

UNIVERSIDAD TÉCNICA DE COTOPAXI



**UNIDAD ACADÉMICA DE CIENCIAS DE LA INGENIERÍA Y
APLICADAS**

**TEMA: “Implementación de Pruebas Caja Negra y Caja Blanca aplicables
al Sistema Escolástico del Colegio Nacional “Primero de Abril” de la
Ciudad de Latacunga Provincia de Cotopaxi” en el periodo Octubre 2008 –
Noviembre 2009**

DIRECTOR DE TESIS:

Ing. Franklin Javier Montaluisa

AUTORAS:

Caisa Guayta Carmen Jimena

Semblantes Chicaiza Lilia Vanessa

LATACUNGA – ECUADOR 2010

AUTORÍA

Las autoras certifican que las investigaciones, redacción y propuestas del presente trabajo son de su exclusiva autoría. Caisa Guayta Carmen Jimena y Chicaiza Semblantes Lilia Vanessa.

Caisa Guayta Carmen Jimena

C.I. 050307540-0

Chicaiza Semblantes Lilia Vanessa

C.I.050307999-8

CERTIFICACIÓN

Cumpliendo con lo estipulado en el Capítulo IV, Art.9, Literal f, del Reglamento del Curso Pre profesional de la universidad técnica de Cotopaxi, informo que el grupo de postulantes conformado por las señoritas egresadas: Caisa Guayta Carmen Jimena y Chicaiza Semblantes Lilia Vanesa, han desarrollado su trabajo de Tesis de Grado, de acuerdo al planteamiento formulado en el Plan de Tesis con el tema: **“Implementación de Pruebas Caja Negra, Caja Blanca aplicables al Sistema Escolástico del Colegio Nacional “Primero de Abril” de la Ciudad de Latacunga Provincia de Cotopaxi”** .

En tal virtud de lo mencionado anteriormente, considero que el grupo se encuentra apto para presentarse a la Defensa del Trabajo de Tesis

Latacunga 07 de abril del 2010.

.....

Ing. Franklin Montaluisa

DIRECTOR DE TESIS

CERTIFICACIÓN DE TRADUCCIÓN

Yo, CRISTINA DE LOS ANGELES ZAMBRANO RENDON, portador de la Cédula de Identidad N° 0502242621, en calidad de Docente del Área de Inglés del COLEGIO PARTICULAR JAN AMOS COMENIUS “ tengo a bien **CERTIFICAR:** que los egresados de la Universidad Técnica de Cotopaxi, señoritas:, han realizado la debida corrección con mi persona del Abstrac de la Tesis de Grado con el Tema: **“Implementación de Pruebas Caja Negra, Caja Blanca aplicables al Sistema Escolástico del Colegio Nacional “Primero de Abril”** de la Ciudad de Latacunga Provincia de Cotopaxi” el cual se encuentra bien estructurado, por lo que doy fe del presente trabajo.

Por tal motivo faculto a los peticionarios hacer uso del presente certificado como a bien lo consideren.

.....
CRISTINA ZAMBRANO RENDON

C.I. N° 0502242621

AGRADECIMIENTO

Nuestros más sinceros agradecimientos a Dios, quien con su infinita voluntad nos impulsó al cumplimiento de tan anhelada meta, la de obtener nuestro título de Ingenierías en Informática y Sistemas Computacionales.

A nuestros queridos padres por incentivarnos día a día, por su desvelo y entrega, gracias por ser los principales gestores y forjadores de nuestras vidas profesionales.

Un agradecimiento muy especial al Colegio Nacional Primero De Abril en especial a sus autoridades y directivos por brindarnos toda la contribución necesaria para la realización de nuestro proyecto.

Del mismo modo agradecemos a todas aquellas personas que de una u otra manera colaboraron en el desarrollo del presente proyecto.

DEDICATORIA

Al culminar nuestra educación superior, queremos dedicar el resultado de nuestros años de estudio a las personas que estuvieron en todo momento a lo largo de nuestra trayectoria, siendo los pilares fundamentales para alcanzar uno de tantos sueños, dedicamos a nuestros padres quienes depositaron su confianza en nosotros brindándonos incondicionalmente su apoyo que ha servido para poder demostrar a la sociedad que somos hombres y mujeres de bien en servicio para los demás.

Como olvidar a nuestros compañeros de aula, con quienes formamos una familia y los consideramos unos segundos hermanos con quienes vivimos momentos inolvidables compartiendo momentos buenos y malos durante esta etapa de nuestra vida.

ÍNDICE GENERAL

PORTADA.....	i
AUTORÍA.....	ii
CERTIFICACIÓN.....	iii
CERTIFICACIÓN DE TRADUCCIÓN.....	iv
AGRADECIMIENTO.....	v
DEDICATORIA.....	vi
ÍNDICE GENERAL.....	vii
RESUMEN.....	xvi
SUMMARY.....	xviii
INTRODUCCIÓN.....	1

Capítulo I

1.- Bases teóricas.....	4
1.1 Sistemas de información.....	4
1.1.2 Elementos de un sistema de información.....	4
1.2 Pruebas de software.....	5
1.2.1 Herramienta informática de pruebas.....	7
1.2.2 Pruebas de caja blanca.....	7
1.2.3 Pruebas de caja negra.....	8
1.2.3.1 Limitaciones.....	10
1.2.3.2 Pruebas de integración.....	10
1.2.3.3 Pruebas de aceptación.....	11
1.3 Plan de pruebas.....	12
1.3.1 Aspectos psicológicos y organización del trabajo.....	14
1.4 Que es una arquitectura.....	16
1.5. Que es un cliente.....	16
1.6 Que es un servidor.....	16
1.6.1 Tipos de servidor.....	17

1.6.2 Base de datos.....	18
1.6.2.1 Base de datos distribuida.....	18
1.7 Windows 2000 server.....	19
1.7.1 Windows xp (experience).....	20
1.8 Interfaces de usuario.....	20
1.8.1 Interfaz de usuario de un programa.....	21
1.8.2 Modelos de interfaz.....	22
1.8.3 Diseño de interfaces.....	26
1.8.4 Proceso de diseño de interfaces de usuario	30
1.8.5 Técnicas avanzadas para el diseño de interfaces de usuario.....	31
1.8.5.1 Diseño iterativo de interfaces	32
1.8.5.2 El proceso de diseño y el equipo de desarrollo.....	33
1.8.5.3 Las etapas y tareas del ciclo.....	34
1.8.5.4 Evolución de las interfaces de usuario.....	36
1.9 Software libre.....	39
1.9.1 Linux.....	39
1.10 Que es una norma.....	40
1.10.1 Tipología de normas.....	40
1.10.2 ISO.....	42
1.10.3 Estándares IEEE.....	43
1.11 Modelos de calidad.....	44
1.12 Auditoría.....	44
1.12.1 Auditoría interna y auditoría externa.....	46
1.12.2 Alcance de la auditoría informática.....	48
1.12.3 Control de integridad de registros.....	48
1.12.4 Control de validación de errores.....	48
1.12.5 Características de la auditoría informática.....	48
1.12.6 Síntomas de necesidad de una prueba de eficiencia.....	49
1.12.7 Fallas predecibles a la instalación del software.....	50

1.13 Gestión informática.....	51
1.13.1 Control de entrada de datos.....	52
1.13.2 Planificación y recepción de aplicaciones.....	52
1.13.2 Planificación y recepción de aplicaciones.....	52
1.14 Desarrollo de proyectos o aplicaciones.....	53
1.14.1 Prueba informática de sistemas.....	54
1.15 Sistemas operativos.....	55
1.15.1 Administración de base de datos.....	55
1.15.2 Investigación y desarrollo.....	56
1.15.3 Evaluación informática de comunicaciones y redes que forman parte del sistema.....	56
1.15.4 Seguridad informática.....	56
1.15.5 Herramientas y técnicas para la evaluación informática.....	57
1.16 Gráfica de la métrica de un sistema.....	59

CAPITULO II

2 Trabajo de campo.....	61
2.1 Recopilación de la información de campo.....	61
2.1.2 Presentación	61
2.2 Análisis e interpretación de resultados de las encuestas realizadas a los docentes del colegio nacional primero de abril.....	62
2.3 Análisis e interpretación de resultados de las encuestas realizadas a los administrativos del colegio nacional primero de abril.....	71
2.4 Análisis.....	78

2.5 Conclusiones.....	79
2.6 Recomendaciones.....	80

CAPITULO III

3. Desarrollo del proyecto	81
3.1.Tema	81
3.1.1 Presentación.....	81
3.2. Objetivos.....	83
3.2.1 Objetivo general.....	83
3.2.2 Objetivos específicos.....	83
3.3 Implementación de la evaluacion	84
3.3.1 Plan de pruebas.....	84
3.3.2 Planificación de las pruebas.....	88
3.3.3. Principio básico de las pruebas.....	90
3.3.4 Realización de pruebas.....	90
3.3.5 Tipos de pruebas.....	90
3.3.6 Desarrollo incremental.....	91
3.3.7 Estrategias de integración.....	91
3.3.8 Pruebas del sistema y de aceptación.....	92
3.5 Estándares para la evaluación.....	93
3.5.1 Modelos de calidad	93
3.5.2 Modelo calidad modelo MC CALL.....	93
3.5.3 Sistema SIAC.....	94
3.6 Modelo iso 9126	100
3.7 Plan de desarrollo de software.....	104

3.7.1 Propósito.....	104
3.7.2 Ámbito.....	104
3.7.3 Definiciones, acrónimos y abreviaturas.....	105
3.7.4 Referencias.....	105
3.7.5 Visión general.....	105
3.7.6 Recursos y miembros del equipo	105
3.7.6.1 Tareas y cronología de las mismas.....	106
3.8 DESCRIPCIÓN DEL PROYECTO.....	106
3.8.1 Propósito del proyecto ámbito y objetivos.....	106
3.8.2 Suposiciones y restricciones.....	107
3.8.3 Evolución del plan de evaluación de software.....	108
3.9 Estructura organizacional.....	108
3.9.1 Interfaces externas.....	109
3.9.2 Roles y responsabilidades.....	109
3.10 Gestión de procesos.....	110
3.10.1 Estimaciones del proyecto.....	110
3.10.2 Recuento de objetos y su clasificación.....	111
3.10.2.1 Experiencia y capacidad de los desarrolladores.....	111
3.10.2.2 Calcular el tiempo estimado.....	111
3.11 Plan del proyecto.....	112
3.11.1 Objetivos de iteración.....	112
3.12 Referencias del proyecto.....	113
3.13 Plantillas de excel.....	114
3.13.1 Resumen del informe técnico.....	115
3.13.2 Informe técnico.....	116
3.13.3 Nueva tecnología.....	118
3.14 Planificación de la elaboración del prototipo.....	119
3.14.1 Utilidad del prototipo del sistema.....	119
3.14.2 Análisis del prototipo.....	120
3.14.3 Necesidades de utilizar un prototipo.....	121
3.14.4 Determinación del sistema a implementar.....	121
3.14.5 Desarrollo de un modelo de trabajo.....	122

3.14.6 Revisión del prototipo.....	123
3.15 Prototipo propuesto.....	123
3.15.1 Utilización del sistema.....	124
3.15.2 Manejo del sistema.....	124
3.15.3Descripciones de procesos claves.....	124
3. 15.4Requerimientos de hardware y software.....	147
3.15.5 Recomendación.....	148
3.16 Recursos especiales.....	149
3.17 Conclusiones.....	150
3.18 Recomendaciones.....	150
ANEXOS.....	152

RESUMEN

Las instituciones públicas y privadas; en particular los colegios de la provincia se han visto en la necesidad de establecer normas tendientes a la estandarización del manejo de los recursos de información, por lo que los departamentos informáticos y administrativos han experimentado una importante evolución en los últimos años.

La implementación de pruebas caja negra, es una investigación basada en la preocupación sobre el manejo del sistema escolástico de la institución educativa: además para que se convierta en un sistema libre de errores, permitiendo así mejorar el aprovechamiento de la tecnología que dispone la institución.

La realización de esta investigación se basó en el objetivo de desarrollar un testing que incluya Hardware, Redes y Conexiones Electricas, aplicando las normas IEEE, ISO para verificar el funcionamiento del sistema escolástico de control de procesos sistemáticos SIAC

La presente implementación de pruebas de caja negra y caja blanca se aplico en el sistema escolástico del Colegio Nacional Primero de Abril, el cual permitio señalar los fallos detectados y las posibles soluciones que permitirá la optimización y buen funcionamiento del sistema.

El presente trabajo está constituido por un I CAPITULO que consta de un marco teórico en el que se describe todo lo relacionado a las pruebas de caja negra, sistemas de informacion, plan de pruebas como también las normas ISO, IEEE sobre la que se realizó la propuesta.

El II CAPITULO que esta dedicado al análisis e interpretacion de los resultados obtenidos en la aplicación de las encuestas a docentes y personal administrativo.

Ademas consta de un III CAPITULO en el cual se desarrollo un plan de trabajo para la evaluacion del proyecto, se aplico las pruebas de caja negra según las normas ISO, basandose en lo indicadores como son estrés, interfaz y seguridad y se evaluo al sistema el entorno de hardware, redes y conecciones electricas.

Finalmente la presente investigación contiene una propuesta la cual servirá para poder elaborar un sistema de información correcto así como también contiene un informe técnico detallando los aspectos más importantes de nuestra investigación.

SUMMARY

The institutions public and private in special the high schools of the province have seen each in the necessity of establishing norms tendents to the standardization of the handling of the resources of the resources of information for that the computer departments and administrativos have experienced important in the evolution the last years.

The implementation of tests of black box and White box is a based investigation in the preoccupation of the system free of mistake allowing this cuay to improve the use of the technology that prepares the institution.

The realization of this investigation is based in the objective of developing a testing that includes hardware software and nets applying the norms IEEE ISO the veriry the operation of systematic processes SIAC.

The present implementation of tests of black box and White box, is applies in the scholastic system of the PRIMERO DE ABRIL high school pointing out the defected the mistakes and the possibles solutions that it will allow the optomition and good operation of the system.

The present work this constitud by a theoretical mark in which is described all the related to the tests of White box and black box as well as the norms iso IEEE on which one carries out the propasal.

Also consists of an surrender to the analysis of the obtained results in the aplicacion of the surveys,

The type of investigation used in the work presently is descriptive, exploratory, bibliographical and documental way and of field for that which uses the inductive method obtaining results that they helped the position of general conclusions and to inclination of these power make decisions as well to delineate plans of action to establishing the proposal of investigation containing suggestions that allow to the directive of the high school to carry out some relating changes to the computer system SIAC.

Finally the present investigation contains a prototype the wish to be able to the elaborate of an correct system of information as well as it contains a technical report detailing the aspects important of our investigation.

INTRODUCCIÓN

Las Pruebas de software son aplicaciones de los principios científicos de la Informática, basadas en procesos que permiten verificar y revelar la calidad de un producto software; así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir posibles errores, no obstante, éste no es su único objetivo y su importancia es reconocida desde tiempos más antiguos.

Las pruebas de tipo CAJA NEGRA, donde los casos de prueba se diseñan considerando exclusivamente las entradas y salidas del sistema, sin preocuparse por la estructura interna del mismo.

Las pruebas de caja negra y caja blanca proveen una evaluación objetiva, imparcial y competente de las actividades de los sistemas de información y constituyen un medio para reorientar los esfuerzos de la institución hacia planes y objetivos en constante cambio. También ayudan a descubrir deficiencias o errores en aquellas partes del sistema que son examinadas.

En la actualidad, es cada vez mayor la necesidad de examinar y evaluar la calidad del producto software que se utiliza y sustenta la operatividad de las instrucciones.

Esto se logra con las pruebas de caja negra y caja blanca, además se pueden conocer los problemas que dificultan el desarrollo operativo u obstaculizan el avance del sistema informático. Es indispensable para valorar la importancia de los problemas, encontrar sus causas y proponer soluciones adecuadas e implementarlas.

Una vez estudiado lo referente a las pruebas de Caja Negra y Caja Blanca procedemos a la realización de nuestro trabajo que es la puesta en práctica en una de las instituciones educativas para lo cual hemos seleccionado al “Colegio Nacional Primero de Abril”

Las pruebas de software a ser aplicadas al sistema escolástico del “Colegio Nacional Primero de Abril” están orientadas a evaluar todos los módulos y submódulos del sistema escolástico, con el fin de dar a conocer a sus miembros el estado y calidad del mismo.

El COLEGIO NACIONAL “PRIMERO DE ABRIL” es uno de los pilares fundamentales en el desarrollo de la educación en la ciudad de Latacunga, colegio cuyo proceso de enseñanza es de óptima calidad y cuenta con 1300 estudiantes que provienen de distintas parroquias del al cantón Latacunga.

Actualmente cuenta con su propia infraestructura y sus principales objetivos fundamentales fueron impartir enseñanzas acordes al avance de la ciencia y tecnología, capacitando e innovando al personal docente que facilite la formación de jóvenes líderes; convertirse en un instrumento activo en el intento de fortalecer la conciencia cotopaxense y por ende nacional.

El Colegio Nacional “Primero de Abril” es una institución que brinda la oportunidad a la juventud del sector urbano y rural de una formación integral con las especializaciones de físico-matemático, químico-biólogo, ciencias sociales, contabilidad e informática. También orienta sus esfuerzos hacia la búsqueda de los mayores niveles de calidad, tratando de lograr niveles adecuados de eficiencia, eficacia y efectividad en su gestión.

El tipo de investigación utilizada en el presente trabajo es de forma bibliográfica, documental y de campo, para lo cual se utilizo el método inductivo, método

estadístico, método de observación obteniendo resultados que ayudaron al planteamiento de conclusiones generales, y a través de éstas poder tomar decisiones, como también a delinear planes de acción para establecer la propuesta de investigación, conteniendo sugerencias que permitan a los directivos del colegio a realizar algunos cambios referentes al sistema informático SIAC.

CAPÍTULO I

1. BASES TEÓRICAS

Es necesario considerar varios términos y aspectos que permitan fundamentar el presente proyecto de tesis a continuación:

1.1 SISTEMAS DE INFORMACION

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio con los datos relacionados que son fundamentales.

1.1.2 ELEMENTOS DE UN SISTEMA DE INFORMACIÓN.

Un **sistema de información** es un conjunto organizado de elementos, estos elementos son de 4 tipos:

- Personas.
- Datos.
- Actividades o técnicas de trabajo.

- Recursos materiales en general (típicamente recursos informáticos y de comunicación).

Todo ese conjunto de elementos interactúan entre si para procesar los datos y la información (incluyendo procesos manuales y automáticos) y distribuirla de la manera más adecuada posible en una determinada organización en función de sus objetivos. Normalmente el término es usado de manera errónea como sinónimo de sistema de información informático, estos son el campo de estudio de la tecnología de la información (IT), y aunque puedan formar parte de un sistema de información (como recurso material), por sí solos no se pueden considerar como sistemas de información, este concepto es más amplio que el de sistema de información informático. No obstante, un sistema de información puede estar basado en el uso de computadoras. Según la definición de Langefors este tipo de sistemas son:

- Un medio implementado tecnológicamente para grabar, almacenar y distribuir expresiones lingüísticas,
- así como para extraer conclusiones a partir de dichas expresiones.

1.2 PRUEBAS DE SOFTWARE

Las Pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto software.

Las Pruebas de software se integran dentro de las diferentes fases del Ciclo del software dentro de la Ingeniería de software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene.

La calidad de un sistema software es algo subjetivo que depende del contexto y del objeto que se pretenda conseguir. Para determinar dicho nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema.

Las pruebas de software, testing, beta testing es un proceso usado para identificar posibles fallos de implementación, calidad o usabilidad de un programa de ordenador o videojuego. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas. Únicamente un proceso de verificación formal puede probar que no existen defectos.

Hay muchos planteamientos para abordar el testeo de software, pero para verificar productos complejos de forma efectiva requiere de un proceso de investigación más que seguir un procedimiento al pie de la letra. Una definición de "testing" es: proceso de evaluación de un producto desde un punto de vista crítico, donde el "tester" (persona que realiza el testeo) somete el producto a una serie de acciones inquisitivas, y el producto responde con su comportamiento como reacción. Por supuesto, nunca se debe testear el software en un entorno de producción. Es necesario testear los nuevos programas en un entorno de pruebas separado físicamente del de producción. Para crear un entorno de pruebas en una máquina independiente de la máquina de producción es necesario crear las mismas condiciones que en la máquina de producción. Existen a tal efecto varias herramientas vendidas por los mismos fabricantes de hardware (IBM, Sun, HP etc.). Esas utilidades reproducen automáticamente las bases de datos para simular un entorno de producción.

En general, los informáticos distinguen entre errores de programación (o "bugs") y defectos de forma. En un defecto de forma, el programa no realiza lo que el usuario espera. Por el contrario, un error de programación puede describirse como un fallo en la semántica de un programa de ordenador. Éste podría presentarse, o no, como un defecto de forma si se llegan a dar ciertas condiciones de cálculo.

Una práctica común es que el testeo de un programa sea realizado por un grupo independiente de "testers" al finalizar su desarrollo y antes de sacarlo al mercado. Una práctica que viene siendo muy popular es distribuir de forma gratuita una versión no final del producto para que sean los propios consumidores los que la testeen. En ambos casos, a la versión del producto en pruebas y que es anterior a la versión final (o "master") se denomina beta, y a dicha fase de testeo, beta testing.

Puede además existir una versión anterior en el proceso de desarrollo llamada alpha, en la que el programa, aunque incompleto, dispone de funcionalidad básica y puede ser testeado.

