



INSTITUTO SUPERIOR MINERO METALÚRGICO

“Dr. Antonio Núñez Jiménez”.

Facultad de Metalurgia - Electromecánica

Trabajo de Tesis

En Opción al Título

Ingeniero en Informática

**SISTEMA DE GESTIÓN ENERGÉTICA ASISTIDA PARA EL
ISMM-MÓDULO DE PRONÓSTICO DEL CONSUMO DE
RECURSOS ENERGÉTICO.**

AUTOR: Yanquiel Borges Matos

TUTOR: Ing. José Rolando Pérez Sandó

CONSULTANTE: Dr. Arístides Alejandro Legrá Lobaina

Moa, Holguín, Cuba

“Año del 53 Aniversario del Triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa para que haga uso de este como estime pertinente.

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

Yanquiel Borges Matos

Nombre completo del Autor

Ing. José Rolando Pérez Sandó

Nombre completo del Tutor

PENSAMIENTO:

Con sabiduría se edifica la casa, con prudencia se afirma y con ciencia se llenan las cámaras de todo bien preciado y agradable.

Proverbios 24.3-4;

DEDICATORIA

*Este trabajo va dedicado especialmente a mis Padres,
por la magnífica educación que me dieron,
por apoyarme en todas mis decisiones,
y porque sé que este es su mayor anhelo.*

*A mi madre, Miriam,
por demostrarme que en la vida no debemos dejar que nos venzan las dificultades y
haberme dado ánimo para seguirme superando
y lograr que este sueño sea posible.*

*A mi padre, Gustavo,
porque me inculcó los deseos de superación y me enseñó a enfrentar la vida con mis
propias decisiones.*

*A mi maravillosa Ele,
por ser mi compañera y amiga en esta larga travesía.*

*A mi hermano Daniel,
por ser el menor y el alma de la casa,
que en su vida pueda lograr empeños mayores.*

AGRADECIMIENTOS

A Dios, por iluminar mi camino y guiar mis pasos.

Por darme la oportunidad de realizar este sueño,

Gracias mi Señor.

A mis padres queridos, porque gracias a ellos estoy aquí, por sus enseñanzas, por sus consejos, por su voluntad, esfuerzo y empeño en mi educación y sobre todo gracias por darme la vida.

A mi querida Ele, por ser mi guía, mi compañera, mi amiga, mi confidente, mi amante, mi luz, mi estrella, por ser la mujer que un día soñé,

y sobre todo por amarme,

Gracias mi Amor.

A mi familia en general por darme siempre apoyo cuando lo necesité.

A la familia de Eylén por acogerme en su casa y darme su apoyo.

A mi prima Iduviza por estar siempre atenta y dispuesta a ayudarme.

A mi colega y amigo Cala, al que le estaré eternamente agradecido por su gran ayuda.

*A mis compañeros de grupo, que sin duda son los mejores que he tenido
en especial a Reidelbis, Roberto y Migue .*

A los profesores que cada día impartieron sus conocimientos para nuestra superación.

A Pikirí, por su tutoría ejemplar.

Al Mello por su ayuda desinteresada.

Al Dr. Legrá por darme la oportunidad de participar en este proyecto.

A todos los que de una forma u otra aportaron su granito de arena en la realización de este trabajo.

GRACIAS

RESUMEN

Nuestro país se encuentra enfrascado en lograr un mejoramiento gradual en todas las esferas de la economía. Un elemento fundamental para lograr este mejoramiento, es el ahorro eficiente de los recursos energéticos. El aumento de los precios de la energía, la necesidad de prestar mejores servicios, así como la imperiosa necesidad de protección del medio ambiente, son factores que impulsan actualmente la crecida de la eficiencia energética. Por tal motivo, en el presente Trabajo de Diploma se desarrolla la implementación de un módulo como parte de del sistema GEA (Gestión Energética Asistida) que brinde información sobre el pronóstico del consumo de los recursos energéticos que le son asignados al ISMMM (Instituto Superior Minero Metalúrgico de Moa), que favorezca el proceso de planificación de dichos recursos. El uso de esta aplicación facilitará la toma de decisiones relacionadas con los pedidos de portadores energéticos, necesarios para la sostenibilidad del instituto en un periodo de tiempo determinado. Esto permite un mejor conocimiento de lo que se ha consumido y de lo que está por consumirse, reduciendo en gran medida el gasto innecesario de estos portadores. Además se generan reportes actualizados como facilitadores en los pedidos.

SUMMARY

Our country is buried in achieving a gradual improvement in all the spheres of the economy. A fundamental element to achieve this improvement, is the efficient saving of the energy resources. The increase of the prices of the energy, the necessity to lend better services, as well as the imperious necessity of protection of the environment, they are factors that impel the grown of the energy efficiency at the moment. For such a reason, presently Work of Diploma is developed the implementation of a module like part of of the system GEA (Attended Energy Administration) that offers information on the presage of the consumption of the energy resources that you/they are assigned to the ISMMM (Institute Superior Mining Metallurgist of Moa) that favors the process of planning of this resources. The use of this application will facilitate the taking of decisions related with the orders of energy payees, necessary for the sustainable of the institute in a period of certain time. This allows a better knowledge of what has wasted away and of what is to waste away, reducing in great measure the unnecessary expense of these payees. Up-to-date reports are also generated as facilitators in the orders.

Índice

INTRODUCCIÓN	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	6
1.1. ESTADO DEL ARTE	6
1.1.1. Términos fundamentales	6
1.1.1.1. Pronóstico	6
1.1.1.2. Clasificación de los pronósticos con respecto al tiempo que abarcan	6
1.1.1.3. Métodos estadísticos para el pronóstico	7
1.1.2. Antecedentes	8
1.1.2.1. Ámbito Nacional	8
1.1.2.2. Ámbito Internacional	8
1.1.2.3. Pronóstico en el ISMMM	9
1.2. TENDENCIAS Y TECNOLOGÍAS ACTUALES	10
1.2.1. Internet	10
1.2.2. Aplicaciones WEB. Ventajas y Desventajas	10
1.2.3. Servidores WEB	13
1.2.4. Lenguajes de Programación WEB	15
1.2.5. Entornos de programación Web	16
1.2.6. Sistemas Gestores de Base de Datos (SGBD)	17
1.3. ARQUITECTURA	22
1.3.1. Patrones Arquitectónicos	22
1.3.2. Modelo Cliente-Servidor	26
1.4. METODOLOGÍA DE DESARROLLO DE SOFTWARE	27
1.4.1. Metodologías Ágiles	28
1.4.2. Metodología XP	29
1.4.2.1. Cuatro valores que promueven la metodología XP:	29
1.4.2.2. Roles XP	29
1.4.2.3. Proceso XP	30
1.4.2.4. Prácticas XP	30
1.4.2.5. Ventajas de XP	32
1.4.2.6. Desventajas de XP	33
1.4.2.7. Fases de la metodología XP	33
1.5. HERRAMIENTAS CASE (COMPUTER – AIDED SOFTWARE ENGINEERING)	35
1.6. LENGUAJE UNIFICADO DE MODELADO (UML)	36
1.7. TECNOLOGÍAS Y HERRAMIENTAS A UTILIZAR	37
CAPÍTULO 2 PLANEACIÓN Y DISEÑO	39
2.1. PERSONAL RELACIONADO CON EL SISTEMA	39

2.2.	LISTA DE RESERVA.....	39
2.3.	HISTORIAS DE USUARIO.....	40
2.4.	PLANIFICACIÓN DE ENTREGAS.....	42
2.4.1.	Estimación de Esfuerzo por Historias de Usuario.....	42
2.4.2.	Planificación de Iteraciones.....	43
2.4.3.	Plan de duración de las Iteraciones.....	44
CAPÍTULO 3 DESARROLLO Y PRUEBAS.....		45
3.1.	MODELO DE DATOS.....	45
3.2.	CLASES, RESPONSABILIDADES Y COLABORADORES.....	46
3.3.	DESARROLLO DE LAS ITERACIONES.....	46
3.3.1.	Tareas por Historias de Usuario.....	47
3.4.	PRUEBAS.....	48
3.4.1.	Desarrollo dirigido por Pruebas.....	49
3.4.2.	Pruebas de Aceptación.....	49
CAPÍTULO 4 ESTUDIO DE FACTIBILIDAD.....		51
4.1.	EVALUACIÓN COSTO-BENEFICIO.....	51
4.2.	EFFECTOS ECONÓMICOS.....	52
4.3.	ELEMENTOS PARA IDENTIFICAR LOS COSTOS Y BENEFICIOS DEL PROYECTO.....	53
4.4.	FACTIBILIDAD ECONÓMICA.....	54
4.5.	EVALUACIÓN ECONÓMICA.....	54
4.5.1.	Beneficios tangibles.....	56
4.5.2.	Beneficios y Costos Intangibles en el proyecto.....	56
CONCLUSIONES.....		57
RECOMENDACIONES.....		58
BIBLIOGRAFÍA.....		59
GLOSARIO DE TÉRMINOS.....		63
ANEXOS.....		64
	ANEXO 1 MINIMOS CUADRADOS GENERALIZADOS.....	64
	ANEXO 2 HISTORIAS DE USUARIOS.....	70
	ANEXO 3 TARJETAS CRC.....	72
	ANEXO 4 TARJETAS DE IGENIERÍA.....	77
	ANEXO 5 PRUEBAS DE ACEPTACIÓN.....	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. Ninguna empresa o entidad podría tener éxito si no es capaz de adaptarse a los diferentes cambios aprovechando los beneficios de esta nueva era, donde la economía global depende cada vez más de sistemas automatizados que en épocas pasadas.

Los actuales intentos del país de prosperar a partir de bases propias de desarrollo, a fin de lograr una paulatina recuperación de su economía, hacen imprescindible que sean adoptadas tecnologías que se encuentran adecuadas a las necesidades de equilibrio con el estado científico técnico del mundo contemporáneo.

El ISMMM (Instituto Superior Minero Metalúrgico de Moa), como entidad enmarcada en el contexto actual del país, en lo que se refiere al desarrollo tanto social como económico, tiene dentro de sus principales metas, la de lograr una mejor utilización de las actuales tecnologías en función de un uso racional de los recursos energéticos que le son asignados.

El CEETAM (Centro de Estudios de Energía y Tecnología Avanzada en Moa), enmarcado en el ISMMM, ha declarado dentro del sistema de objetivos del curso 2010-2011 su compromiso con el trabajo en busca de la eficiencia energética. El Área de resultado Clave No 3: Perfeccionamiento de la Gestión Económica y Financiera; en su Lineamiento 6 plantea: "Los planes de ahorro aseguran el uso eficiente de los portadores energéticos" y el primer criterio de medida establece que "La aplicación del Sistema Informatizado de Gestión Energética Asistida (GEA) en el ISMM, permite lograr ahorros en la planificación y ejecución de cada uno de los portadores energéticos". De este enfoque se derivó la acción: "El CEETAM propondrá una implementación informática viable y eficaz del Sistema de Gestión Energética Asistida a la dirección económica del ISMMM, con el fin de controlar el uso de los portadores energéticos del ISMMM y contribuir a que el Comité de Energía del ISMMM mejore su eficacia en este objetivo"

Como parte de este sistema, y basándonos en investigaciones recientes, se ha establecido la necesidad de desarrollar un módulo que permita brindar información sobre el pronóstico de la demanda del consumo de estos recursos, en intervalos de tiempos, mediante el uso de métodos estadísticos.

Con la realización de este módulo de pronóstico energético, se podrá brindar una información detallada de los posibles parámetros de consumo que se obtendrán en períodos de tiempo próximos al actual. Agilizando en gran medida el proceso de planificación, como parte del flujo de datos en el ciclo del sistema. El uso de esta aplicación, facilitará la toma de decisiones relacionadas con los pedidos de portadores energéticos, necesarios para la sostenibilidad del instituto en un periodo de tiempo determinado.

La importancia de la predicción del consumo energético, no solo radica en su relación con la disponibilidad de combustibles primarios y su financiamiento, sino también en los siguientes factores de carácter técnico económico:

- Conocer con anticipación los futuros gastos financieros por concepto de adquisición de combustibles para las actividades de transportación.
- Prever futuros cambios en las redes eléctricas y líneas de transmisión.
- Necesidad de aplicar nuevas medidas para el ahorro de energía.

En el marco reducido de una institución universitaria, el conocimiento previo de los consumos futuros de energía, se convierte en una herramienta fundamental para la ejecución planificada del control que deben ejercer los sistemas de gestión energética. Al conocer cuánto se va a consumir y cómo será el comportamiento de las principales variables universitarias que inciden en el consumo energético, se crean las condiciones necesarias para prever el cambio de los indicadores de consumo, lo que posibilita la toma de decisiones adecuadas dentro del marco de un sistema de gestión de consumo energético.

Por lo que se identifica como **Problema Científico** a resolver: La necesidad de un módulo para el sistema GEA (Gestión Energética Asistida) que permita realizar el pronóstico adecuado del consumo de recursos energéticos en el ISMMM.

Este problema se enmarca sobre el **Objeto de Estudio**: Pronóstico sobre el consumo de recursos energéticos en el ISMMM.

El **Campo de Acción** considerado es: Pronóstico energético basado en métodos estadísticos.

Objetivo general:

Desarrollar el Módulo de Pronóstico del Consumo de Recursos Energéticos para la aplicación web GEA.

Idea a Defender:

Realizar el pronóstico del consumo de recursos energéticos basado en métodos estadísticos, posibilitará tener una visión adecuada de las necesidades energéticas futuras de la entidad.

