirlos guarda razonable proporción con el resultado que el auditor trata de lograr.

Unicidad. Funcionalidad de un sistema de seguridad para asegurar que la información utilizada será la última, exacta, autorizada y completa.

Unidad de entrada-salida. Elemento responsable de comunicar a la estación de trabajo con los periféricos exteriores.

Unidad de datos de usuario. Conjunto de datos parte de un fichero virtual que está asociada con un nodo del árbol de nodos que modela ese fichero virtual.

CAPITULO 3

NORMAS Y/O ESTÁNDARES INTERNACIONALES

3.1. Estándares de calidad de Software

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

La gestión de la calidad se puede entender como el conjunto de actividades y medios necesarios para definir e implementar un sistema de la calidad, por una parte y responsabilizarse de su control, aseguramiento y mejora continua, por otra.[4]

El control está dirigido al cumplimiento de requisitos, el aseguramiento a inspirar confianza en que se cumplirá el requisito pertinente y el mejoramiento al aumento de su eficiencia y eficacia.

La gestión de calidad a nivel de la organización en entidades de software ha seguido dos líneas que pueden perfectamente complementarse entre sí. Por una parte, se ha seguido la línea marcada por las entidades internacionales de estandarización para todos los productos y servicios a través de las normas ISO 9000 y por otra, el mundo del software ha creado su propia línea de trabajo en la gestión de la calidad, trabajando sobre los procesos de producción de software como medio de asegurar la calidad del producto final.

3.1.1. ISO 9000 - Sistemas de Gestión de la Calidad – Fundamentos y vocabulario

Con el objetivo de estandarizar los sistemas de calidad de las diferentes empresas y sectores, se publican las normas ISO 9000, que son un conjunto de normas editadas y revisadas periódicamente por la Organización Internacional de Normalización (ISO) sobre la garantía de calidad de los procesos.

En la actualidad las normas ISO son requeridas a nivel mundial, debido a que garantizan la calidad de un producto por la utilización de controles exhaustivos, que conllevan el aseguramiento de que todos los procesos

que se involucraron dentro de la fabricación del producto operan bajo las características previstas.

La serie de normas de ISO 9000 consta de requisitos y directrices que permiten establecer y mantener un sistema de calidad en la compañía. En lugar de dictar especificaciones para el producto final, ISO 9000 se centra en los procesos sustantivos, es decir, en la forma en que se produce. Las normas ISO 9000 requieren de sistemas documentados que permitan controlar los procesos que se utilizan para desarrollar y fabricar los productos. Estos tipos de normas se fundamentan en la idea de que hay ciertos elementos que todo sistema de calidad debe tener bajo control, con el fin de garantizar que los productos y servicios de calidad se fabriquen en forma consistente y a tiempo. Cabe recalcar que ISO 9000 no normaliza el sistema de gestión de calidad, ya que esto depende del tipo de sector, tamaño de la empresa, organización interna, etc, sino que normaliza las verificaciones que se han de realizar sobre el sistema de calidad.**[5]**

La serie ISO 9000 fue creada por comités integrados por representantes de 27 países, los cuales a su vez se encargan de revisarlas y mantenerlas actualizadas. Ha sido adoptada por más de 70 países alrededor del mundo como la norma de mayor aceptación que establece

requisitos para los sistemas de calidad. De esta manera, se consolida a nivel internacional la normativa de la gestión y control de calidad.**[5]**

A continuación se presentan las ventajas de aplicar el estándar ISO 9000 dentro de una organización:

- Es un factor competitivo para las empresas.
- Proporciona confianza a los clientes.
- Ahorra tiempo y dinero, evitando re-certificar la calidad según los estándares locales o particulares de una empresa.
- Se ha adaptado a más de 90 países e implantado a todo tipo de organizaciones industriales y de servicios, tanto sector privado como público.
- Proporciona una cierta garantía de que las cosas se hacen tal como se han dicho que se han de hacer.**[5]**

A continuación se presentan las desventajas de aplicar el estándar ISO 9000 dentro de una organización:

- Es costoso.
- Muchas veces se hace por obligación.
- Es cuestión de tiempo que deje de ser un factor competitivo.
- Hay diferencias de interpretación de las cláusulas del estándar.

- No es indicativa de la calidad de los productos, procesos o servicio.
- Hay mucha publicidad engañosa.[5]

3.1.2. ISO 9001- Sistemas de Gestión de la Calidad – Requisitos

La norma ISO 9001 elaborada por la Organización Internacional para la Estandarización, especifica los requisitos para un sistema de gestión de la calidad que pueden utilizarse para su aplicación interna por las organizaciones, para certificación o con fines contractuales. Se centra en la eficacia de la gestión de la calidad para dar cumplimiento a los requisitos del cliente. Esta norma internacional no incluye requisitos específicos de otros sistemas de gestión, tales como aquellos particulares para la gestión ambiental, gestión de la seguridad y salud ocupacional, gestión financiera o gestión de riesgos. Sin embargo esta Norma internacional permite a una organización integrar o alinear su propio sistema de gestión de la calidad con requisitos de sistemas de gestión relacionados.[6]

Este estándar ha sido adoptado por más de 130 países para su uso, y se está convirtiendo en el medio principal con el que los clientes pueden juzgar la competencia de un desarrollador de software. Uno de los problemas con el estándar ISO 9001 está en que no es

específico de la industria: está expresado en términos generales, y puede ser interpretado por los desarrolladores de diversos productos.[6]

Esta norma internacional no incluye requisitos específicos de otros sistemas de gestión, tales como aquellos particulares para la gestión ambiental, gestión de la seguridad y salud ocupacional, gestión financiera o gestión de riesgos. Sin embargo esta norma internacional permite a una organización integrar o alinear su propio sistema de gestión de la calidad con requisitos de sistemas de gestión relacionados.[6]

Otra novedad que presenta es el concepto de mejora continua. Se insiste en que el sistema de gestión de la calidad tiene que ser algo dinámico que se va enriqueciendo continuamente alimentado por la satisfacción/insatisfacción de los clientes y por sus diferentes demandas a lo largo del tiempo. Ya no habrá sitio para sistemas de gestión estáticos que, aun hoy en día, abundan.[6]

La importancia de esta norma de calidad radica en que todos sus requisitos pueden ser aplicados a cualquier compañía sin importar el tamaño o el sector en el cual están inmersas.

Este estándar tiene aplicación en aquellas compañías que diseñan, fabrican y dan servicios sobre sus productos. Consta de 20 "cláusulas", cada una de las cuales establecen los requisitos para las diferentes áreas de su sistema de calidad:

1. Responsabilidad de la Dirección

La dirección es la principal responsable de una organización. La dirección de la organización debe revisar en forma regular los resultados del sistema de calidad.

2. Sistema de Calidad

La dirección deberá definir y documentar su política y objetivos de calidad para asegurar el compromiso con la calidad y con los requerimientos mínimos de ISO 9000. Es necesario tener un manual que incorpore la norma ISO 9000 y así mismo haga referencia a los procedimientos que se emplean para cumplir con la norma.

3. Revisión del contrato

Es preciso contar con un sistema documentado que define como se comunicarán y ejecutarán los cambios al cliente y a la propia organización interna.

4. Control de diseño

Es preciso tener procedimientos documentados que se aseguren que los diseños de los productos cumplen con los requerimientos de los clientes.

5. Control de los documentos y de los datos

Todos los documentos y datos requerirán de la aprobación de una persona autorizada. Es necesario autorizar de manera formal a tales personas y que estas deberán ser capaces de evaluar la validez del documento.

6. Compras

Llevar a cabo las operaciones de compra de forma sistemática que asegure que se obtienen los materiales apropiados para los requerimientos específicos de la organización.

7. Control de los productos suministrados por los clientes

Se deberán establecer procedimientos para la inspección, almacenamiento, manejo y mantenimiento de los materiales que el cliente proporciona.

8. Identificación y rastreabilidad de los productos

La evaluación de un proveedor deberá incluir un método de revisión documentado y formal, la organización deberá mantener los registros de evaluación de un proveedor y un listado formal de aquellos que satisfacen este proceso documentado. La evaluación deberá especificar la calidad de los materiales que se reciben.

9. Control de los procesos

Se refiere al proceso global de producir un artículo y el método por el cual se controla y asegura que se siguen los procesos. El equipo y herramientas que utilicen los empleados deberán contar con las instrucciones de operación y planes de mantenimiento apropiados.

10. Inspección y ensayos

Abarca las pruebas de los materiales que se desplazan por los procesos, así como la inspección final del producto. Las operaciones de prueba deberán realizarse de acuerdo con los procedimientos documentados y apoyarse con registros que indiquen el estado del material y la condición satisfactoria de todos los requerimientos antes del lanzamiento del producto.

11. Control de los equipos de inspección, medición y ensayo

Es preciso asegurar el mantenimiento, revisión y control de todos los equipo de prueba, calibración y cualquier otro, incluyendo moldes,

accesorios, plantillas, patones y programas de computación. Se deberán cumplir los puntos: Identificar la medición a realizar, identificar y calibrar todos los equipos de pruebas a intervalos regulares de tiempo o uso.

12. Estado de inspección y ensayo

A medida que los productos recorren las diversas áreas de prueba, el material y los productos deberán portar la identificación referente a su estado.

13. Control de los productos no conformes

Productos que no están de acuerdo a lo definido inicialmente deben desecharse.

14. Acciones correctivas y preventivas

La norma pide que las personas involucradas enfrenten los problemas de manera sistemática.

15. Manipulación, almacenamiento, embalaje, preservación y entrega

La norma exige revisar los pedidos de los clientes antes de aceptarlos. La norma dicta que es preferible un pedido por escrito.

Independientemente de la revisión de un período de cliente por parte de una persona autorizada, es preciso mantener un registro del pedido y de su revisión. La norma exige realizar una inspección y una prueba completa del producto final, deberán verificar que los datos estén conformes con las especificaciones del producto según las define el plan de calidad. También se exige retener el producto y posponer el envío de este hasta haber concluido todas las inspecciones y verificar que el producto cumple con todas las especificaciones. El registro deberá indicar quien autorizó el envío del producto.

16. Auditorías internas de la calidad

La dirección deberá mantener una verificación interna para el propósito primario de realizar una auditoría interna. El personal de la auditoría deberá contar con la capacitación apropiada para las actividades de verificación. Es necesario realizar estas auditorías al menos una vez al año.

17. Adiestramiento

Es necesario identificar una autoridad capaz de administrar y verificar que los trabajos que influyen en la calidad se realizan en la forma que los documenta el sistema de calidad.

18. Servicios posventa

Soporte brindado posterior a la entrega del producto final al cliente, con esto se cubre cualquier falla posterior.

19. Técnicas estadísticas

Levantamiento de información para uso interno en futuros desarrollos de productos.[7]

3.1.3. ISO/IEC 9126 - Factores de Calidad del Software

ISO 9126 es un estándar internacional para la evaluación de la calidad del software. Está remplazado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos.

El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso.

El **modelo de calidad** establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y sub-características de la siguiente manera:

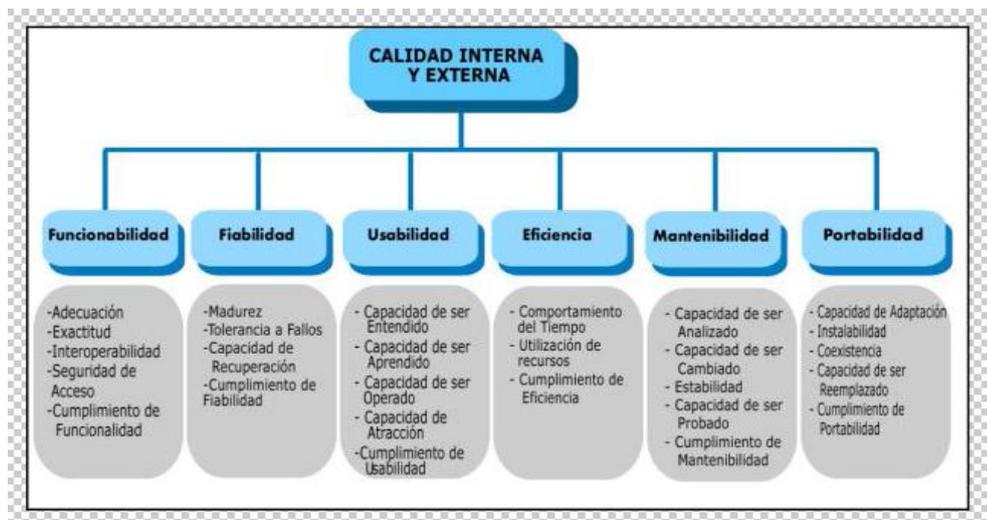


FIGURA 3.1 Modelo de calidad externa e interna

Funcionalidad - Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas.

- ✓ Idoneidad
- ✓ Exactitud
- ✓ Interoperabilidad
- ✓ Seguridad
- ✓ Cumplimiento de normas.

- **Fiabilidad** - Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.

- ✓ Madurez
 - ✓ Recuperabilidad
 - ✓ Tolerancia a fallos
- **Usabilidad o Facilidad de Uso** - Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.
 - ✓ Aprendizaje
 - ✓ Comprensión
 - ✓ Operatividad
 - ✓ Atractividad
- **Eficiencia** - Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.
 - ✓ Comportamiento en el tiempo
 - ✓ Comportamiento de recursos
- **Mantenibilidad** - Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.
 - ✓ Estabilidad

- ✓ Facilidad de análisis
 - ✓ Facilidad de cambio
 - ✓ Facilidad de pruebas
- **Portabilidad** - Conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra.
 - ✓ Capacidad de instalación
 - ✓ Capacidad de reemplazamiento
 - ✓ Adaptabilidad
 - ✓ Co-Existencia

La sub-característica Conformidad no está listada arriba ya que se aplica a todas las características. Ejemplos son conformidad a la legislación referente a usabilidad y fiabilidad.

Cada sub-característica (como adaptabilidad) está dividida en atributos. Un atributo es una entidad la cual puede ser verificada o medida en el producto software. Los atributos no están definidos en el estándar, ya que varían entre diferentes productos software.

Un producto software está definido en un sentido amplio como: los ejecutables, código fuente, descripciones de arquitectura, y así. Como resultado, la noción de usuario se amplía tanto a operadores como a

programadores, los cuales son usuarios de componentes como son bibliotecas software.

El estándar provee un entorno para que las organizaciones definan un modelo de calidad para el producto software. Haciendo esto así, sin embargo, se lleva a cada organización la tarea de especificar precisamente su propio modelo. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos de calidad.

Métricas internas son aquellas que no dependen de la ejecución del software (medidas estáticas).

Métricas externas son aquellas aplicables al software en ejecución.

La calidad en las métricas de uso están sólo disponibles cuando el producto final es usado en condiciones reales.

Idealmente, la calidad interna no necesariamente implica calidad externa y esta a su vez la calidad en el uso.

Este estándar proviene desde el modelo establecido en 1977 por McCall y sus colegas, los cuales propusieron un modelo para especificar la calidad del software. El modelo de calidad McCall está organizado sobre tres tipos de Características de Calidad:

- Factores (especificar): Describen la visión externa del software, como es visto por los usuarios.
- Criterios (construir): Describen la visión interna del software, como es visto por el desarrollador.
- Métricas (controlar): Se definen y se usan para proveer una escala y método para la medida.

ISO 9126 distingue entre fallo y no conformidad. Un fallo es el incumplimiento de los requisitos previos, mientras que la no conformidad es el incumplimiento de los requisitos especificados. Una distinción similar es la que se establece entre validación y verificación.

3.1.4. ISO/IEC 29119 – Pruebas de Software

La industria de la informática ha experimentado continuos cambios y ha evolucionado rápidamente. De estos cambios uno de los que ha tenido mayor incidencia en el desarrollo de las nuevas tecnologías ha sido el movimiento hacia los sistemas de evaluación de software crítico. Realizar una herramienta de trabajo que permita el asesoramiento en el desarrollo software mediante la utilización de ISO/IEC 29119 Software Testing. Este estándar, cuya elaboración comenzó en 2007 tiene como objetivo cubrir todo el ciclo de vida de las pruebas de sistemas software incluyendo los aspectos relativos a la

organización, gestión, diseño y ejecución de las pruebas, para reemplazar varios estándares IEEE y BSI sobre pruebas de software.

La estructura de ISO/IEC 29119 consta de cuatro partes:

1. Conceptos y Vocabulario
2. Proceso de Pruebas
3. Documentación de Pruebas
4. Técnicas de Prueba

De acuerdo con el plan de trabajo aprobado por ISO en una primera fase se elaboran las partes 2 y 3, y posteriormente las 1 y En Agosto de 2008 se liberó el primer borrador de trabajo (WorkingDraft), correspondiente a la parte 2 para revisión por parte de diferentes expertos internos y externos a los comités. Partiendo de los comentarios recibidos y de las sucesivas revisiones internas realizadas, se ha reorganizado y simplificado el modelo de procesos (Septiembre 2009). En 2010 se han liberado al SC7 los WD/CD de las partes 1, 2, y 3 para revisión. **[20]**

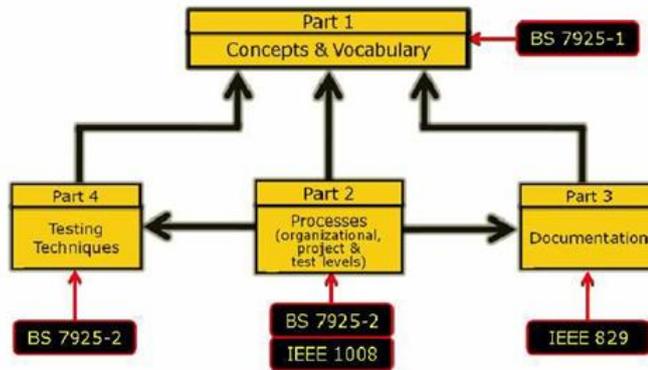


Figura 3.2 Estructura ISO/IEC 29119

Las partes 2 (procesos) y 3 (documentación) han sido planificadas en una primera secuencia, que tras sucesivas revisiones por expertos externos e internos a los diferentes comités producirán los primeros borradores del comité (CommitteeDraft: CD) para continuar con los procesos de tramitación y revisión habituales en la elaboración de los estándares ISO.

1. Conceptos y Vocabulario

EL objetivo es dar una visión general de la norma y de los conceptos generales de pruebas de software y proporcionar un vocabulario de términos de pruebas de software el examen. Esta parte incluyen los siguientes temas:

- Introducción a las pruebas de software
- Pruebas de software en un contexto organizativo
- El proceso de prueba
- La relación entre las pruebas y el desarrollo
- Implicaciones de los modelos de ciclo de vida de desarrollo software (por ejemplo, cascada, espiral, ágil)
- Tipos de pruebas, pruebas técnicas y fases/niveles de prueba
- Pruebas basadas en el Riesgo
- Elementos de Pruebas

2. Proceso de Pruebas

Cubre un modelo de proceso de pruebas genérico que pueda ser utilizado en cualquier desarrollo de software y ciclo d vida de pruebas. En particular, el modelo de procesos tal y como figura en el último borrador de trabajo está formado por tres niveles

- ✓ Procesos de la organización
- ✓ Procesos de Gestión
- ✓ Procesos Fundamentales

En el nivel superior se encuentran los procesos de la organización, que no son específicos de un determinado proyecto de pruebas. Permiten

definir las políticas y estrategias aplicables a toda la organización o a una línea de proyectos.

Para un proyecto de pruebas se tienen los niveles de procesos de gestión y procesos “fundamentales”. Se define un conjunto de procesos de gestión genéricos para permitir flexibilidad y adaptación a diferentes contextos. Estos procesos son la planificación, monitorización y control, y finalización de las pruebas. Los diferentes procesos de gestión se podrán instanciar en uno o en varios proyectos de pruebas dependiendo de la situación. Por ejemplo, en un proyecto simple puede existir solamente un plan de pruebas global (una sola instancia). En un proyecto más complejo puede existir un plan maestro y otros planes subordinados a éste para cubrir.

Una revisión y comparativa de modelos de procesos de prueba

Diferentes niveles de prueba (por ejemplo), pruebas de integración, sistema o aceptación) o diferentes tipos de prueba. Cada uno de ellos sería una instancia del proceso de gestión.

Los procesos fundamentales abarcan los aspectos técnicos de las pruebas: diseño e implementación, puesta a punto del entorno, ejecución de pruebas y la notificación de resultados de las pruebas.

También se incluyen variantes de los procesos para contemplar tantas pruebas dinámicas como estáticas.

Los siguientes diagramas ilustran las diferentes capas del proceso que se describen el Modelo de Proceso de Pruebas.