1.2.1 HERRAMIENTA INFORMÁTICA DE PRUEBAS

Grupo de herramientas que permite reproducir la funcionalidad de una Aplicación Informática mediante el uso de guiones o "scripts", tanto en la interfaz gráfica de usuario como en la comunicación de la aplicación con otras, como puede ser entre una aplicación que se ejecuta en un navegador y el servidor web que le atiende y entre éste y una base de datos.

1.2.2 PRUEBAS DE CAJA BLANCA

Sinónimos:

- Pruebas estructurales
- Pruebas de caja transparente

En estas pruebas estamos siempre observando el código, que las pruebas se dedican a ejecutar con ánimo de "probarlo todo". Esta noción de prueba total se formaliza en lo que se llama "cobertura" y no es sino una medida porcentual de ¿cuánto código hemos cubierto?

Hay diferentes posibilidades de definir la cobertura. Todas ellas intentan sobrevivir al hecho de que el número posible de ejecuciones de cualquier programa no trivial es (a todos los efectos prácticos) infinito. Pero si el 100% de cobertura es infinito, ningún conjunto real de pruebas pasaría de un infinitésimo de cobertura.

1.2.3 PRUEBAS DE CAJA NEGRA

Sinónimos:

- Pruebas de caja opaca
- Pruebas funcionales
- Pruebas de entrada/salida
- Pruebas inducidas por los datos

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle

datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.

Las pruebas de caja negra están especialmente indicadas en aquellos módulos que van a ser interfaz con el usuario (en sentido general: teclado, pantalla, ficheros, canales de comunicaciones, etc) Este comentario no obsta para que sean útiles en cualquier módulo del sistema.

Las pruebas de caja negra se apoyan en la especificación de requisitos del módulo. De hecho, se habla de "cobertura de especificación" para dar una medida del número de requisitos que se han probado. Es fácil obtener coberturas del 100% en módulos internos, aunque puede ser más laborioso en módulos con interfaz al exterior. En cualquier caso, es muy recomendable conseguir una alta cobertura en esta línea.

El problema con las pruebas de caja negra no suele estar en el número de funciones proporcionadas por el módulo (que siempre es un número muy limitado en diseños razonables); sino en los datos que se le pasan a estas funciones. El conjunto de datos posibles suele ser muy amplio (por ejemplo, un entero).

A la vista de los requisitos de un módulo, se sigue una técnica algebraica conocida como "clases de equivalencia". Esta técnica trata cada parámetro como un modelo algebraico donde unos datos son equivalentes a otros. Si logramos partir un rango excesivamente amplio de posibles valores reales a un conjunto reducido de clases de equivalencia, entonces es suficiente probar un caso de cada clase, pues los demás datos de la misma clase son equivalentes.

1.2.3.2 LIMITACIONES

Lograr una buena cobertura con pruebas de caja negra es un objetivo deseable; pero no suficiente a todos los efectos. Un programa puede pasar con holgura millones de pruebas y sin embargo tener defectos internos que surgen en el momento más inoportuno (Murphy no olvida).

Por ejemplo, un PC que contenga el virus Viernes-13 puede estar pasando pruebas de caja negra durante años y años. Sólo falla si es viernes y es día 13; pero ¿a quién se le iba a ocurrir hacer esa prueba?

Las pruebas de caja negra nos convencen de que un programa hace lo que queremos; pero no de que haga (además) otras cosas menos aceptables.

1.2.3.2 PRUEBAS DE INTEGRACIÓN

Las pruebas de integración se llevan a cabo durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto.

Estas pruebas se pueden plantear desde un punto de vista estructural o funcional.

Las pruebas estructurales de integración son similares a las pruebas de caja blanca; pero trabajan a un nivel conceptual superior. En lugar de referirnos a sentencias del lenguaje, nos referiremos a llamadas entre módulos. Se trata pues de identificar todos los posibles esquemas de llamadas y ejercitarlos para lograr una buena cobertura de segmentos o de ramas.

Las pruebas funcionales de integración son similares a las pruebas de caja negra. Aquí trataremos de encontrar fallos en la respuesta de un módulo cuando su operación depende de los servicios prestados por otro(s) módulo(s). Según nos vamos acercando al sistema total, estas pruebas se van basando más y más en la especificación de requisitos del usuario.

Las pruebas finales de integración cubren todo el sistema y pretenden cubrir plenamente la especificación de requisitos del usuario. Además, a estas alturas ya suele estar disponible el manual de usuario, que también se utiliza para realizar pruebas hasta lograr una cobertura aceptable.

En todas estas pruebas funcionales se siguen utilizando las técnicas de partición en clases de equivalencia y análisis de casos límite (fronteras).

1.2.3.4 PRUEBAS DE ACEPTACIÓN

Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo, pues sería

impresentable dar la cara al cliente; sino una vez pasada todas las pruebas de integración por parte del desarrollador.

La experiencia muestra que aún después del más cuidadoso proceso de pruebas por parte del desarrollador, quedan una serie de errores que sólo aparecen cuando el cliente se pone a usarlo.

Muchos desarrolladores ejercitan unas técnicas denominadas "pruebas alfa" y "pruebas beta". Las pruebas alfa consisten en invitar al cliente a que venga al entorno de desarrollo a probar el sistema. Se trabaja en un entorno controlado y el cliente siempre tiene un experto a mano para ayudarle a usar el sistema y para analizar los resultados.

Las pruebas beta vienen después de las pruebas alfa, y se desarrollan en el entorno del cliente, un entorno que está fuera de control. Aquí el cliente se queda a solas con el producto y trata de encontrarle fallos (reales o imaginarios) de los que informa al desarrollador.

Las pruebas alfa y beta son habituales en productos que se van a vender a muchos clientes. Algunos de los potenciales compradores se prestan a estas pruebas bien por ir entrenando a su personal con tiempo, bien a cambio de alguna ventaja económica (mejor precio sobre el producto final, derecho a mantenimiento gratuito, a nuevas versiones.

1.3 PLAN DE PRUEBAS

Un plan de pruebas está constituido por un conjunto de pruebas. Cada prueba debe:

- Dejar claro qué tipo de propiedades se quieren probar (corrección, robustez, fiabilidad, amigabilidad).
- Saber cómo se mide el resultado
- Especificar en qué consiste la prueba (hasta el último detalle de cómo se ejecuta)
- Definir cuál es el resultado que se espera (identificación, tolerancia).

Las pruebas angelicales carecen de utilidad, tanto si no se sabe exactamente lo que se quiere probar, o si no está claro cómo se prueba, o si el análisis del resultado se hace "a ojo".

Estas mismas ideas se suelen agrupar diciendo que un caso de prueba consta de 3 bloques de información:

1. El propósito de la prueba.
2. Los pasos de ejecución de la prueba.
3. El resultado que se espera.

Y todos y cada uno de esos puntos debe quedar perfectamente documentado. Las pruebas de usar y tirar más vale que se tiren directamente, aún antes de usarlas.

Cubrir estos puntos es muy laborioso y, con frecuencia, tedioso, lo que hace desagradable (o al menos muy aburrida) la fase de pruebas. Es mucho más divertido codificar que probar. Tremendo error en el que, no obstante, es fácil incurrir.

Respecto al orden de pruebas, una práctica frecuente es la siguiente:

1. Pasar pruebas de caja negra analizando valores límite. Recuerde que hay que analizar condiciones límite de entrada y de salida.
2. Identificar clases de equivalencia de datos (entrada y salida) y añadir más pruebas de caja negra para contemplar valores normales (en las clases de equivalencia en que estos sean diferentes de los valores límite; es decir, en rangos amplios de valores).

1.3.2 ASPECTOS PSICOLÓGICOS Y ORGANIZACIÓN DEL TRABAJO

- a. Probar es ejercitar un programa para encontrarle fallos. Jamás se debería probar un programa con el ánimo de mostrar que funciona; ese no es el objetivo.
- b. Un caso de prueba tiene éxito cuando encuentra un fallo. Lo gracioso no es encontrar un caso en el que el programa funciona perfectamente. Eso es, simplemente, lo normal.

- c. Las pruebas debe diseñarlas y pasarlas una persona distinta de la que ha escrito el código; es la única forma de no ser "comprensivo con los fallos".

- d. Si en un módulo (o sección de un programa, en general) se encuentran muchos fallos, hay que insistir sobre él. Es muy habitual que los fallos se concentren en pequeñas zonas. Hay mil causas para que ocurra este efecto:
 - código escrito por un programador malo
 - código muy difícil
 - código mal o insuficientemente especificado
 - código escrito en un mal día, con prisas, ...

- e. Las pruebas pueden encontrar fallos; pero jamás demostrar que no los hay.

- f. Las pruebas también tienen fallos. Los errores son propios de los humanos: todo el mundo se equivoca. Si una prueba falla, hay que revisar tanto lo que se prueba como lo que lo prueba. No obstante, la experiencia muestra que (casi siempre) hay más fallos el probado que en el probador.

1.4 QUE ES UNA ARQUITECTURA

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente dentro de la organización.

Debemos señalar que para seleccionar el modelo de una arquitectura, hay que partir del contexto tecnológico y organizativo del momento y, que la arquitectura Cliente/Servidor requiere una determinada especialización de cada uno de los diferentes componentes que la integran.

1.5. QUE ES UN CLIENTE

Es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes LAN o WAN. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente.

1.6 QUE ES UN SERVIDOR

Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs o WANs, para proveer de múltiples servicios a los clientes y ciudadanos

tales como impresión, acceso a bases de datos, fax, procesamiento de imágenes, etc.

1.6.1 TIPOS DE SERVIDOR

a. Servidores de archivos

Servidor donde se almacena archivos y aplicaciones de productividad como por ejemplo procesadores de texto, hojas de cálculo, etc.

b. Servidores de bases de datos

Servidor donde se almacenan las bases de datos, tablas, índices. Es uno de los servidores que más carga tiene.

c. Servidores de transacciones

Servidor que cumple o procesa todas las transacciones. Valida primero y recién genera un pedido al servidor de bases de datos.

d. Servidores de Groupware

Servidor utilizado para el seguimiento de operaciones dentro de la red.

e. Servidores de objetos

Contienen objetos que deben estar fuera del servidor de base de datos. Estos objetos pueden ser videos, imágenes, objetos multimedia en general.

f. Servidores Web

Se usan como una forma inteligente para comunicación entre empresas a través de Internet.

Este servidor permite transacciones con el acondicionamiento de un browser específico.

1.6.2 BASE DE DATOS

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

Existen programas denominados sistemas gestores de bases de datos, abreviado SGBD, que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

1.6.2.1 BASE DE DATOS DISTRIBUIDA

Son las Bases de Datos que no están almacenadas totalmente en un solo lugar físico, (esta segmentada) y se comunican por medio de enlaces de comunicaciones a través de una red de computadoras distribuidas geográficamente.

Ventajas

- Refleja una estructura organizacional - los fragmentos de la base de datos se ubican en los departamentos a los que tienen relación.
- Disponibilidad - un fallo en una parte del sistema solo afectará a un fragmento, en lugar de a toda la base de datos.

- Rendimiento - los datos generalmente se ubican cerca del sitio con mayor demanda, también los sistemas trabajan en paralelo, lo cual permite balancear la carga en los servidores.
- Economía - es más barato crear una red de muchas computadoras pequeñas, que tener una sola computadora muy poderosa.
- Modularidad - se pueden modificar, agregar o quitar sistemas de la base de datos distribuida sin afectar a los demás sistemas (módulos).

Desventajas

- Complejidad - Se debe asegurar que la base de datos sea transparente, se debe lidiar con varios sistemas diferentes que pueden presentar dificultades únicas..
- Economía - la complejidad y la infraestructura necesaria implica que se necesitará una mayor mano de obra.
- Seguridad - se debe trabajar en la seguridad de la infraestructura así como cada uno de los sistemas.
- Integridad - Se vuelve difícil mantener la integridad, aplicar las reglas de integridad a través de la red puede ser muy caro en términos de transmisión de datos.
- Falta de experiencia - las bases de datos distribuidas son un campo relativamente nuevo y poco común por lo cual no existe mucho personal con experiencia o conocimientos adecuados.
- Carencia de estándares - aún no existen herramientas o metodologías que ayuden a los usuarios a convertir un DBMS centralizado en un DBMS distribuido.

1.7 WINDOWS 2000 SERVER

Está destinada a ser el servidor de archivos, impresión, web, FTP de una empresa de chica a mediana. Su antecesor es Windows NT 4.0 Server. Es ideal para

cuando no se requiere de un servidor dedicado a cada tarea o departamento y se puede tener todo centralizado en 1.

1.7.1 WINDOWS XP (EXPERIENCE)

La unión de Windows NT/2000 y la familia de Windows 9.x se alcanzó con Windows XP puesto en venta en 2001 en su versión Home y Professional. Windows XP usa el núcleo de Windows NT. Incorpora una nueva interfaz y hace alarde de mayores capacidades multimedia. Además dispone de otras novedades como la multitarea mejorada, soporte para redes inalámbricas y asistencia remota. Se puede agregar que inmediatamente después de haber lanzado el último Service Pack (SP2), Microsoft diseñó un sistema orientado a empresas y corporaciones, llamado Microsoft Windows XP Corporate Edition, algo similar al Windows XP Profesional, solo que diseñado especialmente para empresas. En el apartado multimedia, XP da un avance con la versión Media Center(2002-2005). Esta versión ofrece una interfaz de acceso fácil con todo lo relacionado con multimedia (TV, fotos, reproductor DVD, Internet...).

1.8 INTERFACES DE USUARIO

La creación de los interfaces de usuario ha sido un área del desarrollo de software que ha evolucionado dramáticamente a partir de la década de los setenta. Esta es una de las partes más importantes de cualquier programa, ya que determina que tan fácil es posible que el programa haga lo que el usuario quiere hacer.

El diseño de una interfaz es fundamental para el éxito de un programa. Un buen programa con una pobre interfaz tendrá una mala imagen, y al contrario, una buena interfaz puede realzar un programa mediocre.

1.8.1 INTERFAZ DE USUARIO DE UN PROGRAMA:

- La interfaz de usuario es el vínculo entre el usuario y el programa de computadora. Una interfaz es un conjunto de comandos o menús a través de los cuales el usuario se comunica con el programa.
- Una interfaz es un programa que permite relacionar dos ambientes diferentes, en este caso nos referimos como interfaz al que le permite a Ud. comunicarse con el programa.
- La interfaz representa el punto de encuentro entre el usuario y la computadora. En esta interacción, el usuario juzga la utilidad de la interfaz; el hardware y el software se convierten en simples herramientas sobre los cuales fue construida la interfaz.
- Conjunto de elementos hardware y software de un ordenador que presentan información al usuario y le permiten interactuar con la información y con el ordenador. El interfaz incluye el hardware que forma el sistema, como el teclado, un dispositivo apuntador tal como un ratón, joystick o trackball, la UCP y el monitor. Los componentes software son los elementos que el usuario ve, oye, a los que apunta o toca en la pantalla

para interactuar con el ordenador, así como la información con la que trabaja. También se puede considerar parte de la interfaz la documentación (manuales, ayuda, referencia, tutoriales) que acompaña al hardware y al software.

- Es ese "algo" entre uno mismo y el objeto de la interacción cuando uno accede e interactúa con el sistema, este algo, nos informa que acciones son posibles, el estado actual del objeto y los cambios producidos, y nos permite actuar con o sobre el sistema o la herramienta. Es a la vez un límite y un espacio común entre ambas partes.
- En el caso de producto informático, la interfaz no es solo el programa o lo que se ve en la pantalla. Desde el momento que el usuario abre la caja, comienza a interactuar con el producto y por lo tanto, comienza su experiencia.

1.8.2 MODELOS DE INTERFAZ

Existen tres puntos de vista distintos en una interfaz de usuario: el del usuario, el del programador y el del diseñador.

Cada uno tiene un modelo mental propio de la interfaz, que contiene los conceptos y expectativas acerca de la interfaz, desarrollados a través de su experiencia.

- **Modelo del usuario.** El usuario tiene su visión personal del sistema, y espera que éste se comporte de una cierta forma, que se puede conocer estudiando al usuario (realizando tests de usabilidad, entrevistas, o a través

de una realimentación). Una interfaz debe facilitar el proceso de crear un modelo mental efectivo.

Para ello son de gran utilidad las metáforas, que asocian un dominio nuevo a uno ya conocido por el usuario; un ejemplo típico es la metáfora del escritorio, común a la mayoría de las interfaces gráficas actuales.

Las metáforas constituyen una ayuda, relacionando el sistema con algo ya conocido por el usuario.

- **Modelo del programador.** Es el más fácil de visualizar, al poderse especificar formalmente. Está constituido por los objetos que manipula el programador, distintos de los que trata el usuario (ejemplo: base de datos - agenda telefónica). Estos objetos deben esconderse del usuario.

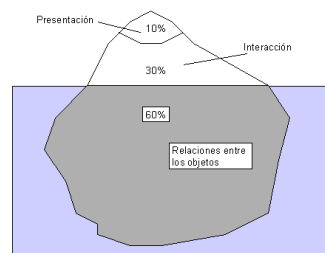
Los conocimientos del programador incluyen la plataforma de desarrollo, el sistema operativo, las herramientas de desarrollo, especificaciones. Sin embargo, esto no significa necesariamente que tenga la habilidad de proporcionar al usuario los modelos y metáforas más adecuadas. Muchos no consideran el modelo del usuario del programa, y sí sus propias expectativas acerca de cómo trabaja el ordenador.

- **Modelo del diseñador.** El diseñador mezcla las necesidades, ideas, deseos del usuario y los materiales de que dispone el programador para diseñar un producto de software. Es, pues, un intermediario entre ambos.

- El modelo del diseñador describe los objetos que utiliza el usuario, su presentación al mismo y las técnicas de interacción para su manipulación.

El modelo consta, pues, de tres partes: presentación, interacción y relaciones entre los objetos.

La presentación es lo que primero capta la atención del usuario, pero más tarde pasa a un segundo plano, y adquiere más importancia cómo el producto cumple las expectativas del usuario. La presentación no es lo más relevante, y un abuso en la misma (por ejemplo, en el color) puede ser contraproducente, distrayendo al usuario.



La segunda parte define las técnicas de interacción del usuario, a través de diversos dispositivos.

La tercera es la más importante, y es donde el diseñador determina la metáfora adecuada que encaja con el modelo mental del usuario. El modelo debe comenzar por esta parte e ir hacia arriba. Una vez definida la metáfora y los objetos del interfaz, los aspectos visuales saldrán de una manera lógica y fácil.

Estos modelos deben estar claros para los participantes en el desarrollo de un producto, de forma que se consiga una interfaz atractiva y a la vez efectiva para el trabajo con el programa. Una interfaz no es simplemente

una cara bonita; esto puede impresionar a primera vista, pero decepcionar a la larga. Lo realmente importante es que el programa se adapte bien al modelo del usuario, cosa que se puede comprobar utilizando el programa más allá de la primera impresión.

Aspectos de la psicología humana. Al diseñar interfaces de usuario deben tenerse en cuenta las habilidades cognitivas y de percepción de las personas, y adaptar el programa a ellas.

Así, una de las cosas más importantes que una interfaz puede hacer es reducir la dependencia de las personas de su propia memoria, no forzándoles a recordar cosas innecesariamente (por ejemplo, información que apareció en una pantalla anterior) o a repetir operaciones ya realizadas (por ejemplo, introducir un mismo dato repetidas veces).

La persona tiene unas habilidades distintas de la máquina, y ésta debe utilizar las suyas para soslayar las de aquella (como por ejemplo la escasa capacidad de la memoria de corto alcance).

Los objetivos de los usuarios son muy variables; incluso un usuario puede cambiar su objetivo cuando realiza una misma tarea.

Los programas son usados por usuarios con distintos niveles de conocimientos (desde novatos hasta expertos). Así pues, no existe una interfaz válida para todos los usuarios y todas las tareas. Debe permitirse libertad al usuario para que elija el modo de interacción que más se adecúe

a sus objetivos en cada momento. La mayoría de los programas y sistemas operativos ofrecen varias formas de interacción al usuario.

1.8.3 DISEÑO DE INTERFACES

Reglas para el diseño de interfaces de usuario

- Dar control al usuario.
- Reducir la carga de memoria del usuario.
- Consistencia.

Regla 1: Dar control al usuario. El diseñador debe dar al usuario la posibilidad de hacer su trabajo, en lugar de suponer qué es lo que éste desea hacer. La interfaz debe ser suficientemente flexible para adaptarse a las exigencias de los distintos usuarios del programa.

En concreto, se pueden enumerar los siguientes principios que permiten al usuario estar en posesión del control:

1. Usar adecuadamente los modos de trabajo.
2. Permitir a los usuarios utilizar el teclado o el ratón.

3. Permitir al usuario interrumpir su tarea y continuarla más tarde.
4. Utilizar mensajes y textos descriptivos.
5. Permitir deshacer las acciones e informar de su resultado.
6. Permitir una cómoda navegación dentro del producto y una fácil salida del mismo.
7. Permitir distintos niveles de uso del producto para usuarios con distintos niveles de experiencia.
8. Hacer transparente la interfaz al usuario, que debe tener la impresión de manipular directamente los objetos con los que está trabajando.
9. Permitir al usuario personalizar la interfaz (presentación, comportamiento e interacción).
10. Permitir al usuario manipular directamente los objetos de la interfaz.

Regla 2: Reducir la carga de memoria del usuario. La interfaz debe evitar que el usuario tenga que almacenar y recordar información.

