

Universidad de las Ciencias Informáticas
Facultad 1



*Sistema de gestión de los medios básicos informáticos de la Unión
de Empresas de Recuperación de Materias Primas.*

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores:

José Antonio Brizuela Santoyo

Néstor Alain Arnet Oviedo


Tutor:

Ing. Marianny Hernández Batista.

Ciudad de La Habana, 2010.

Año del 52 aniversario de la Revolución.

FRASE



“La tontería se pone delante para ser vista, la inteligencia detrás para observar”.
“Cada uno de nosotros es un modelo totalmente nuevo, parecido a otros modelos pero
totalmente diferente”.

Ernesto Che Guevara.

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

José Antonio Brizuela Santoyo

Néstor Alain Arnet Oviedo

Firma del autor

Firma del autor

Ing. Marianny Hernández Batista.

Firma del tutor

DEDICATORIA

José Antonio:

Quisiera dedicarles la tesis a mi mamá y a mi papá por ser lo más grande que tengo en la vida.

A mi novia porque sin ella nunca hubiera terminado este peligroso y difícil camino.

A mis hermanos tantos los de sangre como lo de guerra esto es para ustedes y a mis queridísimos suegros que me adoptaron como un hijo más.

A la Revolución Cubana por darme la posibilidad de convertirme en un profesional.

A Fidel mi comandante que lo seguiré hasta el final del mundo.

AGRADECIMIENTOS

Alain:

Quiero agradecerle a mi familia principalmente a mi mamá y a mi papá, que son las personas por la cual he seguido el camino del estudio, a mi hermana por estar pendiente a mi evolución como estudiante y ayudarme en los momentos más importantes y a mi sobrino que lo quiero con el corazón.

Agradecerles también a mis hermanos del pre-universitario, especialmente a Noel y al Guille, sin olvidarme de mi comadre Orquídea que siempre ha estado al tanto de mí. Seguidamente agradecerle a mis hermanos de la UCI, la Gaby, Imer, Eddy y Tommy que han estado conmigo en las buenas y en las malas, a mis amistades de quinto año, a los programadores del Symphony, Ibrael, Ángel, Reinier y al Payo que fueron de gran ayuda en la evolución de esta tesis.

José Antonio:

A la tierra que me vio nacer y a la Revolución por ofrecerme la posibilidad de estudiar y realizarme como profesional.

A todas las escuelas en las que curse las diferentes etapas de enseñanza. A todos los profesores que de una forma u otra aportaron su granito de arena en mi preparación, como persona y como profesional.

A mis padres por apoyarme en todo y estar ahí cuando los necesito, mamá y papí sin ustedes no sería la persona que soy.

A mi hermana con alma de artista por aguantarme todos estos años, a mis 2 hermanos por ser mi ejemplo.

A mi novia, gracias por cambiar el curso de mi vida.

RESUMEN

La Unión de Empresas de Recuperación de Materias Primas (UERMP), es una institución que fue creada con ese nombre en 1981. Con la renovación e incorporación de nuevas tecnologías y equipos utilizados en los procesos de recuperación y transformación de los desechos reciclables, la dirección de esta empresa ha tomado la decisión de informatizar el sistema de registro y control de los medios básicos informáticos, para así llevar a cabo la centralización de dicho control. Actualmente esta empresa cuenta con una aplicación desarrollada en MS-DOS la cual tiene baja calidad, presenta errores y no cumple con las funcionalidades encomendadas; por lo que desarrollar una aplicación Web que permita automatizar los diferentes procesos que se llevan a cabo en la UERMP ha sido el objetivo general de esta investigación. Se tiene en cuenta para desarrollar la aplicación la metodología XP para el desarrollo del mismo, Symfony como marco de trabajo de desarrollo, como gestor de base datos PostgreSQL y PHP como lenguaje de programación.

ÍNDICE DE CONTENIDO

ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	9
CAPÍTULO 1	13
1.1 INTRODUCCIÓN	13
1.2 CONCEPTOS DE GESTIÓN	13
1.3 GESTIÓN DE LA INFORMACIÓN	14
1.4 SISTEMAS DE GESTIÓN.....	15
1.5 SOFTWARE PARA EL CONTROL DE MEDIOS BÁSICOS	16
1.5.1 EN EL ÁMBITO INTERNACIONAL.....	17
1.5.2 EN EL ÁMBITO NACIONAL	18
1.5.3 EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS (UCI)	19
1.6 ANÁLISIS DE LAS TECNOLOGÍAS.....	20
1.6.1 METODOLOGÍA XP.....	20
1.6.2 RUP	21
1.6.3 SCRUM	23
1.7 ¿QUÉ ES UNA APLICACIÓN WEB?	23
1.8 HTML.....	24
1.8.1 CCS	25
1.9 LENGUAJE DE PROGRAMACIÓN DE LADO DEL CLIENTE (JAVA SCRIPT)	26
1.10 LENGUAJE DE PROGRAMACIÓN DE LADO DEL SERVIDOR	27
1.10.1 PHP.....	27
1.10.2 ASP.NET	28
1.10.3 PHP5.....	28
1.11 SERVIDOR WEB APACHE.....	29
1.12 GESTORES DE BASE DE DATOS A UTILIZAR.....	30
1.12.1 POSTGRE SQL.....	30
1.12.2 ORACLE	31
1.12.3 MYSQL	32
1.13 MARCO DE TRABAJO.....	32
1.13.1 CAKE PHP	32

ÍNDICE DE CONTENIDO

1.13.2	SYMFONY.....	32
1.14	NETBEANS IDE.....	33
1.15	ARQUITECTURA.....	34
1.16	CONCLUSIONES.....	35
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.....		37
2.1	INTRODUCCIÓN.....	37
2.2	PROPUESTA DE SOLUCIÓN.....	37
2.3	FASE DE PLANIFICACIÓN.....	39
2.4	HISTORIAS DE USUARIO.....	39
2.5	PERSONAS RELACIONADAS CON EL SISTEMA.....	41
2.6	REQUERIMIENTOS NO FUNCIONALES.....	41
2.7	ESTIMACIÓN DE ESFUERZO POR HISTORIA DE USUARIO.....	43
2.8	PLAN DE ITERACIONES.....	44
2.9	PLANIFICACIÓN DE DURACIÓN DE ITERACIONES.....	45
2.10	PLAN DE ENTREGAS.....	46
2.11	CONCLUSIONES.....	47
CAPÍTULO 3: DISEÑO IMPLEMENTACIÓN Y PRUEBAS.....		48
3.1	INTRODUCCIÓN.....	48
3.2	FASE DE DISEÑO.....	48
3.2.1	TARJETAS CRC (CLASE, RESPONSABILIDAD Y COLABORACIÓN).....	50
3.2.2	TAREAS DE INGENIERÍA.....	54
3.3	DISEÑO DE LA BASE DATOS.....	55
3.4	DIAGRAMA DE DESPLIEGUE.....	55
3.5	DESARROLLO: CICLO DE VIDA DE XP PARA LA ETAPA DE DISEÑO.....	57
3.6	PRUEBA.....	58
3.7	CONCLUSIONES.....	62
CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD.....		63
4.1	INTRODUCCIÓN.....	63
4.2	CARACTERÍSTICAS DEL PROYECTO.....	63
4.3	ESTIMACIÓN INICIAL.....	66
4.4	CÁLCULO DE INSTRUCCIONES FUERTES, ESFUERZO, TIEMPO DE DESARROLLO, CANTIDAD DE HOMBRES Y COSTO.....	66

ÍNDICE DE CONTENIDO

4.5	BENEFICIOS TANGIBLES E INTANGIBLES	70
4.6	ANÁLISIS DE COSTO Y BENEFICIOS	71
4.7	CONCLUSIONES	71
	CONCLUSIONES GENERALES	72
	RECOMENDACIONES	73
	BIBLIOGRAFÍA REFERENCIADA	74
	BIBLIOGRAFÍA CONSULTADA	76
	GLOSARIO DE TÉRMINOS.....	78

INTRODUCCIÓN

Introducción

La estrategia cubana, en la recuperación de materias primas después del Triunfo de la Revolución fue concebida y proyectada por Ernesto Che Guevara siendo Ministro de Industrias y se origina cuando se crea mediante la Resolución 21-1272 del 7 de noviembre de 1961, la Empresa Consolidada de Recuperación de Materias Primas.

La Empresa Consolidada de Recuperación de Materias Primas transitó históricamente por varios organismos centrales desde su fundación en el Ministerio de Industrias. Luego se subordinó al Ministerio de la Industria Básica y en febrero de 1974 pasó al de la Industria Sideromecánica, donde se creó la rama de Materias Primas para dirigir esta actividad a nivel nacional. En 1981 se convierte en la Unión de Empresas de Recuperación de Materias Primas (UERMP). Esta empresa cuenta actualmente con 26 empresas clasificadas: 15 recuperadoras, una en cada provincia del país incluyendo el municipio especial Isla de la Juventud, 9 especializadas y 2 exportadoras–importadoras. Con la renovación e incorporación de nuevas tecnologías y equipos utilizados en los procesos de recuperación y transformación de los desechos reciclables, la dirección de esta empresa ha tomado la decisión de informatizar el sistema de registro y control de los medios básicos informáticos para así llevar a cabo la centralización de dicho control, actualmente esta empresa cuenta con una aplicación desarrollada en el sistema operativo MS-DOS, con muy baja calidad, que es la que se está utilizando pero hay funcionalidades que no tiene implementada. Actualmente la mayoría de los procesos de gestión de la información que se realizan en la UERMP son de forma manual, archivándose toda la información en formato duro. Entre las funcionalidades que se necesitan automatizar están las de control de entrada y salida de los medios básicos informáticos, control sobre los medios básicos que entran en la lista de mantenimiento, gestión de los medios básicos informáticos por tipo. Actualmente estas informaciones sólo se imprimen y archivan en la máquina del informático de cada empresa expuesto a pérdidas o a negligencias del trabajo.

INTRODUCCIÓN

Por la situación problemática antes expuesta, se define el siguiente **problema científico**:
¿Cómo mejorar la gestión de los medios básicos informáticos de la Unión de Empresas de Recuperación de Materias Primas?

Teniendo como **objetivo general**: Desarrollar un Sistema de Gestión de los Medios Básicos informáticos de la Unión de Empresas de Recuperación de Materias Primas.

Desglosado en los siguientes **objetivos específicos**:

- ✓ Identificar, describir y modelar los procesos relacionados con los medios básicos informáticos de la Unión de Empresas de Recuperación de Materias Primas.
- ✓ Realizar el análisis y diseño del sistema de gestión de los medios básicos informáticos de la Unión de Empresas de Recuperación de Materias Primas.
- ✓ Realizar la implementación del Sistema de Gestión de los Medios Básicos Informáticos de la Unión de Empresas de Recuperación de Materias Primas.
- ✓ Realizar las pruebas pertinentes al Sistema de Gestión de los Medios Básicos Informáticos de la Unión de Empresas de Recuperación de Materias Primas.

Tareas de la investigación:

- ✓ Estudio de la documentación existente sobre el registro y control de los medios básicos informáticos de la Unión de Empresas de Recuperación de Materias Primas.
- ✓ Definición de la metodología de desarrollo de *software* a utilizar.
- ✓ Definición de las herramientas a utilizar (ambiente de desarrollo).
- ✓ Definición de funcionalidades del sistema.
- ✓ Definición de la propuesta del sistema.
- ✓ Identificación de los patrones y estándares a utilizar en la implementación del sistema.
- ✓ Implementación del Sistema de Gestión de los Medios Básicos Informáticos de la Unión de Empresas de Recuperación de Materias Primas.
- ✓ Realización de las pruebas pertinentes para comprobar la funcionalidad y calidad del Sistema de Gestión de los Medios Básicos Informáticos de la Unión de Empresas de Recuperación de Materias Primas.

INTRODUCCIÓN

Hipótesis:

La implementación del Sistema de Gestión de Medios Básicos Informáticos en la Unión de Empresas de Recuperación de Materias Primas que contribuirá a mejorar la gestión de los medios básicos informáticos.

Variables Independientes:

La implementación del sistema de los medios básicos informáticos de la Unión de Empresas de Recuperación de Materias Primas.

Variables dependientes:

Mejorar la gestión de los medios básicos informáticos.

Operacionalización de las variables:

Variable conceptual	Dimensión	Indicadores	Unidad de medida
Implementación del sistema de los medios básicos informáticos de la Unión de Empresas de Recuperación de Materias Primas.	Factibilidad	Tiempo de desarrollo	Corto plazo
			Largo plazo
		Costo de desarrollo	Alto
			Medio
			Bajo
		Mejorar la gestión de los medios básicos informáticos	Funcionalidad
Funcionalidad media	2		
Funcionalidad baja	1		

Tabla 1: Variables de hipótesis

Métodos empíricos:

Entrevista: se realiza a directivos y clientes en general para obtener información relacionada con la aplicación lo que contribuye al perfeccionamiento de la propuesta realizada.

Métodos Teóricos:

INTRODUCCIÓN

Análisis y síntesis: este permite la separación de un todo en sus partes o en sus elementos constitutivos. Se apoya en que para conocer un fenómeno es necesario descomponerlo en sus partes y luego sintetizar, esto no es más que la unión de los elementos para formar un todo.

Histórico y lógico: se emplea para el estudio y profundización de la evolución de las aplicaciones lo que facilita la indagación de soluciones al problema planteado.

Modelación: se utiliza durante la elaboración del sistema, porque se hace necesario explicarle al cliente mediante modelos, cómo se tiene pensado que quede el sistema para saber si cumple con sus necesidades.

