



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

Trabajo de Diploma

En opción al título de Ingeniería Informática.

*Sistema Automatizado para el Registro y
Control de Asistencia para el Banco Popular de
Ahorro.*

Autor: Odelmi Rivera Regalado

*Tutores: Ing. Edilberto Arcaya
Consultante: Ing. Radamés San Miguel*

*Moa, Julio ,2011
"Año 53 de la Revolución."*

Declaración de Autoría

Yo, Odelmi Rivera Regalado, declaro que soy la única autora de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los _____ días del mes de Junio del 2010.

Odelmi Rivera Regalado _____

Nombre completo del primer autor

Ing. Edilberto Arcaya _____

Nombre completo del primer tutor

Ing. Radamés San Miguel _____

Nombre completo del segundo tutor

Pensamiento:



El éxito de los hombres no se mide por su éxito inmediato, sino por su éxito definitivo, no se mide por el dinero que acumularon, sino por el resultado de sus obras.

José Martí.

Agradecimientos

Primeramente agradezco a dios por haberme ayudado en todos los momentos difíciles de mi vida.

A mi amigo Yohandy que siempre me ayudó incondicionalmente.

A mis amigas Daimara y Ariannis, por estar todos estos años junto mí y ayudarme y aconsejarme cuando lo he necesitado.

A mi novio Ariel por estar a mi lado y ayudarme cuando me ha hecho falta.

A Yaniris, Yisel y Nany por haberme soportado todo este tiempo.

A los profesores Ariel Montes de Oca, Juan Manuel y Gaby por brindarme su ayuda.

A mi amiga Bettsy que siempre ha estado conmigo en los momentos buenos y malos de mi vida y me ha apoyado siempre en todas las decisiones que he tomado, ha sido mi paño de lágrimas.

A una persona que no está en estos momentos en la universidad pero aprendí mucho de ella y la admiro por lo decisiva que es, a Mailin que he vivido todos estos años con ella prácticamente y desde niñas somos una.

A mis compañeros de aula por estar siempre a mi lado cuando los necesité principalmente a Padilla, Exneyder, Reidelvis, Alexander y Calzadilla.

A mis tutores por su incondicional apoyo en todo momento, y por su paciencia y comprensión.

A Dany, Dixandra, Mar, Mari, Iris por ser tan buenos conmigo.

A mi familia que es el eslabón principal de mi carrera y todos me han ayudado cuando lo he necesitado.

A mis padres y mi hermano por guiarme por el camino correcto y por la confianza depositada.

A todos

Muchas Gracias

Dedicatoria

A mis padres queridos y mi hermano que me han guiado en todo momento y me han enseñado que la vida es de sacrificios y de pruebas.



Resumen:

La importancia que representa para todo tipo de empresa el control del personal que allí labora es innegable; pues constituye la base para el buen funcionamiento y desarrollo de la entidad, de ahí la razón de que se hayan desarrollado mecanismos para lograr un buen control: existen resoluciones, decretos, proyectos, sistemas manuales y automatizados. De estos últimos, se conoce la existencia de varios creados a nivel mundial utilizando altas tecnologías de reconocimiento de personas, de voz, utilizando tarjetas y varias variantes para tener un control estricto del movimiento de los empleados.

El objetivo del siguiente trabajo es crear un sistema informático que lleve este control y tratar de sustituir el que actualmente está funcionando en el Banco Popular de Ahorro (BPA) por las desventajas que posee dicho sistema.

Se espera que el mismo cumpla con todos los requisitos que se plantean, sirva de mejora para controlar y registrar los datos necesarios y sea un avance para el departamento de Recursos Humanos.

Summary

The importance is for all types of company control of the staff who work there is undeniable, as is the basis for the proper functioning and development of the institution, hence the reason they have developed mechanisms to achieve good control: there resolutions, decrees, plans, manual and automated systems. Of the latter, we know the existence of several established worldwide recognition using high-tech people, voice, using cards and several variants for a strict control of movement of employees.

The aim of this work is to create a computer system to take this control and try to replace the currently operating in the savings bank popular for this system has disadvantages.

It is expected that it meets all the requirements that arise, serve better to monitor and record the necessary data and is a breakthrough for the Human Resources department.

ÍNDICE

Introducción	1
Capítulo 1 – Fundamentación Teórica	5
1.1 Introducción.....	5
1.2 Estado del arte.....	5
1.3 Flujo actual de los procesos	6
1.3.1 Análisis crítico de la ejecución de los procesos.....	6
1.3.2 Sistemas automatizados vinculados al campo de acción.....	7
1.4.2 Sistemas gestores de bases de datos.....	9
1.5 Herramientas CASE.....	11
1.6 Aplicaciones Web.....	12
1.6.1 ¿Qué es una aplicación Web? ¿Por qué la utilizamos?.....	12
1.6.2 Cualidades de las aplicaciones Web.....	12
1.7 Metodologías para el desarrollo de sistemas informáticos.....	13
1.7.1 RUP.....	13
1.7.2 SXP.....	13
1.7.3 XP.....	14
1.7.3.1 Fases de la metodología XP.....	15
1.8 Arquitectura Cliente/Servidor:.....	17
1.8.1 Características de la Arquitectura Cliente/Servidor.....	18
1.9 Herramientas y Tecnologías empleadas en la propuesta de solución.....	22
1.9.1 PHP Como Lenguaje de Programación.....	22
1.9.2 MySQL Como Gestor de Base de Datos.....	22
1.9.3 Servidor para aplicaciones Web.....	23
1.9.4 Macromedia Dreamweaver (versión 8).....	23
1.9.5 Herramienta CASE utilizada.....	24
1.9.6 Metodología Utilizada.....	24
1.10 Conclusiones del Capítulo.....	25
Capítulo 2 Planeación y diseño	26
2.1 Introducción.....	26
2.2 Funcionalidades generales.....	26
2.2.1 Personal relacionado con el sistema	26
2.2.2 Lista de reserva.....	27
2.2.3 Historias de usuario.....	27

2.3	Planificación de entregas.....	29
2.3.1	Estimación de esfuerzo por historias de usuario.....	30
2.3.3	Plan de duración de las iteraciones.....	31
2.3.4	Clases, responsabilidades y colaboradores.....	32
2.4	Conclusiones.....	33
	Capítulo 3 Desarrollo y Pruebas.....	34
3.1	Introducción	34
3.2	Modelo de datos.....	34
3.3	Desarrollo de las iteraciones.....	35
3.3.1	Tareas por historias de usuario.....	35
3.3.2	Historias de usuario abordadas en la primera iteración.....	36
3.4	Pruebas.....	37
3.5	Desarrollo dirigido por pruebas.....	37
3.5.1	Pruebas de aceptación.....	38
3.6	Conclusiones.....	40
	Capítulo 4 Estudio de Factibilidad.....	41
4.1	Introducción.....	41
4.2	Efectos Económicos.....	41
4.3	Beneficios y Costos Intangibles en el proyecto.....	43
4.4	Ficha de costo.....	43
4.5	Conclusiones del Capítulo.....	47
	CONCLUSIONES GENERALES.....	48
	RECOMENDACIONES.....	49
	REFERENCIAS BIBLIOGRÁFICAS.....	49
	ANEXOS.....	54
	Anexo 1 Historias de usuarios.....	54
	Anexo 2 Tarjetas CRC.....	58
	Anexo 3 Tarjetas de tareas.....	61
	Anexo 4 Pruebas de aceptación	70
	Anexo 5 Modelo de base de datos.....	74
	Anexo 6 Vistas de algunas interfaces.....	75

Índice de Figuras

Fig.1 Metodología XP.....	17
Fig. 3 Arquitectura Cliente/Servidor.....	18
Fig. 3 Modelo de datos.....	74
Fig.4 Vista de la Interfaz del administrador.....	75
Fig.5 Vista de la Interfaz de Firmar cuando se llega tarde.....	75
Fig.6 Vista de la Interfaz de Gestionar usuarios.....	76
Fig.7 Vista de la Interfaz de Ausencias.....	76



Introducción

En la actualidad empresas de todos los tamaños requieren contar con herramientas de gestión para controlar y cuantificar los distintos aspectos de su operación. Software de aplicación para control de almacenes, inventarios, planillas, ventas, cuentas por pagar y cobrar, contabilidad, entre muchos otros son usuales en la gran mayoría de empresas independientemente de su tamaño o giro de negocio.

Una de las áreas que más ha tardado en automatizarse ha sido la del control de asistencia de los trabajadores, pues en la mayoría de los centros de trabajo se cuenta con sistemas manuales de control basados en los antiguos relojes mecánicos que utilizan tarjetas de cartón o con hojas de firma donde el trabajador firma y el mismo indica la hora en la cual entra o sale de la empresa.

Existen en la actualidad muchos sistemas de control de asistencia que se basan en tecnología de identificación automática con códigos de barra, banda magnética, tarjetas de proximidad por radio frecuencia e incluso sistemas biométricos de huella digital. Estos sistemas son de muy alto costo por lo cual Cuba se ha visto obligada a utilizar solo algunos de ellos en algunas empresas y mantenerse llevando el control mediante sistemas manuales de registro y asistencia.

En el Banco Popular de Ahorro (BPA) del municipio Sagua de Tánamo, se controla diariamente al igual que en todas las empresas del país la asistencia de los trabajadores, que deben cumplir con un horario de oficinas mediante hojas de firma. Esto permite que la empresa cuente con la información necesaria sobre la asistencia y puntualidad de los mismos.

En la actualidad los informes que se derivan del registro y control de asistencia de los trabajadores del banco no garantizan en calidad y cantidad el sistema de información necesaria para que la administración tome decisiones efectivas, pues existen ciertos problemas con estos sistemas de hojas de firma, ya que en varias ocasiones se han encontrado borrones en la hoja y tachaduras, lo que hace ineficiente este control, además este sistema permite que a veces los mismos trabajadores le firmen a sus compañeros que aún no han llegado violando reglas del ministerio de trabajo.

Atendiendo a lo antes expuesto derivamos como **problema científico** la inexistencia de una herramienta informática que garantice de manera eficiente la calidad del



proceso de registro y control de asistencia en el departamento de Recursos Humanos del Banco Popular de Ahorro (BPA). Teniendo en cuenta el problema planteado, el **objeto de estudio** se enmarca en la informatización del proceso de registro y control de asistencia, centrando el **campo de acción** en la informatización del proceso de registro y control de asistencia del departamento de Recursos Humanos del Banco Popular de Ahorro (BPA) del municipio Sagua de Tánamo.