Para ello, debe seguir los siguientes principios:

1. Aliviar la carga de la memoria de corto alcance (permitir deshacer, copiar y pegar; mantener los últimos datos introducidos).
2. Basarse en el reconocimiento antes que en el recuerdo (ejemplo: elegir de entre una lista en lugar de teclear de nuevo).
3. Proporcionar indicaciones visuales de dónde está el usuario, qué está haciendo y qué puede hacer a continuación (ejemplo: pantalla de Word).
4. Proporcionar funciones deshacer, rehacer y acciones por defecto.
5. Proporcionar atajos de teclado (iniciales en menús, teclas rápidas).
6. Asociar acciones a los objetos (menú contextual).
7. Utilizar metáforas del mundo real (sistema telefónico, agenda).

8. Presentar al usuario sólo la información que necesita (menús simples/avanzados, wizards, asistentes).

9. Hacer clara la presentación visual (colocación/agrupación de objetos, evitar la presentación de excesiva información).

Regla 3: Consistencia. Permite al usuario utilizar conocimiento adquirido en otros programas consistentes con el nuevo programa. Ejemplo: mostrar siempre el mismo mensaje ante un mismo tipo de situación, aunque se produzca en distintos lugares. Principios:

1. Consistencia en la realización de las tareas: proporcionar al usuario indicaciones sobre el proceso que está siguiendo.

2. Consistencia dentro del propio producto y de un producto a otro. La consistencia se aplica a la presentación (lo que es igual debe aparecer igual: color del texto estático), el comportamiento (un objeto se comporta igual en todas partes) y la interacción (los atajos y operaciones con el ratón se mantienen; el usuario espera los mismos resultados cuando interactúa de la misma forma con objetos distintos).

3. Consistencia en los resultados de las interacciones: misma respuesta ante la misma acción. Los elementos estándar del interfaz deben comportarse siempre de la misma forma (las barras de menús despliegan menús al seleccionarse).

4. Consistencia de la apariencia estética (iconos, fuentes, colores, distribución de pantallas).

5. Fomentar la libre exploración de la interfaz, sin miedo a consecuencias negativas.

1.8.4 PROCESO DE DISEÑO DE INTERFACES DE USUARIO

En el proceso de diseño de una interfaz de usuario se pueden distinguir cuatro fases:

1. Reunir y analizar la información del usuario.
2. Diseñar la interfaz de usuario.
3. Construir la interfaz de usuario.
4. Validar la interfaz de usuario.

Reunir y analizar la información del usuario: Qué tipo de usuarios van a utilizar el programa, qué tareas van a realizar los usuarios y cómo las van a realizar, qué exigen los usuarios del programa, en qué entorno se desenvuelven los usuarios (físico, social, cultural).

Diseñar la interfaz de usuario: Es importante dedicar tiempo y recursos a esta fase, antes de entrar en la codificación. En esta fase se definen los objetivos de usabilidad del programa, las tareas del usuario, los objetos y acciones de la

interfaz, los iconos, vistas y representaciones visuales de los objetos, los menús de los objetos y ventanas. Todos los elementos visuales se pueden hacer primero a mano y luego refinar con las herramientas adecuadas.

Construir la interfaz de usuario: Es interesante realizar un prototipo previo, una primera versión del programa que se realice rápidamente y permita visualizar el producto para poderlo probar antes de codificarlo definitivamente.

Validar la interfaz de usuario: Se deben realizar pruebas de usabilidad del producto, a ser posible con los propios usuarios finales del mismo.

1.8.5 Técnicas avanzadas para el diseño de interfaces de usuario

Presentación de información: no se deben colocar demasiados objetos en la pantalla, y los que existen deben estar bien distribuidos. Cada elemento visual influye en el usuario no sólo por sí mismo, sino también por su combinación con el resto de elementos presentes en la pantalla.

Color: Es probablemente el elemento de la interfaz que con más frecuencia es mal utilizado. El color comunica información, no es sólo decorativo (ejemplo: reforzar mensajes de error). Deben utilizarse combinaciones adecuadas (por ejemplo, las paletas proporcionadas por los sistemas operativos). El color debe atraer la atención, pero no cansar después de un rato de trabajo. Es especialmente importante seguir las líneas de diseño existentes. Principio básico: diseñar primero en blanco y negro, y luego añadir el color.

Audio: Primero es preciso ver cuándo es más apropiado que la información visual. Segundo, determinar el sonido adecuado. Tercero, permitir la personalización (volumen y desactivación). Como en el caso de los colores existen guías de uso. En lugares de trabajo abiertos, puede ser poco efectivo; además, puede ser embarazoso para algunas personas. El sonido debe usarse para informar, no cuando no añade nada nuevo (por ejemplo, un mensaje de aviso de correo o de bienvenida, respectivamente, al iniciar una sesión de trabajo).

Animación: Se define como un cambio en el tiempo de la apariencia visual de un elemento gráfico. Ejemplos de su uso: progreso de acciones (copia de ficheros en Windows 95, instalación de programas), estado de procesos (iconos de impresora), acciones posibles (cambios en el cursor al desplazar el ratón). La animación puede ayudar a subrayar iconos importantes, mostrar el estado de un objeto particular o explicar su comportamiento.

1.8.5.1 DISEÑO ITERATIVO DE INTERFACES

El diseño iterativo de interfaces es un proceso independiente de las técnicas utilizadas para llevarlo a cabo.

Actualmente, el proceso del desarrollo de una interfaz se concibe como un ciclo que consta de cuatro etapas en varios niveles:

- Diseño.
- Implementación.
- Medición.
- Evaluación.

El resultado (o output) de cada etapa es la alimentación (o input) de la que sigue, incluso el de la última.

Los resultados de la etapa de evaluación se toman para rediseñar la interfaz, implementarla nuevamente, medir y así sucesivamente.

Debido a esa repetición o auto alimentación se lo llama diseño iterativo.

Es importante comprender que este ciclo no solo se cumple dentro del ciclo de vida de un producto, sino también entre productos y dentro de cada etapa misma.

Mientras tengamos tiempo, trataremos de hacer tantos ciclos de mejoramiento como nos sea posible, hasta la fecha límite.

La siguiente versión tomará al producto existente como su comienzo y otra vez comenzará el ciclo.

1.8.5.2 EL PROCESO DE DISEÑO Y EL EQUIPO DE DESARROLLO

Además de la recursividad, otra característica del enfoque actual del diseño de interfaces es que no sólo involucra a los especialistas en usabilidad o diseño, sino a todo el equipo de desarrollo.

¿Quiénes constituyen el equipo de desarrollo?

Todos aquellos que participan de alguna manera en el desarrollo o comercialización del sistema o herramienta: gente de marketing, comunicación, documentación, sistemas e informática, diseño y usabilidad, packaging, etc.

Cada uno tiene conocimiento acerca de un área específica, y su participación a lo largo del desarrollo aumenta las posibilidades de éxito.

Todos los equipos pueden tener discusiones acerca de la usabilidad de un sitio, o el uso de la aplicación que están haciendo. Muchas de esas discusiones no están basadas en hechos u observaciones, sino en mitos que nos influyen sin que nos demos cuenta (más sobre los mismos en los workshops).

Nada mejor para terminar esas discusiones que un pequeño test de usabilidad: no solo elimina reuniones interminables y defensas acaloradas por opiniones personales, sino que tiene un poder de convencimiento y demostración casi imposible de igualar.

No hay nada tan claro como ver a un usuario tratando infructuosamente de usar ese software que creíamos tan simple, para volver al laboratorio sin discusiones y aceptar que es necesario cambiar la versión actual.

1.8.5.3 LAS ETAPAS Y TAREAS DEL CICLO

Diseño	<ul style="list-style-type: none">• Análisis de requerimientos del producto• Análisis de las tareas• Conocimiento del usuario• Generación de posibles metáforas y análisis de tipo de diálogo• Revisión de posibilidades para la implementación
--------	---

Implementación	<ul style="list-style-type: none"> • Generación de prototipos (profundos o amplios, para investigación general o de ajustes) • Desarrollo de la aplicación, sitio o sistema
Medición (Test de usabilidad)	<ul style="list-style-type: none"> • Planificación (desarrollo del plan, definición de las medidas, selección de participantes, formación de observadores, preparación de los materiales). • Test (prueba piloto, test con usuarios)
Evaluación	<ul style="list-style-type: none"> • Conclusión (análisis de los datos, elaboración del informe, resultados y recomendaciones) • Comparación contra estándares (internos y/o externos), versiones anteriores del mismo producto y productos competidores. • Verificación de las diferencias • Generación de nuevas metas

1.8.5.4 EVOLUCIÓN DE LAS INTERFACES DE USUARIO

La evolución de las interfaces de usuario corre en paralelo con la de los sistemas operativos; de hecho, la interfaz constituye actualmente uno de los principales elementos de un sistema operativo. A continuación se muestran las distintas

interfaces que históricamente han ido apareciendo, ejemplificándolas con las sucesivas versiones de los sistemas operativos más populares.

Interfaces de línea de mandatos (command-line user interfaces, Cuis). Es el característico del DOS, el sistema operativo de los primeros PC, y es el estilo más antiguo de interacción hombre-máquina. El usuario escribe órdenes utilizando un lenguaje formal con un vocabulario y una sintaxis propia (los mandatos en el caso del DOS). Se usa un teclado, típicamente, y las órdenes están encaminadas a realizar una acción.

El usuario no suele recibir mucha información por parte del sistema (ejemplo: indicador del DOS), y debe conocer cómo funciona el ordenador y dónde están los programas (nada está oculto al usuario). El modelo de la interfaz es el del programador, no el del usuario. Ejemplo del DIR-DEL-DIR, por la falta de información de respuesta del DOS. Otras veces, en cambio, es excesiva: etiqueta del volumen en el DIR.

Inconveniente:

Carga de memoria del usuario (debe memorizar los mandatos; incluso la ayuda es difícil de leer); nombres no siempre adecuados a las funciones, significado de los mandatos mal comprendido a veces (varios mandatos con el mismo o parecido significado, como DEL y ERASE); inflexible en los nombres (DEL y no DELETE).

Ventajas:

Potente, flexible y controlado por el usuario, aunque esto es una ventaja para usuarios experimentados. La sintaxis es estricta, y los errores pueden ser graves.

En suma, un CUI es adecuado para usuarios expertos, no para novatos. Para aquellos resultan más rápidos, por lo que se puede diseñar un CUI como parte de una interfaz, para que se pueda utilizar una vez que se tenga experiencia.

Interfaces de menús. Un menú es una lista de opciones que se muestran en la pantalla o en una ventana de la pantalla para que los usuarios elijan la opción que deseen (véase ejemplo). Los menús permiten dos cosas: navegar dentro de un sistema, presentando rutas que llevan de un sitio a otro, y seleccionar elementos de una lista, que representan propiedades o acciones que los usuarios desean realizar sobre algún objeto.

Las interfaces de menús aparecen cuando el ordenador se vuelve una herramienta de usuario y no sólo de programadores. Las actuales interfaces gráficas u orientadas a objetos siguen utilizando este tipo de interfaces (los distintos estilos de interfaces no son mutuamente exclusivos).

Existen distintos tipos de menús. Los primeros fueron los menús de pantalla completa, estructurados jerárquicamente los menús de barra, situados en la parte superior de la pantalla, son profusamente utilizados en las aplicaciones actuales. Contienen una lista de acciones genéricas que dan paso a menús desplegables donde se concretan. Estos menús pueden llevar a su vez a otros: son los menús en cascada. Pueden cambiar dinámicamente y deshabilitar opciones que no estén disponibles en un momento dado (marcándolas habitualmente en gris).

Las paletas o barras de herramientas son menús gráficos con acciones, herramientas y opciones que se pueden colocar en la pantalla. Se utilizan mucho en programas gráficos.

Los menús contextuales o menús pop-up son los más recientes. Se llaman así porque el contenido del menú depende del contexto de trabajo del usuario. Contienen únicamente las opciones que son aplicables al objeto seleccionado, más algunas de uso frecuente que también son accesibles desde el menú de barra.

Las interfaces de menús, bien estructuradas, son buenas para usuarios noveles o esporádicos. Son fáciles de aprender y de recordar. Pueden existir menús simples y avanzados, para adaptarse al tipo de usuario.

Precauciones: no ocupar demasiado espacio de la pantalla, recordar la información acumulada de menús precedentes, no colocar demasiados elementos en el menú, agruparlos de manera lógica (no en orden alfabético, por ejemplo; esto ayuda a recordarlos), permitir la personalización por parte del usuario, usar una terminología adecuada y consistente dentro del programa y con otros programas (Exit, Quit, Escape, Close, Return, Back). Las interfaces de menús serán utilizadas normalmente en conjunción con los otros estilos de interfaces.

Interfaces gráficas (graphical user interfaces, GUIs). Desarrolladas originalmente por XEROX (sistema Xerox Star, 1981, sin éxito comercial), aunque popularizadas por Apple (Steven Jobs se inspiró en los trabajos de Xerox y creó el Apple Lisa, 1983, sin éxito, y Apple Macintosh, 1984, con éxito debido en gran medida a su campaña publicitaria).

Un GUI es una representación gráfica en la pantalla del ordenador de los programas, datos y objetos, así como de la interacción con ellos. Un GUI proporciona al usuario las herramientas para realizar sus operaciones, más que una lista de las posibles operaciones que el ordenador es capaz de hacer.

1.9 SOFTWARE LIBRE

Es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente. El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios como internet

1.9.1 LINUX

Linux es un sistema operativo libre tipo Unix. Es usualmente utilizado junto a las herramientas GNU como interfaz entre los dispositivos de hardware y los programas usados por el usuario para manejar un computador. A la unión de ambas tecnologías, más la inclusión de algunas otras, (como entornos de escritorio e interfaces gráficas) se le conoce como distribución GNU/Linux. Fue lanzado bajo la licencia pública general de GNU y es desarrollado gracias a contribuciones provenientes de colaboradores de todo el mundo, por lo que es uno de los ejemplos más notables de software libre.

Linux fue creado por Linus Torvalds en 1991. Muy pronto, la comunidad de Minix (un clon de Unix), contribuyó en el código y en ideas para el núcleo Linux. Por aquel entonces, el Proyecto GNU ya había creado muchos de los componentes necesarios para conseguir un entorno operador con software libre, pero su propio sistema operativo, el llamado (GNU Hurd), se encontraba incompleto por lo que comenzaron a usar Linux. Linux es el tercer sistema operativo más utilizado en el entorno de escritorio (desktop) y el más utilizado en el entorno de servidores. Actualmente posee una cuota de mercado del 1,05% a nivel mundial.

1.10 QUE ES UNA NORMA

Las normas son un modelo, un patrón, ejemplo o criterio a seguir. Una norma es una fórmula que tiene valor de regla y tiene por finalidad definir las características que debe poseer un objeto y los productos que han de tener una compatibilidad para ser usados a nivel internacional. Pongamos, por ejemplo, el problema que ocasiona a muchos usuarios los distintos modelos de enchufes que existen a escala internacional para poder acoplar pequeñas máquinas de uso personal: secadores de cabello, máquinas de afeitar, etc. cuando se viaja. La incompatibilidad repercute en muchos campos. La normalización de los productos es, pues, importante.

La finalidad principal de las normas ISO es orientar, coordinar, simplificar y unificar los usos para conseguir menores costes y efectividad.

Tiene valor indicativo y de guía. Actualmente su uso se va extendiendo y hay un gran interés en seguir las normas existentes porque desde el punto de vista económico reduce costes, tiempo y trabajo. Criterios de eficacia y de capacidad de respuesta a los cambios. Por eso, las normas que presentemos, del campo de la información y documentación, son de gran utilidad porque dan respuesta al reto de las nuevas tecnologías.

1.10.1 Tipología de normas

La familia de **normas ISO 9000** son normas de "calidad" y "gestión continua de calidad", establecidas por la Organización Internacional para la Estandarización (ISO) que se pueden aplicar en cualquier tipo de organización o actividad sistemática, que esté orientada a la producción de bienes o servicios. Se componen de estándares y guías relacionados con sistemas de gestión y de herramientas específicas como los métodos de auditoría (el proceso de verificar que los sistemas de gestión cumplen con los estándares).

Los principales beneficios son:

- Mejorar la satisfacción del cliente.
- Mejorar continuamente los procesos relacionados con la Calidad.

Otros beneficios adicionales son:

- Reducción de rechazos e incidencias en la producción o prestación del servicio.
- Aumento de la productividad.

La familia de normas apareció por primera vez en 1987 teniendo como base una norma estándar británica (BS), y se extendió principalmente a partir de su versión de 1994, estando actualmente en su versión 2008, publicada el 13 de noviembre de 2008

La principal norma de la familia es actualmente la: ISO 9001:2008 - Sistemas de Gestión de la Calidad - Requisitos.

Otra norma vinculante a la anterior: ISO 9004:2000 - Sistemas de Gestión de la Calidad - Guía de mejoras del funcionamiento.

Las normas ISO 9000 de 1994 estaban principalmente basadas para organizaciones que realizaban proceso productivo y, por tanto, su implantación en la institución educativa era muy dura y por eso se sigue en la creencia de que es un sistema bastante burocrático.

Con la revisión de 2000 se ha conseguido una norma bastante menos burocrática para organizaciones de todo tipo, y además se puede aplicar sin problemas en empresas de servicios e incluso en la Administración Pública.

Para verificar que se cumple con los requisitos de la norma, existen unas entidades de certificación que dan sus propios certificados y permiten el sello. Estas entidades están vigiladas por organismos nacionales que les dan su acreditación.

Para la implantación, es muy conveniente que apoye a la organización una empresa de consultoría, que tenga buenas referencias, y el firme compromiso de la Dirección de que quiere implantar el Sistema.

1.10.2 ISO

International Organization for Standardization, es una organización internacional no gubernamental, compuesta por representantes de los organismos de normalización (ONs) nacionales, que produce normas internacionales industriales y comerciales estándares.



1.10.3 Estándares IEEE

Los estándares IEEE están orientados al aseguramiento de la calidad a nivel del proyecto.

Existen también estándares para otras actividades relacionadas con la calidad como pruebas, verificación y validación, revisiones, etc. Los principales se recogen en la siguiente tabla.

IEEE 730 2002	-	Planes de aseguramiento de calidad de software
IEEE 829 1998	-	Documentación de pruebas del software
IEEE 982.1,982.2		Diccionario estándar de medidas para producir software confiable
IEE E1008 1987	-	Pruebas de unidad del software
IEEE 1012 1998	-	Verificación y validación del software
IEEE 1028 1997	-	Revisiones del software
IEEE 1044 1993	-	Clasificación estándar de anomalías del software
IEEE 1061 1992	-	Standard for a software quality metrics methodology
IEE E 1074 1995	-	Standard for developing software life cycle processes
IEE E 1 228 1994	-	Planes de seguridad del software

1.11 MODELOS DE CALIDAD

Como se menciona en el apartado anterior, los Modelos de Calidad ayudan a definir y poner en marcha el concepto general de Calidad de Software. Los Modelos de Calidad se definen de forma jerárquica es un concepto que se deriva de un conjunto de sub-conceptos, cada uno de los cuales se va a evaluar a través de un conjunto de indicadores o métricas. Existen algunos Modelos de calidad entre los más conocidos los Modelo Mc Call, ISO 9126, IEEE, Boehm, GQM, Gilb, entre otros. La ventaja de los modelos de calidad es que la calidad se convierte en algo concreto, que se puede definir, medir y sobre todo planificar.

1.12 AUDITORÍA:

La auditoría nace como un órgano de control de algunas instituciones estatales y privadas. Su función inicial es estrictamente económico-financiera, y los casos inmediatos se encuentran en las peritaciones judiciales y las contrataciones de contables expertos por parte de Bancos Oficiales.

La función auditora debe ser absolutamente independiente; no tiene carácter ejecutivo, ni son vinculantes sus conclusiones. Queda a cargo de la empresa tomar las decisiones pertinentes. La auditoría contiene elementos de análisis, de verificación y de exposición de debilidades y disfunciones. Aunque pueden aparecer sugerencias y planes de acción para eliminar las disfunciones y debilidades antedichas; estas sugerencias plasmadas en el Informe final reciben el nombre de Recomendaciones.

Las funciones de análisis y revisión que el auditor informático realiza, puede chocar con la psicología del auditado, ya que es un informático y tiene la necesidad de realizar sus tareas con racionalidad y eficiencia. La reticencia del auditado es comprensible y, en ocasiones, fundada. El nivel técnico del auditor es

a veces insuficiente, dada la gran complejidad de los Sistemas, unidos a los plazos demasiado breves de los que suelen disponer para realizar su tarea.

Además del chequeo de los Sistemas, el auditor somete al auditado a una serie de cuestionario. Dichos cuestionarios, llamados Check List, son guardados celosamente por las empresas auditoras, ya que son activos importantes de su actividad. Las Check List tienen que ser comprendidas por el auditor al pie de la letra, ya que si son mal aplicadas y mal recitadas se pueden llegar a obtener resultados distintos a los esperados por la empresa auditora. La Check List puede llegar a explicar cómo ocurren los hechos pero no por qué ocurren. El cuestionario debe estar subordinado a la regla, a la norma, al método. Sólo una metodología precisa puede desentrañar las causas por las cuales se realizan actividades teóricamente inadecuadas o se omiten otras correctas.

El auditor sólo puede emitir un juicio global o parcial basado en hechos y situaciones incontrovertibles, careciendo de poder para modificar la situación analizada por él mismo.

1.12.1 AUDITORÍA INTERNA Y AUDITORÍA EXTERNA:

La auditoría interna es la realizada con recursos materiales y personas que pertenecen a la empresa auditada. Los empleados que realizan esta tarea son remunerados económicamente. La auditoría interna existe por expresa decisión de la Empresa, o sea, que puede optar por su disolución en cualquier momento.

