



**Ministerio de Educación Superior
Instituto Superior Minero Metalúrgico de Moa
Dr. Antonio Núñez Jiménez**

Trabajo de Diploma

En opción al Título de Ingeniería Informática

Sistema Automatizado para la Gestión de Información de la Maestría de Electromecánica

Autor: Daimara Arias Ramírez.

Tutores: Ing. Eulises Peña Duran.

Dr.C Arístides Alejandro Legrá Lobaina.

**Moa,
Enero ,2011
“Año 53 de la Revolución.**

DECLARACIÓN DE AUTORIA

Declaro que soy el único autor de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes _____ del año 2011.

Firma del autor

Firma del tutor

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título:

Autor:

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

- Alto nivel de Independencia
- Adecuada Originalidad
- Creatividad
- Alto nivel de Laboriosidad
- Alto nivel de Responsabilidad

Además, debe destacarse la buena calidad científico-técnica de sus resultados y la adecuada calidad del documento de la tesis.

El software desarrollado por la autora es de inmediata aplicación y su importancia está argumentada por su papel en la Gestión de Postgrado del CEETAM. El sistema informático que resulta de la presente investigación constituye un primer paso para la automatización de la Maestría de Electromecánica en el ISMMM como parte del sistema informático que desarrolla el CEETAM.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de 5-Excelente.

Nombre completo del primer tutor
<Grado científico, Categoría docente
y/o investigativa>
(Si procede)

Nombre completo del segundo tutor
<Grado científico, Categoría docente
y/o investigativa>

Fecha: _____

Pensamiento

“Milito en el grupo de los impacientes y milito en el bando de los apurados de los que siempre presionan para que las cosas se hagan, y de los que muchas veces tratan de hacer más de lo que se puede”.

Fidel Castro
1963.

Agradecimientos.

Quisiera agradecerles en primer lugar a mis padres por siempre estar ahí firmes todos estos años de mi carrera... a mis abuelos por hacerme la convicción de que en este mundo siempre se es necesario de un profesional.

A mi querido y amado esposo Leonel en primer lugar por estar a mi lado en estos años, por brindarme su apoyo incondicional en la realización de este trabajo que sin él no hubiera llegado al final, por haber tenido la paciencia necesaria en estos meses de nervios y de noches sin dormir, agradecerles a mis tutores Arístides Alejandro Legra Lobaina por sus ideas interesantes y su preocupación durante todo el desarrollo del trabajo.

No se pueden pasar por alto a dos queridas personas Gaby y Lili que han sido más que una familia en todos estos años, porque estuvieron en momentos difíciles y buenos en los que me hacían falta tener una familia a mi lado. Los que me brindaron sus manos para construir todo lo que quería ser, una Ingeniera.

Quisiera agradecerle también a mi querido profesor Roilber ya que en los días más difíciles me supo brindar de su conocimiento y apoyo incondicional.

Otros que no pueden faltar son Roiky y Yusmey que en los momentos finales estuvieron presentes y me aportaron mucho conocimiento para realizar una discusión exitosa.

No podía faltar el agradecimiento a mis compañeros de aula que vienen conmigo desde segundo año y me dieron su apoyo en especial a Yohandi, Odelmis, Padilla, Ernesto Leyva, y Alexander.

A todos: Muchas Gracias.

Dedicatoria

A mis queridos padres ,abuelos, y en especial a mi amado Papi Eliecer ,por haberme guiado hasta hacer realidad mi sueño que es el de ellos también, y por enseñarme que la vida es de mucho sacrificio y prueba.

Resumen

El Centro de Estudio y Energía de Tecnología de Avanzada de Moa (CEETAM), se desenvuelve en un entorno de niveles considerables de información.

Específicamente en la Maestría de Electromecánica del CEETAM, la información es manejada en un sistema de información que aún no está informatizado. La gestión de la información se ha convertido en un tema de amplia significación, lo que propicia la necesidad de crear un sistema automatizado que permita funcionar eficientemente a la Maestría.

El objetivo de esta investigación es construir una aplicación informática que posibilite la propagación e intercambio de información de la Maestría de Electromecánica, garantizando la gestión de la información de manera eficaz y segura.

El uso de esta aplicación le facilitará el trabajo al Coordinador, Comité Académico y al Claustro de la Maestría. Además, permitirá la creación y refinamiento de historiales, sin los errores que puedan cometerse si eso se realiza manualmente, contando con funcionalidades para la toma de decisiones.

A partir de los estudios realizados sobre las herramientas para la construcción de sistemas de gestión de información se propone la elaboración de una aplicación web. En este documento se recoge un resumen del estudio realizado, se presentan los artefactos obtenidos durante cada una de las etapas de la metodología de desarrollo de software seleccionada.

Abstract

The Center for Energy Studies and Advanced Technology Moa (CEETAM), operates in an environment of significant levels of information.

Specifically in the Master of Electromechanical the CEETAM, information is managed in an information system that is not yet computerized. The information management has become a topic of broad significance, which promotes the need to create an automated system that allows it to operate efficiently at the Masters.

The objective of this research is to build a software application that facilitates the spread and exchange of information on the Master of Electromechanical, ensuring management information efficiently and securely.

Using this application will facilitate the work of the Coordinating Committee and the Academic Senate of the Master. It will also enable the creation and refinement of records, without the errors that may be committed if it is done manually, with features for decision making.

From studies of tools for building information management systems is proposed to develop a web application. This document provides a summary of the study, are the artifacts obtained during each stage of software development methodology selected.

Índice

Introducción.....	1
Capítulo I- Fundamentación Teórica	5
1- Introducción al Capítulo.....	5
1.2 Estado del Arte	5
1.3 Flujo Actual de los Procesos.....	6
1.3.1 Análisis crítico de la ejecución de los procesos.	6
1.3.2 Sistemas automatizados vinculados al campo de acción.....	7
1.4 Tendencias y Tecnologías Actuales.	7
1.4.1 Lenguajes de programación.....	8
1.4.2 Sistemas Gestores de Base de Datos (SGBD)	10
1.4.2.1 PostgreSQL.....	10
1.4.2.2 MySQL	11
1.4.3 Servidor para aplicaciones Web.....	11
1.4.3.1 Apache	11
1.4.3.2 XAMPP.....	13
1.4.3.3 Macromedia Dreamweaver (versión 8).....	13
1.5 Herramientas CASE.....	14
1.6 Metodologías para el Desarrollo de Software	15
1.6.1 Metodología propuesta para el desarrollo de la aplicación	16
1.6.2 ¿Por qué elegir XP?	16
1.6.3 Valores que promueve XP	17
1.6.4 XP en detalle.....	18
1.6.5 ¿Qué propone XP?	26
1.6.6 Fases de la metodología XP	26
1.7 Arquitectura Cliente/Servidor	28
1.7.1 Características de la Arquitectura Cliente/Servidor.....	29
1.8 Herramientas para el desarrollo de la aplicación	33
1.8.1 PHP Como Lenguaje de Programación.	33
1.8.2 MySQL Como Gestor de Base de Datos.....	33
1.8.3 XAMPP.....	34
1.8.4 MYSQL Workbench 7 OSS	34
CAPÍTULO II. PLANIFICACIÓN Y DISEÑO	36
2.1 Introducción	36
2.2 Funcionalidades generales	36
2.2.1 Personal relacionado con el sistema.....	36
2.2.2- Lista de reserva.....	37
2.2.3 Historias de usuario.....	37
2.3-Planificación de Entregas.	39
2.3.1 Estimación de esfuerzo por historia de usuario.....	39
2.3.2 Planificación de iteraciones.....	40
2.3.3 Plan de duración de las iteraciones.....	41
2.4 Tarjetas CRC	41
CAPITULO III - DESARROLLO Y PRUEBAS	43
3.1 Introducción	43
3.2 Modelo de datos.	43
3.3 Desarrollo de las iteraciones.....	43
3.3.1 Tareas por historias de usuario	43
3.3.2 Iteración No.1	44
3.4 Pruebas	45
3.4.1 Desarrollo dirigido por pruebas	45
3.4.2 Pruebas de aceptación.....	46

CAPITULO IV - ESTUDIO DE FACTIBILIDAD.....	48
4.1 Introducción	48
4.2 Efectos económicos.....	48
4.3 Beneficios y Costos Intangibles en el proyecto.....	50
4.4 Ficha de costo	50
4.4.1 Costos en Moneda Libremente Convertible:	51
4.4.2 Costos en Moneda Nacional:	52
CONCLUSIONES GENERALES.....	54
RECOMENDACIONES	55
Fuentes Bibliográficas	56
GLOSARIO DE TÉRMINOS.....	58
Anexos	59
Anexo 1 Historias de usuarios.....	59
Anexo 2 Tarjetas CRC	65
Anexo 3 Tarjetas de Ingeniería	68
Anexo 4 Pruebas de Aceptación	77
Anexo 5 Modelo de la Base de Datos	82
Anexo 6 Principal Interfaz de la Aplicación.....	83



Introducción

Las tecnologías de la información y las comunicaciones (Tics) con su creciente desarrollo han alcanzado un lugar cimero en el mundo, enmarcado principalmente por las crecientes necesidades que logran satisfacer cada día.

Con la era digital y el auge alcanzado por el mercado de productos informáticos a nivel mundial, así como la dependencia creada por las diferentes ramas de la sociedad hacia esos productos y servicios, se ha hecho necesaria la creación de sistemas que mejoren y faciliten la gestión de la información en los procesos productivos y de servicios, asimismo, como estrategia de la informatización de la sociedad cubana, se hace necesaria una mayor presencia de la universidad en la red de redes.

En el ISMM ambos objetivos han sido considerados en una Estrategia de Informatización que tenga en cuenta todos los procesos sustantivos. Específicamente, en la actividad del Postgrado Académico que desarrolla el Centro de Estudio de Energía y Tecnología Avanzada de Moa (CEETAM) de la Facultad de Metalurgia y Electromecánica (FMEM) del ISMM, se ha propuesto que la propagación e intercambio de información de sus Maestrías y Doctorados se realicen desde la Web. Entonces queda planteada como situación problémica general la necesidad de diseñar e implementar los entornos de informatización de las Maestrías y Doctorados que desarrolla el CEETAM. Específicamente, en nuestro caso se plantea la necesidad de implementar una aplicación informática para el entorno de la Maestría de Electromecánica del ISMMM.

Dicha aplicación facilitará el intercambio con todas las personas que estén interesadas en obtener información académica, que contendría:

- La organización y planificación de las ediciones y asignaturas.
- La divulgación del programa de la maestría.
- La información relacionada con la solicitud y aprobación de la matrícula.

Con este trabajo se pretende dar solución a la situación anteriormente expuesta, definiéndose como **problema científico**: La no existencia de una aplicación informática en la Maestría de Electromecánica del CEETAM, que posibilite la estrategia de informatización del ISMMM y la propagación e intercambio de información. Teniendo en



cuenta el problema planteado el **objeto de estudio** de este trabajo es la construcción de Sistemas Automatizados para la Gestión de Información, el **campo de acción** se enmarca en la Automatización de la Gestión de Información de la Maestría de Electromecánica del CEETAM.

Para darle solución al problema se plantea el siguiente **objetivo general** del trabajo:

Construir una aplicación informática que posibilite la propagación e intercambio de información de la Maestría de Electromecánica del CEETAM.

Como **idea a defender** se plantea que si se desarrolla un entorno de informatización basado en una aplicación WEB, entonces se garantizan la propagación e intercambio de información de la Maestría de Electromecánica del CEETAM.

Para darle paso al objetivo general fue necesario establecer los siguientes **objetivos específicos**:

- A. Establecer el Marco Teórico Conceptual.
- B. Caracterizar el intercambio presente y futuro de información de la Maestría de Electromecánica del CEETAM.
- C. Proponer el diseño de un entorno informático basado en una aplicación WEB que garantice el manejo eficaz y eficiente de la información de la Maestría de Electromecánica del CEETAM.
- D. Desarrollar una aplicación WEB con una calidad tal que garantice una adecuada funcionalidad y visibilidad.

Para cumplir con estos objetivos y resolver la situación problemática planteada, se proponen las siguientes **tareas a desarrollar**:

- A.1. Establecer el sistema de conceptos inherentes a la investigación.
- A.2. Búsqueda de información de investigaciones similares.
- A.3. Escribir el Capítulo 1.
- B.1. Entrevistar a expertos que posean grandes experiencias sobre el tema.
- B.2. Conocer el sistema de información relacionado con el contenido de la maestría.
- B.3. Conocer el sistema administrativo de la maestría.



- C.1 Diseñar una aplicación WEB que garantice el manejo eficaz y eficiente de la información de la Maestría de Electromecánica del CEETAM.
- D.1 Implementar el diseño de la aplicación WEB con calidad que garantice una adecuada funcionalidad y visibilidad.

Para llevar a cabo las tareas antes mencionadas se emplearon métodos teóricos y empíricos de la investigación científica. Entre los métodos teóricos utilizados están:

Análisis y síntesis: Este método se utiliza para desglosar el problema en partes o sub-problemas para de esta forma comprobar el correcto funcionamiento de las mismas, luego integrarlo todo para corroborar las relaciones entre estas y su integración como un todo, llegando así a una mejor solución, también para arribar conclusiones parciales y generales de la investigación.

Histórico-Lógico: Es utilizado para la comprensión de la evolución del empleo de herramientas para la gestión de la información que es generada por la interacción de dos o más personas.

Entre los métodos empíricos utilizados se encuentran:

Entrevista: Se usa con el objetivo de recopilar información, esta será la vía fundamental para la determinación de los requerimientos del sistema.

La **observación:** es útil para entender el comportamiento del sistema y sus especificaciones.

Revisión de documentos: se utiliza para conocer los detalles del funcionamiento del banco es decir, reglas y particularidades de esta entidad además de permitirnos justificar nuestra solución al problema planteado.

El presente trabajo consta de cuatro capítulos:

Capítulo 1 “Fundamentación Teórica”: en este capítulo se ofrece una breve descripción del objeto de estudio, objetivo general, tareas y algunos conceptos fundamentales. Se realiza un estudio acerca de los diferentes sistemas existentes vinculados al campo de acción, además se presenta la metodología y las herramientas a utilizar en el desarrollo del sistema propuesto.

Capítulo 2 “Planeación y Diseño”: se hace uso de la metodología expuesta en el capítulo inicial para el desarrollo del proyecto, abordando en detalles cada uno de sus fases.



Capítulo 3 “Desarrollo y Pruebas”: se presentan los principales métodos y definiciones dentro de la implementación de los flujos de trabajo. También se muestran las interfaces gráficas diseñadas para la interacción de los flujos de trabajo con los usuarios. Se describen además las pruebas realizadas y sus resultados.

Capítulo 4 “ Estudio de Factibilidad”: se realiza un estudio de los esfuerzos requeridos para la construcción del sistema y se valora la sostenibilidad del producto.



Capítulo I- Fundamentación Teórica

1- Introducción al Capítulo

En la actualidad, la informática ha evolucionado considerablemente y junto a ella la producción de diversos software. En este capítulo abordaremos los conceptos fundamentales para el desarrollo de aplicaciones Web, así como también ilustraremos un estudio de las ventajas y desventajas que estas ofrecen, las características generales de los sistemas de bases de datos y lenguajes de programación Web, además se verán las diferentes metodologías existentes para el desarrollo de aplicaciones, y de ellas las adecuadas a utilizar en este trabajo depende de las ventajas que estas ofrecen. Finalmente, se hace una descripción de las herramientas y tecnologías utilizada para el desarrollo del software.

1.2 Estado del Arte

El objeto de estudio del presente trabajo es la construcción de sistemas automatizados para la gestión de información y su campo de acción se ha definido como la automatización de la gestión de información de la Maestría de Electromecánica del CEETAM. Es conveniente, por tanto, conocer los antecedentes de trabajos similares en el extranjero, en Cuba y específicamente en el ISMMM.

La mayoría de los postgrados académicos internacionales se presentan sobre plataformas informáticas que llenan las expectativas que hemos esbozado al principio de la introducción. Se trata de sistemas sobre WEB que permiten una efectiva comunicación entre profesores, profesores y estudiantes y entre estudiantes.

Algunos de estos sistemas han sido diseñados específicamente para ciertas maestrías o doctorados pero otros constituyen bases genéricas donde se pueden “montar” diferentes cursos de postgrado, maestrías, etc.