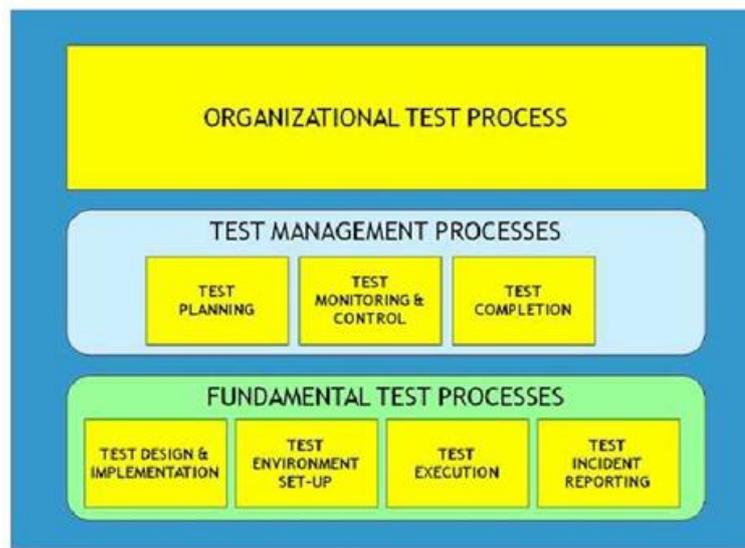
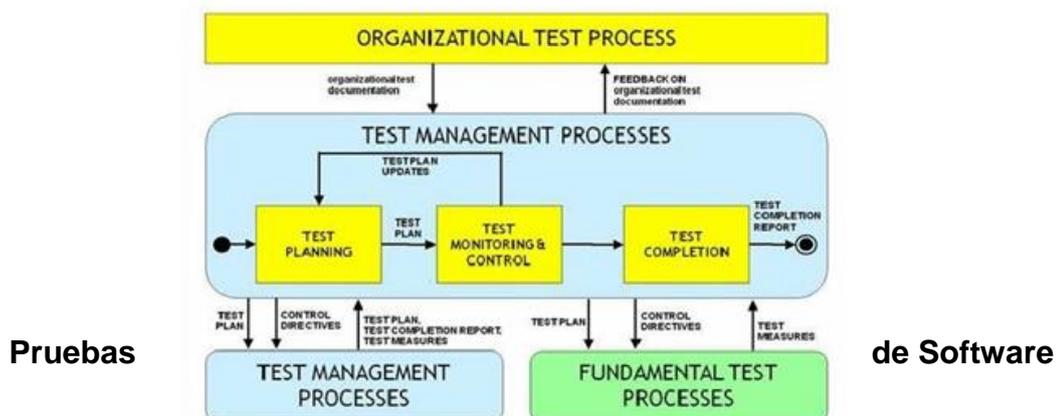


Figura 3.3 Vista General del Modelo de Proceso de Pruebas (Borrador Sep 2009)



En todo proceso de desarrollo de aplicaciones es indispensable la presencia

de un proceso de Pruebas de Software que coexista y se integre con el primero para garantizar así el buen funcionamiento y la calidad del producto final. Para lograr lo antes expuesto se debe partir del concepto de que las mismas desempeñan un papel fundamental en esta disciplina.

“Las pruebas constituyen una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente”. [31]

Toda prueba de software desempeña un papel fundamental en el desarrollo de cualquier tipo de aplicación, pero si se estudia la mejor forma de hacerlo, siguiendo los pasos de acuerdo con los especialistas en el tema, se incrementan las posibilidades de que esta llegue a un feliz término y arroje resultados más cercanos a los esperados, permitiendo así, realizar a posteriori un mejor análisis de la situación. Para ilustrar mejor esta situación, se presentan a continuación algunos objetivos, características y principios con que deben contar las Pruebas de Software.

Objetivos de las Pruebas de Software

- Probar si el software no hace lo que debe.
- Probar si el software hace lo que no debe, es decir, si provoca efectos secundarios adversos.

- Descubrir un error que aún no ha sido descubierto.
- Encontrar el mayor número de errores con la menor cantidad de tiempo y esfuerzo posibles.
- Mostrar hasta qué punto las funciones del software operan de acuerdo con las especificaciones y requisitos del cliente.

Características de las Pruebas de Software

- Alta probabilidad de encontrar un error. El ingeniero de software debe tener un alto nivel de entendimiento de la aplicación a construir para poder diseñar casos de prueba que encuentren el mayor número de defectos.
- No debe ser redundante. Uno de los objetivos de las pruebas es encontrar el mayor número de errores con la menor cantidad de tiempo y esfuerzo posibles, por lo cual no se deben diseñar casos de prueba que tengan el mismo propósito que otros, sino que se debe tratar de diseñar el menor número de casos de prueba que permitan probar adecuadamente el software y optimizar los recursos.
- Una buena prueba no debería ser ni demasiado sencilla ni demasiado compleja.

Tipos de Pruebas de Software

Las pruebas en conjunto tienen como objetivo general verificar y validar un software, independientemente de las características y el entorno donde se desarrollen, además de los recursos y los factores vinculados al proceso de desarrollo.

Funcionalidad

1. **Función:** Pruebas fijando su atención en la validación de las funciones, métodos, servicios, [caso de uso](#).
2. **Seguridad:** Asegurar que los datos o el sistema solamente es accedido por los actores deseados.
3. **Volumen:** Enfocada en verificando las habilidades de los programas para manejar grandes cantidades de datos, tanto como entrada, salida o residente en la [BD](#).

Usabilidad

Prueba enfocada a factores humanos, estéticos, consistencia en la interfaz de usuario, ayuda sensitiva al contexto y en línea, asistente documentación de usuarios y materiales de entrenamiento.

Fiabilidad

1. **Integridad:** Enfocada a la valoración exhaustiva de la robustez (resistencia a fallos).
2. **Estructura:** Enfocada a la valoración a la adherencia a su diseño y formación. Este tipo de prueba es hecho a las [aplicaciones Web](#) asegurando que todos los enlaces están conectados, el contenido deseado es mostrado y no hay contenido huérfano.
3. **Stress:** Enfocada a evaluar cómo el sistema responde bajo condiciones anormales. (extrema sobrecarga, insuficiente memoria, servicios y [hardware](#) no disponible, recursos compartidos no disponible).

Rendimiento

1. **Benchmark:** Es un tipo de prueba que compara el rendimiento de un elemento nuevo o desconocido a uno de carga de trabajo de referencia conocido.
2. **Contención:** Enfocada a la validación de las habilidades del elemento a probar para manejar aceptablemente la demanda de múltiples actores sobre un mismo recurso (registro de recursos, memoria).
3. **Carga:** Usada para validar y valorar la aceptabilidad de los límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante. La variación en carga

es simular la carga de trabajo promedio y con picos que ocurre dentro de tolerancias operacionales normales.

Soportabilidad

1. **Configuración:** Enfocada a asegurar que funciona en diferentes configuraciones de hardware y software. Esta prueba es implementada también como prueba de rendimiento del sistema.
2. **Instalación:** Enfocada a asegurar la instalación en diferentes configuraciones de [hardware](#) y [software](#) bajo diferentes condiciones (insuficiente espacio en disco, etc.) **[22]**

3. Documentación de prueba

La tercera parte de la norma cubre la documentación de prueba a través del ciclo de vida completo del software de prueba. Esto incluirá las plantillas que se pueden personalizar y que abarcan todas las fases del proceso de pruebas, incluyendo:

Prueba de la organización Documentación de Procesos:

- Política organizativa de prueba
- Estrategia organizativa de prueba

Gestión de pruebas proceso de documentación:

- Plan de pruebas (incluye la estrategia de prueba)
- Informe de las pruebas de estado
- Prueba de Informe Final

Prueba dinámica de la documentación del proceso:

- Prueba de las Especificaciones de Diseño
- Especificación de Casos de Prueba
- Procedimiento de Ensayo
- Requisitos de los datos de prueba
- Requisitos detallados entorno de prueba
- Entorno de prueba de la disponibilidad del informe
- Prueba Final
- Resultado de la prueba
- Prueba de registro de ejecución
- Prueba de Informe de Incidentes

4. Técnicas de ensayo

Las técnicas de ensayo de la norma cubre una variedad de técnicas dinámicas comunes de pruebas de software:

Especificación basada en Técnicas de prueba:

- ✓ Separación de equivalencia
- ✓ Clasificación método del árbol

- ✓ Análisis de Valor Límite
- ✓ Exámenes del Estado de Transición
- ✓ Decisión de la prueba de la mesa
- ✓ Causa-Efecto gráfica
- ✓ Sintaxis de Prueba
- ✓ Técnicas de ensayo combinatorias, incluyendo:
 - Todas las combinaciones
 - Prueba de pares
 - Cada opción Pruebas
 - Base de Pruebas de Selección
- ✓ Escenario de las pruebas
- ✓ Error Adivinar
- ✓ Pruebas al azar

Las técnicas basadas en la estructura de prueba:

- ✓ Declaración de Prueba
 - ✓ Rama de pruebas
 - ✓ Decisión de la Prueba
 - ✓ Condición de verificación, incluyendo:
 - Rama Condición de verificación
 - Rama Condición de verificación Combinación
 - Modificaba la decisión de Estado de Condición (MCDC)
- Pruebas

✓ Las pruebas de flujo de datos, incluyendo:

- Todas las definiciones
- All-c-usa
- Todos los utiliza-p-
- Todos los usos
- Todos los caminos-du-

También proporciona las definiciones de informativos de una variedad de calidad relacionados con los tipos de pruebas:

- Pruebas de accesibilidad
- Backup / Recuperación de Pruebas
- Compatibilidad de Pruebas
- Conversión de Pruebas
- Pruebas de recuperación de desastres
- Pruebas funcionales
- Pruebas de mantenimiento
- Portabilidad Pruebas
- Procedimiento de prueba
- Pruebas de confiabilidad
- Pruebas de Seguridad
- Ensayos de estabilidad
- Test de usabilidad

CAPÍTULO 4

METODOLOGÍA DE RESULTADOS

4.1. Antecedentes

La Empresa, donde se aplicó el estudio, considera necesaria la aplicación de una metodología de calidad de sistemas para la realización de las pruebas de los sistemas a fin de alcanzar la Calidad en el desarrollo de Software.

4.2. Introducción

La empresa objeto de nuestro estudio, es una Institución Privada, sin fines de lucro, que tiene como propósito prestar servicios asistenciales en las ramas de la salud, educación y atención a los ancianos de conformidad con las necesidades de la ciudad de Guayaquil, por razones de privacidad, no podemos especificar su nombre.

La empresa objeto de nuestro de estudio, fue creada el 29 de Enero de 1988. Su sede es Guayaquil, capital económica del Ecuador y ciudad en la que se asienta el 27% de la población nacional.

Actualmente ofrece trabajo a más de 6.140 personas, jefes o madres de familia, con lo que aporta significativamente a la ocupación laboral de profesionales en diferentes áreas y de técnicos medios.

MISIÓN: Es una Institución de servicios para las personas de escasos recursos económicos en Guayaquil y sus zonas de influencia, prioritariamente en: salud (general, maternidad, infantil y mental), educación y ancianidad.

VISIÓN: La Empresa objeto de nuestro estudio será un modelo en la prestación de servicios asistenciales para la población más necesitada con gestión moderna, autónoma y auto sustentable.

4.3. Objetivos del trabajo Diseño de un sistema de calidad de pruebas de software para una empresa de servicios basado en el estándar ISO 9126.

- Elaborar un plan de calidad de Software para garantizar la funcionalidad, confiabilidad y seguridad de los Sistemas de Información de la Organización.

- Establecer una Metodología de pruebas para ser desarrolladas en todos los sistemas de Información previo su puesta a producción.
- Aplicar la Metodología de pruebas al Diseño de un sistema de calidad de pruebas de software para una empresa de servicios basado en el estándar ISO 9126.
- Diseñar los instrumentos y formularios necesarios para la ejecución de pruebas
- Establecer las técnicas y herramientas que se deben usar en la realización de pruebas.

4.4. Metodología de Investigación

Para la realización de la presente investigación, se han considerado algunos métodos de acuerdo a las características específicas del tema de estudio, por lo que se ha utilizado en forma general el método científico, que se entiende como un procedimiento organizado y planificado que sirve para descubrir la existencia de procesos y resultados de la investigación, así como el método inductivo y deductivo.

Esta investigación se ha diseñado luego de haber realizado un análisis de las diversas situaciones que se presentan en la empresa xyz objeto de estudio.

Método Científico.

Tenemos tres definiciones básicas que nos explican el concepto de lo que es el método científico y son:

1. El método científico es el conjunto de procedimientos lógicos que sigue la investigación para descubrir las relaciones internas y externas de los procesos de la realidad natural y social.
2. Llamamos método científico a la serie ordenada de procedimientos de que se hace uso en la investigación científica para obtener la extensión de nuestros conocimientos.
3. Se entiende por método científico al conjunto de procesos que el hombre debe emplear en la investigación y demostración de la verdad.

En el estudio se decidió realizar una encuesta para obtener la información necesaria, esto fue debido a que las encuestas proveen de un mecanismo para obtención de información, el cual es muy práctico y fácil de comprender.

4.4.1. Técnicas e Instrumentos de Investigación

La técnica es un estudio observacional en el cual el investigador busca recaudar datos por medio del instrumento de investigación que es un cuestionario prediseñado, y no modifica el entorno ni controla el proceso que está en

observación (como sí lo hace en un experimento). Los datos se obtienen a partir de realizar un conjunto de preguntas normalizadas dirigidas a una muestra representativa o al conjunto total de la población estadística en estudio, formada a menudo por personas, empresas o entes institucionales, con el fin de conocer estados de opinión, características o hechos específicos. El investigador debe seleccionar las preguntas más convenientes, de acuerdo con la naturaleza de la investigación. [3]

Esta encuesta se la realizaría de manera presencial, ya que de esta manera se puede estar presente en el proceso para supervisar y aclarar cualquier duda que tenga el encuestado, así como también poder obtener mayor precisión en cuanto a la información que se obtenga.

Las razones que nos llevaron a aplicar esta metodología, de realizar las encuestas de manera presencial fueron:

- Se procedió a la elaboración de un plan piloto, en el cual se planteo la posibilidad de realizar las encuestas vía email, pero no se obtuvieron buenos resultados ya que las personas no daban respuestas a nuestra solicitudes enviadas por correo

electrónico, lo que nos llevo a decidir encuestar de manera presencial.

- También al elaborar la encuesta nos dimos cuenta de que las personas que iban a ser encuestadas no estarían familiarizadas con muchos de los términos utilizados en las preguntas de la encuesta, por tanto si queríamos obtener resultados claros y precisos era necesario estar presentes mientras ellos daban contestación a las preguntas con el fin de aclarar las dudas que tuvieran.

Lo primero que se definió en cuanto al plan de levantamiento de información, fue la herramienta que se iba a utilizar para obtener los datos necesarios, la herramienta escogida fue la encuesta, por razones que ya se explicaron anteriormente. Posterior a esto se elaboró la lista de preguntas que iban a conformar la encuesta, para luego hacer correcciones a dicha encuesta en base a opiniones de personas con experiencia en este tema, hasta finalmente obtener la versión final de la encuesta.

De todas formas se necesitó pulir varias veces la encuesta, ya que era importante poder obtener resultados fiables, los cuales dependerían en la mayoría de los encuestados, de la claridad y familiarización de la terminología usada en esta encuesta. **Ver Anexo 2.**

A las personas encuestadas, pertenecientes al departamento de sistema, se les proporcionó la encuesta impresa para que sea fácil de llenar, además de que a cada persona encuestada se le pidió que firmara una hoja de constancia de haber participado en la encuesta.

Una vez culminada la fase de recolección de información, se procedió a realizar el análisis estadístico de los datos con el fin de transformar dichos datos en información relevante para el contexto del presente estudio.

Para el análisis estadístico se utilizó como herramienta a Excel, por la familiarización y el manejo completo que se posee de esta herramienta por parte del grupo de desarrollo.

A continuación presentamos de manera gráfica los resultados del estudio:

Sección 1. De la encuesta: Acerca del entrevistado.

Esta sección consiste en los datos de la persona encuestada y que nos sirven de referencia, principalmente para saber el cargo que ocupa dentro de la organización.

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL INSTITUTO DE CIENCIAS MATEMÁTICAS Guayaquil - Ecuador	
1. ACERCA DEL ENTREVISTADO	
<i>Nombre:</i>	_____
<i>Edad:</i>	_____
	<i>A) Masculino</i> <input type="checkbox"/>
	<i>B) Femenino</i> <input type="checkbox"/>
<i>Cargo:</i>	_____
<i>Email</i>	_____

Sección 2. De la encuesta: Acerca de la documentación del proceso de desarrollo.

Pregunta 2.1: ¿Con qué frecuencia realizan documentación del proceso de desarrollo de software?

Cuadro N° 2.1 Frecuencia de la documentación en el proceso del desarrollo de software.

	PERSONAL	
	#	%
<i>En todo proyecto</i>	2	11%
<i>En algunos</i>	4	19%
<i>Pocos</i>	6	30%
<i>Ninguno</i>	8	40%

Gráfico N° 2.1 Frecuencia de la documentación en el proceso del desarrollo de software.

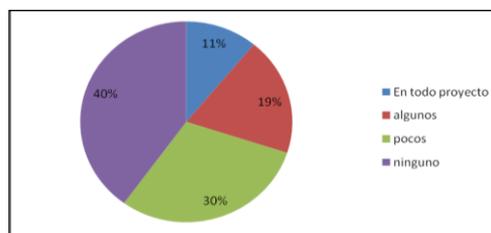


Gráfico 2.1 Frecuencia de la documentación en el proceso del desarrollo de software

Análisis e Interpretación.

Del total de encuestados, el 40% respondió que no documentaban ninguno de sus proyectos, mientras que un 30% contestó que si documenta aunque sea pocos proyectos, lo más preocupante es que solamente un 19% documenta algunos proyectos y únicamente un 11% documenta todos sus proyectos. Si observamos estos resultados podemos darnos cuenta que la mayoría de software no están debidamente documentado lo cual dificulta la calidad del producto final.

Pregunta 2.2: ¿Cuán importante cree ud. Que es la documentación en el proceso de desarrollo de software?

Cuadro N° 2.2 Importancia de la documentación en el proceso del desarrollo de software.

PERSONAL		
	#	%
<i>Muy Importante</i>	11	55%
<i>Poco Importante</i>	7	35%
<i>No tiene Importancia</i>	2	10%

Gráfico N° 2.2 Importancia de la documentación en el proceso del desarrollo de software.

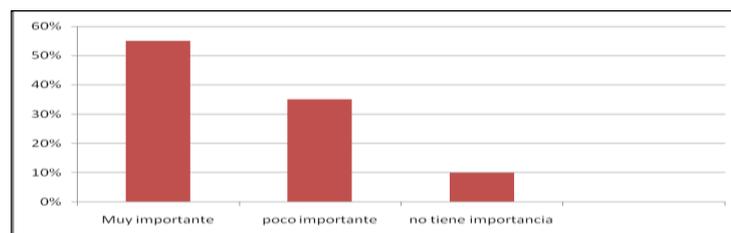


Gráfico 2.2 Importancia de la documentación en el proceso del desarrollo de software

Análisis e Interpretación.

En contraste con los resultados de la pregunta anterior podemos ver que el 55% de los encuestados considera muy importante la documentación dentro de un proyecto, esto nos indica que están conscientes de la importancia de la documentación, sin embargo no lo aplican en el desarrollo del mismo.

Pregunta 2.3: ¿Por Qué motivo no incorporan documentación al proceso de desarrollo de software?

Cuadro N° 2.3 Incorporación de la documentación en el proceso de desarrollo de software.

	PERSONAL	
	#	%
<i>Falta de organización de los miembros del equipo</i>	7	35%
<i>Desconocimiento de cómo documentar</i>	10	50%
<i>No le interesa incorporar documentación al proceso</i>	1	5%
<i>Otro</i>	2	10%

Gráfico N° 2.3 Incorporación de la documentación en el proceso de desarrollo de software.

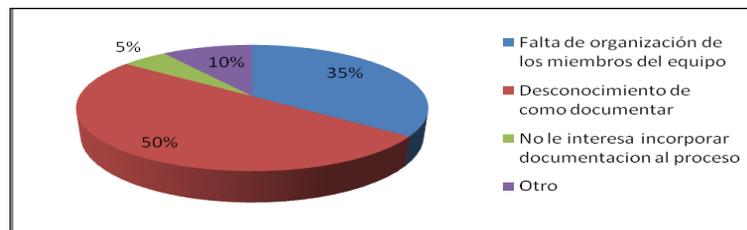


Gráfico 2.3. Incorporación de la documentación en el proceso de desarrollo de software

Análisis e Interpretación.

Como podemos observar el principal motivo por el cual no se incorpora la documentación al proceso de desarrollo 55% de encuestados es por el desconocimiento de las normas para documentar, y se debe poner énfasis también en la falta de organización que existe dentro de los equipos de desarrollo, éstos son los principales motivos por los cuales no existe documentación en los proyectos.

Sección 3: Acerca del equipo de desarrollo.

Pregunta 3.1: ¿Cuántas personas integran su equipo?

Análisis e Interpretación.

Los resultados de esta pregunta nos dieron como indicativo de que los equipos de desarrollo por proyecto constan de mínimo 3 personas, teniendo como máximo llegar a 10 personas por equipo de desarrollo.

Pregunta 3.2: ¿Cuál es el número de personas designadas para el desarrollo de un proyecto?

Análisis e Interpretación.

Reafirmando lo analizado en la pregunta anterior, podemos comentar que para un proyecto pequeño, el equipo de desarrollo se conforma de 3 personas, mientras que para un proyecto mediano 5 personas es el promedio y en un proyecto grande la constante es de 8 a 10 personas.

Pregunta 3.3. ¿Su equipo de desarrollo tiene roles definidos?

Cuadro N° 3.3 Roles Definidos

PERSONAL		
	#	%
<i>SI</i>	11	55%
<i>NO</i>	9	45%

Gráfico N° 3.3 Roles Definidos

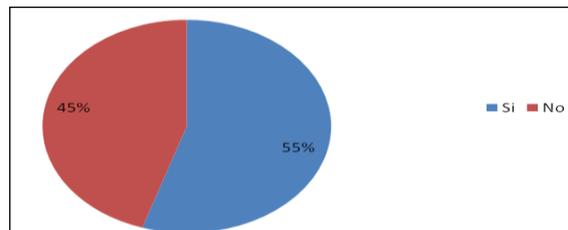


Gráfico 3.3. Roles Definidos

Análisis e Interpretación.