Enfoque de sistema: se aplica para el estudio y profundización constante del tema de investigación, lo que propicia la introducción de proyecciones concebidas en las diferentes tareas integradoras realizadas en función del objetivo propuesto.

El trabajo está estructurado como sigue:

Capítulo 1: Fundamentación Teórica: Se abarca los principales conceptos manipulados en el transcurso de la investigación y una referencia al estado del arte de los procesos de gestión de los medios básicos informáticos y se definirán las herramientas y metodologías utilizadas para realizar el sistema.

Capítulo 2: Características del sistema: En este capítulo se realiza todo el levantamiento de los requisitos del sistema, ya sean funcionales y no funcionales y demás artefactos definidos por la metodología a utilizar para esta fase.

Capítulo 3: Análisis, Diseño e Implementación y Prueba: Se realiza el análisis y diseño de la aplicación, elaborando los artefactos para esta fase, como son las tarjetas CRC y las tareas de ingeniería. Además se realiza la implementación del sistema, definiendo el modelo de despliegue, y las pruebas pertinentes para comprobar la factibilidad de la aplicación.

Capítulo 4: Estudio de factibilidad: En este capítulo se describe el estudio de factibilidad realizado correspondiente al sistema propuesto, teniendo en cuenta el costo estimado y los beneficios que reportará al ser implantado.

Capítulo 1

1.1 Introducción

En este capítulo se hace un estudio acerca de los sistemas de gestión, detallando conceptos importantes utilizados durante la investigación referente al objeto de estudio. Se explicarán también las diferentes tecnologías y herramientas que se selecciona para desarrollar el sistema.

Introducción a la Gestión

Desde la comunidad primitiva, el hombre se ha planteado la necesidad de regular sus acciones y recursos. Este proceso que inicialmente era una actividad intuitiva, fue perfeccionándose gradualmente y con el tiempo evolucionó a modelos que han ido profundizando y refinando sus mecanismos de funcionamiento y formas de ejecución, hasta convertirse en sistemas que adaptados a características concretas y particulares, han pasado a formar parte elemental y punto de atención de cualquier organización. Derivados de este proceso surgen ideas y términos como la gestión y todo lo que ella representa. Con el desarrollo de la sociedad y de los sistemas de producción, influenciados por el desarrollo científico técnico y las revoluciones industriales, los principios de gestión existentes han ido evolucionando, convirtiéndose en un proceso de gran importancia para cualquier actividad humana. La gestión está caracterizada por una visión más amplia de las posibilidades reales de una organización, para resolver determinada situación o arribar a un fin determinado. Puede asumirse, como la disposición y organización de los recursos de un individuo o grupo, para obtener los resultados esperados. Pudiera generalizarse como una forma de alinear los esfuerzos y recursos para alcanzar un fin determinado. (Solán, 2008)

1.2 Conceptos de gestión

Gestión: es el proceso mediante el cual se obtiene, despliega o utiliza una variedad de recursos básicos para apoyar los objetivos de la organización. Gestionar es coordinar todos los recursos disponibles para conseguir determinados objetivos, implica amplias y fuertes

interacciones fundamentalmente entre el entorno, las estructuras, el proceso y los productos que se deseen obtener. (Salazar, 2000)

1.3 Gestión de la Información

En la actualidad constantemente se escucha mencionar el término “era de la información”, lo que se deja ver, la importancia que tiene este recurso para los procesos de desarrollo en los que se encuentra inmersa la sociedad vigente. La gestión de la información es el proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, así como, mejorar los procesos, productos y servicios de la organización. El objetivo fundamental de la gestión de la información, es crear sistemas que permitan socializar la información que tiene una entidad y los individuos que la conforman. Estos sistemas incluyen operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y colaboración de la información adquirida por la organización a través de diferentes fuentes. Gestionan además el acceso y los derechos de los usuarios sobre la misma. La gestión de la información se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar y posteriormente, recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades. En el centro de la gestión de la información se encuentra la gestión de la documentación (la información que queda plasmada en documentos) y que puede ser de tres tipos:

Interna: Hace referencia a aquella documentación generada o recibida por la organización, en el ejercicio de sus funciones, es decir, son documentos que surgen de la actividad diaria de esa

Institución: en este grupo, se encuentra tanto la documentación típicamente administrativa: contabilidad, correspondencia, como la documentación de gestión: informes, actas de reuniones, procedimientos de trabajo, y la documentación técnica que refleja la propia actividad de la organización.

CAPÍTULO I

Externa: además de la documentación producida por la propia organización, esta y las personas que trabajan en ella, necesitan a menudo, consultar y manejar fuentes de información externas: libros revistas, base de datos, internet.

Pública: es aquella documentación que la organización produce de cara al público, para comunicarles hechos, actividades, acontecimientos, por ejemplo, las memorias, los catálogos de productos y servicios, la página web. (Ruesta, 2001)

1.4 Sistemas de Gestión

El uso de sistemas de gestión, permite a las organizaciones automatizar un conjunto de funciones importantes dentro del proceso empresarial, dentro de ellas se encuentra la contabilidad, distribución de productos, recursos humanos e información. La implantación de un sistema de gestión trae consigo mejoras a la empresa, ya que permite facilitar y agilizar procesos que generalmente se realizan de forma manual. Además permite tener un mayor control de los recursos, trabajadores e información que en ella se manipula. Por lo que se puede considerar a un sistema de gestión como: Un conjunto de elementos que interactúan entre sí, con el fin de apoyar las actividades de una empresa o negocio. Estos elementos son de naturaleza diversa y normalmente incluyen: el equipo informático o hardware necesario para que pueda operar (lo constituyen las computadoras y el equipo periférico que puede conectarse a ellas); los recursos humanos que interactúan con él, formados por personas que utilizan el sistema y los programas o *software*, que harán que los datos de entrada introducidos sean procesados correctamente y generen los resultados que se esperan. El proceso de gestión tiene entre sus objetivos las siguientes funciones:

Planificar: Es el proceso de establecer objetivos con el fin de alcanzar determinados resultados, así como, identificar las acciones necesarias para alcanzarlos. Dentro de este concepto se contemplan un conjunto de decisiones o una selección de alternativas para el logro de tales resultados. Esta función se registra en diversos tipos de documentos: planes, programas, pronósticos y políticas.

CAPÍTULO I

Organizar: Es el proceso de dividir el trabajo a realizar y de coordinar el logro de resultados que tienen un propósito común. Organizar es la química de la organización donde se mezclan todos los elementos que interactúan entre sí a fin de obtener los resultados esperados. Es el acto de combinar habilidades, posibilidades técnicas, experiencias, recursos y todos los elementos que podrían convertirse en resultados.

Dirigir: Es el proceso de conducir y coordinar los esfuerzos laborales de las personas que integran una organización, ayudándolos a desarrollar tareas relevantes dentro de ella. La dirección es la función mediante la cual se ponen en marcha las actividades programadas. Comprende el compromiso de alcanzar un objetivo mediante el liderazgo de un grupo. La dirección ejerce una influencia notable en las personas para que trabajen voluntaria y entusiastamente para el logro de las metas colectivas de equipos y de la organización en su conjunto.

Controlar: Es el proceso de supervisar las actividades y resultados, comparándolos con los objetivos y tomando las acciones correctivas, si son necesarias. Para ello se compara el desempeño con metas y planes, se muestran las desviaciones y al emprender medidas para corregir las desviaciones, se ayuda a asegurar el logro de los planes. Esta función comprende el establecimiento de normas de desempeño como base para la medida de los resultados, investigación, análisis, diseño, implantación y operación de los sistemas de información, registros contables y estadísticos, auditorías, inspecciones, controles y otros métodos de verificación directa. (Gómez, 2010)

1.5 Software para el control de medios básicos

A continuación se realiza un estudio de diferentes sistemas destinados al control de medios básicos informáticos ya sea a nivel internacional o nacional, así como el sistema que se encuentra en uso en la Universidad de las Ciencias Informáticas (UCI).

1.5.1 En el ámbito internacional

En el ámbito internacional se encuentran varios *software* de gestión, entre los que se encuentran:

GestiCam

GestiCam es un sistema informático integrado de control económico para autónomos. El proyecto nació con el fin de dotar a los autónomos de Castilla-La Mancha (región de Europa) de un software de gestión y control económico en código libre y con una integración dentro del Sistema Operativo MoLinux que distribuye la junta de comunidades de Castilla-La Mancha. Posee una gran diversidad de módulos especializados como:

- Gestión de compras.
- Gestión de ventas.
- Gestión de empresas (clientes/proveedores/parámetros).
- Gestión financiera.
- Gestión de inventario.
- Gestión de productos.
- Gestión de producción.
- Administración del sistema, copias, restauraciones, etc.
- Informes.

GestiCam es técnicamente avanzado, está basado en tecnología de bases de datos PostgreSQL. Es un sistema multiplataforma que genera archivos en formato PDF, HTML y OpenOffice. El mismo es completamente basado en productos de código abierto estándares, permite su libre distribución, modificable y adaptable incluso por el usuario final, gratuito y sin costes.

Permite que la interfaz se adapte a las necesidades del usuario además contempla varias monedas, presentaciones, etc. sin una configuración compleja. Multilenguaje, multi-empresa, multi-departamentos, multi-usuario. Total integración en la empresa. Sin dudas es un potente sistema, pero este gran alcance ha provocado rechazo entre los usuarios que desean

controlar solo pequeños grupos de medios básicos un número de ventas poco complicado, entre otras acciones. (2006, 2010)

1.5.2 En el ámbito nacional

En el ámbito nacional se encuentran varios *software* de gestión, como son:

-Sistema automatizado de control de los activos fijos tangibles para el sector de salud (MEBUS)

El sistema denominado MEBUS (Medios básicos informáticos para Unidades de Salud) forma parte del proyecto SEUS (Sistema Económico para Unidades de Salud) que se está aplicando en las unidades del Sector de la Salud. Es un sistema automatizado para la gestión del control de los medios básicos informáticos en las unidades presupuestadas de Salud Pública, con el propósito de brindar el control y flujo de información administrativa para contribuir a la toma de decisiones de forma oportuna, con mayor rapidez y más completa. Valiéndose de sus potentes herramientas en la programación visual, así como en la orientada a objetos, permite la obtención de una serie de informes sumamente importantes para el control de los medios básicos informáticos y el procesamiento de informaciones fundamentales que se brindan a niveles superiores. Incluye toda la información de los submayores de los medios básicos, brinda la depreciación por cada medio, da la posibilidad de revalorizar el medio cuya depreciación haya caducado y, en caso de no revalorizar, permite que el medio deje de depreciarse, avisa cuándo la depreciación de cada activo está llegando a su fin y contiene una base de datos pasiva con las bajas de los medios básicos informáticos por 5 años. Las informaciones procesadas que son necesarias para otros subsistemas (contabilidad general, costos, cobros y pagos, así como presupuesto), se envían automáticamente, ya sea a través de la red o por correo electrónico. (González, 2003)

-Sistema automatizado para el control de los inventarios en uso (FONDUS)

Software realizado para automatizar el subsistema de inventario en uso (fondos fijos) como parte integral del Sistema Económico para Unidades de Salud (SEUS). Se crea un sistema con aspectos teóricos relacionados con la programación orientada a objetos y la teoría de

CAPÍTULO I

base de datos. Este sistema no solo garantiza el cambio del sistema operativo MS-DOS a *Windows*, con todas las ventajas de este último, sino que brinda informes más completos sobre los movimientos realizados y el registro de los fondos fijos, así como también mejora cualitativamente la seguridad de la aplicación y la generación automática de los comprobantes de operaciones. (Echagarrúa, 2003)

A pesar del gran avance informático y su gran desarrollo en la producción del software se considera que dichos productos nacionales presenta un adecuado funcionamiento y garantiza una eficacia de la gestión y control de los medios básicos informáticos en sus respectivas ramas pero no es factible para todas las empresas ya que no cumplirían con todos los requerimientos.

1.5.3 En la Universidad de las Ciencias Informáticas (UCI)

ASSET (Sistema de Gestión Integral)

Es un sistema integral modular concebido para el control de la actividad económica empresarial. Permite realizar, controlar y contabilizar todas las transacciones relacionadas con el proceso de compra-venta de productos y servicios, los cobros, pagos y anticipos asociados a los mismos, recursos humanos y nóminas, los activos fijos y útiles y herramientas de su entidad. Es un sistema que facilita el uso de la parametrización para adaptarse a las exigencias de cada cliente en particular, en la emisión de varios reportes que tendrán la forma y el contenido que el usuario les defina.

Posibilita el control del inventario perpetuo de múltiples almacenes y la generación automática de comprobantes de operaciones, ofrece un control estricto de las existencias, reservas y disponibilidad de productos, así como de las cuentas por cobrar y pagar debidas a la facturación y recepción de productos. Está diseñado para multicompañía, con una estructura organizativa a varios niveles, en la que podrán asistir grupos corporativos, corporativos, grupo de agrupaciones, agrupaciones, almacenes y centros de costo. Para entidades de esta estructura se brinda un modulo de comunicaciones que facilita poder intercambiar información

entre ellas, con el fin de poder consolidar información sobre la gestión comercial y contable pudiéndose obtener los estados financieros, resúmenes de compra, venta etc.

Este producto es uno de los más usados en nuestro país ya que está diseñado no solo para una empresa en específico, si no que presenta una funcionalidad en general o sea que todas las compañías pueden usar dicho software .Por su generalidad un software de este nivel no es conveniente para la empresa de la UERMP ya que dada su solicitud ellos necesitan un software que presente las características específicas de su empresa.

