



INSTITUTO SUPERIOR MINERO METALÚRGICO
"Dr. Antonio Núñez Jiménez"

Facultad Metalurgia - Electromecánica
Carrera de Ingeniería Informática.

Trabajo de diploma para optar por el título
de Ingeniería en Informática

Tema:

Herramienta para el control de reportes de averías y mantenimientos de los medios informáticos del ISMM.

Autor:

Yadriel Vargas Padilla.

Moa, Cuba
Junio, 2011

"Año del 53 Aniversario del Triunfo de la Revolución."



INSTITUTO SUPERIOR MINERO METALÚRGICO
"Dr. Antonio Núñez Jiménez"

Facultad Metalurgia - Electromecánica
Carrera de Ingeniería Informática.

Trabajo de diploma para optar por el título
de Ingeniería en Informática

Tema:

Herramienta para el control de reportes de averías y mantenimientos de los medios informáticos del ISMM.

Autor:

Yadriel Vargas Padilla.

Tutor:

Ing. Marcos A. Martínez Rodríguez.

Moa, Cuba
Junio, 2011

"Año del 53 Aniversario del Triunfo de la Revolución."

DECLARACIÓN DE AUTORÍA

Yo, Yadriel Vargas Padilla declaro que soy el único autor de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa (ISMMM) y al departamento de informática para hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del 2011.

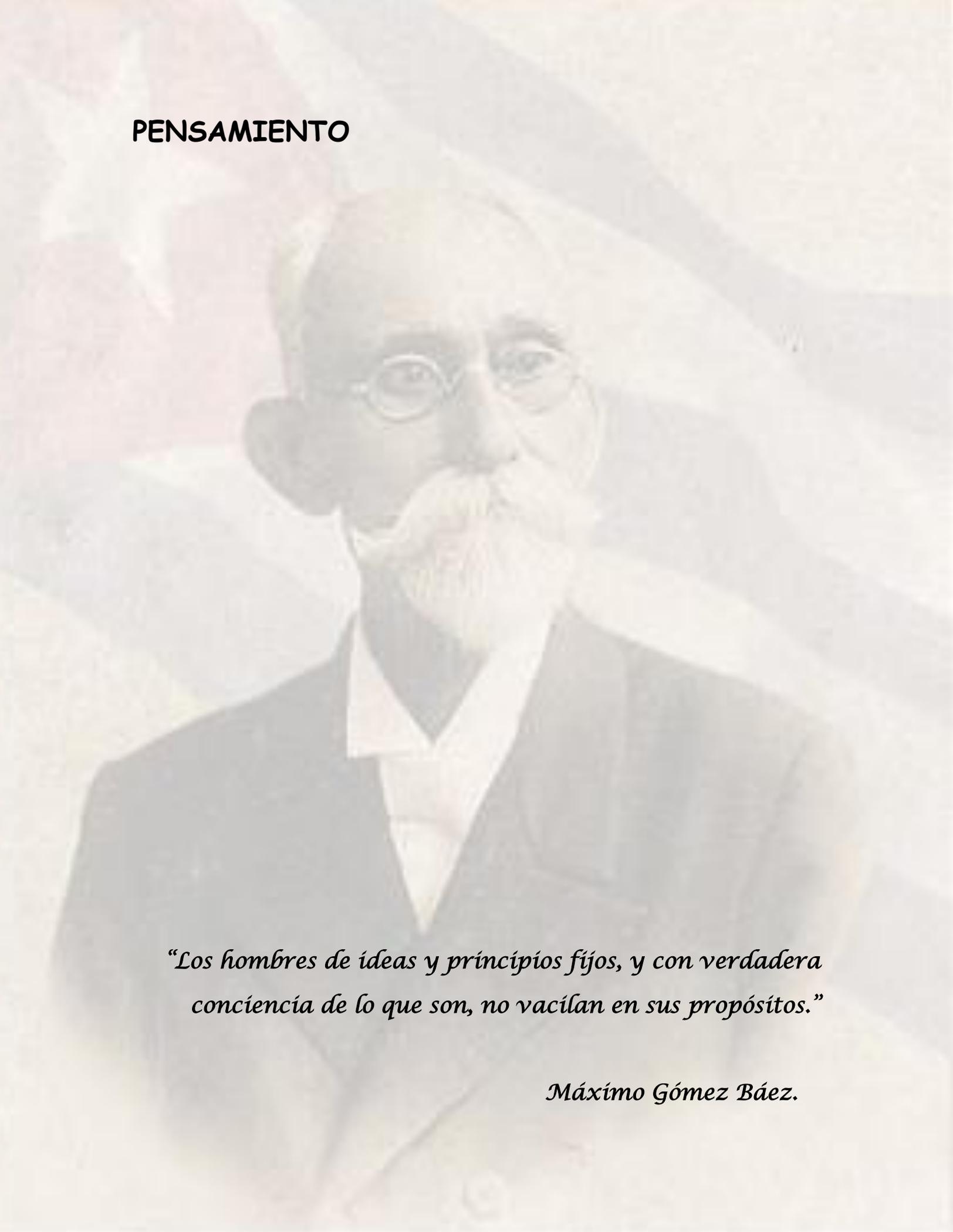
Yadriel Vargas Padilla.

Firma del autor.

Ing. Marcos A. Martínez Rodríguez.

Firma del tutor.

PENSAMIENTO

A faded, grayscale portrait of Máximo Gómez Báez, a prominent leader in the Cuban War of Independence. He is depicted from the chest up, wearing a dark suit jacket, a white shirt, and a light-colored tie. He has a full, white beard and mustache, and is wearing round-rimmed spectacles. The background is a light, textured surface with some faint, abstract shapes.

“Los hombres de ideas y principios fijos, y con verdadera conciencia de lo que son, no vacilan en sus propósitos.”

Máximo Gómez Báez.

DEDICATORIA

Este trabajo va dedicado a mi incansable madre que ha sido el artífice y prácticamente la autora de este trabajo. A mi padre a mis abuelos que me han apoyado en todo este tiempo de estudiante, en fin a toda la familia, las personas y amigos que de cierta forma se han visto involucrados con la culminación exitosa de mi vida universitaria.

AGRADECIMIENTOS

Primeramente quisiera agradecer a mis padres, a mis abuelos, a mi tía Marlene que han estado ahí de forma sacrificada para que esta investigación tenga éxito.

A mis compañeros de grupo por haberme soportado durante estos largos cinco años.

Al sacrificado claustro de profesores del departamento que dieron el máximo de su profesionalidad para formarme como un profesional de bien para esta sociedad.

A mis amigos del barrio, a mi amigo Randy, Tita y demás que muy atentamente me han apoyado para mi formación como ingeniero.

Y un aparte especial para quien ha sido en estos meses finales más que una amiga, ha sido el faro que ha alumbrado cada paso que doy en afán de conquistar mis propósitos; gracias a ti mi Cosí. ! Ah! Se me quedaba mi hermano Paumier como se le conoce jocosamente, por haberme ayudado en todo lo que pudo.

No quisiera ser egoísta y que alguien querido se me quedara fuera por lo que vuelvo a declarar muchas:

... ¡GRACIAS A TODOS LOS QUE ME APOYARON! ...

Resumen

El ISMM cuenta con una red de cómputo bastante extensa que se distribuye por todas las facultades y centros de estudios que componen dicha red. Por el acelerado deterioro de los medios y el descontrol que en ocasiones se tienen sobre estos, el departamento de informática, a propuesta de los directivos responsables del departamento de mantenimiento de equipamiento informático del centro de altos estudios, se dio a la tarea de implementar una herramienta para el control de reportes de averías y mantenimiento de los medios informáticos. En la actualidad en dicho centro se carece de una herramienta que sea capaz de recopilar la información referente a una queja de avería o mantenimiento, todo esto tributando a un desempeño ineficiente de este factor de vital importancia. El resultado concreto de este trabajo fue proporcionar una vía eficiente, la cual brinde la posibilidad de mejorar todo este proceso, para la cual se implementó una aplicación informática que permitió realizar las tareas pertinentes. Además se presenta la propuesta metodológica de Ingeniería de Software, las herramientas y lenguajes de programación que se utilizaron para el desarrollo de la aplicación, como vía de solución a la problemática encontrada.

Abstract

The ISMM has a quite extensive computation network that is distributed throughout all the faculties and learning centres that compose this network. Due to the faster deterioration and inadequate control of the medium that occur occasionally, the department of computer science, owing to the proposal of the directive of the department of computer equipment maintenance, a study task was given in order to implement a tool for controlling the hardware damage reports and the maintenance of information mediums. At the present time the centre has no tool that is able to gather the information with respect to complaint hardware damage or maintenance, all this contributing to inefficiency in the role of this factor of vital importance. The principal objectives of this work are to provide a platform that would offer the possibility to improve the efficiency of this process. This would be possible with the implementation of a computer application that allows the conduction of the pertinent tasks. Besides, a proposed Software Engineering methodology, the tools and programming languages that were used for the development of the application that serve as a solution platform to the existing problem have been presented.

ÍNDICE

Resumen	V
Abstract	VI
INTRODUCCIÓN	1
FUNDAMENTACIÓN TEÓRICA	6
Introducción	6
1 Estado del Arte.	6
1.1 Cuestiones generales del proceso de control de los medios informáticos en el ISMMM.	6
1.2 Sistemas automatizados existentes vinculados al campo de acción.	7
1.3 Aplicaciones Web.	9
1.3.1 ¿Qué es una aplicación Web? ¿Por qué la utilizamos?	9
1.3.2 Cualidades de las aplicaciones Web.	10
1.3.3 Arquitectura cliente servidor.	11
1.4 Arquitecturas para el desarrollo e implementación de producto software.	12
1.4.1 Arquitecturas en tres capas.	12
1.4.2 Patrón Arquitectónico Modelo Vistas Controladores (MVC)	14
1.5 Tecnologías para el desarrollo del sistema.	15
1.5.1 Lenguaje empleado del lado del cliente.	16
1.5.2 Lenguajes del lado del servidor.	16
1.6 Herramientas utilizadas en la propuesta de solución.	18
1.6.1 Dreamweaver como entorno de desarrollo.	18
1.6.2 Sistema gestor de Bases de Datos.	18
1.6.2.1 ¿Qué es PostgreSQL?	18
1.6.2 Paquete de tecnología Web.	20
1.6.3 Embarcadero ER/Studio 8.0.	20
1.6.4 PhpDesigner 2008 v6.0.0	21
1.7 Fundamentación de la selección de las herramientas utilizadas para la implementación del producto software.	21
1.8 Tendencias y tecnologías actuales.	22
1.8.1 Metodologías para el desarrollo de productos software.	22
1.8.2 Proceso Unificado de Desarrollo (RUP).	22
1.8.3 Metodología ágil desarrollo de software.	24
1.8.3.1 Un proceso es ágil cuando el desarrollo de software es:	25
1.8.3.2 Programación Extrema.(eXtreme programming, XP)	25
1.8.3.3 Características fundamentales de XP.	26
1.8.3.4 Proceso XP	27
1.8.3.5 SCRUM	28

1.9	Fundamentación de la selección de la metodología de desarrollo de software.	29
1.10	Conclusiones del Capítulo	30
PLANIFICACIÓN Y DISEÑO		31
	Introducción	31
2	Funcionalidades generales.	31
2.1	Lista de reserva.	32
2.2	Historias de Usuarios (HU).	33
2.3	Planificación de entregas.	34
2.3.1	Estimación de esfuerzo por HU.	35
2.3.2	Planificación de entregas.	35
2.3.3	Plan de duración de las iteraciones.	36
2.4	Tarjetas Clases-Responsabilidades-Colaboración (CRC).	37
2.4.1	Tarjeta C.R.C Módulo #1: Gestionar Elementos.	37
2.5	Conclusiones del Capítulo	38
IMPLEMENTACIÓN Y PRUEBAS		39
	Introducción	39
3	Implementación	39
3.1	Tareas por HU	40
3.1.1	Tareas de la HU #1	40
3.2	Prueba	40
3.2.1	Pruebas de Aceptación (PA).	40
3.2.2	Pruebas del módulo #2: Obtención de reportes.	41
3.3	Modelo lógico de datos	42
3.4	Modelo físico de datos.	43
3.5	Conclusiones del Capítulo	43
ESTUDIO DE FACTIBILIDAD DEL PROYECTO		44
	Introducción	44
4	Efectos Económicos.	44
4.1	Beneficios y Costos Intangibles en el proyecto	46
4.2	Ficha de costo	47
4.3	Conclusiones del Capítulo	49
CONCLUSIONES GENERALES		50

RECOMENDACIONES	51
FUENTES BIBLIOGRÁFICAS	52
ANEXOS	52
Anexo-1 <i>Historias de Usuarios</i>	
Anexo-2 <i>Tarjetas CRC</i>	
Anexo-3 <i>Tareas de Ingeniería</i>	
Anexo-4 <i>Pruebas de Aceptación</i>	
Anexo-5 <i>Modelo Físico de Datos</i>	

ÍNDICE DE ILUSTRACIONES

Figura 1: Representación de la arquitectura en tres capas. _____	12
Figura 2: Modelo Vistas Controladores. _____	14
Figura 3: Tecnologías para el desarrollo aplicaciones Web. _____	15
Figura 4. Metodología XP. _____	25
Figura 5. Modelo de un proyecto XP. _____	28
Figura 6. Modelo de datos. _____	42
Figura 7. Gráfico punto de equilibrio de soluciones. _____	49
Figura 8. Modelo físico de datos. _____	74