Como podemos observar, es importante recalcar que en un 55% los equipos de desarrollo si poseen roles bien definidos, lo cual es importante para asegurar el papel y la responsabilidad de cada miembro del equipo.

Pregunta 3.4: ¿Cuán importante cree Ud. Que es definir roles en el equipo para la organización del proyecto?

Cuadro N° 3.4 Roles para la equipo de organización del proyecto.

	PERSONAL	
	#	%
<i>Muy Importante</i>	18	90%
<i>Poco Importante</i>	1	5%
<i>No tiene importancia</i>	1	5%

Gráfico N° 3.4 Roles para la equipo de organización del proyecto.

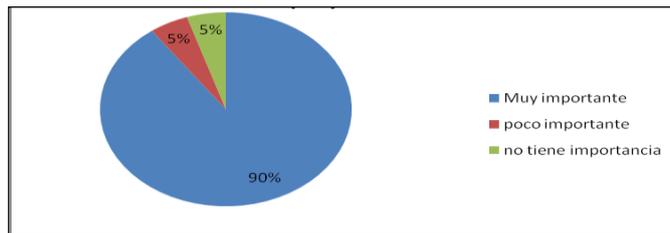


Gráfico 3.4. Roles para el equipo de organización del proyecto.

Análisis e Interpretación.

Es muy importante recalcar que en un 90% de los encuestados piensan que es de suma importancia el definir roles dentro del equipo de desarrollo, esto nos garantiza que al menos existe una organización y tareas definidas en los proyectos.

Pregunta 3.5: ¿Qué roles tiene su equipo de desarrollo?

Cuadro N° 3.5 Roles para el rol del equipo de desarrollo

	PERSONAL	
	#	%
<i>Programador</i>	20	100%
<i>Equipo de Pruebas</i>	14	70%
<i>Diseñadores</i>	10	50%
<i>Gerente de Proyectos</i>	20	100%
<i>Integradores</i>	10	50%
<i>Otros</i>	1	5%

Gráfico N° 3.5 Roles para el rol del equipo de desarrollo

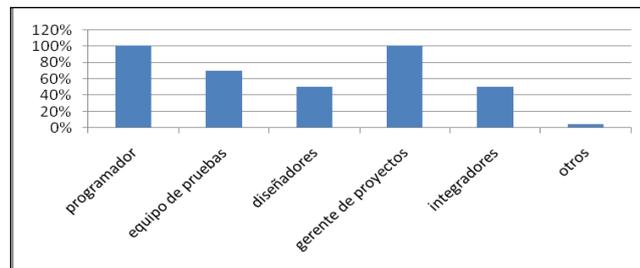


Gráfico 3.5. Roles para el rol del equipo de desarrollo

Análisis e Interpretación.

Todos los equipos de prueba poseen gerentes de proyecto y desarrolladores, pero sin duda alguna preocupa que aún no todos los equipos de desarrollo posean equipo de prueba, y es en base a esto que este estudio cobra importancia.

Sección 4: Acerca de las metodologías y estándares de calidad de software.

Pregunta 4.1: Ud. realiza esfuerzos a informarse sobre la calidad de software:

Cuadro N° 4.1 Esfuerzos sobre la calidad de software.

PERSONAL		
	#	%
<i>SI</i>	12	60%
<i>NO</i>	8	40%

Gráfico N° 4.1 Esfuerzos sobre la calidad de software.

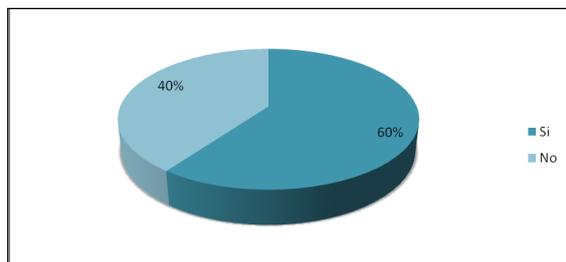


Grafico 4.1. Esfuerzos sobre la calidad de software.

Análisis e Interpretación.

En esta pregunta, se consultó a los encuestados acerca de si dedicaban esfuerzos a informarse acerca de la calidad de software, un 60% dijo que si mientras un 40% dijo que no.

Pregunta 4.2: ¿Cuán importante cree Ud. que es la aplicación de alguna metodología en la calidad del producto final?

Cuadro N° 4.2 Aplicación de la metodología en la calidad del producto final

	PERSONAL	
	#	%
<i>Muy Importante</i>	10	50%
<i>Poco Importante</i>	6	30%
<i>No tiene Importancia</i>	4	20%

Gráfico N° 4.2 2 Aplicación de la metodología en la calidad del producto final

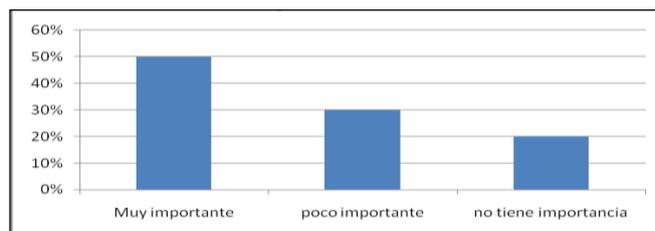


Gráfico 4.2. Aplicación de la metodología en la calidad del producto final

Análisis e Interpretación.

El 50% de los encuestados considera muy importante la metodología de calidad de software en el resultado del producto final, mientras que un 30% lo considera poco importante y un 20%, que preocupa, considera que no tiene importancia.

Pregunta 4.3: ¿Qué metodología(s) o técnica(s) de aseguramiento de calidad de software conoce?

Cuadro N° 4.3 Metodologías y técnicas de calidad de software

	PERSONAL	
	#	%
<i>CMMI</i>	7	36%
<i>Paradigma GQM</i>	1	4%
<i>Modelo de Boehm</i>	2	9%
<i>Modelo de McCall</i>	3	14%
<i>Ninguna</i>	6	32%
<i>Otros</i>	1	5%

Gráfico N°4.3 Metodologías y técnicas de calidad de software

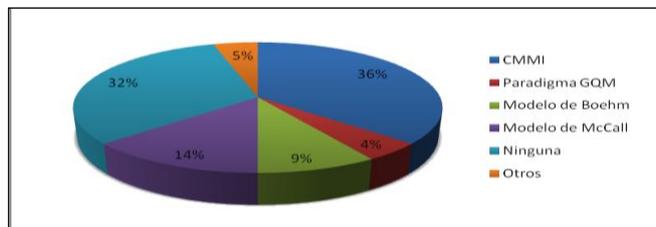


Gráfico 4.3. Metodologías y técnicas de calidad de software

Análisis e Interpretación.

Esta pregunta se centra en establecer el nivel de conocimiento del entrevistado en el campo de la calidad de software, específicamente en cuanto a las metodologías y técnicas de aseguramiento de calidad. El 36% de los entrevistados afirmó conocer acerca de CMMI, el 5% conocía el Paradigma GQM, el 9% conocía el modelo de Boehm, el 14% tenía conocimiento sobre el modelo de McCall, mientras el 32% no tenía conocimientos acerca de ninguna metodología o técnica de aseguramiento de calidad, y por último el 5% conocían otras metodologías (SPICE).

Pregunta 4.4: ¿Cuáles son las ventajas que usted conoce de la(s) metodología(s) escogida(s) anteriormente sobre las demás mencionadas?

Ventajas de las metodologías de calidad de software	PERSONAL	
	#	%
<i>Permitir cumplir con los tiempos establecidos</i>	6	30%
<i>Fácil de aplicar a los procesos</i>	6	30%
<i>Logran mayor satisfacción en el cliente</i>	7	35%
<i>Existe un mayor conocimiento sobre esta(s) metodología(s)</i>	5	25%
<i>Otra</i>		0%

Gráfico N° 4.4 Ventajas de las metodologías de calidad de software

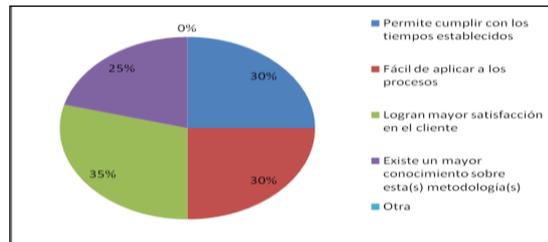


Gráfico 4.4. Ventajas de las metodologías de calidad de software

Análisis e Interpretación.

Esta pregunta se enfoca en establecer cuál o cuáles son los puntos fuertes de las metodologías escogidas por los encuestados. El 30% dijo que cumplían con los tiempos establecidos, el 30% dijo que eran fáciles de aplicar, el 35% contestó que ayudaban a alcanzar mayor satisfacción por parte de los clientes, el 25% dijo que su fuerte era que existía un mayor conocimiento de dichas metodologías, el 0% contestó que no le veía ninguna ventaja.

Pregunta 4.5: ¿Conoce Ud. Organizaciones que publiquen estándares de calidad?

Cuadro N° 4.5 Conoce organizaciones de estándares de calidad

PERSONAL		
	#	%
<i>SI</i>	14	70%
<i>NO</i>	6	30%

Gráfico N° 4.5 Conoce organizaciones de estándares de calidad

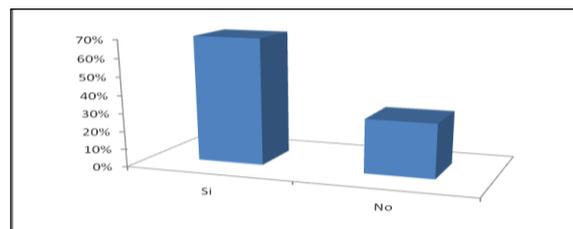


Gráfico 4.5. Conoce organizaciones de estándares de calidad

Análisis e Interpretación.

El 70% de los encuestados afirmó conocer alguna organización que emita estándares de calidad, mientras que el restante 30% afirmó no conocer organización alguna que emita estándares de calidad.

Pregunta 4.6: ¿Qué Organizaciones que publiquen estándares de calidad conoce?

Cuadro N° 4.6 Organizaciones de estándares de calidad

PERSONAL		
	#	%
<i>ISO</i>	20	100%
<i>IEEE</i>	9	45%
<i>Otros</i>	0	0%

Gráfico N° 4.6 Organizaciones de estándares de calidad

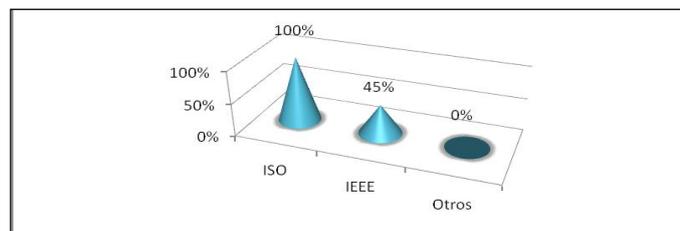


Gráfico 4.6. Organizaciones de estándares de calidad

Análisis e Interpretación.

La totalidad de los encuestados respondió que conocían acerca de la ISO, mientras un 45% contestó que conocían acerca de la IEEE y ninguno de los encuestados tenía conocimiento acerca de otras organizaciones de este tipo.

Pregunta 4.7: ¿De la(s) organizaciones escogidas en la pregunta anterior ¿Qué estándares de calidad conoce?

Análisis e Interpretación.

En esta pregunta la mayoría de los encuestados comentaron que conocían los estándares ISO 9000, 9001 y 9126. En esta pregunta no existe gráfico, por ser una pregunta abierta.

Pregunta 4.8: ¿Conoce ud. El estándar de calidad ISO 9126?

Cuadro N° 4.8 Estándar de Calidad ISO 9126

PERSONAL		
	#	%
<i>SI</i>	8	38%
<i>NO</i>	12	62%

Gráfico N° 4.8. Estándar de Calidad ISO 9126

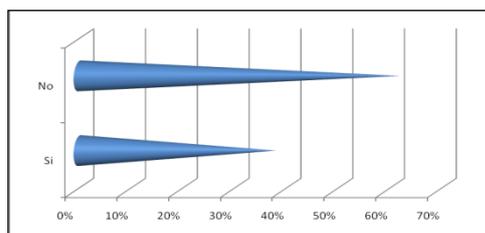


Gráfico 4.13. Estándar de Calidad ISO 9126

Análisis e Interpretación.

En esta pregunta el 70% de los encuestados afirmó conocer acerca del estándar ISO 9126, esto es importante ya que aunque no lo hayan investigado a profundidad, al menos tienen una noción de este estándar o para qué sirve.

Pregunta 4.9: ¿Cuál cree Ud. que es el atributo más importante del estándar ISO 9126?

Cuadro N° 4.9 Atributos ISO 9126

	PERSONAL	
	#	%
<i>Funcionalidad</i>	7	35%
<i>Confiabilidad</i>	2	10%
<i>Usabilidad</i>	2	10%
<i>Eficiencia</i>	4	20%
<i>Facilidad de mantenimiento</i>	5	23%
<i>Portabilidad</i>	0	2%

Gráfico N° 4.9 Atributos ISO 9126

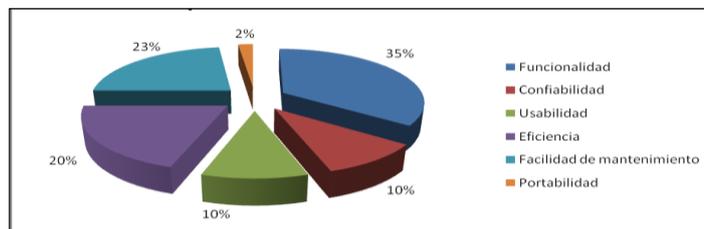


Gráfico 4..9. Atributos ISO 9126

Análisis e Interpretación.

Aunque en esta pregunta las respuestas fueron en porcentajes similares, podemos resaltar que la funcionalidad, la eficiencia y la facilidad de mantenimiento son las características que nuestros encuestados buscan de lograr en sus productos finales, mientras que no se preocupan tanto por la usabilidad, lo cual podría traer inconvenientes con los usuarios.

Sección 5: Acerca de las pruebas de calidad.

Pregunta 5.1: ¿Incorporan una fase de pruebas en las etapas de su desarrollo?:

Cuadro N° 5.1 Pruebas en el desarrollo

PERSONAL		
	#	%
<i>SI</i>	14	70%
<i>NO</i>	6	30%

Gráfico N° 5.1 Pruebas en el desarrollo

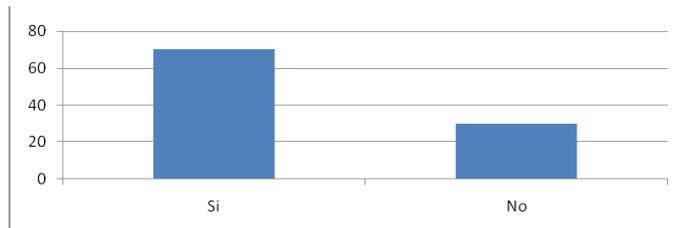


Gráfico 5.1. Pruebas en el desarrollo

Análisis e Interpretación.

El 70% de los encuestados contestaron que si tenían una fase de pruebas en sus proyectos, mientras que un 30% no incorporan fases de pruebas en sus proyectos, lo cual compromete mucho la calidad final del entregable.

Pregunta 5.2: ¿Para el control de los errores ud. Realiza?

Cuadro N° 5.2 Control de los errores

	PERSONAL	
	#	%
<i>Inspecciones tempranas</i>	10	50%
<i>Análisis de defectos</i>	5	25%
<i>Revisiones periódicas</i>	4	20%
<i>Se controlan cuando estos se producen</i>	1	5%
<i>Otros</i>	0	0%

Gráfico N° 5.2 Control de los errores

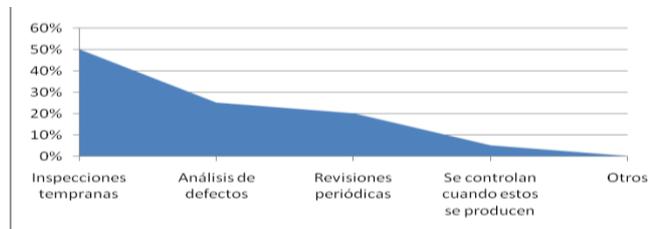


Gráfico 5.2. Control de los errores

Análisis e Interpretación.

La mitad de los encuestados realizan ligeras impresiones tempranas a sus proyectos, es decir que en cada paso del desarrollo incorporan pequeñas pruebas pero de modo empírico, aunque lo ideal sería que crezca el porcentaje de equipos que realicen revisiones periódicas y establecidas en los cronogramas. También cabe recalcar que cada vez existen menos equipos que actúan de forma reactiva.

Pregunta 5.3: ¿Qué tipo de pruebas se realizan?

Cuadro N° 5.3 Tipo de Pruebas

	PERSONAL	
	#	%
<i>Pruebas Unitarias</i>	5	25%
<i>Pruebas de Integración</i>	5	25%
<i>Pruebas orientas a objetos</i>	10	48%
<i>Pruebas caja negra</i>	0	1%
<i>Pruebas caja blanca</i>	0	1%
<i>Otros</i>	0	0%

Gráfico N° 5.3 Tipo de Pruebas

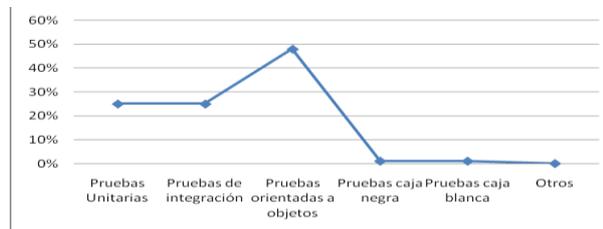


Gráfico 5.3. Tipo de Pruebas

Análisis e Interpretación.

De esta pregunta es importante señalar que los encuestados no incorporan las pruebas de caja blanca para revisar el correcto funcionamiento del sistema con entradas aleatorias ni de caja negra para revisar que las salidas de una determinada entrada sean las deseadas, esto es crítico ya que este tipo de pruebas son importantes para el correcto funcionamiento del sistema. Al contrario los equipos se centran más en pruebas a objetos, pruebas a módulos del sistema por separado y pruebas al sistema completo.

Pregunta 5.4: ¿Qué porcentaje de defecto encuentran en las pruebas?

Cuadro N° 5.4 Porcentaje de defectos en las pruebas

PERSONAL		
	#	%
[0, 25%)	7	35%
[25,50%)	9	45%
[50,75%)	3	15%
[75,100%)	1	5%
Desconoce	0	0%

Gráfico N° 5.4 Porcentaje de defectos en las pruebas

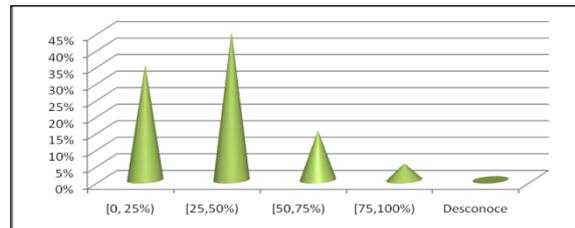


Gráfico 5.4. Porcentaje de defectos en las pruebas

Análisis e Interpretación.

Podemos observar que el porcentaje de defectos que se encuentran sigue siendo relativamente alto, entre el 25 y 50%, lo que nos lleva a pensar que no hay una mejora de procesos ni un análisis correcto previo al desarrollo. En este aspecto aún queda mucho por trabajar.

4.4.2. Análisis e Interpretación de resultados.

De la presentación de los resultados podemos extraer los siguientes puntos que son importantes recalcar:

- Todos los encuestados habían participado en proyectos de desarrollo de software por lo cual, nuestra población seleccionada (departamento de sistemas de la empresa) para la encuesta fue la correcta, permitiéndonos obtener resultados con un alto grado de confiabilidad.
- Gran parte los encuestados conocían acerca de calidad de software y estándares de calidad, por lo que no hubo que invertir mucho tiempo explicándole cual era el objetivo en torno a esta encuesta.
- En ciertos temas, como los estándares de calidad de software, fue necesario realizar una introducción de los mismos para que puedan entender el objetivo de las preguntas.

4.5 Presentación del sistema de calidad de pruebas

4.5.1 Metodología de pruebas diseñada

Antecedentes

La Empresa, donde se aplicó el estudio, considera necesaria la aplicación de una metodología de calidad de sistemas para la realización de las pruebas de los sistemas y por ello surge el presente documento que pretende aplicar una metodología de aseguramiento de calidad de software acorde a sus necesidades.

Introducción.

Este documento está destinado a dar una orientación sobre la forma de desarrollar las pruebas de un Sistema, para lograr asegurar la calidad de software, identificando las técnicas y herramientas que son aplicables en cada etapa, dando especial énfasis a los lineamientos de cómo deben ser desarrolladas.

Sus usuarios o destinatarios serán los programadores y analistas de pruebas, tanto en sus pruebas individuales como con usuarios, entre otros que estén vinculados al área de sistema y que se encarguen de certificar la calidad de los sistemas desarrollados tanto interna como externamente.

4.5.2 Fases de la metodología de prueba

Luego de varias sesiones de trabajo y de revisión de metodologías y procesos, se establecieron las actividades o fases involucradas en la metodología.