En Cuba, el desarrollo de estos sistemas es escaso y principalmente se ha trabajado la presentación de postgrados en la Plataforma MOODLE. En el ISMMM no se tienen experiencias de este tipo.



1.3 Flujo Actual de los Procesos

Desde la perspectiva de las personas interesadas en realizar y obtener información de la maestría, podemos decir que saben de la existencia de la maestría por alguna persona que lo recomendara. Para poder solicitar matrícula en dicha maestría es necesario cumplir con una serie de requisitos. Para realizar esta operación es necesario que las personas manden por correo toda la información o se presenten personalmente. Si estas personas fueran aprobadas en la matrícula de la maestría tienen que presentarse físicamente para recibir sus respectivas clases.

Desde la perspectiva del coordinador de la maestría, podemos decir que este recibe todas las solicitudes de matrícula de la maestría por correo o por las personas interesadas presentándose físicamente. Este lleva manualmente mucho contenido académico acerca de la maestría. También toda la información de sus alumnos y el claustro de la maestría.

1.3.1 Análisis crítico de la ejecución de los procesos.

Desde la perspectiva de las personas interesadas en realizar y obtener información de la maestría, podemos decir que en estos momentos existe una página Web en el CEETAM que tiene un vínculo hasta la página Web de la Maestría Electromecánica.

En el menú de esta página esta la opción de matricularse, donde las personas que están interesadas pueden solicitar matrícula en la maestría, este proceso anteriormente era muy tedioso y a la vez difícil para algunas personas que no vivieran en nuestro país. Todos los requisitos necesarios para matricularse pueden subirse al sitio y esperar la respuesta del coordinador para saber si fue aprobado o no. Se quiere que las clases de los maestrantes sean recibidas por el sitio para comodidad de los mismos ya que estos no tendrían la necesidad de viajes desde su país.

Desde la perspectiva del coordinador de la maestría, podemos decir que la información académica que anteriormente se llevaba manualmente ahora se llevara por el sitio, al igual que la información de los maestrantes y el claustro de la maestría.



1.3.2 Sistemas automatizados vinculados al campo de acción.

Moodle es un sistema de gestión de cursos de libre distribución (course management system, CMS) que ayuda a los educadores a crear comunidades de aprendizaje en línea. Promueve una pedagogía constructivista social (colaboración, actividades, reflexión crítica, etc.). Tiene una interfaz de navegador de tecnología sencilla, ligera, y compatible. (Dougiamas, y otros, 2008)

Moodle Promueve una pedagogía constructivista social (colaboración, actividades, reflexión crítica, etc.). Su arquitectura y herramientas son apropiadas para clases en línea, así como también para complementar el aprendizaje presencial. Tiene una interfaz de navegador de tecnología sencilla, ligera, y compatible. La instalación es sencilla requiriendo una plataforma que soporte PHP y la disponibilidad de una base de datos. (Dougiamas, y otros, 2008)

Moodle tiene una capa de abstracción de bases de datos por lo que soporta los principales sistemas gestores de bases de datos. Se ha puesto énfasis en una seguridad sólida en toda la plataforma. Todos los formularios son revisados, las cookies cifradas, etc. La mayoría de las áreas de introducción de texto (materiales, mensajes de los foros, entradas de los diarios, etc.) pueden ser editadas usando el editor HTML, tan sencillo como cualquier editor de texto. (Dougiamas, y otros, 2008)

En la presente investigación se ha propuesto el desarrollo de una nueva plataforma debido a que MOODLE no permite estructurar de forma completa el sistema de trabajo de las maestrías en Cuba. Además, se trata de un software australiano al cual no se tiene acceso a su código fuente lo cual es contraproducente para su desarrollo local. Finalmente ese software está sujeto a que en algún momento sea presentado en términos comerciales imposibles de cumplir. (Dougiamas, y otros, 2008)

1.4 Tendencias y Tecnologías Actuales.

La humanidad con el desarrollo de las tecnologías lucha por hacer más rápido y seguro el manejo de las comunicaciones y el control de datos. Muchas tecnologías han revolucionado el mundo, en el campo de la informática, la tecnología Cliente-Servidor toma fuerza progresivamente debido a sus ventajas. La Web deja de ser un medio de



divulgación de información para convertirse en una forma operativa, eficaz y accesible, desde cualquier parte para el control de datos.

1.4.1 Lenguajes de programación

Personal Home Page - (PHP) es el acrónimo de procesador hipertexto (Hipertexto Preprocessor). Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado y embebido en el HTML. Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. (Basulto, 2010)

PHP es un lenguaje de programación de estilo clásico, esto significa que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, etc. No es un lenguaje de marcas como podría ser HTML, XML o WML. (Basulto, 2010)

A diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso permite acceder a los recursos que tenga el servidor, como por ejemplo podría ser, una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML. (Basulto, 2010)

PHP es la gran tendencia en el mundo de Internet. Últimamente se puede observar un ascenso imparable, puesto que cada día son muchas las páginas Web que lo utilizan para su funcionamiento, según las estadísticas, PHP se utiliza en más de 10 millones de páginas, y cada mes realiza un aumento del 15%. Como síntesis, PHP corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unas 20 bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de aplicaciones Web muy robustas, y contiene unas 40 extensiones estables sin contar las que se están experimentando, igualmente tiene soporte para comunicarse con otros servicios usando protocolos tales



como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros, además de que: (Basulto, 2010)

- Es software libre, lo que implica menos costes y servidores más baratos que otras alternativas.
- Es muy rápido. Su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.
- Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si está familiarizado con esta sintaxis, resultará un poco mejor aprender PHP.
- Su librería estándar es realmente amplia, lo que permite reducir los llamados “costes ocultos”, uno de los principales defectos de ASP.
- PHP tiene una de las comunidades más grandes en internet, esto permite encontrar fácilmente ayuda, documentación, artículos, noticias y otros recursos.
- Permite las técnicas de Programación Orientada a Objetos (POO).
- Posibilita crear los formularios para la Web.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

HTML – El Lenguaje HTML es el idioma de la Web. Se basa en el uso de “Etiquetas” para la definición del formato del texto, los distintos elementos que conforman la página, sus propiedades y disposición. Este lenguaje es interpretado por los navegadores, procesado y convertido en una Web tal como la vemos en la pantalla, con imágenes, tablas, texto, videos y toda clase de elementos. El lenguaje está compuesto por etiquetas o marcas, gracias a ellos es posible darles forma a todos los componentes de una página o un documento HTML. Las etiquetas de HTML están divididas en etiquetas de apertura y de cierre, aunque no siempre existen estas últimas. (Cuenca Muguercia, 2010)



JavaScript – ES un lenguaje pensado para agregar interactividad con el usuario a las páginas HTML. Permite ejecutar secuencias de comandos en el mismo navegador del usuario. Con JavaScript se puede realizar cálculos rápidos y complejos, verificar formularios antes de enviarlos, crear calendarios, convertir divisas. Es un lenguaje que distingue entre minúscula y mayúscula, no exige la declaración explícita de las variables, es posible crear las variables. Es importante saber que JavaScript no lo soportan todos los navegadores por lo que nos vemos en la situación de probar el código resultante en más de un navegador. La sintaxis es muy parecida a C o C++, por lo que es un lenguaje fácil para el que lo domine. (Cuenca Muguercia, 2010)

1.4.2 Sistemas Gestores de Base de Datos (SGBD)

Un Sistema de Gestión de Bases de Datos (SGBD) es el conjunto de programas que permiten definir, manipular y utilizar la información que contienen las bases de datos, realizar todas las tareas de administración necesarias para mantenerlas operativas, mantener su integridad, confidencialidad y seguridad. Una base de datos nunca se accede ó manipula directamente, sino a través del SGBD. Se puede considerar al SGBD como la interfaz entre el usuario y la base de datos. El funcionamiento del SGBD está muy interrelacionado con el del Sistema Operativo, especialmente con el sistema de comunicaciones. El SGBD utilizará las facilidades del sistema de comunicaciones para recibir las peticiones del usuario (que puede estar utilizando un terminal físicamente remoto) y para devolverle los resultados. Las peticiones se realizan generalmente en forma de sentencias SQL (Structure Query Lenguaje), que no es más que un lenguaje de consultas estructurado compuesto por comandos, cláusulas, operadores y funciones de agregado. (Rodríguez Granada, 2009)

1.4.2.1 PostgreSQL

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS), que ha sido desarrollado de varias formas desde 1977, es una herramienta muy potente para los desarrolladores de sistemas de bases de datos. PostgreSQL



tiene transacciones, integridad referencial, vistas, y multitud de funcionalidades. PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto. PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características, algunas son la alta concurrencia y Amplia variedad de tipos nativos, entre otras. (Rodríguez Granada, 2009)

1.4.2.2 MySQL

MySQL - En el mundo de las base de datos cliente/servidor existe una feroz competencia. Muchos “grandes” compiten por ser la prestación más rápida, más segura, más confiable, más robusta. Los principales colosos de este mundo son, sin dudas Microsoft SQL Server y Oracle, y otros no tan conocidos como DB2, Sybase, Informix y Postgres. Sin embargo, MySQL no se queda atrás y desde hace poco se ha convertido en una importante competencia para estos productos, ya que cuenta con características comparables y muchas veces mejores. La empresa que desarrolla MySQL es MySQL AB, de origen sueco. MySQL es un sistema de gestión de base de datos relacional, multi-hilo y multiusuario, con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. MySQL AB pertenece a Sun Microsystems desde enero de 2008. Por un lado lo ofrece bajo la GNU GPL, pero, entidades que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso. Está desarrollado en su mayor parte en ANSIC. Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL está poseído y patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. (Wikipedia)

1.4.3 Servidor para aplicaciones Web

1.4.3.1 Apache

Apache, sustancialmente, es un proyecto nacido para crear un servidor de Web estable, fiable y veloz para plataformas Unix. Apache nace, por una parte, de un código



ya existente y de una serie de parches (patch) para mejorar su fiabilidad y sus características; de ahí su nombre. (Htmlpoint.com, 2008)

Características: (Ciberaula.com, 2005)

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es una tecnología gratuita con un código fuente disponible. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esta característica le ofrece al software un grado de transparencia tal que es posible determinar en todo momento qué es lo que se está instalando, sin secretos ni puertas traseras.
- Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que se instalen cuando se necesiten. Otra cosa importante es que cualquiera que posea alguna experiencia en la programación de C o Perl puede escribir un módulo para realizar una función determinada. (Ciberaula.com, 2005)
- Trabaja con Perl, PHP y otros lenguajes de script. Perl destaca en el mundo del script y Apache utiliza su parte del pastel de Perl tanto con soporte CGI como con soporte mod perl. También trabaja con Java y páginas JSP. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs³⁰. Permite la creación de ficheros de log a la medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.

Se pueden extender las características de Apache hasta donde la imaginación y los conocimientos lleguen, debido a que el equipo de desarrollo está formado por



voluntarios, diseminados por todo el mundo, que sigue manteniendo este servidor de Web libre. (Ciberaula.com, 2005)

1.4.3.2 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 para Microsoft Windows, GNU/Linux, Solaris, y MacOS X. (BLANCO CRIADO, 2008)

XAMPP es regularmente actualizado para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL, y PhpMyAdmin. Para instalar XAMPP requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar programas por separado.

Oficialmente, los diseñadores de XAMPP solo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios webs y programadores testear su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica sin embargo, XAMPP es utilizado actualmente para servidor de sitios webs en WWW, y con algunas modificaciones es generalmente lo suficientemente seguro para serlo. Una herramienta especial es suministrada para proteger fácilmente las partes más importantes del paquete. (BLANCO CRIADO, 2008)

1.4.3.3 Macromedia Dreamweaver (versión 8)

Macromedia Dreamweaver – Es un editor de texto, un entorno de desarrollo donde el Webmaster puede olvidarse de las partes más tediosas del diseño, como tablas, formularios, y demás elementos.

Es una de las herramientas más utilizadas para el trabajo de aplicaciones visuales, el programa se adapta increíblemente a las necesidades de todo tipo de profesional del diseño Web tanto como para los que deseen programar el código como para los que gustan de una metodología totalmente visual.



Soporta varios lenguajes tales como: PHP, ASP, HTML, JavaScript o CSS. Otra característica interesante del programa es su integración con Flash y Fireworks también productos de Macromedia. (Hernán, 2006)

Permite insertar algunos elementos básicos en Flash sin necesidad de tener este programa instalado, como botones, viñetas y textos. Finalmente si queremos potenciar el programa podemos instalarle gran cantidad de plug-ins, o extensiones, los cuales pueden ser descargados del sitio de Macromedia o bien podemos programarlos nosotros mismos. (Hernán, 2006)

1.5 Herramientas CASE

- ERStudio
- MYSQL Workbench 7 OSS

Embarcadero ER/Studio-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. (Bureaudeprensa.com, 2008)

MYSQL Workbench 5.1 OSS: 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. Se diseña para hacer más fácil de entender el estado actual de los datos. Simple y fácil al usuario, ayuda a organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia. (Basulto, 2010)

Existen otras como es el caso del Embarcadero y Rational Rose, el primero solo para relacionar entidades para la base de datos y el segundo además de ser una herramienta que se basa especialmente en la realización de diagramas. (Basulto, 2010)



1.6 Metodologías para el Desarrollo de Software

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en muchos otros. (FERRER, 2008)

Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. (FERRER, 2008)

El Proceso Unificado de Desarrollo (RUP).

RUP. (Proceso Unificado del Racional)-Es una metodología para proyectos más largos, debido a su gran cantidad de diagramas que lleva consigo, además se documenta poco sobre el Sistema que se está llevando a cabo. (FERRER, 2008)

XP. (Programación Extrema)

Metodología que adopta 12 prácticas que se pueden utilizar todas o no, eso lo deciden el programador y el cliente según las necesidades de este último o si la aplicación no requiere de todas. Se centra especialmente en documentar en forma de plantillas, tiene cuatro fases: Planeación, Diseño, Desarrollo o Implementación y Pruebas. En la primera fase se generan como artefactos los usuarios del negocio, las historias de usuarios, la lista de reserva del producto, el plan de iteraciones, entre otros. En la segunda se tiene el modelo de datos, tarjetas CRC. En tercera fase se desarrollaron las tareas de ingeniería y la cuarta fase son efectuadas las pruebas al software para verificar que el mismo cumpla con todas las funcionalidades acordadas, estas pruebas pueden ser aceptadas por el cliente o denegadas por el mismo. (FERRER, 2008)



SXP. (Scrum + Programación. Extrema)

Esta Metodología fue elaborada en la Universidad de las Ciencias Informáticas, escogieron una parte de Scrum y otra parte de XP, para una mayor organización divide en carpetas todas sus plantillas. (FERRER, 2008)

1.6.1 Metodología propuesta para el desarrollo de la aplicación

La metodología XP o Extreme Programming es una de las variantes de las metodologías ágiles con más aceptación en la comunidad internacional de desarrollo. Esta metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Universidad de Oriente)

1.6.2 ¿Por qué elegir XP?

Actualmente XP es la metodología ágil más documentada (hay una colección de libros "XP Series" de Addison Wesley) y extendido. Existe una gran comunidad de desarrolladores XP. Otra de las ventajas de XP es que no es necesario adoptarlo en forma completa, sino que pueden utilizarse varias de sus prácticas en forma independiente. Esto hace que el costo de su implementación sea mucho más accesible que el de otras metodologías. Un estudio a la bibliografía (Welicki, 2003), muestra las ventajas que tiene XP y que exponemos en los puntos siguientes: (JEFFRIES, 2009)

- Puede ser implementado en forma parcial (elegir sólo algunas de las prácticas)
- Puede ser implementado en forma gradual.
- Puede adaptarse a las necesidades de cualquier equipo de desarrollo. De hecho, Kent Beck recomienda a los equipos que lo adapten a sus necesidades. (JEFFRIES, 2009)



- Exige que se establezca una comunicación más fluida con el cliente y que este tenga mayor participación en el proceso de desarrollo. La consecuencia de esto es que el cliente se involucre más en el desarrollo del producto.
- Actualmente es la metodología ágil más extendida y documentada.
- Se realizan pruebas constantemente del sistema.