ÍNDICE DE TABLAS

Tabla1: Personal relacionada con el sistema. _____	31
Tabla2: Plantilla de Historia de Usuario. _____	33
Tabla3: HU Reportar Averías. _____	34
Tabla4: Estimación de esfuerzo por HU. _____	35
Tabla5: Plan de entregas. _____	35
Tabla6: Duración de iteraciones. _____	36
Tabla7: Tabla de releases. _____	36
Tabla8: Plantilla de tarjeta CRC. _____	37
Tabla9: Tarjeta CRC. _____	37
Tabla10: Plantilla de Tareas de Ingeniería. _____	39
Tabla11: Tareas de Ingeniería. _____	40
Tabla12: Plantilla Prueba de Aceptación. _____	41
Tabla13: P.A Reportar averías. _____	41
Tabla14: Costo en Moneda Libremente Convertible. _____	47
Tabla15: Costo en Moneda Nacional. _____	48
Tabla16: HU Buscar reporte. _____	54
Tabla17: HU Obtener estadísticas de los reportes gestionados en el módulo. ____	54
Tabla18: HU Obtener reportes. _____	55
Tabla19: HU Gestionar usuarios del sistema. _____	55
Tabla20: HU Elaborar reporte de mantenimiento. _____	56
Tabla21: Gestionar Computadoras. _____	56
Tabla22: HU Autenticar usuario. _____	57
Tabla23: HU Atender avisos de mantenimiento. _____	57
Tabla24: HU Asignar avisos. _____	58
Tabla25: HU Exportar los datos a formato de impresión. _____	58
Tabla26: Gestionar dispositivos. _____	58
Tabla27: Tarjeta C.R.C Gestionar_User. _____	59
Tabla28: Tarjeta C.R.C Gestionar_PCs. _____	59
Tabla29: Tarjeta C.R.C Gestionar_Dispositivos. _____	59

Tabla30: Tarjeta C.R.C Report_Averia. _____	59
Tabla31: Tarjeta C.R.C Buscar_Report. _____	60
Tabla32: Tarjeta C.R.C Estadísticas_Report. _____	60
Tabla33: Tarjeta C.R.C Mostrar Reportes. _____	60
Tabla 34: Tarjeta C.R.C Imprimir. _____	60
Tabla35: Tarjeta C.R.C Clases: Plan_Mtto. _____	61
Tabla36: Tarjeta C.R.C. _____	61
Tabla37: Tarea de programación Procesar información de avería. _____	61
Tabla38: Tarea de programación Búsqueda de reporte específico. _____	62
Tabla39: Tarea de programación obtención de estadísticas. _____	62
Tabla40: Tarea de programación Reporte pendiente. _____	62
Tabla41: Tarea de programación Reporte pendiente por piezas. _____	63
Tabla42: Tarea de programación Reporte terminado. _____	63
Tabla43: Tarea de programación Insertar usuarios. _____	63
Tabla44: Tarea de programación Eliminar usuarios. _____	64
Tabla45: Tarea de programación Modifica usuarios. _____	64
Tabla46: Tarea de programación Cambiar contraseña. _____	64
Tabla47: Tarea de programación Ver usuarios. _____	65
Tabla48: Tarea de programación Plan de mantenimiento. _____	65
Tabla49: Tarea de programación Insertar computadoras. _____	65
Tabla50: Tarea de programación Baja de computadoras. _____	66
Tabla51: Tarea de programación Alta computadora. _____	66
Tabla52: Tarea de programación Loguear usuario. _____	66
Tabla 53: Tarea de programación atención de avisos averías. _____	67
Tabla54: Tarea de programación atención mtto. _____	67
Tabla55: Tarea de programación Exportar formato de impresión. _____	67
Tabla56: Tarea de programación Insertar dispositivo. _____	68
Tabla 57: Tarea de programación eliminar dispositivo. _____	68
Tabla 58: Tarea de programación modificar dispositivo. _____	68
Tabla59: Prueba de aceptación. _____	69
Tabla60: Prueba de aceptación. _____	69

Tabla61: Prueba de aceptación	69
Tabla62: Prueba de aceptación.	70
Tabla63: Prueba de aceptación.	70
Tabla64: Prueba de aceptación.	71
Tabla65: Prueba de aceptación.	71
Tabla66: Prueba de aceptación.	72
Tabla67: Prueba de aceptación.	72
Tabla68: Prueba de aceptación.	73
Tabla69: Prueba de aceptación.	73

INTRODUCCIÓN

La esfera de influencia de la informática en el mundo contemporáneo es muy amplia y multidisciplinaria, relacionando disciplinas establecidas, como son la ciencia de la computación y la ingeniería; que abarca las relaciones informativas en sistemas hombre-hombre y hombre-máquina.

El objeto de trabajo del ingeniero en informática es la automatización de los sistemas de información en organismos bases.

En particular este ingeniero tiene su campo de acción asociado a la concepción diseño-desarrollo, implementación y mantenimiento de los sistemas automatizados de información para organizaciones bases, productivas y de servicios y debe contribuir a incrementar la eficiencia en el funcionamiento de esas entidades.

Este profesional trabaja de forma multidisciplinaria con los especialistas de las ramas correspondientes, debe concebir e implementar el sistema de información automatizado que de respuestas a las necesidades informativas; lo que a su vez implica que ellos mismo seleccionen los equipos, técnicas, y métodos más efectivos para la captación de conocimiento de acuerdo a las características de la aplicación que se trate (Morejón 1990).

En el mundo actual la mayoría de las actividades, acciones y operaciones hacen uso de la computadora. La informática soporta cambios continuos que repercuten a diario en nuestra sociedad. Hoy no se conciben entidades que no utilicen las Tecnología de la Información y las Comunicaciones (TIC) ya que esta es capaz de aportar soluciones eficientes y eficaces permitiendo actuar con rapidez, obtener el máximo rendimiento del personal y tomar mejores decisiones.

Nuestro país no se encuentra ajeno al desarrollo del producto software y en los últimos años ha tomado cierto auge, con el objetivo siempre de brindar una mejor información a la sociedad. Las universidades cubanas juegan un papel fundamental en este proceso sometido a cambios y transformaciones ante los retos que le plantea la actual sociedad del conocimiento.

El Instituto Superior Minero Metalúrgico de Moa (ISMMM), no está ajeno a esas transformaciones y cambios por lo que se ha trazado una política de automatizar la gran parte de los procesos docentes y extracurriculares por que atraviesa la

institución. El ISMMM cuenta con una red de cómputo bastante amplia, por lo que el reporte de las averías, gestión de mantenimiento y la reparación de los medios informáticos que componen esta red se hace tedioso, y se ha dado la tarea de implementar un aplicación informática que sustituya, todo este proceso.

Como consecuencia a lo planteado anteriormente se tiene la siguiente **situación problemática**: que debido a la densidad del equipamiento informático con el que cuenta la institución, y que el proceso de reporte de averías se efectúa de forma manual actualmente, ello trae consigo que se produzcan errores en la manipulación de la información, el uso irracional de papeles y otros recursos de oficina, la demora en el mantenimiento y reparación de los medios informáticos de las distintas áreas que componen la red de cómputo del ISMMM, perjudicando así a los usuarios implicados.

Dada esta situación podemos definir como **problema científico**: la no existencia de una herramienta automatizada para el control y gestión de los medios informáticos del ISMMM.

Para encontrarle solución al problema planteado se tiene como **objeto de estudio**: la creación de aplicaciones Web para la automatización de los reportes de averías y mantenimiento del equipamiento informático. Y como **campo de acción**: la automatización de la generación de reportes de averías y mantenimientos en la red de cómputo del ISMMM.

De manera que el **objetivo general** de la investigación viene encaminado en desarrollar una aplicación con tecnología Web que permita la gestión y control de la información de forma más rápida, segura y eficiente en el proceso de reportes de averías, mantenimientos de los medios informáticos del ISMMM.

Para dar respuesta al problema científico se plantea como **idea a defender**: qué, si se implementa una aplicación Web, para la gestión y control de la información en el proceso de reportes de averías y mantenimientos de los medios informáticos, entonces se agilizará y se hará de forma más fiable el desarrollo de esta actividad en el ISMMM.

Los **objetivos específicos** a seguir son:

- Desarrollar todo el marco teórico pertinente con el sistema de reporte de averías, mantenimiento de los medios informáticos.
- Analizar y diseñar la aplicación que posibilite mostrar los principales parámetros a tener en cuenta para llevar a cabo las fases de desarrollo e implementación.
- Desarrollar e implementar la aplicación para realizar el reporte de averías, mantenimiento de los medios informáticos.
- Realizar el estudio de factibilidad que permita mostrar los costos y beneficios del sistema.

Para lograr un eficaz desarrollo de la investigación y darle seguimiento a los objetivos trazados se plantearon las siguientes **tareas científicas**:

- Estudio de los referentes teóricos acerca de los procesos de automatización de los reportes de averías.
- Fundamentación teórica de la investigación.
- Diagnóstico de la situación existente en los procesos objetos de estudio.
- Elaboración de una aplicación informática para el reporte de averías y mantenimiento del equipamiento informático.
- Evaluación del nivel de factibilidad del desarrollo del producto software.

Para cumplimentar estas tareas científicas se han empleado métodos de investigación científica.

- **Métodos teóricos** empleados se encuentran:

Histórico y lógico: para la búsqueda de antecedentes del software, las herramientas utilizadas, así como la forma de cómo se llevaba a cabo el control de los medios informáticos en el ISMMM.

Análisis y síntesis: Se utilizó en los fundamentos teóricos, en el procesamiento de la información y en la descomposición de cada uno de los requerimientos del sistema.

Hipotético deductivo: Fue empleado para la elaboración de la idea a defender del trabajo, la cual permitirá deducir la solución del problema una vez demostrada por la investigación.

Revisión de documentos: Se utilizó fundamentalmente para constatar y recopilar los requerimientos del software a desarrollar en las diferentes documentaciones y bibliografías.

➤ **Métodos empíricos** que se utilizaron fueron:

Observación: Se empleó para caracterizar y analizar el previo funcionamiento del problema planteado, así como para llevar a cabo la implementación y diagnóstico del resultado obtenido en la investigación.

El presente trabajo está estructurado por un resumen, introducción general, 4 capítulos, conclusiones generales, recomendaciones, bibliografías y anexos:

En el Capítulo1. ***Fundamentación teórica.***

Se ofrece una breve descripción de diferentes conceptos imprescindibles que le sirven de pilar a la investigación que se llevó a cabo. Así como algunos elementos importantes de las herramientas a utilizar para la implementación del sistema.

En el Capítulo2. ***Planeación y Diseño.***

Se hace uso de la metodología propuesta en el capítulo inicial para el desarrollo del sistema, abordando en detalles cada una de las fases.

En el Capítulo3. ***Implementación y Pruebas.***

Se presentan los principales métodos y definiciones dentro de la implementación de los flujos de trabajo. Se describen además las pruebas realizadas y los resultados que esta arroja.

En el Capítulo4. ***Estudio de factibilidad.***

Se realiza un estudio de los esfuerzos requeridos para la realización del sistema. Se hace referencia a los beneficios tangibles e intangibles y se analizan los costos de desarrollo de la aplicación contra los beneficios para ver si es factible o no la aplicación.

Capítulo 1

FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo se abordarán cuestiones generales sobre el control de los medios informáticos en el ISMMM. Se realizará una exhaustiva investigación sobre algunos sistemas automatizados existentes unidos al campo de acción. Se definirán cuestiones básicas relacionadas con las aplicaciones Web, así como las tecnologías para su desarrollo y el sistema de bases de datos utilizados en la misma. Además se expone la metodología de Ingeniería de Software que se usará para el desarrollo de la aplicación.

1 Estado del Arte.

1.1 Cuestiones generales del proceso de control de los medios informáticos en el ISMMM.

Seguidamente se expondrá de forma breve el funcionamiento interno del problema a resolver. El problema inicia cuando a un usuario de la red de cómputo del ISMMM se le avería algún medio informático. Este debe informárselo a los responsables que cumplen con dicha tarea, este localiza el área a la que pertenece la queja de avería y lo reporta como medio informático averiado. El encargado de reportar la avería lo asigna a la oficina de Copextel del centro, el técnico de taller de Copextel disponible es el encargado de ir hasta el área donde se localiza el equipo y revisarlo. Si el problema puede ser solucionado inmediatamente lo hace y abre un plan de mantenimiento a dicho equipo, sino si al problema no se le puede dar solución inmediatamente se remite al taller y se le asigna al técnico de reparación hasta que sea reparado. También existen técnicos en mantenimiento que son los responsables de dar mantenimiento a los medios informáticos del ISMMM, determinado por un período de tiempo impuesto por el taller de Copextel. Cada vez que surge un aviso de avería se abre una orden de

servicio que recoge lo que se le realizó al equipo en su reparación, esta orden también es empleada cuando un equipo pasa a taller o se le da mantenimiento.

1.2 Sistemas automatizados existentes vinculados al campo de acción.

El ISMMM no posee una herramienta que satisfaga las necesidades de los usuarios de la red de cómputo, en cuanto al reporte de averías de los medios informáticos se refiere. Por lo que se incurre en insuficiencias en cuanto al mantenimiento y reparación de estos medios. En cuanto a la investigación teórica realizada se determinó que los sistemas vinculados al campo de acción a nuestro trabajo que existen en el país no cumplen con las exigencias del entorno donde se va implementar dicho software o sea que algunos no cumplen con los controles deseados y otros realizan funcionalidades que van más allá de los intereses que se propone el ISMMM. Muestra de esto es la herramienta que se implementó en la CUJAE que cumple en gran parte con las funcionalidades que se requiere pero no está a fin con lo que se quiere en nuestro trabajo. Las exigencias funcionales que debe tener el software a desarrollar son:

- ❖ Que los administradores de las áreas de la red de cómputo del ISMMM sean los que provean las averías que presentan los equipos al área de las que son responsables.
- ❖ El administrador de la aplicación es el encargado directo de realizar la planificación de mantenimiento que recibirán los equipos en sus diferentes áreas, además va ser el encargado de gestionar los usuarios que interactuaran con el sistema así como la gestión de los medios informáticos que se les incorporen al sistema como medios físicos de la red de cómputo del ISMM.
- ❖ El técnico de Copextel máximo responsable de los técnicos será el encargado de interactuar con el sistema atendiendo los reportes de averías así como los de mantenimiento.

En nuestro país se han desarrollado herramientas con algunas de las funcionalidades que se quieren en el presente trabajo.

Herramienta para el control de los medios informáticos. (Emp. Cdte. Ernesto Che Guevara).

Sistema de reparación y mantenimiento de los medios informáticos. (CUJAE).

Internacionalmente existen compañías que comercializan sistemas que realizan algunas de las funcionalidades que se quiere.

- ❖ **Aranda Asset Management (AAM):** La versión 6.2 de AAM ofrece facilidades de administración y control remoto, realización de certificados y cuentas, y generación de reportes desde la misma aplicación. Junto con ello, realiza licenciamiento de software y medición de software para conocer el nivel de sus aplicativos en la empresa. Además, permite manejar datos de un equipo tales como SO, procesadores, capacidad de memoria, discos lógicos, espacio disponible y utilizado, y recursos compartidos y entre otros. Esta herramienta ofrece dos niveles de ayuda fundamentales: operativo, que permite ejecutar distintos tipos de distribuciones y controlar de manera remota una máquina; y administrativo, pues mantiene los inventarios del día y también puede desarrollar reportes.