1.6 Análisis de las tecnologías

1.6.1 Metodología XP

La Programación Extrema (PX), mejor conocida por su nombre en inglés *Extreme Programming* (XP), es una de las llamadas metodologías ágiles de desarrollo de *software* más exitosas de los tiempos recientes. Cada día se genera incontable información sobre el tema, generalmente en inglés. La programación extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Es una metodología para proyectos de corta duración, con el objetivo de responder rápido a las necesidades del cliente, además de que potencia al máximo el trabajo en equipo o sea todo el grupo está completamente vinculado al proyecto a realizar.

Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se puede hacer pruebas de las fallas que pudieran ocurrir. Es como adelantarse a obtener los posibles errores.

Re fabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

Características principales de XP

- ✓ Ligerero.
- ✓ Cercano al desarrollo.
- ✓ Se basa en historias de usuario
- ✓ Fuerte comunicación con el cliente.
- ✓ El código fuente pertenece a todos.
- ✓ Programación por parejas.
- ✓ Prueba como base de la funcionalidad.
- ✓ Solo el mínimo de la organización.
- ✓ Pobre en cuanto a documentación.

(Programación Extrema, 2007)

1.6.2 RUP

El *Rational Unified Process* unifica todo el equipo de desarrollo de software y optimiza su comunicación proveyendo a cada miembro de una aproximación al desarrollo de software con una base de conocimiento en línea de acuerdo a las necesidades específicas del proyecto. Usando la navegación en línea del navegador, cada miembro del equipo tiene acceso instantáneo a la base de conocimiento y guía de procesos de RUP desde su escritorio. La base de conocimiento unifica aún más al equipo identificando y asignando responsabilidades, artefactos y tareas de forma que cada miembro del equipo comprenda su contribución al proyecto. Unificando al equipo, se simplifica la comunicación, asegurando la asignación de recursos en forma eficiente, la entrega de los artefactos correctos, y el cumplimiento de los tiempos límite.

Entrega del software operativo con confianza

RUP mantiene al equipo enfocado en producir incrementalmente software operativo a tiempo, con las características requeridas y con la calidad requerida. Las mejores prácticas probadas en la industria, contenidas en RUP, incorporan las lecciones aprendidas de cientos de líderes de la industria y miles de proyectos. Ya no hay necesidad de re-inventar soluciones a desafíos

de la ingeniería de software bien conocidos. Siguiendo el acercamiento al desarrollo iterativo de RUP, es posible entregar a tiempo y con confianza el software.

Características y beneficios

No existen dos proyectos de desarrollo de software que sean iguales, cada uno tiene prioridades, requerimientos, y tecnologías muy diferentes. Sin embargo, en todos los proyectos, se debe minimizar el riesgo, garantizar la predictibilidad de los resultados y entregar software de calidad superior a tiempo. RUP, es una plataforma flexible de procesos de desarrollo de software que ayuda proveyendo guías consistentes y personalizadas de procesos para todo el equipo de proyecto.

- ✓ Las mejores prácticas más probadas de la industria son las mejores prácticas de desarrollo adoptadas en cientos de proyectos mundialmente enseñados como parte de los currículos en cientos de universidades, la metodología RUP se convirtió rápidamente en el estándar del factor para el proceso de desarrollo en la industria de software.
- ✓ Proceso hecho práctico diferente de otras metodologías comerciales, la plataforma RUP hace que el proceso sea práctico con bases de conocimiento y guías para ayudar en el despegue de la planificación del proyecto, integrar rápidamente a los miembros del equipo y poner en acción el proceso personalizado.
- ✓ Se adapta a las necesidades de los proyectos. Solo la plataforma RUP proporciona un marco de trabajo de proceso configurable que permite seleccionar e implantar los componentes específicos de proceso necesarios para proporcionar un proceso consistente y customizado para cada equipo y proyecto.

Una de las mejores prácticas centrales de RUP es la noción de desarrollar iterativamente. RUP organiza los proyectos en términos de disciplinas y fases, consistiendo cada una en una o más iteraciones. Con esta aproximación iterativa, el énfasis de cada trabajo flexible variará a través del ciclo de vida. La aproximación iterativa ayuda a mitigar los riesgos en forma temprana y continua, con un progreso demostrable y frecuentes versiones ejecutables. (GSInnova, 2010)

1.6.3 Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar altamente productivos.

En Scrum se realizan entregas parciales y regulares del resultado final del proyecto, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto. (proyectos agiles.org, 2009)

1.7 ¿Qué es una aplicación Web?

Una aplicación Web es una interface entre un formulario diseñado específicamente para cubrir con las necesidades de su negocio y su información que actualmente tiene hacia el interior de su empresa, como pueden ser sistemas administrativos, inventarios, facturación, cuentas por cobrar, productos, etc. La información puede ser de dominio público o restringido a ciertas personas a través de un nombre de usuario y contraseña con el objetivo de que cualquier persona pueda consultarla e interactuar con ella desde Internet.

El desarrollo de una aplicación Web le permitirá publicar un catálogo electrónico de productos, manejo de inventarios, órdenes de compra, publicación de información con acceso restringido a ciertos usuarios, actualización y mantenimiento de su sitio Web y en general, le permitirá publicar cualquier tipo de información que se pueda incorporar a una base de datos.

CAPÍTULO I

Conjunto de aplicaciones o tecnologías para ínter operar en la web, estas tecnologías o aplicaciones intercambian datos entre si con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la web. En todo este proceso intervienen una serie de tecnologías que hacen posible una circulación de información que pueden ser transmitidos a través HTTP, SMTP entre otras. (Navarmedia)

1.8 HTML

El HTML, *Híper Texto Marcus* Lenguaje (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto. Fue creado en 1986 por el físico nuclear Tim Berners-Lee; el cual tomo dos herramientas preexistentes: El concepto de Hipertexto (Conocido también como *link* o ancla) el cual permite conectar dos elementos entre si y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizara 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) o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen, se deben guardar con la extensión htm o html.

Estos documentos pueden ser mostrados por los visores o "navegadores" de paginas Web en Internet, como Netscape Navegador, Mosaico, Opera y Microsoft Internet Explorer.

También existe el HTML Dinámico (DHTML), que es una mejora de Microsoft de la versión 4.0 de HTML que le permite crear efectos especiales como, por ejemplo, texto que vuela desde la página palabra por palabra o efectos de transición al estilo de anuncio publicitario giratorio entre página y página.

Para crear una página web se pueden utilizar varios programas especializados en esto, como por ejemplo, el Microsoft Front Page o el Macromedia Dreamweaver 3. Otra forma de diseñar

un archivo .html, es copiar todo en el Bloc de Notas del Windows, ya que este sencillo programa cumple con un requisito mínimo que es la posibilidad de trabajar con las etiquetas con las que trabaja este lenguaje.

A continuación se muestra las etiquetas más comunes que deben aprenderse para hacer una página Web.

Estructura de los documentos de HTML

Si se tiene en cuenta el contenido del documento, todos los documentos de HTML bien escritos comparten una estructura en común. Un documento de HTML empieza con la etiqueta <HTML>, que es la que encerrará el documento actual. Contiene dos secciones primordiales: la cabecera y el cuerpo encerrados respectivamente por los elementos cabeza y cuerpo. La cabecera puede contener información y siempre contiene el título del documento encerrado por el elemento. En el cuerpo se encuentra todo el contenido del documento, ya sea, texto, imágenes, sonidos, hipervínculos, etc. (Ravioli, 200)

1.8.1 CCS

CSS es un lenguaje de programación web, usado para dar mayor personalización a nuestras páginas web y mantener el mismo estilo en múltiples páginas. Desgraciadamente no todos los navegadores soportan muy bien CSS pero pronto eso mejorara.

Usando CSS, uno puede definir los colores, fondos, bordes, margen, alineaciones, letra, tamaño y muchas cosas más para casi todas las partes de una paginan web.

Cascada, describe muchas cosas; primero que todo significa que muchos estilos pueden ser usados mientras son integrados por el navegador para proveer un estilo para cada elemento.

Cascada también significa que objeto hereda por su objeto padre o clase. Los objetos son identificados por los tags de elementos. El objeto más grande en la estructura es el cuerpo. Si un elemento no tiene su propio CSS definido y el cuerpo si, este heredara las propiedades del cuerpo.

1.9 Lenguaje de programación de lado del cliente (Java script)

Java script es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

Muchos confunden el Java script con el Java pero ambos lenguajes son diferentes y tienen sus características singulares. Java script tiene la ventaja de ser incorporado en cualquier página web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado.

Java por su parte tiene como principal característica ser un lenguaje independiente de la plataforma. Se puede crear todo tipo de programa que puede ser ejecutado en cualquier ordenador del mercado: Linux, Windows, Apple, etc. Debido a sus características también es muy utilizado para internet.

Como síntesis se puede decir que Java script es un lenguaje interpretado, basado en prototipos, mientras que Java es un lenguaje más orientado a objetos.

Java script es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con Java script se puede crear diferentes efectos e interactuar con nuestros usuarios.

Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros. (EcuRed, 2010)

1.10 Lenguaje de programación de lado del servidor

1.10.1 PHP

El PHP es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. Con PHP se puede hacer cualquier cosa que se pueda realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. Un sitio con páginas dinámicas es el que permite interactuar con el visitante, de modo que cada usuario que visita la página vea la información modificada para requisitos particulares. Las aplicaciones dinámicas para el Web son frecuentes en los sitios comerciales, donde el contenido visualizado se genera de la información alcanzada en una base de datos u otra fuente externa.

PHP ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día con los códigos comparables en otros lenguajes. Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores.

Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.

PHP es la opción natural para los programadores en máquinas con Linux que ejecutan servidores web con Apache, pero funciona igualmente bien en cualquier otra plataforma de UNIX o de Windows, con el software de Netscape o el servidor de web de Microsoft. Para trabajar con capacidades PHP, se puede conseguir mayor información en PHP.net, sitio encargado de mantener al día a todos los desarrolladores con las últimas descargas relacionadas con el lenguaje y documentación. (S, 2003)

1.10.2 ASP.net

El ASP.net de Microsoft es una tecnología de scripts que corren en el servidor y pueden ser utilizados para crear aplicaciones dinámicas e interactivas en el Web. Una página ASP.net es una página de HTML que contiene scripts que son procesados por un servidor Web antes de ser enviados al navegador del usuario. Usted puede combinar el ASP con el Lenguaje Extensible de Marcas (XML) y el Lenguaje de Marcas de Hipertexto (HTML) para crear poderosos sitios Web interactivos. El código ASP.net es más “compacto” que el código ASP; los scripts requeridos para realizar una función dada son más cortos en ASP.net que en ASP. Debido a que los scripts que corren del lado del servidor están construidos en una página regular de HTML, pueden ser entregados en casi cualquier navegador, Un archivo ASP.net puede ser creado utilizando cualquier herramienta de edición, como el bloc de notas (Definición de ASP.net, 2008).

1.10.3 PHP5

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 un potente lenguaje de *script* del lado del servidor, que se utiliza principalmente para generar páginas de forma dinámica, es tan potente que se utiliza para muchísimas cosas más: generar imágenes, generar PDF, atacar diferentes protocolos de Internet (LDAP, IMAP, POP3, FTP, TELNET, JABBER), como lenguaje de *script* genérico. La sintaxis gramática y funciones del lenguaje son muy sencillas, su compatibilidad con otros sistemas se incrementa agregando nuevas funciones. La manera de manejar los datos, las funciones disponibles y la documentación son realmente excepcionales. La mayor parte de su sintaxis ha sido tomada de C, Java y PERL con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML.

Ventajas de PHP5:

- ✓ Mejor soporte para la Programación Orientada a Objetos, que en versiones anteriores era extremadamente rudimentario.
- ✓ Mejoras de rendimiento.
- ✓ Mejor soporte a XML (XPath, DOM, etc.).
- ✓ Soporte integrado para SOAP.
- ✓ Manejo de excepciones.
- ✓ Soportado por una gran comunidad de desarrolladores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- ✓ Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- ✓ Entre las ventajas de PHP se pueden destacar que este lenguaje es completamente expandible, está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código. Muchas interfaces distintas para cada tipo de servidor.

1.11 Servidor web Apache

El servidor Apache se desarrolla dentro del proyecto HTTP Servidor de la Apache *Software* Fundación. Presenta características como, mensajes de errores altamente configurables, bases de datos de autenticación y negociado de contenido. Tiene amplia aceptación en la red convirtiéndose en el servidor HTTP más usado y el mejor servidor de páginas web en el mundo.

La arquitectura que utiliza es cliente-servidor, el cliente hace la petición al servidor y este atiende dicha petición. El protocolo utilizado para la transferencia de hipertexto es HTTP que está basado en el envío de mensajes y establece el conjunto de normas mediante las cuales se envían las peticiones de acceso a una web y la respuesta de esa web.

Ventajas de Apache:

- ✓ Código abierto.
- ✓ Extensible.

- ✓ Fácil de conseguir ayuda/soporte.
- ✓ Es multiplataforma, aunque idealmente está preparado para funcionar bajo Linux.
- ✓ Muy sencillo de configurar.
- ✓ Amplias librerías de PHP y PERL a disposición de los programadores.
- ✓ Posee diversos módulos que permiten incorporarle nuevas funcionalidades, estos son muy simples de cargar.
- ✓ Es capaz de utilizar lenguajes como PHP, TCL, Pitón, etc.

1.12 Gestores de base de datos a utilizar

1.12.1 Postgre sql

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. El director de este proyecto es el profesor Michael Stonebraker, y fue patrocinado por la Agencia (DARPA), y la Fundación Nacional de Ciencias (NSF).