Las fases que integran la Metodología de Prueba de Sistemas son:

1. Investigación Preliminar
2. Conocimiento del sistema
3. Planeación
4. Diseño
5. Ejecución
6. Documentación
7. Análisis de errores/Depuración
8. Pruebas con los Usuarios
9. Pruebas en el ambiente de producción

En todo proyecto de sistemas se debe establecer un equipo de trabajo el cual debe participar activamente en todas las fases de la metodología de prueba

Fase 1. Investigación Preliminar

Objetivo

Descubrir lo que el sistema debe hacer. En esta primera fase es donde se descubrirá por qué surgió la necesidad de crear el sistema, se conocerán los requerimientos del área usuaria y se revisará el documento de especificaciones que contiene en forma detallada lo que el sistema debe hacer.

La investigación preliminar proporciona una comprensión inicial del sistema y permite conocer o recordar las especificaciones.

Entradas:

Entre las principales entradas a esta fase, tenemos:

1. Formulario de Requerimientos. **ANEXO 5A**

2. Documento de especificaciones, el cual contendrá los elementos del sistema: consulta, procesos y reportes sugeridos o propuestos, así como los elementos requeridos.

Salidas:

Narrativa resumida de especificaciones. **ANEXO 5B**

Técnicas aplicadas: ANEXO 5C

Análisis

Documentación

Entrevistas a usuarios

Herramientas: ANEXO 5C

Procesadores de texto

Fase 2. Conocimiento del sistema

Objetivo

Descubrir lo que el sistema hace.

En esta fase con el conocimiento previo de la documentación existente, el Analista de pruebas se reúne con el Analista-

programador, para conocer el funcionamiento y la operatividad del sistema, tal cual el programador lo desarrolló, basado en el requerimiento y en el documento de especificaciones.

Entradas:

Documento de especificaciones técnicas del programador

Salidas:

Con esta información se creará:

- 1.-Un diagrama de los elementos del sistema
- 2.-Un documento denominado ***Instructivo del módulo o sistema.***

Técnicas aplicadas: ANEXO 5C

Análisis

Documentación

Entrevistas a programadores (En la interacción con el sistema)

Herramientas:

Procesadores de texto

Microsoft Visio

Fase 3. Planeación de las pruebas

Objetivo

1. Establecer un plan efectivo de pruebas para garantizar que sus sistemas sean preparados eficientemente y cumplan con los requerimientos exigidos por el usuario.
2. Coordinar de manera eficiente y coherente los esfuerzos internos, con los usuarios.

3. Identificar en qué puntos del proceso podrían existir debilidades para tomar los criterios necesarios que garantice la adecuada operación de los sistemas

4. Definir las pruebas a realizar.

Planificar es el acto de idear o determinar un curso de acción para lograr un resultado final. Planificar implica saber, establecer objetivos, desarrollar estrategias que determinarán las metas: establecer prioridades, secuencias, agenda de actividades y verificar cumplimientos de políticas y procedimientos.

Durante esta fase se definen los problemas de negocios, se articula la visión y el alcance del sistema, se identifican los objetivos y restricciones de acuerdo como fueron planteados por el usuario, y se determinan los factores de éxito.

Entradas:

Problemas del negocio o casos

Objetivos

Restricciones (Fechas críticas)

Riesgos

Sistema de control interno (políticas, normas y procedimientos)

Criterios de aceptación

El elemento quizás más importante es el criterio de aceptación que se va a emplear para los componentes o módulos del producto. Es importante notar que la Calidad del producto no es lo que se quiere

probar, puesto que este debe ser un elemento inmerso en la construcción y desarrollo de toda la solución.

Salidas:

Se debe elaborar un plan de pruebas documentado para sus módulos. **ANEXO 5D**

El plan debe contener:

1. Alcance
 - a. Dependencias o entidades vinculadas o relacionada con el sistema que se prueba
2. Descripción o enfoque de pruebas
3. Tipos de pruebas (Uso de múltiples sistemas de pruebas) **ANEXO 5E**
4. Herramientas de pruebas a utilizar
5. Calendario
6. Recursos requeridos (Personal, terceros, equipos, entre otros)
7. Necesidades de personal y de formación
 - a. Usuario (s) involucrado (s)
 - b. Administrador de seguridad
 - c. Soporte de Sistema (Personal de Redes, Comunicaciones, Soporte a usuario, entre otros miembros del área de Sistemas, según corresponda)
 - d. Coordinador de la prueba
8. Responsabilidades en la organización y realización de las pruebas.
9. Necesidades generales de entorno.
 - a. Ambiente de prueba
 - b. Usuario (s) de prueba
 - c. Enfoque de Seguridad requerido en la prueba

- d. Bibliotecas, archivos
- e. Generación de datos en caso de que sea requerido
- 10. Esquema de tiempos (con tiempos estimados, hitos, etc).
- 11. Procedimientos de aceptación y rechazo de los resultados
- 12. Plan de contingencia

Note que puede haber un plan global que explicita el énfasis a realizar sobre los distintos tipos de pruebas (ej: verificación, integración)

En los tipos de pruebas se deberá hacer una adecuada selección de los tipos y se deberá describir brevemente los procedimientos de pruebas (escenarios, técnicas, casos).

En particular, podemos (y debemos) crear pruebas unitarias, de integración y de sistema

Que incluyan los casos de uso, pruebas que busquen errores en el manejo de las restricciones, entre otros aspectos.

Técnicas aplicadas: ANEXO 5C

Análisis

Documentación

Herramientas:

Procesadores de texto

Microsoft Project

Fase 4. Diseño de las pruebas

Objetivo:

Especificar los detalles necesarios sobre el enfoque general reflejado en el plan e identificar las características que se deben probar.

Para realizar este sistema de pruebas utilizaremos varias herramientas, entre ellas:

- Experiencia de los desarrolladores, con la guía de los participantes podemos definir cuáles son los errores más comunes en sus proyectos.
- Se utiliza caso de pruebas, todo en base a que las características de funcionamiento estén acorde al levantamiento de requerimientos. Al momento que se descubre una desviación de las especificaciones se crea una lista de deficiencias.
- Aunque es casi imposible saber cómo los usuarios utilizarán el software se pueden realizar simulaciones del uso del sistema.

Entradas:

Plan general de pruebas. **ANEXO 5D**

Plan detallado de pruebas. **ANEXO 5D**

Matrices de pruebas.

Salidas:

Plan detallado de pruebas que incluye los procedimientos de pruebas

El procedimiento de prueba detallado debe contener por cada tipo de prueba, lo siguiente: **ANEXO 5D**

a) Identificador (único) para la especificación.

Debe proporcionar también una referencia del plan asociado (si existe).

b) Objetivo del procedimiento y lista de caso que se ejecutan con él.

c) Introducción y resumen de elementos y características a probar.

d) Requisitos especiales para la ejecución (p. ej., entorno especial o personal especial).

e) Características a probar de los elementos software (y combinaciones de características, controles, riesgos, entre otros).

f) Características que no se prueban.

g) Validación de datos.

Detalles sobre el plan de pruebas del que surge este diseño, incluyendo las actividades, técnicas de prueba específicas, herramientas y los métodos de análisis de resultados.

- 1) Pasos en el procedimiento. Además de la manera de registrar los resultados y los incidentes de la ejecución, se debe especificar:
- 2) La secuencia necesaria de acciones para preparar la ejecución.
- 3) Acciones necesarias para empezar la ejecución.
- 4) Acciones necesarias durante la ejecución.
- 5) Cómo se realizarán las medidas (p. ej., el tempo de respuesta).
- 6) Acciones necesarias para suspender la prueba (cuando los acontecimientos no previstos lo obliguen).

h) Especificación de los tipos de pruebas y casos de prueba según corresponda para el modelo

i) Datos a usar y resultados esperados descritos de Matrices de pruebas.

1. Determinar los tipos de registros maestros que van a ser utilizados (si se requieren)
2. Determinar los tipos de operaciones que van a ser incluidas en los datos de prueba
3. Preparar papeles de trabajo adecuados
4. Obtener los registros que van a ser procesados con las operaciones de prueba; determinar previamente los resultados para compararlos con los datos reales de salida del procesamiento de prueba
5. Si las operaciones de prueba son corridas con los archivos maestros regulares o con la corrida regular de procesamiento, investigar los efectos que el procesamiento de prueba tendrá sobre la información de salida del sistema.
6. Las aplicaciones pueden tratar datos clasificados o que tengan transacciones económicas o de riesgo especial.
7. Los datos de prueba deben incluir las operaciones que determinan el procesamiento y manejo de las siguientes condiciones generales:
 - i. Condiciones válidas
 - ii. Condiciones fuera de secuencia
 - iii. Condiciones fuera de los límites
 - iv. Procesamiento alternativo que se efectúa como resultado de la comparación de los registros de operaciones con los registros maestros (por ejemplo, comparación de los números de identificación de la operación con el número de identificación del registro maestro)
 - v. Diferencias en unidades de medida
 - vi. Información incompleta, no válida u omitida
 - vii. Validación de datos

- viii. Campos que tienen caracteres numéricos vez de caracteres alfabéticos y a la inversa
- ix. Campos que tienen demasiados caracteres
- x. Condiciones ilógicas en los campos de los datos
- xi. Condiciones en que las claves o importes de las operaciones no coinciden con las claves o importes establecidos en las tablas almacenadas internamente

j) Tipos de pruebas

1. Pruebas unitarias, por cada uno de los módulos del sistema
2. Luego de las pruebas unitarias, van las de integración conformen vayan uniendo dos o más módulos del sistema.
3. Pruebas de caja blanca para constatar la respuesta del sistema ante diversas entradas.
4. Pruebas de caja negra, para verificar las entradas y salidas que emite el sistema.

k) Validación de datos.

Controles de edición y validación de Programas

Los controles de edición y validación de datos de entrada a diseño de prueba a programas se basan en la necesidad de contar con datos coherentes e íntegros, lo que garantizará obtener una salida fiable de un proceso determinado

Seguimiento de formatos establecidos y tipo de datos

El control de formato permite verificar si los datos son apropiados (Numéricos, alfabéticos o alfanuméricos) y corresponden a la longitud específica, es decir que no sean muy pequeño para que se puedan visualizar todo el contenido del campo.

Verificar el tipo de datos que contienen los campos es muy importante así como verificar si estos están comprendidos dentro del rango permitido de valores.

Campo faltante

Los campos sensibles deben ser llenados antes de aceptar alguna transacción en el Sistema. Esta verificación es hecha en la evaluación de los controles, es decir los campos importantes deben estar completos.

Razonabilidad

Los datos no deben exceder de su valor razonable. Así en Horas: 24; Meses: 12; etc.

La razonabilidad también considera la verificación automática de tablas, de códigos, límites mínimos y máximos o la revisión de determinadas condiciones establecidas previamente.

- Para ver cómo se integra con otros sistemas (integración)
 - Para comprobar datos y su integridad con otros datos y sistemas de información (Conversión/Integración de datos)
-
- l) Definir el inventario detallado de los recursos financieros, humanos, tecnológicos y de infraestructura física necesarios para llevar a buen término los procedimientos de pruebas.
 - m) Revisión y aprobación del diseño detallado.
 - n) Cronograma de pruebas, esquema de tiempos (tiempos estimados, hitos, etc).
 - o) Horarios de pruebas.
 - p) Necesidades detalladas del entorno.

Especificar las propiedades necesarias y deseables del ambiente de prueba, incluyendo las características del hardware, el software de sistemas (p. ej. el sistema de operación), cualquier otro software necesario para llevar a cabo las pruebas, así como la colocación específica del software a probar (p. ej. qué módulos se coloca en qué máquinas de una red local) y la configuración del software de apoyo.

- Programas que van a ser probados verificar que los programas son utilizados en el procesamiento de las operaciones de prueba
 - Usuarios (perfiles, acceso al sistema)
 - Creación de datos
- q) Elementos de respaldo
- r) Documentos a entregar (Pantallas de error, entre otros).
- s) Aprobaciones y firmas con nombre y descripción de cargo de los involucrados en los procedimientos de prueba.
- t) Matrices de prueba

Las Matrices de pruebas son planillas que permiten identificar los componentes que serán probados junto con los datos a utilizar, así como las condiciones previas y posteriores a la misma. Una matriz de prueba consta de tres secciones.

- 1) Precondiciones:** En esta sección se registran las: condiciones que deben cumplirse para poder iniciar la pruebas, estas pueden ser datos seleccionados (confirmar que toda la data seleccionada se encuentre de acuerdo a lo requerido a través de la matriz de prueba), por ej: tener creados a los pacientes (nombres, numero de historia clínica), tabla de especialistas, etc. Esto obliga a identificar los datos a crear en el transcurso de la prueba o previo a la prueba.

- 2) Proceso a Evaluar:** Se registran los casos de pruebas que se van a evaluar, valores de la prueba y los factores, características y condiciones (criterios de validación) que debe cumplir o satisfacer dicho caso de prueba.

Generalmente los casos de pruebas se definen en función de las opciones o transacciones que tiene la aplicación probar.

- 3) Post condiciones:** Se registrarán las actividades o acciones de validación que permitan certificar que el caso de prueba que se está evaluando se ha realizado con éxito, por ejemplo: Consultar en opción, que la transacción esté grabada correctamente en el sistema.

Técnicas aplicadas: ANEXO 5C

Análisis

Documentación

Herramientas:

Procesadores de texto

Excel

Fase 5. Ejecución de las pruebas

Objetivo:

Aplicar los procedimientos de prueba especificados en el plan detallado.

Cualquier pieza de software puede sufrir problemas al tiempo de ejecución. Estos problemas, que se mostrarán como fallas, provienen

de defectos introducidos por equivocación humana o de tallas del hardware.

Las fallas causadas por el hardware difieren en su tratamiento según su naturaleza. La alternativa es identificar defectos en el software durante la ejecución de las pruebas.

- Probar la disponibilidad del ambiente requerido
- Constatar el funcionamiento del entorno de prueba

El entorno de prueba debe tener en cuenta aspectos tales como: la base de la complejidad de sus operaciones, sus sistemas, su nivel de interconexión con redes internas o externas, propias o de terceros, su grado de exposición y su dependencia de terceras personas para los productos y servicios basados en la tecnología informática

Verificar que el responsable del departamento de producción haya efectuado la creación del ambiente de prueba necesario, con las bibliotecas, los archivos y programas que se requieren para ejecutar la prueba

Comprobar que el responsable del departamento de seguridad haya efectuado la creación del usuario(s) de prueba, con los perfiles y menús que se requieren para ejecutar la prueba

Comprobar que el responsable del departamento de seguridad haya coordinado con el responsable del departamento de Producción la puesta en ambiente de prueba de los archivos y programas solicitado con los permisos correspondientes en cada librería, objetos y archivos.

Verificar que el responsable del departamento de Producción en caso que haya sido requerido haya efectuado la generación masiva de datos de prueba

- Ejecutar las pruebas siguiendo el procedimiento de prueba detallado.
- Capturar las pantallas de ejecución principalmente aquellas con defectos, faltas o errores, teniendo en cuenta que estos términos que coloquialmente solemos manejar como sinónimos, difieren un poco, así:

Error (error):

Es una equivocación cometida por un desarrollador.

Defecto (fault, defect):

Un error puede de desarrollo puede conducir a defectos mayores en módulos y en integración del sistema. Por tanto se debe cuidar cada aspecto por mas pequeño que sea.

Falla (failure):

Una falla es la discrepancia visible que se produce al ejecutar un programa con un defecto, es el fracaso de lo esperado.

Respaldo de la Data:

Coordinar el respaldo inmediato de la data para su utilización en otra prueba (prueba de interrelación)

La meta del proceso de prueba es ejecutar un código con datos de entrada que obligan a los defectos a manifestarse como fallas.

Técnicas aplicadas:

Análisis

Documentación

Verificación

Inspección

Herramientas:

Procesadores de texto

Visualizadores de imágenes

Capturador de pantalla

Herramientas de prueba

Fase 6. Documentación de la ejecución de las pruebas

Objetivo:

Es importante mantener documentación soporte de las pruebas realizadas. La empresa debe conservar documentación escrita y organizada para respaldar cada una de las etapas del proceso de pruebas. Esta documentación suministra un rastro de auditoría y debe facilitar la rectificación de los problemas cuando se presentasen. La documentación deberá incluir, cuando menos, los siguientes aspectos:

- Plan general de prueba **ANEXO 5D**
- Plan detallado de prueba con sus procedimientos **ANEXO 5D** Los tipos de pruebas que se llevaron a cabo, matrices de prueba y los casos de prueba aplicados **ANEXO 5D**

- Los criterios utilizados para seleccionar las operaciones y datos La planificación para la corrección y nuevas pruebas de los sistemas informáticos
- Las personas responsables de la autorización de la planificación de pruebas y de la aceptación de los resultados de las pruebas
- Resultados de las pruebas
- Documentar las novedades presentadas en la prueba
- Firma de los responsables de ejecución de las pruebas (Programadores, analistas O&M)

- **Previo a la prueba**

- Hora de inicio de la prueba
- Ambiente de la prueba
- Estado de la data
- Librería

- **En la prueba**

- Errores encontrados, pasos que se siguieron para llegar al error, data utilizada.
- Impresión de los errores (pantallas, reportes, formularios)
- Impresión de los resultados positivos (pantallas, reportes, formularios)
- Calificar la severidad del error
- Seguimiento a las correcciones y nuevas pruebas
- Otra documentación que apoye sus decisiones y conclusiones histórico de pruebas (Documenta todos los hechos relevantes ocurridos durante la ejecución de las pruebas)
- Informe de resumen de las pruebas realizadas (Resume los resultados de las actividades de prueba (las reseñadas en el propio informe) y aporta una evaluación del software basada en dichos resultados)

El informe de resumen debe contener criterios tal como, se muestra en el **ANEXO 5D**

Para cada criterio definido se deben haber desarrollado las pruebas respectivas.

Técnicas aplicadas: ANEXO 5C

Análisis

Documentación

Herramientas:

Procesadores de texto

Microsoft Excel

Visualizador de imágenes

Capturador de pantallas

Fase 7. Análisis de defectos y/o fallos.

Objetivo:

Localizar, analizar y corregir los defectos que se presentaron en la ejecución de las pruebas

En esta fase se debe encontrar la causa de defectos y/o fallos que se hayan producido, analizarlos y corregirlos en forma oportuna, sin dilatar más la duración de las pruebas. Esto es responsabilidad del programador. El analista de pruebas es el encargado de la ejecución de las pruebas y el que entregará el informe resumido de defectos y/o fallos al analista programador para las respectivas correcciones.

En caso de errores se deberá volver a ejecutar la misma hasta que los defectos sean cero.

Cada vez, que el programa sea cambiado, los datos utilizados en la prueba fallida deben ser nuevamente considerados como datos de prueba de entrada y deberán ser comparados con los datos originales de salida de prueba. Este procedimiento ayuda a eliminar los errores en los programas que algunas veces afectan de forma inesperada a los mismos.

Debemos definir 2 etapas:

- Localización del error (mayor esfuerzo)
- Corrección del defecto

Las preguntas que el programador se debe hacer son:

- ¿Cuándo se cometió?
- ¿Quién lo hizo?
- ¿Qué se hizo mal?
- ¿Cómo se podría haber prevenido?
- ¿Por qué no se detectó antes?
- ¿Cómo se podría haber detectado antes?
- ¿Cómo se encontró el error?

La depuración aparece como resultado de una prueba efectiva, es decir, cuando en un caso de prueba se encuentra un error, la depuración es el proceso que resulta en la eliminación de un error. La depuración siempre se da como consecuencia de la prueba.

La categoría de depuración “eliminación de causas” se manifiesta mediante inducción o deducción. Los datos relacionados con la ocurrencia del error se organizan para llegar a las posibles causas; se desarrolla una lista de las causas y se llevan a cabo las pruebas para eliminar cada una.

La depuración una vez concluida genera un nuevo ciclo de pruebas, el programa no será aceptado hasta que su depuración sea total y el programa sea aceptado.

Entradas:

Programas no aceptados. **ANEXO 5C**

Salidas:

Programas aceptados

Informe de defectos/fallos y/o errores

Técnicas aplicadas: ANEXO 5C

Análisis

Programación

Herramientas:

Lenguajes de programación

Otros vinculantes

Fase 8. Pruebas con los Usuarios

Objetivo:

Ejecutar pruebas con los usuarios una vez que los programas cumplen todos los requerimientos y funcionan correctamente

Esta función debe empezar a planearse como la forma en la que se hará la educación a los usuarios para que cuando el sistema esté listo, todos los usuarios sepan cómo usarlo.

Esta labor de prueba es realizada principalmente por los usuarios en coordinación con Recursos Humanos, Pruebas y Desarrollo.

Esta etapa permite que el usuario vea el adecuado funcionamiento de los programas tal cual fue su requerimiento inicial para su posterior entrega formal de programas por parte del personal de O&M, quien luego de las pruebas llevará un registro de firmas de aceptación en el formulario de requerimiento de usuarios

En esta fase la labor se centra a identificar claramente quién es el receptor/usuario de negocio y de procesos:

- Identificar grupos de usuarios
- Identificar las necesidades de automatización
- Identificar expectativas/metás de los usuarios
- Identificar las restricciones que los perfiles puedan tener

Debe hacerse un levantamiento de los perfiles de usuarios que intervienen en todo el proceso de negocio. Este trabajo se facilita con la intervención de los jefes de área¹ el personal de recursos humanos y de aquellos que conocen los modelos, los cargos y los puestos de todas las áreas integrando dichos perfiles.

Se sugiere elaborar una matriz de perfiles de usuarios que luego facilitará la identificación de sus actividades.