Las **tareas** que se llevarán a cabo para darle cumplimiento al objetivo general planteado son:

- Analizar las soluciones informáticas existentes relacionadas con el pronóstico de consumo energético.
- Analizar los métodos estadísticos existentes para la realización de pronósticos.
- Valorar las tendencias actuales de las herramientas y tecnologías.
- Seleccionar las herramientas y tecnologías más adecuadas para el desarrollo del Módulo de Pronóstico del Consumo de Recursos Energéticos.
- Determinar los requisitos del Módulo del Pronóstico de Consumo de Recursos Energéticos.
- Diseñar la base de datos.
- Implementar el Módulo de Pronóstico del Consumo de Recursos Energéticos.
- Realizar pruebas funcionales.
- Crear el Manual de usuarios.

Los **métodos de investigación** empleados son los teóricos y empíricos. Los métodos empíricos empleados fueron la observación y análisis de documentos para la recopilación de la información. La observación se utilizó para conocer la funcionalidad del CEETAM,

del ISMMM y el comportamiento del problema. Mediante el análisis de documentos se pudo conocer el comportamiento de los algunos parámetros de consumo.

Los métodos teóricos utilizados fueron: el método histórico y lógico para la búsqueda de los antecedentes del software. El análisis y síntesis para la recopilación y el procesamiento de la información obtenida en los métodos empíricos y arribar a las conclusiones de la investigación. Mediante la modelación se realizó el estudio de la gestión de la información del CEETAM.

Las etapas de la investigación son:

- El estudio de las distintas metodologías de desarrollo de software, de las herramientas, tecnologías y patrones arquitectónicos para hacer una selección de las mejores, y utilizarlos en la confección del software de acuerdo a sus características.
- La contextualización del problema, enmarcando los objetivos específicos requeridos por el cliente.
- Análisis y diseño del software con todas las especificaciones requeridas.
- Realización del estudio de factibilidad.

El presente trabajo consta de introducción, cuatro capítulos, conclusiones, recomendaciones, glosario de términos, bibliografía y anexos.

En el Capítulo uno **Fundamentación Teórica**, se hace una descripción de los principales conceptos asociados al dominio del problema; se explica brevemente algunos conceptos sobre el modelo estadístico empleado y otros. Se hace referencia sobre algunos proyectos existentes que tienen relación con este trabajo. Se fundamenta la selección de las herramientas utilizadas para el desarrollo del trabajo.

En el Capítulo dos **Planeación y Diseño**, se hace uso de la metodología expuesta en el capítulo inicial para el desarrollo del proyecto, abordando en detalles cada una de sus fases. Se describen las funcionalidades que serán objeto de automatización mediante el empleo de las historias de usuarios (HU), se realiza una estimación del esfuerzo necesario para las mismas y se establece un plan de iteraciones necesarias sobre el sistema, para su terminación.

El Capítulo tres **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.

En el Capítulo cuatro **Estudio de Factibilidad y Sostenibilidad**, se realiza un estudio de los esfuerzos requeridos para la construcción del sistema, y se valora la sostenibilidad del producto.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

En el mundo actual, las TIC han evolucionado notablemente y junto con ella la producción de diversos productos de software. En este capítulo se realiza un esbozo de las características del pronóstico y el método estadístico empleado para su solución. Además se hace una descripción sobre las aplicaciones web, se brindan algunos conceptos y se exponen tanto la metodología a seguir en el desarrollo de la aplicación, como las herramientas a emplear en la misma.

1.1. Estado del arte

1.1.1. Términos fundamentales

1.1.1.1. Pronóstico

Es el arte y la ciencia de predecir los eventos futuros. Para ello se pueden usar datos históricos y su proyección hacia el futuro mediante algún tipo de modelo matemático. Además, generalmente es un proceso de estimación en situaciones de incertidumbre. El pronóstico ha evolucionado hacia la práctica del plan de demanda en el pronóstico diario de los negocios. Es un proceso crítico y continuo que se necesita para obtener buenos resultados durante la planificación de un proyecto. ([Mayorga Peralta, 2009](#))

1.1.1.2. Clasificación de los pronósticos con respecto al tiempo que abarcan

1. Pronósticos a corto plazo: En las empresas modernas, este tipo de pronóstico se efectúa cada mes, y su tiempo de planeación tiene vigencia de un año. Se utiliza para programas de abastecimiento, producción, asignación de mano de obra a las plantillas de trabajadores, y planificación de los departamentos de fabricación.
2. Pronósticos a mediano plazo: Abarca un lapso de seis meses a tres años. Este se utiliza para estimar planes de ventas, producción, flujos de efectivo y elaboración de presupuestos.
3. Pronósticos a largo plazo: Este tipo de pronóstico se utiliza en la planificación de nuevas inversiones, lanzamiento de nuevos productos y tendencias tecnológicas de materiales, procesos y productos, así como en la preparación de proyectos. El

tiempo de duración es de tres años o más. (Wikipedia.org)

1.1.1.3. Métodos estadísticos para el pronóstico

➤ Método de Serie Temporal. (Wikipedia.org)

Los métodos basados en las series temporales utilizan datos históricos como base para estimar resultados futuros. Se asume que la demanda es función del tiempo, y que además pueden estar involucrados los siguientes componentes:

- Tendencia.
- Ciclos.
- Estacionalidades.
- Irregularidades.

➤ Método de Mínimos Cuadrados. ([Freund, 1960](#))

Puede utilizarse para computar la tendencia de una línea recta o curva en este caso la teoría desarrollada está limitada al método para computar la tendencia de la línea recta por el método de mínimos cuadrados; el cálculo de la tendencia de la línea curva utiliza los mismos principios, pero implica matemáticas más complicadas; las estimaciones de la tendencia se computan de tal manera que, por ejemplo: la suma de las desviaciones al cuadrado de las ventas reales en relación a las estimadas llegan a un mínimo, de ahí el término de mínimos cuadrados. Las modalidades o cambios de este método de análisis estadístico en relación al expuesto anteriormente, son:

- **X** simboliza períodos de tiempo, siendo en este caso los meses, desarrollando una serie de dígitos, lo cual es conocido por el analista y no es difícil de determinar.
- **Y** representa a los consumos, producción, inventarios, etc., o sea, la variable cuyas fluctuaciones en relación al tiempo son estudiadas; en este caso la sigla Y se sustituye por la **P=** pronóstico.
- El valor estimado de la tendencia a largo plazo puede ser calculado en cuanto a los valores **b** y **a** (constantes numéricas) sean determinados, cuyas formulas son las denominadas cortas, como sigue:

$$b = \frac{n(\sum x_i y_i) - (\sum x_i)(\sum y_i)}{n(\sum x_i^2) - n(\sum x_i)^2} \quad a = \frac{\sum y_i - b(\sum x_i)}{n}$$

y la fórmula general sería: $P = a + bx$

1.1.2. Antecedentes

1.1.2.1. Ámbito Nacional

En la Universidad de Cienfuegos Carlos Rafael Rodríguez se realizó un estudio y aplicación de los principales métodos de pronóstico de consumo de energía eléctrica. Se utilizaron los métodos tradicionales y los más novedosos, como son análisis de series temporales con los métodos de Box-Jenkins (ARIMA) y el uso de la inteligencia artificial. Se empleó como elemento de comparación el EMC (Error Medio Cuadrático). Como fuente de datos se tomaron los relacionados con los últimos 10 cursos, en cuanto a la variación de los distintos tipos de matrícula, la variación del equipamiento y el consumo de energía eléctrica. Para el pronóstico de series temporales y la aplicación de métodos paramétricos y de regresión múltiple, se empleó el software STATGRAPH -5. ([Fernández Pérez, 2008](#))

En la Universidad de Cienfuegos se presentó un método de determinación experimental del coeficiente de resistencia del camino en una ruta concreta, en el cual se muestran importantes resultados que permiten el pronóstico de consumo de combustible de los vehículos automotores en los ciclos teóricos de viaje, resueltos a partir de la base de datos experimental del coeficiente de resistencia al camino, destacándose que el error relativo entre los resultados experimentales del consumo y los resultados teóricos del consumo solucionados con el valor real del coeficiente de resistencia al camino, no exceden el 10% y son mucho más precisos que los obtenidos a través de los valores del coeficiente recomendados en las tablas. ([Pérez Gálvez, 2008](#))

1.1.2.2. Ámbito Internacional.

Varios han sido los métodos empleados para pronosticar la demanda energética. En general estos métodos pueden ser clasificados en tres grandes grupos: los primeros se basan en modelos estadísticos (regresión, series de tiempo y/o econométricos); los segundos en inteligencia artificial y, finalmente, aquellos que dependen del juicio y la intuición humana. En Valencia (2005) se puede encontrar un cuadro detallado de cada uno de estos grupos, al igual que sus ventajas y desventajas.

Realizan un estudio usando modelos de tipo ARIMA. En este caso el pronóstico de demanda de energía eléctrica se basa en datos históricos suministrados por la Empresa de Energía de Pereira en el periodo comprendido entre el 1 de enero de 2001 y el 31 de diciembre de 2001.

Por otra parte, Blaconá & Abril (2000) realizan un estudio para el MEM (Mercado Eléctrico Mayorista) en Argentina. De acuerdo con los autores, las series con la demanda de energía eléctrica contienen procesos estacionales muy complejos, que resultan difíciles de modelar con la teoría básica de series de tiempo. Por esta razón prefieren apoyarse en los MBEE (Modelo Básico de Espacio de Estado). La serie de tiempo de la demanda diaria promedio de energía eléctrica de Argentina contiene dos tipos de estacionalidad, una semanal y otra anual. La estacionalidad semanal resulta fácil de modelar con variables DUMMY, mientras la estacionalidad anual se modela mediante un modelo spline de regresión no paramétrica descrito por Poirier (1973).

Harvey & Koopman (1993) emplean también MBEE y la técnica Time-Varying splines para obtener un modelo de pronóstico horario de la demanda de electricidad de la compañía americana Puget Sound Power and Light. Poirier et al. (1979) realizaron un estudio para el Electric Power Research Institute, el cual involucró modelos de tipo econométrico y splines con el fin de estimar la demanda residencial para un pequeño conjunto de consumidores en Estados Unidos de América, de los cuales se contaba con información cada 15 minutos. (Revista Colombiana de Estadística, 2007)

1.1.2.3. Pronóstico en el ISMMM

Actualmente en el ISMM no se realiza el proceso de pronóstico de consumo energético, lo que conlleva a no tener una predicción de los posibles parámetros de consumo de los portadores energéticos en próximos períodos. No se tiene una base sólida para la realización y configuración del proceso de planificación. Además no se conoce con anticipación los futuros gastos financieros por concepto adquisición de recursos energéticos.

1.2. Tendencias y tecnologías actuales

Ante el incesante avance de las tecnologías, la sociedad, ávida de nuevas herramientas y funcionalidades, exige a los desarrolladores de software nuevos retos y nuevas concepciones para satisfacer sus exigencias, cada vez, más ambiciosas. Para satisfacer estas exigencias, los desarrolladores deben buscar nuevas ideas surgiendo así nuevas metodologías y formas de desarrollo que permitan confeccionar productos cada vez más complejos.

1.2.1. Internet

Algunos de los nombres más usados para definir Internet son “Red de Redes” o “Autopista de la Información”. Internet es una Red de Redes porque está hecha a base de unir muchas redes locales de ordenadores. Además, es considerada "La Red de Redes" porque es la más grande.

Por Internet circulan constantemente cantidades increíbles de información. Por este motivo se le llama también la Autopista de la Información. Actualmente existen alrededor de 50 millones de personas con acceso a este recurso. La información circula de manera casi instantánea, lo que hace que el mundo se encuentre conectado a cada segundo, y con ello, se pueden realizar actividades como transacciones económicas, comunicación a distancia, etc. (Universidad Nacional de Lomas de Zamora, 2008)

Uno de los fundamentos de Internet es el TCP/IP, un protocolo de transmisión que asigna a cada máquina que se conecta un número específico, llamado «número IP». El protocolo TCP/IP sirve para establecer una comunicación entre dos puntos remotos mediante el envío de información en paquetes. Internet brinda además una serie de servicios, los más usados son Correo Electrónico, World Wide Web, FTP, Grupos de Noticias, IRC y Servicios de Telefonía. (Hervás Gómez, 2006)

1.2.2. Aplicaciones WEB. Ventajas y Desventajas

Con la llegada de la Web 2.0, propiciada esta por el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) y los nuevos retos de Internet, se abre paso una nueva idea en la concepción del software; el software orientado a servicios o SaaS (acorde a sus siglas en inglés). SaaS es un modelo de distribución del software que

proporciona a los clientes el acceso al mismo a través de la red (generalmente Internet), de manera que les libra del mantenimiento de las aplicaciones, de operaciones técnicas y de soporte. Las aplicaciones distribuidas en la modalidad SaaS pueden llegar a cualquier tipo de empresa sin importar su tamaño o su ubicación geográfica. Se trata de un modelo que une el producto (software) al servicio, para dotar a las empresas de una solución completa que permita optimizar sus costes y sus recursos. (CIIN, 2005)

Ventajas WEB: (Graham, 2001)

Compatibilidad multiplataforma: Las aplicaciones Web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.

Actualización: Las aplicaciones basadas en Web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo, pues no se hace necesario iniciar nuevas descargas y/o procedimientos de instalación (algunas veces imposible cuando usted está trabajando dentro de grandes organizaciones).

Inmediatez de acceso: Las aplicaciones basadas en Web no necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online y están listas para trabajar sin importar cuál es su configuración o su hardware.

Menos requerimientos de memoria: Las aplicaciones basadas en Web tienen menos demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del proveedor, esas aplicaciones basadas en Web usan en muchos casos la memoria de las computadoras donde ellas corren, dejando más espacio para correr múltiples aplicaciones sin incurrir en frustrantes deterioros en el rendimiento.

