

Universidad de las Ciencias Informáticas

Facultad 1



Título: Sistema para la gestión de los procesos económicos de la Casa Financiera de Unión de Empresas Recuperadoras de Materias Primas.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Autores:

Dariel Osorio Plasencia
Alain Garcia Asencio

Tutores:

Ing. Marianny Hernández Batista
Ing. Hiraldo Manuel Anaya González

Ciudad de La Habana, Junio del 2010
"Año 52 de la Revolución"

“¿Por qué esta magnífica tecnología científica, que ahorra trabajo y nos hace la vida más fácil, nos aporta tan poca felicidad? La respuesta es está, simplemente: porque aún no hemos aprendido a usarla con tino.”

Albert Einstein

Alain

*Dedico este trabajo con todo mi corazón
a mi mamá por nunca rendirse.*

Daniel

A mis padres por enseñarme que el conocimiento es el mayor tesoro que puede tener un ser humano. Por dejarme como legado valores que defenderé a cualquier precio.

A mi tía porque a pesar de la distancia siempre sus consejos y su experiencia han estado cerca de nosotros.

A mis abuelas que tanto les hubiese gustado verme hoy convertido en lo que soy.

AGRADECIMIENTOS

Alain

A mi "Toti" por una vida de sacrificio, por ser mi guía.

A mi "casi esposa" por aguantarme...

A mis primos por entenderme: Yasnel, Greity, Jandy, Yudith, Yoalis y Eduardito.

A mi tíos: Alba, Canda, Ana, Moises, Alina, Yoyi, Aloima y Chuchi.

A mis niños por su cariño: Jonhan, Yendry, Zulima y Misselita

A mis cuñados: Lorenzo y Adriancito.

A mis amigos por el apoyo: Dariel, Javier, Yankiel, Yeny, Daily, Medaimy, Leduar y Yoslandys.

A los que nos han ayudado en este trabajo: Eddy y Alejandro.

A los profes de estos últimos cinco años por sembrar en mi la semilla del saber.

A los integrantes del equipo de beisbol de la Universidad, por dejarme jugar con ustedes.

A mi profe Betty por enseñarme el camino que culmina aquí.

Y en especial a todos los que dijeron que nunca me graduaría, a los que nunca me apoyaron, a los que me cerraron las puertas por demostrarme lo fuerte que soy para lograr lo que me propongo.

Daniel

A mi mamá, por ponernos a mí y a mis hermanos como prioridad número uno en su vida desde que vinimos al mundo. Por apoyarme ciegamente en todo momento.

A mi papá, por poner el amor de padre por encima de todo, por formar mi carácter y por enseñarme a enfrentar la vida y a levantar la frente. Por ser mi primer maestro.

A mi tía, por lo mucho que le hubiera gustado compartir este triunfo junto a nosotros.

A mi hermano, por ser mi amigo, mi consejero y sobre todas las cosas mi ejemplo más cercano de lo que quería y logré ser.

A mi hermana por ser la parte que me falta, mi amiga siempre y por ser tan iguales y a la vez de diferentes.

A mi sobrino por demostrarme que la madurez nunca es suficiente.

A Meda, por ser mi amiga novia y compañera durante estos años y soportarme durante tanto tiempo.

A mi tía Celita por quererme como su niño mayor.

A Alain, mi amigo y compañero de tesis.

A mis maestros y profesores a lo largo de estos años de estudio. En especial a Matilde, Marlén, Isidro, Carlos de la Vega. Por enseñarme a luchar contra la incultura, con la que irremediabilmente nace el ser humano.

A mis amigos de la Lenin por compartir con ellos momentos inolvidables.

Al grupo 1104 que ser parte del es una de las mejores cosas que me paso en esta Universidad. En especial, Javier, Pedro, María.