- ❖ **NTRsupport:** Es una herramienta de asistencia y control remotos de ordenadores a través de Internet. Se trata de una solución diseñada para complementar y agilizar el servicio de atención al cliente tanto en empresa como en organismos públicos, ofreciendo la posibilidad de conectarse a su ordenador a través de Internet. Así pues, NTRsupport ser usado para proporcionar asistencia remota tanto a clientes externos como a trabajadores internos. NTRsupport es una solución SaaS (el Software como un Servicio, por sus siglas en inglés) lo que implica que, además de evitar instalaciones, cualquier empresa que lo use no deberá dedicar ni recursos – servidores, SO, BD, redes, etc. – ni personal para mantenerlo en funcionamiento ni actualizarlo.

- ❖ **Microsoft Systems Management Server (SMS):** Es un producto del software de la gerencia de sistemas de la Corporación Microsoft para manejar grandes grupos de sistemas de ordenadores basadas en Windows. SMS provee distribución de control remoto del software, hardware e inventario del software.
- ❖ **El SAP (Sistemas, Aplicaciones y Productos):** Es un sistema integral que incluye un módulo relacionado con el objeto de estudio a precios inalcanzables para nuestras universidades.
- ❖ **El Manage Engine ServiceDesk:** Es una aplicación de escritorio que se encarga del control de los medios informáticos. Pero hay funcionalidades como el reporte por un administrador de equipos que no cumple (Rodríguez 2008).

Aunque algunos de las aplicaciones expuestas anteriormente tienen una amplia gama de funcionalidades similares a las que se quieren obtener en el presente trabajo, estos no se adaptan a las necesidades del ISMMM, para resolver de forma eficiente y factible el problema a resolver.

1.3 Aplicaciones Web.

1.3.1 ¿Qué es una aplicación Web? ¿Por qué la utilizamos?

Una aplicación Web es un sistema informático que los usuarios usan accediendo a un servidor Web a través de los protocolos de Internet. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero. La facilidad para actualizar y mantener las aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su creciente popularidad.

Una aplicación Web está comúnmente estructurada como una aplicación en tres-capas. En su forma más común el navegador es la primera capa, un motor usando alguna tecnología de Web dinámica (Ej.: CGI, PHP, ASP, Java Servlets o Pearl) es la capa del medio, y algún tipo de gestor de bases de datos como última capa.

consultas y actualizaciones a la bases de datos generando una interfaz de usuarios(Subiros 2009).

En tiempos recientes se ha usado la estrategia de generalizar esta arquitectura mediante la adición de piezas de hardware que permitan balancear la carga de los servidores Web y de las aplicaciones.

Las Aplicaciones Web son de un desarrollo poco costoso, sencillo y rápido. Presenta acceso ubicuo, sin necesidad de distribución e idealmente, con pocos requerimientos técnicos. Con datos centralizados y fácil integración de datos múltiples fuentes. Debido a estas ventajas que presenta es la razón por la cual es utilizada en este trabajo para llevar a cabo la implementación del software(Subiros, 2009).

1.3.2 Cualidades de las aplicaciones Web.

Para que la Web sea efectiva, debe poseer algunas cualidades indispensables, como la seguridad, la escalabilidad, portabilidad y un diseño eficiente de la interfaz de usuario. La escalabilidad y la portabilidad son las que generalmente caen en olvido.

Escalabilidad: la aplicación tendrá la capacidad de crecer junto a las necesidades de sus usuarios.

Portabilidad: se enlaza el concepto de escalabilidad, y se refiere a la compatibilidad de la aplicación con múltiples sistemas operativos y plataformas utilizadas en el mercado. En general, cuando la demanda de los usuarios crece, las aplicaciones deben cambiar de plataforma para pasar de un servidor pequeño a uno mediano o grande. Es entonces, cuando la portabilidad es crítica para poder “escalar” sin problemas independientemente del entorno que se requiera.

Sin dudas, las aplicaciones Web aún tienen un largo camino por recorrer, debido a que son una opción muy interesante, especialmente, en tareas donde las bases de datos juegan un papel predominantes y los usuarios se encuentran dispersos(Monmany 2009).

1.3.3 Arquitectura cliente servidor.

Esta arquitectura consiste básicamente en que un programa –el cliente informático – realiza peticiones a otro programa –el servidor – que le da respuesta.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de las responsabilidades, lo que facilita y clarifica el diseño del sistema. La separación entre el cliente y el servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni necesariamente en un solo programa. El servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

La **arquitectura cliente-servidor** sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico.

Ventajas.

- Centralización del control: los accesos, los recursos, la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- Escalabilidad: se puede aumentar la capacidad de los clientes y el servidor por separado.

El servidor del cliente es una arquitectura de red que separa al cliente (a menudo un uso que utiliza una interfaz gráfica de usuario) de un servidor. Cada caso del software del cliente puede enviar peticiones a un servidor. Los tipos específicos de servidores incluyen servidores Web, los servidores de archivo, los servidores del correo, entre otros. Mientras que sus prototipos varían algo, su arquitectura básica sigue siendo la misma. Aunque esta idea se aplica en una variedad de maneras, en diversas clases de usos, el ejemplo más fácil de visualizar es el uso actual de páginas Web en internet(Monografias.com. 2007).

1.4 Arquitecturas para el desarrollo e implementación de producto software.

1.4.1 Arquitecturas en tres capas.

Arquitectura en capas: es donde se define como organizar el modelo de diseño a través de capas, que pueden estar físicamente distribuidas, lo que quiere decir que los componentes de una capa solo puede hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son consistentes de ningún detalle o interfaz de las superiores.

La programación por capas es un estilo de programación en el que el objetivo primordial es separar la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación del usuario.

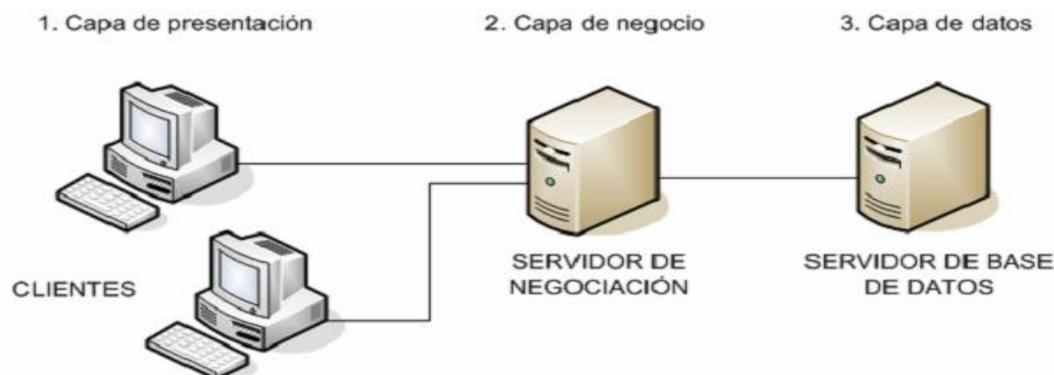


Figura 1: Representación de la arquitectura en tres capas.

Capas o niveles.

- Capa de presentación o interface: es la capa que le permite al usuario interactuar con el sistema, captura y le comunica la información al mismo, dando un mínimo de proceso (realiza un filtrado previo para asegurarse que no haya errores de formato). Esta capa se comunica únicamente con la de de negocio.
- Capa de negocio: es donde reside los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el

proceso. Se denomina capa de negocio (e incluso lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitarle al gestor de bases de datos almacenar o recuperar datos de él.

- Capa de datos: es donde residen los datos y es la encargada de acceder ellos. Está formada por uno o más gestores de bases de datos que se encargan de realizar el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en un mismo ordenador aunque no es lo típico. Lo más usual es que haya una multitud de ordenadores donde reside la capa de interface (son los clientes de la arquitectura cliente/servidor). Las capas de negocios y de datos pueden residir en un mismo ordenador, y si el crecimiento de las necesidades lo aconseja, pueden dividirse en dos o más ordenadores. Así, si el tamaño o complejidad de las base de datos aumenta, pueden dividirse en varios ordenadores los cuales recibirán las peticiones del ordenador donde resida la capa de negocio. Si por el contrario, la complejidad fuese en la capa de negocio lo que obligase a la separación, esta lógica del negocio podría residir en uno o más ordenadores que realizarían las solicitudes a una única base de datos.

Ventajas:

- ❖ El estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual, permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales.
- ❖ El estilo admite muy naturalmente optimizaciones y refinamientos.
- ❖ Proporciona una amplia reutilización. Al igual que los tipos de datos abstractos, se pueden utilizar diferentes implementaciones o versiones de una misma capa a medida que soporten las mismas interfaces de cara a las capas adyacentes. Esto conduce a la posibilidad de definir interfaces de

capa estándar, a partir de las cuales pueden construirse extensiones o presentaciones específicas (Rodríguez 2008).

1.4.2 Patrón Arquitectónico Modelo Vistas Controladores (MVC)

La Arquitectura es el esqueleto o base de una aplicación. Representa la organización fundamental de un sistema. Desde los pequeños programas hasta los sistemas más grande poseen una estructura y un comportamiento que los hace clasificables según su "arquitectura". En la Web es muy común la utilización de la arquitectura "3-capas", "n-capas" "MVC", entre otras. El patrón arquitectónico MVC, es por ello que adoptamos esta arquitectura para el desarrollo de la propuesta de solución. Modelo Vista Controlador (MVC)

Es un patrón de diseño de arquitectura de software que separa los datos de una aplicación la interfaz de usuario, y la lógica de control en tres componentes distintos.



Figura 2: Modelo Vistas Controladores.

- **Modelo:** Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- **Vista:** Presenta el modelo en un formato adecuado, como en una página Web que le permite al usuario interactuar con ella, usualmente un elemento de interfaz de usuario.
- **Controlador:** Responde a eventos, usualmente acciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) logrando un mantenimiento más rápido y sencillo de las aplicaciones.

Ejemplo, para el caso de la Web, si se fuera a mostrar una misma aplicación en un navegador estándar, como en un navegador de un dispositivo móvil, sólo es necesario crear una vista nueva por cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (Aplicación de escritorio, HTTP, consola de comandos, e-mail, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación (Basulto 2010).

Ventajas del MVC

- La separación del Modelo de la Vista, es decir, separa los datos de la representación visual de los mismos.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Permite el escalamiento de la aplicación en caso de ser requerido.

1.5 Tecnologías para el desarrollo del sistema.

El código asociado a una página, o servicio Web, puede ser ejecutado en el servidor (PHP, ASP, Java, etc.) o en el cliente (JavaScript, VBScript), permitiendo el envío/recibo de información entre el servidor y la máquina cliente a través de un documento con formato estándar ejecutado del lado del cliente (HTML, XML).

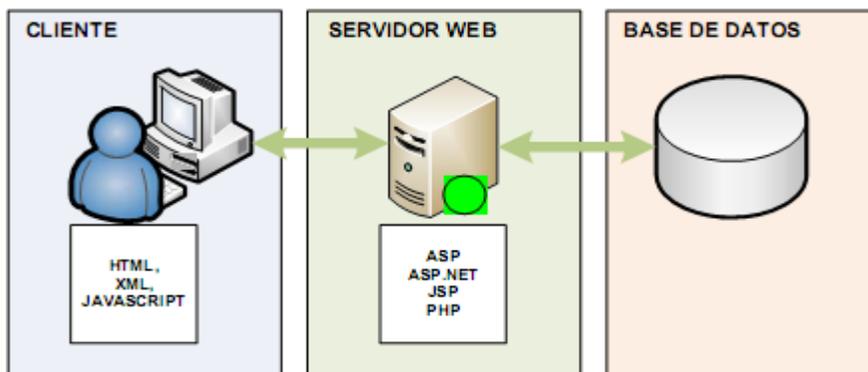


Figura 3: Tecnologías para el desarrollo aplicaciones Web.

1.5.1 Lenguaje empleado del lado del cliente.

- **JavaScript:** este es un lenguaje interpretado por lo que no requiere compilación. Fue creado por la empresa Netscape Communication. Es similar al Java, aunque no es un lenguaje orientado a objetos, no dispone de ninguna de los privilegios de la programación orientada a objetos (POO). La mayoría de los navegadores en sus últimas versiones interpretan JavaScript. El código de JavaScript puede ser integrado a nuestras páginas Web.

Ventajas:

- Lenguaje de scripting fiable y seguro.
- Los scripts tienen capacidad limitada, por razones de seguridad.
- El código se ejecuta en lado del cliente.

Desventajas:

- Código visible por cualquier usuario.
 - El código debe descargarse completamente.
 - Puede poner en riesgo la seguridad del sitio.
-
- **HTML:** significa lenguaje de etiquetas de hipertexto, en donde hipertexto hace referencia a la capacidad del lenguaje para a utilización de los conocidos hipervínculos. Este lenguaje está compuesto por etiquetas o marcas, y gracias a ella es posible dar forma a todos los componentes de una página.
Un documento HTML no es otra cosa que un archivo de texto que se puede crear con cualquier editor.

1.5.2 Lenguajes del lado del servidor.

- **PHP** acrónimo de *Hypertext Preprocessor*. Fue creado originalmente en 1994 Rasmus Lerdorf, pero como está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores.

PHP es un lenguaje de programación independiente de plataforma, rápido, con una gran librería de funciones, interpretado del lado del servidor utilizado para la generación de páginas Web dinámicas, embebidas con HTML. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. PHP es un producto de código abierto lo que quiere decir que se puede acceder a su código, puede utilizarlo, modificarlo, y redistribuirlo sin coste alguno. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Es independiente de plataforma, puesto que existe un módulo PHP para casi cualquier servidor Web, permite que cualquier sistema sea compatible con él. Permite además la Programación Orientada a Objetos y posee capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, a partir de PHP en su versión 4 se destaca su conectividad con MySQL y PostgreSQL (Rufino 2007).

Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje rápido.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Soporta en cierta medida la programación orientada a objetos (P.O.O). Clases y Herencias.
- Es libre por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de ningún tipo de variables ni manejo detallado del bajo nivel.

Desventajas:

- Se necesita instalar un servidor Web.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más eficiente a medida que las solicitudes aumenten de número.