Menos Bugs: Las aplicaciones basadas en Web deberían ser menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones basadas en Web, todos utilizan la misma versión, y todos los bugs pueden ser corregidos tan pronto como son descubiertos.

Precio: Las aplicaciones basadas en Web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional. Esto permite que las aplicaciones online cuesten una fracción de sus contrapartes descargables si no totalmente gratuitas, mientras que ofrecen componentes adicionales y servicios Premium como una opción.

Los datos también van online: El hecho de que el manejo de los datos sea realizado de forma remota libra al usuario de la responsabilidad en la protección de los mismos, y al mismo tiempo logra que los recursos sean accesibles en cualquier momento.

Múltiples usuarios concurrentes: Las aplicaciones basadas en Web pueden ser utilizadas por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta.

Los datos son más seguros: Si bien la ruptura de discos no va a desaparecer, es probable que los usuarios escuchen mucho menos del tema. A medida que las compañías se hagan cargo del almacenamiento de los datos del usuario, granjas de almacenamiento de datos redundantes, altamente fiables, los usuarios van a tener mucho menos riesgo de perder sus datos debido a una ruptura de disco impredecible o a un virus de la computadora. Las compañías que proveen aplicaciones basadas en Web van a brindar amplios servicios de resguardo de datos ya sea como una parte integral del servicio básico o como una opción paga.

Desarrollar aplicaciones en el lenguaje que usted quiera: Una vez que las aplicaciones han sido separadas de computadoras locales y sistemas operativos específicos, pueden también ser escritas en prácticamente cualquier lenguaje de programación. Debido a que las aplicaciones Web son esencialmente una colección de programas más que un simple programa, ellas podrían ser escritas en cualquier lenguaje de programación existente.

Desventajas WEB: (Masternewmedia.org, 2005)

- Acceso limitado, la necesidad de conexión permanente y rápida a Internet hacen que el acceso a estas aplicaciones no esté al alcance de todos.
- La interactividad no se produce en tiempo real, en las aplicaciones Web cada acción del usuario conlleva un tiempo de espera hasta que se obtiene la reacción del sistema.

- Elementos de interacción muy limitados. En comparación con el software de escritorio, las posibilidades de interacción con el usuario que ofrecen las aplicaciones Web (mediante formularios principalmente) son muy escasas.
- Diferencias de presentación entre plataformas y navegadores. La falta de estándares ampliamente soportados dificulta el desarrollo de las aplicaciones.

1.2.3. Servidores WEB

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición. Actualmente uno de los servidores Web más difundido es Apache. (Vegas, 2002)

Apache

Apache 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.

- 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 logs. 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.

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)

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. (Wikipedia.org)

1.2.4. Lenguajes de Programación WEB

PHP

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994.

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK. (Achour, et al.)

Ventajas de PHP: (Dondo, 2005)

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones.

Desventajas de PHP: (Dondo, 2005)

- No posee una abstracción de base de datos estándar, sino bibliotecas especializadas para cada motor (a veces más de una para el mismo motor).
- No posee adecuado manejo de internacionalización, unicode, etc.
- Por su diseño dinámico no puede ser compilado y es muy difícil de optimizar.

- Por sus características promueve la creación de código desordenado y complejo de mantener.
- Está diseñado especialmente para un modo de hacer aplicaciones web que es ampliamente considerado problemático y obsoleto (mezclar el código con la creación de la página web).

JSP

La tecnología Java para la creación de páginas web con programación en el servidor. JSP es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es una tecnología orientada a crear páginas web con programación en Java. Java Server Pages es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. (Universidad de Valparaíso, 2006)

Esta tecnología es un desarrollo de la compañía Sun Microsystems. La Especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible la Especificación JSP 2.1.

Las JSP's permiten la utilización de código Java mediante scripts. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Librerías de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.

Con JSP se puede crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP podrán escribirse con el editor HTML/XML habitual. (Millán Tejedor, 2005)

1.2.5. Entornos de programación Web

Dreamweaver: Sin lugar a dudas, es una de las herramientas más utilizadas por los webmasters para el trabajo con aplicaciones visuales en este caso en el diseño e implementación de páginas Web. Se adapta increíblemente a las necesidades de todo tipo de profesional de diseño Web, tanto para lo que prefieren programar el código

directamente en el editor de texto como para los que gustan del ambiente visual. Se trata de un editor de texto especialmente diseñado para trabajar con documentos Web como HTML, PHP, ASP, JavaScript, entre otros.

PhpDesigner 2008 v6.0.0: Es un completo entorno de desarrollo y programación especialmente diseñado para los gurús de PHP, aunque también permite trabajar con comodidad en otros lenguajes de programación como HTML, XHTML, CSS y SQL.

Ofrece toda una serie de asistentes y diálogos integrados que facilitan en todo momento las tareas, además de acceso directo a librerías de código o scripts de uso habitual, utilidades diversas y toda suerte de herramientas, todo ello en una interfaz de diseño sencillo y elegante que se puede personalizar con nada menos que dieciocho temas distintos.

Cuenta con cliente de FTP y navegador de ficheros integrado, utilidades de corrección y autocompletado, búsqueda integrada en Google y soporte para proyectos, además de usar un práctico esquema de color para la sintaxis del código fuente que facilita enormemente la programación. PhpDesigner soporta: PHP, HTML, XHTML, CSS, Java, Perl, JavaScript, VB, C# y SQL.

1.2.6. Sistemas Gestores de Base de Datos (SGBD)

Un Sistema Gestor de Bases de Datos (SGBD) o DBMA es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. (CAVSI, 2004)

Algunos ejemplos de SGBD son PostgreSQL, MySQL, SQL Server, etc.

Un SGBD debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Características de un Sistema Gestor de Base de Datos: (CAVSI, 2004)

Abstracción de la información. Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

Independencia. La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

Redundancia mínima. Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.

Consistencia. En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

Seguridad. La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra protegida frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipularla o destruirla; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

Integridad. Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.

Respaldo y recuperación. Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

Control de la concurrencia. En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

PostgreSQL (Eaprende.com, 2001)

Es un servidor de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD.

Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Principales características:

- Alta concurrencia: Mediante un sistema denominado MVCC (Acceso concurrente Multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.
- Amplia variedad de tipos nativos. PostgreSQL provee nativamente soporte para:
 - Números de precisión arbitraria.
 - Texto de largo ilimitado.
 - Figuras geométricas (con una variedad de funciones asociadas)
 - Direcciones IP (IPv4 e IPv6).
 - Bloques de direcciones estilo CIDR.
 - Direcciones MAC.
- Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

Gracias a su licencia BSD, se permite la utilización del código para ser comercializado. Uno de los casos ejemplo es la de Enterprise DB (POstgresql Plus), la cual incluye varios agregados y una interfaz de desarrollo basada en Java. Entre otras empresas que utilizan Postgresql para comercializar se encuentra CyberTech (Alemania), con su producto CyberCluster.

MySQL

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.

MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales. (Nieves Borrero, et al., 2007)

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.

- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas.

Ventajas: (Eaprende.com, 2001)

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad

Desventajas

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).

SQL Server

SQL Server es un conjunto de objetos eficientemente almacenados. Los objetos donde se almacena la información se denominan tablas, y éstas a su vez están compuestas de filas y columnas. En el centro de SQL Server está el motor de SQL Server, el cual procesa los comandos de la base de datos. Los procesos se ejecutan dentro del sistema operativo y entienden únicamente de conexiones y de sentencias SQL.

SQL Server incluye herramientas para la administración de los recursos que el ordenador proporciona y los gestiona para un mejor rendimiento de la base de datos.

Transact-SQL es el lenguaje que utiliza SQL Server para poder enviar peticiones tanto de consultas, inserciones, modificaciones, y de borrado a las tablas, así como otras peticiones que el usuario necesite sobre los datos. En definitiva, es un lenguaje que utiliza SQL Server para poder gestionar los datos que contienen las tablas.

El lenguaje estándar SQL (Structured Query Language) se emplea para los sistemas de bases de datos relacionales RDBMS (Relational Database Management System), es el estándar ANSI (American National Standards Institute). (FormaSelect)

Características de Microsoft SQL Server: (Isrosoft.com, 2006)

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita.

1.3. Arquitectura

1.3.1. Patrones Arquitectónicos

Patrón de arquitectura MCV (Modelo Vista Controlador): Para el diseño de aplicaciones con sofisticadas interfaces se emplea el patrón de diseño MCV. La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si se realiza un diseño ofuscado, es decir, una forma de mezclar los componentes de interfaz y de negocio, entonces, la consecuencia será que, cuando se necesite cambiar la interfaz, tendrá que modificarse trabajosamente los componentes de negocio, por lo que propiciará mayor trabajo y más riesgo de error. Se trata de realizar un diseño que desacople la vista del modelo, con el fin de perfeccionar la reusabilidad. De este modo las modificaciones son las que impactan en menor medida en la lógica de negocio o de datos. (JIMÉNEZ IGLESIAS, 2009)

Elementos del patrón:

- Modelo: datos y reglas de negocio.
- Vista: muestra la información del modelo al usuario.
- Controlador: gestiona las entradas del usuario.

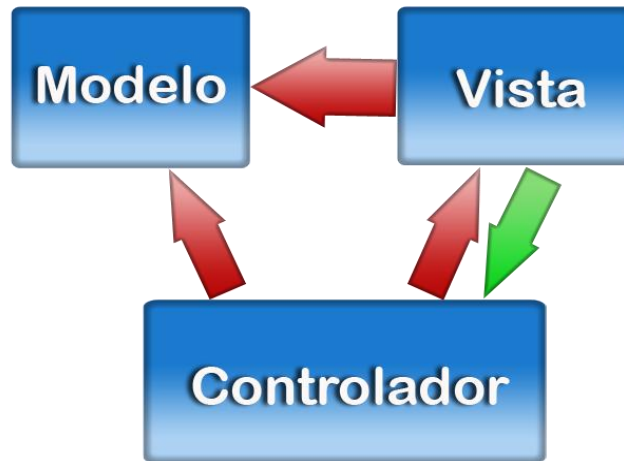


Figura 1.1 Modelo vista controlador.

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador. Un ejemplo clásico es el de la información en una base de datos, que puede presentarse de diversas formas: diagrama de pastel, de barras, tabular, etc. Analizan cada componente:

El modelo es responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Definir las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: “si la mercancía solicitada no está en el almacén, consultar el tiempo de entrega estándar del proveedor”.
- Llevar un registro de las vistas y controladores del sistema.
- Si se está en presencia un modelo activo, el mismo notificará a las vistas los cambios que en los datos pueda producir un agente externo (ejemplo: un fichero bat que actualiza los datos, un temporizador que desencadena una inserción, etc.).

El controlador es responsable de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).

- Contiene reglas de gestión de eventos, del tipo “si el evento z, entonces acción w”. estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada a actualizar.

Las vistas son responsables de:

- Recibir datos del modelo y mostrarlo al usuario.
- Tienen un registro de su controlador asociado, (normalmente porque además lo demanda).
- Pueden suministrar el servicio de actualizar, para que sea solicitado por el controlador o por el modelo, (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

Arquitectura en capas

Los sistemas o arquitecturas en capas constituyen uno de los estilos que aparecen con mayor frecuencia mencionados como categorías mayores del catálogo o por el contrario, como una de las posibles imágenes de algún estilo envolvente. Definen el estilo en capas como una organización jerárquica tal, que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones de la inmediatamente inferior. (REYNOSO, y otros, 2007)

La arquitectura por capas es un estilo de arquitectura en la que el objetivo primordial es la separación de la lógica de negocio de la lógica de diseño, un ejemplo básico es separar la capa de datos, de la capa de presentación al usuario. La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles, simplemente es necesario conocer las API que existen entre niveles.

El diseño de sistemas informáticos suele usar las arquitecturas multinivel o programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables, (que pueden ampliarse con facilidad en caso de que las necesidades aumenten). El diseño más en boga actualmente es el diseño en tres capas.



Figura 1.2 Arquitectura en tres capas.

Capas o niveles:

- Capa de presentación o interface: es la capa de que le permite al usuario interactuar con el sistema, captura y le comunica la información al mismo, dando un mínimo de proceso, (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la del negocio.
- Capa de lógica o de negocio: es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio e incluso lógica del negocio, pues es aquí donde se establecen las reglas que deben cumplirse. Esta capa se comunica con la de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos para solicitar al gestor de bases de datos para almacenar o recuperar datos de él.
- Capa de datos: es donde se ubican los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de los mismos, reciben solicitudes de almacenamiento o de recuperación de información desde la lógica del negocio.

Todas estas capas pueden residir en un único ordenador, esto no sería lo normal, lo más usual es que haya una multitud de ordenadores donde reside la capa de interface (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en un mismo ordenador, y si el crecimiento de las necesidades lo aconseja, pueden dividirse en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos

aumenta, pueden separarse en varios ordenadores los cuáles recibirán las peticiones del ordenador en que resida la capa de negocio. Si por el contrario, la complejidad fuese en la capa de negocio lo que obligase a la separación, esta lógica del negocio podría residir en uno o más ordenadores que realizarían las solicitudes a una única base de datos.

1.3.2. Modelo Cliente-Servidor

El esquema cliente-servidor “es un modelo de computación en el que el procesamiento requerido para ejecutar una aplicación o conjunto de aplicaciones relacionadas se divide entre dos o más procesos que cooperan entre sí “.