PostgreSQL es una derivación libre (OpenSource) de este proyecto, y utiliza el lenguaje SQL92/SQL99, así como otras características que se comentan más adelante.

Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

Características de PostgreSQL:

A continuación se enumeran las principales características de este gestor de bases de datos:

- ✓ Implementación del estándar SQL92/SQL99.
- ✓ Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. También permite la creación de tipos propios.

CAPÍTULO I

- ✓ Incorpora una estructura de datos array.
- ✓ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- ✓ Permite la declaración de funciones propias, así como la definición de disparadores.
- ✓ Soporta el uso de índices, reglas y vistas.
- ✓ Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- ✓ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

1.12.2 Oracle

Es manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información.

Es el conjunto de datos que proporciona la capacidad de almacenar y acude a estos de forma consecuente con un modelo definido como relacional. Además es una suite de productos que ofrece una gran variedad de herramientas.

Es el mayor y más usado Sistema Manejador de Base de Datos Relacional (RDBMS) en el mundo. La Corporación Oracle ofrece este RDBMS como un producto incorporado a la línea de producción. Además incluye cuatro generaciones de desarrollo de aplicación, herramientas de reportes y utilitarios.

Oracle corre en computadoras personales (PC), microcomputadoras, y computadoras con procesamiento paralelo masivo. Soporta unos 17 idiomas, corre automáticamente en más de 80 arquitectura de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código. Esto es porque más el 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos.

1.12.3 MySQL

Es un gestor de base de datos. Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda. Es una base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL. El código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales. Es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet.

1.13 Marco de trabajo

1.13.1 Cake php

Es un Marco de trabajo para Php. Facilita la creación de administradores para el manejo de contenido estándar (insertar, editar y eliminar registros), facilita la validación de los campos de los formulario, previene de ataques por inyección de SQL, facilita el testeado de las aplicaciones y también tiene un sistema de plantillas rápido y flexible.

1.13.2 Symfony

Symfony es un Marco de trabajo PHP que facilita el desarrollo de las aplicaciones web. Symfony se encarga de todos los aspectos comunes y aburridos de las aplicaciones web, dejando que el programador se dedique a aportar valor desarrollando las características únicas de cada proyecto.

Si todavía programas aplicaciones PHP a mano, estás pidiendo el tiempo y mucho dinero. Symfony aumenta exponencialmente tu productividad y te ayuda a mejorar la calidad de tus aplicaciones web aplicando todas las buenas prácticas y patrones de diseño que se han definido para la web.

Symfony es además el Marco de trabajo más documentado del mundo, ya que cuenta con miles de páginas de documentación distribuidas en varios libros gratuitos y decenas de tutoriales.

Características

- ✓ Fácil de instalar y configurar en sistemas Windows, Mac y Linux
- ✓ Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MS SQL Server)
- ✓ Compatible solamente con PHP 5 desde hace años, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP
- ✓ Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional
- ✓ Preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización
- ✓ Flexible hasta cualquier límite y extensible mediante un completo mecanismo de plugins
- ✓ Publicado bajo licencia MIT de software libre y apoyado por una empresa comprometida con su desarrollo
- ✓ Traducido a más de 40 idiomas y fácilmente traducible a cualquier otro idioma

1.14 NetBeans IDE

Nació como un proyecto estudiantil en República Checa en 1996. Su nombre original era Xelfi. Este fue el primer IDE para Java, escrito en Java. Róman Stanek, emprendedor, invierte en Xelfi y nace un negocio. De la unión de *Network* y Java Beans nace NetBeans. Un IDE multilenguaje, completo y modular con soporte para Java SE, Java EE, Java ME. Tiene gran cantidad de módulos de terceros (*plugins*), desarrollo intuitivo arrastrar y soltar, y completa código. Es gratis y de código abierto y posee una gran comunidad de usuarios y desarrolladores.

Características principales:

- ✓ Mejoras en el editor de código.

- ✓ Instalación y actualización más simple.
- ✓ Enlazar datos con el *Swing* GUI.
- ✓ Integrado de perfiles, perfiles de "puntos".
- ✓ Características visuales para el desarrollo web.
- ✓ Creador gráfico de juegos para celulares.
- ✓ Mejoras para SOA y UML.
- ✓ Soporte para PHP.

1.15 Arquitectura

La arquitectura es el esqueleto o base de una aplicación. Representa la organización fundamental de un sistema. Desde los pequeños programas hasta los sistemas más grandes poseen una estructura y un comportamiento que los hace clasificables según su "arquitectura". En la Web es muy común la utilización de la arquitectura "3-capas", "n-capas", "MVC", entre otras. El marco de trabajo Symfony, seleccionado para desarrollar la aplicación, implementa a su vez el patrón arquitectónico MVC, es por ello que se adopta esta arquitectura para el desarrollo de la propuesta de solución.

Modelo Vista Controlador (MVC)

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

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

CAPÍTULO I

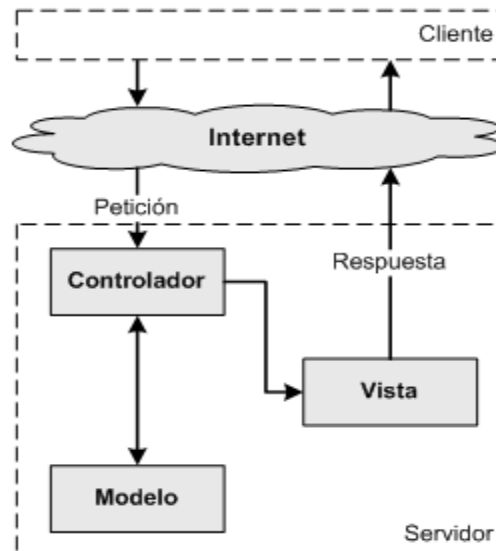


Figura 1. Patrón Modelo Vista Controlador

1.16 Conclusiones

Como conclusiones del capítulo se puede decir que de los sistemas estudiados, los mismos destacan por su potencia, alcance y probada eficacia en el manejo y control de los medios informáticos en sus respectivas ramas. Sin embargo ninguno se ajusta a las necesidades y requerimientos específicos de la UERMP, lo que hace descartable su uso. Basado en todo lo anteriormente descrito se propone una solución utilizando para su desarrollo, herramientas libres de costo, con una probada reputación y tiempo de explotación en el mercado.

- ✓ Como metodología de desarrollo se optó por una metodología ágil por todos los beneficios que está trae consigo, en este caso se habla de Xp que respalda con sus prácticas todo el proceso de desarrollo, obteniéndose de esta forma un proceso de *software* completo.
- ✓ Para realizar el diseño se utilizará la herramienta case Visual Paradigm Enterprise Edición (EE_VP_UML), porque soporta el ciclo de vida completo de un *software* y la Universidad de las Ciencias Informática paga la licencia.
- ✓ Como marco de trabajo se utilizará Symfony porque es un *marco de trabajo* maduro, bien documentado y con una gran comunidad que lo apoya. Su estructura modelo vista controlador facilita su estudio.

CAPÍTULO I

- ✓ Como sistema gestor de base datos por sus características se utilizará PostgreSQL, puesto que está disponible en cualquier plataforma. Además, posee una gran escalabilidad haciéndolo idóneo para sitios web.
- ✓ Para la programación del lado del cliente se utilizará como lenguaje, JavaScript para mejorar la gestión de la interfaz cliente/servidor y CSS para realizar la presentación de la página estructurada en HTML y para el lado del servidor PHP5 debido a que es orientado a objeto, multiplataforma, y permite crear páginas web dinámicas.
- ✓ Como servidor web, Apache que es un servidor web de código abierto que funciona sobre cualquier plataforma, potente y que ofrece un servicio estable y sencillo de mantener y configurar.
- ✓ Como IDE de desarrollo se utilizará NetBeans porque es el único que da soporte para Symfony.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En el presente capítulo se abordan temas de las principales características del sistema en desarrollo, así como la propuesta de solución elaborada por el equipo de trabajo. Además se plasman las fases iniciales de la metodología XP (*Extreme Programming*) que se aplicó a este proyecto informático, estas fases son: la de Exploración, la de Planificación y la de Diseño.

2.2 Propuesta de solución

Para dar solución a la problemática planteada en la investigación se decidió crear una herramienta que debe permitir gestionar los medios básicos informáticos de la empresa, mediante una interfaz gráfica, sencilla y amigable. Además será capaz de mostrar en una interfaz, el control de entrada y salidas de todos los medios básicos informáticos dada una fecha determinada, así como la posibilidad de imprimir dicho control y dicho de paso un motor de búsqueda donde se pueda buscar, controlar y localizar dichos medios en cualquiera de las empresas de UERMP. Así como una interfaz para gestionar los medios básicos informáticos por tipo de equipos, por empresas y por marcas. Por lo que se quiere crear un software capaz de automatizar muchos de los procesos que se realizan manualmente y mejorar la gestión de los que se realizan informativamente.

A continuación se muestran los procesos del negocio:

CAPÍTULO II

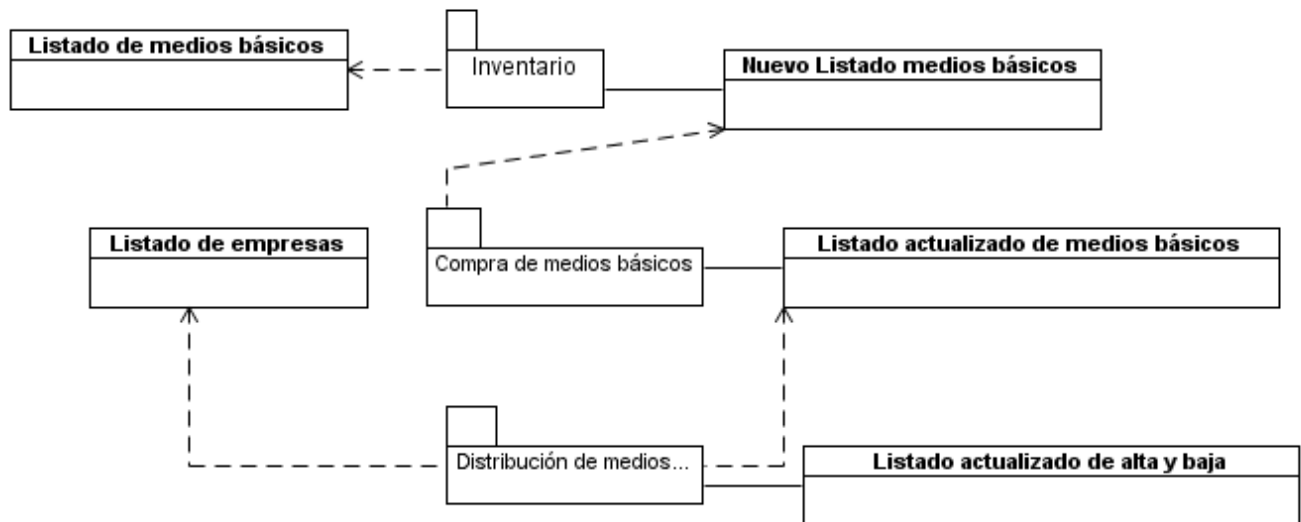


Figura 2: Procesos del negocio.

Especificación de los requerimientos del software

Requerimientos funcionales

1. Autenticar usuario.
 - .Introducir nombre de usuario y contraseña.
 - Validar los datos introducidos.
 - Otorgar los accesos al usuario
2. Gestionar Medios básicos informáticos
 - Buscar medios básicos informáticos.
 - Adicionar medios básicos informáticos.
 - Eliminar medios básicos informáticos.
 - Modificar medios básicos informáticos.
3. Controlar entrada y salida de los medios básicos informáticos.
4. Mostrar entrada y salida de los medios básicos informáticos.
5. Obtener datos de los medios básicos informáticos.
6. Mostrar Medios básicos informáticos por sucursales.

7. Mostrar Medios básicos informáticos por tipo de equipos.
8. Control de los movimientos de los medios básicos informáticos.
9. Mostrar Listado de mantenimiento de los medios básicos informáticos.
10. Imprimir listado de todos los medios básicos informáticos.

2.3 Fase de planificación

Es la primera de las 4 fases de la metodología XP y en ella se identifican las historias de Usuarios y también se crea la cobertura idónea para la familiarización del equipo de trabajo con las herramientas y las nuevas tecnologías elegidas para realizar el proyecto. Se sientan las bases de la comunicación que es un pilar fundamental cuando se utiliza la mencionada metodología.

2.4 Historias de Usuario

La historia de usuarios es el primer artefacto generado en este capítulo. Es una forma particular en la que se llaman a los requerimientos en esta metodología. Estas historias de usuarios se redactan desde el punto de vista del cliente aunque el criterio de los desarrolladores también se tiene en cuenta en el momento de formularlas. En la fase se detectaron requerimientos que se convirtieron en historias de usuarios, las cuales se listan a continuación:

- ✓ Autenticar Usuario.
- ✓ Gestionar Medios básicos informáticos
- ✓ Controlar entrada y salida de los medios básicos informáticos.
- ✓ Mostrar entrada y salida de los medios básicos informáticos.
- ✓ Obtener datos de los medios básicos informáticos.
- ✓ Mostrar Medios básicos informáticos por sucursales.
- ✓ Mostrar Medios básicos informáticos por tipo de equipos.
- ✓ Control de los movimientos de los medios básicos informáticos.
- ✓ Mostrar Listado de mantenimiento de los medios básicos informáticos.

CAPÍTULO II

- ✓ Imprimir listado de todos los medios básicos informáticos.

Estas historias de usuario no son más que los requerimientos funcionales, que especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. Por lo general se describen mejor a través del modelo de casos de uso (MCU) y los casos de uso (CU) como tal. Por lo tanto los requerimientos funcionales especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. A continuación se muestra unas de las historia de usuario que se desarrolla en este capítulo.