Declaración de autoría

Dariel Osorio Plasencia y Alain Garcia Asencio declaran que son los únicos autores de la investigación científica: “Sistema para la gestión de los procesos económicos de la Casa Financiera de Unión de Empresas Recuperadoras de Materias Primas”, y otorgamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Dariel Osorio Plasencia

Alain Garcia Asencio

Firma del Autor

Firma del Autor

Ing. Hiraldo Manuel Anaya González

Ing. Marianny Hernández Batista

Firma del Tutor

Firma del Tutor

RESUMEN

La presente investigación científica, titulada: Sistema de gestión de los *procesos económicos*^{*1} para la casa financiera de la Unión de Empresas Recuperadoras de Materias Primas, se inicia al identificar problemas fundamentales existentes en la mencionada institución. Emergiendo desde entonces los principales objetivos de la investigación, que sentaron las bases para crear una aplicación que automatice la gestión de los procesos económicos en la empresa; proveer una base de datos que sustituyera la obsoleta existente, así como conservar los datos registrados por un sistema de más de doce años de explotación; fueron premisas superadas con creces por el alcance del producto final.

El logro de una aplicación que a partir de las nuevas tecnologías y herramientas que se investigaron, realice la emisión periódica de reportes auditables, emita estados momentáneos de subcuentas y del presupuesto general de la empresa, además provea mayor seguridad en el trabajo con la información al establecer un trabajo por roles salvaguardando los datos de la empresa y de los usuarios que la manejan. Otro beneficio de la aplicación es el uso más eficiente de la jornada laboral. A partir de la implantación del sistema, la acumulación física de reportes será mucho menor y el uso de las impresoras será más racional. Trae mejoras notables para los trabajadores de la casa financiera desde el punto de vista organizativo, ya que pueden disponer de los datos con mayor facilidad y en menor espacio de tiempo. Es un ejemplo del uso de las tecnologías para el desarrollo de la humanidad.

Todas las palabras que lo adelante aparezcan en cursiva y tengan un (*) serán conceptualizadas en el glosario de términos.

ÍNDICE

Introducción	1
CAPÍTULO 1: Fundamentación teórica	5
1.1. Conceptos asociados al dominio del problema.....	5
1.1.1. Contabilidad	5
1.1.2. Sistema contable.....	6
1.1.3. Contabilidad de costos	6
1.1.4. Datos primarios	6
1.1.5. Sistema informático:.....	6
1.1.6. Gestión de procesos	7
1.1.7. Gestión de información	7
1.1.8. Herramientas CASE	7
1.2. Funciones de la casa financiera.....	7
1.3. Sistemas contables	8
1.3.1. Sistemas contables utilizados en el mundo.....	8
1.3.2. Sistemas contables utilizados en Cuba.....	9
1.3.3. Sistemas contables utilizados en la UCI.	11
1.4. Fundamentación de la tecnología a utilizar.....	11
1.4.1. Sistemas Gestores de Base de Datos	12
1.4.2. Lenguajes de programación.....	15
1.4.3. Herramientas para la migración de base de datos.....	20
1.5. Metodologías para la Ingeniería de Software	21
1.5.1. Proceso Unificado de Desarrollo de Software	21
1.5.2. Programación extrema.....	21
1.5.3. Scrum.....	22
1.6. Lenguajes y herramientas para el modelado de software	23
1.6.1. Lenguaje UML.....	23
1.6.2. Visual Paradigm para UML	23
1.7. Conclusiones del capítulo	23
Capítulo 2: Características del Sistema	25
2.1. Fase de Exploración-Planificación	25

2.1.1. Historias de usuario.....	25
2.1.2. Personas relacionadas con el sistema	35
2.1.3. Requisitos no funcionales del sistema	35
2.1.4. Estimación de esfuerzo por historias de usuario	38
2.1.5. Plan de iteraciones.....	39
2.1.6. Plan de duración de iteraciones	39
2.1.7. Tareas de ingeniería	41
2.1.8. Plan de entregas	50
2.2. Fase de Diseño	51
2.2.1. Metáfora para el sistema	51
2.2.2. Diseño de la base de datos	51
2.3. Tarjetas CRC	52
2.4. Estimación de esfuerzo por el COCOMO II.....	59
2.5. Conclusiones del capítulo	61
Capítulo 3: Implementación, prueba y validación del sistema propuesto	62
3.1. Técnicas de programación en el lenguaje C#	62
3.1.1. Descripción del algoritmo más complejo en el desarrollo.	63
3.2. Beneficios tangibles e intangibles	64
3.3. Pruebas	64
3.3.1. Pruebas unitarias	65
3.3.2. Pruebas de aceptación	65
3.4. Conclusiones del capítulo	65
Conclusiones	67
Recomendaciones	68
Referencias Bibliográficas	69
Glosario de términos	71