1.6.3 Valores que promueve XP

Comunicación

XP se nutre del ancho de banda más grande que se puede obtener cuando existe algún tipo de comunicación como la comunicación directa entre personas. Es muy importante entender cuáles son las ventajas de este medio. Cuando dos o más personas se comunican directamente pueden no sólo consumir las palabras formuladas por la otra persona, sino que también aprecian los gestos, miradas etcétera, que hace su compañero. Sin embargo, en una conversación mediante el correo electrónico, hay muchos factores que hacen de esta una comunicación, por así decirlo, mucho menos efectiva. (Beck, 1999)

Coraje

El coraje es un valor muy importante dentro de la programación extrema. Un miembro de un equipo de desarrollo extremo debe tener el coraje de exponer sus dudas, miedos, experiencias sin “embellecer” éstas de ninguna de las maneras. Esto es muy importante ya que un equipo de desarrollo extremo se basa en la confianza para con sus miembros. Faltar a esta confianza es grave. (Beck, 1999)

Simplicidad

Dado que no se puede predecir como va a ser en el futuro, el software que se desarrolla; por un equipo de programación extrema intenta mantener el mismo lo más sencillo posible. Esto quiere decir que no se va a invertir ningún esfuerzo en hacer un desarrollo que en un futuro pueda llegar a tener valor. En XP, frases como, “...en un futuro vamos a necesitar...” o, “...haz un sistema genérico de...”, no tienen ningún sentido ya que no aportan ningún valor en el momento. (Beck, 1999)



Retroalimentación

La agilidad se define, entre otras cosas, por la capacidad de respuesta ante los cambios que se van haciendo necesarios a lo largo del camino. Por este motivo uno de los valores que nos hace más ágiles es el continuo seguimiento o retroalimentación que recibimos a la hora de desarrollar en un entorno ágil de desarrollo. La retroalimentación se toma del cliente, de los miembros del equipo y de todo el entorno en el que se mueve un equipo de desarrollo ágil. (Beck, 1999)

1.6.4 XP en detalle.

Relatos de usuario.

- a) Similares a los Casos de Uso pero no lo mismo; estiman el tiempo con vista al plan de entrega.
- b) Las escriben los clientes, describen algo que el sistema debe hacer, son 2 o 3 oraciones en el lenguaje del usuario, sin jerga técnica, con poco detalle. La descripción detallada se dará al momento de iniciar el desarrollo.
- c) Definen una o más pruebas de aceptación.
- d) Corresponden a 1, 2 ó 3 semanas de desarrollo a tiempo completo.
- e) En un plan de entregas típico, se definen unas 80 historias \pm 20%.

Planificación de entrega.

- a) En una reunión de planificación se define un plan de entrega para todo el proyecto. El plan de entrega se usará para crear planes de iteración.
- b) Se toman decisiones de desarrollo y de negocios, se negocia un cronograma factible por todos.
- c) Se parte de los relatos de usuario, impresos o en tarjetas; moviéndolas en la mesa se define el primer (o el siguiente) conjunto a implementar, algo utilizable, verificable, con sentido para la empresa, entregable a corto plazo.
- d) Puede planearse por tiempo (cuántas historias antes de tal fecha) o por alcance (cuánto tiempo para estas historias), a la velocidad de programación estimada para el equipo.
- e) Las iteraciones se planifican en detalle recién al encararse, no antes. Las reglas están en el Portland Pattern Repository.



Capítulo I- Fundamentación Teórica

- f) Negociar hasta la coincidencia de clientes, desarrolladores y gerentes; no cambiar las estimaciones de los relatos de usuario.
- g) Un proyecto puede ser medido en 4 variables: 1) Alcance: qué se hará; 2) Recursos: gente disponible; 3) Fecha de entrega; 4) Calidad: bien hecho, bien probado. Sólo pueden elegirse 3 de las 4; la 4a. resulta de las anteriores. Además, la calidad no debiera cambiarse. (Beck, 1999)

El Plan de Entrega.

- a) Define los conjuntos de relatos de usuario a implementar en cada iteración, y sus fechas de entrega.
- b) Cuando se verifica un cambio de la velocidad de programación durante 2 ó 3 iteraciones deberá rehacerse el Plan de Entrega. (Beck, 1999)

Entregas frecuentes.

- a) El Plan de Entrega busca definir unidades funcionales con sentido para producir entregas frecuente.

Velocidad de programación.

- a) Es la cantidad de relatos de usuario o tareas de programación realizadas en una iteración (un simple conteo).
- b) Es inútil sacar medias por programador o por tarea; cada equipo tiene sus propias características.
- c) Se esperan variaciones en la velocidad. Si son extremas, rehacer el Plan de Entrega.

Desarrollo de una iteración.

- a) Dividir el Plan de Entrega en unas 12 iteraciones, más o menos. Cada iteración demora de 1 a 3 semanas.
- b) No adelantar planificación: cada iteración se planifica inmediatamente antes de comenzar. No adelantar implementación, desarrollar sólo lo previsto en la presente iteración.



Plan de iteración.

- a) Incluye los relatos de usuario y las pruebas de aceptación. Ambas se dividen en tareas de programación. Una tarea de programación abarca de 1 a 3 días de trabajo ideales, sin interrupciones ni otras actividades.
- b) El programador estima la tarea que toma.
- c) Estimar el tiempo de acuerdo a la velocidad de programación ya verificada. Agregar o quitar tareas si no alcanza el tiempo; si es preciso se renegociará. No engañarse ignorando o alterando el cálculo frío.
- d) Insistir en implementar primero lo esencial; si no se llega será más fácil renegociar lo accesorio.
- e) No sacrificar la refactorización ni las pruebas de unidad: descuidar esto atrasará más.

Rotación del personal.

- a) Mover al personal de un área a otra para evitar pérdidas de conocimiento o cuellos de botella en la codificación; todos deben conocer bastante código de otras secciones. La rotación y la programación en parejas produce el entrenamiento cruzado que evita las "islas de conocimiento".
- b) En cada iteración todos deben trabajar parte del tiempo en algún área nueva. En cada pareja de programadores se rota sólo uno por vez.

Reunión diaria de pie.

- a) Organizar una única reunión diaria, breve, de pie, con todos los integrantes, sólo para plantear problemas. Estos se resolverán luego en pequeños grupos de involucrados.

Corregir XP.

- a) Adaptar las reglas cuando no se obtienen los resultados esperados. Las reglas se pueden cambiar pero no pueden dejar de existir: cada desarrollador debe saber exactamente qué puede esperar de los demás, y los demás de él.



Capítulo I- Fundamentación Teórica

Buscar la simplicidad.

- a) Un diseño simple es más fácil de crear y mantener. Buscar afanosamente el diseño más simple que funcione. No implementar funcionalidad antes de tiempo.
- b) Advertencia: lograr un diseño simple es un gran trabajo.

Esquema de nombres ("system metaphor").

- a) Nombrar clases y métodos en forma consistente y clara. Debe ser posible inferir la significación real de un nombre nunca visto antes, a qué cosa se refiere.

Tarjetas CRC.

- a) Usar tarjetas CRC, "Clase, Responsabilidad, Colaboración", en el diseño grupal. Ayudan a evitar el enfoque procedimental y destacan la orientación a objetos.
- b) Cada tarjeta CRC representa un objeto. El nombre de Clase va arriba, las responsabilidades (qué debe hacer) a la izquierda, las clases asistentes (que colaboran) a la derecha.
- c) No suele ser necesario escribir la tarjeta completa; los participantes se familiarizan rápidamente con el propósito de cada clase.
- d) En la reunión CRC alguien simula el sistema discutiendo los mensajes intercambiados entre objetos. Limitar a 1 o 2 personas de pie exponiendo, mientras los demás permanecen sentados.

Solución rápida ("spike solution").

- a) Crear una solución rápida, un programa muy simple para explorar una solución posible, enfocada en el problema central sin atender el resto. Suele ser descartado; se emplea para probar la solución, mejorar la estimación, reducir el riesgo potencial.

Nunca agregar funcionalidad extra.

- a) No agregar funcionalidades extra sólo porque ahora se ve claro como hacerlo, o porque parezca mejorar mucho el sistema: el 90% de las funcionalidades adicionales no llegan a usarse. No prever requerimientos futuros ni flexibilidad extra. Concentrarse en el desarrollo de hoy.



Refactorizar a todo trance.

- a) Refactorizar es mejorar el código existente, la estructura interna del software, no su comportamiento visible.
- b) Refactorizar para mantener el diseño simple: quitar redundancia, eliminar funcionalidad no usada, rehacer diseños obsoletos, mantener el código limpio y conciso para que sea fácil de entender, modificar y extender.

El cliente siempre presente.

- a) El cliente debe participar activamente a lo largo de todo el proceso, el experto, no un ayudante. Debe haber una pareja de representantes del cliente asignadas al proyecto, a veces en tiempo completo.
- b) Los clientes escriben los relatos de usuario, participan en las estimaciones, asignan prioridades, verifican el cumplimiento de las funcionalidades. En la reunión de planificación de entrega negocian el conjunto de relatos de usuario a implementar en cada entrega.
- c) Los relatos de usuario no contienen detalle; los clientes deben estar presentes al definir las tareas de programación necesarias para implementar cada relato.
- d) Los clientes deben estar presentes en las pruebas de verificación para revisar el resultado y determinar cuando puede pasarse a producción el sistema.

Normas de codificación.

- a) Debe elegirse y respetarse una norma de codificación.

Codificar primero la prueba de unidad ("Unit Test").

- a) Las pruebas de unidad se escriben una vez y se corren reiteradamente a lo largo de todo el proyecto, asegurando siempre el funcionamiento correcto; evitan las ambigüedades, los requerimientos quedan duros en la prueba.
- b) Para cada unidad, se codifica primero una prueba simple para una función simple; se van agregando prueba y funcionalidad en etapas sucesivas, hasta implementar todo y probar todo en esa unidad.
- c) Una funcionalidad está terminada cuando pasa todas sus pruebas de unidad.



Programación en parejas.

Todo el código incluido en una entrega de producción es escrito por dos personas trabajando juntas ante un único computador. Uno digita y piensa tácticamente sobre el método en construcción; el otro piensa estratégicamente en la integración de ese método en la clase. (Beck, 1999)

- a) La programación en pareja produce la misma cantidad de código a un nivel de calidad superior.
- b) Lleva tiempo acostumbrarse a programar en pareja; puede resultar incómodo al principio.

Integración secuencial.

- a) Los grupos trabajan en paralelo, pero integran uno por vez: sólo una pareja de programadores integra, prueba y entrega los cambios en el repositorio de código, en un momento dado. Allí se fija una nueva versión.
- b) Una forma de asegurar la integración secuencial es hacerla en una máquina única (si los programadores trabajan en un mismo lugar físico).
- c) Los programadores pueden integrar libremente en sus máquinas de trabajo habituales.

Integración frecuente.

- a) Los programadores deben integrar su código al repositorio común cada pocas horas, al menos una vez por día cada pareja. Todos deben trabajar con la última versión.
- b) La integración continua evita los esfuerzos divergentes o fragmentados donde no se comunica lo que se puede compartir o reutilizar; las incompatibilidades se detectan temprano.



Código de propiedad colectiva.

- a) No hay una única persona responsable por el código: la construcción y la responsabilidad están distribuidas, todos los desarrolladores pueden agregar funcionalidad, corregir errores o refactorizar.
- b) El código incluye siempre las pruebas de unidad; todos los cambios serán controlados por la suite de prueba. El código sólo puede entregarse luego de haber pasado el 100 % de las pruebas de unidad.

Optimizar al final.

- a) Dejar la optimización para el final. No suponer dónde estarán los puntos de estrangulamiento; medirlos.
- b) "Make it work, make it right, then make it fast." (Hágalo funcionar, hágalo bien, después hágalo andar rápido).

No trabajar horas extra.

- a) El trabajo en horario extendido desmoraliza al equipo; no se cubrirá un atraso trabajando horas extra. Renegociar el alcance o la fecha de entrega en una reunión de planificación.
- b) Agregar personal a un proyecto atrasado es una mala idea.

Pruebas de unidad.

- a) Crear u obtener un marco de prueba ("test framework") para crear una suite automática de pruebas de unidad. Probar todas las clases del sistema (métodos triviales de "set" y "get" suelen omitirse).
- b) Crear las pruebas antes de escribir el código. No puede integrarse código sin sus pruebas de unidad.
- c) Las pruebas de unidad evolucionan junto con el código. No pueden crearse al final, ni dejar de escribirse.
- d) El tiempo de escribir las pruebas de unidad se gana con creces en la reiteración continua de las pruebas y la confianza al encarar cambios.



Capítulo I- Fundamentación Teórica

- e) Las pruebas de unidad posibilitan la propiedad colectiva de código, la refactorización, la integración frecuente. El agregado de funcionalidad incluye el agregado de pruebas.

Marco de prueba.

- a) El marco de prueba no es una herramienta de prueba sino de desarrollo, se debe usar desde el principio. Puede crearse desde cero, pero los hay disponibles para la mayoría de los lenguajes.

Cuando aparece un error.

- a) Se crean pruebas para ese error, para evitar su reaparición.
- b) La aparición de un error en producción requiere crear una prueba de aceptación. Ante una prueba de aceptación fallida, los programadores crean pruebas de unidad para ubicar el defecto en el código. Una vez superadas el 100% de las pruebas de unidad se vuelve a correr la prueba de aceptación fallida para verificar la desaparición del error.

Prueba de aceptación.

- a) Las pruebas de aceptación se crean a partir de los relatos de usuario. El cliente define los escenarios de prueba para verificar si el relato de usuario ha sido correctamente implementado. Un relato de usuario puede tener una o varias pruebas de aceptación. El cliente es responsable de verificar el pasaje de las pruebas de aceptación y priorizar la corrección de las pruebas fallidas.
- b) Las pruebas de aceptación son pruebas tipo caja negra a nivel del sistema: cada prueba de aceptación corresponde a un resultado producido por el sistema.
- c) Las pruebas de aceptación deben ser automáticas, correrse frecuentemente, publicarse sus resultados y programarse su corrección para la próxima iteración.
- d) Deben crearse pruebas de aceptación en cada iteración. Si no hay pruebas de aceptación nuevas no se ha hecho nada nuevo.
- e) Un relato de usuario no está completo hasta no haber pasado todas sus pruebas de aceptación.



1.6.5 ¿Qué propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes de que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo. (Beck, 1999)

1.6.6 Fases de la metodología XP

Fase I: Planificación

1- Se escriben historias de usuario, cuya idea principal es describir un caso de uso en dos o tres líneas con terminología del cliente (de hecho, se supone que deben ser escritos por el mismo), de tal manera que se creen test de aceptación para historias de usuarios (user store) y permita hacer una estimación de tiempo de desarrollo del mismo.

2- Se crea un plan de lanzamiento (release planning), que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto. Se usa como base las historias de usuario, participando el cliente en la elección de las que se desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto.

3- El desarrollo se divide en iteraciones, cada una de las cuales comienzan con un plan de iteración, para el que se eligen las historias de usuario a desarrollar y las tareas de desarrollo.

4- Se cambia el proceso cuanto sea necesario, para adaptarlo al proyecto. (Beck, 1999)

Fase II: Diseño

1- Se eligen los diseños funcionales más simples.

2- Se elige una metáfora del sistema para que el nombrado de clases, siga una misma línea, facilitando la reutilización y la comprensión del código.

3- Se escriben tarjetas de clase-responsabilidades-colaboración (CRC) para cada objeto, que permitan abstraerse al pensamiento estructurado y que el equipo de desarrollo completo participe en el diseño.



Capítulo I- Fundamentación Teórica

Fase III: Codificación

- 1- El cliente está siempre disponible, de ser posible, cara a cara. La idea es que forme parte del equipo de desarrollo, y esté presente en todas las fases de XP. La idea es usar el tiempo del cliente para estas tareas en lugar de crear una detallada especificación de requisitos, y evitar la entrega de un producto insuficiente, que le hará perder tiempo.

- 2- El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código.

- 3- Las pruebas unitarias se codifican antes que el código en sí, haciendo que la codificación de este último sea más rápida, y que cuando se afronte la misma se tenga más claro, qué objetivos tiene que cumplir lo que se va a codificar.

- 4- La programación del código se realiza en parejas, para aumentar la calidad del mismo. En cada momento, sólo habrá una pareja de programadores que integre código.

- 5- Se integra código y se lanza dicha integración de manera frecuente, evitando divergencias en el desarrollo y permitiendo que todo el mundo trabaje con la última versión del desarrollo. De esta manera, se evitará pasar grandes períodos de tiempo integrando el código al final del desarrollo, ya que las incompatibilidades serán detectadas enseguida.