Para darle solución al problema se plantea el siguiente **objetivo general**: Diseñar e Implementar una aplicación web que favorezca y agilice el proceso de registro y control de asistencia para el departamento de Recursos Humanos del Banco Popular de Ahorro (BPA).

Como **Idea a defender** se establece que: Si se diseña un sistema informático que registre con eficiencia y calidad la asistencia y puntualidad de los trabajadores de esta entidad, el mismo agilizará el control para el departamento de Recursos Humanos del Banco Popular de Ahorro del municipio Sagua de Tánamo.

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

- Estudiar cómo se lleva a cabo el proceso de registro y control de asistencia en el Banco Popular de Ahorro del municipio Sagua de Tánamo.
- Establecer el marco teórico de la investigación.
- Realizar un estudio de sistemas similares existentes.
- Diseñar e implementar el sistema para el proceso de registro y control de asistencia para el BPA.

Para que los objetivos específicos se lograran fue necesario plantearse las siguientes: **tareas de investigación**:

- Realizar la entrevista a la persona encargada de llevar a cabo el proceso de registro y control de asistencia en el BPA.
- Determinar las insuficiencias del sistema de control vigente en el BPA.
- Realizar una búsqueda de información sobre sistemas informáticos para el control de la asistencia y puntualidad laboral en Cuba.



- Realizar el diseño e implementación del sistema informático para el control de la asistencia y puntualidad laboral en el BPA.
- Elaborar el manual de usuario.
- Realizar el estudio de factibilidad.

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

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

Histórico-Lógico: Para la búsqueda de antecedentes del software, las herramientas utilizadas, así como la forma de cómo se llevaba a cabo el proceso de registro y control de asistencia en el banco.

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

Entre los métodos empíricos utilizados se encuentran:

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

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

El presente trabajo consta de introducción, cuatro capítulos, conclusiones, recomendaciones, referencias bibliográficas, glosario de términos y anexos:



En el capítulo 1: Fundamentación teórica, se analizan aspectos relacionados con la asistencia y puntualidad del banco, se realiza un estudio acerca de los diferentes sistemas existentes vinculados al campo de acción, además se presenta la metodología y las herramientas a utilizar en el desarrollo del sistema propuesto.

En el capítulo 2: Planeación y diseño, se hace uso de la metodología expuesta en el capítulo inicial para el desarrollo del proyecto, abordando en detalles cada una de sus fases.

El capítulo 3: Desarrollo y Pruebas, se presentan los principales métodos y definiciones dentro de la implementación de los flujos de trabajo. Se describen además las pruebas realizadas y sus resultados.

En el capítulo 4: Estudio de factibilidad y sostenibilidad, se realiza un estudio de los esfuerzos requeridos para la realización del sistema, y se valora la sostenibilidad del producto.



Capítulo 1 – Fundamentación Teórica.

1.1 Introducción.

En la actualidad, la informática ha evolucionado considerablemente y junto a ella la producción de diversos software. Las nuevas tecnologías de la informática posibilitan de manera más efectiva la atención a las diferentes ramas de los procesos empresariales, pues para la aplicación de cualquier estrategia es necesario o aconsejable apoyarse en algún software que ayude a recoger los datos históricos de los resultados obtenidos y permita optimizar la fluidez de estos (Cuenca, 2009). En este capítulo se abordarán aspectos básicos relacionados con el tema de registro y control de asistencia del personal del Banco Popular de Ahorro. Se analizarán características y beneficios de su utilización, entre otros. Estos elementos constituyen el soporte teórico del proyecto, que contribuyen al logro de un mejor entendimiento de la situación problemática y de una valoración adecuada para su solución.

1.2 Estado del arte.

Con el desarrollo de la ciencia y la digitalización las empresas a nivel mundial tienen sistemas de control de asistencia de personal, con el objetivo de tener al día todas las informaciones referentes de sus obreros o trabajadores. Llevar el control de asistencia y permanencia de los empleados en una empresa es una labor muy sencilla en algunos casos y muy complejas en otra.

Sencilla si podemos contar con una persona que tome nota del ingreso y salida del personal, es todo lo que podría necesitarse.

Compleja cuando se requiera llevar un control de los días de vacaciones tomados por un empleado particular, o también tener conocimiento exacto de los movimientos (comisiones de servicio, descanso médico, capacitación, etc). Como solución a esta necesidad existen distintos tipos de Sistemas Computarizados del Control de Asistencia, con uso de fotochecks con códigos de barras, bandas magnéticas o tarjetas de proximidad. Todos los cuales se basan en algo que el empleado posee (el fotocheck) pero que puede ser prestado, olvidado, robado o perdido y sin tener la certeza que la persona que lo presenta es el usuario auténtico (suplantación). (Borrero, 2008)



1.3 Flujo actual de los procesos.

Actualmente el proceso de firma en el Banco Popular de Ahorro (BPA) del municipio Sagua de Tánamo se realiza de forma manuscrita. Se conoce, que los trabajadores en el banco trabajan dos sesiones mañana y tarde, donde se recogen cuatro firmas dos en la sesión mañana y dos en la tarde. Los trabajadores se dirigen adonde existe una persona encargada de llevar el control de asistencia de cada trabajador.

En el banco, la firma de entrada se realiza a las (7:50) después de esta hora se traza una franja roja que indica las llegadas tardes. Luego a la hora de almuerzo se da un receso donde cada trabajador debe firmar para salir (12:20) .Para la sección de la tarde se firma a la (1:20 p.m. y 5:00 p.m.).Este registro de forma manuscrita se hace con el objetivo de elaborar el registro de tiempo de cada empleado.

1.3.1 Análisis crítico de la ejecución de los procesos.

El Control Interno ha sido preocupación de las entidades en mayor o menor grado, con diferentes enfoques y terminologías, lo que ha permitido que al pasar del tiempo se hayan planteado diferentes concepciones acerca del Control Interno, sus principios y elementos que se deben conocer e instrumentar en la entidad cubana actual.

(Borrero ,2008)

En la Resolución No. 297 dictada el 23 de septiembre del 2003 en la ciudad de La Habana se define el Control Interno como sigue:

Control Interno: El Control Interno se define ampliamente como un proceso realizado por las direcciones de las empresas y administradores para proporcionar seguridad razonable de los objetivos en las siguientes actividades:

- Confiabilidad de la información.
- Eficiencia y eficacia de las operaciones.
- Cumplimiento de las leyes, reglamentos y políticas establecidas.
- Control de los recursos, de todo tipo, a disposición de la entidad.

Además, en el acápite UNDÉCIMO de la Resolución No. 188/2006 dada en la ciudad de La Habana a los 21 días del mes de agosto, refiere cuales son las prohibiciones



comunes a todos los trabajadores. La primera de ellas expresa: “marcar las tarjeta o firmar el registro de asistencia de otro trabajador...”. (Borrero, 2008)

No obstante existe problemas con este control de la asistencia y puntualidad de los trabajadores pues se está presentando la situación en que un empleado le firme a otro indebidamente, se provoquen errores a la hora de poner la hora dejando así mucho que desear en la veracidad y calidad del proceso realizado. Además, la introducción manual de los datos en el sistema que actualmente se está llevando a cabo, está provocando retrasos en el tema de la obtención de la nómina para realizar el pago de los trabajadores del banco popular de ahorro del municipio Sagua de Tánamo debido a lo engorroso de la operación.

1.3.2 Sistemas automatizados vinculados al campo de acción.

Durante el período investigativo del presente trabajo, se conoció que en el Banco de Crédito y Comercio (BANDEC) del municipio Sagua de Tánamo presenta un sistema de registro y asistencia que controla diariamente su personal, SANRA (Sistema Automatizado de Nómina y Registro de Asistencia), este sistema fue diseñado para controlar el personal que allí labora y de ahí se confecciona la nómina de cada trabajador. Este sistema cumple con todos los requisitos, pero por políticas de la entidad y por otras razones, este se encuentra solamente en las sucursales BANDEC (Banco de Crédito y Comercio).

1.4 Tendencias y tecnologías actuales.

1.4.1 Lenguajes de programación.

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



sentencias condicionales, bucles, funciones, etc. No es un lenguaje de marcas como podría ser HTML, XML o WML. (Basulto, 2010)

A diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso permite acceder a los recursos que tenga el servidor, como por ejemplo podría ser, una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. PHP es la gran tendencia en el mundo de Internet. Últimamente se puede observar un ascenso imparable, puesto que cada día son muchas las páginas Web que lo utilizan para su funcionamiento, según las estadísticas PHP se utiliza en más de 10 millones de páginas, y cada mes realiza un aumento del 15%. Como síntesis, PHP corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unas 20 bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de aplicaciones Web muy sólidas, y contiene unas 40 extensiones estables sin contar las que se están experimentando, igualmente tiene soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros, además de que: (Basulto, 2010)

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



JavaScript

JavaScript – ES un lenguaje pensado para agregar interactividad con el usuario a las páginas HTML. Permite ejecutar secuencias de comandos en el mismo navegador del usuario. Con JavaScript se puede realizar cálculos rápidos y complejos, verificar formularios antes de enviarlos, crear calendarios, convertir divisas. Es un lenguaje que distingue entre minúscula y mayúscula, no exige la declaración explícita de las variables, es posible crear las variables. Es importante saber que JavaScript no lo soportan todos los navegadores por lo que nos vemos en la situación de probar el código resultante en más de un navegador. La sintaxis es muy parecida a C o C++, por lo que en lenguaje fácil para el que lo domine. (Hernán, 2006)

HTML

HTML significan (HiperText Markup Language, Lenguaje de Etiquetas e Hipertexto). Es el que permite saltar de una página a otra en un mismo documento o hacia otro que podría estar localizado al extremo opuesto del planeta.

El HTML no es más que una aplicación del SGML (Standard Generalized Markup Language, Lenguaje Estándar de Marcación General). Este último no es más que un sistema para definir tipos de documentos estructurados y lenguajes de marcas para representar esos mismos documentos.

HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas; además no presenta ningún compilador, por lo tanto, algún error de sintaxis que se presente éste no lo detectará y se visualizará en la forma como éste lo entienda. El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos Windows (Bloc de notas), UNIX (el editor vi o ed) o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen se deben guardar con la extensión .htm o .html. (Cuenca, 2009)

1.4.2 Sistemas gestores de bases de datos.

Un Sistema de Gestión de Bases de Datos (SGBD) es el conjunto de programas que permiten definir, manipular y utilizar la información que contienen las bases de datos, realizar todas las tareas de administración necesarias para mantenerlas operativas, mantener su integridad, confidencialidad y seguridad. Una base de datos nunca se accede ó manipula directamente, sino a través del SGBD. Se puede considerar al



SGBD como la interfaz entre el usuario y la base de datos. El funcionamiento del SGBD está muy interrelacionado con el del Sistema Operativo, especialmente con el sistema de comunicaciones. El SGBD utilizará las facilidades del sistema de comunicaciones para recibir las peticiones del usuario (que puede estar utilizando un terminal físicamente remoto) y para devolverle los resultados. Las peticiones se realizan generalmente en forma de sentencias SQL (Structure Query Lenguaje), que no es más que un lenguaje de consultas estructurado compuesto por comandos, cláusulas, operadores y funciones de agregado. (Rodríguez, 2009)

My SQL.

En el mundo de las base de datos cliente/servidor existe una feroz competencia. Muchos “grandes” compiten por ser la prestación más rápida, más segura, más confiable, más robusta. Los principales colosos de este mundo son, sin dudas Microsoft SQL Server y Oracle, y otros no tan conocidos como DB2, Sybase, Informix y Postgres. Sin embargo, MySQL no se queda atrás y desde hace poco se ha convertido en una importante competencia para estos productos, ya que cuenta con características comparables y muchas veces mejores. La empresa que desarrolla MySQL es MySQL AB, de origen sueco. (Cuenca, 2009) MySQL es un sistema de gestión de base de datos relacional, multi-hilo y multiusuario, con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL, pero, entidades que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso. Está desarrollado en su mayor parte en ANSIC. Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL está poseído y patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. (My SQL, 2008)

PostgreSQL.

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



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

1.5 Herramientas CASE.

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: (Basulto, 2010)

“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”.

- **CaseStudio:** Es una herramienta que nos posibilita relacionar las tablas de una base de datos y normalizar las mismas, con el objetivo de eliminar campos repetidos en esta. Desarrolla todo el proceso de una manera más rápida, está hecho para relacionar una gran cantidad de tablas en un pequeño período de tiempo, aunque se puede relacionar pocas tablas también. (Introducción a herramientas CASE y System Architect, 2004)
- **MYSQL Workbench 5.1 OSS:** Es una herramienta de modelado de datos, se usa para el diseño y la construcción lógica y física de bases de datos. Su ambiente es de gran alcance y multinivel. Se diseña para hacer más fácil de entender el estado actual de los datos. Simple y fácil al usuario, ayuda a organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia. (Introducción a herramientas CASE y System Architect, 2004)



1.6 Aplicaciones Web.

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

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

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

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

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

1.6.2 Cualidades de las aplicaciones Web.

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

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



Portabilidad: se enlaza el concepto de escalabilidad, y se refiere a la compatibilidad de la aplicación con múltiples sistemas operativos y plataformas utilizadas en el mercado.

En general, cuando la demanda de los usuarios crece, las aplicaciones deben cambiar de plataforma para pasar de un servidor pequeño a uno mediano o grande. Es entonces, cuando la portabilidad es crítica para poder “escalar” sin problemas independientemente del entorno que se requiera.

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

1.7 Metodologías para el desarrollo de sistemas informáticos.

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

Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. (MONMANY,2009)

Esta es la filosofía de las metodologías ágiles, las cuáles dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software. Algunas de las metodologías para el desarrollo informático son:

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

1.7.2 SXP. (Scrum + Programación. Extrema). Esta Metodología fue elaborada en la Universidad de las Ciencias Informáticas, escogieron una parte de Scrum y otra parte de XP, para una mayor organización dividen en carpetas todas sus plantillas.



1.7.3 XP .La metodología XP o Extreme Programming es una de las variantes de las metodologías ágiles con más aceptación en la comunidad internacional de desarrollo. Esta metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software. (Metodología XP, 2008)

Es la más destacada de los procesos ágiles de desarrollo de software formulada por Kent Beck. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los programadores que la practican consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Es utilizada para proyectos de corto plazo, equipo pequeño y cuyo plazo de entrega era ayer. Las características fundamentales son: (Subirós, 2009)

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Programación por parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera el código es revisado y discutido mientras se escribe es más importante que la posible pérdida de productividad inmediata.
- Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.



- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento.
- Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

1.7.3.1 Fases de la metodología XP.

Fase I: Planificación.

- 1- Se escriben historias de usuario, cuya idea principal es describir un caso de uso en dos o tres líneas con terminología del cliente (de hecho, se supone que deben ser escritos por el mismo), de tal manera que se creen test de aceptación para historias de usuarios (user storie) y permita hacer una estimación de tiempo de desarrollo del mismo.
- 2- Se crea un plan de lanzamiento (release planning), que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto. Se usa como base las historias de usuario, participando el cliente en la elección de las que se desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto.
- 3- El desarrollo se divide en iteraciones, cada una de las cuales comienzan con un plan de iteración, para el que se eligen las historias de usuario a desarrollar y las tareas de desarrollo.
- 4- Se cambia el proceso cuanto sea necesario, para adaptarlo al proyecto.



Fase II: Diseño.

- 1- Se eligen los diseños funcionales más simples.
- 2- Se elige una metáfora del sistema para que el nombrado de clases, siga una misma línea, facilitando la reutilización y la comprensión del código.
- 3- Se escriben tarjetas de clase-responsabilidades-colaboración (CRC) para cada objeto, que permitan abstraerse al pensamiento estructurado y que el equipo de desarrollo completo participe en el diseño.

Fase III: Codificación.

- 1- El cliente está siempre disponible, de ser posible, cara a cara. La idea es que forme parte del equipo de desarrollo, y esté presente en todas las fases de XP. La idea es usar el tiempo del cliente para estas tareas en lugar de crear una detallada especificación de requisitos, y evitar la entrega de un producto insuficiente, que le hará perder tiempo.
- 2- El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código.
- 3- Las pruebas unitarias se codifican antes que el código en sí, haciendo que la codificación de este último sea más rápida, y que cuando se afronte la misma se tenga más claro, qué objetivos tiene que cumplir lo que se va a codificar.
- 4- La programación del código se realiza en parejas, para aumentar la calidad del mismo. En cada momento, sólo habrá una pareja de programadores que integre código.
- 5- Se integra código y se lanza dicha integración de manera frecuente, evitando divergencias en el desarrollo y permitiendo que todo el mundo trabaje con la última versión del desarrollo. De esta manera, se evitará pasar grandes períodos de tiempo integrando el código al final del desarrollo, ya que las incompatibilidades serán detectadas enseguida.
- 6- Se usa la propiedad colectiva del código, lo que se traduce en que cualquier programador puede cambiar cualquier parte del código. El objetivo es fomentar la contribución de ideas por parte de todo el equipo de desarrollo.
- 7- Se deja la optimización para el final.
- 8- No se hacen horas extra de trabajo.

Fase IV: Pruebas.

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

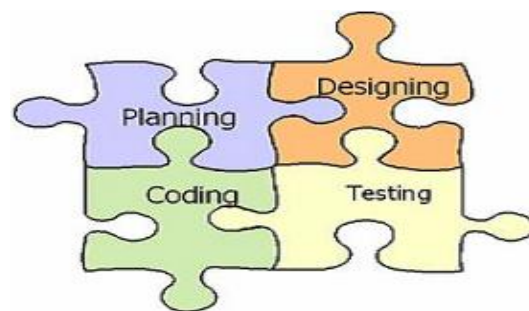


Fig.1 Metodología XP.

1.8 Arquitectura Cliente/Servidor

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

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

1.8.1 Características de la Arquitectura Cliente/Servidor.

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

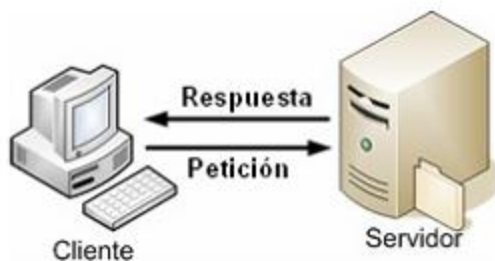


Fig. 3 Arquitectura Cliente/Servidor.

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



modelo. Se puede decir que la arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información, estableciendo así un enlace de comunicación transparente entre los elementos que conforman la estructura. No existe una definición específica adoptada universalmente de la Arquitectura Cliente/Servidor, las empresas de cómputo enfocan el concepto basándose en la funcionalidad que representa según los servicios que ellas mismas ofrecen. (Renaud, 1996)

Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes:

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

Todos los sistemas desarrollados en arquitectura Cliente/Servidor poseen las siguientes características distintivas de otras formas de software distribuido. (Renaud, 1996):

- Servicio: El servidor es un proveedor de servicios; el cliente es un consumidor de servicios.
- Recursos compartidos: Un servidor puede atender a muchos clientes al mismo tiempo y regular su acceso a recursos compartidos.
- Protocolos Asimétricos: La relación entre cliente y servidor es de muchos a uno; los clientes solicitan servicios, mientras los servidores esperan las solicitudes pasivamente.
- Transparencia de ubicación: El software Cliente/Servidor siempre oculta a los clientes la ubicación del servidor.
- Mezcla e igualdad: El software es independiente del hardware o de las plataformas de software del sistema operativo; se puede tener las mismas o diferentes plataformas de cliente y servidor.



- Intercambio basados en mensajes: Los sistemas interactúan a través de un mecanismo de transmisión de mensajes: la entrega de solicitudes y respuestas del servicio.
- Encapsulamiento de servicios: Los servidores pueden ser sustituidos sin afectar a los clientes, siempre y cuando la interfaz para recibir peticiones y ofrecer servicios no cambie.
- Facilidad de escalabilidad: Los sistemas Cliente/Servidor pueden escalarse horizontal o verticalmente. Es decir, se pueden adicionar o eliminar clientes (con apenas un ligero impacto en el desempeño del sistema); o bien, se puede cambiar a un servidor más grande o a servidores múltiples.
- Integridad: El código y los datos del servidor se conservan centralmente; esto implica menor costo de mantenimiento y protección de la integridad de los datos compartidos. Además, los clientes mantienen su individualidad e independencia.