ÍNDICE DE TABLAS

Tabla 1: Historia de Usuario: Ingreso por cheque.....	26
Tabla 2: Historia de Usuario: Egreso por cheque.	26
Tabla 3: Historia de Usuario: Ingreso por traspaso.....	27
Tabla 4: Historia de Usuario: Egreso por traspaso.	27
Tabla 5: Historia de Usuario: Estado de cuenta por subcuenta.....	28
Tabla 6: Historia de Usuario: Clasificador de epígrafe.....	28
Tabla 7: Historia de Usuario: Ejecución de presupuesto.	29
Tabla 8: Historia de Usuario: Entrada de subcuenta.	29
Tabla 9: Historia de Usuario: Funcionario autorizado.	30
Tabla 10: Historia de Usuario: Ajuste y cancelación.....	30
Tabla 11: Historia de Usuario: Chequeo de cuenta y presupuesto.....	31
Tabla 12: Historia de Usuario: Hallar mejor cliente.....	31
Tabla 13: Historia de Usuario: Consolidar flujo de caja.	32
Tabla 14: Historia de Usuario: Cuadre diario-semanal.	32
Tabla 15: Historia de Usuario: Cuadre mensual-anual.	33
Tabla 16: Historia de Usuario: Conformación de reporte.....	33
Tabla 17: Historia de Usuario: Autenticar usuario.....	34
Tabla 18: Personas relacionadas con el sistema.....	35
Tabla 19: Estimación de esfuerzo por historia de Usuario.....	38
Tabla 20: Planificación de duración de iteraciones.....	40
Tabla 21: Tarea de Ingeniería: Establecer Período.	41
Tabla 22: Tarea de Ingeniería: Realizar Monto.....	41
Tabla 23: Tarea de Ingeniería: Revisar Chequeo.	42

ÍNDICE DE TABLAS

Tabla 24: Tarea de Ingeniería: Seleccionar Egresos.....	42
Tabla 25: Tarea de Ingeniería: Chequera.....	43
Tabla 26: Tarea de Ingeniería: Revisar Chequera.....	43
Tabla 27: Tarea de Ingeniería: Traspasos.....	44
Tabla 28: Tarea de Ingeniería: Llevar Estado de Cuenta.....	44
Tabla 29: Tarea de Ingeniería: Crear Epígrafe.....	45
Tabla 30: Tarea de Ingeniería: Diseñar Reportes por Epígrafes.....	45
Tabla 31: Tarea de Ingeniería: Verificar Presupuesto.....	46
Tabla 32: Tarea de Ingeniería: Asignar Presupuesto.....	46
Tabla 33: Tarea de Ingeniería: Entrada por subcuentas.....	47
Tabla 34: Tarea de Ingeniería: Funcionarios.....	47
Tabla 35: Tarea de Ingeniería: Ajustes Cancelaciones.....	48
Tabla 36: Tarea de Ingeniería: Cuentas y Presupuesto.....	48
Tabla 37: Tarea de Ingeniería: Caja Semanal.....	49
Tabla 38: Tarea de Ingeniería: Caja Anual.....	49
Tabla 39: Tarea de Ingeniería: Consolidar.....	50
Tabla 40: Planificación de entrega.....	50
Tabla 41: Ajustes.....	52
Tabla 42: Auditor.....	53
Tabla 43: Cambiar Contraseña.....	53
Tabla 44: Chequeo de Presupuesto.....	53
Tabla 45: Clasificador.....	54
Tabla 46: Usuario.....	54
Tabla 47: Egresos.....	54

ÍNDICE DE TABLAS

Tabla 48: Ejecución.	55
Tabla 49: Entrada de Sub Cuentas.	55
Tabla 50: Estado.	55
Tabla 51: Informático.	55
Tabla 52: Ingresos.	56
Tabla 53: Mostrar Cancelaciones.	56
Tabla 54: Mostrar Estado.	56
Tabla 55: Operador.	57
Tabla 56: Reporte Cancelación.	57
Tabla 57: Solicitud Egreso.	58
Tabla 58: Solicitud Ingreso.	58
Tabla 59: Solicitud Traspaso.	58
Tabla 60: Sub-Cuentas.	59
Tabla 61: Traspasos.	59
Tabla 62: Valores de los coeficientes para los modos del modelo de estimación y costos.	60