- La P.O.O es muy deficiente para aplicaciones grandes.
- Dificulta la organización por capa de la aplicación.

1.6 Herramientas utilizadas en la propuesta de solución.

1.6.1 Dreamweaver como entorno de desarrollo.

Dreamweaver, sin lugar a dudas, es una de las herramientas más utilizadas por los webmasters para el trabajo con aplicaciones visuales en este caso en el diseño e implementación de páginas Web. Se adapta increíblemente a las necesidades de todo tipo de profesional de diseño Web, tanto para lo que prefieren programar el código directamente en el editor de texto como para los que gustan del ambiente visual.

Se trata de un editor de texto especialmente diseñado para trabajar con documentos Web como HTML, PHP, ASP, JavaScript, entre otros (Ruiz 2004).

1.6.2 Sistema gestor de Bases de Datos.

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes y acrónimos respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System. Actualmente existen muchos sistemas gestores de bases de datos, entre ellos, están: MySQL, PostgreSQL y Microsoft SQL Server.

1.6.2.1 ¿Qué es PostgreSQL?

- PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde la década de 1980.

- El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto.
- PostgreSQL es ampliamente considerado como una de las alternativas de sistema de bases de datos de código abierto.

Ventajas:

- Inhalación ilimitada: Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.
- Soporte: Además de nuestras ofertas de soporte, tenemos una importante comunidad de profesionales y entusiastas de PostgreSQL de los que su compañía puede obtener beneficios y contribuir.
- Ahorros considerables en costos de operación: PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento.
- Estabilidad y Confiabilidad Legendarias: PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad. Ni una sola vez. Simplemente funciona.
- Extensible: El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.
- Multiplataforma: PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y ahora en versión nativa para Windows.

- Diseñado para ambientes de alto volumen: PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes.
- Herramientas gráficas de diseño y administración de BD: Existen varias herramientas gráficas de alta calidad para administrar las bases de (RUP)Data Architect).

1.6.2 Paquete de tecnología Web.

XAMPP: es un paquete formado por un servidor Web apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl. El nombre proviene de **X** (para cualquier sistema operativo), **A** (Apache), **M** (MySQL), **P** (PHP), **P** (Perl). El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible los sistemas operativos Windows, Linux, Solaris, MacOS X. XAMPP es regularmente actualizado para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. Incluye otros módulos como OpenSSL y phpMyAdmin(XAMPP 2008).

1.6.3 Embarcadero ER/Studio 8.0.

Es una herramienta de modelado de datos, se usa para el diseño y la construcción lógica y física de bases de datos. Su ambiente es de gran alcance y multinivel. Simple y fácil al usuario, ayuda a las organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia y alcanza en última instancia usos de más alta calidad que entreguen datos más eficientes y exactos a la empresa de desarrollo(ER/STUDIO).

Ventajas:

- Si se está comenzando un nuevo diseño o está manteniendo una base de datos existente, ER/Studio se combina con las características para ayudarle a conseguir el trabajo hecho con eficacia.
- La creación de diagramas es clara y rápida.

- Tiene la posibilidad de realizar diagramas con desempeño rápido.

1.6.4 PhpDesigner 2008 v6.0.0

Es un completo entorno de desarrollo y programación especialmente diseñado para los “gurús” de PHP, aunque también permite trabajar con comodidad en otros lenguajes de programación como HTML, XHTML, CSS y SQL.

Ofrece toda una serie de asistentes y diálogos integrados que facilitan en todo de uso habitual, utilidades diversas y toda suerte de herramientas, todo ello en una interfaz de diseño sencillo y elegante que se puede personalizar con nada menos que dieciocho temas distintos.

Cuenta con cliente de FTP y navegador de ficheros integrado, utilidades de corrección y autocompletado, búsqueda integrada en google y soporte para proyectos, además de usar un práctico esquema de color para la sintaxis del código fuente que facilita enormemente la programación.

PhpDesigner soporta: PHP, HTML, XHTML, CSS, Java, Perl, JavaScript, VB, C# y SQL.(Professional 2005)

1.7 Fundamentación de la selección de las herramientas utilizadas para la implementación del producto software.

Independientemente de que en esta investigación no se abordaron otras tecnologías para el desarrollo de aplicaciones Web, estas no pierden su factibilidad y su competitividad, por lo que solo se hace referencia a las tecnologías propuestas para el desarrollo de la aplicación:

- ❖ Lenguaje de programación *Processor Hypertext* (PHP).
- ❖ Gestor de Base de Datos, PostgreSQL.
- ❖ Servidor Web, Apache del paquete de tecnologías XAMMP.

Debido también que en la política de desarrollo de software en el ISMMM se estipula que se debe desarrollar bajo el paradigma de “**Open Source**”.

1.8 Tendencias y tecnologías actuales.

1.8.1 Metodologías para el desarrollo de productos software.

Hoy en día, llevar a cabo el desarrollo de un buen software depende de un gran número de actividades y etapas donde elegir la mejor metodología, para el equipo influye directamente en el futuro éxito del producto. El papel preponderante de las metodologías es sin duda esencial en un proyecto y en el paso inicial, que debe encajar en el equipo, guiar y organizar actividades que conlleven a las metas trazadas en el grupo.

Una metodología para el desarrollo de un proceso de software es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas informáticos.

Las metodologías existentes en la actualidad se dividen en dos grandes grupos atendiendo a sus características: **las metodologías tradicionales** (RUP, MSF) y **las metodologías ágiles** (XP, SCRUM). Las primeras están pensadas para el uso exhaustivo de documentación durante todo el ciclo del proyecto mientras que las segundas ponen vital importancia en la capacidad de respuesta a los cambios, la confianza en las habilidades del equipo y al mantener una buena relación con el cliente. **(RUP)**

Teniendo en cuenta ambos enfoques damos paso al análisis de dos de las metodologías más usadas actualmente.

1.8.2 Proceso Unificado de Desarrollo (RUP).

Es un proceso para el desarrollo de un software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Como tres características esenciales está dirigido por casos de uso: que orientan al proyecto a la importancia para el usuario y lo que se quiere, está centrado en la arquitectura: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es iterativo e incremental: donde divide el proyecto en mini-proyectos donde los casos de uso y al arquitectura cumplen sus objetivos de manera depurada. RUP propone cuatro etapas para el desarrollo de un producto:

Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. Estas etapas revelan que para producir una versión del producto en desarrollo se emplean todas las actividades de ingeniería pero con diferente énfasis; en las primeras versiones se hace más énfasis en el modelado del negocio, requisitos, análisis y diseño; mientras en las posteriores el énfasis recae sobre las actividades de implementación, pruebas y despliegue. Además contempla flujos de trabajo de soporte que involucran actividades de planificación de recursos humanos tecnológicos y financieros. El Proceso Unificado de Desarrollo tiene 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. Flujos de trabajo(Tradicional):

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

1.8.3 Metodología ágil desarrollo de software.

Se entiende como desarrollo ágil software a un paradigma de desarrollo de software basados en procesos ágiles. Los procesos ágiles de software, conocido anteriormente como metodologías livianas, intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

Es un marco de trabajo conceptual de la ingeniería de software que promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto. Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando software en cortos lapsos de tiempo. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener un demo (sin errores) al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.

Los métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación. La mayoría de los equipos ágiles están localizados en una simple oficina abierta, a veces llamadas "plataformas de lanzamiento" (bullpen en inglés). La oficina debe incluir revisores, escritores de documentación y ayuda, diseñadores de iteración y directores de proyecto. Los métodos ágiles también enfatizan que el software funcional es la primera medida del progreso. Combinado con la preferencia por las comunicaciones cara a cara, generalmente los métodos

ágiles son criticados y tratados como "indisciplinados" por la falta de documentación técnica(Ferrer 2007).

1.8.3.1 Un proceso es ágil cuando el desarrollo de software es:

- **Incremental.** Entregas pequeñas de software, con ciclos rápidos.
- **Cooperativo.** Cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación.
- **Sencillo.** El método en sí mismo es simple, fácil de aprender y modificar.
- **Esta bien documentado y es adaptable.** Permite realizar cambios de último momento).

Programación Extrema (Extreme Programming ->XP).

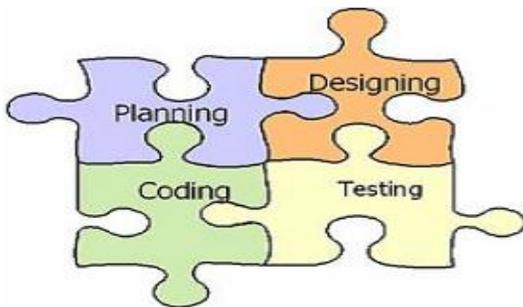


Figura 4. Metodología XP.

1.8.3.2 Programación Extrema.(eXtreme programming, XP)

XP, es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, que describe la filosofía de XP en el "Manifiesto Ágil" 5, sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea(Wesley 2000).

1.8.3.3 Características fundamentales de XP.

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias: continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Programación por parejas: se recomienda que las tareas de desarrollo e lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera -el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.
- Frecuente interacción: del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Refactorización del código:, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo(Letelier 2008).

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre

este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

Ventajas

- Apropiado para entornos volátiles.
- Estar preparados para el cambio, significa reducir su coste.
- Planificación más transparente para los clientes, ya conocen las fechas de entrega de funcionalidades. Vital para su negocio.
- Permite definir en cada iteración cuales son los objetivos de la siguiente.
- Permite la retroalimentación.
- La presión esta a lo largo de todo el proyecto y no en una entrega final.

Desventajas

- Delimitar el alcance del proyecto con nuestro cliente.

Para mitigar esta desventaja se plantea definir un alcance a alto nivel basado en la experiencia.

1.8.3.4 Proceso XP

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

- 1) El cliente define el valor de negocio a implementar.
- 2) El programador estima el esfuerzo necesario para su implementación.
- 3) El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- 4) El programador construye ese valor de negocio.
- 5) Vuelve al paso 1.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

Modelo de un Proyecto XP.

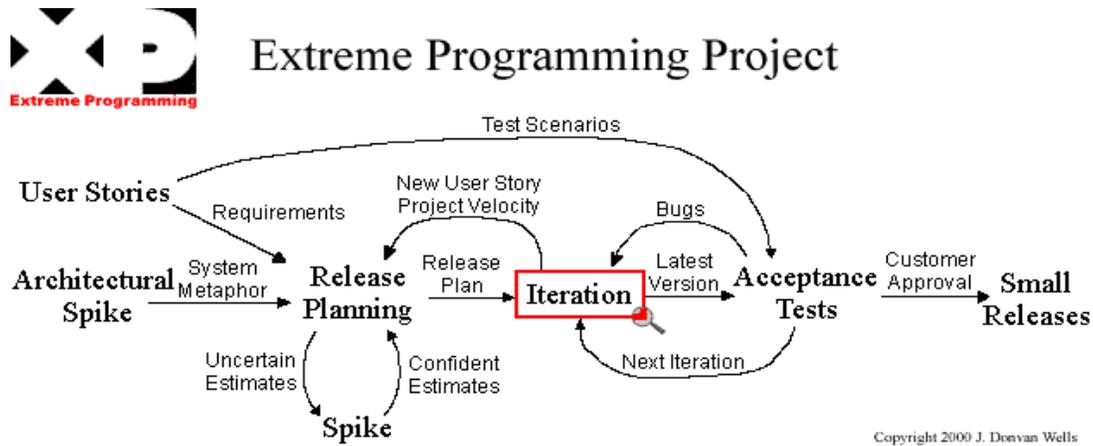


Figura 5. Modelo de un proyecto XP.

1.8.3.5 SCRUM

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el *ScrumMaster*, que mantiene los procesos y trabaja de forma similar al director de proyecto, el *ProductOwner*, que representa a los *stakeholders* (clientes externos o internos), y el *Team* que incluye a los desarrolladores.

Durante cada *sprint*, un periodo entre 15 y 30 días (la magnitud es definida por el equipo), el equipo crea un incremento de software *potencialmente entregable* (utilizable). El conjunto de características que forma parte de cada sprint viene del *ProductBacklog*, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del *ProductBacklog* que forman parte del sprint se determinan durante la reunión de *Sprint Planning*. Durante esta reunión, el *ProductOwner* identifica los elementos del *Product Backlog* que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint.

(Zulueta 2010)Scrum permite la creación de equipos autoorganizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado *requirements churn*), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes(Zulueta 2010).

1.9 Fundamentación de la selección de la metodología de desarrollo de software.

XP, SCRUM y RUP son grandes metodologías que hoy en día son de las más utilizadas para el análisis, diseño e implementación de un producto software por medianas y grandes empresas ligadas al desarrollo de producto software. Se ha determinado en la concepción de este trabajo la implantación de XP como metodología de desarrollo, ya que es una metodología ligera que requiere de poco requerimiento de documentación y planificación para el desarrollo de la aplicación. Si se escogiese una metodología tradicional o sea RUP traería grandes problemas como son:

- ❖ **Multitud de artefactos:** El hecho de realizar varios artefactos y mantenerlos actualizados consume mucho tiempo.
- ❖ **Requisitos cambiantes:** Los cambios en un proceso de desarrollo son inevitables, al aparecer un nuevo requisito hace que se tenga que comenzar una nueva iteración para dar cumplimiento a su funcionalidad. Como el proyecto esta en plena investigación y en un ambiente de desarrollo sujeto a cambios repentinos se sugieren una gran adaptatividad y pronta respuesta, lo cual RUP no ofrece.

- ❖ **Planificación inexistente:** La planificación que se realiza en las fases iniciales está sujeta a muchas variaciones en dependencia de los cambios que se experimenten en los requisitos. Por tanto se hace muy difícil planificar actividades específicas si no se tiene claro que se debe hacer realmente.
- ❖ **Se necesita:** La documentación mínima necesaria para el futuro soporte y mantenimiento del producto final.

Todos estos inconvenientes son erradicados con la implementación de la metodología **XP**.

1.10 Conclusiones del Capítulo

En el presente capítulo se abordaron los principales conceptos y definiciones en el que se enmarca el dominio del problema en cuestión, para cumplimentar el estudio del estado del arte del presente trabajo. Se realizó una investigación profunda de los sistemas automatizados existentes unidos al campo de acción de nuestra investigación, lo que arrojó que ninguno cumple con las exigencias requeridas.