- 6- Se usa la propiedad colectiva del código, lo que se traduce en que cualquier programador puede cambiar cualquier parte del código. El objetivo es fomentar la contribución de ideas por parte de todo el equipo de desarrollo.

- 7- Se deja la optimización para el final.

- 8- No se hacen horas extra de trabajo.



Capítulo I- Fundamentación Teórica

Fase IV: Pruebas

- 1- Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado.
- 2- Cuando se encuentra un error de codificación o bug, se desarrollan pruebas para evitar volver a caer en el mismo.
- 3- Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las user stories elegidas para la iteración, y son "pruebas de caja negra", en las que el cliente verifica el correcto funcionamiento de lo que se está probando. Cuando se pasa la prueba de aceptación, se considera que el correspondiente user storie se ha completado. (Beck, 1999)

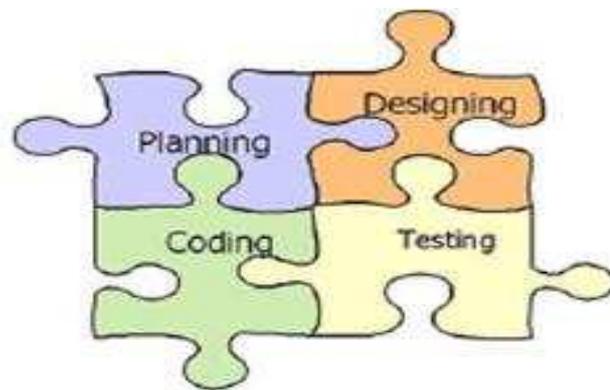


Figura # 1 Fases de la Metodología XP. Fuente: elaboración propia.

1.7 Arquitectura Cliente/Servidor

Arquitectura Cliente/Servidor: El que se utiliza lógicamente en el software es la Arquitectura Cliente/Servidor. Esta arquitectura es la plataforma abierta por excelencia, por la variedad de combinaciones de clientes y servidores que permite conectar en red. Sin embargo, elegir las plataformas, las herramientas, los proveedores y las bases de administración de la arquitectura Cliente/Servidor, además de la tecnología de creación, es una decisión difícil de tomar. Elegir un servidor es una cuestión muy complicada; para aplicaciones pequeñas y medianas, todos los servidores han probado ser muy buenos, las diferencias se darán cuando se necesiten altísimos regímenes transaccionales, y dependerán de cómo cada uno vaya incorporando nuevas características como paralelismo, "read ahead". Cada nueva versión puede modificar las posiciones y los principales fabricantes están trabajando al ritmo de una gran versión nueva por año. Hoy en día, el modelo Cliente/Servidor se considera clave para abordar las necesidades de



Capítulo I- Fundamentación Teórica

las empresas. El proceso distribuido se reconoce actualmente como el nuevo paradigma de sistemas de información, en contraste con los sistemas independientes. Este cambio fundamental ha surgido como consecuencia de importantes factores (negocio, tecnología, proveedores), y se apoya en la existencia de una gran variedad de aplicaciones estándar y herramientas de desarrollo, fáciles de usar que soportan un entorno informático distribuido. (Renaud, 1996)

1.7.1 Características de la Arquitectura Cliente/Servidor.

La arquitectura Cliente/Servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

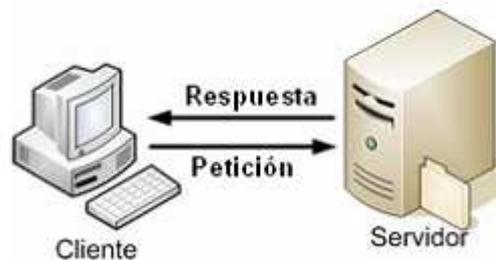


Fig. 2 Arquitectura Cliente/Servidor.

En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, inicia un proceso de diálogo: produce una demanda de información o solicita recursos. La computadora que responde a la demanda del cliente se conoce como servidor. Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura. Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet. Cliente/Servidor es el modelo de interacción más común entre aplicaciones en una red. No forma parte de los conceptos de la Internet como los protocolos IP, TCP o UDP, sin embargo todos los servicios estándares de alto nivel propuestos en Internet funcionan según este modelo. Se puede decir que la arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados



Capítulo I- Fundamentación Teórica

física y lógicamente, compartiendo datos, procesos e información, estableciendo así un enlace de comunicación transparente entre los elementos que conforman la estructura. No existe una definición específica adoptada universalmente de la Arquitectura Cliente/Servidor, las empresas de cómputo enfocan el concepto basándose en la funcionalidad que representa según los servicios que ellas mismas ofrecen. Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes: (Renaud, 1996)

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Todos los sistemas desarrollados en arquitectura Cliente/Servidor poseen las siguientes características distintivas de otras formas de software distribuido:

- Servicio: El servidor es un proveedor de servicios; el cliente es un consumidor de servicios.
- Recursos compartidos: Un servidor puede atender a muchos clientes al mismo tiempo y regular su acceso a recursos compartidos.
- Protocolos Asimétricos: La relación entre cliente y servidor es de muchos a uno; los clientes solicitan servicios, mientras los servidores esperan las solicitudes pasivamente.
- Transparencia de ubicación: El software Cliente/Servidor siempre oculta a los clientes la ubicación del servidor.
- Mezcla e igualdad: El software es independiente del hardware o de las plataformas de software del sistema operativo; se puede tener las mismas o diferentes plataformas de cliente y servidor.
- Intercambio basados en mensajes: Los sistemas interactúan a través de un mecanismo de transmisión de mensajes: la entrega de solicitudes y respuestas del servicio.
- Encapsulamiento de servicios: Los servidores pueden ser sustituidos sin afectar a los clientes, siempre y cuando la interfaz para recibir peticiones y ofrecer servicios no cambie.



Capítulo I- Fundamentación Teórica

- Facilidad de escalabilidad: Los sistemas Cliente/Servidor pueden escalarse horizontal o verticalmente. Es decir, se pueden adicionar o eliminar clientes (con apenas un ligero impacto en el desempeño del sistema); o bien, se puede cambiar a un servidor más grande o a servidores múltiples.
- Integridad: El código y los datos del servidor se conservan centralmente; esto implica menor costo de mantenimiento y protección de la integridad de los datos compartidos. Además, los clientes mantienen su individualidad e independencia.

La arquitectura Cliente/Servidor es una infraestructura versátil modular y basada en mensajes que pretende mejorar la portabilidad, la interoperabilidad y la escalabilidad del cómputo; además es una apertura del ramo que invita a participar a una variedad de plataformas, hardware y software del sistema.

Cliente

El cliente es la entidad por medio de la cual un usuario solicita un servicio, realiza una petición o demanda el uso de recursos. Este elemento se encarga, básicamente, de la presentación de los datos y/o información al usuario en un ambiente gráfico. Los clientes se suelen situar en PC's o en estaciones de trabajo se encargan de realizar el FRONT END, que es la parte de la aplicación que interactúa con el usuario, en ellos permanecen las aplicaciones particulares de cada usuario, y realizan funciones como: (Renaud, 1996)

- Manejo de la interfaz del usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.
- Como ejemplos de clientes pueden citarse interfaces de usuario para enviar comandos a un servidor, APIs para el desarrollo de aplicaciones distribuidas, herramientas en el cliente para hacer acceso a servidores remotos (por ejemplo, servidores de SQL) o aplicaciones que solicitan acceso a servidores para algunos servicios.

Servidor

El servidor es la entidad física que provee un servicio y devuelve resultados; ejecuta el procesamiento de datos, aplicaciones y manejo de la información o recursos. En el servidor se realiza el BACK END que es la parte destinada a recibir las solicitudes del



Capítulo I- Fundamentación Teórica

cliente y donde se ejecutan los procesos. En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Además, deben manejar los interbloques, la recuperación ante fallas, y otros aspectos afines. Por las razones anteriores, la plataforma computacional asociada con los servidores es más poderosa que la de los clientes. Por esta razón se utilizan PC's poderosas, estaciones de trabajo, minicomputadores o sistemas grandes. Además deben manejar servicios como administración de la red, mensajes, control y administración de la entrada al sistema ("login"), auditoría y recuperación, y contabilidad. Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.
- Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste, le responde proporcionándolo.

Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en computadoras personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo. En el servidor permanecen las aplicaciones que deben ser compartidas por varios usuarios. Normalmente, aunque con excepciones, el cliente y el servidor están ubicados en distintos procesadores; incluso, un servidor puede fungir como cliente de otros servidores. Existen diversos servidores mismos que se clasifican basándose en su funcionalidad; estos son denominados servidores dedicados ya que administran el uso de algún recurso en particular, por ejemplo, Servidor de Archivos: El cliente envía solicitudes de registros de archivos al servidor, es un simple servicio de datos compartidos por medio de la red. Servidor de bases de datos: El cliente envía solicitudes de SQL en calidad de mensajes (un mensaje por instrucción); el servidor hace uso de su propia capacidad de procesamiento para encontrar los datos solicitados y devolverlos por medio de la red. Servidores Web: Se usan como una forma inteligente para comunicación entre empresas a través de Internet. Este servidor permite transacciones con el acondicionamiento de un browser específico. Este modelo está integrado por clientes compactos y portátiles en comunicación con servidores amplios. Tal comunicación se da mediante un protocolo denominado HTTP del inglés Hypertext Transfer Protocol. (Renaud, 1996)



1.8 Herramientas para el desarrollo de la aplicación

1.8.1 PHP Como Lenguaje de Programación.

Luego de hacer el análisis entre los lenguajes que implementan servicios Web, se decide utilizar el PHP embebido en el código HTML ya que:

- Está soportado en la mayoría de las plataformas de Sistemas Operativos, mientras que con ASP por ser propiedad de Microsoft no es multiplataforma.
- El PHP no tiene costo oculto, o sea que cuando se adquiere incluye un sin número de bibliotecas que proporcionan el soporte para la mayoría de las aplicaciones Web, por ejemplo e-mail, generación de ficheros PDF y otros. En caso de que no se tengan las bibliotecas estas se pueden encontrar gratis en Internet. En el caso de ASP forma parte del Internet Information Server que viene integrado en Windows NT-2000 Server con su elevado costo de adquisición.
- PHP y ASP son parecidos en cuanto a la forma de utilización, pero PHP es más rápido, gratuito y multiplataforma. (Intercambiosvirtuales.org, 2006)

1.8.2 MySQL Como Gestor de Base de Datos

Luego de analizadas las características y facilidades de los SGBD presentados, y de las herramientas a desarrollar, se decide usar el MySQL como SGBD, por las siguientes razones: MySQL es muy rápido, fiable y fácil de usar. Es un sistema multiplataforma de base de datos relacionales, lo que da velocidad y flexibilidad, cuenta con un sistema de privilegios contraseñas, muy seguro que permite la autenticación básica para el acceso al servidor. (Wikipedia)

- No se necesitará de un manejo complejo de la información.
- El PHP maneja más fácil al MySQL que a PostgreSQL, debido a la gran cantidad de funciones que tiene explícitas.
- El MySQL tiene mayor rendimiento y mayor velocidad mientras PostgreSQL es de 2 a 3 veces más lento que MySQL.
- Mejores utilidades de administración (backup, recuperación de errores, etc.).
- No suele perder información ni corromper los datos.
- No hay límites en el tamaño de los registros, PostgreSQL tiene un límite del tamaño de cada fila de las tablas a 8k.



Capítulo I- Fundamentación Teórica

- Mejor control de acceso, en el sentido de que usuarios tienen acceso a que tablas y con que permisos.
- MySQL se comporta mejor que PostgreSQL a la hora de modificar o añadir campos a una tabla "en caliente".
- MySQL por su bajo consumo lo hace apto para ser ejecutado en una máquina con escasos recursos sin ningún problema, PostgreSQL consume más recursos y carga más el sistema.

1.8.3 XAMPP

Es un servidor independiente de plataforma, software libre, incluye el servidor web Apache, los servidores de datos MySQL y SQLite, sus respectivos gestores phpMyAdmin y phpSQLiteAdmin, el intérprete del lenguaje homónimo PHP con los extras incluidos en Perl, el intérprete del lenguaje Perl, servidores de FTP como ProFTPD ó FileZilla FTP Server, las estadísticas Webalizer y OpenSSL, eAccelerator, Freetype2, libjpeg, libpng, zlib, Ming, etc. entre otros. El programa esta 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 para Microsoft Windows, GNU/Linux, Solaris y MacOS X. (BLANCO CRIADO, 2008)

- ❖ Incluye Chequeo de seguridad.
- ❖ Contiene un Panel de control.

Incluye herramientas extras (Webalizer, Mercury Mail, Conmutador PHP).

1.8.4 MYSQL Workbench 7 OSS

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. Se diseña para hacer más fácil de entender el estado actual de los datos. Simple y fácil al usuario, ayuda a organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia. (Wikipedia)

Ventajas:

- Si se está comenzando un nuevo diseño o está manteniendo una base de datos existente, se combina con las características para ayudarle a conseguir el trabajo hecho con eficacia.



Capítulo I- Fundamentación Teórica

- La creación de diagramas es clara y rápida.
- Tiene la posibilidad de realizar diagramas con desempeño rápido.
- Te da la posibilidad de darle un nombre a la Base de Datos ante de exportarla.

En este capítulo se abordaron elementos necesarios para la comprensión y fundamentación de la solución propuesta. Las tendencias y tecnologías actuales relacionadas con el tema, se hizo una valoración del lenguaje de programación, el sistema gestor de bases de datos, y la metodología de desarrollo. Una vez conocidas las herramientas y conceptos a usar se puede proseguir con el diseño y la construcción de la solución propuesta.



CAPÍTULO II. PLANIFICACIÓN Y DISEÑO

2.1 Introducción

En este capítulo se aborda la fase de planeación y diseño, donde se describen las necesidades del cliente, se detallan 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.

2.2 Funcionalidades generales.

Lo que se busca con la automatización de la Maestría de Electromecánica es que los que cumplan con los requisitos para realizar la maestría puedan acceder a ella de forma digital a través de un sitio web, con la facilidad de que estos no tengan la necesidad de viajar desde su país hasta Cuba para realizar la misma. Este es un proceso que actualmente se hace de forma manual y con el maestrante personalmente lo que hace más complicado el proceso para los mismos. En la actualidad los maestrantes deben pedir matrícula entregando una serie de requisitos y con el sitio este podrá subir estos requisitos y esperar respuesta del coordinador de la maestría, si puede o no realizar la maestría.

2.2.1 Personal relacionado con el sistema.

Personas relacionadas con el sistema	Justificación
Coordinador	Se le denomina Coordinador a todos aquellos usuarios que se le asignan ciertos privilegios para operar sobre el sistema (no puede ser Administrador del sistema).
Estudiantes	Se le denomina Estudiantes a todos aquellos usuarios que van a recibir el curso.
Usuarios	Se le denomina Usuarios a todos aquellos usuarios que pueden ejecutar las opciones no privilegiadas que brinda el sistema (invitados).
Pensum	Es un conjunto de asignaturas y actividades no lectivas que conforman el currículum de una actividad de Postgrado en este caso de la Maestría de Electromecánica

Tabla 2.1 Personas relacionadas con el sistema



2.2.2- Lista de reserva.

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

- Autenticar Usuarios.
- Insertar Usuarios.
- Eliminar usuarios.
- Modificar usuarios.
- Mostrar listado de usuarios.
- Insertar pensum.
- Eliminar pensum.
- Mostrar listado de pensum.
- Insertar ediciones vinculantes.
- Eliminar ediciones vinculantes.
- Mostrar listado de ediciones vinculantes.
- Insertar asignatura.
- Eliminar asignatura.
- Mostrar listado de asignatura.
- Insertar documentación.
- Eliminar documentación.
- Mostrar listado de documentación.
- Solicitar matrícula en una edición.
- Mostrar información de un pensum según la edición
- Mostrar listado de solicitudes.
- Mostrar información de un estudiante.
- Mostrar información de un profesor.
- Mostrar ediciones dada una modalidad.