Por otro lado, la auditoría externa es realizada por personas afines a la empresa auditada; es siempre remunerada. Se presupone una mayor objetividad que en la Auditoría Interna, debido al mayor distanciamiento entre auditores y auditados.

La auditoría informática interna cuenta con algunas ventajas adicionales muy importantes respecto de la auditoría externa, las cuales no son tan perceptibles como en las auditorías convencionales. La auditoría interna tiene la ventaja de que puede actuar periódicamente realizando Revisiones globales, como parte de su Plan Anual y de su actividad normal. Los auditados conocen estos planes y se habitúan a las Auditorías, especialmente cuando las consecuencias de las Recomendaciones habidas benefician su trabajo.

En una empresa, los responsables de Informática escuchan, orientan e informan sobre las posibilidades técnicas y los costes de tal Sistema. Con voz, pero a menudo sin voto, Informática trata de satisfacer lo más adecuadamente posible aquellas necesidades. La empresa necesita controlar su Informática y ésta necesita que su propia gestión esté sometida a los mismos Procedimientos y estándares que el resto de aquella. La conjunción de ambas necesidades cristaliza en la figura del auditor interno informático.

En cuanto a empresas se refiere, solamente las más grandes pueden poseer una Auditoría propia y permanente, mientras que el resto acuden a las auditorías externas. Puede ser que algún profesional informático sea trasladado desde su puesto de trabajo a la Auditoría Interna de la empresa cuando ésta existe. Finalmente, la propia Informática requiere de su propio grupo de Control Interno, con implantación física en su estructura, puesto que si se ubicase dentro de la estructura Informática ya no sería independiente. Hoy, ya existen varias organizaciones Informáticas dentro de la misma empresa, y con diverso grado de autonomía, que son coordinadas por órganos corporativos de Sistemas de Información de las Empresas.

Una Empresa o Institución que posee auditoría interna puede y debe en ocasiones contratar servicios de auditoría externa. Las razones para hacerlo suelen ser:

- Necesidad de auditar una materia o asunto de gran especialización, para la cual los servicios propios no están suficientemente capacitados.
- Contrastar algún Informe interno con los resultados del externo, en aquellos supuestos de emisión interna de graves recomendaciones que chocan con la opinión generalizada de la propia empresa.
- Servir como mecanismo protector de posibles auditorías informáticas externas decretadas por la misma empresa.
- Aunque la auditoría interna sea independiente del Departamento de Sistemas, sigue siendo la misma empresa, por lo tanto, es necesario que se le realicen auditorías externas como para tener una visión desde afuera de la empresa.

La auditoría informática, tanto externa como interna, debe ser una actividad exenta de cualquier contenido o matiz "político" ajeno a la propia estrategia y política general de la empresa. La función auditora puede actuar de oficio, por iniciativa del propio órgano, o a instancias de parte, esto es, por encargo de la dirección o cliente.

1.12.2 ALCANCE DE LA AUDITORÍA INFORMÁTICA:

El alcance ha de definir con precisión el entorno y los límites en que va a desarrollarse la auditoría informática, a fin de que se complemente con los objetivos de ésta. El alcance ha de figurar expresamente en el Informe Final, de

modo que quede perfectamente determinado no solamente hasta que puntos se ha llegado, sino cuales materias fronterizas han sido omitidas.

1.12.3 CONTROL DE INTEGRIDAD DE REGISTROS:

Hay Aplicaciones que comparten registros, es decir; son registros comunes. Si una Aplicación no tiene integrado un registro común, cuando lo necesite utilizar no lo va a encontrar y, por lo tanto, la aplicación no funcionara como debería.

1.12.4 CONTROL DE VALIDACIÓN DE ERRORES:

Debe asegurarse que el sistema que se aplica para detectar y corregir errores sea eficiente.

1.12.5 CARACTERÍSTICAS DE LA AUDITORÍA INFORMÁTICA:

La información de la empresa y para la empresa, siempre importante, se ha convertido en un Activo Real de la misma, como sus Stocks o materias primas si las hay. Por ende, han de realizarse inversiones informáticas, materia de la que se ocupa la Auditoría de Inversión Informática.

Del mismo modo, los Sistemas Informáticos han de protegerse de modo global y particular: a ello se debe la existencia de la Auditoría de Seguridad Informática en general, o a la auditoría de Seguridad de alguna de sus áreas, como pudieran ser Desarrollo o Técnica de Sistemas.

1.12.6 SÍNTOMAS DE NECESIDAD DE UNA PRUEBA DE EFICIENCIA:

Estos síntomas pueden agruparse en clases:

Síntomas de descoordinación y desorganización:

- 1.- No coinciden los objetivos de la Informática del sistema de la institución.

- 2.- Los estándares de productividad se desvían sensiblemente de los promedios conseguidos habitualmente.

Síntomas de mala imagen e insatisfacción de los usuarios:

- 1.- No se atienden las peticiones de cambios de los usuarios. Ejemplos: cambios de Software en los terminales de usuario, refrescamiento de paneles, variación de los ficheros que deben ponerse diariamente a su disposición, etc.

- 2.- No se cumplen en todos los casos los plazos de entrega de resultados periódico exactos del sistema.

- **Síntomas de Inseguridad: Evaluación de nivel de riesgos**

- 1.- Seguridad Lógica
- 2- Seguridad Física
- 3.-Confidencialidad

- Continuidad del Servicio. Es un concepto aún más importante que la Seguridad. Establece las estrategias de continuidad entre fallos mediante Planes de Contingencia Totales y Locales del mismo sistema.
- Centro de Proceso de Datos fuera de control. Esa es la razón por la cual, en este caso, el síntoma debe ser sustituido por el mínimo indicio de resultados de la prueba de caja negra.

1.12.7 FALLAS PREDECIBLES A LA INSTALACIÓN DEL SOFTWARE

Todas las Aplicaciones que se desarrollan son super-parametrizadas, es decir, que tienen un montón de parámetros que permiten configurar cual va a ser el comportamiento del Sistema. Una Aplicación va a usar para una y otra cosa cierta cantidad de espacio en el disco. Si no se analizó cual es la operatoria y el tiempo que le va a llevar ocupar el espacio asignado, y se pone un valor muy pequeño, puede ocurrir que un día la Aplicación reviente, se caiga. Si esto sucede en medio de la operatoria y la Aplicación se cae, el volver a levantarla, con la nueva asignación de espacio, si hay que hacer reconversiones o lo que sea, puede llegar a demandar muchísimo tiempo, lo que significa un riesgo enorme.

1.13 GESTIÓN INFORMÁTICA:

Una vez conseguida la Operatividad del Sistemas, el segundo objetivo es la verificación de la observancia de las normas teóricamente existentes en el departamento de Informática así como el departamento administrativo . Para ello, habrán de revisarse sucesivamente y en este orden:

- .1 Las Normas Generales de la Instalación Informática. Se realizará una revisión inicial sin estudiar a fondo las contradicciones que pudieran existir, pero registrando las áreas que carezcan de normativa, y sobre todo verificando que esta Normativa General Informática no está en contradicción con alguna Norma General no informática de la empresa.

- .2 Los Procedimientos Generales Informáticos. Se verificará su existencia, al menos en los sectores más importantes. Por ejemplo, la recepción definitiva de las máquinas debería estar firmada por los responsables de Explotación. Tampoco el alta de una nueva Aplicación podría producirse si no existieran los Procedimientos de Backup y Recuperación correspondientes.

- .3 Los Procedimientos Específicos Informáticos. Igualmente, se revisara su existencia en las áreas fundamentales. Así, Explotación no debería explotar una Aplicación sin haber exigido a Desarrollo la pertinente documentación. Del mismo modo, deberá comprobarse que los Procedimientos Específicos no se opongan a los Procedimientos Generales. En todos los casos anteriores, a su vez, deberá verificarse que no exista contradicción alguna con la Normativa y los Procedimientos Generales de la propia empresa, a los que la Informática debe estar sometida.

1.13.1 CONTROL DE ENTRADA DE DATOS:

Se analizará la captura de la información en soporte compatible con los Sistemas, el cumplimiento de plazos y calendarios de tratamientos y entrega de datos; la correcta transmisión de datos entre entornos diferentes. Se verificará que los controles de integridad y calidad de datos se realizan de acuerdo a Norma.

1.13.2 PLANIFICACIÓN Y RECEPCIÓN DE APLICACIONES:

Se auditarán las normas de entrega de Aplicaciones por parte de Desarrollo, verificando su cumplimiento y su calidad de interlocutor único. Deberán realizarse muestreos selectivos de la Documentación de las Aplicaciones explotadas. Se inquirirá sobre la anticipación de contactos con Desarrollo para la planificación a mediano y largo plazo.

1.13.3 CENTRO DE CONTROL Y SEGUIMIENTO DE TRABAJOS:

Se analizará cómo se prepara, se lanza y se sigue la producción diaria. Básicamente, la explotación Informática ejecuta procesos por cadenas o lotes sucesivos (Batch), o en tiempo real (Tiempo Real). Mientras que las Aplicaciones de Teleproceso están permanentemente activas y la función de Explotación se limita a vigilar y recuperar incidencias, el trabajo Batch absorbe una buena parte de los efectivos de Explotación. En muchos Centros de Proceso de Datos, éste órgano recibe el nombre de Centro de Control de Batch. Este grupo determina el éxito de la explotación, en cuanto que es uno de los factores más importantes en el mantenimiento de la producción.

Batch y Tiempo Real:

Las Aplicaciones que son Batch son Aplicaciones que cargan mucha información durante el día y durante la noche, se corre un proceso enorme y lo que hace es relacionar toda la información, calcular cosas y obtener como salida, por ejemplo, reportes. O sea, recolecta información durante el día, pero todavía no procesa nada. Es solamente un tema de

"Data Entry" que recolecta información, corre el proceso Batch (por lotes), y calcula todo lo necesario para arrancar al día siguiente.

Las Aplicaciones que son Tiempo Real u Online, son las que, luego de haber ingresado la información correspondiente, inmediatamente procesan y devuelven un resultado. Son Sistemas que tienen que responder en Tiempo Real.

1.14 DESARROLLO DE PROYECTOS O APLICACIONES:

La función de Desarrollo es una evolución del llamado Análisis y Programación de Sistemas y Aplicaciones. A su vez, engloba muchas áreas, tantas como sectores informatizables tiene la empresa. Muy escuetamente, una Aplicación recorre las siguientes fases:

- Prerequisitos del Usuario (único o plural) y del entorno
- Análisis funcional
- Diseño
- Análisis orgánico (Preprogramación y Programación)
- Pruebas
- Entrega a Explotación y alta para el Proceso.

Estas fases deben estar sometidas a un exigente control interno, caso contrario, además del disparo de los costes, podrá producirse la insatisfacción del usuario. Finalmente, la auditoría deberá comprobar la seguridad de los programas en el sentido de garantizar que los ejecutados por la máquina sean exactamente los previstos y no otros.

Una auditoría de Aplicaciones pasa indefectiblemente por la observación y el análisis de cuatro consideraciones:

- Revisión de las metodologías utilizadas:
- Control Interno de las Aplicaciones:
- Estudio de Vialidad de la Aplicación.
- Definición Lógica de la Aplicación.
- Desarrollo Técnico de la Aplicación
- Diseño de Programas.

1.14.1 PRUEBA INFORMÁTICA DE SISTEMAS:

Se ocupa de analizar la actividad que se conoce como Técnica de Sistemas en todas sus facetas. Hoy, la importancia creciente de las telecomunicaciones ha propiciado que las Comunicaciones, Líneas y Redes de las instalaciones informáticas así como los sistemas se verifiquen por separado, aunque formen parte del entorno general de Sistemas

1.15 SISTEMAS OPERATIVOS:

Debe verificarse en primer lugar que los Sistemas estén actualizados con las últimas versiones del fabricante, indagando las causas de las omisiones si las hubiera. El análisis de las versiones de los Sistemas Operativos permite descubrir las posibles incompatibilidades entre otros productos de Software Básico adquiridos por la instalación y determinadas versiones de aquellas. Deben revisarse los parámetros variables de las Librerías más importantes de los Sistemas, por si difieren de los valores habituales aconsejados por el constructor.

1.15.1 ADMINISTRACIÓN DE BASE DE DATOS:

El diseño de las Bases de Datos, sean relaciones o jerárquicas, se ha convertido en una actividad muy compleja y sofisticada, por lo general desarrollada en el ámbito de Técnica de Sistemas, y de acuerdo con las áreas de Desarrollo y usuarios de la empresa. Al conocer el diseño y arquitectura de éstas por parte de Sistemas, se les encomienda también su administración. Los auditores de Sistemas han observado algunas disfunciones derivadas de la relativamente escasa experiencia que Técnica de Sistemas tiene sobre la problemática general de los usuarios de Bases de Datos.

La administración tendría que estar a cargo de Explotación. El auditor de Base de Datos debería asegurarse que Explotación conoce suficientemente las que son accedidas por los Procedimientos que ella ejecuta. Analizará los Sistemas de salvaguarda existentes, que competen igualmente a Explotación. Revisará finalmente la integridad y consistencia de los datos, así como la ausencia de redundancias entre ellos.

1.15.2 INVESTIGACIÓN Y DESARROLLO:

Las empresas que utilizan y necesitan de informáticas desarrolladas, saben que sus propios efectivos están desarrollando Aplicaciones y utilidades que, concebidas inicialmente para su uso interno, pueden ser susceptibles de adquisición por otras empresas, haciendo competencia a las Compañías del ramo.

1.15.3 EVALUACION INFORMÁTICA DE COMUNICACIONES Y REDES QUE FORMAN PARTE DEL SISTEMA:

Para el informático, el entramado conceptual que constituyen las Redes Nodales, Líneas, Concentradores, Multiplexores, Redes Locales, etc. no son sino el soporte físico-lógico del Tiempo Real. Que son las terminales para cierto proceso del entorno grafico de las pantallas para la obtención de datos del sistema

1.15.4 SEGURIDAD INFORMÁTICA:

La computadora es un instrumento que estructura gran cantidad de información, la cual puede ser confidencial para individuos, empresas o instituciones, y puede ser mal utilizada o divulgada a personas que hagan mal uso de esta. También puede ocurrir robos, fraudes o sabotajes que provoquen la destrucción total o parcial de la actividad computacional. Esta información puede ser de suma importancia, y el no tenerla en el momento preciso puede provocar retrasos sumamente costosos.

En la actualidad y principalmente en las computadoras personales, se ha dado otro factor que hay que considerar: el llamado "virus" de las computadoras, el cual, aunque tiene diferentes intenciones, se encuentra principalmente para paquetes que son copiados sin autorización ("piratas") y borra toda la información que se tiene en un disco. Al auditar los sistemas se debe tener cuidado que no se tengan copias "piratas" o bien que, al conectarnos en red con otras computadoras, no exista la posibilidad de transmisión del virus.

1.15.5 HERRAMIENTAS Y TÉCNICAS PARA LA EVALUACIÓN INFORMÁTICA:

- **Cuestionarios:**

Las auditorías informáticas se materializan recabando información y documentación de todo tipo. Los informes finales de los auditores dependen de sus capacidades para analizar las situaciones de debilidad o fortaleza de los diferentes entornos. El trabajo de campo del auditor consiste en lograr toda la información necesaria para la emisión de un juicio global objetivo, siempre amparado en hechos demostrables, llamados también evidencias.

Para esto, suele ser lo habitual comenzar solicitando la complementación de cuestionarios preimpresos que se envían a las personas concretas que el auditor cree adecuadas, sin que sea obligatorio que dichas personas sean las responsables oficiales de las diversas áreas a auditar.

Estos cuestionarios no pueden ni deben ser repetidos para instalaciones distintas, sino diferentes y muy específicos para cada situación, y muy cuidados en su fondo y su forma.

Sobre esta base se estudia y analiza la documentación recibida, de modo que tal análisis determine a su vez la información que deberá elaborar el propio auditor. El cruzamiento de ambos tipos de información es una de las bases fundamentales de la auditoría.

Cabe aclarar que, esta primera fase puede omitirse cuando los auditores hayan adquirido por otro medios la información que aquellos preimpresos hubieran proporcionado.

- **Entrevistas:**

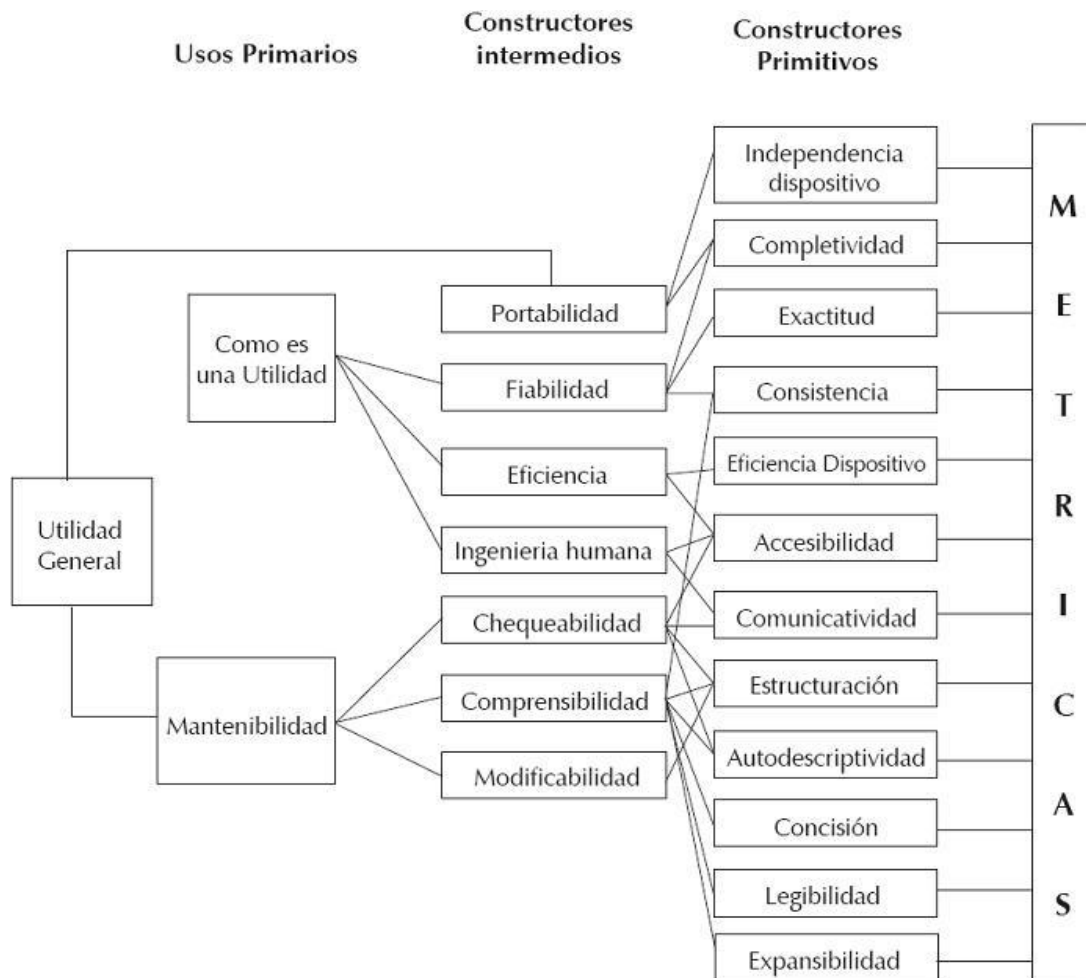
El auditor comienza las relaciones personales con el auditado. Lo hace de tres formas:

- Mediante la petición de documentación concreta sobre alguna materia de su responsabilidad.
- Mediante "entrevistas" en las que no se sigue un plan predeterminado ni un método estricto de sometimiento a un cuestionario.
- Por medio de entrevistas en las que el auditor sigue un método preestablecido de antemano y busca unas finalidades concretas.

La entrevista es una de las actividades personales más importante del auditor; en ellas, éste recoge más información y mejor matizada, que la proporcionada por medios propios puramente técnicos o por las respuestas escritas a cuestionarios.

Aparte de algunas cuestiones menos importantes, la entrevista entre auditor y auditado se basa fundamentalmente en el concepto de interrogatorio; es lo que hace un auditor, interroga y se interroga a sí mismo. El auditor informático experto entrevista al auditado siguiendo un cuidadoso sistema previamente establecido, consistente en que; bajo la forma de una conversación correcta y lo menos tensa posible, el auditado conteste sencillamente y con pulcritud a una serie de preguntas variadas, también sencillas. Sin embargo, esta sencillez es solo aparente. Tras ella debe existir una preparación muy elaborada y sistematizada, y que es diferente para cada caso particular.

1.16 GRAFICA DE LA MÉTRICA DE UN SISTEMA



CAPITULO II

2.- Trabajo de campo

2.1 RECOPIACIÓN DE LA INFORMACIÓN DE CAMPO

2.1.2PRESENTACION

Las técnicas que se aplicaron para la presente investigación fueron encuestas, las mismas que fueron diseñadas con preguntas de diferente tipo, de acuerdo a cada uno de los sectores involucrados, como son: personal administrativo y docentes, procurando obtener información que permita realizar adecuadamente un diagnóstico del sistema escolástico, que actualmente se encuentra en funcionamiento en el colegio Primero de Abril. El mismo instrumento fue aplicado para los directivos de la institución educativa, con preguntas orientadas a conocer cuál es el criterio que tienen sobre la importancia y la necesidad de la implementación de las pruebas de caja negra y caja blanca, en los actuales momentos.