La arquitectura Cliente/Servidor es una infraestructura versátil modular y basada en mensajes que pretende mejorar la portabilidad, la interoperabilidad y la escalabilidad del cómputo; además es una apertura del ramo que invita a participar a una variedad de plataformas, hardware y software del sistema . (Renaud, 1996)

Cliente

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

- Manejo de la interfaz del usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Como ejemplos de clientes pueden citarse interfaces de usuario para enviar comandos a un servidor, APIs para el desarrollo de aplicaciones distribuidas, herramientas en el



cliente para hacer acceso a servidores remotos (por ejemplo, servidores de SQL) o aplicaciones que solicitan acceso a servidores para algunos servicios.

Servidor

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

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

Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en computadoras personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo. En el servidor permanecen las aplicaciones que deben ser compartidas por varios usuarios. Normalmente, aunque con excepciones, el cliente y el servidor están ubicados en distintos procesadores; incluso, un servidor puede fungir como cliente de otros servidores. Existen diversos servidores mismos que se clasifican basándose en su funcionalidad; estos son denominados servidores dedicados ya que administran el uso de algún recurso en particular, por ejemplo, Servidor de Archivos: El cliente envía solicitudes de registros de archivos al servidor, es un simple servicio de datos compartidos por medio de la red. Servidor de bases de datos: El cliente envía solicitudes de SQL en calidad de mensajes (un mensaje por instrucción); el servidor



hace uso de su propia capacidad de procesamiento para encontrar los datos solicitados y devolverlos por medio de la red. Servidores Web: Se usan como una forma inteligente para comunicación entre empresas a través de Internet. Este servidor permite transacciones con el acondicionamiento de un browser específico. Este modelo está integrado por clientes compactos y portátiles en comunicación con servidores amplios. Tal comunicación se da mediante un protocolo denominado HTTP del inglés Hypertext Transfer Protocol.

1.9 Herramientas y Tecnologías empleadas en la propuesta de solución.

1.9.1 PHP Como Lenguaje de Programación.

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

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

1.9.2 MySQL Como Gestor de Base de Datos.

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

- No se necesitará de un manejo complejo de la información.



- El PHP maneja más fácil al MySQL que a PostgreSQL, debido a la gran cantidad de funciones que tiene explícitas.
- El MySQL tiene mayor rendimiento y mayor velocidad mientras PostgreSQL es de 2 a 3 veces más lento que MySQL.
- Mejores utilidades de administración (backup, recuperación de errores, etc.).
- No suele perder información ni corromper los datos.
- No hay límites en el tamaño de los registros, PostgreSQL tiene un límite del tamaño de cada fila de las tablas a 8k.
- Mejor control de acceso, en el sentido de que usuarios tienen acceso a que tablas y con que permisos.
- MySQL se comporta mejor que PostgreSQL a la hora de modificar o añadir campos a una tabla "en caliente".
- MySQL por su bajo consumo lo hace apto para ser ejecutado en una máquina con escasos recursos sin ningún problema, PostgreSQL consume más recursos y carga más el sistema.

1.9.3 Servidor para aplicaciones Web.

XAMPP - Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los interpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de **X** (para cualquiera de los diferentes sistemas operativos), **A**pache, **M**ySQL, **P**HP, **P**erl. El programa esta liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP esta disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.. XAMPP es regularmente actualizado para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. Incluye otros módulos como OpenSSL, y phpMyAdmin.(citado en http://wikipedia.uo.edu.cu/es/articles/x/a/m/XAMPP_f875.html.)



1.9.4 Macromedia Dreamweaver (versión 8).

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

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

Permite al usuario utilizar la mayoría de los navegadores Web instalados en su ordenador para previsualizar las páginas web. También dispone de herramientas de administración de sitios dirigidas a principiantes como, por ejemplo, la habilidad de encontrar y reemplazar líneas de texto y código por cualquier tipo de parámetro especificado, hasta el sitio web completo. (Hernán ,2006)

1.9.5 Herramienta CASE utilizada.

MYSQL Workbench 5.1 OSS- Después de un estudio de las herramientas que se abordaron anteriormente se escogió esta ya que nos proporciona muchas ventajas como las siguientes ventajas:

- Si se está comenzando un nuevo diseño o está manteniendo una base de datos existente, se combina con las características para ayudarle a conseguir el trabajo hecho con eficacia.
- La creación de diagramas es clara y rápida.
- Tiene la posibilidad de realizar diagramas con desempeño rápido.
- Te da la posibilidad de darle un nombre a la Base de Datos ante de exportarla.

1.9.6 Metodología Utilizada.

Actualmente XP es la metodología ágil más documentada (hay una colección de libros “XP Series” de Addison Wesley) y extendido. Existe una gran comunidad de desarrolladores XP. Otra de las ventajas de XP es que no es necesario adoptarlo en forma completa, sino que pueden utilizarse varias de sus prácticas en forma independiente. Esto hace que el costo de su implementación sea mucho más accesible que el de otras metodologías. Algunas de las ventajas que tiene 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.
- 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.
- Se realizan pruebas constantemente del sistema.

1.10 Conclusiones del Capítulo.

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



Capítulo 2 Planeación y diseño.

2.1 Introducción.

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, se presentan además las tarjetas, clases, responsabilidades y colaboradores, que permitirán trabajar con una metodología basada en objetos.

2.2 Funcionalidades generales.

El control del personal en el BPA del municipio Sagua de Tánamo se realiza de forma manual, lo que hace el proceso más lento, con el consiguiente aumento de gastos de papel y otros materiales de oficina e incrementa la posibilidad de que se produzcan errores en el momento de manipular la información de los mismos, lo que trae como consecuencia afectaciones en la elaboración de la pre Nómina pues a veces ocurren errores y hay que hacer una nueva pre Nómina.

2.2.1 Personal relacionado con el sistema

Personas relacionadas con el sistema	Justificación
Especialista de Recursos Humanos	Es la persona que tiene conocimiento de como se lleva a cabo el proceso de asistencia y puntualidad en la entidad.
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 Personas relacionadas con el sistema.



2.2.2 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.

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

- 1-) Autenticar usuarios.
- 2-) Insertar usuarios.
- 3-) Eliminar usuarios.
- 4-) Mostrar listado de los usuarios.
- 5-) Modificar contraseña del empleado.
- 6-) Modificar nivel de acceso.
- 7-) Modificar usuarios.
- 8-) Registrar hora de entrada y salida
- 9-) Mostrar o visualizar horario de llegada y salida.
- 10-) Mostrar fecha actual.
- 11-) Mostrar historial diario por usuario.
- 12-) Mostrar el historial del empleado por mes.
- 13-) Controlar las ausencias al trabajo.
- 14-) Exportar datos de asistencia del mes.
- 15-) Cambiar Imagen.

2.2.3 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. (Basulto, 2010)

Modelo de planilla de historias 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: Autenticar usuarios.

Historia de usuario	
Número: 1	Usuario: Especialista de Recursos Humanos
Nombre: Autenticar usuarios.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Odelmi Rivera Regalado	
Descripción: Los usuarios del Sistema ingresan sus datos para entrar al sistema (nombre de usuario y contraseña). El sistema verifica que los datos estén correctos, en caso de que no sean correctos los mismos la aplicación muestra un mensaje de error.	
Observaciones: Confirmado con el cliente.	

-Tabla 2.3 HU No.1: Autenticar usuarios.

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

2.3 Planificación de entregas.

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

Las estimaciones asociadas a la implementación de las historias se establecen empleando como medida el punto de estimación. Un punto de estimación equivale a una semana ideal de programación, donde los miembros de los equipos de desarrollo, trabajan el tiempo planeado sin ningún tipo de interrupción, este punto de estimación que se utiliza para representar la semana ideal, es de 5 días. Las historias generalmente tienen un valor de 1 a 3 puntos. Además, se mantiene un registro de la velocidad de desarrollo, establecida por puntos de iteración, basado fundamentalmente en la suma de los puntos de estimación correspondientes a las HU, que fueron terminadas en la última iteración. (Basulto, 2010)



2.3.1 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:

Historias de usuario	Puntos de estimación
Autenticar usuarios.	1 día.
Gestionar usuarios.	3 semanas.
Gestionar hora de entrada y salida.	2 semanas.
Mostrar fecha actual.	1 semana.
Mostrar el historial del empleado.	2 semanas.
Exportar datos de asistencia del mes.	1 semana.
Controlar las ausencias al trabajo.	2 semanas.
Cambiar Imagen.	1 día.

Tabla 2.11 Estimación de esfuerzo por historias de usuario.

2.3.2 Planificación de iteraciones.

A partir de las HU antes expuestas y la estimación del esfuerzo propuesto para la realización de las mismas, se procede a realizar la planificación de la etapa de implementación del sistema, apoyándose en el tiempo e intentando concentrar las funcionalidades relacionadas en una misma iteración. En este plan se establece cuantas 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:

Esta iteración tiene como objetivo darle cumplimiento a las HU que se consideraron de mayor importancia para el desarrollo de la aplicación. Al concluir dicha iteración se contará con todas las funcionalidades descritas en la HU 1, 2, 3 y 4 las cuales hacen alusión a la autenticación de los usuarios, gestión de los usuarios, al registro y visualización de la hora, y a mostrar la fecha actual.

Segunda iteración:

Esta iteración tiene como finalidad desarrollar las HU 5 y 6. La misma brinda la funcionalidad de mostrar el historial del empleado y de poder exportar datos de asistencia de los empleados en el mes.

Tercera iteración:

Esta última iteración del módulo tiene como propósito llevar a cabo el desarrollo de la HU 7 y 8 que permite que el encargado pueda controlar las ausencias de los trabajadores y se pueda cambiar la imagen que existe actualmente en el sistema.

Estas HU serán integradas 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.