Entradas: ANEXO 5A

Formulario de requerimiento

Salidas: ANEXO 5C

Programas aceptados por los usuarios

Firma de aceptación de usuarios

Técnicas aplicadas:

Análisis

Documentación

Herramientas:

Sistemas en ambiente de prueba

Otros vinculantes

Fase 9. Pruebas en Producción**Objetivo:**

Efectuar evaluaciones de las bibliotecas, programas, archivos, datos puestos en Producción a fin de verificar que se comportan como había sido planeado y que todos los requerimientos funcionan correctamente en el entorno de Producción.

En esta fase se puede incluso efectuar pruebas en producción con datos de prueba o registros de prueba que permitirán corroborar el adecuado funcionamiento del sistema o de los programas, pero éstos deben luego ser eliminados con la finalidad de no alterar los datos reales.

El paso a Producción debe de ser controlado sin descartarse la previsión de contar con un plan de contingencia en caso de fallas de cualquier índole

para así poder regresar hacia atrás y evitar fallas trascendentales en los sistemas y los datos.

También se puede solicitar la revisión de los programas en Producción por parte de técnicos de la misma empresa o independientes a fin de determinar la ausencia de caballos de troya, bombas lógicas y similares, que puedan llegar a alterar la calidad, factor fundamental en los sistemas.

Esta labor de prueba es realizada principalmente por personal experto y técnico de sistemas.

Entradas: ANEXO 5C

Programas aceptados

Salidas: ANEXO 5D

Programas operativos

Informe de prueba

Técnicas aplicadas:

Análisis

Herramientas:

Sistemas en Producción

4.5.3 Aplicación de la metodología establecida en el caso práctico.

Antecedentes

La empresa XYX considera necesario la creación del módulo de Activos Fijos que requiere de modificaciones solicitadas por el Departamento de Contabilidad para optimizar los ingresos de información y el manejo del proceso de revisión y aplicación dentro del módulo además, minimizar o racionalizar el proceso de elaboración del diario contable de las compras y donaciones de Activos Fijos, así como controlar y validar el ingreso de los datos referentes a creación de activos fijos de tal manera que lleguen a contabilidad sin errores.

FASES DE LA METODOLOGÍA DE PRUEBA

Fase 1. Investigación Preliminar

El módulo de activos fijos se diseñó por qué surgió la necesidad de registrar la operación deseada mediante un asistente especial (ej. una compra o una venta), el sistema interpreta la información y hace en forma automática todos los asientos necesarios en la contabilidad y en el módulo de activos fijos.

Por ejemplo la baja de un activo únicamente requiere que se especifique el identificador del activo y en segundos el sistema dará de baja años de historia del activo en la contabilidad: costo histórico, mayores valores, ajustes por inflación, depreciaciones, ajustes a las depreciaciones y valorizaciones.

Objetivo

- Realizar el seguimiento de información clave, como los costos, la fecha de la puesta en servicio, el propietario y la vida útil de los activos fijos (los equipos y bienes raíces con una vida útil superior a un año)
- Registrar números de modelo o de serie, ubicación, número de identificación de la etiqueta, garantía, e información referente al servicio y al custodio.
- Procesar los costos de la depreciación de acuerdo con las reglas contables seguidas por la organización y expresar el valor de los activos menos la depreciación acumulada

Entradas.

Entre las principales entradas a esta fase, tenemos:

1. Formulario de Requerimiento #134 de descripción del movimiento del asiento contable en altas y bajas. **ANEXO 5A.**
2. Documento de especificaciones de Activos Fijos, donde se precisa la optimización del ingreso de transacciones al sistema de distribución de datos en las pantallas de ingreso y por el ingreso masivo de registros, por lo tanto minimizará el tiempo que se empleará en el proceso operativo y disminuirá el índice de errores.

Salidas:

Narrativa que describe lo que hace el sistema:

El módulo de Activos Fijos permitirá al usuario optimizar el ingreso de información y el manejo del proceso de revisión y aplicación del

sistema, automatizará el proceso de elaboración del diario contable de las compras y donaciones de Activos Fijos, además controlará y validará el ingreso de los datos referentes a la creación de activos fijos de tal manera que lleguen a contabilidad sin errores.

En el análisis efectuado para el nuevo diseño del ingreso de registros de activos fijos, se han estructurado 3 procesos para el mantenimiento de Lotes, aprobación de Lotes, contabilización de Activos Fijos

En el Mantenimiento de Lotes se ha procedido a optimizar las pantallas del Mantenimiento de Activos Fijos, de tal forma que se eliminen los datos redundantes, así como también lograr una mejor ubicación de los campos. Esto va a permitir un ingreso más ágil de la información por parte de los usuarios que registran toda la documentación relacionada para la creación de los Activos Fijos. **ANEXO 5F**

Además, para efecto de disminuir los errores y supervisar el trabajo de los digitadores se ha creado un nuevo proceso que es el de Aprobación de Lotes de Activos Fijos, el cual consiste en facilitar al supervisor de Activos Fijos la revisión de la información, a través de la verificación tanto en pantalla como en reportes de los lotes ingresados por los digitadores.

Una vez que la información ha sido aprobada se asignan los códigos definitivos de Activos Fijos y se graba en sus respectivos archivos.

Como proceso final está la contabilización de los lotes una vez que estos han sido aprobados.

Este proceso consistirá en la creación del lote, el cual va a permitir al usuario final organizar la documentación que recibe para la creación posterior de los activos fijos, esta documentación refiere a facturas locales, liquidación de

importaciones, actas por donaciones, memorandos de las áreas que reciben activos fijos entre otros documentos.

Técnicas aplicadas: ANEXO 5C

Análisis

Documentación

Entrevistas a usuarios

Herramientas:

Procesadores de texto

Fase 2. Conocimiento del sistema

Objetivo

El área Contable de la empresa, representada por el Sr. Jorge Icaza, ha solicitado crear el módulo de Activos Fijos del sistema administrativo ADAM, para realizar un mejor control de los registros de activos fijos y agilizar las transacciones de ingreso de los mismos, así como disminuir el volumen de asientos diarios que tienen que procesarse e imprimirse.

Estos requerimientos se plantearon y definieron en dos reuniones que se tuvo con las siguientes personas: Sr. Pablo Monca y Sra. Maria Cres (Contabilidad); Sr Luís Casto y Srta. Alicia Rivadavia, (Financiero) y Srta. Ana Oga (Pruebas).

Interface de registro de activos fijos

Creación de una interface que permita el registro y control en el proceso de ingreso de los activos fijos, para que luego de su ingreso y revisión se puedan transferir a los archivos pertenecientes al módulo de

Activos Fijos (AF). Esto permitirá validar las transacciones ingresadas y corregirlas, antes de grabarlas en el módulo de Activos Fijos y de esta manera minimizar el índice de errores.

Procesos de la aplicación

- Crear proceso de aplicación de los ingresos validados y revisados, para cargar los registros directamente al módulo de activos fijos.
- Crear proceso de contabilización por lotes, según criterios de selección.

Reportes de control de ingreso de activos fijos

Crear un reporte que muestre las transacciones de activo fijos que hayan sido ingresadas en el día, que incluya el centro d costo, la referencia, tipo de documento (factura, acta, DUI), el usuario que ingresó, el usuario que revisó y el usuario que aplicó. Esto permitirá controlar mejor el ingreso de información al módulo de Activos Fijos.

Control de Validación

Crear un proceso de selección de transacciones para obtener sólo las que se hayan validado, de tal manera que sólo se procesen los activos fijos que hayan sido revisados por el responsable del ingreso.

Creación de una nueva interface de activos fijos, para permitir el ingreso de la información sin afectar directamente a los archivos de Activos Fijos.

Generación de un reporte con el detalle de los items, y de la persona que hizo la validación de los lotes.

Proceso contable

Crear un proceso que permita contabilizar los activos fijos, al finalizar el día, seleccionando rango de lotes o de fechas y generar asientos de diario por dependencia y tipo de activo.

Soporte contable

Crear un reporte resumido de los ingresos de los activos fijos que muestre los totales por dependencia y por tipo de activo.

Entorno

En el proceso operativo actual del módulo Activos Fijos (AF), comprende el ingreso de los activos fijos, verificación de los datos ingresados contra los documentos fuentes, generación de archivo EXCEL para control de la información ingresada, validación y aplicación de los asientos contables automáticos por altas e activos fijos.

Registro de Activos Fijos.

- Cada usuario debe ingresar inmediatamente, máximo hasta dos días, indistintamente, sin ningún orden establecido, facturas con una cantidad variable de ítem. En el caso de las liquidaciones por importaciones u órdenes de compra, la información es preparada en el Dpto. de Contabilidad en una hoja Excel.
- Si los ítem son suministros médicos, la información ingresar es repetitiva, es decir, el detalle no cambia absolutamente en nada, por lo que se vuelve tedioso el ingreso de los mismos.

Emisión de Reportes de Control.

- El reporte de los activos fijos puede generarse de acuerdo a criterios seleccionados por el usuario y los datos de cada activo fijo se imprimen en dos hojas. Esta información está distribuida de la misma manera como aparece en las pantallas de ingreso.

Generación de archivo de datos de activos fijos.

- El responsable del módulo genera un archivo con los datos relevantes de activos fijos para control y contabilización de los mismos, el cual es convertido a Excel para poder revisar las transacciones ingresadas por los usuarios.
- La información de este archivo se filtra de acuerdo a los requerimientos de información del momento.

Validación de transacciones.

- Después de que se ingresaron y contabilizaron los activos fijos, el responsable del módulo compara los totales de la hoja Excel con los datos de los ingresos del mes, y a su vez con la hoja Excel preparada por Contabilidad.
- Si existe algún error, se modifica la transacción correspondiente al activo fijo.

Oportunidades de Negocio

- La modificación a estos procesos del módulo de Activos Fijos, optimizará el ingreso de transacciones al sistema, por las mejoras a la distribución de datos en las pantallas de ingreso y por el ingreso masivo de registros, por lo tanto minimizará el tiempo que se emplea en el proceso operativo y disminuirá el índice de errores.

- Con los nuevos reportes y la modificación a los procesos se podrá validar de mejor manera el ingreso al módulo de activos fijos.

Objetivos de Negocios

Con la modificación de las opciones de ingreso de altas al módulo de Activos Fijos, se logrará:

- Rapidez en el registro de transacciones de facturas de activos fijos, mediante una mejora en la interface de registro de activos fijos, que permitirá adicionalmente disminuir el volumen de asientos contables.
- Mejoras en el control de creación de activos fijos y validación y corrección de datos.
- Disminuir la cantidad de asientos que se necesitan para contabilizar los activos fijos.
- Proporcionar un soporte contable adecuado y ágil para la revisión del usuario de contabilidad.

Requerimientos

Las nuevas opciones del Sistema y sus modificaciones involucran los siguientes requerimientos:

- Creación de la interface de ingreso de activos fijos, que permita ingresar automáticamente un volumen de ítems al mismo tiempo, dependiendo del tipo de activo, siendo óptimo aplicarla a los suministros médicos.
- Creación de reportes.
 1. Creación del reporte detallado de transacciones de activos fijos.
 2. Creación del reporte resumido de transacciones de activos fijos.

- Creación del proceso de contabilización, definiendo criterios de selección, basados en rango de lotes o de fechas.
- Normalizar datos en las tablas de agrupaciones de activos fijos, eliminando información duplicada.
- Crear tablas de relación de campos, que filtre solo la información correspondiente.
- Ingresar masivamente las transacciones de activos fijos, utilizando un mecanismo que agrupe a las mismas y que podría definírsele como lote de ingreso.

Beneficio provisto a los usuarios

- Aumentar la productividad del personal encargado de los registros de activos fijos.
- Minimizar errores humanos de criterio y de digitación.
- Facilitar tareas de análisis con los nuevos reportes.
- Controlar el ingreso de activos fijos.
- Disminuir los asientos diarios de contabilización de activos fijos.
- Disminuir las bajas de activos fijos por mal ingreso.
- Permitirá etiquetar más rápidamente los activos fijos con su código generado.
- Pronta distribución de los activos fijos al lugar solicitado.

Riesgos del Negocio

Se ha detectado como factor de riesgo:

- Que los usuarios no utilicen correctamente el proceso de verificación de la información.
- Perder el detalle de cada activo fijo en la contabilización de los mismos.
- No se ha definido un proceso que reverse las transacciones mal aplicadas.

Entradas:

Documento de especificaciones técnicas del programador

Salidas:

Con esta información se creará:

- 1.- Un diagrama de los elementos del sistema
- 2.- Un documento denominado ***Instructivo del módulo o sistema.***

Técnicas aplicadas: ANEXO 5C

Análisis

Documentación

Entrevistas a programadores (En la interacción con el sistema)

Herramientas:

Procesadores de texto

Microsoft Visio

Fase 3. Planeación de las pruebas

El módulo de Activos Fijos le permite al usuario controlar de manera más fácil, la información monetaria y no monetaria de los activos que forman parte de la empresa. Este módulo le permitirá contar siempre con información sobre la ubicación de los activos, los centros de costos

responsables, el proveedor y toda la información necesaria para el control administrativo de los mismos.

Además, permite calcular la depreciación y reevaluación de cada activo e integrar estos rubros en los movimientos contables asociados que se realizan con dichos activos fijos para registrarlos en la contabilidad general de la empresa.

Objetivo

- Facilitar el análisis de la situación del activo fijo de la empresa al permitir la definición de vidas útiles y tipos de depreciación distintos para la contabilidad fiscal.
- Mantener el saldo contable de ley, permite disponer de información útil para el análisis gerencial y la toma de decisiones.
- Brindar a las empresas información útil para la toma de decisiones al permitir la consulta e impresión de reportes para cualquier período instantáneamente.
- Apoyar la identificación de activos que requieren mantenimiento y apoya la programación tanto del mantenimiento preventivo como correctivo.

Entradas:

Riesgos

- No mantener la base actualizada puede producir un riesgo ya que se muestran datos antiguos que no tendrían beneficio para la empresa.
- El Sistema de Riesgo del módulo de Activo Fijo no permitirá el registro por unidad de medidas de la propiedad además de cálculo de depreciación por los diversos métodos, por empresa, iniciativa, cuentas contables, entre otros criterios. Dicha facilidad propicia el control de los bienes por local, responsabilidad y su naturaleza.

- Otro aspecto que el módulo es que no se podría contempla es el registro de la revaluación del activo y de todo mecanismo de averiguación de depreciación.

Sistema de control interno (políticas, normas y procedimientos)

Criterios de aceptación

El elemento quizás más importante es el criterio de aceptación que se va a emplear para los componentes o módulos del producto. Es importante notar que la Calidad del producto no es lo que se quiere probar, puesto que este debe ser un elemento inmerso en la construcción y desarrollo de toda la solución.

Salidas: ANEXO 5D

Plan de pruebas para modulo de activo fijo.

Se han creado pruebas positivas y negativas, que incluyen los casos de uso, pruebas que busquen errores en el manejo de las restricciones, entre otros aspectos.

Técnicas aplicadas: ANEXO 5C

Análisis

Documentación

Herramientas:

Procesadores de texto

Microsoft Project

Fase 4. Diseño de las pruebas

Objetivo:

Especificamos los detalles necesarios sobre el enfoque general reflejado en el plan e identificar las características detalladas que se deben probar.

Entradas: ANEXO 5D

Esquema de pruebas.

Salidas: ANEXO 5D

Para obtener el documento resultante se debe tener en cuenta lo siguiente:

Tipos de pruebas: Pruebas de caja negra y prueba de integración.

Matrices de prueba: Las Matrices de pruebas son planillas que permiten identificar los componentes que serán probados junto con los datos a utilizar, así como las condiciones previas y posteriores a la misma. Una matriz de prueba consta de tres secciones (Precondiciones, Proceso a Evaluar, Pos condiciones).

Técnicas aplicadas: ANEXO 5C

Análisis

Documentación

Herramientas:

Procesadores de texto

Microsoft Excel

Fase 5. Ejecución de las pruebas

Los recursos están disponibles para realizar todas las pruebas de comprobación del correcto funcionamiento de los componentes del sistema

de información. Se crearon los usuarios de pruebas, el ambiente, las autorizaciones, perfiles y accesos con la participación activa del personal de seguridad informática expertos en el establecimiento de las opciones de seguridad propias del equipo y del personal de Operaciones de la empresa.

La comprobación se realizó de acuerdo con las verificaciones establecidas en el plan de pruebas para el nivel de pruebas unitarias. Para cada verificación establecida se realizan son las pruebas con los casos de prueba asociados, efectuando el correspondiente análisis y evaluación de los resultados y generando un registro conforme a los criterios establecidos en el plan de pruebas.

Una vez que todos los componentes han sido probados se procede a realizar las pruebas de integración, verificando de acuerdo al plan de pruebas, si estos interactúan correctamente a través de sus interfaces, tanto internas como externas.

Para estas pruebas se seleccionan los datos necesarios para elaborar casos de seguimiento con un flujo normal de información y otros con flujos alternos o de excepción.

Finalmente, se realizan las pruebas para comprobar la integración en forma global, verificando el funcionamiento de las interfaces y la cobertura de los requerimientos no funcionales. Es importante mencionar que los requerimientos no funcionales pueden comprometer la aceptación del sistema en caso de que no cumplan con las expectativas fijadas.

Técnicas aplicadas:

Análisis

Documentación

Fase 6. Documentación de la ejecución de las pruebas

Objetivo:

Es necesario archivar las pruebas realizadas con sus respectivos anexos, para que sea analizado en conjunto con los programadores con toda la documentación de soporte de las pruebas físicas y/o lógicas realizadas. Este trabajo consiste en la elaboración de una carpeta física/lógica que contenga toda la información recopilada de todas las fases anteriores, debidamente etiquetada y archivada.

La empresa, debe conservar documentación escrita y organizada para respaldar cada una de las etapas del proceso de pruebas y lo hacen en la carpeta del proyecto sección prueba establecida para el efecto.

Técnicas aplicadas:

Análisis

Documentación

Herramientas:

Procesadores de texto

Microsoft Excel

Fase 7. Análisis de errores/Depuración

Objetivo:

Al realizar las pruebas respectivas en el análisis del sistema de Activos Fijos encontramos errores, los cuales fueron reportados a los programadores quienes hicieron el análisis y las correcciones de los defectos que se presentaron en la ejecución de las pruebas.

La principal causa del error fue una mala relación de las tablas de activo fijo.

- Repetición de los mismos activos a diferentes usuarios

- No presenta todos los activos que le corresponden al usuario
- Existen usuarios que ya no laboran en las dependencias y tienen asignado activos existentes en la empresa
- Existencia de algunos activos en mal estado que se encontraban asignados como buen estado

Entradas: ANEXO 5C

Programas no aceptados

Salidas: ANEXO 5C

Programas aceptados

Técnicas aplicadas:

Análisis

Programación

Herramientas:

Lenguajes de programación

Otros vinculantes

Fase 8. Pruebas Finales aplicadas a usuarios.

Objetivo:

Se realizaron pruebas con los usuarios una vez que los programas cumplieron todos los requerimientos y funcionan correctamente.

Esta labor de prueba fue realizada por el personal de Seguridad, Operaciones, Programadores y personal de pruebas con la coordinación del departamento de Recursos Humanos.

Esta etapa permitió que el usuario vea el adecuado funcionamiento de los programas tal cual fue su requerimiento inicial para su posterior entrega formal de programas por parte del personal de pruebas. Luego de las pruebas se llenó el registro de firmas de aceptación en el formulario de requerimiento de usuarios

Entradas: ANEXO 5A

Formulario de requerimiento

Salidas: ANEXO 5C

Programas aceptados por los usuarios

Técnicas aplicadas:

Análisis

Documentación

Herramientas:

Sistemas en ambiente de prueba

Otros vinculantes

CONCLUSIONES Y RECOMENDACIONES.

A partir del estudio podemos emitir las siguientes conclusiones:

- La mayoría de proyectos de software en la actualidad no incorporan documentación, lo cual constituye una desventaja muy grande en cuanto a alcanzar la calidad de software, a pesar de que los encuestados consideran importante documentar, pudimos conocer que no aplican esto debido a desconocimiento de normas de documentación y desorganización dentro de los grupos de trabajo.
- Los grupos de trabajo están empezando a definir roles entre sus integrantes, esto es importante ya que vemos que están avanzando un paso hacia la organización en los proyectos. Esto va de la mano con el que los encuestados considera muy importante para el éxito del proyecto, el definir roles entre sus integrantes.
- Algo que preocupa es que en los equipos de trabajo, no todos incorporan equipos de pruebas internos, esto nos da a pensar que aún se sigue desarrollando software de manera artesanal.
- En la actualidad los grupos de trabajo consideran importante la aplicación de una metodología de aseguramiento de calidad de

software, para que su producto final satisfaga al usuario, pero no se esta aplicando estándares de calidad en la organización.

- La mayoría de participantes en proyectos, conocen de organizaciones que emiten estándares de calidad, principalmente de la ISO, y de la IEEE. Lo que falta es que se informen mejor sobre estándares de calidad para que puedan aplicarlos en sus proyectos.
- Continuando con los estándares de calidad de software, vemos que el estándar ISO 9126 es aún desconocido por la mayoría de desarrolladores de software.
- La mayoría de grupos de proyectos incorporan una fase de pruebas, pero en su mayoría se dedican a realizar inspecciones tempranas y no aplican inspecciones periódicas y organizadas dentro de un plan de pruebas.