Luego de haber aplicado los diferentes instrumentos a los directivos y docentes, ponemos a consideración los siguientes resultados, los mismos que serán

analizados y representados gráficamente, para luego extraer las respectivas conclusiones y recomendaciones.

2.2 Análisis e interpretación de resultados de las encuestas realizadas a los DOCENTES del colegio Nacional Primero de Abril.

De los 90 docentes existentes en el Colegio Nacional Primero de Abril en un 67% se realizaron la encuesta y el 33% restante son encuestas no realizadas por falta de conocimiento de los docentes y en mucho de los casos por la poca colaboración prestada para el avance de la institución.

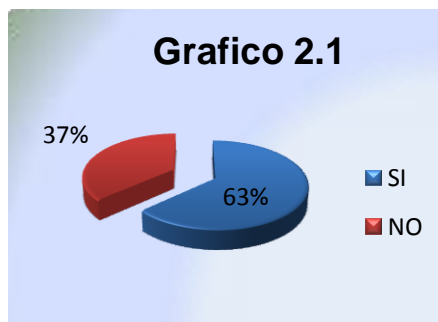
PREGUNTA N°1

¿Usted tiene conocimiento sobre el funcionamiento del sistema Escolástico?

TABLA N° 2.1

Conocimiento que tienen los docentes sobre el funcionamiento del sistema escolástico

Denominación	Frecuencia	Porcentaje
SI	38	63%
NO	22	37%
TOTAL	60	100%



Fuente: Encuesta

Elaborado por: Las Autoras

De las encuestas realizadas a 60 docentes del Colegio Nacional Primero De Abril: 38 encuestados que equivalen al 63%, manifestaron que conocen el funcionamiento del sistema escolástico mientras que 22 docentes que equivalen al 37% no tienen conocimiento de las funciones del sistema escolástico de la institución.

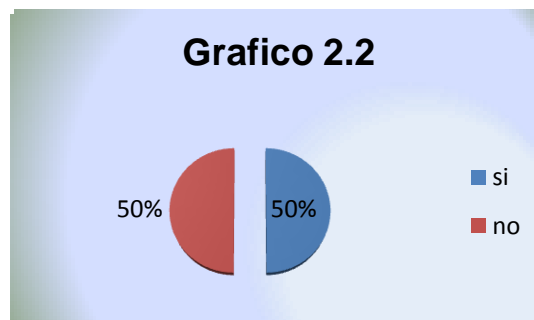
PREGUNTA N°2

¿Usted tiene acceso al sistema escolástico?

TABLA N° 2.2

Acceso que tienen los docentes al sistema escolástico

Denominación	Frecuencia	Porcentaje
SI	30	50%
NO	30	50%
Total	60	100%



Fuente: Encuesta

Elaborado por: Las Autoras

De los 60 docentes encuestados 30 maestros, que equivale al 50%, manifiestan que tienen acceso al sistema escolástico mientras que 30 docentes que equivalen al 50%, manifestaron que no tienen acceso al sistema escolástico.

PREGUNTA N°3

¿Qué opina del sistema escolástico que existe en la actualidad?

Opiniones sobre el sistema escolástico existente en la institución.

Al aplicar las encuestas en la institución educativa muchos de los docentes supieron manifestar que el sistema escolástico está incompleto, necesita cambios, solicitan que sea actualizado debido a que tiene muchas falencias y funciona a

medias, especialmente en lo que se refiere a notas de conducta, y al finalizar el año falta el cuadro total de calificaciones; además, manifiestan que el tiempo de pase de notas no es el adecuado y que el sistema escolástico lleve un control correcto de los alumnos que ingresan y se retiran de la institución. Los maestros en su gran mayoría opinan que necesitan capacitación para poder manejar el sistema y de esta manera no sea tan restringido para que la institución logre tener avances significativos.

Fuente: Encuesta

Elaborado por: Las Autoras

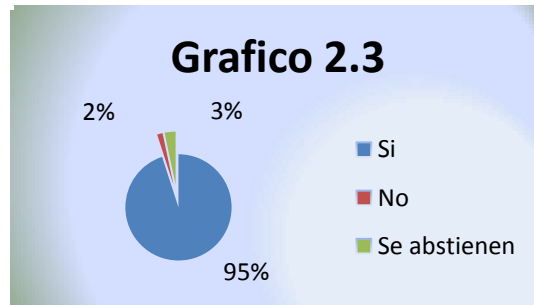
PREGUNTA N°4

¿Cree usted que el sistema escolástico debe ir acorde con los nuevos avances tecnológicos?

TABLA N° 2.3

Nivel de conocimiento e importancia de conocer si el sistema escolástico debe ir acorde a la tecnología actual

Denominación	Frecuencia	Porcentaje
Si	57	95%
No	1	2%
Se abstienen	2	3%
Total	60	100%



Fuente: Encuesta

Elaborado por: Las Autoras

De los 60 encuestados, 57 docentes, que equivale al 95% manifestaron que el sistema escolástico debe ir acorde con los avances tecnológicos para brindar una información rápida y oportuna, mientras que 1 maestro sugiere que no debe ir acorde a la tecnología actual, 2 docentes, que equivale al 3%, se abstienen de dar una respuesta.

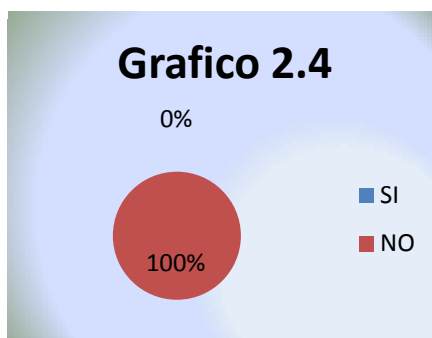
PREGUNTA N°5

¿Usted ha escuchado acerca de las pruebas de caja negra y caja blanca?

TABLA N° 2.4

Nivel de conocimiento de los docentes sobre las pruebas de caja negra y caja blanca

Denominación	Frecuencia	Porcentaje
SI	0	0%
NO	60	100%
Total	60	100



Fuente: Encuesta

Elaborado por: Las Autoras

Al aplicar las encuestas en la institución educativa podemos manifestar que, en un 100% de los docentes, desconocen del tema de caja negra caja blanca, debido a que es un sistema de pruebas nuevo; es por tal razón que creemos necesario su implementación.

PREGUNTA N°6

¿A qué áreas del sistema escolástico tiene usted acceso?

TABLA N° 2.5

Acceso de los docentes al sistema escolástico

Denominación	Frecuencia	Porcentaje
Área de notas	58	97%
Se abstienen	2	3%
Total	60	100%



Fuente: Encuesta

Elaborado por: Las Autoras

De los 60 encuestados, 58 profesores, que equivale al 97% de los docentes, tienen un acceso limitado al sistema escolástico; es decir, simplemente pueden acceder al área de pase de notas académicas y de disciplina; cada maestro solo accede al área de su asignatura y dos docentes, que corresponden al 3%, se abstienen de responder.

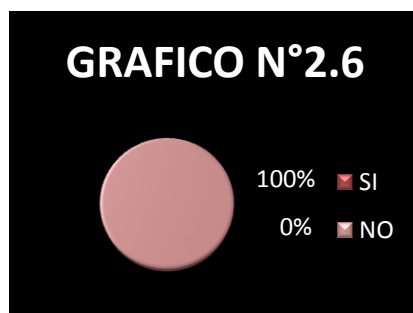
PREGUNTA N°7

¿Le gustaría conocer el funcionamiento del sistema escolástico?

TABLA N° 2.6

Interés de los docentes por conocer el funcionamiento del sistema escolástico

Denominación	Frecuencia	Porcentaje
SI	60	100%
NO	0	0%
Total	60	100%



Fuente: Encuesta

Elaborado por: Las Autoras

De los 60 encuestados, el 100% de los docentes, manifestaron que creen necesario conocer el funcionamiento del sistema escolástico para poder tener acceso seguro a todas las áreas.

PREGUNTA N°8

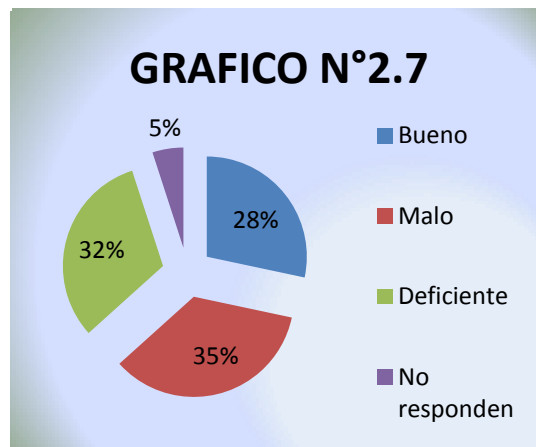
¿Cómo califica usted el funcionamiento del sistema escolástico?

TABLA N° 2.8

Evaluación al funcionamiento del sistema escolástico.

Denominación	Frecuencia	Porcentaje
Bueno	17	28%
Malo	21	35%
Deficiente	19	32%
No responden	3	5%

TOTAL	60	100%
--------------	----	------



Fuente: Encuesta

Elaborado por: Las Autoras

De los 60 encuestados, 17 docentes que equivalen al 28% manifestaron que el sistema escolarístico es bueno, mientras que 21 maestros, que equivalen al 35% consideran que el sistema escolarístico no funciona correctamente que es malo; 19 encuestados, que equivale al 32% sugieren que el sistema escolarístico es deficiente; por otra parte 3 de los encuestados, que equivale al 5% decidieron no dar respuesta alguna.

2.3 Análisis e interpretación de resultados de las encuestas realizadas a los ADMINISTRATIVOS del colegio Nacional Primero de Abril.

De los 10 administrativos existentes en el Colegio Nacional Primero de Abril en un 20% se realizaron las encuestas ya que solo estas 2 personas encuestadas tenían pleno conocimiento del funcionamiento del sistema .

NOTA.- en virtud de lo mencionado anteriormente tomaremos en cuenta a las dos personas como el 100% debido a que son los únicos que tiene conocimiento del funcionamiento del sistema y podrán aportar positivamente al desarrollo de nuestra investigación.

PREGUNTA N°1

¿Considera usted que el sistema escolástico es eficiente?

De los 02 administrativos encuestados, manifiestan que el sistema escolástico no es eficiente debido a que existen muchos problemas, especialmente en el ingreso de calificaciones tanto de disciplina como académicas y en la emisión de todo tipo de reportes, además de esto es un sistema que trabaja a la mitad de su requerimiento. Por esta razón el personal administrativo no se encuentra satisfecho y solicitan la actualización y mejora del sistema así como de capacitación permanente.

Fuente: Encuesta

Elaborado por: Las Autoras

PREGUNTA N°2

¿Qué grado de dificultad considera usted que tiene el sistema escolástico?

De los 02 administrativos encuestados, uno de los encuestados manifiesta que el sistema escolástico tiene un grado de dificultad bajo debido a que es fácil de manejar, pero respecto a la programación no tienen conocimiento, mientras que la otra persona encuestada opina que el grado de dificultad es medio, debido a que el sistema escolástico no está completo.

Fuente: Encuesta

Elaborado por: Las Autoras

PREGUNTA N°3

¿Conoce si se ha realizado algún tipo de pruebas de software en el sistema escolástico actual?

Los encuestados manifiestan que en el sistema escolástico no se han implementado ningún tipo de pruebas, es por esto que esta tesis aportaría positivamente al desarrollo del sistema.

Fuente: Encuesta

Elaborado por: Las Autoras

PREGUNTA N°4

¿Qué cambios plantearía usted para el sistema escolástico?

Las dos personas encuestadas manifiestan que los cambios necesarios deberían ser necesariamente en el software del sistema y su actualización.

Fuente: Encuesta

Elaborado por: Las Autoras

PREGUNTA N°5

¿Considera importante realizar cambios en el sistema escolástico?

Los 02 administrativos encuestados opinan que es importante realizar cambios en el sistema escolástico para que de esta manera podamos satisfacer las necesidades tanto de autoridades como de docentes.

Fuente: Encuesta

Elaborado por: Las Autoras

PREGUNTA N°6

¿Usted ha escuchado acerca de las pruebas de caja negra, caja blanca?

Las 02 personas encuestadas manifiestan que no tienen conocimiento sobre las pruebas de software (pruebas de caja negra y caja blanca).

Fuente: Encuesta

Elaborado por: Las Autoras

PREGUNTA N°7

¿Cree usted que las pruebas que se apliquen al sistema escolástico ayuden al mejoramiento del mismo?

Los 02 administrativos encuestados, manifiestan que las pruebas si ayudarían al desarrollo del sistema escolástico

Fuente: Encuesta

Elaborado por: Las Autoras

PREGUNTA N°8

¿Cómo califica el funcionamiento del sistema escolástico?

Los 02 administrativos encuestados, opinan que el sistema es bueno pero necesita algunos cambios porque tiene varias falencias en la interfaz de usuario y en reportes.

Fuente: Encuesta

Elaborado por: Las Autoras

PREGUNTA N°9

¿A través de qué actividades realizadas conoce usted el sistema escolástico?

De los 02 administrativos encuestados, los mismos expresan que si hay acceso al sistema a través de módulos de calificaciones y de los reportes.

Fuente: Encuesta

Elaborado por: Las Autoras

PREGUNTA N°10

¿Usted ha escuchado los beneficios que producen las pruebas de software?

Los 02 encuestados supieron decir que no conocen los beneficios que pueden ofrecer las pruebas de software, pero creen necesario la implementación de dichas pruebas.

Fuente: Encuesta

Elaborado por: Las Autoras

PREGUNTA N°11

¿El sistema escolástico cuenta con manuales de usuario?

Los encuestados aseguran que no existe ningún tipo de manual, tomando en cuenta que la existencia de dichos manuales es de gran utilidad especialmente para el personal administrativo.

Fuente: Encuesta

Elaborado por: Las Autoras

PREGUNTA N°12

¿Los problemas más frecuentes en el sistema escolástico son en?

De los 02 administrativos encuestados, los mismos manifiestan que los cambios necesarios debería ser necesariamente en el software del sistema debido a que existen falencias en la programación y esto no permite que las funciones se realicen satisfactoriamente.

Fuente: Encuesta

Elaborado por: Las Autoras

PREGUNTA N°13

¿La plataforma informática instalada en el sistema escolástico de qué forma satisface los requerimientos de los usuarios?

De los 02 administrativos encuestados, ellos manifiestan que la plataforma instalada funciona correctamente y la califican como muy buena.

Fuente: Encuesta

Elaborado por: Las Autoras

PREGUNTA N°14

¿Tienen algún convenio con empresas proveedoras para garantizar la calidad del sistema escolástico?

De los 02 administrativos encuestados, los 02 aseguran que no existe ningún convenio con la empresa creadora del programa ni con otra empresa.

Fuente: Encuesta

Elaborado por: Las Autoras

2.4 ANALISIS

Con los porcentajes obtenidos en cada una de las preguntas efectuadas a los administrativos y docentes del colegio Nacional Primero de Abril, podemos manifestar que las funciones que realiza el sistema escolástico son incompletas y en algunos los casos son erróneas, por lo que proponemos que el sistema sea actualizado con los avances tecnológicos, en función de las políticas de la institución, para ofrecer información oportuna e inmediata a los docentes y de una forma fácil y más segura a los administrativos.

Las opiniones y sugerencias de los administrativos y docentes son valiosas ya que el Colegio Nacional Primero de Abril necesita obtener un sistema escolástico eficiente y con un margen mínimo de errores. Ya que esta es la única forma que la institución educativa puede emitir las actividades que se realiza en cuanto a notas y reportes.

Además de esto es de gran beneficio contar con un sistema escolástico porque nos permite minimizar tiempo y facilitar el trabajo tanto a docentes como a secretaria y así poder cumplir con un pase de notas rápido confiable y con un tiempo adecuado.

Podemos observar claramente que la mayoría de los docentes no conocen el funcionamiento del sistema escolástico y el acceso al mismo es muy restringido es por esta razón que la institución educativa no ha logrado avances significativos.

En cuanto a los administrativos manifiestan que conocen el funcionamiento del sistema pero carecen de conocimientos en lo que corresponde a la programación además de esto no cuentan con manual alguno que les pueda ayudar para salir de sus dudas al ingresar al programa escolástico.

Ante estos resultados se observa claramente la gran acogida a la propuesta presentada por las investigadoras “Implementación de Pruebas Caja Negra, Caja Blanca aplicables al Sistema Escolástico del Colegio Nacional “Primero de Abril” de la Ciudad de Latacunga Provincia de Cotopaxi” en el período Octubre 2008 – Agosto 2009” que mejore la calidad de los servicios y promueva al desarrollo educativo.

2.5 CONCLUSIONES

- Con los resultados obtenidos de las encuestas realizadas a los docentes y administrativos, las investigadoras manifiestan que hay una gran acogida para el desarrollo de la propuesta ya mencionada, ante la imperiosa necesidad de tener un sistema escolástico actual y con un margen mínimo de errores que permita desarrollar las actividades eficientemente en cuanto a pase de notas académicas y disciplina, que vaya acorde a los intereses de los docentes y administrativos por lo que es factible la realización de una implementación de pruebas caja negra caja blanca.

- El sistema escolástico implementado en cuanto a facilidad de uso tiene un entorno amigable pero se accede través de claves de usuario.

- El software necesita de cambios en cuanto a calificaciones y reportes debido a que en ellos se presentan errores notables y el mismo trabaja a mitad de sus requerimientos.

- Además concluimos que es un sistema extremadamente cerrado debido a que no permite cambios ni modificaciones en su estructura y los docentes solo pueden acceder al modulo de notas y solicitan constante capacitación.
- El sistema no está acorde a las necesidades del personal administrativo ni docente debido a que el tiempo de pase de notas no es el adecuado y además falta el cuadro total de calificaciones.

2.6 RECOMENDACIONES

- Recomendamos que exista capacitación permanente tanto para docentes como para el personal administrativo para el buen manejo del sistema.
- Además que exista más apertura en la utilización del sistema ya que los docentes desean tener más conocimiento sobre el mismo.
- Debe existir un técnico en sistemas de planta para el departamento de administrativo, con el fin de dar seguridad e integridad a la información.

CAPITULO III

3.- Desarrollo del proyecto

3.1 TEMA:

“Implementación de Pruebas Caja Negra y Caja Blanca aplicables al Sistema Escolástico del Colegio Nacional “Primero de Abril” de la Ciudad de Latacunga Provincia de Cotopaxi”

3.1.1 Presentación

El Colegio Nacional “Primero de Abril” al ser una institución educativa que busca el bienestar de docentes y estudiantes aprovecha la potencialidades de los sistemas informáticos, ello implica una nueva realidad y nuevos riesgos.

Por lo tanto el enfoque de este trabajo es diseñar pruebas que sistemáticamente saquen a la luz errores que podría tener un ambiente automatizado. Esto ayuda a que la institución logre adecuar niveles de excelencia en la protección y aprovechamiento de sus datos, en conjunto con los involucrados y así obtener

soluciones adecuadas para optimizar el uso de los recursos informáticos que posee el establecimiento.

La implementación de las pruebas en hardware y en redes en la institución educativa constituye una herramienta para verificar y evaluar el cumplimiento obligatorio de normas, reglamentos, estatutos, etc. En el cual se inclinan los requisitos mínimos de los testings.

Mejorar la eficiencia del servicio informático, disminuir costos, mejorar la relación servicio-usuario y en general mejorar el control del servicio informático del Colegio Nacional Primero De Abril.

3.2. OBJETIVOS

3.2.1 Objetivo General

- Implementar pruebas de caja negra al sistema escolástico del colegio nacional primero de abril para evaluar el funcionamiento del mismo.

3.2.2 Objetivos Específicos

- Mejorar la eficiencia del sistema escolástico para disminuir el tiempo de ejecución.
- Mejorar la relación servicio- usuario para crear un ambiente de trabajo amigable.
- Diseñar un modelo prototipo que indique la estructura correcta de un sistema informático para permitir a las partes responsables de su creación experimentar, probarlo en situaciones reales y explorar su uso.

3.3 IMPLEMENTACIÓN DE LA EVALUACION

3.3 1 PLAN DE PRUEBAS

Un plan de pruebas está constituido por un conjunto de pruebas. Cada prueba debe:

- Dejar claro qué tipo de propiedades se quieren probar (corrección, robustez, fiabilidad, amigabilidad.
- Dejar claro cómo se mide el resultado
- Especificar en qué consiste la prueba (hasta el último detalle de cómo se ejecuta)
- Definir cuál es el resultado que se espera (identificación, tolerancia.

Las pruebas angelicales carecen de utilidad, tanto si no se sabe exactamente lo que se quiere probar, o si no está claro cómo se prueba, o si el análisis del resultado se hace "a ojo".

Estas mismas ideas se suelen agrupar diciendo que un caso de prueba consta de 3 bloques de información:

1. El propósito de la prueba
2. Los pasos de ejecución de la prueba

3. El resultado que se espera

Y todos y cada uno de esos puntos debe quedar perfectamente documentado. Las pruebas de usar y tirar más vale que se tiren directamente, aún antes de usarlas.

Cubrir estos puntos es muy laborioso y, con frecuencia, tedioso, lo que hace desagradable (o al menos muy aburrida) la fase de pruebas. Es mucho más divertido codificar que probar. Tremendo error en el que, no obstante, es fácil incurrir.