2.3.3 Plan de duración de las iteraciones.

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



Iteración	Historias de usuarios	Duración Total
Iteración 1	Autenticar usuarios Gestionar usuarios. Gestionar horario de entrada y salida. Mostrar fecha actual.	5 semanas y 1 día
Iteración 2	Mostrar el historial del empleado. Exportar datos de asistencia del mes.	4 semanas
Iteración 3	Controlar las ausencias al trabajo. Cambiar Imagen	2 semanas y 1 día

Tabla 2.12 Plan de duración de las iteraciones.

2.3.4 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.



Tarjeta CRC No.1 Autenticar usuarios.

Nombre de la clase: Autenticar usuarios	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Autenticar Usuarios	Conexión

Tabla 2.13 Tarjeta CRC No.1

Para consultar el resto de las tarjetas CRC ir: [Anexo 2](#)

2.4 Conclusiones.

En este capítulo se abordó la fase de planeación y diseño donde se delinearon las HU con la participación del cliente, se llevó a efecto la planificación de iteraciones de cada HU a partir de la estimación del esfuerzo necesario de las mismas, se realizaron la tarjetas CRC culminando así esta fase y se determina que el trabajo está listo para pasar a la siguiente etapa de desarrollo.



Capítulo III - Desarrollo y pruebas.

3.1 Introducción.

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

3.2 Modelo de datos.

En esta parte se muestra el modelo de datos empleado para la aplicación. En la gestión de usuarios es empleada la tabla usuarios, donde se almacenan los datos de los usuarios y se registran los datos necesarios. En la tabla asistencia es donde se guarda todo lo referente a la asistencia de los empleados, hora de entrada y salida, fecha entre otras características. Para la justificación de las llegadas tardes se utiliza la tabla justificación, ahí se va a recoger todas las justificaciones de los empleados.

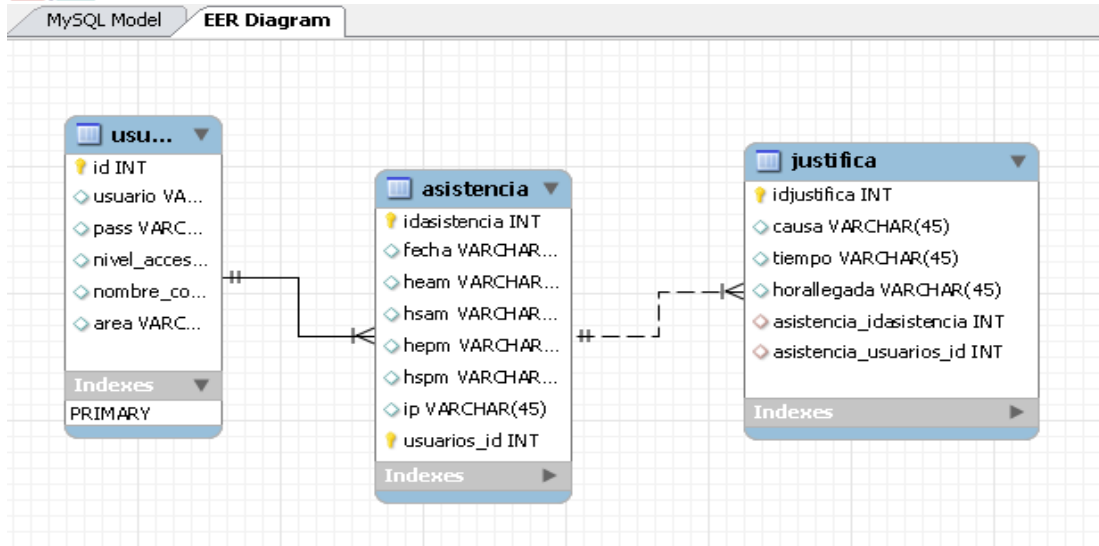


Fig.3.2 Modelo de datos

3.3 Desarrollo de las iteraciones.

Durante la fase planificación y diseño fueron detalladas las HU 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.

Autenticar usuarios.	Insertar los datos para entrar al Sistema.
Gestionar usuarios.	Insertar usuarios. Eliminar usuarios. Mostrar listado. Modificar contraseña. Modificar nivel de acceso. Modificar usuario.
Gestionar horario de entrada y salida de los usuarios.	Registrar hora de entrada y salida. Mostrar o visualizar horario de entrada y salida.
Exportar datos de asistencia del mes.	Exportar datos.
Mostrar fecha actual.	Mostrar fecha.
Mostrar el historial del empleado.	Mostrar historial diario por usuario. Mostrar historial por mes.
Controlar ausencias al trabajo.	Controlar ausencias.
Cambiar Imagen.	Cambiar Imagen.

Tabla3.1 Distribución de tareas por historia de usuario.

3.3.2 Historias de usuario abordadas en la primera iteración.

Historias de usuario	Tiempo de estimación (semanas)
Autenticar Usuarios	1 día
Gestionar Usuarios	2 semanas
Gestionar hora de entrada y salida.	2 semanas
Mostrar fecha actual	1 semana

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



Tarjeta de tarea No.1: Autenticar Usuario.

Tarea ingeniería	
Número tarea: 1	Número historia: 1
Nombre tarea: Autenticar Usuario.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha inicio: 2/2/2011	Fecha fin: 3/2/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea facilita que los usuarios se puedan autenticar y entrar al sistema.	

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

Para consultar el resto de las tareas ir: [Anexo3](#)

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 de un producto con mayor calidad, y se ofrece 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, se verifica 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%.

3.5 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. En esencia el TDD y las pruebas de caja blanca se enfocan en la lógica del negocio.

3.5.1 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, solo 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 la HU 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 3.14 Planilla de prueba de aceptación.



Pruebas de aceptación para la HU: Autenticar usuarios.

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

Tabla 3.15 Pruebas de aceptación para la HU: Autenticar usuarios.

Para consultar el resto de las pruebas ir: [Anexo 4](#)

3.6 Conclusiones.

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



Capítulo 4 Estudio de factibilidad

4.1 Introducción

Para estudiar la factibilidad de este proyecto se utilizará la **Metodología Costo Efectividad (Beneficio)**, la cual plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores:

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

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

4.2 Efectos Económicos

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

Efectos directos:

POSITIVOS:

- Los usuarios autorizados a manipular el sistema tendrán la cobertura de poder registrarse y entrar al sistema para poder llevar un control de ellos.
- Se mejora la eficiencia del proceso de registros y control de asistencia para el departamento de Recursos Humanos de Banco Popular de Ahorro.



NEGATIVOS:

- Para el uso de la esta aplicación implementada con tecnología Web se necesitará que la misma sea ejecutada con el navegador de 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 navegadores.

Efecto indirecto:

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

Efecto Externo:

- Se tendrá una herramienta disponible que le facilitará gran parte del trabajo al encargado de llevar a cabo el proceso de registro y control de asistencia en el Banco Popular de Ahorro.

Intangibles:

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

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

- **SITUACIÓN SIN PROYECTO**

En el Banco Popular de Ahorro se controla de manera manual la asistencia de los trabajadores, que deben cumplir con un horario de oficinas mediante hojas de firma. Cada trabajador cuando llega al banco debe dirigirse adonde existe una persona encargada de llevar a cabo este proceso y firmar en una hoja de firma su entrada y así sucesivamente cada vez que vaya a salir o a entrar al banco.



• SITUACIÓN CON PROYECTO

Mediante este sistema es posible lograr un mejor control sobre el personal que allí labora y que pueda ser accedida así como clasificada en base a los requerimientos del BPA. Los usuarios con solo entrar al sistema podrán firmar su entrada y salida de manera digital. En dependencia del usuario que sea es decir si es el encargado, va a tener la posibilidad de ver las llegadas tardes en el día, de poder ver el historial de un empleado en particular, así como de realizar las operaciones comunes de un administrador como insertar, modificar entre otras operaciones.

4.3 Beneficios y Costos Intangibles en el proyecto

Costos

- Resistencia al cambio.

Beneficios

- Mejor comodidad para el departamento de Recursos Humanos.
- Menor tiempo empleado en la introducción de los datos.
- Mayor rapidez en la búsqueda de información.

4.4 Ficha de costo

Para determinar el costo económico del proyecto se utilizará el procedimiento para elaborar Una Ficha de Costo de un Producto Informático [Pérez García].

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

Costos en Moneda Librementemente Convertible:

Tabla: Costo en Moneda Librementemente Convertible

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

Costos en Moneda Nacional:

Tabla: Costo en Moneda Nacional.

Ficha de Costo.	
	Precio(s)
Costos Moneda Nacional	
Costos Directos	
Salario del personal que laborará en el proyecto	100,00
12,5% del total de gastos por salarios se dedica a la seguridad social	0,00
9.09% de salario total, por concepto de vacaciones a acumular	0,00
Gasto por consumo de energía eléctrica	170,70
Gastos en llamadas telefónicas	0,00
Gastos administrativos	0,00
Subtotal	270,70
Costos Indirectos	
Know How	0,00
Subtotal	
Total	270,70

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



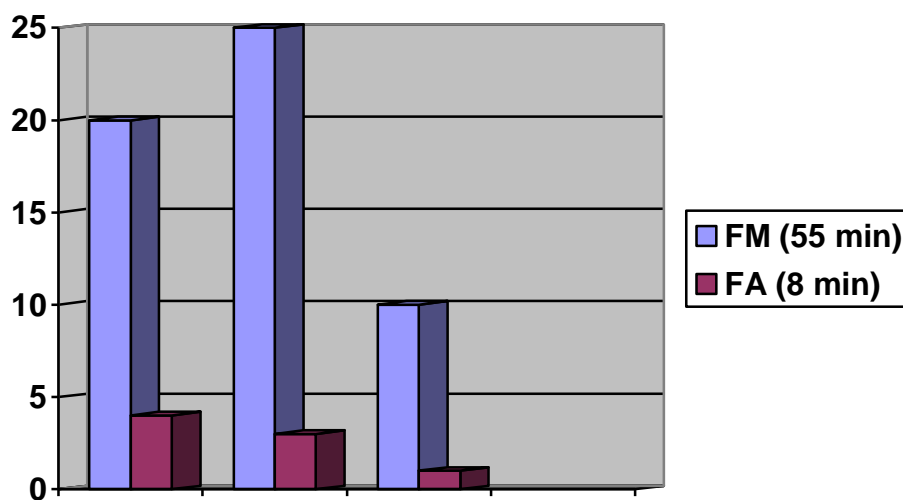
Valores de la variable (Solución manual):

- i. Prueba durante el tiempo de realización de las firmas (20 min.)
- ii. Buscar un historial mensualmente de cada trabajador (25 min)
- iii. Buscar asistencia diaria de cada trabajador (10 min)

Valores de la variable (Solución con el software):

- iv. Realizar firmas (4 min).
- v. Buscar historial de un trabajador mensualmente (3 min).
- vi. Buscar asistencia (1 min).