2.2.3 Historias de usuario

Se plantean las historias de usuario previstas por el cliente, técnica utilizada por XP para especificar los requisitos del sistema. Estas tienen la misma finalidad que los casos de uso de la metodología RUP pero con algunas diferencias. Teniendo en cuenta el esfuerzo asociado a las historias de usuario y las prioridades del cliente se define una versión que sea de valor para el cliente. Se describen brevemente las



CAPITULO II – PLANIFICACIÓN Y DISEÑO

características que el sistema debe tener desde la perspectiva del cliente. (Basulto, 2010)

Historia de usuario	
Número: No. Historia de usuario	Usuario: Usuario entrevistado para obtener la función requerida a automatizar.
Nombre historia: Nombre de la historia de usuario que sirve para identificarla mejor entre los desarrolladores y el cliente.	
Prioridad en el negocio: Importancia de la historia para el cliente: Alta / Media / Baja	Riesgo en desarrollo: Dificultad para el programador: Alta / Media / Baja
Puntos estimados: Estimación: de 1 a 3 puntos	Iteración asignada: Iteración a la que corresponde
Programador responsable: Daimara Arias Ramírez	
Descripción: Se especifican las operaciones por parte del usuario y las respuestas del sistema.	
Observaciones: Algunas observaciones de interés, como glosario, información sobre usuario etc.	

Tabla 2.2. Modelo de planilla de historias de usuario

Las historias de usuario descritas por el cliente:

Historia de usuario	
Número: 1	Usuario: Arístides A. Legrá Lobaina.
Nombre : Autenticación de usuarios	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Daimara Arias Ramírez.	
Descripción: El Administrador debe de ingresar sus datos para necesitar ingresar al sistema (nombre de usuario y contraseña. El sistema guarda los datos.	
Observaciones: Confirmado.	

Tabla 2.3 HU No.1: Autenticar usuarios

Las demás Historias de Usuarios están en él: [Anexo1](#)



2.3-Planificación de Entregas.

En esta fase se establece la prioridad de cada HU, y a continuación, se realiza una estimación del esfuerzo necesario de cada una de ellas por parte de los programadores. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debe obtenerse en no más de dos a tres meses.

Las estimaciones asociadas a la implementación de las historias se establecen empleando como medida el punto de estimación. Un punto de estimación equivale a una semana ideal de programación, donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción, este punto de estimación que se utiliza para representar la semana ideal, es de 5 días. Las historias generalmente tienen un valor de 1 a 3 puntos. Además, se mantiene un registro de la velocidad de desarrollo, establecida por puntos de iteración, basado fundamentalmente en la suma de puntos de estimación correspondientes a las HU, que fueron terminadas en la última iteración.

2.3.1 Estimación de esfuerzo por historia de usuario

Para el buen desarrollo del sistema propuesto, se realizó una estimación para cada una de las HU identificadas, y se obtienen los resultados que se muestran a continuación:

Historias de usuarios	Puntos de estimación
Autenticar Usuarios	1 día
Gestionar Usuarios.	1 semana
Gestionar Pensum	1 semana
Gestionar ediciones vinculantes	2 semana
Gestionar asignaturas	1 semana
Gestionar Documentación	1 semana
Solicitar Matricula en una edición	2 semana
Mostrar información de un pensum según la edición	1 semana
Mostrar listado de solicitudes	2 semana
Mostrar información de un estudiantes	1 semana
Mostrar información de un profesor	1 semana
Mostrar información de ediciones dada una modalidad.	2 semana

Tabla 2.16 Estimación de esfuerzo por historia de usuario



2.3.2 Planificación de iteraciones

A partir de las HU antes expuestas y la estimación del esfuerzo propuesto para la realización de las mismas, se procede a realizar la planificación de la etapa de implementación del sistema, apoyándose en el tiempo e intentando concentrar las funcionalidades relacionadas en una misma iteración. En este plan se establece cuántas iteraciones serán necesarias realizar sobre el sistema para su terminación. El plan de iteraciones puede contener indicaciones sobre cuáles HU se incluirán en un release, lo cual debería ser consistente con el contenido de una o dos iteraciones. En relación con lo antes tratado se decide realizar el sistema en 3 iteraciones, las cuales se explican de forma detalla a continuación:

Primera iteración:

Esta iteración tiene como objetivo darle cumplimiento a las HU que se consideraron de mayor importancia para el desarrollo de la aplicación. Al concluir dicha iteración se contará con todas las funcionalidades descritas en las HU 1 ,2 ,3 y 4, las cuales hacen alusión a la Autenticación de usuarios, Gestión de usuarios, Gestión de Pensum y Gestión de ediciones vinculantes.

Segunda iteración:

Esta iteración tiene como finalidad desarrollar las HU 5, 6,7 y 8. Las mismas son las que brindan las funcionalidades de Gestionar Asignatura, Gestionar Documentación, Solicitar matricula en una edición y Mostrar información de un pensum según la edición.

Tercera iteración:

Esta última iteración de la aplicación tiene como propósito llevar a cabo el desarrollo de la HU 9, 10, 11, y 12. La cuáles proporcionan las funcionalidades de: la historia de usuario N0.9 nos va a generar un listado con todas las solicitudes, la NO.10 nos va a mostrar la información de un estudiante, la N0.11 nos muestra la información de un profesor y la N0.12 muestra una edición dada una modalidad (tiempo completo o parcial). Estas historias de usuario serán integradas con el resultado de las iteraciones anteriores, y como fruto de esta integración se obtendrá la versión 1.1 del producto final. A partir de este momento la aplicación será puesta a un proceso de prueba para evaluar el desempeño de la misma.



2.3.3 Plan de duración de las iteraciones

Como parte del ciclo de vida de un proyecto guiado por la metodología de desarrollo de software XP, se crea el plan de duración de cada una de las iteraciones que se llevarán a cabo durante el desarrollo del mismo. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas las HU en cada una de las mismas.

Iteración	Historias de usuario	Duración total
Iteración 1	Autenticar Usuarios	4 semanas y 1 día
	Gestionar Usuarios.	
	Gestionar Pensum	
	Gestionar Ediciones Vinculantes	
Iteración 2	Gestionar Asignatura	5 semanas
	Gestionar Documentación	
	Solicitar matricula en una edición	
	Mostrar información de un pensum según la edición	
Iteración 3	Mostrar listado de solicitudes	6 semanas
	Mostrar información de un Estudiantes	
	Mostrar información de un profesor	
	Mostrar información ediciones dada una modalidad	

Tabla 2.17 Plan de duración de las iteraciones

2.4 Tarjetas CRC

En este epígrafe tiene lugar la realización de las tarjetas de clases, responsabilidades y colaboración, conocidas tradicionalmente como *tarjetas CRC*, las cuáles se realizan con el objetivo de facilitar la comunicación y documentar los resultados. Además, las mismas permiten la total participación y contribución del equipo de desarrollo en la tarea de diseño. Una tarjeta CRC representa un objeto, por tanto es una clase, cuyo nombre se ubica en forma de título en la parte superior de la tarjeta, los atributos y las responsabilidades más significativas se colocan a la izquierda y las clases implicadas con cada responsabilidad a la derecha, en la misma línea de su requerimiento



CAPITULO II – PLANIFICACIÓN Y DISEÑO

correspondiente. Para mejor comprensión de las mismas, se determina agruparlas por HU. (Basulto, 2010)

Clases	
Responsabilidades	Colaboraciones
Clase A	Clase B

Tabla 2.18: Plantilla de tarjeta CRC

Clase Autenticar usuarios	
Responsabilidades	Colaboraciones
Autenticar_usuarios	Conexión

Tabla 2.19: Tarjeta CRC No.1 Autenticar Usuarios.

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

En este capítulo se abordó la fase de planeación y diseño donde se delinearon las HU con la participación del cliente, se llevó a efecto la planificación de iteraciones de cada HU a partir de la estimación del esfuerzo necesario de las mismas. Además presentando las principales clases mediante el empleo de las tarjetas CRC, estamos listos para pasar a la siguiente etapa de desarrollo y pruebas.



CAPITULO III - DESARROLLO Y PRUEBAS

3.1 Introducción

En este capítulo se inicia la fase de desarrollo y pruebas conforme a la metodología XP. Se presenta el modelo de datos empleado para la aplicación concluyente, y se realiza el desarrollo de las iteraciones a partir del desglose de las historias de usuario en tareas. Asimismo aparecen las interfaces gráficas de usuario diseñadas para la aplicación final.

3.2 Modelo de datos.

El modelo físico de la base de datos con la cual trabaja esta Aplicación Web, nos sirve para entender con más facilidad como esta diseñada nuestra base de datos y las relaciones de las mismas, así como el tipo de relación que tienen estas.

Para ver el Modelo de Datos de la Aplicación ir al: [Anexo 5](#)

3.3 Desarrollo de las iteraciones

Durante la fase planificación y diseño fueron detalladas las historias de usuario correspondientes a cada una de las iteraciones a desarrollar, teniendo en cuenta las prioridades y restricciones de tiempo, previstas por el cliente.

Para darle cumplimiento a cada HU, primeramente se debe realizar una revisión del plan de iteraciones, y si es necesario, se le hacen modificaciones a este.

3.3.1 Tareas por historias de usuario

Dentro del contenido de este plan, las HU se descomponen en tareas de programación o ingeniería, y a su vez, estas son asignadas al equipo de desarrollo para su implementación. Las tareas no tienen que ser entendidas necesariamente por el cliente, pues las mismas, sólo son utilizadas por los miembros del equipo de desarrollo, por lo que pueden ser escritas en lenguaje técnico. Las mismas se representan mediante las tarjetas de tareas. (Basulto, 2010)



CAPITULO III – DESARROLLO Y PRUEBAS

Historia de usuario	Tareas
Autenticar Usuarios	Introducir los datos para entrar al Sistema.
Gestionar Usuarios	Insertar usuarios. Eliminar usuarios. Modificar usuarios. Mostrar listado de usuarios.
Gestionar Pensum	Insertar Pensum. Eliminar Pensum. Mostrar listado de Pensum. Buscar y visualizar pensum dada una edición.
Gestionar Ediciones Vinculantes	Insertar ediciones vinculantes. Eliminar ediciones vinculantes. Mostrar listado de ediciones vinculantes.
Gestionar Asignaturas	Insertar Asignaturas. Eliminar Asignaturas. Mostrar listado de Asignaturas.
Gestionar Documentación	Insertar documentación Eliminar documentación Mostrar listado de documentación
Solicitar matrícula en una edición	Solicitud de matrícula
Mostrar información de un pensum según la edición	Mostrar información de pensum dada la edición.
Mostrar listado de solicitudes	Mostrar solicitudes
Mostrar información de un estudiante	Mostrar estudiante
Mostrar información de un profesor	Mostrar profesor
Mostrar ediciones dada una modalidad	Mostrar ediciones dada una modalidad

Tabla3.1 Distribución de tareas por historia de usuario

3.3.2 Iteración No.1

En esta iteración se le dio cumplimiento a la implementación a las HU que se consideraron de mayor importancia para el desarrollo de la aplicación, con el fin de obtener una versión del software.

Historias de usuario	Tiempo de estimación (semanas)
Autenticar Usuarios	1 día
Gestionar Usuarios	1
Gestionar Pensum	1
Gestionar Ediciones vinculantes	2

Tabla3.2: Historias de usuario abordadas en la primera iteración.



Tarea ingeniería	
Número tarea: 1	Número historia: 2
Nombre tarea: Insertar Usuario.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea inserta nuevos Usuario que el Administrador estime necesario, los que van a interactuar con el sistema.	

Tabla 3.3: Tarjeta de tarea No.1 Insertar Usuario

Para ver las otras Tarjetas de Tareas ir al: [Anexo 3](#)

3.4 Pruebas

En la Metodología XP las pruebas juegan un papel fundamental, pues esta permite la comprobación continua del código. El desarrollo constante de las pruebas da lugar a que se desarrolle un software con mayor calidad dando una mayor seguridad de lo que se está haciendo. En esta metodología hay dos tipos de pruebas; las unitarias o desarrollo dirigido por pruebas, desarrolladas por los programadores verificando su código de forma automática, y las pruebas de aceptación, las cuáles son evaluadas luego de culminar una iteración verificando así que se cumplió la funcionalidad requerida por el cliente. (Basulto, 2010)

3.4.1 Desarrollo dirigido por pruebas

El desarrollo dirigido por pruebas, se enfoca en la implementación orientada a pruebas. El código debe ser probado paso a paso para lograr un resultado, aunque no con lógica para el negocio, pero si funcional. Algunas personas confunden este término con las llamadas “pruebas de caja blanca” las cuáles se les practican a los métodos u operaciones para medir la funcionalidad del mismo, desde el punto de vista de validez del cliente. Sin embargo, el TDD se aplica antes de comenzar a implementar cada paso



de la tarea en desarrollo, asumiendo que la prueba es insatisfactoria desde un inicio. Sólo una vez que se haya cumplido de la forma más sencilla posible la lógica del código a probar se asume como cumplida. Luego se realiza un proceso conocido como “refactorización” de código perteneciente a una de las doce prácticas planteadas por la metodología XP, el cual consiste en mantener el código en buen estado, modificándolo activamente para que conserve claridad y sencillez. (Basulto, 2010)

3.4.2 Pruebas de aceptación

Las pruebas de aceptación en XP, se pueden asociar con las pruebas de caja negra que se aplican en la metodología RUP, sólo que se crean a partir de las historias de usuario y no por un listado de requerimientos. Durante las iteraciones, las HU se traducen a pruebas de aceptación. En ellas se especifican desde la perspectiva del cliente, los escenarios para probar que una historia de usuario ha sido implementada correctamente. La misma puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo que persiguen estas pruebas, es garantizar que las funcionalidades solicitadas por el cliente han sido realizadas satisfactoriamente. Una HU no se considera completa hasta que no ha transitado por sus pruebas de aceptación.

Luego de ver lo anteriormente escrito y el Autor reunirse con el cliente para su análisis, se decidió llevar a cabo el proceso mediante las pruebas de aceptación.

La planilla utilizada para plasmar el contenido de las pruebas de aceptación se muestra a continuación. (Basulto, 2010)

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 realice.
Entrada/Pasos ejecución: Pasos para probar la funcionalidad.
Resultado: Resultado que se desea de la prueba.
Evaluación de la prueba: (Aceptada o denegada) o (Satisfactoria o no satisfactoria)

Tabla 3.27 Planilla de prueba de aceptación



CAPITULO III – DESARROLLO Y PRUEBAS

Prueba de aceptación
HU: Autenticar usuarios
Nombre: Prueba para comprobar la Autenticación de usuarios. (Nombre de usuario y contraseña).
Descripción: Validación de entrada de los datos de los usuarios.
Condiciones de ejecución: El usuario debe introducir su nombre de usuario y contraseña.
Entrada/Pasos ejecución: El usuario escribe su nombre de usuario y contraseña y luego da clic en el botón Entrar.
Resultado: <ul style="list-style-type: none">• Si el usuario tiene acceso para entrar a la aplicación e inserta sus datos correctamente entrará sin problemas al Sistema. Se emite un mensaje de error en caso de que: <ul style="list-style-type: none">• Se inserte los datos de un usuario no válido para el Sistema o incorrectos. (Ya sea su nombre de usuario o su contraseña).• Se dé clic en el botón Iniciar sin insertar nada en los campos de texto.
Evaluación de la prueba: Aceptada.

Tabla 3.28 Pruebas de aceptación para la HU: Autenticar usuarios

Para observar las demás Pruebas de Aceptación ir al: [Anexo 4](#)

En este capítulo se llevó a cabo la fase de desarrollo y diseño donde se presenta el modelo de datos de la aplicación a obtener, logrando una visión detallada de sus atributos y las relaciones entre sus clases. Se realiza el desarrollo de las iteraciones a partir de la distribución de tareas por HU, y se les practica las pruebas de aceptación a las funcionalidades de mayor importancia.



CAPITULO IV - ESTUDIO DE FACTIBILIDAD

4.1 Introducción

En el presente capítulo se muestran los resultados obtenidos en el Estudio de Factibilidad del Proyecto. El mismo se realizó basado en la Metodología Costo Efectividad (Beneficio), la cual plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores:

- 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 la capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo para 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 de cumplimiento del objetivo).

4.2 Efectos económicos

- Efectos directos
- Efectos indirectos
- Efectos externos
- Intangibles

Efectos directos:

POSITIVOS:

- El administrador y los usuarios del sistema (alumnos, profesores y coordinador de la Maestría de Electromecánica) tendrán la posibilidad de realizar de forma integrada sus actividades de matrícula y de acceso a la información docente y metodológica de las asignaturas del pensum de estudio.
- Se establece un mecanismo eficaz y eficiente de control interno del postgrado académico.
- Se obtiene información actualizada, resumida y detallada, del proceso docente de la maestría.
- Se integra la gestión del conocimiento en el área de postgrado con el sistema de gestión del CEETAM.