Historia de Usuario	
Número: 3	Usuario: Informático
Nombre historia: Modificar medios básicos informáticos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Néstor Alain Arnet	
Descripción: El sistema será capaz de modificar los Medios Básicos Informáticos, guardándolo en la base de datos.	

Tabla 2 Historia de usuario Modificar medios básicos informáticos

2.5 Personas relacionadas con el sistema

Existen diferentes tipos de personas que trabajan directamente con el sistema o sea interactúan con él obteniendo resultados en los procesos desarrollados. Ellos trabajan directamente con el sistema en funcionamiento e inciden en el resultado de la automatización de los procesos que realiza esta aplicación, a estos individuos se les denomina: personas relacionadas con el sistema. Esto se muestra en la tabla No.3

Personas relacionadas con el Sistema	Justificación
Informático	Es el encargado de interactuar y efectuar todas las tareas del sistema.

Tabla 3. Personas relacionadas con el sistema

2.6 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Teniendo en cuenta las necesidades del cliente, así como el nivel tecnológico que posee el mismo, y las características del entorno en que se desarrolla esta aplicación, se definen las siguientes propiedades o cualidades que dicho sistema debe poseer.

Como se utiliza la metodología XP este tipo de requisitos no que queda plasmado en las planillas principales que en este caso son las historias de usuario. Ya que los mismos son objeto generalmente del desconocimiento de los clientes. Pero es tarea del equipo de desarrollo hacerle entender algunos conceptos al cliente mediante un intercambio profundo y hasta cierto punto darle opciones de cómo hacerlos cumplir siendo él, el más favorecido.

CAPÍTULO II

Requisitos de usabilidad

El software está diseñado para ser utilizado por personas que tengan relación con la información que se maneja. Los usuarios deben de tener conocimientos informáticos elementales. El sistema deberá poseer una disponibilidad total para ser usado durante todo el horario laboral, siempre que no existan problemas técnicos que anulen o afecten el funcionamiento de la maquina servidora o cualquiera de las estaciones de trabajo.

Requisitos de rendimiento.

Es importante hacer una programación orientada a objetos muy exquisita ya que es necesario que el software sea funcional a un 100 % y utilice además de forma eficaz los recursos físicos de cada una de las computadoras en donde se ejecute. La aplicación debe ejecutarse con el mínimo de recursos para el que fue prevista y además debe el tiempo de respuesta de la aplicación a las diferentes peticiones de los usuarios debe ser muy breve o el menor posible.

Requisitos de portabilidad.

La aplicación está diseñada solamente para computadoras con sistema operativo Windows XP Service Pack 2 o cualquier versión superior de Windows. El servidor de Base de Datos estará ubicado en la PC que se encuentra en la empresa principal aquí en Ciudad de la Habana en Miramar.

Requisitos de seguridad.

La empresa objeto de nuestra investigación cuenta con políticas de seguridad establecidas así como sistemas informáticos de seguridad.

Integridad:

Debe generarse una información consistente y protegida contra alteraciones de cualquier tipo. La información debe permanecer intacta bajo cualquier condición. Por lo que es preciso tomar medidas lógicas, y físicas de protección en la institución.

Disponibilidad:

El sistema debe estar disponible durante todo el horario laborable ya que los usuarios necesitan laborar con el mismo y así puedan estar haciéndole peticiones continuas y recibiendo la respuesta adecuada en el momento adecuado.

CAPÍTULO II

Requisitos de software.

En el servidor donde se hospedará la base de datos se requiere del sistema operativo *Windows XP Service Pack 2*. Y en el resto de las estaciones de trabajo la misma versión o superiores, pero además el sistema puede ser accedido desde estaciones con sistema operativo *Linux*, ya que el sistema que se elabora es multiplataforma.

Requisitos de hardware.

El servidor debe tener al menos 512 de RAM y al menos 60 GB de memoria física. Las computadoras clientes deben tener al menos 256 de RAM.

2.7 Estimación de esfuerzo por Historia de Usuario

A continuación se muestran las historias de usuarios en correspondencia con los puntos estimados (en semanas) por parte de los desarrolladores para satisfacer las necesidades del cliente.

Historia de Usuario	Puntos Estimados
Autenticar Usuario	1,5
Gestionar Medios básicos informáticos	3,0
Controlar entrada y salida de los Medios básicos informáticos	2,0
Mostrar entrada y salida de los Medios básicos informáticos	2,0
Obtener datos de los Medios básicos informáticos	1
Mostrar Medios básicos informáticos por sucursales	1
Mostrar Medios básicos informáticos por tipo de equipos	1
Control de los movimientos de los medios básicos informáticos	2,0

CAPÍTULO II

Mostrar Listado de mantenimiento de los medios básicos informáticos	1
Imprimir Listado de medios básicos informáticos	1

Tabla 4. Estimación de esfuerzo por historia de Usuario

2.8 Plan de iteraciones

Conocidas ya cada una de las historias de usuario y el esfuerzo que se requiere para desarrollar cada una de ellas. Se procede a dividir el trabajo en iteraciones para así poder tener un trabajo incremental donde la principal idea sea la comunicación entre el equipo de trabajo en el cual también está insertado el cliente. Para que exista un equilibrio entre las diferentes secuencias de trabajo se procede a dividir el proyecto en 3 iteraciones que se describen a continuación. Para ello se trató de hacer un balance para que cada iteración fuera escenario de conjuntos de historias de usuarios que demandaran un desempeño aproximado del equipo de trabajo.

Iteración 1

Durante la primera iteración se realizarán las tareas más sencillas y de menos costo computacional. Las cuales requieran de menos esfuerzo por parte de los desarrolladores. Y a la vez son las funcionalidades más independientes, que no requieran de muchas funcionalidades para ejecutarse. Al terminar esta iteración ya habrá una concepción más aterrizada de cómo será el sistema en sí pero de forma sencilla.

Iteración 2

En esta iteración se les dará cumplimiento a las historias de usuario de mayor complejidad. Haciendo que el sistema esté funcionalmente casi listo. Esta es la etapa de trabajo en la que se necesita de mayor esfuerzo ya que exige mucho de los desarrolladores. Todas las historias que incidan en la lógica de la programación del sistema deben quedar implementadas.

Iteración 3

CAPÍTULO II

Deben realizarse las funcionalidades menos trascendentales que no incidan en la lógica, que de no implementarse el sistema cumpla con las funcionalidades básicas establecidas por el cliente. Pero se implementan como parte del contrato que se adquirió con el cliente.

2.9 Planificación de duración de iteraciones

La planificación no es más que el establecimiento de un intervalo de tiempo para realizar las actividades que se encuentran en cada una de las fases. Como nuestro equipo de trabajo cuenta con dos integrantes, se tomó la decisión de que algunas de las historias de usuario serian realizadas en conjunto pero otras se piensan que no son necesarias. Entonces para hacer un cálculo estimado de la iteración se suman los tiempos aproximados de cada uno de los integrantes en la iteración y finalmente el tiempo de la iteración no es más que el mayor de las sumatorias de los tiempos estimados de cada miembro. O sea el tiempo de la iteración será igual al mayor de los tiempos de la sumatoria de la estimación de miembros, lo cual se muestra a continuación en la tabla 3.

Iteración	Historia de Usuario	Puntos estimados	Puntos de iteración
1	Autenticar Usuario	1,5	3
	Gestionar medios básicos informáticos	1,5	
2	Controlar entrada y salida de los medios básicos informáticos	2	6
	Mostrar entrada y salida de los medios básicos informáticos	2	

CAPÍTULO II

	Obtener datos de los medios básicos informáticos	2	
3	Mostrar medios básicos informáticos por sucursales	1	5
	Mostrar medios básicos informáticos por tipo de equipos	1	
	Control de los movimientos de los medios básicos informáticos	1	
	Mostrar Listado de mantenimiento de los medios básicos informáticos	1	
	Imprimir Listado de medios informáticos	1	

Tabla 5. Planificación de duración de iteraciones.

2.10 Plan de entregas

Artefacto	Hito	Entrega
Sistema de gestión de los medios básicos informáticos	Final de la primera iteración.	15-18 febrero 2010
Sistema de gestión de los medios básicos informáticos	Final de la segunda iteración.	5 abril- 9 abril 2010

CAPÍTULO II

Sistema de gestión de los medios básicos informáticos	Final de la tercera iteración.	15 junio 2010
---	--------------------------------	---------------

Tabla 6. Planificación de entrega

2.11 Conclusiones

Con la elaboración de este capítulo se realizó el estudio de los procesos del negocio donde se pudo definir la inexistencia de los mismos, por lo que fue necesario realizar un modelo conceptual para lograr un entendimiento preciso de lo que se requería. Se tiene una vista del sistema a desarrollar, se plasmaron los requisitos funcionales y no funcionales a tener en cuenta, obteniendo una idea general de las funcionalidades que debe cumplir la aplicación. Se planifica además, el desarrollo de las iteraciones a realizar, así como la planificación de entrega y el orden en que se implementarán las historias de usuario de acuerdo con la prioridad que se le asignó.

Capítulo 3: Diseño Implementación y pruebas

3.1 Introducción

En el presente capítulo se modelan los artefactos que exige la metodología ágil XP para la fase de desarrollo. Para el diseño de las aplicaciones, esta metodología no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC que en este capítulo se exponen. Además, se realizan las pruebas que es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

3.2 Fase de diseño

Patrones de diseño

Modelo Vista Controlador

- ✓ El patrón de diseño MVC* define una manera de organizar el código de acuerdo con su naturaleza. Este patrón separa el código en tres capas:
- ✓ La capa **Modelo** define la lógica de negocio (la base de datos pertenece a esta capa). Symfony guarda todas las clases y archivos relacionados con el modelo en el directorio `lib/model/`.
- ✓ La **Vista** es con lo que el usuario interactúa. En Symfony, la vista es principalmente la capa de plantillas PHP. Estas son guardadas en el directorio `templates/`.
- ✓ El **Controlador** es la pieza de código que llama al Modelo para obtener algunos datos que le pasa a la Vista para la presentación al cliente. Todas las solicitudes son gestionadas por un controlador frontal (`index.php` y `Aplicación_dev.php`). Estos controladores frontales delegan la verdadera labor a las acciones, estas acciones son, lógicamente, agrupadas en módulos.

CAPÍTULO III

El patrón representa un mecanismo de mejora de procesos de desarrollo de software, fácil de comprender y aplicar. Permite obtener un sistema claro y bien especificado. La separación de la interfaz del resto del código eliminará problemas como:

- ✓ Que a la complejidad de los cálculos del programa se le sume la complejidad de la interfaz de usuario, lo que proporcionaría un código de mayor calidad y de menor dificultad.
- ✓ Menor rigidez en la interfaz con el usuario, dado que el código HTML no estaría entremezclado con el lenguaje de programación y sería una tarea más fácil cambiar el diseño de la interfaz.

Patrones GRASP

Creador: Todos los módulos del sistema tienen una clase `actions.class.php` que contiene las acciones que hacen al sistema funcional. En esta clase las acciones se encargan de crear los objetos de las clases que representan las entidades, evidenciando de este modo que la clase `actions.class.php` es el “creador” de las entidades.

Experto: Se evidencia este patrón puesto que Doctrine es la librería externa que utiliza Symfony para realizar su capa de abstracción al modelo de datos, encapsulando toda la lógica de los datos y generando las clases con funcionalidades comunes de las entidades. Por tanto, cada clase creada por Doctrine a partir de una entidad es experta en manejar su información.

Controlador: Todas las peticiones web son manejadas por un solo controlador frontal (`frontend.php`), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

Alta Cohesión: Symfony permite la asignación de responsabilidades con alta cohesión, por ejemplo la clase `actions.class.php` tiene la responsabilidad para definir las acciones sobre las plantillas y colabora con otras para realizar diferentes operaciones y crear objetos, está

formada por diferentes funcionalidades que se encuentran estrechamente relacionadas, proporcionando que el software sea flexible frente a grandes cambios.

Bajo Acoplamiento: La clase `actions.class.php` hereda solamente de `sfActions` para lograr un bajo acoplamiento de clases.

Patrones GOF

Creacionales:

Singleton (Instancia Única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Es el caso del controlador frontal, donde hay una llamada a la función `sfContext::getInstance()` que garantiza que siempre se acceda a la misma instancia.

Abstrac Factory (Fábrica Abstracta): Se utiliza este patrón al trabajar con objetos de distintas familias de manera que no se mezclen entre sí, haciendo transparente el tipo de familia concreta que se esté usando. Cuando el marco de trabajo necesita, por ejemplo crear un nuevo objeto, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea.

Estructurales:

Decorator (Decorador): Añade funcionalidad a una clase, dinámicamente. El archivo `layout.php`, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el Layout decorando la misma.

Composite (Objeto compuesto): Permite tratar objetos compuestos como si de uno simple se tratase. Sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol. Esto simplifica el tratamiento de los objetos creados, ya que al poseer todos ellos una interfaz común, se tratan todos de la misma manera.

3.2.1 Tarjetas CRC (Clase, Responsabilidad y Colaboración)

Las tarjetas CRC son técnicas de modelado creadas para ayudar a los desarrolladores de software a crear diseños de clases orientados a responsabilidades. Dichas tarjetas constan de

CAPÍTULO III

tres secciones, nombre de la clase, responsabilidades y colaboradores. La sección responsabilidades se lista cada una de las funciones o tareas que debe ser capaz cumplir un objeto de dicha clase mientras que la sección colaboradores contiene otras clases del diseño que pueden colaborar para proveer datos o funcionalidades.