Los principales componentes del esquema cliente-servidor son entonces los Clientes, los Servidores y la infraestructura de comunicaciones. (Monografias.com, 2007)

Modelo cliente-servidor de 2 capas

Las aplicaciones cliente-servidor clásicas o de 2 capas, como su nombre lo indica, agrupan la lógica de presentación (interfaz) y la lógica de aplicación en la máquina cliente y acceden a fuentes de datos compartidos a través de una conexión de red que se encuentran en el servidor de datos.

La ventaja que presenta este tipo de aplicaciones es que los datos están centralizados. Esta centralización beneficia a la empresa pues es más fácil compartir los datos, se simplifica la generación de reportes y se proporciona consistencia en el acceso a los datos.

A continuación se enumeran las principales desventajas que presentan este tipo de aplicaciones:

- **Difíciles de mantener:** Esto viene dado por el hecho de que son difíciles de mantener las reglas de negocio de la lógica de aplicación ya que estas están programadas en cada cliente y esto implica que cualquier cambio tiene que ser redistribuido en todos los clientes.
- **Se compromete la confidencialidad:** Al tener programada la lógica de aplicación en el cliente este tiene a su disposición todas las reglas de negocio de la empresa.

- **Están estrechamente limitadas a una fuente de datos:** Los clientes casi siempre están configurados para acceder a una base de datos en particular por lo que mover los datos a una base de datos diferente se hace realmente complicado. (Janium.com, 2009)

Modelo cliente-servidor de 3 capas

En la arquitectura de 3 capas, la presentación, la lógica de aplicación y los elementos de datos están conceptualmente separados. Los componentes de la capa de presentación manejan la interacción con el usuario y realizan las peticiones del cliente a los componentes de la capa intermedia. Los componentes de la capa intermedia, manipulan la lógica de negocio y hacen las peticiones a la base de datos.

A continuación se enumeran algunas ventajas de las aplicaciones de 3 capas:

- Los componentes de la aplicación pueden ser desarrollados en cualquier lenguaje general lo que posibilita que el grupo de desarrolladores no se centre en el uso de un solo lenguaje.
- Los componentes están centralizados lo que posibilita su fácil desarrollo, mantenimiento y uso.
- Los componentes de la aplicación pueden estar esparcidos en múltiples servidores permitiendo una mayor escalabilidad.
- Los problemas de limitación para las conexiones a las bases de datos se minimizan ya que la base de datos solo es vista desde la capa intermedia y no desde todos los clientes. Además que las conexiones y los drivers de las bases de datos no tienen que estar en los clientes.
- Los componentes de aplicación de la capa intermedia pueden ser asegurados centralmente usando una infraestructura común. Se pueden conceder o denegar los permisos componente a componente simplificando la administración. (Ronda Amador, y otros, 2002)

1.4. Metodología de Desarrollo de Software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Pueden ser comparadas con un plan de contingencias en el que se va indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además

quienes deben participar en el desarrollo de las actividades y qué papel deben de tener. Detallan además la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

1.4.1. Metodologías Ágiles

En una reunión celebrada en febrero de 2001 en Utah-EEUU, nace el término "ágil" aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Varias de las denominadas metodologías ágiles ya estaban siendo utilizadas con éxito en proyectos reales, pero les faltaba una mayor difusión y reconocimiento.

Tras esta reunión se creó The Agile Alliance, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía "ágil". (Crystal Methodologies, 2008)

Actualmente, existe una amplia variedad de metodologías ágiles. Las más difundidas se enumeran a continuación.

- Extreme Programming
- Crystal Clear
- SCRUM
- DSDM (Dynamic System Development Method)
- FDD

1.4.2. Metodología XP.

(Beck,1999), **XP** es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

1.4.2.1. Cuatro valores que promueven la metodología XP:

Simplicidad: XP propone el principio de hacer las cosas más simple que pueda funcionar, en relación al proceso y la codificación. Es mejor hacer hoy algo simple, que hacerlo complicado y probablemente nunca usarlo mañana.

Comunicación: Algunos problemas en los proyectos tienen su origen en que alguien no dijo algo importante en algún momento. XP hace imposible la falta de comunicación.

Retroalimentación: Retroalimentación concreta y frecuente del cliente, del equipo y de los usuarios finales da una mayor oportunidad de dirigir el esfuerzo eficientemente.

Coraje: El coraje (valor) existe en el contexto de los otros 3 valores. (Beck,1999)

1.4.2.2. Roles XP

Programador. El programador escribe las pruebas unitarias y produce el código del sistema.

Cliente. Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.

Encargado de pruebas (Tester). Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

Encargado de seguimiento (*Tracker*). Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.

Entrenador (*Coach*). Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

Consultor. Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

Gestor (*Big boss*). Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación. (BECK, 1999)

1.4.2.3. Proceso XP

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

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

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración. (BECK, 1999)

1.4.2.4. Prácticas XP

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías

disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las siguientes prácticas. (BECK, 1999)

- **El juego de la planificación.** Hay una comunicación frecuente el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.
- **Entregas pequeñas.** Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.
- **Metáfora.** El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema , ayudando a la nomenclatura de clases y métodos del sistema).
- **Diseño simple.** Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.
- **Pruebas.** La producción de código está dirigida por las pruebas unitarias. Éstas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema.
- **Refactorización (Refactoring).** Es el proceso de modificar el código de un sistema de software de modo que no se altere su comportamiento externo pero se mejore su estructura interna. Es una técnica disciplinada de reestructuración de código. Parte del proceso puede automatizarse y de hecho existen herramientas que facilitan la tarea.
- **Programación en parejas.** Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores).

- **Propiedad colectiva del código.** Cualquier programador puede cambiar cualquier parte del código en cualquier momento.
- **Integración continua.** Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.
- **Cuarenta horas por semana.** Se debe trabajar un máximo de cuarenta horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo.
- **Cliente in-situ.** El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.
- **Estándares de programación.** XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible.

1.4.2.5. Ventajas de XP

- 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.
- 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.

1.4.2.6. Desventajas de XP

- XP no es escalable a equipos de muchos desarrolladores (a lo sumo 15)
- Es una metodología nueva y no está ampliamente probada
- Requiere un equipo de programadores altamente especializados y/o con experiencia considerable
- Requiera alto compromiso del equipo de desarrollo, lo cual a veces es difícil debido a la precaria situación contractual de la gente que trabaja en consultoras.

1.4.2.7. Fases de la metodología XP

Fase I: Planificación

- 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 storie*) y permita hacer una estimación de tiempo de desarrollo del mismo.
- 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.
- 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.
- Se cambia el proceso cuanto sea necesario, para adaptarlo al proyecto.

Fase II: Diseño

- Se eligen los diseños funcionales más simples.
- Se elige una metáfora del sistema para que el nombrado de clases, siga una misma línea, facilitando la reutilización y la comprensión del código.
- 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.

Fase III: Codificación

- 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.
- El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código.
- 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.
- 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.
- 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.
- 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.
- Se deja la optimización para el final.
- No se hacen horas extra de trabajo.

Fase IV: Pruebas

- Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado.
- Cuando se encuentra un error de codificación o bug, se desarrollan pruebas para evitar volver a caer en el mismo.
- Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las historias de usuarios 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 la correspondiente historia de usuario se ha completado.

1.5.Herramientas CASE (Computer – Aided Software Engineering).

Las herramientas CASE (Ingeniería de Software Asistida por Computadoras), son aplicaciones informáticas que tienen como objetivo fundamental solucionar y afrontar los problemas de mala calidad de software y documentación inadecuada. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo de un determinado proyecto o software, ya que brindan la posibilidad de realizar cálculos de costos, generan código fuente automáticamente de un diseño previamente dado, poseen compilación automática, ayudan con la documentación y juegan un papel importante en la detección de errores.

Por tales ventajas algunas personas han dado su criterio personal sobre el concepto CASE, que se define como:

“Herramientas individuales para ayudar al desarrollador de software o administrador de proyecto durante una o más fases del desarrollo de software o mantenimiento del mismo”.

Rational Rose: Es una herramienta software para el modelado visual mediante el lenguaje unificado de modelado (unified modeling lenguaje UML) de sistemas software, permite especificar, analizar, diseñar el sistema antes decodificar. Mantiene la consistencia de los modelos del sistema software, además de la generación de documentos automáticamente. Otra funcionalidad de la herramienta es que permite realizar la ingeniería inversa de un producto.

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. ER/Studio se diseña para hacer más fácil de entender el estado actual de los datos de las empresas. Simple y fácil al usuario, ayuda a 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)

Ventajas:

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

1.6. Lenguaje Unificado de Modelado (UML)

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. Capta la información sobre la estructura estática y el comportamiento dinámico de un sistema, no es un lenguaje de programación.

Las herramientas CASE pueden ofrecer generadores de código de UML para una gran variedad de lenguaje de programación, así como construir modelos por ingeniería inversa a partir de programas existentes. UML es un lenguaje de propósito general para el modelado orientado a objetos, es también un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes. (Neuron Corp., 2006)

Objetivos del UML: (Vico.org, 2002)

- Es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- No pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso, incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. Necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.

- Ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Imponer un estándar mundial.

1.7. Tecnologías y Herramientas a utilizar

Este módulo que se propone está concebido bajo el uso de herramientas libres, garantizando la independencia tecnológica de la solución, y apoyándose en los lineamientos existentes para el desarrollo informático en el ISMM.

Se hizo uso de la plataforma web brindada por **Apache** HTTP Server apoyándose en sus características principales. Apache es uno de los servidores más utilizados por ser multiplataforma, de tecnología gratuita, diseño modular y código fuente disponible. (Ciberaula.com, 2005)

Como lenguaje de programación se utilizó **PHP** (Personal Home Page Tools): lenguaje interpretado creado inicialmente para la creación de páginas dinámicas. Es un lenguaje multiplataforma con capacidad de conexión hacia múltiples gestores de base de datos, con una amplia documentación y comunidad de soporte, permite las técnicas de programación orientada a objeto. (Achour, 2005)

El gestor de Base de Datos seleccionado fue **MySQL** el cual es un sistema de gestión de bases de datos relacional, y tiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQL es un software de código abierto, licenciado bajo la Licencia Pública General (GNU/GPL) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones. (Nieves Borrero, y otros, 2007)

El programa seleccionado para la realización del diseño de la interfaz gráfica del sitio web fue **Dreamweaver** que, sin lugar a dudas, es una de las herramientas más utilizadas por los webmasters para el trabajo con aplicaciones visuales, en este caso en el diseño e implementación de páginas Web. Se adapta increíblemente a las necesidades de todo tipo de profesional de diseño Web, tanto para los que prefieren programar el código directamente en el editor de texto como para los que gustan del ambiente visual. Se trata de un editor de texto especialmente diseñado para trabajar con documentos Web como HTML, PHP, ASP, JavaScript, entre otros.

Como resultado del análisis realizado durante el presente capítulo se concluye que: los proyectos relacionados con pronósticos energéticos analizados, no responden a las necesidades del centro, pues no forman parte de un sistema informático y analizan los portadores energéticos de forma individual. Debido a esto se necesita implementar un módulo para el pronóstico del consumo energético. Para ello se realizó un estudio de las tecnologías actuales y se seleccionaron las más adecuadas para la implementación del sistema.

CAPÍTULO 2 PLANEACIÓN Y DISEÑO

En este capítulo, se introduce la fase de planeación y diseño, donde se detallan las necesidades del cliente, se describen las funcionalidades que serán objeto de automatización mediante el empleo de las historias de usuarios (HU), se realiza una estimación del esfuerzo necesario para las mismas y se establece un plan de iteraciones necesarias sobre el sistema, para su terminación.

2.1. Personal relacionado con el Sistema

Personas relacionadas con el sistema	Justificación
Especialista ó Energético	Esta es la persona que tiene conocimiento en la materia de pronóstico de consumo energético, y está encargada de la gestión de la información del mismo.
Administrador	Es la persona encargada de asesorar y dar seguimiento del estado del proceso de desarrollo.
Desarrollador	Es la persona responsable de llevar a cabo la implementación del sistema.

Tabla 2.1 Personal relacionado con el Sistema.

2.2. Lista de Reserva

Después de conocer el personal relacionado, se procede a realizar el análisis de las funcionalidades que debe cumplir la aplicación para dar respuesta a los mismos. Para ello se enumerarán mediante una lista de reserva, las funcionalidades que el sistema debe ser capaz de cumplir. Para lograr que el módulo cumpla con lo antes expuesto, se tuvo en cuenta la siguiente lista de reservas:

- Insertar y realizar cálculo de pronóstico de actividad de transporte.
- Mostrar resultado del pronóstico de actividad de transporte.
- Eliminar pronóstico de actividad de transporte.
- Insertar y realizar cálculo de pronóstico de consumo eléctrico.
- Mostrar resultado del pronóstico de consumo eléctrico.
- Eliminar pronóstico de consumo eléctrico.
- Insertar y realizar cálculo de pronóstico por áreas administrativas.
- Mostrar resultado del pronóstico por áreas administrativas.
- Eliminar pronóstico por áreas administrativas.
- Generar reporte de pronóstico de consumo eléctrico.
- Generar reporte de pronóstico de actividad de transportación.
- Generar reporte de pronóstico por áreas administrativas.

2.3. Historias de Usuario

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

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

Modelo de planilla de historia de usuario

Historia de usuario	
Número: No. Historia de usuario	Usuario: Usuario entrevistado para obtener la función requerida a automatizar.
Nombre: nombre de la historia de usuario que sirve para identificarla mejor entre los desarrolladores y el cliente.	
Prioridad en el negocio: Importancia: Alta / Media / Baja	Riesgo en desarrollo: Dificultad: Alta / Media / Baja
Puntos estimados: Estimación: de 1 a 3 puntos	Iteración asignada: Iteración a la que corresponde
Programador responsable: Nombre de encargado de programación.	
Descripción: 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 Planilla de historia de usuario.