BIBLIOGRAFÍA

- [1] Daniel Chudnovsky, Andrés López, Silvana Melitzko “El sector de software y servicios informáticos (SSI) en la Argentina: Situación Actual y perspectivas de desarrollo” Argentina 2002
- [2] PROSOFT, “Estudio del nivel de madurez y capacidad de procesos de la industria de tecnologías de información”, México 2004.
- [3] AESOFT, “Publicación de la AESOFT sobre la industria del Software en Ecuador”, disponible en <http://www.aesoft.com.ec>, última visita: Septiembre 2006.
- [4] Clases de Ingeniería de Software. Ing. Galo Valverde
Tema: Calidad de Software
- [5] Wikipedia.com, “ISO 9000”, disponible en:
http://es.wikipedia.org/wiki/Normas_ISO_9000 Última visita: Julio 2007
- [6] Monografias.com, “ISO 9001”, disponible en:
<http://www.monografias.com/trabajos6/inso/inso.shtml>
Última visita: Julio 2007

- [7] Wikipedia, "ISO 9001", disponible en
http://es.wikipedia.org/wiki/ISO_9001, última visita: Septiembre 2006.
- [8] tecnomestros, "estándares iso", disponible en:
http://tecnomaestros.awardspace.com/estandares_iso.php
Última visita: Julio 2007
- [9] Clases de Ingeniería de Software. Ing. Galo Valverde
Tema: Modelos de proceso y calidad de Software
Parte acerca de las métricas
- [10] Monografias.com, "Ingeniería del software", disponible en:
[http://www.monografias.com/Ingeniería del software](http://www.monografias.com/Ingeniería%20del%20software)
Última visita: Julio 2007
- [11] Boehm, B. W., Brown, J. R., Kaspar, H., Lipow, M., McLeod, G.,
and Merritt, M., Characteristics of Software Quality, North Holland,
1978.
- [12] Boehm, Barry W., Brown, J. R, and Lipow, M.: Quantitative
evaluation of software quality, International Conference on Software

Engineering, Proceedings of the 2nd international conference on Software engineering, 1976.

- [13] McCall, J. A., Richards, P. K., and Walters, G. F., "Factors in Software Quality", Nat'l Tech. Information Service, no. Vol. 1, 2 and 3, 1977.
- [14] Marciniak, J. J., Encyclopedia of software engineering, 2vol, 2nd ed., Chichester : Wiley, 2002.
- [15] Grady, R. B., Practical software metrics for project management and process improvement, Prentice Hall, 1992.
- [16] Clases de Ingeniería de Software. Ing. Galo Valverde
Tema: Modelos de proceso y calidad de Software
Parte acerca de las Modelo de McCall
- [17] Jacobson, I., Booch, G., and Rumbaugh, J., The Unified Software Development Process, Addison Wesley`Longman, Inc., 1999.

- [18] Kruchten, P., The Rational Unified Process An Introduction - Second Edition, Addison Wesley Longman, Inc., 2000.
- [19] Rational Software Inc., RUP - Rational Unified Process, www.rational.com , 2003.
- [20] Dromey, R. G., "Concerning the Chimera [software quality]", IEEE Software, no. 1, pp. 33-43, 1996.
- [21] Dromey, R. G., "A model for software product quality", IEEE Transactions on Software Engineering, no. 2, pp. 146-163, 1995.
- [22] Wikipedia, "CMMI", disponible en <http://es.wikipedia.org/wiki/CMMI>, última visita: Julio 2007
- [23] Patricia González Herrera, María Lorente Pantoja, Eva Ludeña Pérez-Higuera, Ramón Villahermosa Jiménez, Carmelo Torres Plata, "Marco de evaluación CMMI-SW (por etapas)"
- [24] Manuel de la Villa, Mercedes Ruiz, Isabel Ramos "Modelos de Evaluación y Mejora de procesos: Análisis Comparativo" tabla 1.7 y 1.8

- [25] www.ie.inf.uc3m.es/grupo/Investigacion/LineasInvestigacion/Articulos/spice.doc
- [26] D. Salazar, M. Villavicencio, V. Macías, M. Snoeck, “Estudio Estadístico Exploratorio de las Empresas Desarrolladoras de Software Asentadas en Guayaquil, Quito y Cuenca”
- [27] AESOFT “Primer estudio de la industria de software en el Ecuador”
- [28] Wikipedia, “Encuesta”, disponible en <http://es.wikipedia.org/wiki/Encuesta>, última visita: Julio 2007
- [29] [<http://utemti2009.blogspot.com/2009/07/gane-sarson.html>]
- [30] MinguetMelián, La calidad del software y su medida.
[Juan Manuel Cueva Lovelle, Calidad del Software.](http://www.monografias.com/trabajos59/calidad-software/calidad-software.shtml)
<http://www.monografias.com/trabajos59/calidad-software/calidad-software.shtml>
- [31] [http://www.ecured.cu/index.php/Pruebas de Calidad de Software](http://www.ecured.cu/index.php/Pruebas_de_Calidad_de_Software)
- [32] [http://es.wikipedia.org/wiki/ISO/IEC_9126]

ANEXOS

ENCUESTA APLICADA AL ESTUDIO

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
INSTITUTO DE CIENCIAS MATEMÁTICAS
Guayaquil - Ecuador**

“Diseño de un Sistema de Calidad de Pruebas de Software para una empresa de servicios basado en el Estándar ISO 9126”

1. ACERCA DEL ENTREVISTADO

Nombre:	_____	
Edad:	_____	
Género:	A) Masculino	<input type="checkbox"/>
	B) Femenino	<input type="checkbox"/>
Cargo:	_____	
Email	_____	

2. ACERCA DE LA DOCUMENTACION DEL PROCESO DE DESARROLLO.

2.1 Con que frecuencia realizan documentación del proceso de desarrollo de software?			
	En todo proyecto	<input type="checkbox"/>	Pocos <input type="checkbox"/>
	En algunos	<input type="checkbox"/>	Ninguno <input type="checkbox"/>
2.2 Cuán importante cree Ud. Que es la documentación en el proceso de desarrollo de software?			
	Muy importante	<input type="checkbox"/>	Poco importante <input type="checkbox"/>
	No tiene importancia	<input type="checkbox"/>	
2.3 Por Qué motivo no incorporan documentación al proceso de desarrollo de software:			
A)	Falta de organización de los miembros del equipo		<input type="checkbox"/>
B)	Desconocimiento de como documentar		<input type="checkbox"/>
C)	No le interesa incorporar documentación al proceso		<input type="checkbox"/>
D)	Otro		<input type="checkbox"/>
	Especifique	_____	

3. ACERCA DEL EQUIPO DE DESARROLLO

Cuántas personas integran su equipo:	
3.1	Por favor, indique el número de personas designadas para el desarrollo de un:
3.2	
A) Proyecto pequeño	<input type="checkbox"/>
B) Proyecto mediano	<input type="checkbox"/>
C) Proyecto grande	<input type="checkbox"/>
3.3	Su equipo de desarrollo tiene roles definidos:
A) SI	<input type="checkbox"/>
B) NO	<input type="checkbox"/>
Si escogió la opción B pase a la cuarta parte	
Cuán importante cree Ud. Que es definir roles en el equipo para la organización del proyecto?	
3.4	
A) Muy importante	<input type="checkbox"/>
B) Poco importante	<input type="checkbox"/>
C) No tiene importancia	<input type="checkbox"/>
3.5	Qué roles tiene su equipo de desarrollo?
(Señale con una X todas las que apliquen)	
A) Programador	<input type="checkbox"/>
B) Equipo de pruebas	<input type="checkbox"/>
C) Diseñadores	<input type="checkbox"/>
D) Gerente de proyecto	<input type="checkbox"/>
E) Integradores	<input type="checkbox"/>
F) Otros	<input type="checkbox"/>
	Especifique _____

4. ACERCA DE LA METODOLOGIA Y ESTANDARES DE CALIDAD DE SOFTWARE.

4.1	<i>¿Ud. realiza esfuerzos a informarse sobre la calidad de software:</i>	
	A) SI	<input type="checkbox"/>
	B) NO	<input type="checkbox"/>
4.2	<i>¿Cuán importante cree Ud. que es la aplicación de alguna metodología en la calidad del producto final?</i>	
	A) Muy importante	<input type="checkbox"/>
	B) Poco importante	<input type="checkbox"/>
	C) No tiene importancia	<input type="checkbox"/>
4.3	<i>¿Qué metodología(s) o técnica(s) de aseguramiento de calidad de software conoce?</i>	
	A) CMMI	<input type="checkbox"/>
	B) Paradigma GQM	<input type="checkbox"/>
	C) Modelo de McCall	<input type="checkbox"/>
	D) Modelo de Boehm	<input type="checkbox"/>
	E) Modelo SPICE	<input type="checkbox"/>
	F) Otro	<input type="checkbox"/>
	Especifique	_____

4.4 Cuáles son las ventajas que usted conoce de la(s) metodología(s) escogida(s) anteriormente sobre las demás mencionadas

A) Permite cumplir con los tiempos establecidos

B) Fácil de aplicar a los procesos

C) Logran mayor satisfacción en el cliente

D) Existe un mayor conocimiento sobre esta(s) metodología(s)

E) Otra

Especifique _____

4.5 Conoce ud. Organizaciones que publiquen estándares de calidad?

A) SI

B) NO

Si escogio la opcion B pase a la quinta parte

4.6 Qué Organizaciones que publiquen estándares de calidad conoce?

A) ISO

B) IEEE

C) Otra

Especifique _____

4.7 De la organizaciones escogidas en la pregunta anterior ¿Qué estándares de calidad conoce?

4.8 Conoce Ud. el estándar de calidad ISO 9126?

A) SI

B) NO

4.9 ¿Cuál cree Ud. que es el atributo más importante del estándar ISO 9126?

A) Funcionalidad

B) Confiabilidad

C) Usabilidad

D) Eficiencia

E) Facilidad de mantenimiento

F) Portabilidad

5. ACERCA DE LAS PRUEBAS DE CALIDAD.

5.1	Incorporan una fase de pruebas en las etapas de su desarrollo	
	A) SI	<input type="checkbox"/>
	B) NO	<input type="checkbox"/>
5.2	¿Para el control de los errores Ud. Realiza? (Señale con una X todas las que apliquen)	
	A) Inspecciones tempranas	<input type="checkbox"/>
	B) Análisis de defectos	<input type="checkbox"/>
	C) Revisiones periódicas	<input type="checkbox"/>
	D) Se controlan cuando estos se producen	<input type="checkbox"/>
	E) Otros	<input type="checkbox"/>
	Especifique _____	
5.3	¿Qué tipo de pruebas se realizan? (Señale con una X todas las que apliquen)	
	A) Pruebas Unitarias	<input type="checkbox"/>
	B) Pruebas de integración	<input type="checkbox"/>
	C) Pruebas orientadas a objetos	<input type="checkbox"/>
	D) Pruebas caja negra	<input type="checkbox"/>
	E) Pruebas caja blanca	<input type="checkbox"/>
	F) Otros	<input type="checkbox"/>
	Especifique _____	
5.4	¿Qué porcentaje de defecto encuentran en las pruebas?	
	(Maque con una X una opción)	
	A) [0, 25%)	<input type="checkbox"/>
	B) [25,50%)	<input type="checkbox"/>
	C) [50,75%)	<input type="checkbox"/>
	D) [75,100%)	<input type="checkbox"/>

ESTÁNDARES BASES DE NUESTRO ESTUDIO

ESTÁNDARES BASES DE NUESTRO ESTUDIO:

Estándar de calidad de software	Aplicación a la encuesta	Aplicación al sistema de pruebas	Aplicación al marco teórico del estudio	Metodología	Documentos desarrollados
ISO 9000	X		X		
ISO 9001	X		X		
ISO 29119	X	X	X	X	X
ISO 9126	X	X	X	X	X

Estándares bases de nuestra investigación y su aplicación a la encuesta

Estándar de calidad de software	Para evaluar conocimiento sobre calidad de software	Para evaluar conocimiento sobre fases del software	Para evaluar conocimientos sobre pruebas
ISO 9000	X		
ISO 9001	X		
ISO 29119	X	X	
ISO 9126	X	X	X

Estándares bases de nuestra investigación y su aplicación a la metodología de prueba.

Estándar de calidad de software	En la investigación preliminar	En la fase de diseño del plan	En el plan de pruebas entregado
ISO 9000	X		
ISO 9001	X		
ISO 29119	X	X	X
ISO 9126	X	X	X

Estándares bases de nuestra investigación y su aplicación al marco teórico del estudio.

Estándar de calidad de software	Capítulo 1	Capítulo 2	Capítulo 3
ISO 9000			X
ISO 9001			X
ISO 29119		X	
ISO 9126		X	X

FORMULARIO DE REQUERIMIENTO

Área Solicitante:		EVOLUCIÓN DEL REQUERIMIENTO			
Sistema:		Requerimiento solicitado en:			
Módulo:		Entregado para revisión usuarios: Mayo 28 / 2004			
Función:		Recibido por Sistemas en: Mayo 27 / 2004 – Mayo 27 / 2004 – Mayo 31 / 2004			
Proceso:		Puesto en Producción en:			
Prioridad:		Sustituir requerimiento #			
Aprobado por:					
<input checked="" type="checkbox"/> Cuarto <input type="checkbox"/> Semanal <input type="checkbox"/> Mensual <input type="checkbox"/> Otros		<input type="checkbox"/> Iniciales <input type="checkbox"/> Fecha-hora entrevista <input type="checkbox"/> Entrevista realizada <input type="checkbox"/> Firma			
<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		<input type="checkbox"/> Usuarios Entrevistados <input type="checkbox"/> Iniciales <input type="checkbox"/> Fecha-hora entrevista <input type="checkbox"/> Entrevista realizada <input type="checkbox"/> Firma			
SITUACIÓN ACTUAL					
No	Usuario	Detalle			
REQUERIMIENTO					
No	Usuario	Detalle		Aprobación Inicial Usuario	Cumplido en Fecha
COMENTARIOS					
PRUEBAS					
No	Fecha	Participantes	Firma de aprobación	Observaciones	
CAPACITACIÓN					
Fecha	Participantes		Firma de aprobación	Observaciones	
AFROBACIONES					
Director o Jefe de Área		Sistemas		O&M	Director o Jefe de Área

NARRATIVA RESUMIDA DE ESPECIFICACIONES.

DATOS GENERALES DEL SISTEMA DE ACTIVOS FIJOS.

Gerencia ó Departamento solicitante:

Fecha de elaboración: 25/11/05

Nombre de la Analísis O&M:

DEFINICIÓN DEL SISTEMA

Objetivos y alcances del sistema:

Beneficios esperados:

Descripción del proceso actual:

Requerimientos funcionales del sistema

Prioridades a aplicar:

RESULTADO DE LA SOLICITUD

Fecha de Recepción (DD/MM/AA):

Aprobado

Fecha de Aprobado:

Rechazado

Fecha de Rechazo:

Justificación del rechazo:

Fecha de Requerimiento:

Vo.Bo. de Supervisor de O&M

Vo.Bo. del Usuario

OBSERVACIONES

TÉCNICAS APLICADAS.

Fases	Técnica	Herramientas
Investigación preliminar	Entrevistas Cuestionarios Documental Narrativas	Word Excel Requisito pro
Conocimiento del sistema	Entrevistas Cuestionarios Documental Narrativas	OS400 Comandos del sistema Visio Bpwin
Planeación	Tormenta de ideas Documental Narrativas	Project Visio BPWIN Word
Diseño	Reqpro Software de gestión de biblioteca(Seguridad)	Reqpro Software de gestión de biblioteca(Seguridad)
Ejecución	Comandos AS400 (Ver lista de comandos)	Comandos AS400 (Ver lista de comandos)
Documentación	Reqpro Word (Plantilla preestablecida)	Reqpro Word (Plantilla preestablecida)
Depuración	Comandos AS400 (Ver lista de comandos)	Comandos AS400 (Ver lista de comandos)
Pruebas finales		

DATOS GENERALES DEL SISTEMA DE ACTIVOS FIJOS.

DATOS GENERALES DEL SISTEMA DE ACTIVOS FIJOS.

Gerencia ó Departamento solicitante:

Fecha de elaboración: 25/11/05

Nombre de la Analista O&M:

DEFINICIÓN DEL SISTEMA

Objetivos y alcances del sistema:

Beneficios esperados:

Descripción del proceso actual:

Requerimientos funcionales del sistema

Prioridades a aplicar:

RESULTADO DE LA SOLICITUD

Fecha de Recepción (DD/MM/AA):

Aprobado

Fecha de Aprobado:

Rechazado

Fecha de Rechazo:

Justificación del rechazo:

Fecha de Requerimiento:

Vo.Bo. de Supervisor de O&M

Vo.Bo. del Usuario

OBSERVACIONES

PROCEDIMIENTOS Y ESTÁNDARES PARA EL DISEÑO Y DESARROLLO DE PRUEBAS

HISTÓRICO DE CAMBIOS

Fecha	Versión	Descripción	Autor
14/05/2012	1.0	Primera versión	DH

ÍNDICE

✓ Procesos de la organización.....	113
✓ Procesos de Gestión.....	113
✓ Procesos Fundamentales.....	113

En el nivel superior se encuentran los procesos de la organización, que no son específicos de un determinado proyecto de pruebas. Permiten definir las políticas y estrategias aplicables a toda la organización o a una línea de proyectos. 113

Para un proyecto de pruebas se tienen los niveles de procesos de gestión y procesos “fundamentales”. Se define un conjunto de procesos de gestión genéricos para permitir flexibilidad y adaptación a diferentes contextos. Estos procesos son la planificación, monitorización y control, y finalización de las pruebas. Los diferentes procesos de gestión se podrán instanciar en uno o en varios proyectos de pruebas dependiendo de la situación. Por ejemplo, en un proyecto simple puede existir solamente un plan de pruebas global (una sola instancia). En un proyecto más complejo puede existir un plan maestro y otros planes subordinados a éste para cubrir. 114

Una revisión y comparativa de modelos de procesos de prueba 114

Diferentes niveles de prueba (por ejemplo), pruebas de integración, sistema o aceptación) o diferentes tipos de prueba. Cada uno de ellos sería una instancia del proceso de gestión. 114

Los procesos fundamentales abarcan los aspectos técnicos de las pruebas: diseño e implementación, puesta a punto del entorno, ejecución de pruebas y la notificación de resultados de las pruebas. También se incluyen variantes de los procesos para contemplar tantas pruebas dinámicas como estáticas. 114

Los siguientes diagramas ilustran las diferentes capas del proceso que se describen el Modelo de Proceso de Pruebas. 115

Pruebas de Software..... 115

Objetivos de las Pruebas de Software 116

Características de las Pruebas de Software 117

Tipos de Pruebas de Software 118

Funcionalidad 118

 Usabilidad 118

 Fiabilidad..... 119

 Rendimiento 119

 Soportabilidad 120

1.1. Alcance y Objetivos	225
1.2. Organización de Documentos DE PRUEBAS DE SISTEMAS	225
6.1. Estándares de Desarrollo del Sistema DE PRUEBAS	232
6.2. Estándares de Documentación.....	232
6.3 Estándares de programa	232
6.4. Estándares de Archivos usados	232
6.5. Estándares de Casos de Pruebas	233
Factores de Calidad DE PRUEBA	238
Criterios de Calidad.....	239
Métricas y Medidas de Calidad	242
- Pruebas Unitarias	249
- Pruebas de integración.....	249
- Pruebas de regresión.....	249
- Pruebas de aceptación	249
4.1 Pruebas Unitarias	252
4.2. Pruebas de integración.....	252
4.3. Pruebas de regresión.....	254
4.4. Pruebas de aceptación	254
7.1. Pruebas de interfaz de usuario (GUI).....	260
7.2. Pruebas de API (Application programming interface).....	260
7.3. Pruebas de la base de datos	261
8.1. DESCRIPCION DE ERRORES.....	262
8.2. CLASIFICACIÓN DE ERRORES POR PRUEBAS	263

1. INTRODUCCIÓN

1.1. Alcance y Objetivos

Este manual documenta los estándares y procedimientos que se usan para llevar a cabo las actividades de diseño y desarrollo de pruebas de sistema.

1.2. Organización de Documentos DE PRUEBAS DE SISTEMAS

Este documento describe los estándares y procedimientos para el diseño y el desarrollo de pruebas de sistema, en las siguientes áreas:

- Procesos de diseño y desarrollo de pruebas de sistemas
- Preparación de entregables y prototipos de soporte
- Documentación del diseño en el repositorio del proyecto
- Pruebas unitarias
- Estándares y procedimientos de desarrollo

2. PROCESOS DE DISEÑO Y DESARROLLO DE PRUEBAS DE SISTEMAS

A continuación se resumen los procesos de diseño y desarrollo que se deben seguir para las pruebas de sistema.

Describir los procesos de diseño y desarrollo a seguir incluyendo la asignación de tareas, las técnicas a usar y los puntos en los que es necesario llevar a cabo inspecciones y revisiones.

Documentar los estándares y procedimientos en forma de lista, haciendo referencia a las tareas y técnicas de la base de conocimiento y anotando cualquier excepción de los procedimientos.

- Fase 1.** *Investigación Preliminar*
- Fase 2.** *Conocimiento del sistema*
- Fase 3.** *Planeación de las pruebas*
- Fase 4.** *Diseño de las pruebas*
- Fase 5.** *Ejecución de las pruebas*
- Fase 6.** *Documentación de la ejecución de las pruebas*
- Fase 7.** *Análisis de defectos y/o fallos.*
- Fase 8.** *Pruebas con los Usuarios*
- Fase 9.** *Pruebas en Producción*

3. PREPARACIÓN DE ENTREGABLES Y PROTOTIPOS DE SOPORTE DE PRUEBAS DE SISTEMA.

Resumen de los estándares y procedimientos que se han de seguir para la preparación de entregables y los prototipos de soporte para las pruebas de sistema.