La metodología XP estipula en toda la bibliografía consultada el uso de las tarjetas CRC como un artefacto obligatorio durante el desarrollo de un proyecto. Debido a los beneficios que aporta en cuanto a la comunicación y transparencia del desarrollo, la mayoría de los autores recomiendan que se encuentren en un lugar público a la vista de todo el equipo, donde todos los programadores puedan contribuir a mejorar y refinar el diseño resultante; así como verificar que diferentes clases no dupliquen responsabilidades o que una interfiera con las funcionalidades de la otra; problemas muy frecuentes a la hora de desarrollar una arquitectura bajo paradigmas de orientación a objetos. A continuación se muestran dichas tarjetas:

Historia Usuario: Gestionar Medios básicos informáticos	
Funcionalidades	Colaboraciones(HU)
Adicionar Eliminar Modificar Buscar	Autenticar Usuario

Tabla 7. Tarjeta CRC de Gestionar Medios básicos informáticos

Historia Usuario: Autenticar Usuario	
Funcionalidades	Colaboraciones(HU)
Iniciar sección Cerrar sección	-

Tabla 8. Tarjeta CRC de Autenticar Usuario

CAPÍTULO III

Historia Usuario: Controlar entrada y salida de los Medios Básico Informáticos	
Funcionalidades	Colaboraciones(HU)
Obtener inventario de los medios básicos informáticos.	Autenticar Usuario Mostrar entrada y salida de los medios básicos informático Buscar medios básicos informáticos

Tabla 9. Tarjeta CRC de Controlar entrada y salida de los Medios Básico Informáticos

Historia Usuario: Mostrar entrada y salida de los Medios Básico Informáticos	
Funcionalidades	Colaboraciones(HU)
Buscar inventario de los medios básicos informáticos.	Autenticar Usuario Buscar medios básicos informáticos Obtener inventario de los medios básicos informáticos

Tabla 10. Tarjeta CRC Mostrar entrada y salida de los Medios Básico Informáticos

Historia Usuario: Obtener datos de los Medios básicos informáticos	
Funcionalidades	Colaboraciones(HU)
Obtener datos de los medios básicos informáticos.	Autenticar Usuario Buscar medios básicos informáticos Mostrar medios básicos informáticos por sucursales

Tabla 11. Tarjeta CRC de Obtener datos de los Medios básicos informáticos

CAPÍTULO III

Historia Usuario: Mostrar Medios básicos informáticos por sucursales	
Funcionalidades	Colaboraciones(HU)
Mostrar datos de los medios básicos informáticos	Autenticar Usuario Buscar medios básicos informáticos Obtener datos de los medios básicos informáticos por sucursales Buscar inventario de los medios básicos informáticos

Tabla 10. Tarjeta CRC de Mostrar Medios básicos informáticos por sucursales

Historia Usuario: Mostrar Medios básicos informáticos por tipo de equipos	
Funcionalidades	Colaboraciones(HU)
Mostrar datos de los medios básicos informáticos por tipos de equipos Buscar datos de los medios básicos informáticos por tipo	Autenticar Usuario Buscar medios básicos informáticos Obtener datos de los medios básicos informáticos por sucursales

Tabla 11. Tarjeta CRC Mostrar Medios básicos informáticos por tipo de equipos

Historia Usuario: Control de los movimientos de los Medios básicos informáticos	
Funcionalidades	Colaboraciones(HU)
Mostrar Datos del equipo	Autenticar Usuario Buscar medios básicos informáticos Obtener datos de los medios básicos informáticos por sucursales

Tabla 12. Tarjeta CRC de Control de los movimientos de los Medios básicos informáticos

CAPÍTULO III

Historia Usuario: Mostrar Listado de mantenimiento de los Medios básicos informáticos	
Funcionalidades	Colaboraciones(HU)
Mostrar Listado de mantenimiento	Autenticar Usuario Obtener datos de los medios básicos informáticos por sucursales

Tabla 13. Tarjeta CRC Mostrar Listado de mantenimiento de los Medios básicos informáticos

Historia Usuario: Imprimir Listado de los Medios básicos informáticos	
Funcionalidades	Colaboraciones(HU)
Mostrar Listado de medios básicos informáticos	Autenticar Usuario Obtener datos de los medios básicos informáticos
Mostrar Listado de mantenimiento	

Tabla 14. Tarjeta CRC de Imprimir Listado de los Medios básicos informáticos

3.2.2 Tareas de ingeniería

Las tareas de ingeniería son un conjunto de acciones que se realizan con el objetivo de resolver las historias de usuarios, una historia de usuario puede tener una o más tareas de ingeniería en dependencia de la complejidad de la funcionalidad a desarrollar. A continuación se muestra en la tabla No.15 una de las tareas de la ingeniería “Autenticar usuario”, las demás se encuentran anexadas en el documento.

CAPÍTULO III

Tarea de Ingeniería	
Número de la Tarea: 1	Número de la HU: 1
Nombre de la tarea: Autenticar usuario	
Tipo de Tarea: Desarrollo Desarrollo/Corrección/Mejora/Otra(especificar)	
Fecha de Inicio: 09/05/10	Fecha Fin: 10/05/10
Programador Responsable: José Antonio Brizuela Santoyo	
Descripción: El sistema debe crear un formulario con los datos de inicio de sección (usuario y contraseña).	

Tabla 15. Tarea de ingeniería "Autenticar usuario"

3.3 Diseño de la Base Datos

En el proceso de abstracción que conduce a la creación de una base de datos, desempeña una función prioritaria el modelo de datos. El modelo de datos, como abstracción del universo de discurso, es el enfoque utilizado para la representación de las entidades y sus características dentro de la base de datos. Es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia.

3.4 Diagrama de Despliegue

Los diagramas de despliegue muestran nodos, conexiones, componentes y objetos. Los nodos representan objetos físicos con recursos computacionales como procesadores y periféricos; pueden mostrarse como una clase o una instancia, por lo que su nombre sigue la misma sintaxis establecida para clases y objetos. Las conexiones son asociaciones de comunicación entre los nodos, y se etiquetan con un estereotipo que identifica el protocolo de comunicación o la red utilizada. Los componentes son archivos de código ejecutable, que residen y se ejecutan dentro de un nodo; se pueden representar relaciones de dependencia entre los componentes que, de manera similar a las dependencias entre paquetes,

CAPÍTULO III

corresponden al uso de servicios. El diagrama de despliegue de la propuesta de solución en particular estará compuesto por 3 nodos fundamentales. Se tendrá un nodo servidor donde estará desplegado o instalado el sistema. De igual manera la base datos con la que interactúa el sistema estará instalada en otro servidor, la comunicación entre el sistema y la base de datos se realizará mediante el protocolo (el loco). Existe además un nodo que representaría a la PC Cliente con que el usuario se conectará y trabajará con el sistema , esta conexión se realizará a través del protocolo HTTP. Alternativamente la PC Cliente deberá tener conectada una impresora a través de la conexión USB para el manejo de todo lo referente a la impresión de los diferentes reportes que emite el sistema.

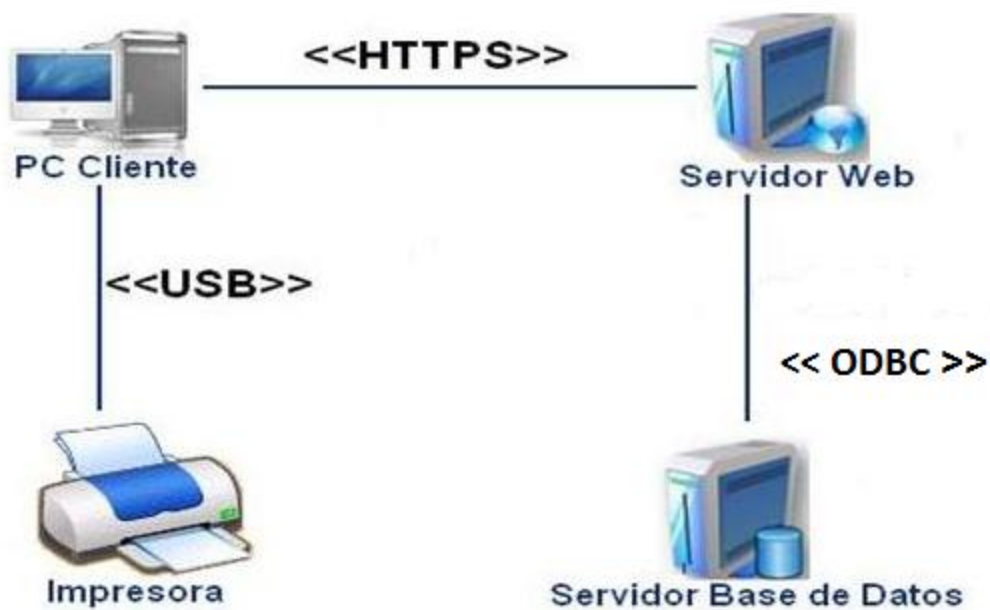


Figura 3. Diagrama de despliegue

3.5 Desarrollo: Ciclo de vida de XP para la etapa de diseño

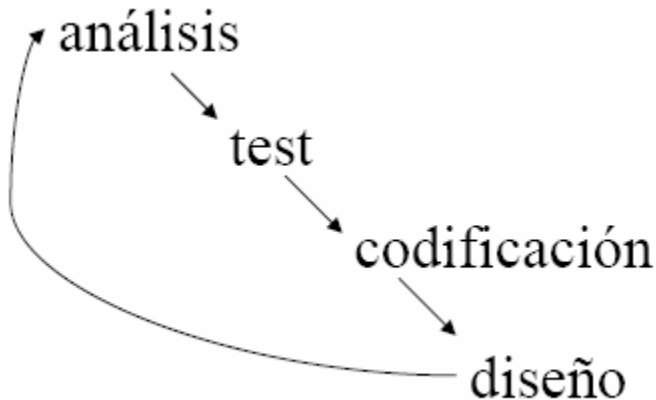


Figura 4. Ciclo de vida de XP para esta etapa

Disponibilidad del cliente

Una de las pocas condiciones que impone la metodología XP es tener al usuario siempre disponible. No sólo para ayudar al equipo de desarrollo, sino formando parte de él. Todas las fases que se realizan en un proyecto XP requieren de comunicación con el usuario, preferiblemente cara a cara, en persona, sin intermediarios.

Durante la reunión del plan de entregas, el usuario propondrá qué historia de usuario se incluye en cada plan. También se negociarán los plazos de entrega. El usuario o cliente tomará las decisiones que le afecten para alcanzar los objetivos de su negocio.

También es necesario que el cliente colabore en la realización de los test. Estos test comprobarán que el sistema está listo para pasar a la fase de producción.

Comprobará los resultados obtenidos y tomará decisiones en cuanto a la utilización o no del sistema realizado. (Escribano, 2002)

Estándares de implementación

El código ha de ser desarrollado siguiendo los estándares de desarrollo para facilitar su lectura y modificación por cualquier miembro del equipo de desarrollo.

Es decisiva, para poder plantear con éxito la propiedad colectiva del código. Ésta sería impensable sin una codificación basada en estándares que haga que todo el mundo se sienta cómodo con el código escrito por cualquier otro miembro del equipo. (Escribano, 2002)

Desarrollar unidad de pruebas

Cuando los test son creados antes que el código, la implementación del código será mucho más rápida. El tiempo empleado en desarrollar un test y algo de código para probarlo es aproximadamente el mismo tiempo que se emplea en crear exclusivamente dicho código. La creación de las unidades de test ayuda al programador a tener una visión a cerca del cómo, en definitiva, del comportamiento del programa. Además, aún se está a tiempo de dar marcha atrás, ya que el programador no ha concluido la implementación. Esta manera de trabajar resulta especialmente beneficiosa en el diseño de complicados sistemas software.

Cualquier característica de un programa para la que no haya un test automatizado, simplemente no existe. (Escribano, 2002)

Programación en parejas

Todo el código que forma parte del plan, será desarrollado por dos personas que trabajarán de forma conjunta en un ordenador. De esta manera, se incrementa la calidad del software desarrollado sin afectar al tiempo de entrega. Se parte de la idea de que este equipo de dos personas posee unos conocimientos similares en cuanto a la tarea, qué van a realizar, es decir, están aproximadamente al mismo nivel. Mientras uno de ellos se encarga de pensar la táctica con la que se va a abordar el problema, el otro se encarga de pensar las estrategias que permiten llevar dichas tácticas a su máximo exponente. Ambos roles son intercambiables. (Escribano, 2002)

3.6 Prueba

¿Dónde encajan las pruebas del sistema en XP?

Uno de los pilares de XP es el proceso de pruebas. XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y

CAPÍTULO III

su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

Las pruebas del sistema tienen como objetivo verificar la funcionalidad del sistema a través de sus interfaces externas comprobando que dicha funcionalidad sea la esperada en función de los requisitos del sistema. Generalmente las pruebas del sistema son desarrolladas por los programadores para verificar que su sistema se comporta de la manera esperada, por lo que podrían encajar dentro de la definición de pruebas unitarias que propone XP.

Sin embargo, las pruebas del sistema tienen como objetivo verificar que el sistema cumple los requisitos establecidos por el usuario por lo que también pueden encajar dentro de la categoría de pruebas de aceptación.

Las pruebas de aceptación son más importantes que las pruebas unitarias dado que significan la satisfacción del cliente con el producto desarrollado y el final de una iteración y el comienzo de la siguiente por esto, el cliente es la persona adecuada para diseñar las pruebas de aceptación.