Se conceptualizó y caracterizó de forma flexible las herramientas y lenguajes para la implementación de la aplicación. Además se expuso el porque de ajustarnos a las cualidades de las metodología Ingeniería de Software XP. Se hizo también referencia a la arquitectura en tres capas ya que será un pilar fundamental para la implementación del futuro sistema.

Capítulo 2

PLANIFICACIÓN Y DISEÑO

Introducción

En este capítulo, se introduce a las fases de planeación y diseño, donde se especifican las necesidades del cliente, se describen las funcionalidades que serán objeto de automatización mediante el empleo de las historias de usuarios (HU), se realiza una estimación del esfuerzo necesario para las mismas y se establece un plan de iteraciones necesarias sobre el sistema, para su terminación. Uno de los artefactos principales es la creación de las tarjetas CRC (Clases-Responsabilidades-Colaboración) la cuales son un paradigma de la programación orientada a objetos, que permitirá trabajar con objetos

2 Funcionalidades generales.

El control y la gestión de los reportes de las averías de los equipos de cómputo en el ISMMM, se realiza de forma manual, lo que hace el proceso más lento e ineficiente, por lo que se incurre en la demora de las reparaciones de los equipos, todo esto tributando con el gasto excesivo de papeles y otros recursos de oficina. Todo esto incrementando la posibilidad que se produzcan errores a la hora de manipular la información, lo que arroja muchas veces afectaciones en el proceso docente y productivo del ISMMM.

Tabla1: Personal relacionada con el sistema.

Personas relacionadas con el sistema	Justificación
Administrador	Este es el encargado de dar bajas y altas del sistema, mostrar distintos reportes que arroja el sistema como parte de sus funcionalidades.
Usuarios	Es la persona encargada de reportar las quejas de averías de las PCs y las de mantenimiento.
Técnicos	Este es le encargado de atender los reportes de averías y mantenimiento.

2.1 Lista de reserva.

Después de conocer el personal relacionado e identificar los requisitos generales, se procede a realizar el análisis de las funcionalidades que debe cumplir la aplicación para dar respuesta a los mismos. Para ello se enumerarán mediante una lista de reserva, las funcionalidades que el sistema debe ser capaz de cumplir.

De acuerdo a lo antes expuesto, el sistema debe ser capaz de:

- Reportar averías.
- Buscar reportes específicos.
 - Búsqueda por número de inventario.
 - Búsqueda por nombre de usuarios.
- Obtener estadísticas de los reportes gestionados en el módulo.
- Obtener reportes:
 - Reportes pendientes.
 - Equipos pendientes por piezas.
 - Reportes terminados.
- Gestionar usuarios del sistema:
 - Insertar usuarios.
 - Modificar usuarios.
 - Eliminar usuarios.
 - Ver usuarios del sistema.
- Elaborar plan de mantenimiento.
- Gestionar computadoras.
 - Insertar computadoras.
 - Modificar computadoras.
 - Dar de baja a computadoras.
 - Ver estado de computadoras.
- Autenticar usuario.
 - Cambiar contraseña.
- Atender avisos de reportes de averías.
- Atender avisos de mantenimiento.
- Exportar los datos arrojados por el sistema a formatos de impresión digitales.
- Asignar Subareas.
- Reportar avisos de mantenimiento.

- Gestionar dispositivos.
 - Eliminar dispositivos.
 - Inserta dispositivos.
 - Modificar dispositivos.

2.2 Historias de Usuarios (HU).

Las HU, son la técnica utilizada en XP para detallar los requisitos del software. Son el resultado directo del intercambio entre los usuarios y desarrolladores a través de reuniones donde las conocidas tormenta de ideas (brain storm) arrojan no solo los requerimientos, sino también las posibles soluciones; representan una forma rápida de administrar las necesidades de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para gestionarlos, debido a que un requerimiento de software es descrito de forma concreta y sencilla utilizando el lenguaje común del usuario.

Las HU permiten responder ágilmente a los requerimientos cambiantes y aunque se redactan desde las perspectivas de los clientes, también los desarrolladores pueden brindar ayuda en la identificación de las mismas. Para definir las se emplea la siguiente plantilla (Beck 2002).

Modelo de planilla de historia de usuario

Tabla2: Plantilla de Historia de Usuario.

Historia de usuarios	
Número: No. HU	Usuario: Usuario entrevistado para obtener información.
Nombre: Nombre de la historia de usuario para identificarla.	
Prioridad en el negocio: Importancia: Alta / Media / Baja	Riesgo en desarrollo: Dificultad: Alta / Media / Baja
Puntos estimados: Estimación: de 1 a 3 puntos	Iteración asignada: Iteración a la que corresponde
Programador responsable: Nombre de encargado de programación.	
Descripción: Una breve descripción del lo que lo que realizará la HU.	
Observaciones: Algunas observaciones de interés.	

HU No.1 Reportar Averías.

Tabla3: HU Reportar Averías.

Historia de usuarios	
Número: 1	Usuario: Usuarios y Administrador del sistema.
Nombre: Reportar de averías.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Los usuarios y el administrador (en caso que lo requiera) del sistema deben suministrarle al sistema la información pertinente al reporte de una avería de una computadora, estos datos se almacenarán en BD para su posterior procesamiento.	
Observaciones: Confirmado con el cliente.	

Para consultar el resto de las HU ir: [Anexo1](#)

2.3 Planificación de entregas.

En esta parte se establece la prioridad de cada Historia de Usuario así como una estimación del esfuerzo necesario de cada una de ellas con el fin de determinar un cronograma de entregas. Las estimaciones de esfuerzo asociado a la implementación de las historias se establecen utilizando como medida, el punto. Un punto, equivale a una semana ideal de programación (6 días). Las historias generalmente valen de 1 a 3 puntos. Por otra parte, se mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de

puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

2.3.1 Estimación de esfuerzo por HU.

Tabla4: Estimación de esfuerzo por HU.

Historia de usuario	Número	Puntos Estimados
Reportar averías	1	1
Buscar reportes	2	2
Obtener estadísticas de reportes	3	3
Obtener reportes	4	4
Gestionar usuarios del sistema	5	3
Elaborar plan de mantenimiento	6	1
Gestionar computadoras	7	1
Autenticar usuarios	8	1
Atender avisos de reportes de averías	9	1
Atender avisos de mantenimiento	10	1
Exportar a formato digital	11	3
Gestionar dispositivos.	12	2
Reportar avisos de mantenimiento	13	2
Asignar Subareas	14	1

El plan de entregas se realiza teniendo en cuenta las unidades funcionales que se quieren entregar y cada uno de estos módulos abarca un número de historias de usuarios a implementar para dar cumplimiento al funcionamiento del mismo.

2.3.2 Planificación de entregas.

Tabla5: Plan de entregas.

Módulo	Historia(s) de Usuario que abarca
✓ Gestionar Elementos	5, 8, 7, 12, 14
✓ Obtención de Reportes	1, 2, 3, 4
✓ Exportar	11
✓ Plan de Mantenimiento	6, 9, 10, 13

2.3.3 Plan de duración de las iteraciones.

Con respecto a las Historias de Usuario previamente presentadas realizamos una planificación en 4 iteraciones basándonos en el tiempo y procurando obtener la funcionalidad relacionada en la misma iteración.

Tabla6: Duración de iteraciones.

Iteración	Orden de implementación por Historias de Usuario	Duración total de la iteración en semanas
1	<ul style="list-style-type: none"> ✓ Gestionar usuarios del sistema ✓ Gestionar computadoras. ✓ Autenticar usuarios. ✓ Gestionar dispositivos ✓ Asignar subareas 	2+2+1+1+1=7
2	<ul style="list-style-type: none"> ✓ Reportar averías ✓ Buscar un reporte específico ✓ Obtener estadísticas de reportes ✓ Obtener reportes 	1+2+1+2=6
3	<ul style="list-style-type: none"> ✓ Elaborar reporte de mantenimiento ✓ Atender avisos de mantenimiento ✓ Reportar avisos de mantenimiento 	1+1+1=3
4	<ul style="list-style-type: none"> ✓ Exportar a formato digital 	1

Combinando el plan de entrega y el plan de iteraciones se harán releases o liberaciones al sistema en las fechas mostradas a continuación:

Módulo #1 = Gestionar Elementos.

Módulo #2 = Obtención de reportes.

Módulo #3 = Exportar.

Módulo #4 = Plan de mantenimiento.

Tabla7: Tabla de releases.

Iteración \ Módulo	Modulo #1	Modulo #2	Modulo #3	Modulo #4
Final 1ra iteración	20/03/2011			
Final 2da iteración		5/06/2011		
Final 3ra iteración			8/06/2011	
Final 4ta iteración				25/05/2011

2.4 Tarjetas Clases-Responsabilidades-Colaboración (CRC).

El uso de las tarjetas C.R.C (Clases, Responsabilidades y Colaboración) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica.

Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad. Esta nueva técnica de diseño es adoptada como alternativa a los diagramas UML de las clases, pues en estas se plasman las responsabilidades que tienen cada objeto y las clases con las que tienen que interactuar para darles respuesta brindando así la información que se necesita a la hora de implementar(Beck 2004).

Tabla8: Plantilla de tarjeta CRC.

Clases	
Responsabilidades	Colaboraciones
Clase A	Clase B

Para una mejor comprensión de las tarjetas C.R.C de nuestro sistema procedemos a agruparlas por módulos identificados en el [Plan de Entregas](#) en el Capítulo 2.

2.4.1 Tarjeta C.R.C Módulo #1: Gestionar Elementos.

Tabla9: Tarjeta CRC.

Clases: Autenticar_users	
Responsabilidades	Colaboraciones
loguearse_Users Cambiar_Contrasenna.	Conexxion.

Para consultar el resto de las tarjetas C.R.C ir: [Anexo-2](#)

2.5 Conclusiones del Capítulo

Con la culminación de este capítulo se han desarrollado las bases con las que se sustentarán las necesidades del cliente, se identificaron las HU con la participación conjunta del cliente y usuarios, destacando la planificación de cada HU por la prioridad de sus iteraciones o sea a partir del esfuerzo de las mismas, culminado así esta fase y se determina que el equipo de trabajo está listo para pasar a la siguiente etapa. Enfocándose en la programación orientada a objetos dentro de la fase de diseño de la metodología XP, se elaboraron las tarjetas CRC.

Capítulo 3

IMPLEMENTACIÓN Y PRUEBAS

Introducción

En este capítulo se desarrollará la fase de implementación y pruebas conforme a lo que determina la metodología XP. Por otra parte se describen cada una de las tareas confeccionadas para cumplir con el desarrollo de cada una de las HU definidas. Además se mostrarán las pruebas de aceptación confeccionadas por el cliente para comprobar que la aplicación funcione correctamente. Estas pruebas fueron realizadas durante la entrega que se efectuaban a lo largo del desarrollo del proyecto.

3 Implementación

En la metodología XP se convierte en un integrante más del equipo de desarrollo el cliente pues él crea las historias de usuario bajo la supervisión de los desarrolladores. Estas historias quedan confeccionadas cuando el cliente es capaz de identificar con precisión la funcionalidad deseada, además, también debe estar presente cuando se realicen las pruebas de aceptación para cada historia, por lo que su presencia es imprescindible.

En XP generalmente cada historia de usuario se divide en tareas de ingeniería (TI) o tareas de programación. Estas se crean para obtener una mejor planificación de la historia; con ellas se pretende cumplir con las funcionalidades básicas que luego conformarán las funcionalidades generales de cada historia.

Tabla10: Plantilla de Tareas de Ingeniería.

Tarea de Programación	
Número tarea:	Número historia:
Nombre tarea:	
Tipo de tarea:	Puntos estimados:
Fecha inicio:	Fecha fin:
Programador responsable:	
Descripción:	

A continuación se presentan las Tareas de Ingeniería agrupadas por las respectivas historias de usuario a las que pertenecen.

3.1 Tareas por HU

3.1.1 Tareas de la HU #1

Tabla11: Tareas de Ingeniería.

Tarea de Programación	
Número tarea: 1	Número historia: 1
Nombre tarea: Presentación de la vista de los datos de reporte de averías.	
Tipo de tarea: Diseño.	Puntos estimados:1
Fecha inicio: 10/04/2011	Fecha fin:12/04/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Los datos que se suministran por el formulario son evaluados para ver el grado de veracidad de los datos; o sea se valida la correcta entrada de datos y si no hay campos vacíos.	

Para consultar el resto de las Tareas de Ingenierías ir: [Anexo-3](#)

3.2 Prueba

3.2.1 Pruebas de Aceptación (PA).

Uno de las mejores características de la metodología XP es el proceso de pruebas. Esta metodología propone probar tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos no deseados a la hora de realizar modificaciones y refactorizaciones. XP propone la realización de pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente(Wesley 2000).

En este proyecto al codificar no se sigue la regla de XP que aconseja crear pruebas unitarias con entornos de desarrollo antes de programar. Las obtuvieron de la descripción de requisitos plasmados en las historias de usuarios, y estas especifican las barreras que deben pasar las distintas funcionalidades del programa, procurando codificar teniendo en cuenta las pruebas que se deben vencer. Para realizar las pruebas de aceptación el cliente utiliza la siguiente plantilla.

Tabla12: Plantilla Prueba de Aceptación.

Prueba de aceptación
HU: Nombre de la historia de usuario que va a comprobar su funcionamiento.
Nombre: Nombre del caso de prueba.
Descripción: Descripción del propósito de la prueba.
Condiciones de ejecución: Precondiciones para que la prueba se pueda realizar.
Entrada/Pasos de ejecución: Pasos para probar la funcionalidad.
Resultado esperado: Resultado que se desea de la prueba.
Evaluación de la prueba: Aceptada o Denegada.

A continuación se presentan las pruebas que se tuvieron en cuenta para verificar el buen funcionamiento de cada módulo en las entregas que se le hacen al cliente cumpliendo con lo establecido en el cronograma de entregas.

3.2.2 Pruebas del módulo #2: Obtención de reportes.

Tabla13: P.A Reportar averías.

Prueba de aceptación
HU: Reportar averías
Nombre: reporte de averías
Descripción: los usuarios del sistema y el administrador (en caso que lo requiera), le suministra los datos pertinentes al sistema.
Condiciones de ejecución: El usuario debe entrar los datos correctos al sistema.
Entrada/Pasos de ejecución: Se verifica que el reporte nuevo no se encuentre en la BD.
Resultado esperado: Se efectúa correctamente el reporte de avería.
Evaluación de la prueba: Satisfactoria.