Respecto al orden de pruebas, una práctica frecuente es la siguiente:

3. Pasar pruebas de caja negra analizando valores límite. Recuerde que hay que analizar condiciones límite de entrada y de salida.
4. Identificar clases de equivalencia de datos (entrada y salida) y añadir más pruebas de caja negra para contemplar valores normales (en las clases de equivalencia en que estos sean diferentes de los valores límite; es decir, en rangos amplios de valores).
5. Añadir pruebas basadas en "presunción de error". A partir de la experiencia y el sentido común, se aventuran situaciones que parecen proclives a padecer defectos, y se buscan errores en esos puntos. Son pruebas del tipo "¡Me lo temía!"
6. Medir la cobertura de caja blanca que se ha logrado con las fases previas y añadir más pruebas de caja blanca hasta lograr la cobertura deseada.

Normalmente se busca una buena cobertura de ramas (revise los comentarios expuestos al hablar de caja blanca).

✓ Aspectos Sicológicos y Organización del Trabajo

1. Probar es ejercitar un programa para encontrarle fallos. Jamás se debería probar un programa con el ánimo de mostrar que funciona; ese no es el objetivo.
2. Un caso de prueba tiene éxito cuando encuentra un fallo. Lo gracioso no es encontrar un caso en el que el programa funciona perfectamente. Eso es, simplemente, lo normal. Lo guai es encontrar el caso en el que falla.

Las pruebas debe diseñarlas y pasarlas una persona distinta de la que ha escrito el código; es la única forma de no ser "comprensivo con los fallos".

3. Hacer una "obra maestra" cuesta mucho esfuerzo y requiere gran habilidad. Encontrarle fallos a una "obra maestra" cuesta aún más esfuerzo y exige otro tipo de habilidad.
4. Las pruebas no pueden esperar a que esté todo el código escrito para empezar a pasarlas. Deben irse pasando pruebas según se va generando el código para descubrir los errores lo antes posible y evitar que se propaguen a otros módulos. En realidad el nombre "fase de pruebas" es engañoso, pues hay muchas actividades que se desarrollan concurrentemente o, al menos, no se necesita cerrar una fase antes de pasar a la siguiente. Algunos autores llegan al extremo de afirmar que "primero hay que probar y luego codificar". Frase graciosa que se plasma en aspectos mas concretos como que el programa se escriba pensando en que hay que probarlo.

5. Si en un módulo (o sección de un programa, en general) se encuentran muchos fallos, hay que insistir sobre él. Es muy habitual que los fallos se concentren en pequeñas zonas. Hay mil causas para que ocurra este efecto:

- o código escrito por un programador malo

- o código muy difícil

- o código mal o insuficientemente especificado

- o código escrito en un mal día, con prisas, ...

Además, cuanto más se parchea un trozo de código, tanto más ruinoso queda y susceptible a derrumbamientos. A la larga hay que acabar tirándolo y empezando de nuevo.

6. Si se detecta un fallo aislado, puede bastar una corrección aislada. Pero si se detectan muchos fallos en un módulo, lo único práctico es desecharlo, diseñarlo de nuevo, y recodificarlo. La técnica de ir parcheando hasta que se pasan una serie de pruebas es absolutamente suicida y sólo digna del avestruz.

7. Las pruebas pueden encontrar fallos; pero jamás demostrar que no los hay.

Es como las brujas: nadie las ha visto; pero haberlas, haylas.

Ningún programa (no trivial) se ha probado jamás al 100%.

8. Las pruebas también tienen fallos. Los errores son propios de los humanos: todo el mundo se equivoca. Si una prueba falla, hay que revisar tanto lo que

se prueba como lo que lo prueba. No obstante, la experiencia muestra que (casi siempre) hay más fallos el probado que en el probador.

3.3.2 Planificación de las pruebas.

La fase de pruebas, por su envergadura e importancia, necesita una organización seria y fiable. Ante una fase de pruebas se debe tomar como axioma que se van a encontrar errores.

Los componentes de una planificación son:

- **Objetivos:** Definir los objetivos o propósitos de cada fase de las pruebas.
- **Criterios de terminación:** Especificar cuando se deben finalizar las pruebas.
- **Cronología:** Fijar los tiempos necesarios para cada fase diseño, escritura, ejecución.
- **Responsabilidades:** Especificar los responsables de cada fase, así como quién corregirá los errores detectados.
- **Bibliotecas de caso, de prueba y de normas:** Crear técnica de identificación, escritura y almacenamiento de casos de prueba.
- **Herramientas:** Establecer cuáles pueden ser las herramientas de pruebas que se van a utilizar.

- Tiempo de máquina: Determinar el tiempo que se necesita en cada fase del proyecto de prueba.
- Configuración de equipo: Detallar la necesidad de hardware y software especiales de equipo o de un período concreto.
- Integración: Describir el plan de integración del sistema.
- Métodos de seguimiento: Especificar los métodos que se han de utilizar en las pruebas.
- Depuración: Definir un mecanismo para informar sobre los errores detectados, para seguir el proceso de las correcciones y para incorporar éstas al sistema.
- Terminación de las pruebas. Es difícil que el último error detectado era el único que quedaba. Sin embargo, existen métodos para mostrar cuando esta próximo el final. Los dos más comunes son:
 - Terminar la prueba cuando el tiempo establecido para la misma ha expirado.
 - Terminar la prueba cuando todos los casos de prueba se ejecutan sin detectar errores.
- ✓ **Otros métodos más complicados de aplicar pero más efectivos son:**
 - Estimación de número total de errores del programa.

- Estimación del porcentaje de estos errores que pueden encontrarse fácilmente.
- Estimación de qué fracción de errores se originan en procesos particulares de diseño.

Para realizar cualquiera de estas estimaciones es necesario contar con una historia o experiencia previa que permita definir dichas estimaciones.

3.3.3. Principio básico de las pruebas

La prueba del equipo lógico es el método más usado para determinar sistema funciona como debe. El proceso de pruebas es uno de los componentes de un conjunto de actividades que permiten asegurar la calidad del producto realizado.

Uno de los principios básicos en la realización de pruebas es que estas han de ser puestas a prueba por personas distintas a los diseñadores de los programas, tanto para evitar una simple verificación de que el programa funcione correctamente, como para comprobar que ese programa ha sido concebido e interpretado correctamente.

3.3.4 Realización de pruebas

Para cada sistema se realizarán diferentes clases de pruebas:

Pruebas Unitarias: Todos los componentes del sistema deben desarrollarse individualmente para comprobar su correcto funcionamiento.

Pruebas de integración: Corresponde probarse la integración entre los componentes del sistema para demostrar que se puede acoplar correctamente.

Pruebas de sistemas: Prueba el sistema globalmente.

3.3.5 Tipos de pruebas

Existen dos tipos de pruebas:

Pruebas del tipo CAJA BLANCA, que permite examinar la estructura interna del programa.

Pruebas del tipo CAJA NEGRA, donde los casos de prueba se diseñan considerando exclusivamente las entradas y salidas del sistema, sin preocuparse por la estructura interna del mismo.

3.3.6 Desarrollo incremental

Al probar cada módulo individualmente es necesario crear módulos auxiliares que simulen las acciones de los módulos indicados por el módulo que se está probando.

El tipo de prueba incremental consiste en agregar cada módulo o componente individual al conjunto de componentes existentes y el conjunto resultante de prueba. Esto reduce la necesidad de crear módulos conductores y permitiendo además, examinar en detalle las interfaces. Cuando las pruebas unitarias y de integración se realizan separadamente, es difícil examinar los componentes individuales o módulos que causan resultados incorrectos, por el contrario, con el tipo de prueba incremental es probable que surjan al incorporar un nuevo componente a un grupo previamente probado, sean debidos precisamente a este último o a las interfaces entre él y los otros componentes.

3.3.7 Estrategias de integración

Es importante determinar la secuencia en que van a producir e integrar los componentes. Para una estructura de programas organizada jerárquicamente, se podrá enfocar el problema de la prueba utilizando varias estrategias diferentes:

- Estrategias de arriba hacia abajo (Top-Down)
- Estrategias de abajo hacia arriba (Bottom-Up)
- Estrategias combinadas
- Comparación de estrategias.

3.3.8 Pruebas del sistema y de aceptación

a) Pruebas Globales. Una vez que se han probado los componentes individuales y se han integrado, se ha de probar el sistema global. En esta etapa pueden distinguirse los siguientes tipos de prueba, cada uno con un objetivo claramente definido:

- Pruebas funcionales
- Pruebas de comunicación
- Pruebas de rendimiento
- Pruebas de volumen
- Pruebas de sobrecarga
- Pruebas de disponibilidad de datos
- Pruebas de facilidad de uso
- Pruebas de operación

- Pruebas de entorno
- Pruebas de seguridad

b) Pruebas de aceptación. Son aquellas pruebas que realiza el usuario con el objeto de comprobar si el sistema es aceptable. Estas pruebas son del mismo tipo que las mencionadas anteriormente, pero son determinadas por el usuario, en lugar de serlo por el equipo de desarrollo.

Un lugar especial de estas pruebas es el de la ejecución en paralelo con el viejo sistema, para comparar los resultados producidos por ambas ejecuciones.

3.5 ESTÁNDARES PARA LA EVALUACIÓN

Se dice de lo que sirve como modelo, tipo, patrón, norma o referencia

3.5.1 MODELOS DE CALIDAD

Como se mencionaba en el apartado anterior los Modelos de Calidad ayudan a definir y poner en marcha el concepto general de Calidad de Software. Los Modelos de Calidad se definen de forma jerárquica es un concepto que se deriva de un conjunto de sub-conceptos, cada uno de los cuales se va a evaluar a través de un conjunto de indicadores o métricas. Existen algunos Modelos de calidad; entre los más conocidos el Modelo Mc Call, ISO 9126, IEEE, Boehm, GQM, Gilb, entre otros. La ventaja de los modelos de calidad es que la calidad se convierte en algo concreto, que se puede definir, medir y sobre todo planificar.

3.5.2 MODELO CALIDAD MODELO MC CALL

El Modelo Mc Call es el más antiguo (1977) y difundido, de él se han derivado la mayoría de los modelos que existen en la actualidad.

El Modelo descompone el concepto de calidad en tres puntos de vista importantes para un producto de software: Operación del producto, revisión del producto y transición del producto, éstos a su vez descomponen en 11 FACTORES de calidad o Atributos de calidad

Externos que muestran la calidad desde el punto de vista del usuario; y éstos a su vez se subdividen en 23 CRITERIOS de calidad o atributos de calidad Internos se trata de una visión de calidad desde el punto de vista del desarrollador; para cada uno de los criterios se definen métricas que son medidas cuantitativas de ciertas características del producto.

3.5.3 SISTEMA SIAC

PUNTO DE VISTA	FACTORES (Vista De Usuario)	CRITERIOS (Vista del Desarrollador)
Operación del producto	<p>FACILIDAD DE USO</p> <p>¿PUEDO UTILIZARLO?</p> <p>El sistema SIAC es de fácil operatividad una vez utilizada la clave de usuario.</p> <p>Se maneja con un entorno gráfico amigable,</p> <p>El tiempo de capacitación para un usuario promedio para el manejo del sistema es de diez</p>	<ul style="list-style-type: none"> • Facilidad de operación • Facilidad de comunicación • Facilidad de aprendizaje

	horas	
	<p>INTEGRIDAD</p> <p>¿ES SEGURO?</p> <p>El sistema se muestra con un nivel de seguridad medio, frágil en su base de datos y asequible a través de la red para sufrir modificaciones dentro de las tablas del sistema.</p> <p>Posee claves de seguridad encriptados, pero con ciclo sin fin de ingresos de nuevas claves hace que disminuya el nivel de seguridad del sistema, ya que ayuda a usuarios no deseados aumentar la posibilidad de búsqueda.</p> <p>El caso de la clave de administradores del sistema da acceso total a la modificación del mismo, incluso modificación de fechas que debería ser un campo inamovible por seguridad de auditoría informática.</p> <p>El sistema crea una auditoría interna de sí mismo para saber quien ingresó datos, cuando y en qué momento en función de usuario y no como</p>	<ul style="list-style-type: none"> • Control de accesos • Facilidad de Auditoria

	administrador.	
	<p>CORRECCIÓN</p> <p>¿HACE LO QUE YO NECESITO?</p> <p>Efectúa el sistema en un 51.6% de las necesidades del usuario, ya que en el campo de calificación de disciplina realiza cálculos que no reflejan la realidad del estudiante dentro del sistema escolástico.</p> <p>Es decir arroja datos erróneos de conducta de los estudiantes, convirtiéndose en un campo cerrado ya que no permite modificaciones y ajustes.</p> <p>En cuanto a los reportes posee un bajo nivel de presentación y con diagramación de formularios obsoleto. Muestra cuadros de error por siete ocasiones para mostrar los resultados de reportes.</p>	<ul style="list-style-type: none"> • Completitud • Consistencia • Trazabilidad
	<p>FIABILIDAD</p> <p>¿LO HACE CORRECTAMENTE TODO EL TIEMPO?</p>	<ul style="list-style-type: none"> • Precisión • Consistencia • Tolerancia a fallos • Modularidad • Simplicidad

	<p>Impresión en datos de conducta</p> <p>Cuellos de botella en reportes</p> <p>No existe una modularidad adecuada para evitar caducidad de reportes.</p> <p>La forma de presentación para realizar las impresiones es muy grande, se debe realizar alrededor de 6 pasos para conseguir la impresión</p>	
	<p>EFICIENCIA</p> <p>¿SE EJECUTARÁ LO MEJOR POSIBLE?</p> <p>La velocidad de altas, bajas y modificaciones dentro de los datos del sistema se ejecuta en un tiempo real aceptable.</p> <p>La forma de almacenamiento es de manera unitaria, es decir, no genera respaldos en otra unidad y se almacena la información en el directorio raíz de la base de datos, es decir si se presenta un problema de virus se podría perder la información de la base de datos.</p> <p>Esto podría suceder si la maquina sufriera de un corte eléctrico se perdería</p>	<ul style="list-style-type: none"> • Eficiencia en ejecución • Eficiencia en almacenamiento

	definitivamente la información.	
Revisión del Producto		
	<p>FACILIDAD DE MANTENIMIENTO</p> <p>¿PUEDO ARREGLARLO?</p> <p>La facilidad de mantenimiento no es una ventaja de este sistema es por lo que se puede catalogar como un sistema cerrado, no es simple el manejo de impresiones, y el soporte técnico es deficiente.</p>	<ul style="list-style-type: none"> • Modularidad • Simplicidad • Auto Descripción • Instrumentación
	<p>DE PRUEBA</p> <p>¿PUEDO PROBARLO?</p> <p>La instrumentación del sistema es aceptable, visualización de colores adecuada, la simplicidad en la mayoría de los módulos es bueno.</p>	<ul style="list-style-type: none"> • Modularidad • Simplicidad • Auto Descripción • Instrumentación
	<p>FLEXIBILIDAD</p> <p>¿PUEDO MODIFICARLO?</p> <p>Otra desventaja del sistema escolástico es la falta de flexibilidad e incompatibilidad de utilitarios de office para la extracción de back up en caso</p>	<ul style="list-style-type: none"> • Auto descripción • Capacidad de expansión • Generalidad • Modularidad

	<p>de fallas del sistema.</p> <p>No se puede incrementar módulos externos como lo que solicitó el personal administrador del sistema para la adecuación de las necesidades del sistema.</p>	
Transición del Producto		
	<p>REUSABILIDAD</p> <p>¿PODRÉ REUTILIZAR ALGUNAS PARTES?</p> <p>Por el tipo de software utilizado para el diseño del sistema escolástico</p>	<ul style="list-style-type: none"> • Auto descripción • Generalidad • Modularidad • Independencia entre sistema y software • Independencia del hardware
	<p>INTEROPERABILIDAD</p> <p>¿PODRÉ COMUNICARLO CON OTROS SISTEMAS?</p>	<ul style="list-style-type: none"> • Modularidad • Compatibilidad de comunicaciones • Compatibilidad de datos
	<p>PORTABILIDAD</p> <p>¿PODRÉ UTILIZARLO EN OTRA MÁQUINA?</p>	<ul style="list-style-type: none"> • Auto descripción • Modularidad • Independencia entre sistema y software

		<ul style="list-style-type: none"> • Independencia del hardware
--	--	--

Tabla1.2 Organización Puntos de Vista, Factores y Criterios Modelo Mc Call

3.6 MODELO ISO 9126

ISO plantea un Modelo genérico de calidad de software en su norma ISO 9126, en éste el concepto de calidad se divide en 6 sub-conceptos funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad y estos a su vez Incluyen otros como se muestra en la siguiente tabla:

CARACTERISTICA	DESCRIPCION
Funcionalidad	<p>¿El software cumple con el propósito para el que fue creado?</p> <p>EN UN 51.6 % DE SUS FUNCIONALIDAD</p>
• Adecuación	<p>¿Posee capacidad adecuada de proporcionar un conjunto de funciones para cumplir las tareas y objetivos especificados por el usuario?</p> <p>NO</p>
• Exactitud	<p>¿Proporciona la capacidad d emitir los resultados correctos con el grado de precisión necesario?</p> <p>NO</p>
• Interoperabilidad	Dispone de capacidad de interactuar con

	<p>uno o más sistemas especificados.</p> <p>NO A NIVEL DE USUARIO</p>
<p>• Seguridad</p>	<p>¿Posee habilidad para prevenir accesos no autorizados accidentales o intencionales, protección de programas y datos?</p> <p>60 % DE EFICACIA</p>
<p>• Conformidad</p>	<p>El software debe estar de la mano con estándares, convenciones, regulaciones en leyes y protocolos, ¿lo realiza?</p> <p>NO</p>
<p>FIABILIDAD</p>	<p>¿El software puede mantener su correcto desempeño bajo qué condiciones y por que período de tiempo?</p> <p>MANTIENE SU DESEMPEÑO MIENTRAS NO SE REQUIERA DE REPORTES</p>
<p>• Madurez</p>	<p>El sistema ofrece la capacidad de evitar fallar con frecuencia los errores que se dan en el software.</p> <p>CADA IMPRESIÓN FISICA</p>
<p>• Tolerancia a fallos</p>	<p>Presente habilidad de mantener su nivel y rendimiento en casos de fallos o errores de ingreso.</p> <p>SI</p>
<p>• Recuperación</p>	<p>¿Restablece su nivel, de rendimiento y recuperación de datos ante posibles fallos.</p> <p>NINGUNO</p>
<p>USABILIDAD</p>	<p>¿Cuánto esfuerzo necesito para utilizarlo?</p> <p>CONOCIMIENTOS BÁSICOS DE COMPUTACIÓN</p>

•Capacidad de Entendimiento	¿Cuál es el esfuerzo necesario para que un usuario entienda la lógica y la aplicabilidad del producto software? POCO ESFUERZO
• Capacidad de Aprendizaje	¿Cuánto es el esfuerzo requerido para que el usuario aprenda la aplicación, proceso, entradas y salidas? POCO ESFUERZO
• Operatividad	¿La operatividad y control de usuarios, cualitativamente es? BUENA
EFICIENCIA	¿Los recursos que utiliza el software para funcionar son los óptimos? SI
• Tiempo respuesta	¿Es optimo el tiempo de respuesta y procesamiento dentro de lo necesario para realizar determinada función? NIVEL LÓGICO BUENO, NIVEL FÍSICO MALO
• Optimización de Recursos	¿El promedio de recursos es el necesario para realizar una función ? MEDIO DEBIDO A QUE NO CUMPLE CON LAS EXPECTATIVAS DE LOS USUARIOS
MANTENIBILIDAD	¿Cuánto esfuerzo es necesario para realizar modificaciones, incluir correcciones, mejoras o adaptaciones al entorno? DIFÍCIL
• Capacidad de análisis	¿Cuánto esfuerzo se necesita para

	<p>encontrar las diferencias o causas de fallas o para identificar las partes a modificar? ESFUERZO ALTO</p>
<p>• Flexibilidad</p>	<p>¿Cuál es la capacidad de reducir el esfuerzo necesario para realizar cambios remover errores o cambios de entorno específicos? NIVEL ALTO, SOPORTE TÉCNICO INADECUADO</p>
<p>• Estabilidad</p>	<p>¿El software posee la capacidad de evitar efectos no esperados luego de las modificaciones? INESTABLE ERROR DE REDUNDANCIA EN DATOS DE REPORTE</p>
<p>PORTABILIDAD</p>	<p>¿Es posible transferir el software desde una organización a un entorno hardware o software diferente? SI</p>
<p>• Adaptabilidad</p>	<p>¿El software puede adaptarse a diferentes entornos? MIENTRAS SEA DENTRO DEL ENTORNO WINDOWS</p>
<p>• Inestabilidad</p>	<p>¿El esfuerzo necesario para instalar el software en un entorno es? ESFUERZO MÍNIMO</p>
<p>• Coexistencia</p>	<p>¿El sistema puede trabajar con otros sistemas adaptados a estándares y convenciones existentes?</p>

	CAPACIDAD NULA
•Capacidad reemplazar	¿Se puede utilizar y reemplazar partes del software en entornos puntuales? CAPACIDAD NULA

✓ Historia de Revisiones

FECHA	VERSIÓN	DESCRIPCIÓN	AUTOR
	1.0	Versión inicial del plan de evaluación de software	<ul style="list-style-type: none">• Jimena Caisa• Lilia Semblantes

3.7 PLAN DE DESARROLLO DE EVALUACION

3.7.1. PROPÓSITO

Mediante este documento, se pretende distinguir, repartir y ordenar las diferentes órdenes que se llevarán a cabo durante el desarrollo del proyecto, asignando a cada tarea o cada grupo de tareas un periodo de tiempo y unos recursos para su realización.

Además, se mostrará y quedará reflejada la organización de los miembros dentro del grupo, así como sus principales responsabilidades. Por último, otro de los objetivos de este documento es el de mostrar el funcionamiento del sistema mediante una serie de estimaciones para conocimiento de la institución.