NEGATIVOS:

- Para el uso de esta aplicación implementada en plataforma Web se necesitará que la misma sea ejecutada preferentemente con el navegador Mozilla Firefox, porque es con el que se trabajó en la elaboración del producto por lo que el diseño está adaptado a este tipo de navegador.

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 venta.

Efectos externos:

- Se obtendrá un producto disponible que le facilitará gran parte del trabajo a los Coordinadores y Profesores encargados de la Maestría de Electromecánica.

Intangibles:

- En la valoración económica siempre hay elementos 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 dos situaciones:

- **SITUACIÓN SIN EL PRODUCTO**

Para llevar a cabo la Maestría de Electromecánica debe tener en cuenta los siguientes pasos:

1. Los maestrantes deben presentarse y entregar varios documentos que son necesarios para ser aprobados en la maestría, estos pueden ser de distintos países.
2. Los maestrantes deben recibir las clases personalmente o en algunos casos por sistemas no presenciales.
3. El trabajo del coordinador se torna engorroso, debido al gran volumen de información que este maneja.



CAPITULO IV - ESTUDIO DE FACTIBILIDAD

- **SITUACIÓN CON EL PRODUCTO**

Para llevar a cabo la Maestría de Electromecánica se realizan los siguientes pasos:

1. Entrar los datos correspondientes del usuario ya sea Administrador, Coordinador, Profesor o Estudiante que hará uso de la aplicación.
2. Según el tipo de usuario podrá realizar operaciones específicas del Sistema.
3. Los maestrantes podrán enviar sus datos utilizando el sistema y una vez que el coordinador los revise podrá aprobar o no la solicitud del maestrante.
4. Los maestrantes encontraran todos los documentos necesarios para poder realizar la maestría.
5. El trabajo del coordinador se facilitara porque tendrá toda la información informatizada.

4.3 Beneficios y Costos Intangibles en el proyecto

Costos

- Resistencia al cambio.

Beneficios

- Más comodidad para los usuarios.
- Los procesos informativos de la Maestría de Electromecánica se realizan con mayor rapidez y fiabilidad.
- Conectividad desde cualquier PC que esté conectada a la red.

4.4 Ficha de costo

Para determinar el costo económico del proyecto se utilizará el procedimiento para elaborar una Ficha de Costo de un producto.

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



CAPITULO IV - ESTUDIO DE FACTIBILIDAD

4.4.1 Costos en Moneda Libremente Convertible:

Ficha de Costo.		Precio(s)
Costos Moneda Libremente 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		25,00
Materiales directos		0,00
Subtotal		25,00
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		25,00

Tabla 4.4.1: Costo en Moneda Libremente Convertible



CAPITULO IV - ESTUDIO DE FACTIBILIDAD

4.4.2 Costos en Moneda Nacional:

Ficha de Costo.	
	Precio(s)
Costos Moneda Nacional	
Costos Directos	
Salario del personal que laborará en el proyecto	100,00
12,5% del total de gastos por salarios se dedica a la seguridad social	0,00
9.09% de salario total, por concepto de vacaciones a acumular	0,00
Gasto por consumo de energía eléctrica	270,70
Gastos en llamadas telefónicas	0,00
Gastos administrativos	0,00
Subtotal	370,70
Costos Indirectos	
Know How	0,00
Subtotal	
Total	370,70

Tabla 4.4.2: Costo en Moneda Nacional.

Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la Metodología Costo- Efectividad. Dentro de la misma 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 la Automatización de la Maestría de Electromecánica del CEETAM y la variable sería la complejidad de las pruebas que se realizan durante este proceso.

Valores de la variable (Solución manual):

- Realizar solicitud de matrícula a la Maestría. 9 variables (70 min).
- Recibir clases personalmente. 1 variable (90 min).
- El coordinador de forma manual lleva mucho contenido Académico así como toda la información de sus alumnos y de su claustro de profesores. 3 variables (480 min.).

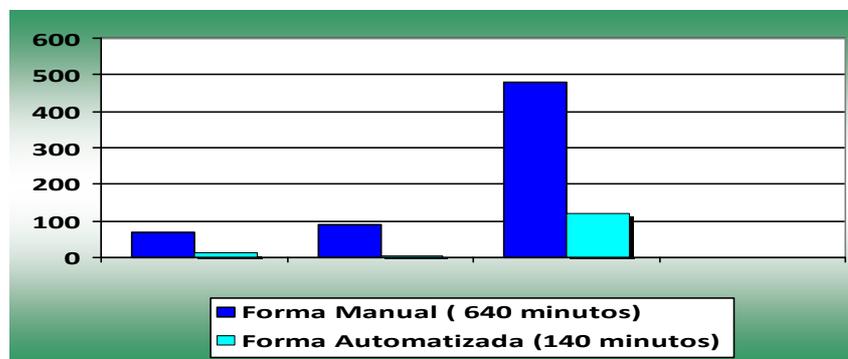


CAPITULO IV - ESTUDIO DE FACTIBILIDAD

Valores de la variable (Solución con el software):

- Cargar la planilla con los datos necesarios y llenar el formulario correspondiente para la realización de solicitud de matrícula. 7 variables (15 min).
- Descargar toda la documentación necesaria para recibir las clases. 1 variable (5 min.).
- Todo el contenido académico, la información de los alumnos y claustro de la maestría que lleva el coordinador estará plasmado en el sitio. 1 variable (120 min.).

Gráfica de la solución manual y solución automatizada



Teniendo en cuenta los resultados reflejados en la gráfica queda demostrada la factibilidad del sistema el tiempo que demora la solución del mismo de forma manual y automatizada.

En este capítulo se realizó el estudio profundo del costo real en que se incurrió durante el diseño e implementación de la aplicación, software mediante la Metodología Costo Efectividad (Beneficios), se analizaron todos los factores directos, indirectos, externos e intangibles, así como se calculó el costo de ejecución del producto software mediante la ficha de costo arrojando como resultados (**costo**) demostrándose la conveniencia de la elaboración del sistema.



CONCLUSIONES GENERALES

Al finalizar el desarrollo de esta investigación, el cual consta de cuatro capítulos donde recogen ampliamente lo realizado en el informe se arribaron a las siguientes conclusiones:

- Se desarrolló el producto final que consistía en el Diseño e Implementación de una aplicación Web para la Gestión de la Información de la Maestría Electromecánica del ISMMM.
- Se realizó la planeación y diseño de la aplicación, en los cuales se identificaron y especificaron los requisitos funcionales, así como se llevaron a cabo las posteriores fases de codificación y pruebas, 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 confeccionaron las tarjetas CRC, además de que se desarrollaron las tareas correspondientes para dar solución a las historias de usuario.
- Se realizaron las pruebas de aceptación definidas por el usuario, lo que arrojó como resultado su aceptación, con lo cual se demostró el cumplimiento satisfactorio de las historias de usuarios.
- Se efectuó un estudio detallado de la factibilidad del producto final, el cual arrojó los resultados esperados.



RECOMENDACIONES

Con vista al desarrollo futuro de este proyecto se recomienda:

- Primeramente comenzar a utilizar el módulo propuesto, para apoyar la extracción del conocimiento del Sitio Web de la Maestría Electromecánica del ISMMM.
- Incentivar la creación de nuevas herramientas para continuar el desarrollo del Sitio Web de la Maestría Electromecánica del ISMMM.
- Agregar nuevas funcionalidades acorde a nuevos requisitos que pudiesen surgir por alguna causa o para aumentar el rendimiento del Sistema.
- Realizar un estudio más profundo de este sistema en vista a perfeccionarlo en versiones futuras.
- Extender el software a cualquier Entidad de nuestro país tanto dentro y fuera de este ya que el producto lo permite.



Fuentes Bibliográficas

Basulto, Jorge Mario. 2010. *Sistema de Gestión integral de la empresa Empleadora del NIQUEL "EMPLANI". MODULO GESTION DE CONTRATOS DE COMPRAS.* Moa : Instituto Superior Minero Metalúrgico de Moa, 2010.

Beck, Kent. 1999. *Extreme Programming Explained. Embrace Change.* [trad.] Addison Wesley. s.l. : Pearson Education, 1999.

BLANCO CRIADO, A. 2008. *XAMPP.* [En línea] 2008.

<http://www.adictosaltrabajo.com/tutoriales>.

Bureaudeprensa.com. 2008. *Embarcadero ER/Studio.* [En línea] 2008.

http://bureaudeprensa.com/es/view.php?bn=bureaudeprensa_software&key=1153755975.

Ciberaula.com. 2005. *Ciberaula.com. Una Introducción a APACHE.* [En línea] 2005.

http://linux.ciberaula.com/articulo/linux_apache_intro.

Cuenca Muguercia, Ariel Ricardo. 2010. *Sistema Automatizado para el control de los indicadores de gestión de un cuadro de mando integral.* 2010.

Dougiamas, Martin y Moodle, Equipo de desarrollo de. 2008. *Moodle.* 2008.

FERRER, J. 2008. *Metodologías Ágiles.* [En línea] 2008.

<http://libresoft.es/downloads/ferrer-20030312.pdf> .

Intercambiosvirtuales.org. 2006. *Intercambiosvirtuales.org. PHP Designer 2008 v6.0.2.0*

Professional. [En línea] 2006. <http://www.intercambiosvirtuales.org/software/php-designer-2008-v6020-professional>.

JEFFRIES, R. 2009. *What is Extreme Programming?* [En línea] 2009.

<http://www.xprogramming.com/xpmag/whatisxp.htm>.

Monmany, J. *Aplicaciones Web.* [En línea] <http://www.webvillage.info>.



Fuentes bibliográficas

Renaud, Paul E. 1996. *Introduction to client/server systems. A practical guide for systems professionals.* . New York : John Wiley & Sons, 1996.

Rodríguez Granada, Yordis. 2009. *Sistema Automatizado para la gestión de información en la secretaria general del ISMMM.* . 2009.

Universidad de Oriente. *Metodología XP.* [En línea]
http://wikipedia.uo.edu.cu/es/articles/p/r/o/Programación_Extrema_3b63.html.

Wikipedia. Wikipedia, la enciclopedia libre. MySQL. [En línea]
<http://es.wikipedia.org/wiki/MySQL>.

Hernán Ruiz, Marcelo. 2006. *Programación Web Avanzada.* La Habana, Editorial Félix Varela.



GLOSARIO DE TÉRMINOS

Testing: Las pruebas de software, en inglés *testing* son los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador o videojuego. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

Open-Source: Se dice así cuando estamos en presencia de un software libre al cual permite estudiar su código y adecuarlo a las necesidades que se necesite, o sea es un código abierto.



Anexos

Anexo 1 Historias de usuarios

Historia de usuario	
Número: 1	Usuario: Arístides A. Legrá Lobaina.
Nombre : Autenticación de usuarios	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Daimara Arias Ramírez.	
Descripción: El Administrador debe de ingresar sus datos para necesitar ingresar al sistema (nombre de usuario y contraseña. El sistema guarda los datos.	
Observaciones: Confirmado.	

Tabla 2.3 HU No.1: Autenticar usuarios

Historia de usuario	
Número: 2	Usuario: Arístides A. Legrá Lobaina
Nombre: Gestión de Usuarios.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Daimara Arias Ramírez.	
Descripción: El Administrador debe de insertar los usuarios que van a trabajar con el Sistema, además eliminarlos, modificarlos y mostrar un listado de los mismos. El sistema guarda los datos.	
Observaciones: Confirmado.	

Tabla 2.4 HU No. 2: Gestionar Usuarios



Anexos

Historia de usuario	
Número: 3	Usuario: Arístides Legrá Lobaina.
Nombre: Gestionar Penum.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Daimara Arias Ramírez.	
Descripción: El Administrador debe de insertar, eliminar, mostrar un listado de los mismos, además buscar y visualizar el pensum de una edición. El sistema guarda los datos.	
Observaciones: Confirmado.	

Tabla 2.5 HU No. 3: Gestionar Penum.

Historia de usuario	
Número: 4	Usuario: Arístides A. Legrá Lobaina.
Nombre: Gestionar ediciones vinculantes.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Daimara Arias Ramírez.	
Descripción: El Administrador debe de insertar las ediciones vinculantes de la maestría, además eliminar y mostrar un listado de los mismos. El sistema guarda los datos.	
Observaciones: Confirmado.	

Tabla 2.6 HU No.4: Gestionar ediciones vinculantes.



Anexos

Historia de usuario	
Número: 5	Usuario: Arístides A. Legrá Lobaina.
Nombre: Gestionar Asignaturas.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Daimara Arias Ramírez.	
Descripción: El Administrador debe de insertar las asignaturas de la maestría, además eliminar y mostrar un listado de las mismas. El sistema guarda los datos.	
Observaciones: Confirmado.	

Tabla 2.7 HU No.5: Gestionar Asignaturas.

Historia de usuario	
Número: 6	Usuario: Arístides A. Legrá Lobaina.
Nombre: Gestionar Documentación.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Daimara Arias Ramírez.	
Descripción: El Administrador debe de insertar la documentación de la maestría, además eliminar y mostrar un listado de las mismas. El sistema guarda los datos.	
Observaciones: Confirmado.	

Tabla 2.8 HU No.6: Gestionar Documentación.



Anexos

Historia de usuario	
Número: 7	Usuario: Arístides A. Legrá Lobaina.
Nombre: Solicitar Matrícula en una Edición.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Daimara Arias Ramírez.	
Descripción: El usuario que visite el sitio y desee hacer la maestría podrá solicitar matrícula.	
Observaciones: Confirmado.	

Tabla 2.9 HU No.7: Solicitar Matrícula en una Edición.

Historia de usuario	
Número: 8	Usuario: Arístides A. Legrá Lobaina.
Nombre: Mostrar información de un pensum según la edición.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Daimara Arias Ramírez.	
Descripción: El administrador o cualquier otro usuario buscaran la información de un pensum según la edición.	
Observaciones: Confirmado.	

Tabla 2.10 HU No.8: Mostrar información de un pensum según la edición.



Anexos

Historia de usuario	
Número: 9	Usuario: Arístides A. Legrá Lovaina.
Nombre: Mostrar listado de solicitudes.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Daimara Arias Ramírez.	
Descripción: El coordinador es el encargado de aprobar las solicitudes, estas se mostraran en un listado.	
Observaciones: Confirmado.	

Tabla 2.11 HU No.9: Mostrar listado de solicitudes.

Historia de usuario	
Número: 10	Usuario: Arístides A. Legrá Lovaina.
Nombre: Mostrar información de un estudiante.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Daimara Arias Ramírez.	
Descripción: El administrador o el coordinador buscaran las características de los estudiantes.	
Observaciones: Confirmado.	

Tabla 2.12 HU No.10: Mostrar información de un estudiante.



Anexos

Historia de usuario	
Número: 11	Usuario: Arístides A. Legrá Lovaina.
Nombre: Mostrar información de un profesor.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Daimara Arias Ramírez.	
Descripción: El administrador o el coordinador buscaran las características de un profesor.	
Observaciones: Confirmado.	

Tabla 2.13 HU No.11: Mostrar información de un profesor

Historia de usuario	
Número: 12	Usuario: Arístides A. Legrá Lovaina.
Nombre: Mostrar información de ediciones dada una modalidad.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Daimara Arias Ramírez.	
Descripción: El administrador o el coordinador buscaran las características de una edición dada una modalidad.	
Observaciones: Confirmado.	

Tabla 2.14 HU No.12: Mostrar información de ediciones dada una modalidad.



Anexos

Anexo 2 Tarjetas CRC

Clase Autenticar usuarios	
Responsabilidades	Colaboraciones
Autenticar_usuarios	Conexión

Tabla 2.19: Tarjeta CRC No.1 Autenticar Usuarios.

Clase Gestionar usuarios	
Responsabilidades	Colaboraciones
insertar _ usuario eliminar _ usuario listar _ usuario	Conexión

Tabla 2.20: Tarjeta CRC No.2 Gestionar Usuarios.

Clase Gestionar Penum.	
Responsabilidades	Colaboraciones
Insertar_penum eliminar_penum listar_penum	Conexión

Tabla 2.21: Tarjeta CRC No.3 Gestionar Penum.