Para consultar el resto de las Pruebas de Aceptación ir: [Anexo-4](#)

3.3 Modelo lógico de datos.

Aquí se presentan todas las tablas con la que interactúa con el sistema propuesto en esta investigación; se recogen y modelan todos los datos con lo que dispondrá la aplicación.

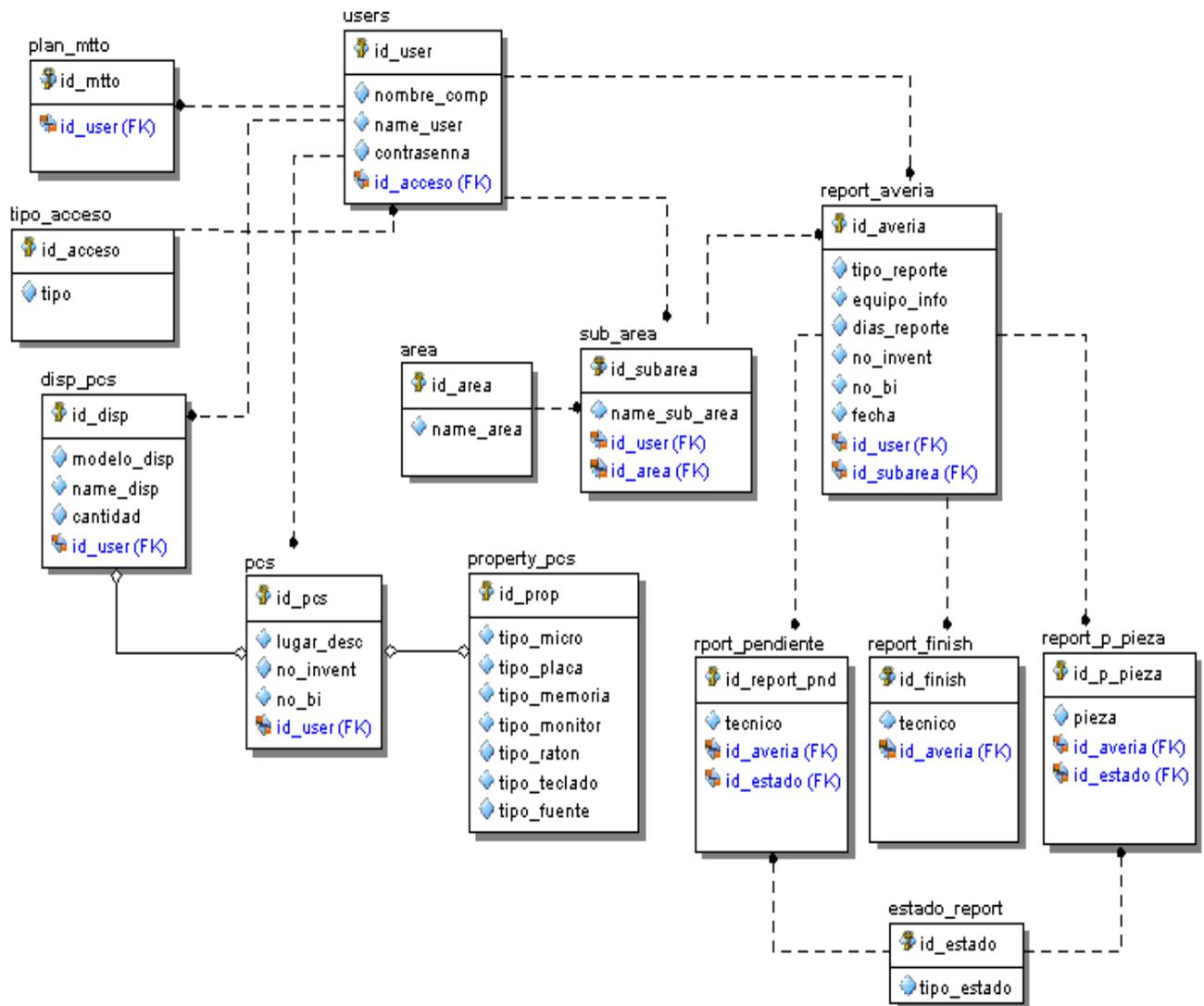


Figura 6. Modelo de datos.

3.4 Modelo físico de datos.

Para obtener más información sobre el modelado físico de los datos ir [Anexo-5](#)

3.5 Conclusiones del Capítulo

Para lograr la completa implementación de cada historia de usuario en la fecha acordada con el cliente, estas se dividieron en tareas de ingeniería. A cada TI se le asignó un tiempo de desarrollo que se cumplió de manera eficiente garantizando así el objetivo principal de su confección.

Con la realización de las pruebas de aceptación el cliente se asegura de que las funciones implementadas cumplan su objetivo satisfactoriamente, probando individualmente cada módulo y asignándole la evaluación correspondiente. Todas las pruebas que se realizaron fueron positivas y el cliente estuvo conforme, cumpliendo entonces la aplicación con las historias de usuarios definidas inicialmente.

Capítulo 4

ESTUDIO DE FACTIBILIDAD DEL PROYECTO

Introducción

En la actualidad para un satisfactorio desarrollo de cualquier proyecto se hace imprescindible el estudio de factibilidad para tener en cuenta una estimación de los costos a incurrir logrando así definir si será factible o no desarrollar el mismo. Para el estudio de factibilidad de este proyecto se utilizará la Metodología Costo Efectividad (Beneficio), la cual plantea que la conveniencia de la ejecución de un proyecto se determina por la observación de dos factores en conjunto:

- El costo que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware/software y los costos de operación asociados.
- La efectividad que se entiende como capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo por el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad del cumplimiento del objetivo).

Esta es una de las partes más importantes en la elaboración de cualquier proyecto ya que haciendo un correcto estudio de factibilidad se puede ahorrar meses e incluso años de trabajo, hasta evitar poner en duda la reputación profesional si se realiza un sistema mal planificado desde una etapa temprana.

Para estudiar la factibilidad de este proyecto se utilizará la metodología.

4 Efectos Económicos.

- ❖ Efectos directos.
- ❖ Efectos indirectos.
- ❖ Efectos externos.
- ❖ Intangibles.

Efectos directos:

POSITIVOS:

- ❖ Los usuarios autorizados a manipular el sistema tendrán la cobertura de reportar de forma on-line cualquier tipo de anomalía que presenten los medios informáticos en las áreas correspondientes a ellos.
- ❖ Se mejora la eficiencia del proceso de gestión de reportes de averías de los medios de cómputos.
- ❖ Se agilizará el proceso de obtención de reportes de las computadoras que están en espera de reparación, la que ya están terminadas y la que están falta de algún dispositivo.
- ❖ Se facilitará el proceso de llevar toda la gestión de datos a un tipo de formato de impresión.

NEGATIVOS:

- ❖ Para el uso de la esta aplicación implementada con tecnología Web se necesitara que la misma sea ejecutada con el navegador de Firefox debido que el que viene con el SO Windows no interpreta bien los estándares de estilos de diseño que se emplean en la aplicación.

Efectos indirectos:

- ❖ Los efectos económicos observados que pudiera repercutir sobre otros mercados no son perceptibles, aunque este proyecto no está construido con la finalidad de comercializarse.

Efectos externos:

- ❖ Se tendrá una herramienta disponible que le facilitará gran parte del trabajo a los encargados de la gestión y control de los medios informáticos.

Intangibles:

- ❖ En la valoración económica siempre hay elementos perceptibles por una comunidad como perjuicio o beneficio, pero al momento de ponderar en unidades monetarias esto resulta difícil o prácticamente imposible.

A fin de medir con precisión los efectos, deberán considerarse tres situaciones:

- **SITUACIÓN SIN PROYECTO**

Para llevar a cabo la gestión de las averías los responsables del reporte se debe tener en cuenta los siguientes pasos:

1. Ir personalmente donde esta el medio de computo averiado y recopilar los datos correspondientes.
2. Hacerle llegar esta información a los técnicos para que estos procedan con la posible reparación.
3. Gastos excesivos de recursos de oficinas dígase hojas y otros recursos no menos importantes.

Nota #1 Mientras más medios de cómputos averiados existan más tediosos se hará el proceso de reportes por lo que podría incurrirse en errores.

- **SITUACIÓN CON PROYECTO**

Para la entrada de los datos al sistema propuesto debemos seguir los siguientes pasos:

1. Entrar los datos correspondiente del usuario o técnico que hará uso de la aplicación.
2. Proveer al sistema con los datos de la avería correspondiente.
3. Mostrará el estado en que esta los medios informáticos.
4. Obtener de forma inmediata un reporte en formato de impresión.

4.1 Beneficios y Costos Intangibles en el proyecto

Costos

- Resistencia al cambio.

Beneficios

- Mejor como comodidad para los usuarios.
- Mejora la calidad de información por la integridad, oportunidad de la información y confiabilidad.
- Menor tiempo empleado el registro de los datos.
- Facilidad a la hora de interpretar la gestión de reportes de medios de cómputos así como su mantenimiento.

4.2 Ficha de costo

Para determinar el costo económico del proyecto se utilizará el procedimiento para elaborar una Ficha de Costo de un producto informático (García, A.M).

Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional.

Costos en Moneda Librementemente Convertible:

Tabla14: Costo en Moneda Librementemente Convertible.

Ficha de Costo.	
	Precio(s)
Costos Moneda Librementemente Convertible	
Costos Directos	
Compra de equipos de cómputo	0,00
Alquiler de equipos de cómputo	0,00
Compra de licencia de Software	0,00
Depreciación de equipos	25,5
Materiales directos	0,00
Subtotal	25,5
Costos Indirectos	
Formación del personal que elabora el proyecto	0,00
Gastos en llamadas telefónicas	0,00
Gastos para el mantenimiento del centro	0,00
Know How	0,00
Gastos en representación	0,00
Subtotal	0,00
Gastos de Distribución y Venta	
Participación en ferias o exposiciones	0,00
Gastos en transportación	0,00
Compra de materiales de propagandas	0,00
Subtotal	0,00
Total	25,5

Costos en Moneda Nacional:

Tabla15: Costo en Moneda Nacional.

Ficha de Costo.	
	Precio(s)
Costos Moneda Nacional	
Costos Directos	
Salario del personal que laborará en el proyecto	100,00
Seguridad Social	0,00
Vacaciones $100 \times 9,09\% = 9.09$	0,00
Gastos en llamadas telefónicas	0,00
Impuesto por la Fuerza de Trabajo	0,00
Subtotal	100,00
Costos Indirectos	
Know How	0,00
Gasto por consumo de energía eléctrica	278,7
Subtotal	278,7
Total	378,7

Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la Metodología Costo-Efectividad. Dentro de esta metodología la técnica de punto de equilibrio aplicable a proyectos donde los beneficios tangibles no son evidentes, el análisis se basa exclusivamente en los costos. Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tomaremos como costo el tiempo en minutos empleado para resolver los reportes de la verías y el posible mantenimiento y la variable sería la todo el proceso de de reporte de averías y mantenimientos.

Valores de la variable (Solución manual):

- i. Emitir un reporte de avería con todos los parámetros establecidos (5 min.).
- ii. Planificar mantenimientos de computadoras (15 min.).

- iii. Arrojar reportes en formatos digitales para una mejor gestión de lo datos (20 min.).

Valores de la variable (Solución con el software):

- i. Emitir un reporte de avería con todos los parámetros establecidos (0,18 min.).
- ii. Planificar mantenimientos de computadoras (0,78 min.).
- iii. Arrojar reportes en formatos digitales para una mejor gestión de lo datos (0,98min.).

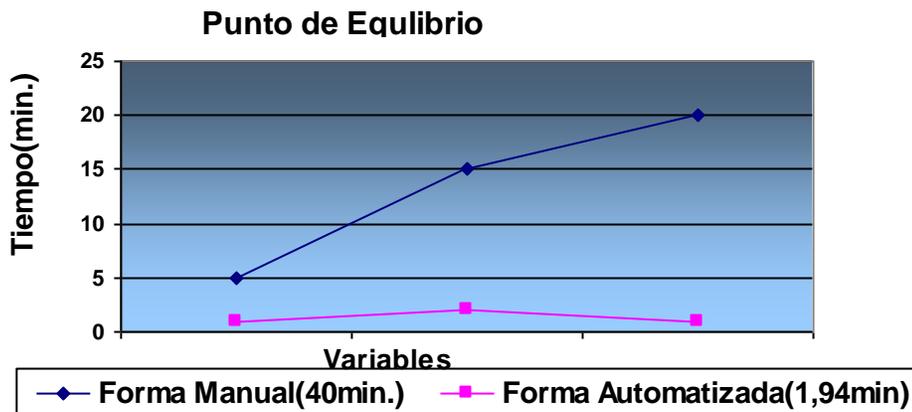


Figura 7. Gráfico punto de equilibrio de soluciones. Teniendo en cuenta los resultados reflejados en la gráfica en cuanto al Punto de Equilibrio queda demostrada la factibilidad del sistema evidenciado por la relación entre la complejidad del problema (cantidad de variables) y el tiempo que demora la solución del mismo de forma manual y automatizada.

4.3 Conclusiones del Capítulo

Al terminar este capítulo se llegó a la conclusión que el producto obtenido es rentable, ya que al aplicar la metodología para calcular la factibilidad se obtuvo que el costo de desarrollo del producto software fue factible porque no se incurrieron en gastos excesivos para su desarrollo. En cuanto los beneficios obtenidos, tantos intangibles como tangibles, decir que: se mejora considerablemente el proceso de reportes de quejas de averías, se obtienen reportes más precisos y confiables y se acelera el mantenimiento de los equipos informáticos. Y un factor importante todo esto tributa en el ahorro de recursos de oficinas.

CONCLUSIONES GENERALES

Llegada la culminación de esta investigación que nos llevó a implementar y desarrollar la aplicación propuesta, se arrojaron las siguientes conclusiones:

- ❖ Se desarrolló el producto final que consistía en la implementación de una aplicación informática con tecnología Web para la gestión y control del reporte de averías y mantenimientos de los medios informáticos en el ISMMM.
- ❖ Se realizó la planificación y diseño de la aplicación, en los cuales se identificaron y especificaron los requerimientos funcionales, así como se llevaron a cabo las posteriores fases de implementación y prueba, que se definen en la metodología de desarrollo utilizada.
- ❖ Se llevó a cabo un estudio de las principales metodologías, lenguajes y herramientas que se consideraron factibles para el desarrollo del sistema.
- ❖ Se realizaron las pruebas de aceptación definidas por el cliente, lo que arrojó como resultado su aprobación, con lo cual se demostró el cumplimiento satisfactorio de las funcionalidades de la aplicación.
- ❖ Se efectuó un estudio detallado de la factibilidad del producto final, el cual arrojó los resultados esperados con costo para el desarrollo del producto software que ascendió a los a los **378,7 CUP** y a **25,5 CUC**.