3.7.2 ÁMBITO

Este documento de plan de revisión de software, describirá todas las fases, las tareas y el orden entre éstas, además de los resultados a entregar y todas las fechas de entrega relacionadas con el proyecto SIAC.

Además, será de utilidad tanto para el Director de la institución como a los diferentes miembros del Colegio “Primero de Abril” a lo largo de la duración del desarrollo del mismo.

3.6.1.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

SIAC. Sistema de Información Académico

3.7.4 REFERENCIAS

Este documento hace referencia a los siguientes documentos:

- Plan de Iteración
- Evaluación de la Iteración
- Lista de riesgos
- Documento de visión

3.7.5 VISIÓN GENERAL.

En este apartado describiremos de una forma general los aspectos que se detallan en este documento. Principalmente son tres:

3.7.6 RECURSOS Y MIEMBROS DEL EQUIPO: en el documento describiremos los diferentes miembros que forman parte del equipo de trabajo (niveles de usuarios y agentes administrativos), así como las responsabilidades y

habilidades de cada uno de ellos. Además, como ya se ha dicho antes, se les asignará ya las tareas a realizar.

3.7.6.1 TAREAS Y CRONOLOGÍA DE LAS MISMAS: en el documento se describirán las diferentes tareas a realizar, así como los recursos que tendrán asignadas cada una, las fechas y duraciones correspondientes. De esta forma, y una vez repartidas las tareas, podremos estimar la validez del sistema.

3.8 DESCRIPCIÓN DEL PROYECTO

3.8.1 PROPÓSITO DEL PROYECTO ÁMBITO Y OBJETIVOS

Mediante este proyecto se pretende evaluar un software educativo para registro de notas y matrículas.

Los principales objetivos de de nuestro proyecto son:

a. Gestión de usuarios:

- Tener los tipos de usuarios.
- Obtener diferentes privilegios.
- Poseer un registro de usuarios.

b. Gestión de estadísticas:

- Almacenar estadísticas de uso de software.
- Visualizar las estadísticas.
- Revisar estadísticas por parte del administrador.

- c. Gestión de edición:
 - Crear nuevos temas.
 - Modificar los temas existentes.

Los anteriores objetivos están basados en los requisitos mostrados en el documento de visión complementado en esta misma iteración, por medio de clientes y desarrolladores.

3.8.2 SUPOSICIONES Y RESTRICCIONES

Las pruebas se realizarán principalmente en el departamento administrativo del Colegio Nacional “Primero de Abril”.

La interfaz del sistema debe ser muy amigable, ya que principalmente será utilizada por usuarios novatos en el manejo informático.

Todas las fechas de entrega de documentos se deberán cumplir rigurosamente.

3.8.3 EVOLUCIÓN DEL PLAN DE EVALUACIÓN DE SOFTWARE

3.9 ESTRUCTURA ORGANIZACIONAL

En el proyecto descrito en este plan de evaluación del software participarán 2 alumnas de la prestigiosa Universidad Técnica de Cotopaxi, que se les conoce como testing, las mismas que serán asesoradas por un Director de Tesis.

Las relaciones entre los miembros del equipo se pueden ver en el siguiente esquema, donde las flechas representan la jerarquía dentro del equipo de trabajo en el proyecto del Sistema Escolástico:



Los roles quedan definidos de la siguiente manera:

Rol	Persona
Diseñador de Interfaz	SIAC empresa SETECOMPU
Implementador	Ing. Eduardo Rodríguez
Administrador	Lic. Luis Morales
Director de tesis	Ing. Xavier Montaluisa
Tester	Caisa Jimena Semblantes Lilia
Rector del Colegio	Msc. Hugo Armas
Secretaría	Lic. Carmen Bastidas

3.9.1 INTERFACES EXTERNAS

En este proyecto, las interfaces externas que se relacionarán con el mismo serán los profesores, a quienes se deben presentar la información entregable en cada iteración, así como al departamento administrativo, la matriculación de los discentes en los diferentes cursos por parte del administrador y los reportes generados por el sistema. Esto ocurre debido a que en el proyecto, los profesores actuarán como clientes limitados y, por otra parte, los usuarios de la aplicación serán los mismos miembros del equipo administrativo, por lo que no se contemplan otras interfaces externas de este proyecto.

3.9.2 ROLES Y RESPONSABILIDADES

Rol	Responsabilidad
Diseñador de Interfaz	Es el encargado de diseñar la interfaz a utilizar en la aplicación (teniendo en cuenta los diferentes términos de calidad y usabilidad de la misma)
Implementador	Es el encargado de desarrollar el código de la aplicación una vez conocidos los casos de uso, por medio de las diferentes aplicaciones utilizadas.
Administrador	Persona encargada de almacenar los datos además de distribuir y controlar las contraseñas de los diferentes usuarios, salvaguardando la integridad como veracidad de la información.
Director de tesis	Encargado de orientar al tester en los diferentes casos de uso, actores, (especificación, elicitación y modelado) para describir de una forma clara la funcionalidad del sistema

Tester	Es el encargado de realizar las pruebas sobre el sistema, así como de analizar e interpretar los resultados obtenidos en dichas pruebas.
Rector del Colegio	Es el encargado de repartir las tareas a los recursos disponibles (prioridades, planificación, etc.), así como de coordinar al resto de los miembros del grupo administrativo. En este caso, será el principal enlace entre el grupo y los profesores.
Secretaría	Segunda persona al mando después del administrador para suplir al mismo en caso de ser necesario

3.10 GESTIÓN DE PROCESOS

3.10.1 ESTIMACIONES DEL PROYECTO

Para el análisis del proyecto se ha aplicado el modelo de evaluación de caja negra, ya que contempla el prototipado para resolver problemas de interfaces, rendimiento o madurez de tecnología. Los pasos seguidos han sido los siguientes:

3.10.2 RECUENTO DE OBJETOS Y SU CLASIFICACIÓN:

Pantallas:

- Título.
- Login.
- Elegir idioma.
- Elegir el tema a modificar.
- Elegir usuario.
- Mostrar estadísticas sobre el manejo de un usuario.
- Modificar información de un usuario.

- Registrar usuario.
- Modificar el tema seleccionado.

Informes:

- Elegir el tema a modificar e imprimir
- Elegir usuario.
- Mostrar estadísticas sobre el manejo del sistema por parte de un usuario.
- Modificar el tema seleccionado.
- Manejo de Reportes

Componentes:

- Aplicación.
- Interfaz de usuario.

3.10.2.1 EXPERIENCIA Y CAPACIDAD DE LOS DESARROLLADORES:

- La experiencia que avaliza la empresa diseñadora del software SIAC, presenta una trayectoria aceptable por referencias presentadas al colegio.

- Supondremos que la capacidad del sistema adquirida está a la par que la experiencia.

3.9.3.2 CALCULAR EL TIEMPO ESTIMADO:

- El tiempo máximo para evaluar, asignado a los testing es de 2 horas al día (la cuarta parte del horario normal), necesitaremos 4 veces más al mes para terminar el proyecto de evaluación.

- Como el grupo de evaluadores se compone de 2 personas, el número de meses necesario para acabar la evaluación será: $37.72 / 2 = 18.86$ meses.

3.11 PLAN DEL PROYECTO

3.11.1 OBJETIVOS DE ITERACIÓN

Fases	Descripción
Inicio	<ul style="list-style-type: none">- Definición del ámbito del proyecto a realizar.- Identificar los casos de uso críticos.- Estimar riesgos potenciales.- Establecer una descripción inicial de la arquitectura derivada desde los escenarios significativos.
Evaluación	<ul style="list-style-type: none">- Asegurar que la arquitectura, requisitos y planes son estables para la fase de evaluación.- Establecer la línea base de la arquitectura.

- Refinar la arquitectura y seleccionar los componentes a hacer/comparar/rehusar. Dicha selección puede conllevar.

- Establecer un entorno estable de soporte describiendo las políticas de cambios, estructuras de directorio, esquemas de nombramiento y políticas de salvaguarda.

3.12 REFERENCIAS DEL PROYECTO

Implementación de Pruebas de Caja Negra y Caja Blanca al sistema escolarístico del Colegio Nacional “Primero de Abril”.

3.13 PLANTILLAS DE EXCEL

3.13.1 RESUMEN DEL INFORME TÉCNICO

- a. Realización del plan de trabajo para la evaluación del proyecto
- b. Se realizó pruebas según las normas ISO y pruebas de caja negra
- c. Se evaluó al sistema, al entorno como hardware, redes y conexiones eléctricas
- d. Los indicadores para seguimiento y evaluación del proyecto se dan según las normativas de estrés, interfaz y seguridad.

Observaciones:

- No existe una asesoría técnica real, eficiente para el área de sistemas ya que no posee la institución un técnico de planta; por ende, el sistema acoplado a realizar tareas escolásticas a más de presentar falencias frente a las necesidades reales del colegio, no posee un entorno seguro para los equipos informáticos que alojan o se relacionan con el sistema.
- No existe un concientización de lo que realmente significa la automatización es por ello que posee redundancia de procesos manuales que deberían ser guiados por el sistema.
- El sistema diseñado para esta tarea no posee un lineamiento exclusivo para resolver las necesidades del personal administrativo y docente.
- El sistema no es diseñado bajo una plataforma real de base de datos, más bien es un gestor obsoleto de base de datos.
- La empresa diseñadora del sistema existe en el mercado, al no haber manual de máquina y manual de usuario, lo que impide que se realice cambios y modificaciones en la estructura del mismo.
- La información de un manejo inadecuado de redes hace que el sistema se vuelva vulnerable ante usuarios que desean realizar cambios dentro de su estructura interna.

- La red que utiliza para enviar datos y recolectar información no posee un servidor y menos una red estructura, lo que poseen es una red doméstica.
- El sistema de red se encuentra junto al sistema eléctrico
- Las claves del computador que aloja el sistema son fáciles de cifrar y como es bajo Windows XP se puede borrar a través de la cuenta del administrador que está sin clave y sin la menor seguridad bajo red.
- El PC que se posee para efectos del sistema es un core duo, genérica, conectada las 24 horas a internet.
- Posee un antivirus pirata y crackeado para proteger el equipo
- Es una institución pública, sin embargo, no posee ninguna licencia de software, violando el Código Informático vigente en nuestro país.
- No existen manuales para usuario del sistema.
- No poseen un documento escrito que valide la pertenencia del sistema.
- No posee Copyright.
- Los usuarios ocasionales en un gran porcentaje no poseen experiencia informática.
- El sistema lo manipulan usuarios externos al sistema (hijos de los docentes).

3.13.2 INFORME TÉCNICO

El sistema SIAC fue creado en FOX PRO para Windows.

Pruebas realizadas al sistema según las normas ISO. Y, de acuerdo a las normativas de la IEEE, arrojan los siguientes resultados de acuerdo a los tres indicadores de evaluación dentro de su conjunto como son seguridad, estrés e interfaz.

El porcentaje de eficiencia global del sistema es de apenas un 39% ya que sus niveles de seguridad son deplorables y mínimos, solo se limita a poseer claves al ingreso pero no guarda informes reales de modificaciones, el administrador posee toda la libertad para realizar cambios y su base de datos es de fácil acceso a través de archivos de programa y no sólo de manera física en el PC. sino también a través de la red; es alarmante porque no posee las restricciones mínimas para protección de datos peor aun del sistema; la seguridad se ve afectada frente a los riesgos del Internet, y no solamente frente a los cracker o hacker sino también frente a problemas comunes de virus, ya que su antivirus actual es una copia pirata, que además de ser ilegal no proporciona la seguridad e integridad de la información, ya que la base de datos viral no se actualiza de la misma manera que un original; los riesgos de seguridad aumentan cuando la información puede ser perdida con facilidad, sólo se necesita una avería en la partición del disco duro y se perdería toda la información de años o meses de trabajo. Además, no genera backup eficientes y la recuperación de la información se requiere del técnico diseñador, es decir son problemas de nivel diez en cuanto a seguridad. Debemos recalcar que, el uso de claves son encriptados, pero permite al usuario realizar infinidad de intentos para descifrar la clave del administrador o de otros usuarios poniendo en riesgo la integridad de la información.

Los formatos utilizados en este nivel son de impresionante lentitud por la redundancia de proceso y errores en la programación que el sistema deja observar en su parte de reportes, errores de datos en cuanto a al campo disciplina, y la difícil manipulación dentro de un sistema extremadamente cerrado hace una tarea tediosa y estresante para el usuario.

La falta de guías de usuario, rotulaciones adecuadas, ayuda eficiente produce malestar en el usuario, y en muchos casos llevando a la utilización de usuarios no

autorizados para realizar tareas de llenado, causando así una fragilidad del sistema frente al ingreso de datos.

Las redundancias del sistema en el nivel de reportes es muy lamentable porque se debe a un error de compilación, el mismo que resulta de un enrutamiento inadecuado del módulo hacia el reporte, que se presenta en inglés (un lenguaje poco entendible para el usuario), siendo interpretado por un experto en sistemas.

La programación del casillero de disciplina es automática y éste, a su vez, no ofrece la información que se requiere porque no es ajustado a la realidad de la necesidad que se quiere solucionar.

La interfaz es creada con un tecnología cerrada, no permite realizar ninguna modificación en su forma y en su estructura, peor anexar y eliminar objetos para la visualización más cómoda del usuario.

El porcentaje de eficiencia global del sistema es del 51.6%, el mismo que no es aceptable para poder seguir utilizando el sistema.

3.13.3 NUEVA TECNOLOGÍA

- Cambiar el tipo de red, de doméstica a una red estructurada
- Cambiarse a un sistema operativo bajo LINUX.
- Utilizar un sistema diseñado en plataformas gratuitas para no pagar licencias.
- Reestructurar el sistema eléctrico.

- Capacitar al personal en el manejo del sistema escolástico y realizar pruebas de evaluación al usuario.
- Eliminar software innecesario para el usuario, programas que causan perjuicios involuntarios al sistema operativo y por ende al sistema.
- Mejorar las claves de ingreso a la base de datos y al sistema operativo
- Realizar en el servidor las restricciones de acceso de páginas con contenido de código sospechoso para evitar conflictos en las máquinas
- Particionar el disco duro para la generación de backup o instalar otro disco duro.

3.14 PLANIFICACIÓN DE LA ELAVORACION DEL PROTOTIPO

3.14.1 UTILIDAD DEL PROTOTIPO DEL SISTEMA

El realizar un modelo del sistema tiene por objetivo ejecutar los programas para encontrar las posibles falencias del mismo. Se desarrolla un prototipo del programa para probar que no existen errores en el desarrollo de un programa, es decir, hacer que una implementación no falle en su ejecución.

La confiabilidad es un aspecto del diseño, por lo que debe estar dentro del sistema, más adelante se indican las estrategias específicas que se utilizaron en el prototipo del sistema.

3.14.2 ANÁLISIS DEL PROTOTIPO

A pesar de los mejores esfuerzos para la implementación del sistema, las necesidades de información no siempre se establecen correctamente. Esto puede ocurrir porque los usuarios pueden saber sólo lo que necesitan mejorar el sistema en ciertas áreas, o lo que deben modificar en los procedimientos existentes, o simplemente no están muy seguros del tipo de información que necesitan.

Por lo tanto, se pueden considerar los prototipos para proporcionar la información necesaria y formular el diseño o desarrollar el sistema, debido a que un prototipo es un sistema de trabajo que se desarrolla con rapidez para probar las ideas y el funcionamiento del nuevo sistema, en otras palabras no sólo es un diseño en papel, sino un software que produzca información impresa o en pantalla.

Los objetivos de esta técnica del prototipo son:

- Validar la funcionalidad del sistema.
- Probar los principios y verificar que el sistema funciona
- Incrementar el nivel de productividad del desarrollo del sistema.
- Busca un acercamiento (interacción) entre el usuario y el sistema.
- Presentar en forma jerárquica las pantallas y el diálogo con el usuario.
- Obtener una realimentación (evolución del prototipo).

Temas complejos de procesamiento de datos tienen la necesidad de prototipos, mediante éstos se busca obtener una retroalimentación del análisis y del diseño antes de realizar su construcción.

Para tal efecto se trabaja con el usuario final creando diálogos (pantallas) para entrada de datos, consultas, menús, efectuar demostraciones de ellos y realiza generación de informes, manipulación de información por pantalla.

Al construir el prototipo se facilitó el desarrollo del sistema, éste se realizó de las tres formas siguientes; un prototipo en papel que describa la interacción hombre-máquina, de forma que facilite al usuario la comprensión de como se produce tal interacción, un prototipo que funcione y por ultimo uno que implemente algunos subconjuntos de la función requerida del sistema requerido.

3.14.3 NECESIDAD DE UTILIZAR UN PROTOTIPO

No se conocen los requerimientos

La naturaleza de esta aplicación es tal que, la información disponible en la base de datos creada en SQL Server no está actualizada con respecto a las características que debe tener el sistema para satisfacer los requerimientos de los usuarios.

3.14.4 DETERMINACIÓN DEL SISTEMA A IMPLEMENTAR

El desarrollo de este prototipo y su evaluación final se llevó a cabo en una forma ordenada, sin importar las herramientas utilizadas. Los pasos seguidos fueron los siguientes:

Identificación de requerimientos conocidos

La determinación de los requerimientos de esta aplicación es tan importante para los requerimientos de hardware, software e interfaz de usuario y así establecer un estándar del sistema, además de conocer si se contaba con las herramientas, método y ciclo del desarrollo del mismo. Por consiguiente se determinaron necesarias las siguientes actividades:

3.14.5DESARROLLO DE UN MODELO DE TRABAJO

Es útil comenzar el proceso de construcción del prototipo con el desarrollo de un plan general que permita a los usuarios conocer lo que se espera de ellos y del proceso de desarrollo. Un cronograma para el inicio y el fin de la implementación es de gran ayuda, al igual que la utilización de un bosquejo.

Para el desarrollo del prototipo se prepara los siguientes componentes:

- La interfaz del usuario. El lenguaje para el diálogo entre el usuario y el sistema, pantallas y formatos para la entrada de datos.
- Módulos esenciales de procesamiento. Módulo de digitalización de imágenes, módulo de consulta y búsqueda de información y módulo de actualización de datos.
- Salida del sistema. Diseño de la consulta y búsqueda de información.

Para construir este prototipo se tuvo que seguir estándares para que los datos que fueron empleados estuvieran ordenados y organizados (longitud de datos, características, tipo, entre otros).

3.14.6 REVISIÓN DEL PROTOTIPO

Durante la evaluación del sistema se captó información sobre la satisfacción del usuario (lo que le agrada y desagrada). Los cambios al prototipo son planificados con los usuarios antes de llevar a cabo tales modificaciones.

Se realizaron revisiones periódicas de la implementación y diseño del sistema; por parte del Director del Proyecto. De esta manera se deberá asegurar que las especificaciones del diseño y arquitectura del sistema se ajusten a las especificaciones funcionales obtenidas en la fase de análisis de sistemas. Y por último, se obtendrá la aprobación formal del presente diseño.

3.15 PROTOTIPO PROPUESTO

PROCESO: MATRICULACIÓN – INGRESO DE NOTAS – CONTROL DE ASISTENCIA – SISTEMA DE CONTROL

✓ OBJETIVO

Este manual tiene como objetivo documentar los procedimientos que se desarrollan en las áreas funcionales una vez que estos ya han sido racionalizados.

✓ SISTEMA DE CONTROL ACADÉMICO

El sistema de software en la Web se debe desarrollar utilizando Metodología de Ingeniería de Software, Herramientas de Programación y Base Datos Relacional de Software Libre (Licencias de Uso sin Costo), que los hacen más intuitivos

fáciles de manejar por los usuarios finales, además de garantizar la calidad de los sistemas construidos bajo este concepto.

3.15.1 UTILIZACIÓN DEL SISTEMA

Para ingresar a la página de Internet, abrimos INTERNET EXPLORER y escribimos en la barra de dirección:

www.primerodeabril.com.edu

Ingresamos a Sistema escolástico dando un clic sobre el texto ubicado en la parte superior derecha.

Accedemos al sistema ingresando el Nombre de Usuario y la Contraseña asignada.

3.15.2 MANEJO DEL SISTEMA

EL manejo de las diferentes herramientas se detalla en los siguientes capítulos, mediante sus propios procedimientos.

3.15.3 DESCRIPCIÓN DE PROCESOS CLAVES

✓ PROCESO DE MATRICULACIÓN

Subproceso: Matriculas Extraordinarias y Listado Final de Alumnos Matriculados

Identificación del Proceso

Nombre		Código
Matriculación y Listado Final de Alumnos Matriculados		1E
Tipo	Área Funcional	No Versión
Clave		1
Frecuencia		Fecha Elaboración
Anual		

Objetivo

Matricular a los alumnos que cumplan con los requisitos para formar parte del colegio.

Ingresar el listado definitivo de alumnos matriculados al Sistema Web de control Académico.

Imprimir los Registros de control diario y de evaluación.

Producto

- Listado Definitivo de Alumnos Matriculados.
- Impresión de Registro de Evaluación.
- Impresión de Registro de Control Diario.

Alcance

La matriculación estará a cargo del administrador del sistema.

Los tiempos de ingreso y modificación de datos al Sistema de Control Académico serán restringidos al periodo de matriculación. Luego del tiempo determinado no se podrá ingresar ni modificar datos.

Responsables

- Rector
- Administrador
- Secretaria

NORMATIVA APLICADA AL PROCESO

- Reglamento de la Dirección de Educación.