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



FM – Forma manual.

FA – Forma automatizada.

Teniendo en cuenta los resultados reflejados en la gráfica en cuanto al Punto de Equilibrio queda demostrada la factibilidad del sistema evidenciado por la relación entre la complejidad del problema (cantidad de fuentes y receptores) y el tiempo que demora la introducción de los datos de forma manual y automatizada.



4.5 Conclusiones del Capítulo

En este capítulo se hace un estudio profundo del costo real en que se incurrió durante el diseño e implementación del producto software mediante la Metodología Costo Efectividad (Beneficios), se analizaron todos los factores directos, indirectos, externos e intangibles, además se calculó el costo de ejecución del producto software mediante la ficha de costo arrojando como resultados **(25,00CUC y 270,70 CUP)** demostrándose la conveniencia de la elaboración del sistema.



CONCLUSIONES GENERALES

El sistema se desarrolla siguiendo la metodología XP, y se utilizaron representaciones para la modelación de todas las fases del proyecto. El sistema resultante está provisto de un ambiente cómodo, fácil de entender, que cumple los estándares de diseño y utiliza técnicas modernas de programación orientada a objetos, para esto se siguieron los siguientes pasos:

- Se elaboró el marco teórico metodológico que fundamenta la investigación, permitiendo el análisis del proceso de control de asistencia y puntualidad en el BPA para lograr una mejor comprensión del negocio.
- Se efectuó un estudio de las diferentes tecnologías y herramientas para la confección del sistema, haciendo una elección, de acuerdo a las especificaciones del cliente.
- Se efectuó un levantamiento de los requerimientos, los que propiciaron un mejor entendimiento de la problemática en cuestión, facilitando el análisis, diseño e implementación del sistema automatizado, el cual solucionó las dificultades antes expuestas.
- Se obtuvo un sistema informático que permite el control de asistencia de los trabajadores del BPA.

Por todo lo anterior se concluye que los objetivos propuestos en el presente proyecto han sido cumplidos satisfactoriamente.



RECOMENDACIONES

De manera general los objetivos trazados al inicio de esta investigación han sido logrados, al mismo tiempo en el transcurso del proceso de desarrollo, ha quedado evidenciado, que la propuesta es solo la primera fase de un proyecto que puede ser mucho más ambicioso. Por tanto se hacen las siguientes recomendaciones.

- Continuar trabajando en el sistema automatizado de registro y control de asistencia con el objetivo de integrarlo en un futuro con un sistema que elabore la nómina de los trabajadores.
- Agregar nuevas funcionalidades acorde a nuevos requisitos que pudiesen surgir por alguna causa o para aumentar el rendimiento del Sistema.
- Realizar un estudio más profundo de este sistema en vista a perfeccionarlo en versiones futuras.



REFERENCIAS BIBLIOGRÁFICAS.

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

Borrero Lagué, Luís Enrique. *Sistema de reporte y control de asistencia*. Moa, 2008.

Basulto, Jorge Mario. *Sistema de Gestión integral de la empresa Empleadora del NIQUEL "EMPLENI". MODULO GESTION DE CONTRATOS DE COMPRAS*. Moa, 2010.

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

Introducción a herramientas CASE y System Architect. Universidad Politécnica de Valencia. [Consultado: 2011-01-15], 2004. Disponible en: http://www.dsic.upv.es/asignaturas/eui/mtp/doc-practicas/intro_case_SA.pdf.

Metodología XP [en línea], 2008. [Consultado: 2011-04-10]. http://wikipedia.uo.edu.cu/es/articles/p/r/o/Programación_Extrema_3b63.html.

MONMANY, J. *Aplicaciones Web*. [Consultado: 2011-01-10]. Disponible en: <http://www.webvillage.info>.

MySQL. [en línea], 2008. [Consultado: 2011-02-20]. Disponible en: http://wikipedia.uo.edu.cu/es/articles/m/y/s/MySQL_62a0.html

PÉREZ GARCÍA, A. M. *Procedimientos para la elaboración de la ficha de costo de un producto informático*. Facultad MFC UCLV. Villa Clara.

Renaud Paul E. (2nd ed). (1996) *Introduction to client/server systems. A practical guide for systems professionals*. John Wiley & Sons New York, USA.



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

Romero Cuza, Edisvel. *Sistema Automatizado para la recuperación de información en entornos virtuales basado en perfiles de usuarios*. Moa, 2010.

Subirós Muñoz, Dariel Raúl. *Desarrollo de una interfaz gráfica de usuario para el preprocesador meteorológico AERMET*”, 2009.

XAMPP [Consultado: 2011-03-25]. Disponible en:

http://wikipedi.uo.edu.cu/es/articles/x/a/m/XAMPP_f875.html.



GLOSARIO DE TÉRMINOS

Aplicación Web (Web-Application): Es un software o sistema que está basado en las especificaciones de la World Wide Web Consortium (W3C) y provee diferentes recursos como contenidos y servicios, los cuales se usan a través de una interfaz de usuario conocida como navegador Web.

Arquitectura: Conjunto de decisiones significativas acerca de la organización de un sistema software. La arquitectura del software se interesa no solo por la estructura y el comportamiento, flexibilidad al cambio, reutilización, comprensión, economía y tecnología, así como por aspectos estéticos.

Base de Datos: Es un conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que una base de datos puede considerarse una colección de datos variables en el tiempo.

HTML: Hyper Text Markup Language, o simplemente HTML, es un lenguaje de etiquetas de programación muy sencillo que se utiliza para crear los textos y las páginas web. Si se hace la traducción de su nombre del inglés al castellano, sería “Lenguaje de Marca de Hipertextos”, ya que es justamente un lenguaje que se basa en las marcas para crear los hipertextos.

JAVA: Es un lenguaje de programación diseñado para ser utilizado en la red, no está instalado localmente y se utiliza para dar a las páginas Web características extras no disponibles en HTML.

MVC: Modelo Vista Controlador. Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio.



Programación Extrema(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.

RUP: El Proceso Unificado Racional o RUP (Rational Unified Process), es un proceso desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos y roles.

XAMPP: Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de **X** (para cualquiera de los diferentes sistemas operativos), **A**pache, **M**ySQL, **P**HP, **P**erl. 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, Linux, Solaris, y MacOSX.



ANEXOS

Anexo 1 Historias de usuarios.

Tabla 2.3 HU No.1: Autenticar usuarios.

Historia de usuario	
Número: 1	Usuario: Especialista de Recursos Humanos
Nombre: Autenticar usuarios.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Odelmi Rivera Regalado	
Descripción: Los usuarios del Sistema ingresan sus datos para entrar al sistema (nombre de usuario y contraseña). El sistema verifica que los datos estén correctos, en caso de que no sean correctos los mismos la aplicación muestra un mensaje de error.	
Observaciones: Confirmado con el cliente.	

Tabla 2.4 HU No.2 Gestionar usuarios.

Historia de usuario	
Número: 2	Usuario: Especialista de Recursos Humanos
Nombre: Gestionar usuarios.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Odelmis Rivera Regalado	
Descripción: El administrador es el encargado de insertar los usuarios existentes en el sistema. Una vez insertado los usuarios, se podrá eliminar, modificar, cambiar la contraseña, mostrar un listado de los mismos y modificar el nivel de acceso.	
Observaciones: Confirmado con el cliente.	



Tabla 2.5 HU No.3: Gestionar horario de entrada y salida.

Historia de usuario	
Número: 3	Usuario: Especialista de Recursos Humanos
Nombre: Gestionar horario de entrada y salida.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Odelmis Rivera Regalado	
Descripción: Cuando el usuario entre al sistema y salga se debe registrar la hora de entrada y salida y además de debe mostrar en el sistema.	
Observaciones: Confirmado con el cliente.	

Tabla 2.6 HU No.4: Mostrar fecha actual.

Historia de usuario	
Número: 4	Usuario: Especialista de Recursos Humanos
Nombre: Mostrar fecha actual.	
Prioridad en el negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Odelmis Rivera Regalado	
Descripción: Cuando el empleado entre al sistema se debe mostrar la fecha actual.	
Observaciones: Confirmado con el cliente.	



Tabla 2.7 HU No.5: Mostrar el historial del empleado

Historia de usuario	
Número: 5	Usuario: Especialista de Recursos Humanos
Nombre: Mostrar el historial del empleado.	
Prioridad en el negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Odelmis Rivera Regalado	
Descripción: Cuando el administrador entre al sistema y desea ver como ha sido la asistencia en el mes o en el día de un empleado determinado.	
Observaciones: Confirmado con el cliente.	

Tabla 2.8 HU No.6: Exportar datos de asistencia del mes.

Historia de usuario	
Número: 6	Usuario: Especialista de Recursos Humanos
Nombre: Exportar datos de asistencia del mes.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Odelmis Rivera Regalado	
Descripción: Cuando el administrador quiera exportar los datos de la asistencia de cada trabajador.	
Observaciones: Confirmado con el cliente.	



Tabla 2.9 HU No.7: Controlar ausencias al trabajo.

Historia de usuario	
Número: 7	Usuario: Especialista de Recursos Humanos
Nombre: Controlar ausencias al trabajo.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Odelmis Rivera Regalado	
Descripción: Cuando el administrador quiera controlar las ausencias del empleado y desee ver las justificaciones por la cual faltó.	
Observaciones: Confirmado con el cliente.	

Tabla 2.10 HU No.8: Cambiar Imagen.

Historia de usuario	
Número: 8	Usuario: Especialista de Recursos Humanos
Nombre: Cambiar Imagen.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Odelmis Rivera Regalado	
Descripción: Cuando el usuario quiera cambiar la imagen que tiene actualmente en el sistema.	
Observaciones: Confirmado con el cliente.	