Como resultado de los objetivos propuestos en la investigación se logró la implementación del producto software que se esperaba. Con el propósito fundamental de simplificar las demoras que se producen en el tratamiento manual de la información, disminuir el grado de errores y para contribuir a elevar la calidad del desarrollo del trabajo.

RECOMENDACIONES

- ❖ Poner en explotación el software en todas las áreas del ISMM para aumentar la capacidad de respuestas en cuanto a reparación y mantenimiento de averías de medios informáticos se refiera.
- ❖ Redefinir las funcionalidades acorde a nuevos requisitos que pudiesen surgir, por inconvergencia o para aumentar el rendimiento del sistema.
- ❖ Realizar un estudio más profundo de este sistema en vista a perfeccionarlo en versiones futuras.
- ❖ Ejecutar preferiblemente la aplicación en navegadores Open-Source debido que traducen los estilos de diseño con mayor claridad.

Con la puesta en práctica de las recomendaciones que se expusieron anteriormente se logrará una capacidad de respuesta más eficiente a la hora de emitir un reporte de queja avería. Además se incentiva para que se creen o mejoren las funcionalidades de la aplicación propuesta, para que conlleven a un mejor rendimiento de las mismas.

FUENTES BIBLIOGRÁFICAS

- Basulto, M. (2010). "Sistema de gestión integral de la empresa empleadora del Níquel - módulo gestión de contratos de compras."
- Beck, K. (2002). " *Test-Driven Development By Example*, Addison-Wesley Professional."
- Beck, K. (2004). " *Extreme Programming Explained: Embrace Change*, Addison-Wesley Professional."
- ER/STUDIO. "Herramienta para el modelaje de datos Embarcadero." from Disponible en:
http://bureaudeprensa.com/es/view.php?bn=bureaudeprensa_software&ke.
- Ferrer, J. (2007). " *Metodología ágil para el desarrollo de software*."
- García, A.M. " *Procedimiento para la elaboración de la ficha de costo de un producto informático*". Facultad MFC UCLV. Villa Clara.
- Letelier, P. y P., M. C. (2008). " *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Valencia, Universidad Politécnica de Valencia."
- Monmany, J. (2009). "Aplicaciones Web." from Disponible en:
<http://www.webvillage.info>
- Monografias.com. (2007). " *Definición arquitectura cliente servidor*." from
<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>.
- Morejón, Y. (1990). " *¿Qué voy a estudiar?. Perfil del ingeniero informático*."
- Professional, P. D. v. (2005). " *Herramienta para el desarrollo Web con PHP*." from Disponible en: <http://www.intercambiosvirtuales.org/software/php-designer-2008-v6020-professional>
- Rodríguez, M. y I. D. (2008). " *Herramienta para el control de los medios informáticos de la Empresa Cdte. "Ernesto Che Guevara"*."
- Rufino, A. (2007). "Lenguaje de programación: PHP." from Disponible en:
<http://1sinfo.blogspot.com/2007/05/lenguaje-de-programacin-php.html>.
- Ruiz, M. H. (2004). " *Programación Web avanzada*."

RUP. "Metodología de desarrollo software RUP." from Disponible en:
http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational

Subiros, D. (2009). "Desarrollo de una interfaz gráfica de usuario para el preprocesador meteorológico AERMET."

Subiros, D. (2009). "Desarrollo de una interfaz gráfica de usuario para el preprocesador meteorológico AERMET." 75.

Tradicional, R. M. "Proceso Unificado de desarrollo de software." from Disponible en: http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado.

Wesley, A. (2000). "Una explicación de la programación extrema. Aceptar el cambio."

XAMPP. (2008). "Paquete de tecnología Web Xampp ", from Disponible en: <http://e.wikipedia.org/wiki/XAMPP>.

Zulueta, A. (2010). "Modulo para la Extracción, Preprocesamiento, Descripción y Almacenaje en formato XML, de la información recuperada por el Sistema Automatizado de Información Virtual del ISMMM."

ANEXOS

Anexo-1 Historias de Usuarios

Historias de Usuario que se abordan en el dominio del sistema que se desea automatizar. (Cont.)

Tabla16: HU Buscar reporte.

Historia de usuarios	
Número: 2	Usuario: Administrador
Nombre: Buscar reporte.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Yadriel Vargas Padilla.	
Descripción: el administrador hace una búsqueda de los reportes, y obtiene una detallada información que le pudiese interesar en un momento determinado.	
Observaciones: la búsqueda se hará según las exigencias que desee el administrador: <ul style="list-style-type: none"> • Reporte específico. • Reporte efectuado por un usuario. 	

Tabla17: HU Obtener estadísticas de los reportes gestionados en el módulo.

Historia de usuarios	
Número: 3	Usuario: Administrador
Nombre: Obtener estadísticas de los reportes.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Yadriel Vargas Padilla.	
Descripción: se crea un reporte con la información precisa de los reportes que se han gestionados en el sistema.	
Observaciones: Confirmado por el administrador. (como usuario)	

Tabla18: HU Obtener reportes.

Historia de usuarios	
Número: 4	Usuario: Administrador(Alexis)
Nombre: Obtener reportes.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Yadriel Vargas Padilla.	
Descripción: se obtienen una serie de reportes en las cuales se arrojan información sintetizada del comportamiento del estado en que se encuentran las PCs: <ul style="list-style-type: none"> • Reportes pendientes • Reportes pendientes por piezas. • Reportes terminados. 	
Observaciones: Confirmado por el administrador.	

Tabla19: HU Gestionar usuarios del sistema.

Historia de usuarios	
Número: 6	Usuario: Administrador(Alexis)
Nombre: Gestionar usuarios del sistema.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Yadriel Vargas Padilla.	
Descripción: el administrador crea el perfil del usuario que va interactuar con el sistema.	
Observaciones: Confirmado por el Administrador y usuarios	

Tabla20: HU Elaborar reporte de mantenimiento.

Historia de usuarios	
Número: 6	Usuario: Administrador(Alexis)
Nombre: Elaborar reporte de mantenimiento.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Yadriel Vargas Padilla.	
Descripción: El administrador del sistema va ser el responsable de asignarle el periodo de mantenimiento a los diferentes locales con sus respectivos medios de cómputo.	
Observaciones: Confirmada por los usuarios.	

Tabla21: Gestionar Computadoras.

Historia de usuarios	
Número: 7	Usuario: Administrador(Alexis)
Nombre: Gestionar computadoras	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Yadriel Vargas Padilla.	
Descripción: En esta etapa el administrador es el encargado de proveerle al sistema la información de las computadoras que se insertaran, eliminaran en la red, al igual que dará la cobertura de observar el estado de los medios de cómputos en sus diferentes locales.	
Observaciones: Confirmado por el administrador.	

Tabla22: HU Autenticar usuario.

Historia de usuarios	
Número: 8	Usuario: Usuarios del sistema (admin., usuarios, técnicos).
Nombre: Autenticar usuario.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Yadriel Vargas Padilla.	
Descripción: los usuarios deben insertar de forma correctas los datos que le permitirán penetrar al sistema.	
Observaciones: Se deberán insertar nombre de usuarios y contraseñas que sean válidos en e sistema.	

Tabla23: HU Atender avisos de mantenimiento.

Historia de usuarios	
Número: 9	Usuario: Técnico
Nombre: Atender avisos de reportes de averías.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Yadriel Vargas Padilla.	
Descripción: El técnico es el encargado de atender estos tipos de reportes para darle atención inmediata.	
Observaciones: Se suministraran los datos precisos de la avería, así como el medio informático, y la ubicación físico de dicho medio de cómputo.	

Tabla24: HU Asignar avisos.

Historia de usuarios	
Número: 10	Usuario: Técnicos
Nombre: Asignar avisos.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Yadriel Vargas Padilla.	
Descripción: El jefe de los técnicos es el encargado de hacer la distribución de los avisos ya sean de reportes de averías así como las de mantenimientos	
Observaciones: Confirmado por los técnicos.	

Tabla25: HU Exportar los datos a formato de impresión.

Historia de usuarios	
Número: 11	Usuario: Administrador(Alexis)
Nombre: Exportar los datos arrojados por el sistema a formatos de impresión digitales.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Yadriel Vargas Padilla.	
Descripción: A partir de todos los reportes arrojados por el sistema se pueden obtener su homologos pero en formato de impresión. (Pdf, Word, Excel, etc...)	
Observaciones: Confirmado por el administrador.	

Tabla26: Gestionar dispositivos.

Historia de usuarios	
Número: 12	Usuario: Técnico(Copextel)
Nombre: Gestionar dispositivos.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Yadriel Vargas Padilla.	
Descripción: El técnico es el encargado de proveerle los dispositivos al sistema así como la cantidad con los que se cuenta para la posterior asignación, reparación o mantenimiento de los dispositivos que lo necesiten.	
Observaciones: Confirmado por el administrador.	

Anexo-2 Tarjetas CRC

Tarjetas C.R.C que están involucradas en el sistema que se desea automatizar.

Tarjeta C.R.C Módulo #1: Gestionar Elementos.

Tabla27: Tarjeta C.R.C Gestionar_User.

Clases: Gestionar_User	
Responsabilidades	Colaboraciones
Insert_User. Eliminar_User. Modificar_User. Change_Contrasenna. Ver_Users_Sistema.	Conexxion.

Tabla28: Tarjeta C.R.C Gestionar_PCs.

Clases: Gestionar_PCs.	
Responsabilidades	Colaboraciones
Insert_PCs. Modificar_PCs. Dar_alta_PCs. Dar_baja_PCs.	Conexxion.

Tabla29: Tarjeta C.R.C Gestionar_Dispositivos.

Clases: Gestionar_Dispositivos	
Responsabilidades	Colaboraciones
Insert_disp. Eliminar_disp. Modif._disp.	Conexxion.

Tarjeta C.R.C Módulo #2: Obtener reportes.

Tabla30: Tarjeta C.R.C Report_Averia.

Clases: Report_Averia	
Responsabilidades	Colaboraciones
Reportar_New_Averia. Atender_averia	Conexxion.

Tabla31: Tarjeta C.R.C Buscar_Report.

Clases: Buscar_Report	
Responsabilidades	Colaboraciones
Buscar_reporte_invt. Buscar_reporte_user.	Conexxion.

Tabla32: Tarjeta C.R.C Estadisticas_Report.

Clases: Estadisticas_Report	
Responsabilidades	Colaboraciones
Mostar_estadisticas_reporte.	Conexxion.

Tabla33: Tarjeta C.R.C Mostrar Reportes.

Clases: Mostrar Reportes	
Responsabilidades	Colaboraciones
Reporte_pendiente Reporte_terminado Reporte_pendiente_pieza	Conexxion.

Tarjeta C.R.C Módulo #3: Exportar.

Tabla 34: Tarjeta C.R.C Imprimir.

Clases: Imprimir	
Responsabilidades	Colaboraciones
Listado_averias Listado_Mtto. Listado_PCs_facultades.	Conexxion.

Tarjeta C.R.C Módulo #4: Plan de mantenimiento.

Tabla35: Tarjeta C.R.C Clases: Plan_Mtto.

Clases: Plan_Mtto.	
Responsabilidades	Colaboraciones
Elaborar_Plan_Mtto. Atender_Mtto	Conexxion.

Tabla36: Tarjeta C.R.C.

Clases: Asignacion_Mtto	
Responsabilidades	Colaboraciones
Asignar_Mtto_Tecnico.	Conexxion.

Anexo-3 Tareas de Ingeniería

Tareas de Ingenierías o tareas de programación en las que se dividen las HU.

Tareas de la Historia de Usuario #1

Tabla37: Tarea de programación Procesar información de avería.

Tarea de Programación	
Número tarea: 2	Número historia: 1
Nombre tarea: Procesar información de avería.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 12/04/2011	Fecha fin: 13/04/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Aquí el usuario especifica algunas exigencias que se necesitan para generar un nuevo reporte de avería.	

Tareas de la Historia de Usuario #2

Tabla38: Tarea de programación Búsqueda de reporte específico.

Tarea de Programación	
Número tarea: 3	Número historia: 2
Nombre tarea: Búsqueda de reporte específico.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 18/04/2011	Fecha fin: 20/04/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: El administrador del sistema realiza una búsqueda de un reporte determinado según una serie de parámetros pertinentes.	

Tareas de la Historia de Usuario #3

Tabla39: Tarea de programación obtención de estadísticas.

Tarea de Programación	
Número tarea: 4	Número historia: 3
Nombre tarea: obtención de estadísticas.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 25/04/2011	Fecha fin: 27/04/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: El administrador obtiene un resumen de todo el equipamiento que ha sido reportado así como a área a que pertenecen.	

Tareas de la Historia de Usuario #4

Tabla40: Tarea de programación Reporte pendiente.

Tarea de Programación	
Número tarea: 5	Número historia: 4
Nombre tarea: Reporte pendiente.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 02/05/2011	Fecha fin: 03/05/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Se obtienen los reportes que están pendientes por atender.	

Tabla41: Tarea de programación Reporte pendiente por piezas.

Tarea de Programación	
Número tarea: 6	Número historia: 4
Nombre tarea: Reporte pendiente por piezas.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 04/05/2011	Fecha fin: 04/05/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Se obtienen los equipos que están falta de pieza.	

Tabla42: Tarea de programación Reporte terminado.

Tarea de Programación	
Número tarea: 7	Número historia: 4
Nombre tarea: Reporte terminado.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 06/05/2011	Fecha fin: 06/05/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Se obtienen los reportes que han sido atendidos satisfactoriamente.	

Tareas de la Historia de Usuario #5

Tabla43: Tarea de programación Insertar usuarios.

Tarea de Programación	
Número tarea: 8	Número historia: 5
Nombre tarea: Insertar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 02/04/2011	Fecha fin: 02/04/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Se inserta los usuarios que van a interactuar con el sistema.	