Pruebas Unitarias

La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial. (Escribano, 2002)

Pruebas de aceptación

El objetivo de estas pruebas es verificar los requisitos, por este motivo, los propios requisitos del sistema son la principal fuente de información a la hora de construir las pruebas de aceptación.

CAPÍTULO III

Las pruebas de aceptación son creadas a partir de las historias de usuario. Durante una iteración la historia de usuario seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. El cliente o usuario especifica los aspectos a testear cuando una historia de usuario ha sido correctamente implementada.

Una historia de usuario puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento y no se considera completa hasta que no supera sus pruebas de aceptación. Esto significa que debe desarrollarse un nuevo test de aceptación para cada iteración o se considerará que el equipo de desarrollo no realiza ningún progreso.

Una prueba de aceptación es como una caja negra. Cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección de las pruebas de aceptación y tomar decisiones a cerca de las mismas.

La garantía de calidad es una parte esencial en el proceso de XP. La realización de este tipo de pruebas y la publicación de los resultados debe ser los más rápido posibles, para que los desarrolladores puedan realizar con la mayor rapidez los cambios que sean necesarios. (Escribano, 2002)

Caso de prueba aceptación	
Número de caso de prueba: 1	Nombre Historia de Usuario: Buscar medios básicos informáticos
Nombre de la persona que realiza la prueba: José Antonio Brizuela Santoyo	
Descripción de la prueba: En la página principal aparecen varias opciones, al ejecutar el link de Gestionar medios básicos informáticos, va hacia la interfaz de Buscar medios básicos informáticos. Se especifica el criterio de búsqueda por el que se desea buscar, se da clic en el botón buscar y aparecerán los resultados de la búsqueda de los medios básicos informáticos en	

CAPÍTULO III

otra página, con la opción de escoger la cantidad de páginas en los que se quiere ver los resultados de la búsqueda.
Condiciones de ejecución:
Entrada/Pasos de ejecución: El administrador escoge los medios básicos informáticos que desea buscar.
Resultado esperado: La búsqueda se realizó con éxito de acuerdo al criterio de búsqueda especificado, se obtuvieron los resultados previstos
Evaluación de la prueba: Satisfactoria

Tabla 16 Buscar Medios Básicos Informáticos

Código y unidad de pruebas

Las unidades de test o pruebas constituyen unos de los pilares básicos de la *Extreme Programming* (XP).

Uno de los errores que se suele cometer es pensar que se puede dejar la construcción de los *test* para los últimos meses en la realización de un proyecto. Descubrir todos los errores que pueden aparecer lleva tiempo, y más si se deja la depuración de todos para el final.

Las unidades de test están directamente relacionadas con el concepto de posesión del código. En cierta manera, una parte del código no será reemplazado si no supera los test que existen para ese código.

Después de cada modificación, se pueden emplear los test para verificar que un cambio en la estructura no introduce un cambio en la funcionalidad. Sin embargo, si se añaden nuevas capacidades al código, se tiene que rediseñar la unidad de test, para adaptarse a la nueva funcionalidad, de esta manera, la probabilidad de que exista un fallo en ambos (test y código) es menor.

Si se crean los test después de la creación del código habría que hacerlos utilizando ese código como un generador, se reubican los fallos del uno al otro. De aquí la importancia de la creación de las unidades de prueba antes que el código, para que sea independiente de este

CAPÍTULO III

Las pruebas se convierten en una herramienta de desarrollo, no un paso de verificación que puede despreciarse si a uno le parece que el código está bien. (Escribano, 2002)

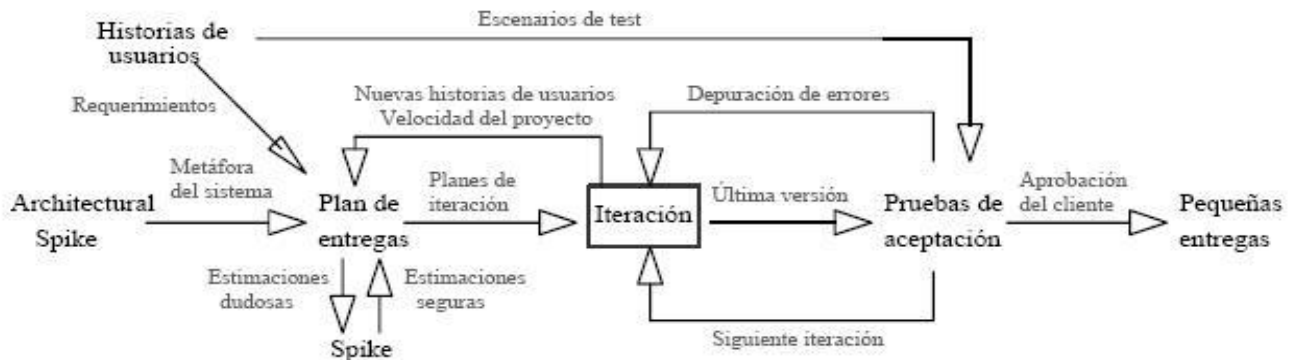


Figura 5. Ciclo completo de XP

3.7 Conclusiones

En este capítulo se desarrolló el modelo del sistema a implementar. Se hizo el diseño del modelo de datos que utilizará la aplicación. Se desarrollaron además las tareas correspondientes para darle solución a las historias de usuarios y las pruebas de aceptación para comprobar que el sistema cumple con los requerimientos del cliente, además de verificar aspectos importantes como es la seguridad de la aplicación. Con este capítulo se culmina la propuesta del sistema a desarrollar. Conviene recordar que ninguna metodología hará el trabajo por ti, porque ninguna metodología trabaja sola.

Capítulo 4: Estudio de factibilidad

4.1 Introducción

Desde los primeros momentos del desarrollo de un *software*, resulta necesario determinar si el mismo resultará factible o no. Para ello se debe realizar un estudio detallado de los beneficios que este aporta y de las inversiones que implicará tanto en la esfera organizativa (entiéndase estructuras, procesos y personas), como en la económica y técnica (teniendo en cuenta habilidades, experiencias, recursos), para llevar a cabo su implementación. Este estudio incluye una planificación del trabajo a realizar referente al tiempo que demorará el desarrollo del *software* a partir de la cantidad de personas requeridas y del tamaño del mismo.

Una de las tareas de mayor importancia en la planificación de proyectos de *software* es el cálculo de la estimación, la cual consiste en determinar con cierto grado de certeza, los recursos de *hardware* y *software*, costo, tiempo y esfuerzo necesarios para el desarrollo de los mismos. En este capítulo se realiza un estudio de la factibilidad para la realización del sistema propuesto, haciendo una estimación del esfuerzo necesario, para el cálculo se decidió utilizar el método COCOMO II, este resulta muy útil para estimar un proyecto en forma global, cuando se tiene un conjunto de casos de uso bastante amplio y con escaso nivel de detalle. Así como los beneficios tangibles e intangibles que reportaría la aplicación y se realiza el análisis de costo y beneficio.

4.2 Características del Proyecto

El primer paso a llevar a cabo para la estimación del proyecto consiste en la obtención de los Puntos de Función desajustados, los cuales están dados por la suma de cada una de las entradas, las salidas y las consultas externas del sistema, así como los archivos lógicos internos y de interfaz externo.

CAPÍTULO IV

Entradas externas.

Entrada de datos del usuario o de control que ingresan desde el exterior del sistema para agregar y/o cambiar datos a un archivo lógico interno. Se definen como un proceso elemental mediante el cual ciertos datos cruzan la frontera del sistema desde afuera hacia adentro.

Entradas externas	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (baja, media o Alta)
Autenticar Usuario	1	10	baja
Gestionar Medios básicos informáticos	1	14	baja

Tabla 17 Entradas externas

Salidas externas

Se definen como un proceso elemental con componentes de entrada y de salida mediante el cual datos simples y datos derivados cruzan la frontera del sistema desde adentro hacia fuera.

Salidas externas	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (baja, media o Alta)
Mostrar entrada y salida de los medios básicos informáticos.	1	1	baja
Obtener datos de los medios básicos informáticos.	1	1	baja

CAPÍTULO IV

Mostrar Medios básicos informáticos por sucursales.	1	1	baja
Mostrar Medios básicos informáticos por tipo de equipos.	1	1	baja
Mostrar Listado de mantenimiento de los medios básicos informáticos.	1	1	baja
Imprimir listado de todos los medios básicos informáticos.	1	1	baja

Tabla 18 Salidas externas

Archivos Lógicos Internos

Constituyen un grupo de datos relacionados lógicamente e identificables por el usuario, que residen enteramente dentro de los límites del sistema y se mantienen a través de entradas externas.

Nombre del fichero Interno	Cantidad de archivos referenciados	Cantidad de elementos de datos	de de	Clasificación (baja, media o Alta)
Control de los movimientos de los medios básicos informáticos.	1	5		baja
Controlar entrada y salida de los medios	1	5		baja

CAPÍTULO IV

básicos informáticos.			
--------------------------	--	--	--

Tabla 19 Archivos Lógicos Internos

4.3 Estimación inicial

Puntos de Función sin ajustar (UFP): se obtiene a través de la suma del aporte de todos los elementos.

Elementos	Baja		Media		Alta		Aportes
	Cant	Valor	Cant	Valor	Cant	Valor	
Entradas externas	2	3	0	4	0	6	6
Salidas externas	6	4	0	5	0	7	24
Consultas externas	0	3	0	4	0	6	0
Archivos Lógicos Internos	2	7	0	10	0	15	14
Archivos de Interfaz Externos	0	5	0	7	0	10	0
Total	UFP =44						

Tabla 20: Puntos de Función desajustados

Una vez que se han obtenido los Puntos de Función sin ajustar del sistema se puede estimar el esfuerzo, para esto se utilizará el método COCOMO II.

4.4 Cálculo de instrucciones fuertes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo

El método COCOMO II consiste básicamente en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar estimados para un proyecto. Estas ecuaciones se encuentran ponderadas por ciertos factores de costo que influyen en el esfuerzo requerido

CAPÍTULO IV

para el desarrollo del *software*. La meta es obtener un número que caracterice completamente al sistema.

Cálculo del esfuerzo nominal.

$$PM_{\text{nominal}} = A * (\text{Size})^E$$

PM nominal: es el esfuerzo nominal requerido en meses-hombre.

A: Es una constante que se utiliza para capturar los efectos multiplicativos en el esfuerzo requerido de acuerdo con el crecimiento del tamaño del *software*. El modelo la calibra con un valor de 2.94.

Size: Es el tamaño estimado del *software*, en Puntos de Función sin ajustar (convertibles a KSLOC). Se calcula el producto de los puntos de función sin ajustar por un factor de conversión que depende del lenguaje a utilizar en el desarrollo del sistema. Se utiliza PHP (factor de conversión = 53 SLOC/UFP).

$$\text{Size} = 53 * 44 = 2332 \text{ SLOC}$$

$$\text{Size} = 2,33 \text{ KSLOC}$$

E: Es una constante denominada *Factor Escalar*. Se calcula ponderando las variables escalares, mediante la ecuación:

$$E = 0.91 + 0.01 * \sum (Wi)$$

Wi = valor de la variable escalar

Donde las *Wi* se muestran en la siguiente tabla:

Variable	Descripción	Valor
PREC	El sistema es muy familiar	2.25
FLEX	Algo de relajación en cuanto a la flexibilidad del desarrollo	2.50

CAPÍTULO IV

RESL	La arquitectura es sólida y los riesgos generalmente se mitigan	<i>3.50</i>
TEAM	La interacción del equipo es altamente cooperativa	<i>2.80</i>
PMAT	La madurez del proceso software es baja	<i>1.80</i>
Total		<i>12.85</i>

Tabla 21 Factor escalar

$$E = 0.91 + 0.01 * 12.85 = 1.167$$

Por tanto:

$$PM \text{ nominal} = A * (\text{Size})^E = 2.94 * (2.3 \text{ KSLOC})^{1.167} = 7,89 \text{ mes-hombre}$$

Cálculo del esfuerzo ajustado.

$$PM \text{ ajustado} = PM \text{ nominal} * \prod (ME_i)$$

Multiplicador	Descripción	Valor
PERS	Se tienen analistas y programadores con alta eficiencia y capacidad de trabajo en equipo.	<i>0.90</i>
RCPX	Las exigencias de confiabilidad, documentación y volumen de datos son moderadas, y la complejidad del producto es baja.	<i>1.05</i>

CAPÍTULO IV

RUSE	No se pretende reutilizar nada	1.07
PDIF	No existen restricciones en cuando al tiempo de CPU o al consumo de memoria, la plataforma es muy estable.	1.00
PREX	Tanto los analistas como los programadores tienen aproximadamente 6 meses de experiencia en la aplicación, la plataforma, el lenguaje y las herramientas utilizadas.	0.95
SCED	Se requiere terminar el proyecto en el tiempo estimado.	1.00
FCIL	Se tienen herramientas CASE simples e infraestructura de comunicaciones básica	0.95
Total		0.96

Tabla 22 Multiplicadores de esfuerzo

$\Pi (MEi) = 0.96.$

$PM \text{ ajustado} = 7.89 * 0.96 = 7.57 \text{ Mes-hombre.}$

Cálculo del tiempo de desarrollo, cantidad de hombres y costo.