Anexo 2 Tarjetas CRC.

Tabla 2.14 Tarjeta CRC No.1 Autenticar usuarios.

Nombre de la clase: Autenticar _ usuarios	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Autenticar Usuarios	Conexión

Tabla 2.15 Tarjeta CRC No.2 Gestionar usuarios.

Nombre de la clase: Gestionar_ usuarios.	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Insertar los datos del usuario.	Insertar _ usuarios. Eliminar _ usuarios. Mostrar _ listado. Modificar _ contraseña. Modificar _ nivel_ acceso.

Tabla 2.16 Tarjeta CRC No.3 Gestionar horario de entrada y salida de los usuarios.

Nombre de la clase: Gestionar _ horario.	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Mostrar y visualizar horario de entrada y salida de los usuarios.	Conexión

**Tabla 2.17 Tarjeta CRC No.4 Mostrar fecha actual.**

Nombre de la clase: Mostrar_ fecha.	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Mostrar fecha actual.	Conexión

Tabla 2.18 Tarjeta CRC No 5 Mostrar el historial del empleado.

Nombre de la clase: Mostrar_ historial.	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Mostrar el historial del empleado.	Conexión.

Tabla 2.19 Tarjeta CRC No 6 Exportar datos de asistencia del mes.

Nombre de la clase: Exportar_ datos.	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Exportar datos de asistencia del mes.	Conexión.



Tabla 2.20 Tarjeta CRC No.7 Controlar ausencias al trabajo.

Nombre de la clase: Controlar_ ausencias.	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Controlar ausencias.	Conexión

Tabla 2.21 Tarjeta CRC No.8 Cambiar Imagen.

Nombre de la clase: Cambiar _ imagen.	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Cambiar Imagen.	Conexión



Anexo 3 Tarjetas de tareas.

Tabla 3.1 Distribución de tareas por historia de usuario.

Autenticar usuarios.	Insertar los datos para entrar al Sistema.
Gestionar usuarios.	Insertar usuarios. Eliminar usuarios. Mostrar listado. Modificar contraseña. Modificar nivel de acceso. Modificar usuario.
Gestionar horario de entrada y salida de los usuarios.	Registrar hora de entrada y salida. Mostrar o visualizar horario de entrada y salida.
Mostrar fecha actual.	Mostrar fecha.
Exportar datos de asistencia del mes.	Exportar datos.
Mostrar el historial del empleado.	Mostrar historial diario de empleados por usuario. Mostrar historial por mes.
Controlar ausencias al trabajo.	Controlar ausencias.
Cambiar Imagen	Cambiar Imagen.

Tabla 3.2 Historias de usuario abordadas en la primera iteración.

Historias de usuario	Tiempo de estimación (semanas)
Autenticar Usuarios	1 día
Gestionar Usuarios	2 semanas
Gestionar hora de entrada y salida.	2 semanas
Mostrar fecha actual	1 semana



Tabla 3.3 Tarjeta de tarea No.1: Autenticar Usuario.

Tarea ingeniería	
Número tarea: 1	Número historia: 1
Nombre tarea: Autenticar Usuario.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha inicio: 2/2/2011	Fecha fin: 2/2/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea facilita que los usuarios puedan autenticarse y entrar al sistema.	

Tabla 3.4 Tarjeta de tarea No.2 Insertar Usuario.

Tarea ingeniería	
Número tarea: 2	Número historia: 2
Nombre tarea: Insertar Usuario.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha inicio: 3/2/2011	Fecha fin: 4/2/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea facilita insertar nuevos usuarios al sistema.	



Tabla3.5 Tarjeta de tarea No.3 Eliminar Usuario.

Tarea ingeniería	
Número tarea: 3	Número historia: 2
Nombre tarea: Eliminar Usuario.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 7/2/2011	Fecha fin: 8/2/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea facilita eliminar los usuarios existentes en el sistema.	

Tabla3.6 Tarjeta de tarea No.4 Modificar contraseña.

Tarea ingeniería	
Número tarea: 4	Número historia: 2
Nombre tarea: Modificar contraseña.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 9/2/2011	Fecha fin: 10/2/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea facilita modificar la contraseña del usuario.	

**Tabla3.7 Tarjeta de tarea No.5 Mostrar listado de los usuarios.**

Tarea ingeniería	
Número tarea: 5	Número historia: 2
Nombre tarea: Mostrar listado de los usuarios.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 11/2/2011	Fecha fin: 12/2/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea facilita mostrar un listado de todos los usuarios existentes en el sistema.	

Tabla3.8 Tarjeta de tarea No.6 Modificar usuarios.

Tarea ingeniería	
Número tarea: 6	Número historia: 2
Nombre tarea: Mostrar listado de los usuarios.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 15/2/2011	Fecha fin: 16/2/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea facilita modificar los usuarios existentes en el sistema.	



Tabla3.9 Tarjeta de tarea No.7 Modificar nivel de acceso.

Tarea ingeniería	
Número tarea: 7	Número historia: 2
Nombre tarea: Modificar nivel de acceso.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 16/2/2011	Fecha fin: 17/2/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea facilita modificar el nivel de acceso de un empleado.	

Tabla3.10 Tarjeta de tarea No.9 Registrar hora de entrada y salida.

Tarea ingeniería	
Número tarea: 9	Número historia: 3
Nombre tarea: Registrar hora de entrada y salida.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 21/2/2011	Fecha fin: 24/2/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea permite registrar en la base de datos la hora de entrada y salida de los usuarios.	

**Tabla3.11 Tarjeta de tarea No.10 Mostrar o visualizar hora de entrada y salida.**

Tarea ingeniería	
Número tarea: 10	Número historia: 3
Nombre tarea: Mostrar o visualizar hora de entrada y salida.	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 25/2/2011	Fecha fin: 28/2/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea permite mostrar en el sistema la hora de entrada y salida de los usuarios.	

Tabla3.12 Tarjeta de tarea No.11 Mostrar fecha actual.

Tarea ingeniería	
Número tarea: 11	Número historia: 4
Nombre tarea: Mostrar fecha actual	
Tipo de tarea : Desarrollo	Puntos estimados: 1
Fecha inicio: 28/2/2011	Fecha fin: 1/3/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea permite que cuando el usuario entre al sistema se muestre la fecha actual.	

**Tabla3.13 Historias de usuarios abordadas en la segunda iteración.**

Historias de usuario	Tiempo de estimación (semanas)
Mostrar el historial del empleado. Exportar datos de asistencia del mes.	4 semanas

Tabla3.14 Tarjeta de tarea No.12 Mostrar historial diario de empleados por usuario.

Tarea ingeniería	
Número tarea: 12	Número historia: 5
Nombre tarea: Mostrar el historial del empleado.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 2/3/2011	Fecha fin: 4/3/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea permite que el encargado pueda ver como ha sido la asistencia de cualquier empleado, en el día.	

Tabla3.15 Tarjeta de tarea No.13 Mostrar historial de empleados por mes.

Tarea ingeniería	
Número tarea: 13	Número historia: 5
Nombre tarea: Mostrar el historial del empleado.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 4/3/2011	Fecha fin: 7/3/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea permite que el encargado pueda ver como ha sido la asistencia de cualquier empleado, en el mes.	

**Tabla3.16 Tarjeta de tarea No.14 Exportar datos de asistencia del mes.**

Tarea ingeniería	
Número tarea: 14	Número historia: 6
Nombre tarea: Mostrar el historial del empleado.	
Tipo de tarea : Desarrollo	Puntos estimados: 2
Fecha inicio: 8/3/2011	Fecha fin: 9/3/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea permite que el encargado pueda exportar los datos de asistencia de un trabajador en el mes.	

Tabla3.17 Historias de usuarios abordadas en la segunda iteración.

Historias de usuario	Tiempo de estimación (semanas)
Controlar las ausencias al trabajo.	2 semanas
Cambiar Imagen.	1 día



Tabla3.18 Tarjeta de tarea No.15 Controlar ausencias.

Tarea ingeniería	
Número tarea: 15	Número historia: 7
Nombre tarea: Controlar ausencias al trabajo.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 10/3/2011	Fecha fin: 12/3/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea permite controlar por parte del encargado las ausencias de cada trabajador cuando este ha llegado tarde.	

Tabla3.19 Tarjeta de tarea No.16 Cambiar Imagen.

Tarea ingeniería	
Número tarea: 16	Número historia: 8
Nombre tarea: Cambiar Imagen.	
Tipo de tarea : Desarrollo	Puntos estimados: 3
Fecha inicio: 13/3/2011	Fecha fin: 13/3/2011
Programador responsable: Odelmi Rivera Regalado.	
Descripción: Esta tarea permite que se pueda cambiar la imagen existente en el sistema.	



Anexo 4 Pruebas de aceptación.

Tabla 3.20 Pruebas de aceptación para la HU: Autenticar usuarios.

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

Tabla 3.21 Pruebas de aceptación para la HU: Gestionar usuarios.

Prueba de aceptación
HU: Gestionar usuarios.
Nombre: Prueba para realizar la gestión de usuarios.
Descripción: Validación de entrada de datos de los usuarios.
Condiciones de ejecución: el administrador debe entrar a la aplicación.
Entrada/Pasos ejecución: El administrador introduce los datos solicitados para realizar las diferentes operaciones de gestión.
<ul style="list-style-type: none">• Resultado: Si se insertó el usuario correctamente muestra un mensaje de confirmación con este mensaje "Usuario insertado correctamente".• Cuando se cambia la contraseña se emite otro mensaje de confirmación "Se ha cambiado su contraseña". <p>Se emite un mensaje de error en caso de que:</p> <ul style="list-style-type: none">• No se introduzcan los datos solicitados.• Se intente insertar un usuario que ya exista.• Falten datos del usuario a la hora de insertarlo.• A la hora de confirmar la contraseña no coincida con la que se puso anteriormente.• Cuando se le va a cambiar la contraseña a un usuario, que se escriba incorrectamente
Evaluación de la prueba: Aceptada.

Tabla 3.22 Pruebas de aceptación para la HU: Gestionar horario de entrada y salida.