Fase 1. Investigación Preliminar

Narrativa resumida de especificaciones.

Fase 2. Conocimiento del sistema

Con esta información se creará:

- 1.-Un diagrama de los elementos del sistema
- 2.-Un documento denominado ***Instructivo del módulo o sistema.***

Fase 3. Planeación de las pruebas

Se debe elaborar un plan de pruebas documentado para sus módulos.

Fase 4. Diseño de las pruebas

Plan detallado de pruebas que incluye los procedimientos de pruebas
El procedimiento de prueba detallado debe contener por cada tipo de prueba, lo siguiente:

Fase 5. Ejecución de las pruebas

Verificar que el responsable del departamento de Producción en caso que haya sido requerido haya efectuado la generación masiva de datos de prueba

Fase 6. Documentación de la ejecución de las pruebas

El informe de resumen debe contener criterios tal como, se muestra en el

Fase 7. Análisis de defectos y/o fallos.

Programas aceptados

Informe de defectos/fallos y/o errores

Fase 8. Pruebas con los Usuarios

Programas aceptados por los usuarios

Firma de aceptación de usuarios

Fase 9. Pruebas en Producción

Programas operativos

Informe de prueba

En la presentación de los entregables se deberá seguir los formatos establecidos en la organización relacionados a estilos y formatos para diagramas y especificaciones.

Los diagramas usados deben seguir los símbolos básicos de diagramas de flujo

La herramienta usada en la documentación de diagramas es Visio.

Se deberá mantener toda la documentación asociada a la prueba así como se deberán mantener todos los registros generados, y se deberá anotar cualquier observación, excepción o problema detectado durante la ejecución de las pruebas

4. DOCUMENTACIÓN DEL DISEÑO EN EL REPOSITORIO DE PRUEBAS DE SISTEMA.

Resumen de los estándares y procedimientos que se han de seguir para la documentación del diseño en el repositorio de las pruebas de sistema.

Fase 1. Investigación Preliminar

Entre las principales entradas a esta fase, tenemos:

1. Formulario de Requerimientos.
2. Documento de especificaciones, el cual contendrá los elementos del sistema: consulta, procesos y reportes sugeridos o propuestos, así como los elementos requeridos.

Fase 2. Conocimiento del sistema

Documento de especificaciones técnicas del programador

Fase 3. Planeación de las pruebas

Problemas del negocio o casos

Objetivos

Restricciones (Fechas críticas)

Riesgos

Sistema de control interno (políticas, normas y procedimientos)

Criterios de aceptación

Fase 4. Diseño de las pruebas

Plan general de pruebas.

Plan detallado de pruebas.

Matrices de pruebas.

Fase 5. Ejecución de las pruebas

Probar la disponibilidad del ambiente requerido
Constatar el funcionamiento del entorno de prueba

Fase 6. Documentación de la ejecución de las pruebas

Hora de inicio de la prueba
Ambiente de la prueba
Estado de la data
Librería

Fase 7. Análisis de defectos y/o fallos.

Programas no aceptados.

Fase 8. Pruebas con los Usuarios

Formulario de requerimiento

Fase 9. Pruebas en Producción

Programas aceptados

Se deberán aplicar todos los formatos a seguir para la documentación del diseño de pruebas tanto en formato, estilos y diagramas.

5. PRUEBAS UNITARIAS

Se deberá presentar el diseño de la prueba unitaria, así como el Resumen de los procedimientos y plantillas para la preparación de planes de pruebas unitarias y el proceso de llevarlas a cabo.

Se deberán describir los procedimientos y plantillas usados en la preparación de los planes de pruebas unitarias .

Se deberá documentar los procedimientos haciendo referencia a las tareas y técnicas de la base de conocimiento apropiadas acorde al sistema evaluado y se deberá anotar cualquier excepción de los procedimientos.

En la documentación de las pruebas se deberá incluir las plantillas que hagan referencia a los entregables y plantillas de la base de conocimiento apropiados.

6. PROCEDIMIENTOS Y ESTÁNDARES DE DESARROLLO DE SISTEMA DE PRUEBAS

En esta sección se especifican:

1. Los estándares y procedimientos a seguir para el desarrollo del sistemas de pruebas
2. Los estándares de documentación
3. Los estándares de programa
4. Los estándares de archivos
5. Los estándares de casos de prueba

6.1. Estándares de Desarrollo del Sistema DE PRUEBAS

Estándares de documentación para elementos tales como:

- Convención de nombres para los ficheros (permanentes y temporales), procedimientos, variables, ficheros de prueba...
- Requisitos de bloqueo y buffering para los ficheros
- Estructuras de los directorios (p.ej. dónde se colocan el código fuente, los ficheros objeto, los archivos de prueba...)
- Estándares de anotación (p.ej. comentarios necesarios al comienzo de cada programa incluyendo el histórico de revisión, etc.)

6.2. Estándares de Documentación

- Actualización de la documentación de la carpeta del proyecto para incluir información adicional
- Formato de la documentación necesaria (p.ej. pseudocódigo detallado, esquemas de los programas)
- Nivel de detalle requerido para la documentación del proyecto
- Paquetes entregables requeridos (p.ej. documentación escrita, casos de pruebas)

6.3 Estándares de programa

- Flujo lógico de procedimientos dentro de un programa
- Tamaño máximo y complejidad de los procedimientos
- Separación de las funciones de entrada/salida de las funciones computacionales
- Procedimiento de manejo de errores
- Uso de variables globales vs. variables locales
- Niveles máximos de anidamiento
- Prohibición del uso de constantes en la llamada de un procedimiento

6.4. Estándares de Archivos usados

- Estándares de anotación (p.ej. comentarios requeridos al inicio del archivo)

- Restricciones de longitud y complejidad de archivos
- Manejo de errores

6.5. Estándares de Casos de Pruebas

- Documentación mínima de las pruebas y los casos de pruebas
- Documentación de los casos de prueba, los resultados esperados y los datos de pruebas
- Métodos para ejecutar pruebas
- Métodos para documentar los resultados reales (Aciertos o fallos)
- Métodos para la corrección de errores

GUÍA DE ESPECIFICACIÓN DE REQUISITOS DE CALIDAD DE PRUEBA

HISTÓRICO DE CAMBIOS

Fecha	Versión	Descripción	Autor
20/05/2012	1.0	Primera versión	DH

ÍNDICE

✓ Procesos de la organización.....	113
✓ Procesos de Gestión.....	113
✓ Procesos Fundamentales.....	113
En el nivel superior se encuentran los procesos de la organización, que no son específicos de un determinado proyecto de pruebas. Permiten definir las políticas y estrategias aplicables a toda la organización o a una línea de proyectos.....	113
Para un proyecto de pruebas se tienen los niveles de procesos de gestión y procesos “fundamentales”. Se define un conjunto de procesos de gestión genéricos para permitir flexibilidad y adaptación a diferentes contextos. Estos procesos son la planificación, monitorización y control, y finalización de las pruebas. Los diferentes procesos de gestión se podrán instanciar en uno o en varios proyectos de pruebas dependiendo de la situación. Por ejemplo, en un proyecto simple puede existir solamente un plan de pruebas global (una sola instancia). En un proyecto más complejo puede existir un plan maestro y otros planes subordinados a éste para cubrir.....	114
Una revisión y comparativa de modelos de procesos de prueba	114
Diferentes niveles de prueba (por ejemplo), pruebas de integración, sistema o aceptación) o diferentes tipos de prueba. Cada uno de ellos sería una instancia del proceso de gestión.....	114
Los procesos fundamentales abarcan los aspectos técnicos de las pruebas: diseño e implementación, puesta a punto del entorno, ejecución de pruebas y la notificación de resultados de las pruebas. También se incluyen variantes de los procesos para contemplar tantas pruebas dinámicas como estáticas.....	114
Los siguientes diagramas ilustran las diferentes capas del proceso que se describen el Modelo de Proceso de Pruebas.....	115
Pruebas de Software.....	115
Objetivos de las Pruebas de Software	116
Características de las Pruebas de Software	117
Tipos de Pruebas de Software	118
Funcionalidad	118
Usabilidad	118
Fiabilidad.....	119
Rendimiento	119
Soportabilidad	120

1.1. Alcance y Objetivos	225
1.2. Organización de Documentos DE PRUEBAS DE SISTEMAS	225
6.1. Estándares de Desarrollo del Sistema DE PRUEBAS	232
6.2. Estándares de Documentación.....	232
6.3 Estándares de programa	232
6.4. Estándares de Archivos usados	232
6.5. Estándares de Casos de Pruebas	233
Factores de Calidad DE PRUEBA	238
Criterios de Calidad.....	239
Métricas y Medidas de Calidad	242
- Pruebas Unitarias	249
- Pruebas de integración.....	249
- Pruebas de regresión.....	249
- Pruebas de aceptación	249
4.1 Pruebas Unitarias	252
4.2. Pruebas de integración.....	252
4.3. Pruebas de regresión.....	254
4.4. Pruebas de aceptación	254
7.1. Pruebas de interfaz de usuario (GUI).....	260
7.2. Pruebas de API (Application programming interface).....	260
7.3. Pruebas de la base de datos	261
8.1. DESCRIPCION DE ERRORES.....	262
8.2. CLASIFICACIÓN DE ERRORES POR PRUEBAS	263

Requisitos de Calidad DE PRUEBA

El objetivo de esta guía es proporcionar una definición exacta y medible de Calidad para el sistema tanto a nivel de producto como de proceso. Esta definición está planteada en términos de Factores de Calidad generales, Criterios de Calidad (ISO 9126) específicos para que se midan cada factor y objetivos de calidad cuantificados y Métricas de Calidad.

Los factores de calidad, los criterios de calidad y las métricas de calidad deben ser tratados de forma adecuada desde el inicio del sistema. Antes de desarrollar cada componente del sistema, se deben definir los factores y criterios de calidad para determinar su aplicabilidad. Los factores y criterios que se consideren relevantes deben servir como factores de éxito críticos para el componente.

Factores de Calidad DE PRUEBA

Los factores de calidad son los aspectos percibidos por el usuario de los componentes del sistema que determinan si el sistema cumple con los requisitos de usuario.

La tabla que se muestra a continuación representa los factores de calidad importantes para el sistema. La tabla identifica y describe cada factor, y clasifica los factores que son los factores de éxito críticos del sistema.

La tabla que se muestra es una guía de factores de calidad relevantes a cada sistema. Las descripciones hay que adaptarlas para que tenga sentido en el sistema específico.

El ranking deberá ser definido para cada caso.

Factor	Descripción	Ranking
Factores de producto		
Accesibilidad	La capacidad de que los usuarios autorizados accedan al sistema cuándo y dónde necesiten acceder.	
Correctitud	La medida en la que el sistema satisface las especificaciones y cumple los objetivos del usuario.	
Eficiencia	La medida en la que el sistema realiza la función para la que se creó con un consumo mínimo de recursos.	
Capacidad de expandirse	La facilidad con la que el sistema puede ser modificado a la hora de añadirle funcionalidad. También la capacidad del sistema de procesar mayores volúmenes de datos sin necesidad de fluctuaciones considerables en el rendimiento.	
Integridad	La medida en la que el sistema está protegido contra accesos no autorizados o modificaciones de software o datos.	
Interoperabilidad	La capacidad del sistema de cambiar información con [lista de	

	nombres de otros sistemas], y de usar la información que se intercambia.	
Mantenibilidad/Facilidad de Mantenimiento	La medida en la que los componentes del sistema se pueden mantener durante la vida de uso esperada o frente a un cambio de proceso.	
Portabilidad	La medida en la que un sistema se puede transferir a nuevos entornos operativos, plataformas hardware o sistemas operativos.	
Idoneidad en la presentación	La medida en la que las cualidades estéticas del sistema presentan la imagen deseada.	
Rentabilidad	La capacidad del sistema de impactar de forma positiva en la productividad y rentabilidad del producto.	
Fiabilidad	La capacidad del sistema de realizar la función requerida bajo las condiciones establecidas durante el periodo de tiempo establecido.	
Reusabilidad	La medida en la que los módulos se pueden usar en múltiples aplicaciones.	
Usabilidad	La medida en la que los componentes funcionales entregados del sistema son entendibles y aplicables por los usuarios finales.	
Factores de Proceso		
Rendimiento financiero	La medida en la que el sistema consigue sus objetivos financieros.	
Cronograma	La medida en la que el sistema se entrega en los plazos establecidos y se cumplen los requisitos.	
Efectividad de recursos	La medida en la que se asignan los recursos óptimos al sistema para asegurar la calidad y la entrega a tiempo.	

Criterios de Calidad

Cada factor de calidad se traduce en uno o más criterios de calidad que pueden ser monitoreados a través del ciclo de vida. Estos criterios sirven como entradas al nivel estratégico para el proceso de analizar y diseñar todos los componentes del sistema.

La siguiente tabla presenta los criterios de calidad que definen cada factor de calidad. Esta tabla es una guía que habría que editar incluyendo los factores y criterios relevantes para cada sistema.

Factor	Criterio	Descripción
Factores de producto		
Accesibilidad	Penetración	La medida en la que un sistema es diseminado de forma satisfactoria a la comunidad de usuarios indicada.
	Virtualidad	La medida en la que los usuarios no requieren tener conocimiento de las características físicas, lógicas o topológicas del sistema.
Corrección	Compleitud	El grado en el que el sistema proporciona la implementación completa de las funciones requeridas.
	Consistencia	El grado en el que el sistema proporciona técnicas y notaciones de diseño e implementación uniformes.
	Exactitud	El grado en el que el sistema proporciona la precisión requerida en cálculos y salidas.
Eficiencia	Efectividad	El grado en el que el sistema utiliza los recursos mínimos al realizar sus funciones.
Capacidad de expansión	Capacidad de aumentar	El grado en el que el sistema proporciona expansión de capacidad, de funciones o datos.
	Modularidad	El grado en el que el sistema proporciona módulos altamente cohesivos con un acoplamiento óptimo.
	Simplicidad	El grado en que el sistema define e implementa sus funciones de la forma menos compleja y más entendible posible.
Integridad	Seguridad	La medida en la que el sistema asigna y utiliza tipos y niveles de seguridad necesarios para controlar el acceso.
	Capacidad de ser auditado	La medida en la que el sistema mantiene un seguimiento de las modificaciones incluyendo los usuarios responsables de funciones y datos.
Interoperabilidad	Estandarización	La medida en la que el sistema utiliza estándares de interfaz para protocolos, rutinas y representaciones de datos.

	Independencia	El grado en el que el sistema no es dependiente del entorno software.
Mantenibilidad/Facilidad de Mantenimiento	Conciso	El grado en el que el sistema implementa sus funciones con una cantidad mínima de código.
	Consistencia	El grado en el que el sistema proporciona técnicas y notaciones de diseño e implementación uniformes.
	Modularidad	El grado en el que el sistema proporciona módulos altamente cohesivos con un acoplamiento óptimo.
	Simplicidad	El grado en que el sistema define e implementa sus funciones de la forma menos compleja y más entendible posible.
Portabilidad	Independencia	El grado en el que el sistema no es dependiente del entorno software.
	Modularidad	El grado en el que el sistema proporciona módulos altamente cohesivos con un acoplamiento óptimo.
	Estandarización	El grado en el que el sistema cumple con estándares de la industria que maximizan la portabilidad.
Idoneidad en la presentación	Imagen	El grado en el que el sistema proporciona una identidad consistente y atractiva que es entendida por todos los usuarios.
Rentabilidad	Productividad	La medida en la que el sistema demuestra una mejora en la productividad de los que lo usan.
	Asequible	El grado en el que los usuarios potenciales se pueden permitir el uso del sistema.
	Coste vs. beneficio	El grado en el que los beneficios tienen más peso que el costo.
Fiabilidad	Gestión de anomalías	La medida en la que el sistema puede proporcionar la continuidad de operaciones bajo condiciones anómalas y la posibilidad de recuperación de estas condiciones.
	Exactitud	El grado en el que el sistema proporciona la precisión requerida en cálculos y salidas.
Reusabilidad	Independencia de aplicación	La medida en la que el sistema es independiente de la aplicación software.
Usabilidad	Simplicidad	El grado en que el sistema define e implementa sus funciones de la forma menos compleja y más

		entendible posible.
	Virtualidad	La medida en la que los usuarios no requieren tener conocimiento de las características físicas, lógicas o topológicas del sistema.
	Formación	La medida en la que el sistema proporciona familiarización de sus funciones y operaciones a sus usuarios.
Factores de proceso		
Rendimiento financiero	Presupuesto	La medida en la que el sistema se entrega de acuerdo a los costos establecidos y aprobados.
Cronograma	Calendario	La medida en la que el sistema se entrega de acuerdo al calendario establecido.

Métricas y Medidas de Calidad

Esta sección define, para aquellos criterios de calidad que pueden ser cuantificados y medidos, el nivel de calidad requerido en el sistema. Para los criterios que son más subjetivos, esta sección también identifica cómo determinar si se cumplen o no los requisitos de calidad.

Para cumplir con factores, criterios y métricas de calidad, deben estar correctamente definidos, a través del ciclo de vida. La siguiente tabla presenta las métricas de calidad que se pueden medir para determinar si se ha conseguido cada factor de calidad. Esta tabla se ha de editar para incluir todos los factores y criterios de calidad según corresponda al sistema evaluado.

Criterios del Cronograma (incluido como ejemplo)		
Cronograma	Método de medida:	Usar una herramienta de seguimiento de sistema para registrar el esfuerzo real y el estimado del sistema y calcular la varianza.
	Responsabilidad:	Gerente de sistema
	Frecuencia:	Semanalmente
	Objetivo:	Cumplir todas las etapas de la planificación
	Objetivos de mejora:	Reducir la fluctuación semanal entre el esfuerzo real y el estimado.

	Actividades de calidad:	Hacer un informe y seguimiento semanal
--	-------------------------	--

NOTA: La empresa se ha comprometido a desarrollar las métricas integrales acorde al formato establecido.

TÍTULO DEL DOCUMENTO

HISTÓRICO DE CAMBIOS

Fecha	Versión	Descripción	Autor
22/05/2012	1.0	Primera versión	DH

ÍNDICE

✓ Procesos de la organización	113
✓ Procesos de Gestión	113
✓ Procesos Fundamentales	113
En el nivel superior se encuentran los procesos de la organización, que no son específicos de un determinado proyecto de pruebas. Permiten definir las políticas y estrategias aplicables a toda la organización o a una línea de proyectos.....	113
Para un proyecto de pruebas se tienen los niveles de procesos de gestión y procesos “fundamentales”. Se define un conjunto de procesos de gestión genéricos para permitir flexibilidad y adaptación a diferentes contextos. Estos procesos son la planificación, monitorización y control, y finalización de las pruebas. Los diferentes procesos de gestión se podrán instanciar en uno o en varios proyectos de pruebas dependiendo de la situación. Por ejemplo, en un proyecto simple puede existir solamente un plan de pruebas global (una sola instancia). En un proyecto más complejo puede existir un plan maestro y otros planes subordinados a éste para cubrir.	114
Una revisión y comparativa de modelos de procesos de prueba.....	114
Diferentes niveles de prueba (por ejemplo), pruebas de integración, sistema o aceptación) o diferentes tipos de prueba. Cada uno de ellos sería una instancia del proceso de gestión.	114
Los procesos fundamentales abarcan los aspectos técnicos de las pruebas: diseño e implementación, puesta a punto del entorno, ejecución de pruebas y la notificación de resultados de las pruebas. También se incluyen variantes de los procesos para contemplar tantas pruebas dinámicas como estáticas.	114
Los siguientes diagramas ilustran las diferentes capas del proceso que se describen el Modelo de Proceso de Pruebas.....	115
Pruebas de Software	115
Objetivos de las Pruebas de Software.....	116
Características de las Pruebas de Software.....	117
Tipos de Pruebas de Software.....	118
Funcionalidad.....	118
Usabilidad.....	118
Fiabilidad	119
Rendimiento	119

Soportabilidad.....	120
1.1. Alcance y Objetivos.....	225
1.2. Organización de Documentos DE PRUEBAS DE SISTEMAS	225
6.1. Estándares de Desarrollo del Sistema DE PRUEBAS	232
6.2. Estándares de Documentación	232
6.3 Estándares de programa.....	232
6.4. Estándares de Archivos usados.....	232
6.5. Estándares de Casos de Pruebas	233
Factores de Calidad DE PRUEBA.....	238
Criterios de Calidad	239
Métricas y Medidas de Calidad	242
- Pruebas Unitarias.....	249
- Pruebas de integración	249
- Pruebas de regresión	249
- Pruebas de aceptación.....	249
4.1 Pruebas Unitarias.....	252
4.2. Pruebas de integración	252
4.3. Pruebas de regresión	254
4.4. Pruebas de aceptación.....	254
7.1. Pruebas de interfaz de usuario (GUI)	260
7.2. Pruebas de API (Application programming interface)	260
7.3. Pruebas de la base de datos.....	261
8.1. DESCRIPCION DE ERRORES.....	262
8.2. CLASIFICACIÓN DE ERRORES POR PRUEBAS.....	263

1. Objetivos

El objetivo de este documento es describir las políticas de pruebas de software aplicables a cualquier proyecto de desarrollo del software dentro de la organización.

El propósito es proporcionar un esquema de trabajo estandarizado de las actividades de prueba del software.