Valores calibrados: $A = 2.94$; $B = 0.91$; $C = 3.67$; $D = 0.24$

$F = D + 0.2 * (E - B)$

$F = 0.24 + 0.2 * (1.167 - 0.91) = 0.29$

$TDEV (\text{Tiempo de desarrollo}) = C * (PM \text{ ajustado}) ^F$

$TDEV (\text{Tiempo de desarrollo}) = 3.67 * (10.11) ^{0.29} = 7.17$

$CH (\text{Cantidad de Hombres}) = PM \text{ ajustado} / TDVE = 10.11 / 7.17 \approx 1.4 \approx 1 \text{ persona.}$

CAPÍTULO IV

Como la cantidad real de hombres disponibles para el desarrollo de la aplicación es **2**, al reajustar el tiempo de desarrollo según la cantidad de hombres, resultó un tiempo equivalente a $(7.17/2)=5.05$ meses.

Salario promedio: Para determinar el salario promedio se tuvo en cuenta que los desarrolladores son estudiantes de 5to año, por lo que se toma como salario correspondiente \$100.00

$$\text{Costo} = 2 * 100 * 7.89 = 1578\$$$

Cálculo de:	Valor
Esfuerzo	7.89 Mes-hombre
Tiempo de desarrollo	5.05 Meses
Cantidad de hombres	2 Personas
Salario medio	\$100.00
Costo	\$1578.00

Tabla 23 Resultados

4.5 Beneficios tangibles e intangibles

Tangibles.

Teniendo en cuenta que la aplicación no es un producto desarrollado para la comercialización, no es válido mencionar beneficios económicos. Se puede decir que el costo por desarrollar la aplicación es de \$1578.00 MN (moneda nacional), el cual es perfectamente reparable si en un futuro se comercializará.

Intangibles

Con el uso de la aplicación se podrá reducir considerablemente el tiempo dedicado al desarrollo de las planificaciones, a la vez que se garantiza que no existan errores en las mismas por doble utilización de recursos, algo que pasa con frecuencia si este proceso se desarrolla manualmente como ocurre en la institución. En este lugar solo están automatizadas algunas acciones, esto trae consigo que un grupo de personas se dedique a la captura de datos, teniendo que pasar largas horas en ello, el *software* que se propone evita este tipo de desgaste físico, así como las dificultades que puede crear no hacerlo de manera correcta, al

proveer a los usuarios de un servicio de captura automática que sigue las planificaciones almacenadas en la BD. La tecnología utilizada para el desarrollo del sistema es totalmente libre, por tanto, no hay que incurrir en gastos en el pago de licencias de uso. El sistema es portable por lo que un cambio de plataforma para la implantación del mismo es viable y factible, y no hay que incurrir en muchos cambios. Es una solución a los problemas encontrados en la aplicación que se usa actualmente y propone una interfaz sencilla y fácil de usar.

4.6 Análisis de costo y beneficios

El desarrollo de la aplicación no constituye un gasto considerable pues todas las herramientas que se han empleado en su desarrollo son libres y de código abierto. El sistema está orientado al trabajador y es de fácil aprendizaje.

4.7 Conclusiones

En este capítulo se describió el estudio de factibilidad realizado correspondiente al sistema propuesto, teniendo en cuenta el costo estimado y los beneficios que reportará al ser implantado. El estudio realizado, ha proporcionado valiosos argumentos que permiten llegar a la conclusión de que la solución de *software* es factible.

CONCLUSIONES GENERALES

Conclusiones generales

A partir del análisis realizado en la Unión de Empresas de Recuperación de Materias Primas se determinaron una serie de procesos que se realizan manualmente. En estos procesos están involucrados datos e informaciones de gran importancia con relación a los medios básicos informáticos de cada empresa. Para solucionar estas dificultades se desarrolló una aplicación para automatizar los procesos principales que tienen lugar en cada empresa.

Para el desarrollo del sistema se efectuaron las siguientes actividades:

- ✓ El objetivo trazado en el diseño teórico – metodológico para el desarrollo de la investigación fue cumplido, comprobándose la hipótesis como respuesta del problema a resolver que originó esta investigación, contribuyendo directamente a la gestión de los procesos que ocurren en la Unión de Empresas de Recuperación de Materias Primas.
- ✓ Se hizo un análisis de los sistemas similares existentes tanto nacional como internacionalmente de los cuales se obtuvo conocimiento para saber como marcha en el mundo el uso de las aplicaciones informáticas dedicadas a los sistemas de gestión de los procesos que ocurren en el mundo.
- ✓ Se llevó a cabo el estudio de las principales metodologías, lenguajes, y herramientas que se consideraron factibles para el desarrollo del sistema.
- ✓ Se realizó todo el proceso de desarrollo del *software* siguiendo lo que establece la metodología ágil utilizada que fue XP, lo cual queda plasmado en el siguiente documento.
- ✓ El estudio de factibilidad realizado, proporciono valiosos argumentos que permiten llegar a la conclusión de que la solución de *software*, es factible pues reportará importantes beneficios sin incurrir en mayores gastos.
- ✓ Con la implementación de la aplicación se da cumplimiento al objetivo propuesto ya que se automatizan los procesos teniendo en cuenta las tareas planteadas.
- ✓ Se logra la confiabilidad, integridad y disponibilidad de los datos almacenados.
- ✓ Se despliega la aplicación, obteniendo resultados satisfactorios en la explotación del producto.

RECOMENDACIONES

Recomendaciones

Como resultado del proceso de investigación y realización de la aplicación han surgido ideas que serían recomendables tener en cuenta para un futuro perfeccionamiento del sistema, por lo cual se aconseja continuar con la implementación agregando nuevas funcionalidades y mejorando las existentes en la medida que sea posible y de acuerdo con las necesidades del cliente. Continuar el desarrollo del sistema, en base al incremento de los requerimientos del cliente. Trabajar en base a obtener una conexión segura (HTTPS).

BIBLIOGRAFÍA REFERENCIADA

Bibliografía referenciada

1-Salazar, A. A. (1 de enero de 2000). *Gestión del Conocimiento.com*. Recuperado el 18 de marzo de 2010, de Gestión del Conocimiento.com: <http://www.gestiondelconocimiento.com/documentos2/apavez/gdc.htm>

2-(MSc Sonia Infante González, 2003)

MSc Sonia Infante González, T. C. (2003). *Sistema automatizado de control de los Activos Fijos Tangibles para el Sector de Salud (MEBUS)*. Recuperado el 18 de 05 de 2010, de Sistema automatizado de control de los Activos Fijos Tangibles para el Sector de Salud (MEBUS): http://bvs.sld.cu/revistas/san/vol7_1_03/san53103.htm

3-Téc. Jennifer González Echagarrúa, L. C. (2003). *Sistema automatizado para el control de los inventarios en uso (FONDUS)*. Recuperado el 18 de 05 de 2010, de Sistema automatizado para el control de los inventarios en uso (FONDUS): http://bvs.sld.cu/revistas/san/vol7_1_03/san55103.htm

4-Programación Extrema. (22 de 12 de 2007). Recuperado el 18 de 05 de 2010, de Programación Extrema: <http://www.programacionextrema.org/>

5-2006, M. A. (2010). *Molinux*. Recuperado el 18 de mayo de 2010, de Molinux: <http://www.molinux.info/index.php>

6-Definición de ASP.net. (2008). Recuperado el 18 de mayo de 2010, de Definición de ASP.net: http://help.websiteos.com/websiteos_sp/definici_n_de_asp.net.htm

7-EcuRed. (2010). Recuperado el 5 de junio de 2010, de EcuRed: <http://www.ecured.cu/index.php/JavaScript>

BIBLIOGRAFÍA REFERENCIADA

- 8-Gómez, A. M. (abril de 2010). *eumed.net*. Recuperado el 18 de mayo de 2010, de eumed.net: <http://www.eumed.net/rev/cccss/08/bgsv.htm>
- 9-GSInnova. (s.f.). Recuperado el 18 de mayo de 2010, de <http://www.rational.com.ar/herramientas/rup.html>
- 10-Navarmedia. (s.f.). Recuperado el 5 de junio de 2010, de Navarmedia: <http://navarmedia.com/?s=internet>
- 11-proyectos agiles.org. (2009). Recuperado el 5 de junio de 2010, de proyectos agiles.org: <http://www.proyectosagiles.org/que-es-scrum>
- 12-Ravioli, P. (200). *monografias.com*. Recuperado el 5 de junio de 2010, de monografias.com: <http://www.monografias.com/trabajos7/html/html.shtml>
- 13-Ruesta, C. B. (marzo de 2001). *www.eubca.edu.uy*. Recuperado el 18 de 05 de 2010, de *www.eubca.edu.uy*: http://www.eubca.edu.uy/materiales/planeamiento_de_servicios_bibliotecarios/gestion_del_conocimiento_y_gestion_de_la_informacion.pdf
- 14-S, C. V. (2003). *maestros del web*. Recuperado el 5 de junio de 2010, de maestros del web: <http://www.maestrosdelweb.com/editorial/phpintro/>
- 15-Solán, P. O. (octubre de 2008). *monografias.com*. Recuperado el 18 de 05 de 2010, de monografias.com: <http://www.monografias.com/trabajos15/sistemas-control/sistemas-control.shtml#intro>
- 16-Escribano, G. F. (2002). *eXtreme Programming / Programación Extrema*. Recuperado el 14 de junio de 2010, de *eXtreme Programming / Programación Extrema*: <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf>

BIBLIOGRAFÍA CONSULTADA

Bibliografía consultada

Fabien Potencier, F. Z. 200. Symfony la guía definitiva.

Pressman, R. (2005). Ingeniería del software. Un enfoque práctico.

Ordenadores, C. (s.f.). Cea Ordenadores. Obtenido de <http://www.ceaordenadores.com/>

RodasXXI. (s.f.). RodasXXI. Obtenido de <http://www.rodasxxi.cu/inventario.php>

SISCONT5. (s.f.). SISCONT5. Obtenido de <http://siscont.tm.minbas.cu/Docs/Documentos/Manuales/Inventario%20SISCONT5.pdf>

DesarrolloWeb.com. (s.f.). DesarrolloWeb.com. Obtenido de Manual completo de HTML: <http://www.desarrolloweb.com/manuales/21/>

Miguel Angel Álvarez, R. A. (s.f.). DesarrolloWeb.com. Obtenido de <http://www.desarrolloweb.com/manuales/34/>

PostgreSQL. (s.f.). Obtenido de <http://www.postgresql.c/>

Foundation, A. S. (2010). The Apache Software Foundation. Obtenido de <http://www.apache.org/>

Corporation, M. (2010). Internet Information server. Obtenido de <http://www.iis.net/>

EllisLab, I. (s.f.). Manual de CodeIgniter. Obtenido de http://codeigniter.com/user_guide/

Rubén Álvarez, P. R. (s.f.). Programación en ASP. Obtenido de www.desarrolloweb.com/manuales/8/

BIBLIOGRAFÍA CONSULTADA

Miguel Angel Alvarez, J. M. (s.f.). Qué es cada tecnología. Obtenido de <http://www.desarrolloweb.com/manuales/15/>

Gracia, J. (2005). IngenieroSoftware.com. Obtenido de <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>

Visual-paradigm.com. (s.f.). Obtenido de <http://www.visual-paradigm.com/>

Object Management Group, I. (s.f.). Unified Modeling Language. Obtenido de <http://www.uml.org/>

Málaga, G. d. (s.f.). BIBLIOTECA VIRTUAL de Derecho, Economía y Ciencias Sociales. Obtenido de <http://www.eumed.net/libros/2007c/306/METODOS%20GENERALES%20DE%20LA%20INVESTIGACION%20CIENTIFICA.htm>

Alvarez, S. (s.f.). Manual de iniciación a la programación. Obtenido de <http://www.desarrolloweb.com/manuales/74/>

ZonaOracle.com. (s.f.). Comunidad Oracle en español. Obtenido de <http://www.zonaoracle.com/>

2009. Entorno virtual de aprendizaje. [Online] 2009. [Cited: febrero 21, 2010.] <http://eva.uci.cu/>

GLOSARIO DE TÉRMINOS

Glosario de términos

Una prioridad al realizar una investigación es trazarse el objetivo de que toda persona que lea el trabajo entienda cada termino utilizado. Quizás el mayor problema es que los términos usados son entendidos por cada ser humano de una manera diferente. A continuación se menciona una definición de los términos principales que resulta imprescindible saber su definición para entender el problema expuesto.

- ✓ **CSS:** Las hojas de estilo en cascada son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).
- ✓ **Plugin:** “Pache” para un programa que le añade características nuevas.
- ✓ **Http:** Protocolo de Transferencia de Hipertexto. Modo de comunicación para solicitar páginas web.
- ✓ **Unix:** Sistema operativo portable, multitarea y multiusuario; desarrollado en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McElroy.
- ✓ **UERMP:** Unión de Empresas de Recuperación de Materias Primas.
- ✓ **Arquitectura:** Organización de los diversos elementos constitutivos de un sistema informático.
- ✓ **Arrays:** Arreglo o alineación es un conjunto o agrupación de variables del mismo tipo.
- ✓ **Hardware:** Corresponde a todas las partes físicas y tangibles de una computadora.
- ✓ **Software:** Se refiere al equipamiento lógico o soporte lógico de una computadora digital.
- ✓ **Protocolo:** Es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red.
- ✓ **Procesos del negocio:** Ordenación lógicamente interrelacionada de tareas desarrolladas en tiempo y espacio (con comienzo y fin, con entradas y salidas

GLOSARIO DE TÉRMINOS

definidas) y que se orienta al logro de un objetivo de negocio, generando un output de valor (total o parcial) para el cliente del proceso.

- ✓ **XML:** Por sus siglas en inglés *Extensible Markup Language* (lenguaje de marcas extensible), es un metalenguaje (es un lenguaje que se usa para hablar acerca de otro lenguaje) extensible de etiquetas
- ✓ **HTML** (Lenguaje de Marcado de Hipertexto): Lenguaje de marcado predominante para la construcción de páginas web.