Clase Gestionar ediciones vinculantes.	
Responsabilidades	Colaboraciones
Insertar_EV eliminar_EV listar_EV	Conexión

Tabla 2.22: Tarjeta CRC No.4 Gestionar Ediciones Vinculantes (EV)



Anexos

Clase Gestionar Asignatura.	
Responsabilidades	Colaboraciones
Insertar_asig eliminar_asig listar_asig	Conexión

Tabla 2.23: Tarjeta CRC No.5 Gestionar Asignatura (asig).

Clase Gestionar Documentación	
Responsabilidades	Colaboraciones
insertar _ doc eliminar _ doc listar _ doc	Conexión

Tabla 2.24: Tarjeta CRC No.6 Gestionar Documentación (doc).

Clase Solicitar Matrícula en una Edición	
Responsabilidades	Colaboraciones
Solicitar _Matricula en una edición.	Conexión

Tabla 2.25: Tarjeta CRC No.7 Solicitar Matricula en una edición.

Clase Mostrar Información de un Pensum según le Edición	
Responsabilidades	Colaboraciones
Mostrar _ información de un pensum según la edición.	Conexión

Tabla 2.26: Tarjeta CRC No.8 Mostrar información de un pensum según la edición.

Clase Mostrar Listado de Solicitudes.	
Responsabilidades	Colaboraciones
mostrar_ ListSolic	Conexión

Tabla 2.27: Tarjeta CRC No.9 Mostrar Listado de Solicitudes (ListSolic).

Clase Mostrar información de un Estudiante.	
Responsabilidades	Colaboraciones
mostrar_ InfEst	Conexión

Tabla 2.28: Tarjeta CRC No.10 Mostrar información de un Estudiante (InfEst).



Anexos

Clase Mostrar Información de un Profesor.	
Responsabilidades	Colaboraciones
mostrar_ InfProf	Conexión

Tabla 2.29: Tarjeta CRC No.11 Mostrar información de un profesor (InfProf).

Clase Mostrar información de ediciones dada la modalidad.	
Responsabilidades	Colaboraciones
mostrar_ información ediciones dada una modalidad	Conexión

Tabla 2.30: Tarjeta CRC No.12 Mostrar información de ediciones dada la Modalidad.



Anexo 3 Tarjetas de Ingeniería

Tarea ingeniería	
Número tarea: 1	Número historia: 2
Nombre tarea: Insertar Usuario.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea inserta nuevos Usuario que el Administrador estime necesario, los que van a interactuar con el sistema.	

Tabla3.3: Tarjeta de tarea No.1 Insertar Usuario

Tarea ingeniería	
Número tarea: 2	Número historia: 2
Nombre tarea: Eliminar Usuario.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea facilita eliminar los usuarios que ya no van a interactuar con el Sistema.	

Tabla3.4: Tarjeta de tarea No.2: Eliminar Usuario



Anexos

Tarea ingeniería	
Número tarea: 3	Número historia: 2
Nombre tarea: Modificar Usuario.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea facilita modificar la contraseña a los usuarios, ya sea porque se le haya olvidado o cualquier otra situación que se presente.	

Tabla3.5: Tarjeta de tarea No.3: Modificar Usuario

Tarea ingeniería	
Número tarea: 4	Número historia: 2
Nombre tarea: Mostrar listado de usuarios.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea muestra un listado con todos los usuarios que tienen acceso al Sistema son los mismos que los Administradores registran.	

Tabla 3.6: Tarjeta de tarea No.4: Mostrar listado de Usuarios



Anexos

Tarea ingeniería	
Número tarea: 5	Número historia: 3
Nombre tarea: Insertar Penum.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea inserta nuevos pensum que el Administrador estime necesario, para poder trabajar después con ellos.	

Tabla 3.7: Tarjeta de tarea No.5: Insertar Penum.

Tarea ingeniería	
Número tarea: 6	Número historia: 3
Nombre tarea: Eliminar pensum.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea elimina los pensum, que ya no se van a utilizar.	

Tabla 3.8: Tarjeta de tarea No.6: Eliminar Penum.



Anexos

Tarea ingeniería	
Número tarea: 7	Número historia: 3
Nombre tarea: Mostrar listado de pensum.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea muestra un listado de todos los pensum que hay en la Base de Datos para tener una visión de los mismos.	

Tabla 3.9: Tarjeta de tarea No.7: Mostrar listado de Pensum.

Tarea ingeniería	
Número tarea: 9	Número historia: 4
Nombre tarea: Insertar ediciones vinculantes.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea inserta nuevas ediciones que el Administrador estime necesario, para poder trabajar después con ellas.	

Tabla 3.10: Tarjeta de tarea No.9: Insertar ediciones vinculantes.

Tarea ingeniería	
Número tarea: 10	Número historia: 4
Nombre tarea: Eliminar ediciones vinculantes.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea elimina las ediciones vinculantes que ya no se vayan a utilizar.	

Tabla 3.11: Tarjeta de tarea No.10: Eliminar ediciones vinculantes.



Anexos

Tarea ingeniería	
Número tarea: 11	Número historia: 4
Nombre tarea: Mostrar listado de ediciones vinculantes.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea muestra un listado de todas de las ediciones vinculantes que hay en la Base de Datos para tener una visión de las mismas.	

Tabla 3.12: Tarjeta de tarea No.11: Mostrar listado de ediciones vinculantes.

3.3.3-Iteración No.2

Historias de usuario	Tiempo de estimación (semanas)
Gestionar Asignatura	1
Gestionar Documentación	1
Solicitar matricula en una edición	2
Mostrar información de un pensum según la modalidad	1

Tabla 3.13: Historias de usuario abordadas en la segunda iteración.

Tarea ingeniería	
Número tarea: 12	Número historia: 5
Nombre tarea: Insertar asignatura.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea inserta nuevas asignaturas que el Administrador estime necesario, para poder trabajar después con ellas.	

Tabla 3.14: Tarjeta de tarea No.12: Insertar asignatura.



Anexos

Tarea ingeniería	
Número tarea: 13	Número historia: 5
Nombre tarea: Eliminar Asignatura.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea elimina las asignaturas que ya no se vayan a utilizar.	

Tabla 3.15: Tarjeta de tarea No.13: Eliminar Asignatura.

Tarea ingeniería	
Número tarea: 14	Número historia: 5
Nombre tarea: Mostrar listado de asignatura.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea muestra un listado de todas las asignaturas, que se encuentran en la base de datos.	

Tabla 3.16: Tarjeta de tarea No.14: Mostrar listado de Asignatura.

Tarea ingeniería	
Número tarea: 15	Número historia: 6
Nombre tarea: Insertar Documentación	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea inserta nuevas documentaciones que el Administrador estime necesario, para que los estudiantes puedan trabajar con ellas.	

Tabla 3.17: Tarjeta de tarea No.15: Insertar Documentación.



Anexos

Tarea ingeniería	
Número tarea: 16	Número historia: 6
Nombre tarea: Eliminar Documentación.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea elimina toda la documentación que ya no se vayan a utilizar.	

Tabla 3.18: Tarjeta de tarea No.16: Eliminar Documentación.

Tarea ingeniería	
Número tarea: 17	Número historia: 6
Nombre tarea: Mostrar listado de documentación.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea muestra un listado de toda la documentación, que se encuentra en la base de datos.	

Tabla 3.19: Tarjeta de tarea No.17: Mostrar listado de documentación.

Tarea ingeniería	
Número tarea: 17	Número historia: 7
Nombre tarea: Solicitar matrícula en una edición.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea será realizada por las personas que deseen hacer la maestría de electromecánica, esta solicitud se encuentra en la base de datos.	

Tabla 3.20: Tarjeta de tarea No.18: Solicitar matrícula en una edición.



Anexos

Tarea ingeniería	
Número tarea: 19	Número historia: 8
Nombre tarea: Mostrar información de un pensum según la edición.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea muestra la información de todos los pensum, que se encuentra en la base de datos.	

Tabla 3.21: Tarjeta de tarea No.19: Mostrar información de un pensum según la edición.

3.3.4-Iteración No.3

Historias de usuario	Tiempo de estimación (semanas)
Mostrar listado de solicitudes	2
Mostrar información de un estudiante.	1
Mostrar información de un profesor.	1
Mostrar información de ediciones dada una modalidad.	2

Tabla 3.22: Historias de usuario abordadas en la tercera iteración.

Tarea ingeniería	
Número tarea: 20	Número historia: 9
Nombre tarea: Mostrar listado de solicitudes.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea muestra un listado de todas las solicitudes que se encuentran en la base de datos.	

Tabla 3.23: Tarjeta de tarea No.20: Mostrar Listado de Solicitudes.



Anexos

Tarea ingeniería	
Número tarea: 21	Número historia: 10
Nombre tarea: Mostrar información de un estudiantes.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea muestra un listado de los estudiantes de la maestría que se han insertado en esta tarea.	

Tabla 3.24: Tarjeta de tarea No.21: Mostrar información de un Estudiante.

Tarea ingeniería	
Número tarea: 22	Número historia: 11
Nombre tarea: Mostrar .información de un profesor	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea muestra la información de los profesores de la maestría que hay en la base de datos para tener una visión de los mismos.	

Tabla 3.25: Tarjeta de tarea No.22: Mostrar información de un profesor.

Tarea ingeniería	
Número tarea: 23	Número historia: 12
Nombre tarea: Mostrar información de ediciones dada una modalidad.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio:	Fecha fin:
Programador responsable: Daimara Arias Ramírez.	
Descripción: Esta tarea muestra la información de las ediciones dada la modalidad, esta tarea la realizan todos los usuarios. .	

Tabla 3.26: Tarjeta de tarea No.23: Mostrar información de ediciones dada una modalidad.



Anexo 4 Pruebas de Aceptación

Prueba de aceptación
HU: Autenticar usuarios
Nombre: Prueba para comprobar la Autenticación de usuarios. (Nombre de usuario y contraseña).
Descripción: Validación de entrada de los datos de los usuarios.
Condiciones de ejecución: El usuario debe introducir su nombre de usuario y contraseña.
Entrada/Pasos ejecución: El usuario escribe su nombre de usuario y contraseña y luego da clic en el botón Entrar.
Resultado: <ul style="list-style-type: none">• Si el usuario tiene acceso para entrar a la aplicación e inserta sus datos correctamente entrará sin problemas al Sistema. <p>Se emite un mensaje de error en caso de que:</p> <ul style="list-style-type: none">• Se inserte los datos de un usuario no válido para el Sistema o incorrectos. (Ya sea su nombre de usuario o su contraseña).• Se dé clic en el botón Iniciar sin insertar nada en los campos de texto.
Evaluación de la prueba: Aceptada.

Tabla 3.28: Pruebas de aceptación para la HU: Autenticar usuarios

Prueba de aceptación
HU: Gestión de usuarios
Nombre: Prueba para verificar la Gestión de usuarios.
Descripción: Validación de la gestión de usuarios.
Condiciones de ejecución: El usuario debe de entrar a la Aplicación y el mismo tiene que ser Administrador de esta para poder insertar usuarios, eliminarlos, cambiarles su contraseña sin necesidad de pedir su contraseña actual, así como listar todos los usuarios con acceso al Sitio.
Entrada/Pasos ejecución: El administrador escribe todos los datos que se le piden para poder insertar un usuario determinado asignándole el nivel que se desee, luego presiona el botón Insertar. Después que el usuario este insertado es que se le puede cambiar su contraseña, eliminarlo y ver sus datos en el listado de usuarios.
Resultado: <p>Se emite un mensaje de error en caso de que:</p> <ul style="list-style-type: none">• Falten datos del usuario a la hora de insertarlo.• Se inserte un usuario que ya exista, es decir que tenga el mismo usuario.• Se inserte un usuario que tenga el mismo número de carnet de identidad.• Cuando se va a eliminar el usuario admin, ya que este está definido por el Sistema.• Se dé clic en botón eliminar sin antes seleccionar ningún usuario.
Evaluación de la prueba: Aceptada.

Tabla 3.29: Pruebas de aceptación para la HU: Gestión de usuarios



Anexos

Prueba de aceptación
HU: Gestión de pensum
Nombre: Prueba para verificar la Gestión de Pensum
Descripción: Validación de la gestión de pensum.
Condiciones de ejecución: El usuario debe de entrar a la Aplicación y el mismo tiene que ser Administrador de esta para poder insertar pensum, eliminarlos, así como listar todos los pensum.
Entrada/Pasos ejecución: El administrador escribe todos los datos que se le piden para poder insertar un pensum determinado, luego presiona el botón Insertar. Después que el pensum este insertado es que puede ser eliminado y ver su información.
Resultado: Se emite un mensaje de error en caso de que: <ul style="list-style-type: none">• Falten datos del pensum a la hora de insertarlo.• Se dé clic en botón eliminar sin antes seleccionar ningún pensum.
Evaluación de la prueba: Aceptada.

Tabla 3.30: Pruebas de aceptación para la HU: Gestión de Pensum

Prueba de aceptación
HU: Gestión de ediciones vinculantes
Nombre: Prueba para verificar la Gestión de ediciones vinculantes.
Descripción: Validación de la gestión de ediciones vinculantes.
Condiciones de ejecución: El usuario debe de entrar a la Aplicación y el mismo tiene que ser Administrador de esta para poder insertar Ediciones Vinculantes, eliminarlos, así como listar todos los Ediciones Vinculantes.
Entrada/Pasos ejecución: El administrador escribe todos los datos que se le piden para poder insertar un Ediciones Vinculantes determinado, luego presiona el botón Insertar. Después que el Ediciones Vinculantes este insertado es que puede ser eliminado y ver su información.
Resultado: Se emite un mensaje de error en caso de que: <ul style="list-style-type: none">• Falten datos del Ediciones Vinculantes a la hora de insertarlo.• Se dé clic en botón eliminar sin antes seleccionar ningún Ediciones Vinculantes.
Evaluación de la prueba: Aceptada.

Tabla 3.31: Pruebas de aceptación para la HU: Gestión de Ediciones Vinculantes



Anexos

Prueba de aceptación
HU: Gestión de Asignaturas
Nombre: Prueba para verificar la Gestión de Asignaturas.
Descripción: Validación de la gestión de asignaturas.
Condiciones de ejecución: El usuario debe de entrar a la Aplicación y el mismo tiene que ser Administrador de esta para poder insertar Asignaturas, eliminarlos, así como listar todos los Asignaturas.
Entrada/Pasos ejecución: El administrador escribe todos los datos que se le piden para poder insertar una Asignatura determinada, luego presiona el botón Insertar. Después que la Asignatura esta insertada es que puede ser eliminada y ver su información.
Resultado: Se emite un mensaje de error en caso de que: <ul style="list-style-type: none">• Falten datos del Ediciones Vinculantes a la hora de insertarlo.• Se dé clic en botón eliminar sin antes seleccionar ninguna Ediciones Vinculantes.
Evaluación de la prueba: Aceptada.

Tabla 3.32: Pruebas de aceptación para la HU: Gestión de Asignaturas

Prueba de aceptación
HU: Gestión de Documentación
Nombre: Prueba para verificar la Gestión de Documentación.
Descripción: Validación de la gestión de documentación.
Condiciones de ejecución: El usuario debe de entrar a la Aplicación y el mismo tiene que ser Administrador de esta para poder insertar la Documentación, eliminarlas, así como listar todas las Documentaciones.
Entrada/Pasos ejecución: El administrador escribe todos los datos que se le piden para poder insertar una Documentación determinada, luego presiona el botón Insertar. Después que la Documentación esta insertada es que puede ser eliminada y ver su información.
Resultado: Se emite un mensaje de error en caso de que: <ul style="list-style-type: none">• Falten datos de la Documentación a la hora de insertarla.• Se dé clic en botón eliminar sin antes seleccionar ningún Documento.
Evaluación de la prueba: Aceptada.

Tabla 3.33 Pruebas de aceptación para la HU: Gestión de Documentación

Prueba de aceptación
HU: Solicitar matricula en una edición.
Nombre: Prueba para Solicitar matricula en una edición.
Descripción: Validación para mostrar las solicitudes de matriculas en una edición.
Condiciones de ejecución: Todos los usuarios del sistema y visitantes al sitio tienen los privilegios para solicitar matricula en una edición.
Entrada/Pasos ejecución: Todos los usuarios del sistema y visitantes al sitio, en la página principal del sitio pueden dirigirse al menú de navegación, a la pestaña Solicitar matricula en una edición, y damos clic. Ahí podemos hacer las solicitudes.
Resultado: <ul style="list-style-type: none">• Se mostrará una interfaz que permitirá hacer la solicitud.
Evaluación de la prueba: Aceptada.