Prueba de aceptación
HU: Gestionar horario de entrada y salida.
Nombre: Prueba para gestionar horario de entrada y salida.
Descripción: Validación de la gestión de horario.
Condiciones de ejecución: el usuario debe entrar a la aplicación.
Entrada/Pasos ejecución: El usuario introduce los datos solicitados para entrar a la aplicación.
Resultado: Si el empleado no firma, no se registra ni visualiza la hora. Si el empleado llega tarde, a la hora de firmar se le mostrará un cartel con el siguiente mensaje "Usted ha llegado tarde, por favor justifíquese" luego de justificarse se mostrará la hora.
Evaluación de la prueba: Aceptada.

Tabla 3.23 Pruebas de aceptación para la HU: Mostrar fecha actual.

Prueba de aceptación
HU: Mostrar fecha actual.
Nombre: Prueba para mostrar fecha.
Descripción: Validación de fecha.
Condiciones de ejecución: el usuario debe entrar a la aplicación.
Entrada/Pasos ejecución: El usuario introduce los datos solicitados para entrar a la aplicación.
Resultado: Si el empleado no entra a la aplicación no se muestra la fecha.
Evaluación de la prueba: Aceptada.

Tabla 3.24 Pruebas de aceptación para la HU: Mostrar el historial del empleado.

Prueba de aceptación
HU: Mostrar el historial del empleado.
Nombre: Prueba para mostrar el historial del empleado.
Descripción: Validación de mostrar el historial.
Condiciones de ejecución: el administrador debe entrar a la aplicación.
Entrada/Pasos ejecución: El administrador introduce los datos solicitados para entrar a la aplicación.
Resultado: Se emite un mensaje de error en caso de que en la búsqueda de empleados por mes, no se seleccione el usuario a buscar.
Evaluación de la prueba: Aceptada.

Tabla 3.25 Pruebas de aceptación para la HU: Exportar datos de asistencia del mes.

Prueba de aceptación
HU: Exportar datos de asistencia del mes.
Nombre: Prueba para exportar datos.
Descripción: Validación de la exportación del resumen.
Condiciones de ejecución: el administrador debe entrar a la aplicación.
Entrada/Pasos ejecución: El administrador introduce los datos solicitados para entrar a la aplicación.
Resultado: Se exportó un resumen detallado con toda la asistencia de los trabajadores en el mes que se está trabajando.
Evaluación de la prueba: Aceptada.

Tabla 3.26 Pruebas de aceptación para la HU: Controlar ausencias al trabajo.

Prueba de aceptación
HU: Controlar ausencias.
Nombre: Prueba para controlar las ausencias.
Descripción: Validación de control de ausencias.
Condiciones de ejecución: el usuario debe entrar a la aplicación.
Entrada/Pasos ejecución: El empleado introduce los datos solicitados para entrar a la aplicación.
Resultado: Si el empleado llega tarde, se le muestra un cartel que dice “Usted ha llegado tarde, justifíquese” y debe clasificar el tipo de ausencia por la que llegó tarde.
Evaluación de la prueba: Aceptada.

Tabla 3.27 Pruebas de aceptación para la HU: Cambiar Imagen.

Prueba de aceptación
HU: Cambiar Imagen.
Nombre: Prueba para Cambiar Imagen.
Descripción: Validación para cambiar la imagen.
Condiciones de ejecución: el usuario debe entrar a la aplicación.
Entrada/Pasos ejecución: El empleado introduce los datos solicitados para entrar a la aplicación.
Resultado: El usuario puede cambiar su imagen cada vez que quiera.
Evaluación de la prueba: Aceptada.

Anexo 5 Modelo de base de datos.

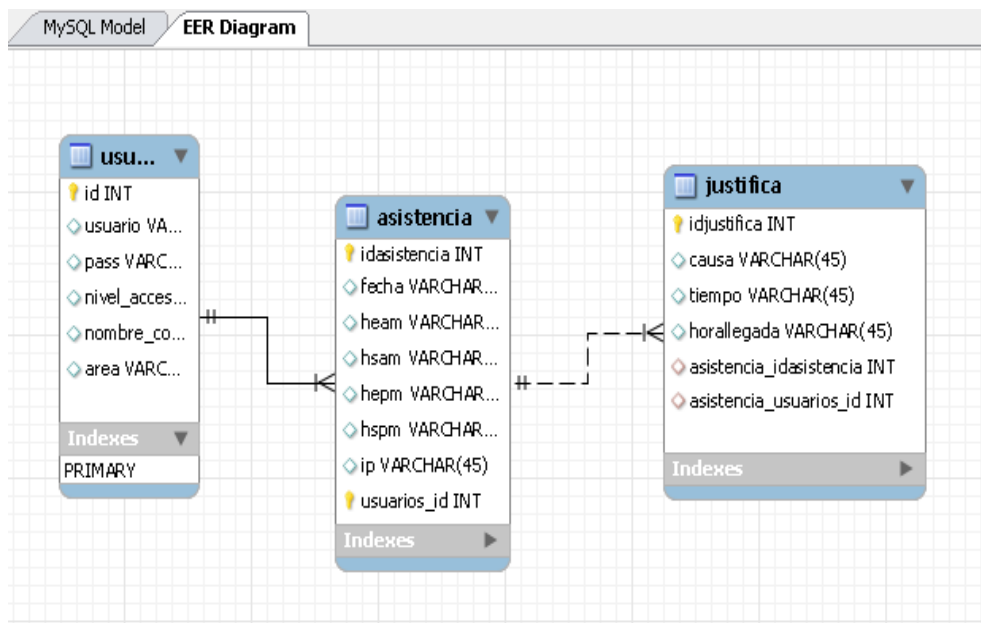


Fig. 3 Modelo de datos.

Anexo 6 Vistas de algunas interfaces.



Fig.4 Vista de la Interfaz del administrador.

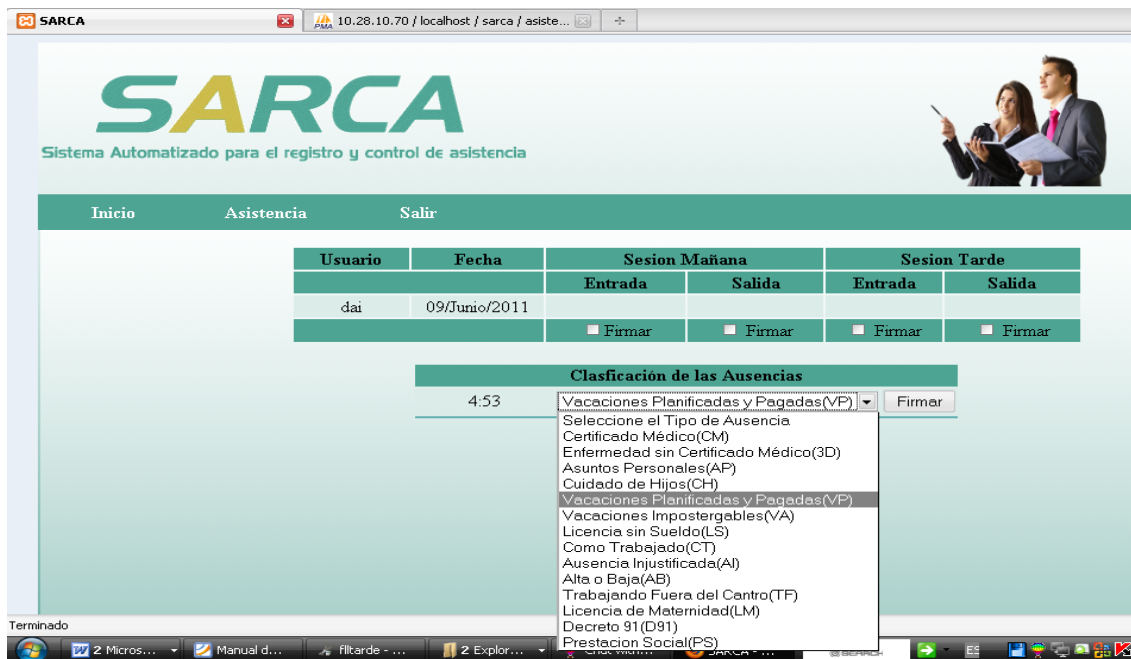


Fig.5 Vista de la Interfaz de Firmar cuando se llega tarde.



Gestionar empleados

Usuario	Nombres y Apellidos	Nivel	Área	Opciones			
ode	Odelmis Rivera Regalado	0	caja	Modificar	Cambiar Clave	Cambiar Nivel Acceso	<input type="checkbox"/>
yoandi	yoandi lopez llorente	1	caja	Modificar	Cambiar Clave	Cambiar Nivel Acceso	<input type="checkbox"/>
daimara	daimara areas ramirez	1	comercial	Modificar	Cambiar Clave	Cambiar Nivel Acceso	<input type="checkbox"/>
dany	dany laurencio gualarte	0	caja	Modificar	Cambiar Clave	Cambiar Nivel Acceso	<input type="checkbox"/>
arcaya	edliberto arcaya matos	0	gerente	Modificar	Cambiar Clave	Cambiar Nivel Acceso	<input type="checkbox"/>
mari	mari romero gongora	1	comercial	Modificar	Cambiar Clave	Cambiar Nivel Acceso	<input type="checkbox"/>
mailin	mailin romero gongora	1	caja	Modificar	Cambiar Clave	Cambiar Nivel Acceso	<input type="checkbox"/>
admin	Administrador del Sistema	0	Redes	Modificar	Cambiar Clave	Cambiar Nivel Acceso	<input type="checkbox"/>

Fig.6 Vista de la Interfaz de Gestionar usuarios.



Realizar Búsqueda

Usuario:

Nombre	Fecha	Tiempo	Ausencia	Hora
mari romero gongora	22/Junio/2011	1:40	Cuidado de Hijos(CH)	9:30
mari romero gongora	23/Junio/2011	1:11	Trabajando Fuera del Contro(TF)	9:01
mari romero gongora	24/Junio/2011	0:14	Licencia sin Sueldo(LS)	8:04
mari romero gongora	24/Junio/2011	6:20	Licencia sin Sueldo(LS)	3:40
mari romero gongora	28/Junio/2011	1:1	Trabajando Fuera del Contro(TF)	8:51

Fig.7 Vista de la Interfaz de Ausencias.