HU No.1: Pronosticar el consumo de actividad de transporte.

Historia de usuario	
Número: 1	Usuario: Energético del ISMM
Nombre: Pronosticar el consumo de actividad de transporte.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 1

Programador responsable: Yanquiel Borges Matos.
Descripción: El usuario debe insertar los datos de pronóstico correspondientes al intervalo de tiempo a pronosticar. Una vez insertados estos datos la información se registrará en el historial de pronóstico de actividad de transporte para poder ser mostrada y actualizada.
Observaciones: Confirmado con el cliente.

Tabla 2.3 HU No.1

Remitirse al [ANEXO 2](#) para ver las Historias de Usuario.

2.4. 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 los puntos de estimación correspondientes a las HU, que fueron terminadas en la última iteración.

2.4.1. Estimación de Esfuerzo por Historias 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:

Estimación de esfuerzo por historias de usuario:

Historias de usuarios	Puntos de estimación
Pronosticar el consumo de actividad de transporte	3 semanas
Pronosticar el consumo eléctrico	2 semanas
Pronosticar el consumo por áreas administrativas	3 semanas
Generar reportes de pronóstico	2 semanas

2.4.2. 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 reléase, lo cual debe 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:

Al darle cumplimiento a dicha iteración, se contará con todas las funcionalidades descritas en las HU 1 y 2, las cuales hacen alusión a la inserción y visualización de la información para el pronóstico del consumo energético por actividad de transporte y por electricidad. Se tendrá la primera versión de prueba, que contará con dos modelos de desarrollo que incorporan todas las funcionalidades antes vistas, estos modelos se presentarán al cliente con el objetivo de obtener una retroalimentación del mismo para posteriores iteraciones del producto.

Segunda iteración:

Esta iteración tiene como finalidad desarrollar las HU 3, las cual brinda las funcionalidades de obtener y visualizar la información relacionada con el pronóstico de consumo por áreas

administrativas. Se tendrá la primera versión de prueba, que contará con un modelo de desarrollo que incorpora todas las funcionalidades antes vistas, este modelo se presentará al cliente con el objetivo de obtener una retroalimentación del mismo para posteriores iteraciones del producto.

Tercera iteración:

Esta última iteración del módulo tiene como propósito llevar a cabo el desarrollo de la HU 4, la cual brinda las funcionalidades de emitir reportes de pronóstico. Esta HU será integrada con el resultado de las iteraciones anteriores, y como fruto de esta integración se obtendrá la versión 1.0 del producto final. A partir de este momento el software será puesto a un proceso de prueba para evaluar el desempeño del mismo.

2.4.3. Plan de duración de las iteraciones

Iteración	Historias de usuario	Duración total
Iteración 1	Pronosticar el consumo de actividad de transporte	5 semanas
	Pronosticar el consumo eléctrico	
Iteración 2	Pronosticar el consumo por áreas administrativas	3 semanas
Iteración 3	Generar reportes de pronóstico	2 semanas

Tabla 2.7 Plan de duración de iteraciones

En este capítulo se abordó la fase de planeación y diseño donde se presentaron las distintas 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, culminando así esta fase y se determina que el equipo de trabajo está listo para pasar a la siguiente etapa de desarrollo.

CAPÍTULO 3 DESARROLLO Y PRUEBAS

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, se presentan las tarjetas CRC (clases, responsabilidades y colaboradores), que permitirán trabajar con una metodología basada en objetos, se realiza el desarrollo de las iteraciones a partir del desglose de las historias de usuario en tareas. Se muestran las interfaces gráficas de usuario diseñadas para la aplicación final. Se describen además, las pruebas realizadas y se indican las respuestas de la aplicación en el empleo de las diferentes funcionalidades, así como los posibles mensajes de error, información o aceptación que emite la misma cuando se utiliza una de estas funcionalidades.

3.1. Modelo de Datos

En esta parte se muestra el modelo de datos empleado para la aplicación. Para pronóstico de consumo de transporte, se crea la tabla Pronostico_Transp y para pronóstico de consumo eléctrico, se crea la tabla Pronostico_Energia. En ambas tablas se recopilan los datos que se obtienen de las tablas del módulo de control y de las tablas Portador e Indicador, los cuales son calculados mediante el método estadístico para hallar pronóstico, mínimos cuadrados y posteriormente almacenados. Dentro de estos datos se encuentran el mes, el año, valor de consumo y el portador energético con su indicador.

Para pronóstico de consumo por áreas administrativas, se creó la tabla Pronostico_Area_Admin. En esta tabla se recopilan los datos de las tablas de los módulos de planificación como la tabla Area_Admin y de control así como de las tablas Portador e Indicador, los cuales son calculados mediante el método estadístico para hallar pronóstico, mínimos cuadrados y posteriormente almacenados. Dentro de estos datos se encuentran el área administrativa, el mes, el año, valor de consumo y el portador energético con su indicador.

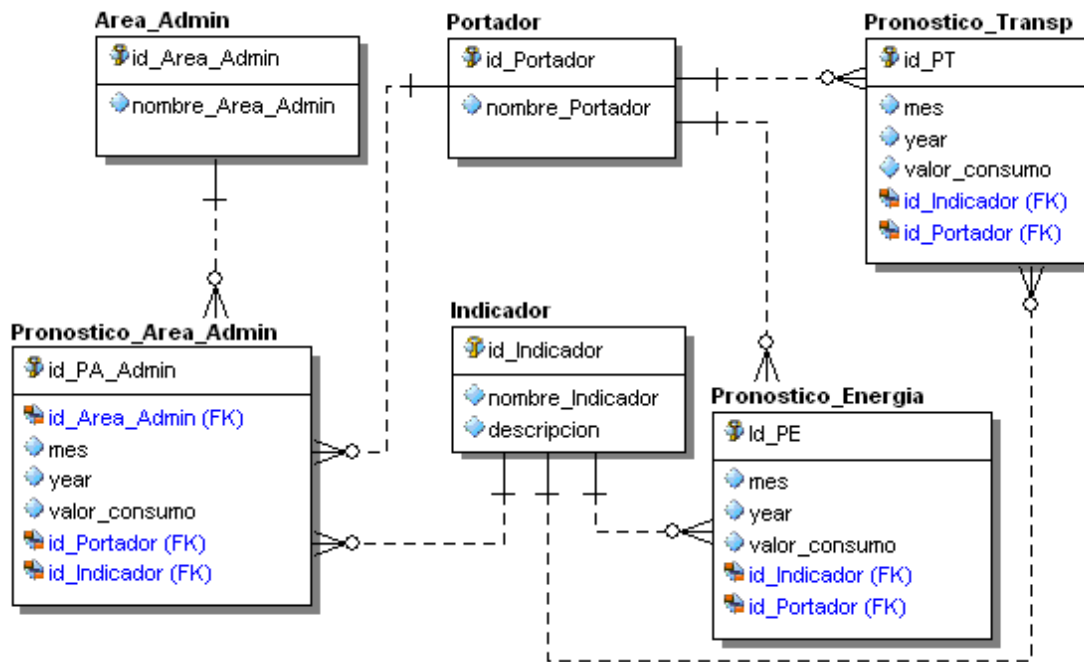


Figura 3.1 Modelo de datos.

3.2. Clases, Responsabilidades y Colaboradores

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 correspondiente. Para mejor comprensión de las mismas, se determina agruparlas por HU.

Remitirse a [ANEXO 3](#) para ver Tarjetas CRC.

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.

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.

Distribución de tareas por historia de usuario

Historia de usuario	Tareas
Pronosticar el consumo de actividad de transporte	<ul style="list-style-type: none">● Insertar y realizar cálculo de pronóstico de actividad de transportación.● Eliminar pronóstico de actividad de transportación.● Mostrar resultado del pronóstico de actividad de transportación.
Pronosticar el consumo eléctrico	<ul style="list-style-type: none">● Insertar y realizar cálculo de pronóstico de consumo eléctrico.● Eliminar pronóstico de consumo eléctrico.● Mostrar resultado del pronóstico de consumo eléctrico.
Pronosticar el consumo por áreas administrativas	<ul style="list-style-type: none">● Insertar y realizar cálculo de pronóstico por áreas administrativas.● Eliminar pronóstico por áreas administrativas.● Mostrar resultado del pronóstico por áreas administrativas.
Generar reportes de pronóstico	<ul style="list-style-type: none">● Generar reporte de pronóstico de consumo eléctrico.● Generar reporte de pronóstico de actividad de transportación.● Generar reporte de pronóstico por áreas administrativas.

Tabla 3.10 Distribución de tareas por historia de usuario

Historias de usuario abordadas en la primera iteración

Historias de usuario	Tiempo de estimación (semanas)	
	Estimación inicial	Real
Pronosticar el consumo de actividad de transporte	3	2
Pronosticar el consumo eléctrico	2	1

Tabla 3.11 Historias abordadas en la primera iteración

Historias de usuario abordadas en la segunda iteración

Historias de usuario	Tiempo de estimación (semanas)	
	Estimación inicial	Real
Pronosticar el consumo por áreas administrativas	3	2

Tabla 3.12 Historias de usuario abordadas en la segunda iteración

Historias de usuario abordadas en la tercera iteración

Historias de usuario	Tiempo de estimación (semanas)	
	Estimación inicial	Real
Generar reportes de pronóstico	2	1

Tabla 3.13 Historias de usuario abordadas en la tercera iteración

Remitirse a [ANEXO 4](#) para ver las tarjetas de ingeniería.

3.4. Pruebas

En la Programación Extrema es esencial el desarrollo de las pruebas, permitiendo probar continuamente el código. Cada vez que se desea implementar las funcionalidades que tendrá el software, XP propone una redacción sencilla de prueba, para ser pasada por el código posteriormente. El proceso constante de las pruebas permite la obtención un producto con mayor calidad ofreciendo a los programadores una mayor certeza en el

trabajo que desempeñan. En la metodología XP hay dos tipos de pruebas; las unitarias o desarrollo dirigido por pruebas (TDD test driven development), 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. Con estas normas se obtiene un código simple y funcional de manera bastante rápida y eficiente. Por esto es importante pasar las pruebas al 100%.

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. Es esencia el TDD, se enfoca en la lógica del negocio y las pruebas de caja blanca en la lógica del negocio.

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 otras metodologías de desarrollo, 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. Una HU

no se considera completa hasta que no ha transitado por sus pruebas de aceptación. Luego de ver los paradigmas anteriores empleados para la realización de las pruebas y reunirse con el cliente para su análisis, el mismo decidió que se lleve 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.

Planilla de prueba de aceptación

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

Tabla 3.29 Planilla de prueba de aceptación.

Remitirse a [ANEXO 5](#) para ver las pruebas de aceptación.

En este capítulo se llevó a cabo la fase de desarrollo y diseño donde se presentó el modelo de datos de la aplicación a construir, logrando una visión detallada de sus atributos y las relaciones entre sus clases, además, las principales clases mediante el empleo de las tarjetas CRC. Se realizó el desarrollo de las iteraciones a partir de la distribución de tareas por HU, y se les practicó las pruebas de aceptación a las funcionalidades de mayor importancia.

CAPÍTULO 4 ESTUDIO DE FACTIBILIDAD

En la actualidad es de suma importancia a la hora de desarrollar un proyecto que el mismo cuente con eficacia y eficiencia a la hora de su implementación, y por supuesto se hace preciso efectuar y evaluar la factibilidad antes de su elaboración. En general los productos informáticos no están exentos de posibles riesgos en la concepción del proyecto, por lo que es válido minimizar de forma razonable recursos humanos, materiales y financieros, de ahí que es de vital importancia estimar la relación costo-beneficio, así como el esfuerzo, capital humano y el tiempo de desarrollo que se demanda en la ejecución de los mismos.

En este capítulo se expone el estudio y factibilidad del proyecto, centrado en estimaciones de esfuerzo humano, tiempo de desarrollo para su ejecución y costo. Se estiman los beneficios tangibles e intangibles que representan para el sistema propuesto, un análisis de costos y beneficios.

4.1. Evaluación Costo-Beneficio

La mayoría, por no decir todos los proyectos de informática, son evaluados según el criterio de Costo-Beneficio. Esta Metodología, plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores (Fernández, 2008):

- 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 satisfacerla 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). El desarrollo de un producto informático, siempre tiene un costo.

Este puede estar justificado por los beneficios tanto tangibles como intangibles que origina el mismo. En este proceso, se necesita de una selección adecuada de los elementos más convenientes para su evaluación.

4.2.Efectos Económicos.

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

Efectos directos

POSITIVOS:

- Los usuarios con acceso al sistema tendrán la posibilidad de crear registros confiables información de consumo energético.
- Se crean bases de información en el proceso de pronóstico del consumo energético
- Se obtendrá un resumen con los pronósticos de consumo de cada área administrativa.
- La información mostrada podrá ser guardada en formato pdf.

NEGATIVOS:

- Para el uso de esta aplicación implementada en plataforma web se necesitará que la misma sea ejecutada 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 obtuvo un producto disponible que permite en gran medida el trabajo de pronóstico de consumo energético.

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.

4.3. Elementos para identificar los Costos y Beneficios del Proyecto

Para la identificación de los costos y beneficios del proyecto que son pertinentes para su evaluación, es necesario definir una situación base o situación sin proyecto; la comparación de lo que sucede con proyecto, versus lo que hubiera sucedido sin proyecto, definirá los costos y beneficios pertinentes del mismo (María, 2009). Estos escenarios, resultan ser una herramienta determinante, puesto que ayudan en gran medida en la definición de los elementos necesarios para la evaluación.