Tabla 3.34: Pruebas de aceptación para la HU: Solicitar matricula en una edición.



Anexos

Prueba de aceptación
HU: Mostrar información de un pensum según la edición.
Nombre: Prueba para mostrar información de un pensum según la edición.
Descripción: Validación para mostrar la información de un pensum según la edición.
Condiciones de ejecución: Todos los usuarios del sistema tienen los privilegios para poder mostrar la información de un pensum según la edición.
Entrada/Pasos ejecución: Todos los usuarios del sistema y visitantes al sitio, en la página principal del sitio pueden dirigirse al menú de navegación, a la pestaña Información de un pensum según la edición, y damos clic. Ahí veremos la información de los pensum.
Resultado: <ul style="list-style-type: none">• Se mostró la información de un pensum según la edición.
Evaluación de la prueba: Aceptada.

Tabla 3.35: Pruebas de aceptación para la HU: Mostrar información de un pensum según la edición.

Prueba de aceptación
HU: Mostrar listado de solicitudes.
Nombre: Prueba para mostrar el listado de las solicitudes
Descripción: Validación para mostrar el listado de las solicitudes.
Condiciones de ejecución: El usuario tiene que tener privilegios de ser Coordinador para poder ver el listado de las solicitudes.
Entrada/Pasos ejecución: El Coordinador en su menú de navegación, se dirige a la pestaña Listado de Solicitudes y al dar clic, se observa una tabla con todas las Solicitudes que han realizado y no se han aprobado.
Resultado: <ul style="list-style-type: none">• Se mostraron todas las Solicitudes no aprobadas.
Evaluación de la prueba: Aceptada.

Tabla 3.36: Pruebas de aceptación para la HU: Mostrar listado de Solicitudes.

Prueba de aceptación
HU: Mostrar listado de estudiantes
Nombre: Prueba para verificar si se muestra la información de los estudiantes
Descripción: Validación para mostrar información de un estudiante.
Condiciones de ejecución: Todos los usuarios del sistema tienen los privilegios para poder mostrar la información de un estudiante.
Entrada/Pasos ejecución: Todos los usuarios del sistema y visitantes al sitio, en la página principal del sitio pueden dirigirse al menú de navegación, a la pestaña Información de un estudiante y damos clic. Ahí veremos las características del estudiante seleccionado.
Resultado: <ul style="list-style-type: none">• Después de seleccionar al estudiante veremos sus características.
Evaluación de la prueba: Aceptada.

Tabla 3.37: Pruebas de aceptación para la HU: Mostrar Información de un Estudiante.



Prueba de aceptación
HU: Mostrar información de un profesor.
Nombre: Prueba para verificar si se muestra la información de un profesor.
Descripción: Validación para mostrar información de un profesor.
Condiciones de ejecución: Todos los usuarios del sistema tienen los privilegios para poder mostrar información de un profesor.
Entrada/Pasos ejecución: Todos los usuarios del sistema y visitantes al sitio, en la página principal del sitio pueden dirigirse al menú de navegación, a la pestaña Información de un profesor y damos clic. Ahí veremos las características del profesor seleccionado.
Resultado: <ul style="list-style-type: none">• Después de seleccionar al profesor veremos sus características.
Evaluación de la prueba: Aceptada.

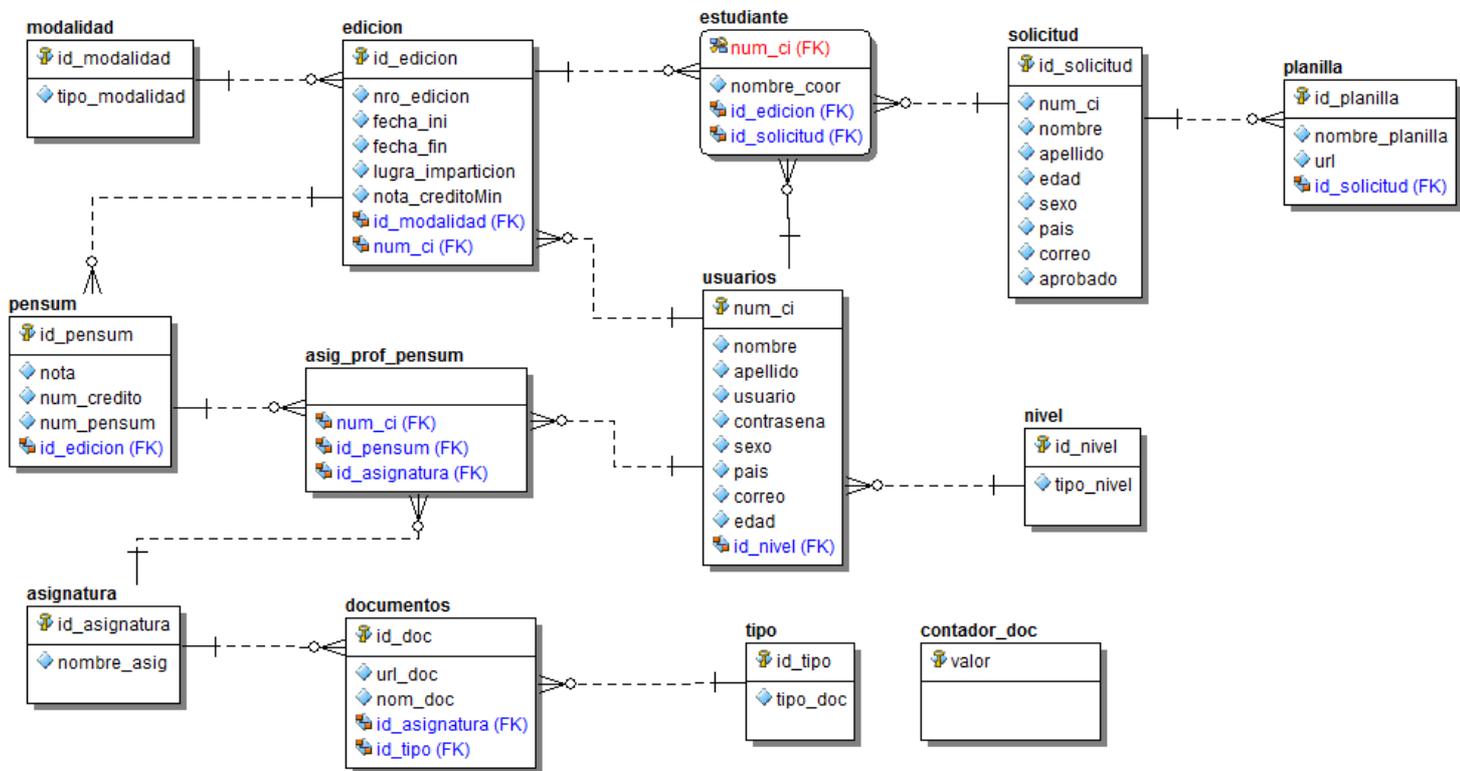
Tabla 3.38: Pruebas de aceptación para la HU: Mostrar información de un profesor

Prueba de aceptación
HU: Mostrar información de ediciones dada una modalidad.
Nombre: Prueba para verificar si se muestra la información de ediciones dada la modalidad.
Descripción: Validación para mostrar información de ediciones dada una modalidad.
Condiciones de ejecución: Todos los usuarios del sistema tienen los privilegios para poder mostrar la información de ediciones dada una modalidad.
Entrada/Pasos ejecución: Todos los usuarios del sistema y visitantes al sitio, en la página principal del sitio pueden dirigirse al menú de navegación, a la pestaña Información de una edición dada una modalidad y damos clic. Ahí veremos las características de una edición seleccionada.
Resultado: <ul style="list-style-type: none">• Después de seleccionar la modalidad veremos las características de la edición.
Evaluación de la prueba: Aceptada.

Tabla 3.39: Pruebas de aceptación para la HU: Mostrar información de ediciones dada una Modalidad.



Anexo 5 Modelo de la Base de Datos





Anexo 6 Principal Interfaz de la Aplicación.

Maestría Electromecánica

Instituto Superior
Metalfúgico de Moa
"Dr. C. Antonio Nieves Jiménez"

Inicio Misión Visión Acerca del Sistema Términos y Condiciones

DESCRIPCIÓN GENERAL DEL PROGRAMA DE MAESTRÍA

El Programa de maestría en ELECTROMECÁNICA aprobado por la COPEP con fecha 17 de Octubre del 2005, tiene antecedentes en dos ediciones de la maestría donde se graduaron más de 20 master en Electromecánica. También han sido impartidos, por los Departamentos de Ingeniería Mecánica y Eléctrica, numerosos cursos y entrenamientos de postgrado y diplomados.

La tercera edición comenzó en el mes de Febrero del 2006 luego de la resolución Ministerial No. 62/2006 que autorizó el inicio de la misma. La cuarta edición se desarrolló en colaboración con la República Bolivariana de Venezuela en el Instituto Universitario Tecnológico (IUT) "Dr. Federico Rivero Palacios" a partir del mes de Mayo del 2007.

En la tercera edición se recibieron 70 solicitudes de 15 entidades del territorio nororiental cubano, de ellas se aprobaron 58, los estudiantes se organizaron en tres grupos incluyendo la tercera edición del Programa Doctoral Electromecánica con salida intermedia de maestría para aquellos que lo solicitaron. De los matriculados 22 vencieron todos los créditos y 29 terminaron con éxito la defensa de la tesis en el período de validez de los créditos para un 90,6%.

En la cuarta edición se recibieron 22 solicitudes de los IUT "Dr. Federico Rivero Palacios" y "Mariscal Sucre", de ellas se aprobaron 22. De los matriculados 18 vencieron todos los créditos y 15 terminaron con éxito la defensa de la tesis en el período de validez de los créditos para un 82,2%.

Entre los principales resultados del Programa se pueden señalar: el amplio espectro de profesionales de diferentes carreras presentes en su matrícula; la estabilidad en la eficiencia de sus dos últimas graduaciones, con más del 82 % de tesis defendidas en cada edición. Los resultados de investigación derivados del Programa han estado relacionados con necesidades actuales y perspectivas del desarrollo socioeconómico local, regional y algunos han tenido impacto nacional e internacional, el incremento del nivel científico del claustro, el elevado número de doctores asociados al Programa y el nivel de cooperación internacional en esta área del conocimiento.

[VER MÁS >>](#)

Usuario:
Contraseña:

Menu

- Solicitar matrícula en una edición
- Información de un alumno
- Información de un profesor
- Información de un peresum según la edición
- Información de una edición según la modalidad
- Información de una asignatura
- Enlaces
- Ceslam
- Intranet ISMM
- Intranet del Mes
- Descargas
- Convocatoria

Interfaz de la página principal.



Maestría Electromecánica



Instituto Superior
Minero Metalúrgico de Moa
"Dr. C. Antonio Núñez Jiménez"

[Inicio](#) [Misión](#) [Visión](#) [Acerca del Sistema](#) [Términos y Condiciones](#) [Cerrar Sesión](#)

Bienvenido Administrador

El Programa de maestría en ELECTROMECAÁNICA aprobado por la COPEP con fecha 17 de Octubre del 2005, tiene antecedentes en dos ediciones de la maestría donde se graduaron más de 20 master en Electromecánica. También han sido impartidos, por los Departamentos de Ingeniería Mecánica y Eléctrica, numerosos cursos y entrenamientos de postgrado y diplomados

La tercera edición comenzó en el mes de Febrero del 2006 luego de la resolución Ministerial No. 62/2006 que autorizó el inicio de la misma. La cuarta edición se desarrolló en colaboración con la República Bolivariana de Venezuela en el Instituto Universitario Tecnológico (IUT) "Dr. Federico Rivero Palacios" a partir del mes de Mayo del 2007.

En la tercera edición se recibieron 70 solicitudes de 15 entidades del territorio nororiental cubano, de ellas se aprobaron 58, los estudiantes se organizaron en tres grupos incluyendo la tercera edición del Programa Doctoral Electromecánica con salida intermedia de maestría para aquellos que lo solicitaron. De los matriculados 32 vencieron todos los créditos y 29 terminaron con éxito la defensa de la tesis en el período de validez de los créditos para un 90,6%.

En la cuarta edición se recibieron 22 solicitudes de los IUT "Dr. Federico Rivero Palacios" y "Mariscal Sucre", de ellas se aprobaron 22. De los matriculados 18 vencieron todos los créditos y 15 terminaron con éxito la defensa de la tesis en el período de validez de los créditos para un 83,3%.

Entre los principales resultados del Programa se pueden señalar: el amplio espectro de profesionales de diferentes carreras presentes en su matrícula; la estabilidad en la eficiencia de sus dos últimas graduaciones, con más del 83 % de tesis defendidas en cada edición. Los resultados de investigación derivados del Programa han estado relacionados con necesidades actuales y perspectivas del desarrollo socioeconómico local, regional y algunos han tenido impacto nacional e internacional, el incremento del nivel científico del claustro, el elevado número de doctores asociados al Programa y el nivel de cooperación internacional en esta área del conocimiento.

Menu

- Gestionar Usuarios
- Gestionar Ediciones
- Gestionar Pensum
- Gestionar Asignaturas
- Información de un alumno
- Información de un profesor
- Información de un pensum según la edición
- Información de una edición según la modalidad
- Información de una asignatura
- Gestionar Contraseña

Páginas del ISMM

- Ceetam
- Intranet ISMM
- Intranet del Mes

Interfaz principal del Administrador.



Maestría Electromecánica



[Inicio](#) [Misión](#) [Visión](#) [Acerca del Sistema](#) [Términos y Condiciones](#) [Cerrar Sesión](#)

Bienvenido Coordinador

El Programa de maestría en ELECTROMECÁNICA aprobado por la COPEP con fecha 17 de Octubre del 2005, tiene antecedentes en dos ediciones de la maestría donde se graduaron más de 20 master en Electromecánica. También han sido impartidos, por los Departamentos de Ingeniería Mecánica y Eléctrica, numerosos cursos y entrenamientos de postgrado y diplomados. La tercera edición comenzó en el mes de Febrero del 2006 luego de la resolución Ministerial No. 62/2006 que autorizó el inicio de la misma. La cuarta edición se desarrolló en colaboración con la República Bolivariana de Venezuela en el Instituto Universitario Tecnológico (IUT) "Dr. Federico Rivero Palacio" a partir del mes de Mayo del 2007.

En la tercera edición se recibieron 70 solicitudes de 15 entidades del territorio nororiental cubano, de ellas se aprobaron 58, los estudiantes se organizaron en tres grupos incluyendo la tercera edición del Programa Doctoral Electromecánica con salida intermedia de maestría para aquellos que lo solicitaron. De los matriculados 22 vencieron todos los créditos y 29 terminaron con éxito la defensa de la tesis en el período de validez de los créditos para un 90,0%.

En la cuarta edición se recibieron 22 solicitudes de los IUT "Dr. Federico Rivero Palacio" y "Mariscal Sucre", de ellas se aprobaron 22. De los matriculados 18 vencieron todos los créditos y 15 terminaron con éxito la defensa de la tesis en el período de validez de los créditos para un 82,2%.

Entre los principales resultados del Programa se pueden señalar: el amplio espectro de profesionales de diferentes carreras presentes en su matrícula; la estabilidad en la eficiencia de sus dos últimas graduaciones, con más del 82 % de tesis defendidas en cada edición. Los resultados de investigación derivados del Programa han estado relacionados con necesidades actuales y perspectivas del desarrollo socioeconómico local, regional y algunos han tenido impacto nacional e internacional, el incremento del nivel científico del claustro, el elevado número de doctores asociados al Programa y el nivel de cooperación internacional en esta área del conocimiento.

© Sistema de Maestría Electromecánica

Menu

- Listado de Solicitudes
- Información de un alumno
- Información de un profesor
- Información de un pensum según la edición
- Información de una edición según la modalidad
- Información de una asignatura
- Cambiar Contraseña

Páginas del ISMM

- Ceetam
- Intranet ISMM
- Intranet del Mes

Interfaz principal del Coordinador.