Es necesario un proceso de pruebas estándar para mejorar la efectividad y eficiencia de las pruebas de todo proyecto dentro de la organización. Este proceso puede llevarse a cabo de acorde a las siguientes fases:

- Definir un proceso de pruebas.
- Crear un proceso de pruebas.
- Asegurarse de que los componentes de riesgo alto del sistema son probados.
- Proporcionar métricas de gestión y control de las pruebas.
- Proporcionar métricas de validación y mejora de las pruebas.
- Proporcionar una base para la automatización de pruebas.
- Proporcionar entregables de prueba específicos.

2. PRUEBAS DE SOFTWARE

Las pruebas de software que deben desarrollarse dentro de la organización a fin de garantizar un grado de calidad mínimo son:

- Pruebas Unitarias
- Pruebas de integración
- Pruebas de regresión
- Pruebas de aceptación

Glosario:

Verificación y validación

La verificación y la validación son actividades de pruebas de software que enfatizan la calidad del software que se está desarrollando. Estas actividades son planificadas y dirigidas a lo largo del ciclo de vida del software.

Las actividades relacionadas con la verificación tratan de comprobar que los elementos del software son consistentes con la especificación asociada. Las pruebas del software son un tipo de verificación que usan técnicas como las revisiones, análisis, inspecciones y walkthroughs para llevar a cabo su propósito.

Por otro lado, la validación es un proceso que comprueba que lo que se ha especificado es lo que el usuario realmente quería. Las actividades de validación pueden empezar cuando la mayoría de las funcionalidades del software están desarrolladas. Las pruebas de validación aseguran que el software cumple todas las funcionalidades que debe cumplir, y que su comportamiento y requisitos de rendimiento son los correctos. Normalmente las pruebas de caja negra son utilizadas para realizar estas actividades con éxito.

La verificación trata de responder a la pregunta: ¿Se está desarrollando el producto correcto? Y la validación contesta a la pregunta: ¿Se está desarrollando el producto de la forma correcta?

Defectos vs Pruebas

El término defecto a menudo se usa para referirse a un problema o a un fallo tanto de hardware como de software.

Las pruebas de software no deberían confundirse con la depuración de estos defectos. La depuración de defectos es el proceso de detectar y analizar defectos cuando el software no se comporta de la forma esperada. Aunque la identificación de algunos defectos sería obvia al trabajar con el software, una aproximación metódica a las pruebas de software sería mucho más exhaustiva y eficiente a la hora de detectar defectos. La depuración de defectos es una actividad que puede soportar a las pruebas, pero no reemplazarlas. Sin embargo ninguna prueba puede garantizar que se han encontrado todos los defectos.

3. USUARIOS PARTICIPANTES DE LAS PRUEBAS DE SOFTWARE?

Las personas con las siguientes habilidades son las más adecuadas para llevar a cabo las pruebas del software:

- Personas con actitud de realizar pruebas con la intención de encontrar la mayor cantidad de defectos posibles.
- Habilidad para entender el punto de vista de los clientes.
- Personas con conocimientos de calidad.
- Personas que prestan mucha atención a los pequeños detalles.
- Personas con experiencia en procesos de desarrollo del software.
- Personas que promuevan una atmósfera positiva con el equipo.
- Personas con habilidades diplomáticas que promuevan mejoras en los procesos de Calidad.
- Con habilidad para superar las presiones y que a pesar de las mismas sepa gestionar de forma correcta los procesos de Control de Calidad de la forma adecuada.
- Buena comunicación e interactividad con clientes, responsables, ingenieros, personas técnicas y no técnicas.
- Habilidades para generar una buena documentación.

4. POLITICAS DE PRUEBAS

4.1 Pruebas Unitarias

Antes de realizar pruebas sobre el código del producto se debe revisar toda la documentación asociada y relacionada a los requerimientos.

Las pruebas unitarias deben asegurar que cada unidad cumple las especificaciones antes de integrarlas con otras unidades. Las pruebas unitarias, comprueban que el código de cada unidad puede ser ejecutado sin problemas.

Las pruebas unitarias pueden realizarse de forma aislada al sistema mediante el uso de stubs y drivers que reemplazan el software omitido o ausente y simulan la interconexión entre los componentes del software de una forma simple. En el apartado de integración se detallará el uso de estos componentes.

Las pruebas unitarias deben incluir pruebas de funcionalidad y sobre características no funcionales. Por ejemplo, pueden probar el comportamiento del medio, el rendimiento o la robustez del sistema.

Este tipo de pruebas debe realizarlas el programador que escribió el código, y el usuario que escribió las especificaciones del programa, dependiendo del nivel de riesgo, para conseguir cierta independencia. Los defectos deben ser corregidos tan pronto como se detectan dejando establecido un registro de incidencias.

Es recomendable preparar y automatizar casos de pruebas antes de comenzar la codificación.

4.2. Pruebas de integración

Una vez que las pruebas unitarias se han desarrollado, esta fase requiere unirlos para crear el sistema. Se deben efectuar pruebas de integración graduales partiendo de un pequeño número de piezas, o partes del programa hasta su integración total.

Las pruebas de integración deben descubrir los defectos de las interconexiones y de la interacción entre sistemas o componentes integrados. Debe establecerse más de un nivel de integración y las pruebas pueden realizarse sobre objetos de tamaño variable.

- Las pruebas de integración de componentes prueban la interconexión entre componentes de software, y deben ser ejecutados únicamente después de las pruebas unitarias.

- Las pruebas de integración de sistemas deben probar la interconexión entre distintos sistemas y deben ser ejecutados después de las pruebas de sistema de cada sistema individual.

Antes de realizar cualquier prueba de integración, es necesario establecer una estrategia y decidir cómo unir las distintas partes de un sistema. Hay tres estrategias que se suelen usar comúnmente:

- **Integración Big-bang**, en la que todas las unidades se unen en una sola resultando un sistema completo. Esta técnica tiene la ventaja de que no es necesario simular ninguna parte porque todo está acabado antes de empezar las pruebas de integración.
- **Integración top-down**, donde el sistema se construye por fases empezando por los componentes que llaman a otros componentes. Este tipo de integración permitirá a los técnicos de pruebas evaluar la conexión entre los componentes empezando de arriba hacia abajo.

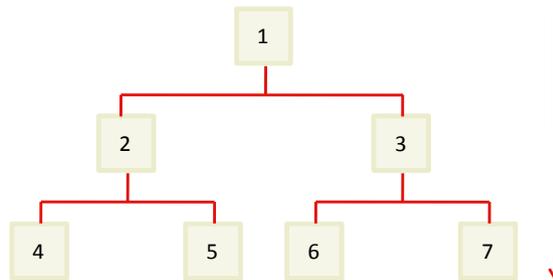


Figura 1 Estructura de control top-down

- **Integración bottom-up**, es lo contrario a la integración top-down y los componentes son integrados en el orden 'de abajo hacia arriba' como muestra la siguiente figura:

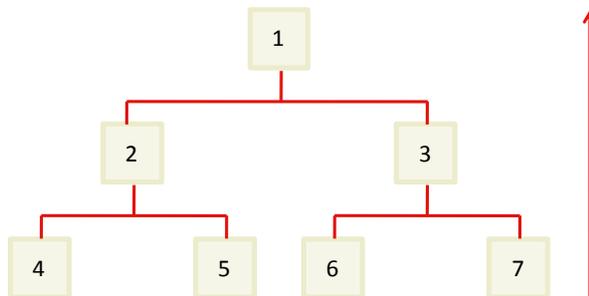


Figura 2 Estructura de control bottom-up

La secuencia y los números de pasos requeridos en la integración dependen de la ubicación de las interconexiones de alto riesgo dentro de la arquitectura del sistema evaluado.

Las personas que lleven a cabo las pruebas deben entender la arquitectura y realizar una buena planificación de la prueba.

4.3. Pruebas de regresión

Al producirse todo cambio en el software o cuando se añade un nuevo módulo, o cuando se crea un nuevo flujo de datos o nuevos controles lógicos genera la aplicación de las pruebas de regresión.

Las pruebas de regresión deben garantizar de manera fiable que las modificaciones no hayan introducido nuevos errores en el código. El personal de Producción debe validar si las modificaciones eliminaron de forma satisfactoria los errores encontrados. Cada vez que el código sea modificado o usado en un nuevo entorno de desarrollo, las pruebas de regresión deben ser usadas para validar la integridad del código.

Las pruebas de regresión sólo deben ser usadas en las pruebas de mantenimiento, para programas existentes o cuando se añaden nuevos módulos al sistema en desarrollo.

4.4. Pruebas de aceptación

Cuando la organización ha realizado sus pruebas de sistemas y ha corregido la mayoría de sus defectos, el sistema será entregado al usuario o al cliente para que le den su aprobación.

Las pruebas de aceptación son, responsabilidad del usuario o del cliente, cualquier persona involucrada en el negocio podría realizarlas únicamente mediante autorización del líder del proceso automatizado. La ejecución de las pruebas de aceptación debe ser realizada en un entorno de pruebas que represente el entorno de producción.

Las pruebas de aceptación tratan de crear confianza en el sistema, en alguna parte del mismo o en sus características no funcionales.

4.5. PRUEBAS DE CAJA BLANCA

Las pruebas de caja blanca obligan al diseño de casos de prueba. Las pruebas deben ser realizadas a partir del conocimiento de la estructura e implementación del software. El analista de pruebas debe estudiar el sistema o parte de él de forma previa. El diseño de los casos de prueba debe enfocarse en la estructura del componente o sistema. Estas pruebas deben analizar la estructura interna del componente/sistema y crear un modelo de pseudo-código a partir del código real.

El método de diseño de casos de pruebas debe considerar entre otros los siguientes casos:

- garantizar que todos los caminos independientes dentro de un módulo han sido ejecutados y probados por lo menos una vez.
- Ejecutar todas las decisiones lógicas.
- Ejecutar todas las iteraciones de las fronteras.
- Ejecutar estructuras de datos internas para asegurar su validez.

4.6. PRUEBAS DE CAJA NEGRA

Estas pruebas deben obtener los casos de pruebas directamente de las especificaciones o requerimientos. El punto más importante sobre las técnicas basadas en la especificación es que las especificaciones o modelos no definen (y no deberían definir) cómo ha de conseguir el sistema el comportamiento especificado cuando se construya; es una especificación del comportamiento requerido (o al menos deseado).

El analista de pruebas deberá estudiar el manual de requerimientos o los requerimientos del sistema antes de ejecutar esta prueba.

Si deben establecer casos de pruebas de tal forma que comprueben que el programa o componente tiene realmente el comportamiento deseado.

Estas pruebas deben incluir tanto pruebas funcionales como no funcionales. Las pruebas funcionales están relacionadas con lo que hace el sistema, sus características y funciones. Las pruebas no funcionales están relacionadas con examinar cuán bien hace algo el sistema, más que examinar qué hace. Los aspectos no funcionales (también conocidos como características o atributos de calidad) deben incluir rendimiento, usabilidad, portabilidad, mantenibilidad, etc. (ISO 9126)

5. AUTOMATIZACIÓN DE PRUEBAS

La organización debe usar la automatización cuando sea necesaria.

Algunas de sus ventajas son:

- ✓ Reducir drásticamente el esfuerzo de las pruebas de regresión.
- ✓ Cuando sea necesario se deben realizar validaciones durante los ciclos de cambios, cosa que sería imposible hacer manualmente debido a restricciones de tiempo.
- ✓ Cuando se requiere validar la consistencia y cobertura lógica.

Para proyectos pequeños, no merece la pena el tiempo necesario para aprender e implementar las herramientas de pruebas. Sin embargo para grandes proyectos es necesario.

Entre las herramientas automatizadas que se autorice su uso se describen las siguientes:

- Analizadores de código, pues ayudan a monitorizar y controlar la complejidad del código, el cumplimiento de estándares.
- Analizadores de cobertura.- Validan qué caminos/partes del código han sido probadas.
- Analizadores de memoria.
- Herramientas de pruebas de rendimiento o de carga.- Estas pruebas se encargan de determinar el comportamiento del sistema bajo diferentes cargas de trabajo.
- Herramientas de pruebas web, que validan si los links son válidos, si el uso de código html es correcto
- Otras herramientas para gestionar los casos de prueba, la documentación, los defectos y la configuración del sistema.

6. ESTRATEGIAS DE PRUEBAS

Las pruebas deben ser actividades planificadas y dirigidas de forma sistemática. Todas las pruebas deben empezar probando pequeñas unidades/módulos e ir avanzando hacia las pruebas de integración y de sistemas, asegurándose de que los defectos, si los hubiese, hayan sido capturados y corregidos tan pronto como sea posible.

Existen muchas estrategias de pruebas. Entre las que se describen:

- **Analítica:** Las estrategias de pruebas analíticas tienen en común el uso de alguna técnica analítica formal o informal normalmente durante la etapa de gestión de requisitos y la de diseño del proyecto.
- **Basado en modelo:** Se pueden crear pruebas basadas en modelos (matemáticos u otros). Si el comportamiento del sistema bajo pruebas se ajusta a lo que se predijo en el modelo, entonces se estima que el sistema está trabajando correctamente. Las estrategias de pruebas basadas en el modelo tienen en común la creación o selección de algún modelo formal o informal para comportamientos de sistemas crítico, normalmente durante las etapas de requisitos y diseño del proyecto.
- **Metódico:** Se debe tener una lista de control que sugiera las áreas principales de ejecución de pruebas o se puede hacer un seguimiento de los estándares de la industria de calidad del software y así tener un resumen de las principales áreas de pruebas. Después, de forma metodológica, se ha de diseñar, implementar y ejecutar las pruebas siguiendo este extracto.
- **Proceso o estándar conformista:** Estas estrategias siguen un desarrollo externo orientado a pruebas a menudo con poca personalización y puede tener un temprano o tardío punto de implicación a pruebas.
- **Dinámico:** Las estrategias dinámicas, como pruebas exploratorias, tienen en común que se concentran en encontrar tantos defectos como sea posible durante la ejecución de pruebas y adaptan todo lo posible el sistema bajo prueba a las condiciones que habrá cuando se libere. Típicamente enfatizan las últimas etapas de pruebas.
- **Consultivo:** Estas estrategias tienen en común la confianza en un grupo de gente que no son técnicos de pruebas para guiar o realizar el esfuerzo de pruebas.
- **Regresión:** Estas estrategias tienen en común el conjunto de procedimientos, normalmente automatizados, que les permiten detectar

defectos de regresión. Esta estrategia suele envolver pruebas funcionales automatizadas antes de la liberación del producto, pero algunas veces se centra por completo en funciones liberadas con anterioridad. Por ejemplo, se puede intentar automatizar todas las pruebas de sistemas de tal forma que cuando se produzca cualquier cambio se pueda volver a ejecutar cada prueba para asegurarse de que ningún área se ha visto afectada.

7. PRUEBAS DE SOFTWARE BASADAS EN LA TECNOLOGÍA

7.1. Pruebas de interfaz de usuario (GUI)

Este tipo de pruebas deben identificar errores de los siguientes tipos:

- Errores de comunicación. Se prueban aspectos como la barra de herramientas o tooltips.
- Falta de información como por ejemplo documentos de formación, mensajes de error, información incorrecta, barra de botones deshabilitada.
- Errores relacionados con acciones del ratón o del teclado, botones por defecto, iconos, escasez de memoria, iconos.
- Estructura de los comandos como los menús, parámetros de comandos, transición de estados, menús popup
- Rigidez del programa en cuanto a las salidas, control u opciones de usuario.
- Preferencias visuales, facilidad de adaptación al usuario o de localización.
- Errores de usabilidad relacionados con la accesibilidad, la eficiencia, la facilidad de uso, la comprensibilidad.
- Localización del sistema teniendo en cuenta temas como la fecha y hora del lugar, dependencias del inglés, traducción, constantes, dependencias culturales

7.2. Pruebas de API (Application programming interface)

Para probar la API existen ciertas tareas que pueden automatizarse como por ejemplo:

- Probar el código
- Probar paquetes de ejecución
- Generación de informes
- Publicación de informes
- Notificación de informes
- Ejecución periódica de paquetes de pruebas

7.3. Pruebas de la base de datos

Al probar la base de datos hay que tener en cuenta aspectos tales como:

- Consideraciones tales como el entorno distribuido, el rendimiento de la base de datos, integración con otros sistemas o aspectos de seguridad.
- Pruebas de integridad relacionadas con cálculos, divisiones por cero, truncamientos, compatibilidad, datos de prueba, consistencia de los datos
- Conectividad de la base de datos: recuperación y almacenamiento de datos.
- Pruebas sobre el esquema de la base de datos: procedimientos de almacenamiento, índices, tablas, campos, restricciones, disparadores
- Pruebas de seguridad: seguridad del usuario y login.

8. PRUEBAS SOBRE ERRORES

8.1. DESCRIPCION DE ERRORES

Descripción de Errores	Detalles	Pruebas
Errores de interfaz de usuario	La interfaz de usuario debería ser consistente con el sistema en cuanto a color, fuentes, tamaño de la fuente, situación de botones, campos, campos de navegación, claridad de la información...	Unitarias/integración
Errores funcionales	La funcionalidad de un programa tiene un error si algo que espera el usuario que haga el programa es confuso e imposible de llevarse a cabo. Este problema es un error de funcionalidad siempre que las expectativas del cliente sean razonables.	Unitarias/integración
Errores de comunicación	¿Hay suficiente información en la pantalla para poder entenderse? ¿La información es útil? ¿Es exacta?	Sistema
Errores de rendimiento	Cualquier cosa que haga pensar al usuario que el programa va despacio es un problema.	Sistema
Errores en la salida/resultados	¿Las salidas del sistema tienen sentido? ¿Los resultados obtenidos pueden almacenarse, redireccionarse o imprimirse?	Sistema
Errores relacionados con la red	Errores al conectar el sistema con una red.	Sistema
Errores en las fronteras	Incapacidad para manejar las fronteras de forma correcta. Cuando el sistema requiere unas entradas siempre comprobar las condiciones frontera.	Unitarias/integración
Errores de cálculo	Errores debidos a fórmulas incorrectas, algoritmos incorrectos...	Unitarias/integración
Errores iniciales	Errores que ocurren solamente la primera o segunda vez que se ejecuta el código.	Unitarias/integración
Errores de flujo de control	Errores debidos a ramas con defectos del flujo de control.	Unitarias y de caja blanca
Errores de manejo de datos	Errores en la transmisión de datos entre distintas rutas.	Unitarias/integración
Problemas en las condiciones	Dos eventos esperando que el otro ocurra. El sistema estará en un bucle infinito.	Sistema
Condiciones de carga	Programas que fallan por un alto volumen de datos, estrés, falta de memoria o de espacio en disco...	Sistema
Manejo del hardware	Errores por excepciones no reconocidas por el hardware, la impresora no tiene papel, espacio en disco...	Sistema

Descripción de Errores	Detalles	Pruebas
Errores de control de versiones	Problemas solucionados que reaparecen en ciertas versiones del producto.	Sistema
Errores de conformidad	Errores debidos a la no conformidad con los estándares o metas del producto.	Sistema

8.2. CLASIFICACIÓN DE ERRORES POR PRUEBAS

Tipo de Pruebas	Detalles	Pruebas
Pruebas de sentencia	Las pruebas unitarias deben incluir los casos de pruebas que prueben todos los caminos del flujo del programa.	Unitarias y de caja blanca
Pruebas de regresión	Referencia punto 4.3	Sistema
Pruebas de rendimiento	Las pruebas de rendimiento necesitan ser planificadas en el diseño de pruebas. Las nuevas versiones que se generen de un producto no deberían tener menos rendimiento que las anteriores y esto requiere cierto mantenimiento.	Sistema
Pruebas de background	En un sistema multiprocesador, cómo de bien el producto realiza muchas tareas. El objetivo es probar que el programa falla cuando maneja más de una tarea.	Sistema
Pruebas de carga	Estas pruebas se encargan de determinar el comportamiento del sistema bajo diferentes cargas de trabajo.	Sistema
Pruebas de estrés	Las pruebas de estrés determinan el punto de ruptura donde el sistema revela el nivel de servicio máximo que puede conseguir	Sistema
Pruebas de almacenamiento	Las pruebas de almacenamiento son otro tipo de pruebas de carga que estudian cómo usa el programa la memoria. Las pruebas de almacenamiento necesitan ser parte del diseño de pruebas cuando sea posible.	Sistema
Pruebas de recuperación de errores	En el caso en que haya errores, las pruebas deberían incluir casos que reproduzcan un mensaje de error por	Unitarias/integración

	lo menos una vez.	
Pruebas de seguridad	Debería probarse el acceso no autorizado. Un error puede ser la entrada al sistema de un usuario no autorizado o que un usuario no tiene suficiente acceso para ser capaz de acceder a los datos con los que debe trabajar.	Sistema
Pruebas de multiusuario	Prueban el comportamiento del sistema ante varios usuarios que trabajan sobre el sistema de forma concurrente. Comprueba la integridad de los datos, las condiciones de la base de datos....	Sistema
Pruebas de configuración	Prueba si los productos trabajan de forma correcta sobre la plataforma y con la configuración especificada.	Sistema
Pruebas de copia de seguridad	Algunos productos pueden necesitar ser probados para asegurar su correcto comportamiento cuando se inician los procedimientos para crear la copia de seguridad.	Sistema
Creación de ejecutables	Se verifica que todos los archivos de código fuente requeridos están incluidos en los ejecutables.	Sistema
Comprobación de la instalación	Verificar que el producto se instala correctamente bajo varios entornos.	Sistema