Situación Sin Proyecto

- No se realiza el proceso de pronóstico del consumo energético.
- No se tienen en cuenta los posibles parámetros de consumo de los portadores energéticos en próximos períodos.
- No se tiene una base sólida para la realización y configuración del proceso de planificación.

Situación con Proyecto

- Se realiza el proceso de pronóstico del consumo energético.
- Se tienen en cuenta los posibles parámetros de consumo de los portadores energéticos en próximos períodos.
- Se tiene una base sólida para la realización y configuración del proceso de planificación.
- Se generan reportes de los parámetros de pronóstico.

Expuestas ambas situaciones, se procede a continuación con la identificación de los costos y beneficios del proyecto.

4.4. Factibilidad Económica

El análisis de factibilidad económica identifica los costos y beneficios asociados con el proyecto. El mismo incluye cuatro categorías (Torres, 2005):

- Costo de desarrollo
- Costos operacionales
- Beneficios tangibles
- Beneficios intangibles

4.5. Evaluación Económica

Los principales factores a considerar para el cálculo de los costos son los relacionados al personal, hardware y software, los que se pueden calcular de diversas maneras, que muchas veces se limitan al buen criterio y a la experiencia.

Para determinar el costo económico del proyecto, se desglosaron los costos en moneda libremente convertible y moneda nacional.

Costos en Moneda Libremente Convertible

Ficha de Costo		Precios (\$)
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		7.50
Subtotal		32.50
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
		0.00

Subtotal		
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	32.50

Tabla 4.1 Costos en moneda libremente convertible

Costos en Moneda Nacional

Ficha de Costo		
		Precios (\$)
Costos Moneda Libremente Convertible		
Costos Directos		
Salario del personal que laborará en el proyecto		110.00
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		6.48
Gastos en llamadas telefónicas		0.00
Gastos administrativos		0.00
Subtotal		116.48
Costos Indirectos		
Know How		
Subtotal		
	Total	116.48

Tabla 4.2 Costos en Moneda Nacional

La evaluación económica se efectúa conjuntamente con la que se puede llamar evaluación técnica del proyecto, que consiste en cerciorarse de la factibilidad técnica del mismo. En el análisis de la Factibilidad Técnica del proyecto, se pudo apreciar que:

- Se cuenta con hardware mayor a los requisitos mínimos
- Se cuenta con la disponibilidad de software

Por lo que se puede inferir que el proyecto es factible técnicamente.

Normalmente no se encuentran problemas en relación con el mercado o la tecnología disponible que se empleará en la fabricación del producto; por tanto la decisión de inversión casi siempre recae en la evaluación económica.

Los costos de los proyectos de informática son relativamente simples de cuantificar, no así los beneficios, que se presentan como ahorro de costos con respecto a la situación base (Rebaza, 2007).

4.5.1. Beneficios tangibles

Son los ingresos adicionales y/o reducción de costos que el nuevo sistema proveerá. Debido a que este proyecto no es construido con la finalidad de comercializarse, no posee ingresos monetarios perceptibles, no así con la reducción de costos, pues el sistema proveerá, mejoras en la gestión y la toma de decisiones.

Estos elementos son de muy difícil cuantificación, por lo que en esta tesis se consideran sólo como intangibles.

4.5.2. Beneficios y Costos Intangibles en el proyecto

Los beneficios intangibles, corresponden a aquellos, cuya valoración económica es difícil de obtener. En este sentido se identificaron los siguientes:

Costos

- Resistencia al cambio.

Beneficios

- El usuario puede realizar predicciones de los valores de consumo.

En este capítulo se mostró el Estudio de Factibilidad Costo-Beneficio con el cual se analizaron los efectos económicos, los beneficios y costos intangibles del proyecto. Al mismo tiempo, se calculó el costo de su ejecución, el cual arrojó como resultado la utilización de \$ 32.50 en CUC y \$ 116.48 en MN que demostraron la factibilidad del mismo.

CONCLUSIONES

Con el desarrollo de la aplicación web para la gestión energética del ISMMM, se darán cumplimiento a los objetivos propuestos en este trabajo, pues se obtendrá como resultado un producto informático en el que se aplican los resultados de la investigación realizada arribándose a las siguientes conclusiones:

- Se elaboró el marco teórico metodológico que fundamenta la investigación, permitiendo analizar la estructura del proceso de pronóstico de recursos energéticos en el ISMM.
- Se realizó el análisis para determinar las principales herramientas usadas en el desarrollo de la aplicación.
- Se creó un sistema informático que permite realizar el pronóstico del consumo de los portadores energéticos. También brinda reportes de los parámetros de pronóstico.
- Se realizó un estudio de factibilidad donde se obtuvo como resultado el tiempo de estimación para el desarrollo del sistema, el esfuerzo y la cantidad de personas necesarias para la realización del mismo, así como el costo.

RECOMENDACIONES

Por la importancia que tiene el presente trabajo para la eficiencia energética en el ISMMM se recomienda lo siguiente:

- Realizar un estudio más profundo respaldado por una mejor coordinación con el propósito de mejorar aún más el presente sistema.
- Implementar un módulo que permita generar gráficas de comparación para la evaluación del pronóstico.
- Establecer funciones evaluadoras para flexibilizar el cálculo complejo de índice de consumos deseados, facilitando al usuario establecer de forma dócil el cálculo de los mismos.

BIBLIOGRAFÍA

Achour Mehdi, y otros Manual de PHP [En línea]. - 2005. - enero de 2011. - [http://es.php.net/manual/es/..](http://es.php.net/manual/es/)

BECK K. Extreme Programming Explained. Embrace Change [Libro]. - [s.l.]: Pearson Education, 1999.

BLANCO CRIADO A. [En línea]// XAMPP. - 2008. - diciembre de 2010. - <http://www.adictosaltrabajo.com/tutoriales>.

Bureaudeprensa.com [En línea]// Embarcadero ER/Studio. - enero de 2011. - http://bureaudeprensa.com/es/view.php?bn=bureaudeprensa_software&key=115375595.

CAVSI [En línea]// ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? . - 2004. - diciembre de 2010. - [http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/..](http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/)

Ciberaula.com Ciberaula.com [En línea]// Una Introducción a APACHE. - 2005. - diciembre de 2010. - http://linux.ciberaula.com/articulo/linux_apache_intro..

CIIN Blog del CIIN (Centro de Innovación en Integración de Cantabria) [En línea]// Software As a Service (SaaS): ¿Qué es?. - 2005. - noviembre de 2010. - <http://geeks.ms/blogs/ciin/archive/2007/10/05/software-as-a-service-sas-191-qu-233-es.aspx>.

Crystal Methodologies [En línea]// Crystal Methodologies. - 2008. - enero de 2011. - <http://crystalmethodologies.blogspot.com>.

Dondo Agustín PHP en castellano [En línea]// ¿Por qué elegir PHP?. - 2005. - diciembre de 2010. - <http://www.programacion.com/php/articulo/porquephp/>.

Eaprende.com Aprende.com. [En línea]// Gestor de Base de Datos: MySQL, PostgreSQL, SQLite. - 2001. - enero de 2011. - <http://www.eaprende.com/gestor-de-basededatos-mysql-postresql-sqlite.html..>

Fernández Pérez Ramón David PRONÓSTICO DEL CONSUMO DE ENERGÍA ELÉCTRICA. ESTUDIO DE CASO EN LA UNIVERSIDAD DE CIENFUEGOS [Libro]. - Cienfuegos : UNIVERSIDAD DE CIENFUEGOS, 2008.

Fernández Y. O. P Sistema informático para la solución de problemas de programación lineal [Libro]. - [s.l.] : Instituto Superior Minero Metalúrgico de Moa (Facultad de Metalurgia - Electromecánica), 2008.

Freund J. E. Estadística Elemental Moderna [Libro]. - Arizona : [s.n.], 1960.

Graham Paul The Other Road Ahead [Libro]. - 2001.

Hervás Gómez Carlos Las nuevas tecnologías en la Educación Primaria. Internet: los procesos de enseñanza-aprendizaje con la world wide web [Libro]. - [s.l.] : Grupo de Tecnología Educativa, Universidad de Sevilla, 2006.

Htmlpoint.com Htmlpoint.com [En línea] // ¿Qué es Apache?. - 2008. - diciembre de 2010. - <http://www.htmlpoint.com/faq/apache/01.htm>.

Intercambiosvirtuales.org [En línea] // PHP Designer 2008 v6.0.2.0 Professional. - 2009. - enero de 2011. - <http://www.intercambiosvirtuales.org/software/php-designer-2008-v6020-professional>.

Islasoft.com Islasoft.com [En línea] // Diferencias entre Microsoft Access y Microsoft SQL Server/MSDE.. - 2006. - enero de 2011. - http://www.islasoft.com/Producto/Diferencias_SQL_Server_Office.pdf..

Janium.com Janium.com [En línea] // Aplicaciones basadas en Web. - 2009. - noviembre de 2010. - <http://www.janium.com/page2/page1/page6/page7/page7.html..>

JIMÉNEZ IGLESIAS E.R Sistema automatizado para la gestión del producto final de los servicios de la UEB Organización empresarial de la Empresa Empleadora del Níquel. (SGPFS) [Libro]. - [s.l.] : Instituto Superior Minero Metalúrgico , 2009.

María F. S. Rincondelvago.com [En línea] // "Evaluación de proyectos.". - 2009. - noviembre de 2010. - <http://html.rincondelvago.com/evaluacion-de-proyectos.html..>

Masternewmedia.org Masternewmedia.org [En línea] // Beneficios De Las Aplicaciones Basadas En Web Y El Anuncio De Microsoft De La Era “En Vivo”. - 2005. - diciembre de 2010. -

http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/Beneficios_De_Las_Aplicaciones_Basadas_En%20Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.htm.

Mayorga Peralta Paulina [En línea] // Gestión de Operaciones.Clase No.3 Pronóstico.. - 2009. - febrero de 2011. - <http://www.slideshare.net.html>.

Millán Tejedor Ramón Jesús [En línea] // Desarrollo de Sitios web dinámicos. - 2005. - enero de 2011. - <http://www.ramonmillan.com/documentos/dhtml.pdf>..

Monografias.com Monografias.com [En línea] // Definición arquitectura cliente servidor. - 2007. - enero de 2011. - <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>..

Monografias.com Monografias.com [En línea]. - 2008. - enero de 2011. - <http://www.monografias.com/trabajos14/modelodebase/modelodebase.shtml> ..

Neuron Corp. [En línea] // ¿Qué es UML?. - 2006. - enero de 2011. - http://www.neuronsrl.com.ar/training/uml/uml_intro.html..

Nieves Borrero Martha y Góngora Rodríguez Asnier Herramienta Informática para automatizar los procesos en el Laboratorio de Calidad de Software: Módulo Gestión de las No Conformidades [Libro]. - Ciudad de La Habana : Universidad de las Ciencias Informáticas, 2007.

Pérez Gálvez Ramón y otros EL COEFICIENTE DE RESISTENCIA AL CAMINO DETERMINADO EXPERIMENTALMENTE Y EL PRONOSTICO DEL CONSUMO DE COMBUSTIBLE [Libro]. - Cienfuegos : [s.n.], 2008.

Rebaza J. C. V. Sociedad de Estudiantes de Ciencia de la Computación SECC [En línea] // Metodología para la Priorización de Proyectos Informáticos. - 2007. - enero de 2011. - <http://www.seccperu.org/node/302>..

Revista Colombiana de Estadística [Publicación periódica]// Revista Colombiana de Estadística. - 2007. - págs. 187–202.

REYNOSO C y KICCILLOF N. [En línea]// Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. - 2007. - <http://www.willydev.net/descargas/prev/Estiloypatron.pdf>.

Ronda Amador Yoelys, Cobas Santos Kadir y Marrero Viñas Daymel El modelo cliente-servidor de tres capas para el trabajo en redes de información [Libro]. - 2002.

Torres D. [En línea]// Proyectos Informáticos. - 2005. - enero de 2011. - <http://notasprisma.tripod.com/Proyectos.htm..>

Universidad de Valparaíso Educnet-Universidad de Valparaíso [En línea]// Evaluación del lenguaje.. - 2006. - enero de 2011. - <http://educnet.decom-uv.cl/educnet/uploads/Implementacion.pdf?nombre=p379/Implementacion.pdf..>

Universidad Nacional de Lomas de Zamora [En línea]// ¿Qué es Internet?. - 2008. - diciembre de 2010. - <http://www.unlz.edu.ar/biblioteca/tutores/histoweb/datosgral.html>.

Vegas Jesús El Servidor Web [Libro]. - [s.l.] : Universidad de Valladolid, 2002.

Vico.org Vico.org [En línea]// Unified Modeling Language. - 2002. - enero de 2011. - <http://www.vico.org/FormMentorOutsourcingUML.pdf..>

Wikipedia.org Wikipedia, la enciclopedia libre [En línea]. - enero de 2011. - <http://www.eswikipedia.org>.

GLOSARIO DE TÉRMINOS

TCP/IP: Protocolo de Control de Transmisión/ Protocolo de Internet (Transmisión Control Protocol/Internet).

FTP: Protocolo de Transferencia de Archivos (File Transfer Protocol)

IRC: Protocolo de Comunicación en Tiempo Real basado en texto (Internet Relay Chat)

Web 2.0: El término Web 2.0 fue acuñado por Tim O'Reilly en 2004 para referirse a una segunda generación en la historia de la Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs y los wikis, que fomentan la colaboración y el intercambio ágil de información entre los usuarios.