DESCRIPCIÓN DE LAS ACTIVIDADES

No.	Responsable	Entrada	Actividades	Salida
2	Administrador	Listado de Alumnos presentados	Controlar el proceso de matriculación y revisar las solicitudes de matrícula. Avalar el no constar con impedimento de matrícula Revisar los impresos de matriculación, firmándolos si son conformes	Formato de matriculación y aprobación de requisitos Carpeta de requisitos
4	Administrador	Formato de matriculación y aprobación de requisitos Listado de Alumnos Admitidos	Realizar Listado Definitivo de Alumnos Matriculados	Listado definitivo de Alumnos Matriculados

FORMULARIO DE CONTROL

- Consta de Control de eventos del formulario.
- Sistema de Control Académico.

MATRICULAS

El sistema de matrículas se divide en dos secciones:

- Ingreso de alumnos Matriculados y

- Consulta de Alumnos Matriculados.

EN MATRICULACIÓN TENEMOS TRES OPCIONES:

- Cambiar los datos de una persona ya matriculada.
- Consultar las personas matriculadas en periodo lectivo.
- Ingresar una nueva matrícula.

MODIFICAR MATRICULA

Permite modificar los datos mal ingresados de los alumnos matriculados.

CONSULTA DE MATRICULADOS

Consulta por apellidos o numero de cédula de los datos ingresados al sistema.

INGRESAR UNA NUEVA MATRICULA

- A.** Es un espacio automático donde no se ingresa ningún dato.
- B.** Es el número de espacios que quedan para escribir.

LIMPIAR

Borra todo lo escrito en ese momento en la pantalla.

SALIR

Sale del sistema e ingresa el Nombre del Usuario y la Contraseña para seguir trabajando.

CONTINUAR

Graba los datos ingresados.

CANCELAR

Se eliminan los datos anteriores y regresa a la primera pantalla de matriculación, donde se ingresan los datos personales.

REGRESAR

Regresa a la página anterior donde se puede modificar únicamente la información sobre las pruebas.

GRABAR

Guarda la información y veremos el siguiente anuncio y posteriormente regresaremos a la primera pantalla de matriculación.

CONSULTAS

Ingresamos a Consulta los Alumnos Matriculados.

Para consultar los alumnos matriculados tenemos dos opciones:

Cedula

Ingresar el número de cédula del alumno.

Apellido y Nombre

Ingresar los apellidos y nombres del alumno.

MATRICULA

Regresa a la pantalla de matriculación de alumnos.

La opción permanece desactivada fuera del periodo de matriculación.

CONSULTA

Al hacer la consulta con el botón se verá la siguiente pantalla, con los datos del alumno.

Regresa a la página de consulta de alumnos matriculados.

PROCESO DE EVALUACIÓN Y APROBACIÓN DE ASIGNATURAS

Subproceso: Registro de Resultados (Notas)

Identificación del Proceso

Nombre		Código
Registro de Resultados (Notas)		2 ^a
Tipo	Área Funcional	No Versión
Clave	Secretaria	1
Frecuencia		Fecha Elaboración
Mensual		

OBJETIVO

Registrar las calificaciones y resultados de las evaluaciones

PRODUCTO

- Cuadro de notas, registro en base de datos.

ALCANCE

Los resultados parciales y finales de las evaluaciones, serán ingresados al sistema escolástico por los profesores en el Formato de Registro de Evaluación, de acuerdo a la planificación aprobada, en las fechas establecidas en el calendario general de actividades.

El Administrador o el Rector es responsable de aprobar los registros del control de notas.

El Secretario es el responsable de introducir y mantener los registros en la base de datos utilizando el Sistema de Control Académico.

RESPONSABLES

- Profesores, Administrador, Secretaria

DESCRIPCIÓN DE LAS ACTIVIDADES

No.	Responsable	Entrada	Actividades	Salida
1	Profesores	Deberes, Lecciones y Pruebas	Evaluación del alumnado	Formato de Registro de Evaluación
2	Secretaria	Formato de Registro de Evaluación	Introducir a la Base de Datos	Base de datos de evaluación de alumnos
3	Rector General Administrativo	Formato de Registro de Evaluación Base de datos de evaluación de alumnos	Evaluar la calidad del proceso de enseñanza aprendizaje	Oficio de correcciones o mejoras al proceso

Formulario de Control

1. Registro de notas:
2. Promedio

Promedio:

La suma todos las Pruebas dividido para tres

Componentes para el promedio

- Deberes
- Pruebas
- Lecciones
- Total
- Observaciones
- Fecha

La suma del promedio de: Cualquier observación con respecto a:

- Notas.
- Deberes.
- Lecciones.
- y Pruebas.

SISTEMA DE CONTROL ACADÉMICO

INGRESO DE NOTAS

El sistema de Notas se divide en dos secciones.

1. Ingreso de Notas al sistema.

2. Consulta de Notas del sistema.

FORMULARIO - ALUMNO

La pestaña del formulario contiene los datos de la materia, el paralelo y el profesor que ingresa la nota.

Una vez finalizada y guardada la información de la pestaña **Formulario** se ingresaran las notas de los alumnos en la pestaña **Alumno**.

FECHA DE REGISTRO

Ingresar la fecha del registro de la nota tal como aparece en el Formulario de Control de Notas

Paralelo – Asignatura - profesor

Copiar del Formulario de Control de Notas los datos correspondientes al paralelo.

Al seleccionar la asignatura, el nombre del profesor se cargará automáticamente.

Damos clic en Guardar para luego acceder a la pestaña de Alumnos.

La opción Regresar cambia a la pantalla principal de Notas.

Formulario - Alumno

Al acceder a la pestaña **Alumno** se desplegará el listado del paralelo seleccionado con las casillas para el ingreso de notas.

Deber – Lección - Prueba

El formulario automáticamente mostrará el listado del paralelo seleccionado con las casillas de ingreso de notas. Es importante seleccionar la casilla de chequeo correspondiente a la nota que se desea ingresar. Copiar del Formulario de Control de Notas los datos correspondientes a cada alumno.

Guardar

Guarda los datos ingresados en la Base de Datos, se muestra la pantalla de registro exitoso.

Aceptando el registro se regresa al menú principal.

CONSULTA DE NOTAS DEL SISTEMA

Cedula

Se ingresa la cedula y se accede a las notas del alumno matriculado.

Regresar

Muestra la pagina principal de reporte de Notas

Paralelo - Materia

Se accede a todas las notas de un paralelo de una materia determinada. Al realizar la consulta se muestra el siguiente reporte:

Consulta

Realiza la consulta respectiva

Regresar

Muestra la pagina principal de ingreso de notas.

Subproceso: Registro de asistencia del registro de control diario

Identificación del Proceso

Nombre		Código
Registro de asistencia		2C
Tipo	Área Funcional	No Versión
Clave		1
Frecuencia	Administrador	Fecha Elaboración
Diario		

Objetivo

La asistencia a las clases es parte fundamental del control.

Producto

- Registro diario de control de asistencia.
- Informe de asistencia mensual de alumnos.
- Base de datos de asistencia de alumnos.

Alcance

La asistencia de los profesores y estudiantes será ingresada al Sistema de Control Académico por la Secretaria, desde el registro de control diario llevado por el profesor.

El Administrador será responsable controlar y aprobar el control llevado por los alumnos conjuntamente con el Rector.

Se tendrá acceso en línea a los reportes de atrasos de los alumnos.

Responsables

- Profesores
- Administrador
- Rector

Descripción de las actividades

No.	Responsable	Entrada	Actividades	Salida
1	Profesor	Asistencia de alumnos y registro de clases impartidas	Tomar lista a los alumnos	Registro de control diario, llenado y firmado diariamente por el profesor.
2	Administrador	Registro de control diario, llenado y firmado diariamente por el profesor.	Aprobación diaria del registro de control	Firma de responsabilidad del registro de control diario
3	Director Pedagógico			
4	Secretaria	Registro de control diario, llenado y firmado	Archivar los registros de control diario e introducir los datos en el sistema de Control Académico	Base de datos de asistencia

SISTEMA DE CONTROL ACADÉMICO

ASISTENCIA

El sistema de Asistencia se divide en tres secciones principales:

- Para el Proceso de Registro de Asistencia del Registro de Control Diario se utilizará la opción de Registro de Control Diario.
- Las opciones de Justificación de faltas y Consultas se trataran en el proceso Justificación de Faltas.
- Al ingresar al Registro de Control Diario el sistema muestra dos pantallas correspondientes a las partes mostradas en el formato impreso. En la primera pantalla se ingresa la asistencia de los profesores y los datos de las materias impartidas; en la segunda pantalla se muestra el listado de los alumnos con la opción de asistencia, atraso o falta.

Se accede al sistema dando clic en Ingresar Registro de Control Diario

FECHA DE REGISTRO

Ingresar la fecha de registro del Formato de Control Diario de Asistencia

Paralelo

Seleccionamos el paralelo al cual ingresaremos los datos. Si no se selecciona un paralelo no se podrá proseguir con la sección de asistencia.

Numero de Formulario

Ingresamos el mismo número de formulario del Formato de Control Diario de Asistencia para correlacionar el registro físico con la base de datos.

Asignatura

Seleccionamos del listado la asignatura correspondiente.

Profesor

El nombre del profesor se mostrara automáticamente al seleccionar la asignatura.

Unidad Académica – Tema/Actividad

Ingresamos la unidad académica y la actividad realizada en la hora de clases como se indica en el formulario del Formato de Control Diario.

Continuar

Seleccionar, “Continuar” si se desea guardar los datos ingresados para poder los datos de asistencia de alumnos.

Regresar

Muestra la pantalla principal de Asistencia.

Para ingresar la asistencia de los alumnos se selecciona la pestaña de ASISTENCIA.

Presente – Atraso – Falta (por cada materia)

El sistema muestra preseleccionado la opción **presente** para las materias impartidas. Si existe un atraso o falta se debe seleccionar la opción por cada materia y en cada alumno de la lista tal como fue ingresado en el formulario del Formato de Control Diario.

Guardar

Salva los datos ingresados en las plantillas control de asistencia de alumnos y profesores.

Muestra la confirmación de ingreso de datos:

Al ingresar a “Regresar”, salimos a la pantalla principal:

Subproceso: Justificación de faltas

Identificación del Proceso

Nombre		Código
Justificación de atrasos y faltas		2A
Tipo	Área Funcional	No Versión
Clave	Administrador	1
Frecuencia		Fecha Elaboración
Cuando sea necesario		

Objetivo

Registrar las justificaciones de faltas

Producto

- Formulario aprobado de justificación, registro en base de datos.

Alcance

Almacenar en una base de datos las justificaciones aprobadas de las faltas emitidas para cada alumno existente en la base de datos de Inasistencias y Atrasos registrada. El modulo permite ingresar los datos aprobados en el formato de justificación de una manera eficiente y rápida

Responsables

- Rector
- Administrador

Descripción de las actividades

No.	Responsable	Entrada	Actividades	Salida
1	Alumno	Petición de Justificación	Llenar el formato de justificación	Formato de justificación
2	Administrador	Formato de justificación	Aprobación de la Justificación solicitada	Formato justificación Aprobado
3	Rector	Formato justificación Aprobado	Aprobación de la Justificación solicitada	Formato justificación Aprobado
4	Secretaria	Formato justificación Aprobado	Ingreso al Sistema de Control Académico	Base de datos de Justificaciones

Formulario de Control

- Fecha del día que es llenado el formulario.
- Paralelo en el que se encuentra el alumno que va a justificar la falta.

- Nombre del colegio.
- Código del Alumno, asignado el momento de la matriculación y que se encuentra en las lista de cada paralelo y que es un número.
- Números de días que se justifican, mínimo uno.
- Si se justifica más de un día se pondrá la fecha del último día que justifica.
- Fecha del día que faltó y está justificando.
- Ejemplo
- Un día de justificación:

1

----- ---- -----

1 5 0 7 0 8

Dos días de justificación:

1 6 0 7 0 8

1 5 0 7 0 8

2

Cinco días de justificación:

2 0 0 7 0 8

1 5 0 7 0 8

5

NOTA:

SI LA PERSONA HA FALTADO EN DOS DÍAS NO SEGUIDOS SE DEBE REALIZAR DOS REGISTROS DE JUSTIFICACIÓN DIFERENTES:

Ejemplo:

FALTA EL 15 / 07 / 08 y el 17 / 07 / 08

Motivo de la falta que se va a justificar, se debe marcar con una X en una sola casilla, de acuerdo a lo indicado.

X

Si el motivo de la falta no está especificado se marcará en 3.Otros y se especificará en las líneas que están a continuación:

Fuera del país

X

Las firmas del Administrador y del Director con las únicas que avalan la autenticidad del registro, sin las cuales el documento no tiene validez.

EJEMPLO DE CÓMO DEBE SER LLENADO EL FORMULARIO

1 2 0 6 0 8

SAN GABRIEL

DELGADO CARRILLO

CAMILO STALIN

15

X

1

1 0 0 6 0 8

-- -- -- -- --

SISTEMA DE CONTROL ACADÉMICO

✓ JUSTIFICACIÓN

El sistema de Asistencia se divide en tres secciones principales; para el Proceso de Justificación de Faltas se utilizará las siguientes opciones.

El sistema de Justificaciones se divide en dos secciones:

1. Justificar Inasistencia.
2. Consultar.

El sistema permite justificar la falta de un día a la vez, si una persona ha faltado por un periodo de tiempo y existe un solo formulario lleno que justifica el mismo periodo, se debe ingresar las justificación por cada día.

✓ RECUPERAR INASISTENTES

Para justificar las faltas es necesario ver la lista de inasistencias de la fecha y el paralelo escogidos, para lo cual; ingresamos la fecha a consultas y el paralelo. Seleccionamos el botón **Recuperar inasistentes**

✓ LISTADO DE ALUMNOS CON FALTAS - NÚMERO DE REGISTRO

Al seleccionar “Recuperar Inasistentes” veremos la lista que nos indica la asignatura, los nombre de los alumnos, el tipo de inasistencia (Falta o Atraso) y el estado de la inasistencia.

✓ ESTADO DE ASISTENCIA

Al seleccionar cualquiera de las celdas marcadas con **Justificar** se mostrará la siguiente pantalla:

Ingresamos la fecha que consta en el formulario de justificación y seleccionamos el motivo de la falta.

Guardamos la Justificación.

✓ RECUPERAR INASISTENTES

Si ingresamos a Recuperar inasistencias podremos observar la lista ya actualizada:

REGRESAR

Al ingresar a Regresar, salimos a la pantalla principal:

✓ MODULO DE CONSULTAS Y ESTADÍSTICAS

El modulo de Consultas y Estadísticas tiene como objetivo mostrar de una manera organizada y objetiva los datos almacenados en las diferentes bases de datos, referentes a matriculación, notas, inasistencias o atrasos y justificaciones.

Está pendiente la implementación de las opciones de estadísticas que mostraran datos procesados en forma alfanumérica y gráfica.

REPORTES

Ingresamos dando clic en **Reportes**

LISTADO GENERAL DE ALUMNOS

Genera una lista de todos los alumnos, en Excel, organizados por orden alfabético.

Para obtenerla damos clic en Ejecutar

Open / Abrir

El listado se verá como una hoja de Excel desde la dirección de Internet, no se guardará en la computadora.

Save / Guardar

Guardará en la computadora el listado generado. En esta opción veremos la siguiente pantalla:

Donde debemos señalar con que nombre y donde se guardará el archivo.

Cancel

Para la generación de la lista y regresa a la pantalla de reportes:

✓ LISTADO DE ALUMNOS POR PARALELO

Genera una lista de alumnos

Genera una lista de los alumnos por paralelo elegido, en Excel, organizados por orden alfabético.

Para obtenerla seleccionamos el paralelo deseado, dando clic sobre el mismo:

Y generamos la lista dando clic en **Ejecutar**.

✓ **INASISTENCIA ENTRE FECHAS**

Genera un listado en Excel de todas las personas que han faltado en un determinado período de tiempo. Esta lista incluye faltas no justificadas y justificadas.

Para generar el listado debemos definir las fechas:

Damos clic en **Ejecutar**.

✓ **MANEJO DE LA INFORMACIÓN**

La tecnología de nuestros días pone al alcance de la humanidad capacidades hasta ahora impensadas. En los últimos años el avance en la capacidad de almacenamiento y la velocidad de acceso a la misma se vio afectada por factores de crecimiento en escala pasando de pocos Mega a cientos de mega y aún miles como el caso de los discos ópticos con cientos de Gigabytes (1024 Megabytes) y más de capacidad.

Toneladas de papel pueden ser reemplazadas en forma simple y segura por su correlato en disco, con las consiguientes ventajas en disponibilidad inmediata, tiempo de búsqueda y acceso a la información y aún seguridad física, tanto en el acceso como ante factores externos.

Descripción del sistema. Nombre, eficiencia, confiabilidad.

3.15.4 REQUERIMIENTOS DE HARDWARE Y SOFTWARE

✓ Hardware y Software

Como se mencionó anteriormente el diseño es una actividad en la cual se toman decisiones importantes, frecuentemente de naturaleza estructural.

Comparte con la programación un interés por la abstracción de la representación de la información y secuencias de procesamiento, pero el nivel de detalle es muy diferente en ambos casos. El diseño constituye representaciones coherentes y bien planificadas de los programas concentrándose en las interrelaciones de los componentes al mayor nivel y en la manipulación de los objetos de datos implicados en los niveles inferiores. A continuación se presenta el documento que conforma el núcleo del sistema propuesto: el documento de especificación de diseño.

La plataforma sobre la que se desarrolla este sistema es Windows XP Professional que solo es compatible con Windows 2000 Professional.

3.14.1 ESPECIFICACIÓN DE LOS REQUISITOS

La especificación de los requisitos del software implica la culminación de la tarea del análisis de sistemas. Dicha especificación se logra estableciendo una completa descripción de las clases que colaboran, su función y el comportamiento del sistema. Este documento y el modelado que contiene deben lograr tres objetivos en mente:

- Describir lo que requiere el usuario.
- Establecer una base para la creación de un diseño de software.

- Definir un conjunto de requisitos que se puedan validar una vez que se ha construido el software.

De esta manera, se logran establecer la base para un buen diseño de sistemas, documentando una descripción del problema que el software va a resolver al definir las clases principales que componen al sistema, así como los atributos y métodos que las componen, además de las relaciones que existen entre ellas.

Sin embargo, los requerimientos de hardware recomendables pueden resumirse en los siguientes puntos:

- Pentium I o equivalente.
- 64 MB de memoria RAM.
- Tarjeta de video de 16-bits de color.
- Tarjeta de Video de 4 MB de memoria.
- 200 MB de Disco Duro disponible.

Para un desarrollo satisfactorio del proyecto, serán requeridos los siguientes componentes de software:

- Linux.
- Antivirus especializado para software free.
- Cortafuegos para Red.
- Un mantenimiento de software periódico.

3.16 RECURSOS ESPECIALES

Para el desarrollo del proyecto han sido necesario manuales y libros especializados en programación en Fox Pro para Windows, así como libros y páginas de Internet que funcionen como fundamentos teóricos referentes al problema que se desea resolver de una manera exitosa.

3.17 CONCLUSIONES

- El sistema escolástico del colegio Nacional Primero de Abril al ser evaluado encontramos varias falencias tanto en interfaz, seguridad y conexiones eléctricas.
- A más de no existe una asesoría técnica real eficiente en el departamento administrativo del colegio es por esta razón que si existe falencias en el sistema tienen necesariamente que acudir a la empresa que implemento al sistema.
- Además que no existe una verdadera concientización de lo que realmente significa la automatización es por esta razón que existe redundancia de procesos manuales.
- También existe un manejo inadecuado de redes y conexiones eléctricas lo que provoca que el sistema se vuelva vulnerable y no exista integridad de datos.

3.18 RECOMENDACIONES

- Debería existir un técnico informático de planta en la institución educativa.
- Además se debería cambiar el tipo de red de domestica a estructurada para brindar seguridad e integridad ala sistema.
- Además que exista capacitación permanente tanto a docentes como al personal administrativo en cuanto al manejo del sistema informático.
- También recomendamos reestructurar el sistema eléctrico ya que una caída de tención ocasionaría perdida de información de años o meses de trabajo.
- La propuesta que recomendamos es la implementación de un nuevo sistema escolástico que vaya acorde a los requerimientos de los usuarios.
- Deberán permitir realizar prácticas pre- profesionales a estudiantes de ingeniería en sistemas, para que de esta manera se puedan hacer cambios significativos con el menor costo

BIBLIOGRAFIA

BIBLIOGRAFÍA CITADA

JAMES Semn (1992), Análisis y Diseño de Sistema de Información, Segunda Edición, México.

<http://es.wikipedia.org/wiki/sistemasdeinformacion>

http://es.wikipedia.org/wiki/cajanegra_gr%C3%A1fica_de_usuario

http://es.wikipedia.org/wiki/Caja_negra

<http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm>

http://es.wikipedia.org/wiki/ISO/IEC_15504

<http://es.wikipedia.org/wiki/ieee/>

http://es.wikipedia.org/wiki/ISO_9001:2000

<http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm#s3>

www.rogeliodavila.com/tcs/.../Parte_16_TestWhite.ppt

BIBLIOGRAFIA CONSULTADA

- ✓ AGUILAR, Marlene (1992), Metodología de la Investigación Científica, Loja Editorial Universidad Técnica Particular de Loja.
- ✓ ULLOA, Francisco (2000), Investigación 2000, Latacunga.
- ✓ LEIVA, Francisco, Nociones de la Metodología de Investigación Científica, Tercera Edición, Quito – Ecuador, págs. 9, 13, 26,27.

AMENOS