Tabla44: Tarea de programación Eliminar usuarios.

Tarea de Programación	
Número tarea: 9	Número historia: 5
Nombre tarea: Eliminar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 03/04/2011	Fecha fin: 04/04/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Elimina usuarios registrados en el sistema.	

Tabla45: Tarea de programación Modifica usuarios.

Tarea de Programación	
Número tarea: 10	Número historia: 5
Nombre tarea: Modifica usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 04/04/2011	Fecha fin: 05/04/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Modifica algunos elementos de los usuarios.	

Tabla46: Tarea de programación Cambiar contraseña.

Tarea de Programación	
Número tarea: 11	Número historia: 5
Nombre tarea: Cambiar contraseña.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 25/03/2011	Fecha fin: 25/03/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: El administrador podrá cambiar la contraseña de los usuarios.	

Tabla47: Tarea de programación Ver usuarios.

Tarea de Programación	
Número tarea: 12	Número historia: 5
Nombre tarea: Ver usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 05/04/2011	Fecha fin: 05/04/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: El administrador verá toda la información de los usuarios.	

Tareas de la Historia de Usuario #6

Tabla48: Tarea de programación Plan de mantenimiento.

Tarea de Programación	
Número tarea: 13	Número historia: 6
Nombre tarea: Plan de mantenimiento.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 20/05/2011	Fecha fin: 23/05/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: El administrador del sistema planificará el mantenimiento de los equipos informáticos.	

Tareas de la Historia de Usuario #7

Tabla49: Tarea de programación Insertar computadoras.

Tarea de Programación	
Número tarea: 14	Número historia: 7
Nombre tarea: Insertar computadoras.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 15/05/2011	Fecha fin: 17/05/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Se inserta nuevas computadoras al sistema.	

Tabla50: Tarea de programación Baja de computadoras.

Tarea de Programación	
Número tarea: 15	Número historia: 7
Nombre tarea: Baja de computadoras.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 18/05/2011	Fecha fin: 18/05/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Se elimina del sistema.	

Tabla51: Tarea de programación Alta computadora.

Tarea de Programación	
Número tarea: 16	Número historia: 7
Nombre tarea: Alta computadora.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 20/05/2011	Fecha fin: 20/05/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Se retira del los reportes terminados y se reasigna a su área.	

Tareas de la Historia de Usuario #8

Tabla52: Tarea de programación Loguear usuario.

Tarea de Programación	
Número tarea: 17	Número historia: 8
Nombre tarea: Loguear usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 15/03/2011	Fecha fin: 15/03/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Para la autenticación de los usuarios.	

Tareas de la Historia de Usuario #9

Tabla 53: Tarea de programación atención de avisos averías.

Tarea de Programación	
Número tarea: 19	Número historia: 9
Nombre tarea: atención de avisos averías.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 03/06/2011	Fecha fin: 05/06/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Esta tarea depende de los reportes que sean provistos por los usuarios.	

Tareas de la Historia de Usuario #10

Tabla54: Tarea de programación atención mtto.

Tarea de Programación	
Número tarea: 20	Número historia: 10
Nombre tarea: Atención Mtto.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 06/06/2011	Fecha fin: 08/06/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Esta tarea depende de la planificación de mantenimiento que realice el administrador.	

Tareas de la Historia de Usuario #11

Tabla55: Tarea de programación Exportar formato de impresión.

Tarea de Programación	
Número tarea: 21	Número historia: 11
Nombre tarea: Exportar formato de impresión.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 10/06/2011	Fecha fin: 12/06/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Los datos de reportes que se obtengan serán exportados a un formato de impresión..	

Tareas de la Historia de Usuario #12

Tabla 56: Tarea de programación Insertar dispositivo.

Tarea de Programación	
Número tarea: 22	Número historia: 8
Nombre tarea: Insertar dispositivo.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 12/06/2011	Fecha fin: 13/06/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Se insertaran nuevos dispositivos al sistema.	

Tabla 57: Tarea de programación eliminar dispositivo.

Tarea de Programación	
Número tarea: 23	Número historia: 8
Nombre tarea: eliminar dispositivo.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 13/06/2011	Fecha fin: 13/06/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Se eliminaran los dispositivos que sean necesarios.	

Tabla 58: Tarea de programación modificar dispositivo.

Tarea de Programación	
Número tarea: 23	Número historia: 8
Nombre tarea: modificar dispositivo.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 14/06/2011	Fecha fin: 15/06/2011
Programador responsable: Yadriel Vargas Padilla.	
Descripción: Se eliminaran los dispositivos que sean necesarios.	

Anexo-4 Pruebas de Aceptación

Pruebas de Aceptación que se les realizan al sistema para comprobar el grado de veracidad de las respuestas que arroja el sistema. (Cont.)

Tabla59: Prueba de aceptación.

Prueba de aceptación
HU: Buscar reportes
Nombre: Búsqueda de reportes específicos.
Descripción: El administrador del sistema busca un reporte específico dado el nombre de usuario o el número de inventario de un equipo.
Condiciones de ejecución: Tienen que existir reportes en la BD.
Entrada/Pasos de ejecución: El administrador escoge el nombre de usuario o no. de inv.
Resultado esperado: Se muestran todos los reportes que pertenezca al usuario o al no. inv. seleccionado.
Evaluación de la prueba: Satisfactoria.

Tabla60: Prueba de aceptación.

Prueba de aceptación
HU: Obtener estadísticas de reportes.
Nombre: Mostrar estadísticas.
Descripción: Se visualiza una estadística completa de los reportes que se han realizado.
Condiciones de ejecución: Existir algún tipo de reporte en la BD.
Entrada/Pasos de ejecución:
Resultado esperado: Visualización de los reportes por áreas y cantidad existentes.
Evaluación de la prueba: Satisfactoria.

Tabla61: Prueba de aceptación

Prueba de aceptación
HU: Obtener reportes.
Nombre: Obtener reportes.
Descripción: Se obtienen reportes pendientes, pendientes por pieza y terminado.
Condiciones de ejecución: Que el técnico tiene que haber atendido al menos un reporte.
Entrada/Pasos de ejecución:
Resultado esperado: Visualización de los reportes.

Evaluación de la prueba: Satisfactoria.

Tabla62: Prueba de aceptación.

Prueba de aceptación
HU: Gestionar usuarios del sistema.
Nombre: Gestionar usuarios.
Descripción: Se insertan, eliminan y modifican los usuarios que interactúan con la aplicación.
Condiciones de ejecución: El usuario a insertar no debe existir en la BD.
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> ➤ Datos para insertar. ➤ Datos para modificar. ➤ Datos para eliminar.
Resultado esperado: La gestión del usuario se efectúa exitosamente.
Evaluación de la prueba: Satisfactoria.

Tabla63: Prueba de aceptación.

Prueba de aceptación
HU: Elaborar plan de mantenimiento.
Nombre: Elaborar plan de mantenimiento.
Descripción: El administrador debe planificar la fecha que se le darán mantenimientos a las computadoras dadas las áreas a que pertenezcan.
Condiciones de ejecución:
Entrada/Pasos de ejecución: asignarle una fecha de mantenimiento a cada computadora.
Resultado esperado: visualizar el plan de mantenimiento para los técnicos.
Evaluación de la prueba: Satisfactoria.

Tabla64: Prueba de aceptación.

Prueba de aceptación
HU: Gestionar computadoras.
Nombre: Gestionar computadoras.
Descripción: El administrador inserta, elimina y modifica computadoras.
Condiciones de ejecución:
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> ➤ Insertar computadora. ➤ Eliminar computadora. ➤ Modificar computadoras. ➤ Visualizar computadoras.
Resultado esperado: Se gestiona una nueva computadora para un área determinada.
Evaluación de la prueba: Satisfactoria.

Tabla65: Prueba de aceptación.

Prueba de aceptación
HU: Autenticar usuarios.
Nombre: Registrar usuario.
Descripción: Un usuario del sistema cuando carga la interfaz de usuario de entrada al sistema tiene que suministrarle los datos correctos para poder acceder al mismo.
Condiciones de ejecución: el usuario debe existir lógicamente en la BD.
Entrada/Pasos de ejecución: El nombre y contraseña de usuario correcta.
Resultado esperado: El usuario del sistema accede al mismo.
Evaluación de la prueba: Satisfactoria.

Tabla66: Prueba de aceptación.

Prueba de aceptación
HU: Atender avisos de reportes de averías.
Nombre: Atender averías.
Descripción: El técnico responsable atiende la solicitud de un reporte de avería tratando de dar solución a la queja.
Condiciones de ejecución: Que existan quejas de reportes.
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> ➤ Seleccionar reportes. ➤ Asignarle pieza si existe. ➤ Reportar como atendido.
Resultado esperado: Se atienda con éxito la queja del reporte seleccionado.
Evaluación de la prueba: Satisfactoria.

Tabla67: Prueba de aceptación.

Prueba de aceptación
HU: Atender avisos de mantenimiento.
Nombre: Atender mantenimiento.
Descripción: El técnico atiende los reportes de quejas de mantenimiento que existan en la BD.
Condiciones de ejecución: Deben existir reportes de quejas de mantenimientos en la BD.
Entrada/Pasos de ejecución: Pasos para probar la funcionalidad.
Resultado esperado: <ul style="list-style-type: none"> ➤ Seleccionar las quejas que se les dará mantenimiento. ➤ Reportar como atendido.
Evaluación de la prueba: Satisfactoria.

Tabla68: Prueba de aceptación.

Prueba de aceptación
HU: Exportar a formato de digital.
Nombre: Formato de impresión.
Descripción: El administrador del sistema visualiza cada reporte que arroja el sistema en un formato digital disponible para impresión.
Condiciones de ejecución: que existen reportes atendidos o reportes de quejas en la BD.
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> ➤ Visualizar reportes ➤ Exportar a formato digital.
Resultado esperado: Un documento de formato pdf o word con la información de los reportes, disponible para impresión.
Evaluación de la prueba: Satisfactoria.

Tabla69: Prueba de aceptación.

Prueba de aceptación
HU: Gestionar dispositivos.
Nombre: Gestionar dispositivos.
Descripción: El técnico debe suministrarle al sistema los dispositivos que se adquieran para darle solución a las quejas de los reportes.
Condiciones de ejecución:
Entrada/Pasos de ejecución: <ul style="list-style-type: none"> ➤ Insertar dispositivos. ➤ Modificar dispositivos. ➤ Eliminar dispositivos. ➤ Ver dispositivos.
Resultado esperado: Gestionar exitosamente los dispositivos.
Evaluación de la prueba: Satisfactoria.

Anexo-5 Modelo Físico de Datos

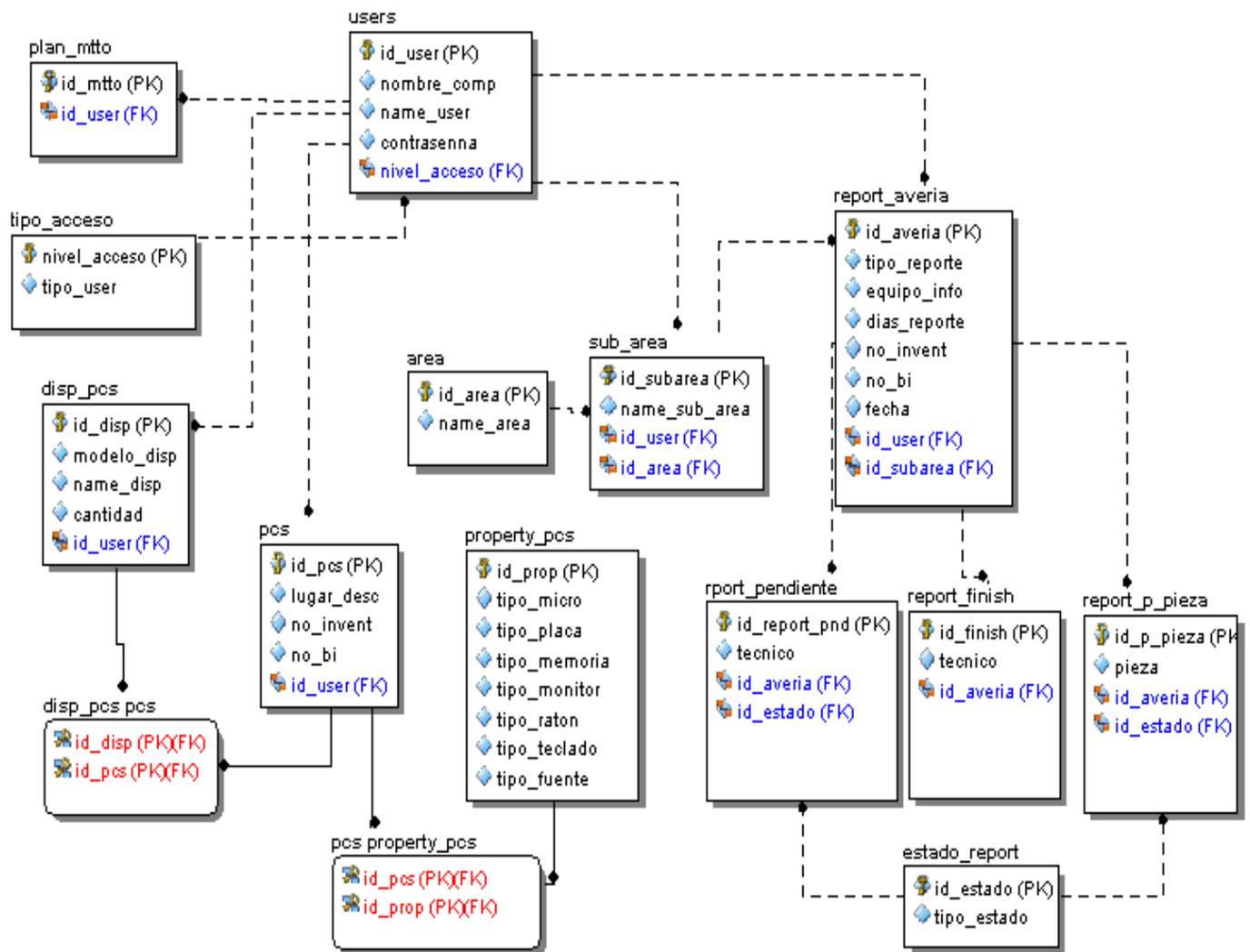


Figura 8. Modelo físico de datos.