RAM: Memoria de Acceso Aleatorio (Random Access Memory)

Bugs: Errores presentes en la aplicación que atentan contra su correcto funcionamiento

HTTP: Protocolo de Transferencia de Hipertexto (HyperText Transfer Protocol)

JSP: Java Server Pages (Ver Epígrafe 1.2.4)

Logs: Término para definir la creación de un fichero donde se registran las acciones de los usuarios

GTK: Conjunto multiplataforma de bibliotecas para el desarrollo de interfaces gráficas de usuarios

DBMA: Por sus siglas en inglés: Database Management System

BSD: Licencia de software libre permisiva (Berkeley Software Distribution)

CIDR: Encaminamiento Inter-Dominios sin Clases (Classless Inter-Domain Router)

IBM: International Business Machines (conocida también como el Gigante Azul)

DDL: Lenguaje de Definición de Datos (Data Definition Language)

DML: Lenguaje de Manipulación de Datos (Data Manipulation Language)

ANEXOS

ANEXO 1 MINIMOS CUADRADOS GENERALIZADOS

Obtener un modelo mínimo cuadrado que exprese explícitamente la relación de las variables independientes (y algunas de sus combinaciones) con respecto a la variable dependiente.

Para ello se siguió el desarrollo siguiente tomado, de las conferencias del Dr. Arístides Alejandro Legrá Lobaina en su curso Modelación Matemática desarrollado en la Maestría de Electromecánica del Instituto Superior Minero Metalúrgico de Moa.

Sean los n puntos P_1, P_2, \dots, P_n pertenecientes al espacio R^m . Cada punto P_i ($i=1,2, \dots,n$) tiene las coordenadas $(x_{1i}, x_{2i}, \dots, x_{mi})$ y puede suceder que dos o más de estos puntos sean iguales. Para cada punto P_i se conocen valores $Y_i \in R$ y por tanto se tiene la siguiente tabla de datos:

i	1	2	3	4	...	n
P_i	P_1	P_2	P_3	P_4	...	P_n
Y_i	Y_1	Y_2	Y_3	Y_4	...	Y_n

De lo que se trata es de encontrar un **modelo funcional lineal** que esté “cercano a todos los puntos” (P_i, Y_i) . El modelo que se busca tiene la forma general:

$$Y = a_0 + a_1 F_1(P) + a_2 F_2(P) + a_3 F_3(P) + \dots + a_k F_k(P)$$

Donde:

$$P = (x_1, x_2, \dots, x_m)$$

a_0, a_1, \dots, a_k son $k+1$ coeficientes que deben determinarse.

$F_1(P), F_2(P), \dots, F_k(P)$ son funciones definidas desde un subconjunto de R^m hasta un subconjunto de R .

El criterio que se seguirá para encontrar a los coeficientes a_0, a_1, \dots, a_k es minimizar la expresión de la llamada Varianza Residual:

$$VR = \sum_{i=1}^n \left(a_0 + a_1 F_1(P_i) + a_2 F_2(P_i) + \dots + a_k F_k(P_i) - Y_i \right)^2 \quad (1)$$

Y si consideramos que $Y_i^* = a_0 + a_1 F_1(P_i) + a_2 F_2(P_i) + \dots + a_k F_k(P_i)$, entonces,

queremos minimizar
$$VR = \sum_{i=1}^n \left(Y_i^* - Y_i \right)^2.$$

Lo cual quiere decir que trataremos de encontrar un conjunto de valores a_0, a_1, \dots, a_k tal que sea mínima la suma de los cuadrados de las diferencias entre los valores estimados Y_i^* y los valores conocidos Y_i .

Para encontrar los valores óptimos de a_0, a_1, \dots, a_k se aplicará una técnica clásica de los métodos de optimización dada por los siguientes pasos:

1. Derivar la expresión (1) con respecto a cada una de los parámetros a_0, a_1, \dots, a_k . Se obtiene

$$\frac{dU}{da_0} = 2 \sum_{i=1}^n \left(a_0 + a_1 F_1(P_i) + a_2 F_2(P_i) + \dots + a_k F_k(P_i) - Y_i \right)$$

$$\frac{dU}{da_1} = 2 \sum_{i=1}^n \left(a_0 + a_1 F_1(P_i) + a_2 F_2(P_i) + \dots + a_k F_k(P_i) - Y_i \right) F_1(P_i)$$

$$\frac{dU}{da_2} = 2 \sum_{i=1}^n \left(a_0 + a_1 F_1(P_i) + a_2 F_2(P_i) + \dots + a_k F_k(P_i) - Y_i \right) F_2(P_i)$$

...

$$\frac{dU}{da_k} = 2 \sum_{i=1}^n \left(a_0 + a_1 F_1(P_i) + a_2 F_2(P_i) + \dots + a_k F_k(P_i) - Y_i \right) F_k(P_i)$$

2. Igualando a 0 cada una de las derivadas anteriores:

$$2 \sum_{i=1}^n \left(a_0 + a_1 F_1(P_i) + a_2 F_2(P_i) + \dots + a_k F_k(P_i) - Y_i \right) = 0$$

$$2 \sum_{i=1}^n \left(a_0 + a_1 F_1(P_i) + a_2 F_2(P_i) + \dots + a_k F_k(P_i) - Y_i \right) F_1(P_i) = 0$$

$$2 \sum_{i=1}^n \left(a_0 + a_1 F_1(P_i) + a_2 F_2(P_i) + \dots + a_k F_k(P_i) - Y_i \right) F_2(P_i) = 0$$

...

$$2 \sum_{i=1}^n \left(F_0 + a_1 F_1(P_i) + a_2 F_2(P_i) + \dots + a_k F_k(P_i) - Y_i \right) F_k(P_i) = 0$$

3. Si asumimos la notación $F_0(P) = 1$, entonces el sistema de ecuaciones que se puede escribir:

$$\left[\sum_{i=1}^n F_0(P_i) F_0(P_i) \right] a_0 + \left[\sum_{i=1}^n F_1(P_i) F_0(P_i) \right] a_1 + \dots + \left[\sum_{i=1}^n F_k(P_i) F_0(P_i) \right] a_k = \left[\sum_{i=1}^n Y_i F_0(P_i) \right]$$

$$\left[\sum_{i=1}^n F_0(P_i) F_1(P_i) \right] a_0 + \left[\sum_{i=1}^n F_1(P_i) F_1(P_i) \right] a_1 + \dots + \left[\sum_{i=1}^n F_k(P_i) F_1(P_i) \right] a_k = \left[\sum_{i=1}^n Y_i F_1(P_i) \right]$$

$$\left[\sum_{i=1}^n F_0(P_i) F_2(P_i) \right] a_0 + \left[\sum_{i=1}^n F_1(P_i) F_2(P_i) \right] a_1 + \dots + \left[\sum_{i=1}^n F_k(P_i) F_2(P_i) \right] a_k = \left[\sum_{i=1}^n Y_i F_2(P_i) \right]$$

...

$$\left[\sum_{i=1}^n F_0(P_i) F_k(P_i) \right] a_0 + \left[\sum_{i=1}^n F_1(P_i) F_k(P_i) \right] a_1 + \dots + \left[\sum_{i=1}^n F_k(P_i) F_k(P_i) \right] a_k = \left[\sum_{i=1}^n Y_i F_k(P_i) \right]$$

Que en notación matricial puede escribirse como $[\mathbf{MS}](\mathbf{A})=(\mathbf{l})$ donde:

$$[\mathbf{MS}] = \begin{bmatrix} \left[\sum_{i=1}^n F_0(P_i) F_0(P_i) \right] & \dots & \left[\sum_{i=1}^n F_k(P_i) F_0(P_i) \right] \\ \left[\sum_{i=1}^n F_0(P_i) F_1(P_i) \right] & \dots & \left[\sum_{i=1}^n F_k(P_i) F_1(P_i) \right] \\ \left[\sum_{i=1}^n F_0(P_i) F_2(P_i) \right] & \dots & \left[\sum_{i=1}^n F_k(P_i) F_2(P_i) \right] \\ \dots & \dots & \dots \\ \left[\sum_{i=1}^n F_0(P_i) F_k(P_i) \right] & \dots & \left[\sum_{i=1}^n F_k(P_i) F_k(P_i) \right] \end{bmatrix}$$

$$\mathbf{(A)} = \begin{pmatrix} a_0 \\ a_1 \\ \dots \\ a_k \end{pmatrix} \quad \mathbf{(I)} = \begin{pmatrix} [\sum_{i=1}^n F_0(P_i)Y_i] \\ [\sum_{i=1}^n F_1(P_i)Y_i] \\ \dots \\ [\sum_{i=1}^n F_k(P_i)Y_i] \end{pmatrix}$$

Es importante destacar que k debe ser siempre menor o igual que n. Por otra parte es necesario aclarar que en la medida que aumenta k el sistema será más difícil de manejar por su tamaño y porque puede convertirse en un sistema mal condicionado (pequeñas variaciones de los datos pueden ocasionar grandes variaciones en los resultados de $\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_k$).

Los dos indicadores más conocidos para determinar la calidad de este modelo son:

1. El valor de $VR = \sum_{i=1}^n (\hat{Y}_i - Y_i)^2$, o sea, para los valores determinados de $\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_k$,

calcular el valor de de las sumas de las diferencias según la expresión 1.

2. El coeficiente de correlación denominado $r = \frac{\sum_{i=1}^n (\hat{Y}_i - Y_M)^2}{\sum_{i=1}^n (Y_i - Y_M)^2}$, donde $Y_M = \frac{\sum_{i=1}^n Y_i}{n}$

es la media aritmética de la variable dependiente Y. Al numerador

$VE = \sum_{i=1}^n (\hat{Y}_i - Y_M)^2$ se le llama Variación Explicada y al denominador

$VT = \sum_{i=1}^n (Y_i - Y_M)^2$ se le llama Variación Total.

Este coeficiente de correlación r tiene su valor entre 0 (peor ajuste) y 1 (mejor ajuste).

También es importante conocer:

- Error Estándar de una Estimación, $EEE = \sqrt{\frac{VE}{n-k-1}}$
- Error probable de una observación $EPO = 0,6745 \sqrt{\frac{VE}{n-k}}$

Pueden determinarse los coeficientes de correlación parcial r_1, r_2, \dots, r_k correspondientes a cada componente del modelo $F_1(P), F_2(P), \dots, F_k(P)$ y realizarse diversas pruebas de inferencia estadísticas para argumentar la validez del modelo.

ANEXO 2 HISTORIAS DE USUARIOS

HU No.2: Pronosticar el consumo eléctrico.

Historia de usuario	
Número: 2	Usuario: Energético del ISMM
Nombre: Pronosticar el consumo eléctrico.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Yanquiel Borges Matos.	
Descripción: El usuario debe insertar los datos de pronóstico correspondientes al intervalo de tiempo a pronosticar. Una vez insertados estos datos la información se registrará en el historial de pronóstico por áreas de consumo final para poder ser mostrada y actualizada.	
Observaciones: Confirmado con el cliente.	

Tabla 2.4 HU No.2

HU No.3: Pronosticar el consumo por áreas administrativas.

Historia de usuario	
Número: 3	Usuario: Responsable del consumo por tipo de actividad
Nombre: Pronosticar el consumo por áreas administrativas.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Yanquiel Borges Matos.	
Descripción: El usuario debe insertar los datos para el pronóstico, correspondientes al intervalo de tiempo a pronosticar. Una vez insertados estos datos la información se registrará en el historial de pronóstico por áreas administrativas para poder ser mostrada y actualizada.	

Observaciones: Confirmado con el cliente.

Tabla 2.5 HU No.3

HU No.4: Generar reportes de pronóstico.

Historia de usuario	
Número: 4	Usuario: Energético del ISMM
Nombre: Generar reportes de pronóstico.	
Prioridad en el negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Yanquiel Borges Matos.	
Descripción: Permite crear un reporte de pronóstico de consumo mensual en un formato de impresión.	
Observaciones: Confirmado con el cliente.	

Tabla 2.6 HU No.4

ANEXO 3 TARJETAS CRC

Tarjeta CRC No.1 para la clase PronostideTransporte

Nombre de la clase: PronostideTransporte	
Tipo de la clase: Lógica de negocio	
Responsabilidades:	Colaboradores:
mes	
año	
consumo	
indicador	
portador	
Set	
Get	

Tabla 3.1 Tarjeta CRC No.1 para la clase PronostideTransporte

Tarjeta CRC No.2 para la clase PronostideTransporteDAO

Nombre de la clase: PronostideTransporteDAO	
Tipo de la clase: Acceso a datos	
Responsabilidades:	Colaboradores:
Insertar datos	PronostideTransporte, MinimosCuadrados
Modificar	PronostideTransporte
Eliminar	PronostideTransporte
Obtener todos los pronósticos	PronostideTransporte
Obtener un pronósticos	PronostideTransporte
Existe	PronostideTransporte

Tabla 3.2 Tarjeta CRC No.2 para la clase PronosticodeTransporteDAO

Tarjeta CRC No.3 para la clase PronostideTransporteService

Nombre de la clase: PronosticodeTransporteService	
Tipo de la clase: Servicio	
Responsabilidades:	Colaboradores:
Insertar datos	PronostideTransporteDAO
Modificar	PronostideTransporteDAO
Eliminar	PronostideTransporteDAO
Obtener todas las pronósticos	PronostideTransporteDAO
Obtener un pronóstico	PronostideTransporteDAO
Existe	PronostideTransporteDAO

Tabla 3.3 Tarjeta CRC No.3 para la clase PronosticodeTransporteService

Tarjeta CRC No.4 para la Clase PronosticodeEnergia

Nombre de la clase: PronosticodeEnergia	
Tipo de la clase: Lógica de negocio	
Responsabilidades:	Colaboradores:
mes	
año	
consumo	
indicador	
portador	
Set	
Get	

Tabla 3.4 Tarjeta CRC No.4 para la clase PronosticodeEnergia

Tarjeta CRC No.5 para la clase PronosticodeEnergiaDAO

Nombre de la clase: PronosticodeEnergiaDAO	
Tipo de la clase: Acceso a datos	
Responsabilidades:	Colaboradores:
Insertar Consumo	PronosticodeEnergia, MinimosCuadrados
Modificar	PronosticodeEnergia
Eliminar	PronosticodeEnergia
Obtener todos los pronósticos	PronosticodeEnergia
Obtener un pronóstico	PronosticodeEnergia
Existe	PronosticodeEnergia

Tabla 3.5 Tarjeta CRC No.5 para la clase PronosticodeEnergiaDAO

Tarjeta CRC No.6 para la clase PronosticodeEnergiaService

Nombre de la clase: PronosticodeEnergiaService	
Tipo de la clase: Servicio	
Responsabilidades:	Colaboradores:
Insertar datos	PronosticodeEnergiaDAO
Modificar	PronosticodeEnergiaDAO
Eliminar	PronosticodeEnergiaDAO
Obtener todas las pronósticos	PronosticodeEnergiaDAO
Obtener un pronóstico	PronosticodeEnergiaDAO
Existe	PronosticodeEnergiaDAO

Tabla 3.6 Tarjeta CRC No.6 para la clase PronosticodeEnergiaService

Tarjeta CRC No.7 para la Clase PronosticodeAreaAdmin

Nombre de la clase: PronosticodeAreaAdmin	
Tipo de la clase: Lógica de negocio	
Responsabilidades:	Colaboradores:
Área admin.	
mes	
año	
consumo	
indicador	
portador	
Set	
Get	

Tabla 3.7 Tarjeta CRC No.7 para la clase PronosticodeAreaAdmin

Tarjeta CRC No.8 para la clase PronosticodeAreaAdminDAO

Nombre de la clase: PronosticodeAreaAdminDAO	
Tipo de la clase: Acceso a datos	
Responsabilidades:	Colaboradores:
Insertar datos.	PronosticodeAreaAdmin, MinimosCuadrados
Modificar.	PronosticodeAreaAdmin
Eliminar.	PronosticodeAreaAdmin
Obtener todos los datos.	PronosticodeAreaAdmin
Obtener un dato.	PronosticodeAreaAdmin
Existe.	PronosticodeAreaAdmin

Tabla 3.8 Tarjeta CRC No.8 para la clase PronosticodeAreaAdminDAO

Tarjeta CRC No.9 para la clase PronosticodeAreaAdminService

Nombre de la clase: PronosticodeEnergiaService	
Tipo de la clase: Servicio	
Responsabilidades:	Colaboradores:
Insertar datos	PronosticodeAreaAdmin DAO
Modificar	PronosticodeAreaAdmin DAO
Eliminar	PronosticodeAreaAdmin DAO
Obtener todas las pronósticos	PronosticodeAreaAdmin DAO
Obtener un pronóstico	PronosticodeAreaAdmin DAO
Existe	PronosticodeAreaAdmin DAO

Tabla 3.9 Tarjeta CRC No.9 para la clase PronosticodeAreaAdminService

ANEXO 4 TARJETAS DE INGENIERÍA

Tarjeta de ingeniería No.1: Insertar y realizar cálculo de pronóstico de actividad de transporte.

Tarea ingeniería	
Número tarea: 1	Número historia: 1
Nombre tarea: Insertar y realizar cálculo de pronóstico de actividad de transporte.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 6/5/2011	Fecha fin: 11/5/2011
Programador responsable: Yanquiel Borges Matos.	
Descripción: Esta tarea es realizada por el responsable del área de transporte.	

Tabla 3.14 Tarjeta de tarea No. 1

Tarjeta de ingeniería No.2: Eliminar pronóstico de actividad de transporte.

Tarea ingeniería	
Número tarea: 2	Número historia: 1
Nombre tarea: Eliminar pronóstico de actividad de transporte.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 12/5/2011	Fecha fin: 13/5/2011
Programador responsable: Yanquiel Borges Matos.	
Descripción: Esta tarea es realizada por el responsable del área de transporte.	

Tabla 3.15 Tarjeta de tarea No. 2

Tarjeta de ingeniería No.3: Mostrar resultado del pronóstico de actividad de transporte.

Tarea ingeniería	
Número tarea: 3	Número historia: 1
Nombre tarea: Mostrar resultado del pronóstico de actividad de transporte.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 17/5/2011	Fecha fin: 19/5/2011
Programador responsable: Yanquiel Borges Matos.	
Descripción: Esta tarea es realizada por el responsable del área de transporte.	

Tabla 3.17 Tarjeta de tarea No. 4

Tarjeta de ingeniería No.5: Insertar y realizar cálculo de pronóstico de consumo eléctrico.

Tarea ingeniería	
Número tarea: 1	Número historia: 2
Nombre tarea: Insertar y realizar cálculo de pronóstico de consumo eléctrico	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 20/5/2011	Fecha fin: 22/5/2011
Programador responsable: Yanquiel Borges Matos.	
Descripción: Esta tarea es realizada por el energético de centro.	

Tabla 3.18 Tarjeta de tarea No. 5

Tarjeta de ingeniería No.6: Eliminar pronóstico de actividad de transporte.

Tarea ingeniería	
Número tarea: 2	Número historia: 2
Nombre tarea: Eliminar pronóstico de consumo eléctrico.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 23/5/2011	Fecha fin: 24/5/2011
Programador responsable: Yanquiel Borges Matos.	
Descripción: Esta tarea es realizada por el energético de centro.	

Tabla 3.19 Tarjeta de tarea No. 6

Tarjeta de ingeniería No.8: Mostrar resultado del pronóstico de consumo eléctrico.

Tarea ingeniería	
Número tarea: 4	Número historia: 2
Nombre tarea: Mostrar resultado del pronóstico de consumo eléctrico.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 26/5/2011	Fecha fin: 27/5/2011
Programador responsable: Yanquiel Borges Matos.	
Descripción: Esta tarea es realizada por el energético de centro.	

Tabla 3.21 Tarjeta de tarea No. 8

Tarjeta de ingeniería No.9: Insertar y realizar cálculo de pronóstico por áreas administrativas.

Tarea ingeniería	
Número tarea: 1	Número historia: 3
Nombre tarea: Insertar y realizar cálculo de pronóstico por áreas administrativas.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 29/5/2011	Fecha fin: 1/6/2011
Programador responsable: Yanquiel Borges Matos.	
Descripción: Esta tarea es realizada por el energético de centro.	

Tabla 3.22 Tarjeta de tarea No. 9

Tarjeta de ingeniería No.10: Eliminar pronóstico por áreas administrativas.

Tarea ingeniería	
Número tarea: 2	Número historia: 3
Nombre tarea: Eliminar pronóstico por áreas administrativas.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 2/6/2011	Fecha fin: 3/6/2011
Programador responsable: Yanquiel Borges Matos.	
Descripción: Esta tarea es realizada por el energético de centro.	

Tabla 3.23 Tarjeta de tarea No. 10

Tarjeta de ingeniería No.12: Mostrar resultado del pronóstico por áreas administrativas.

Tarea ingeniería	
Número tarea: 4	Número historia: 3
Nombre tarea: Mostrar resultado del pronóstico por áreas administrativas.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 6/6/2011	Fecha fin: 8/6/2011
Programador responsable: Yanquiel Borges Matos.	
Descripción: Esta tarea es realizada por el energético de centro.	

Tabla 3.25 Tarjeta de tarea No. 12

Tarjeta de ingeniería No.13: Generar reporte de pronóstico de actividad de transportación.

Tarea ingeniería	
Número tarea: 1	Número historia: 4
Nombre tarea: Generar reporte de pronóstico de actividad de transportación.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 9/6/2011	Fecha fin: 11/6/2011
Programador responsable: Yanquiel Borges Matos.	
Descripción: Esta tarea es realizada por el energético de centro.	

Tabla 3.26 Tarjeta de tarea No. 13

Tarjeta de ingeniería No.14: Generar reporte de pronóstico de consumo eléctrico.

Tarea ingeniería	
Número tarea: 2	Número historia: 4
Nombre tarea: Generar reporte de pronóstico de consumo eléctrico.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 12/6/2011	Fecha fin: 13/6/2011
Programador responsable: Yanquiel Borges Matos.	
Descripción: Esta tarea es realizada por el energético de centro.	

Tabla 3.27 Tarjeta de tarea No. 14

Tarjeta de ingeniería No.15: Generar reporte de pronóstico por áreas administrativas.

Tarea ingeniería	
Número tarea: 3	Número historia: 4
Nombre tarea: Generar reporte de pronóstico por áreas administrativas.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 14/6/2011	Fecha fin: 15/6/2011
Programador responsable: Yanquiel Borges Matos.	
Descripción: Esta tarea es realizada por el energético de centro.	

Tabla 3.28 Tarjeta de tarea No. 15

ANEXO 5 Pruebas de Aceptación

Pruebas de aceptación para la HU: Pronóstico del consumo de transporte

Prueba de aceptación
HU: Pronóstico del consumo de transporte.
Nombre: Prueba para comprobar la entrada y visualización de la información de pronósticos de consumo de transporte.
Descripción: Inserción y muestra correcta de la información de pronósticos de consumo de transporte.
Condiciones de ejecución: El usuario debe insertar los datos para calcular el pronóstico.
<p>Entrada / Pasos ejecución:</p> <ul style="list-style-type: none"> • El usuario selecciona la opción de transporte. • El usuario intenta insertar los datos (mes, año, portador). • El usuario selecciona los datos que desea que se muestren.
<p>Resultado:</p> <ul style="list-style-type: none"> • Se emite un mensaje de error en caso de campos de datos vacíos. • Se emite un mensaje de error en caso de que ya existan los datos que desea calcular. • Se emite un mensaje de error en caso de que no existan los datos necesarios para realizar el cálculo. • En caso de que se inserte los datos correctamente, se realiza el cálculo y se muestra en una tabla.
Evaluación de la prueba: Aceptada.

Tabla 3.30 PA: Prueba para comprobar la entrada y visualización de la información de pronósticos de consumo de transporte.

Pruebas de aceptación para la HU: Pronóstico del consumo eléctrico

Prueba de aceptación
HU: Pronóstico del consumo eléctrico.
Nombre: Prueba para comprobar la entrada y visualización de la información de pronósticos de consumo eléctrico.
Descripción: Inserción y muestra correcta de la información de pronósticos de consumo eléctrico.
Condiciones de ejecución: El usuario debe insertar los datos para calcular el pronóstico.
Entrada / Pasos ejecución: <ul style="list-style-type: none"> • El usuario selecciona la opción de pronóstico eléctrico. • El usuario intenta insertar los datos (mes, año). • El usuario selecciona los datos que desea que se muestren.
Resultado: <ul style="list-style-type: none"> • Se emite un mensaje de error en caso de campos de datos vacíos. • Se emite un mensaje de error en caso de que ya existan los datos que desea calcular. • Se emite un mensaje de error en caso de que no existan los datos necesarios para realizar el cálculo. • En caso de que se inserte los datos correctamente, se realiza el cálculo y se muestra en una tabla.
Evaluación de la prueba: Aceptada.

Tabla 3.31 PA: Prueba para comprobar la entrada y visualización de la información de pronósticos de consumo eléctrico.

Pruebas de aceptación para la HU: Pronóstico del consumo de área administrativa

Prueba de aceptación
HU: Pronóstico del consumo de área administrativa.
Nombre: Prueba para comprobar la entrada y visualización de la información de pronósticos de consumo de área administrativa.
Descripción: Inserción y muestra correcta de la información de pronósticos de consumo de área administrativa.
Condiciones de ejecución: El usuario debe insertar los datos para calcular el pronóstico.
Entrada / Pasos ejecución: <ul style="list-style-type: none"> • El usuario selecciona la opción de pronóstico eléctrico. • El usuario intenta insertar los datos (área administrativa, mes, año, portador). • El usuario selecciona los datos que desea que se muestren.
Resultado: <ul style="list-style-type: none"> • Se emite un mensaje de error en caso de campos de datos vacíos. • Se emite un mensaje de error en caso de que ya existan los datos que desea calcular. • Se emite un mensaje de error en caso de que no existan los datos necesarios para realizar el cálculo. • En caso de que se inserte los datos correctamente, se realiza el cálculo y se muestra en una tabla.
Evaluación de la prueba: Aceptada.

Tabla 3.32 PA: Prueba para comprobar la entrada y visualización de la información de pronósticos de consumo de área administrativa.

Pruebas de aceptación para la HU: Generar reportes de pronósticos

Prueba de aceptación
HU: Generar reportes de pronósticos.
Nombre: Prueba para comprobar la generación de reportes.
Descripción: Permite obtener diferentes reportes de consumo.
Condiciones de ejecución: El usuario debe seleccionar la opción reportes de consumo.
<p>Entrada / Pasos ejecución:</p> <ul style="list-style-type: none"> • El usuario selecciona la opción. • El usuario selecciona el tipo de reporte. • Introduce los datos para generar el reporte.
<p>Resultado:</p> <ul style="list-style-type: none"> • Se visualiza la ventana de diálogo donde el cliente decide si va a guardar el reporte o si lo va a ver en ese momento. • En caso de seleccionar la opción guardar el documento es guardado correctamente. • En caso de seleccionar la opción abrir, es mostrado un documento en formato pdf con los datos del reporte.
Evaluación de la prueba: Aceptada.

Tabla 3.33 PA: Prueba para comprobar la generación de reportes.