Introducción

En la sociedad actual, donde casi todo es desechable, reciclar se ha hecho un negocio. El éxito del reciclaje está dado por factores como: la reducción de basura que contribuye a la limpieza del ambiente y por consiguiente al aumento de la calidad de vida y el reuso de desechos sólidos que se pueden emplear como materias primas, sustituyendo importaciones y generando ganancias a las empresas. (1)

El reciclaje nace con el hombre; siempre el ser humano ha necesitado reutilizar algo que ha creado. En Cuba se enmarca el reciclaje en dos etapas, antes y después del Triunfo de la Revolución. Antes de 1959 en Cuba se reciclaba de forma espontánea y no existía ninguna institución que se encargara de ello. Después del triunfo, el país se trazó un conjunto de estrategias, entre las que se encontraba la concebida y proyectada por el Che, siendo Ministro de Industrias, de crear una empresa que se encargara de la *materia prima** que se generaba en La Isla y así surge mediante la Resolución 21-1272 del 7 de noviembre de 1961, la Empresa Consolidada de Recuperación de Materias Primas que pertenecía al Ministerio de Industrias. (1)

Con el transcurso de los años esta empresa transitó por más de un organismo, poco después de pertenecer al Ministerio de Industrias 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 1977 se trasladó al Comité Estatal de Abastecimiento Técnico Material (CEATM); en ese organismo se convirtió en Unión de Empresas de Recuperación de Materias Primas (UERMP) en 1981, a partir de ese momento se fortalece la actividad económica y tecnológica. En noviembre de 1994 la organización del reciclaje fue trasladada del CEATM al Ministerio de Economía y Planificación. La Unión de Empresas de Recuperación de Materias Primas fue trasladada en marzo de 1997 al entonces Ministerio de la Industria Sideromecánica y la Electrónica. En ese momento la integraban 14 empresas recuperadoras territoriales, empresas de servicios y especializadas, así como empresas importadoras y exportadoras. En la actualidad la UERMP permanece en la rama de la siderurgia, hoy con el nombre de Ministerio de la Industria Sideromecánica y del Reciclaje. Cuenta con 26 empresas

clasificadas en: 15 empresas recuperadoras, una en cada provincia y en el municipio especial Isla de la Juventud; 9 empresas especializadas y 2 exportadoras-importadoras.
(1)

Como parte del perfeccionamiento empresarial que lleva a cabo el país y la vinculación de las Tecnologías de la Informática y las Comunicaciones (TIC) para mejorar el trabajo de las empresas, se toma la idea de informatizar esta entidad.

Teniendo en cuenta los problemas del actual sistema contable de la Casa Financiera (CF) de la UERMP se pretende mejorar el funcionamiento del mismo, el cual no permite una correcta gestión de los procesos económicos que tienen lugar en dicho departamento; presentando dificultades en la asignación de los permisos correspondientes a cada uno de los usuarios. La base de datos no permite hacer cambios en las tablas; mantiene campos que ya no se usan; no se pueden dejar campos nulos, lo que obliga a las técnicas, a inventar valores en muchos casos. Todo lo anteriormente expuesto dificulta el *control físico y auditable** de la información con que trabaja dicho sistema.

Por las insuficiencias planteadas se formula el siguiente **Problema a resolver**: ¿Cómo mejorar la gestión de los procesos económicos de la Casa Financiera de la Unión de Empresas Recuperadoras de Materias Primas?

El **Objeto de estudio** está definido por: Los procesos de gestión de información que se desarrollan en la Unión de Empresas Recuperadoras de Materias Primas.

El cual se enmarca en el **Campo de acción**: El proceso de gestión de información que se desarrolla en la Casa Financiera de la Unión de Empresas Recuperadoras de Materias Primas.

Se plantea como **Objetivo general**: Desarrollar un sistema informático que automatice la gestión económica de la Casa Financiera de la Unión de Empresas Recuperadoras de Materias Primas.

Y como **Objetivos específicos**:

- ✓ Desarrollar una base de datos lo suficientemente robusta para llevar un control auditable de las operaciones.
- ✓ Mantener los registros archivados hasta la fecha.

INTRODUCCIÓN

- ✓ Crear un sistema que permita automatizar las funciones de la Casa Financiera.

Se define como **Hipótesis:** La implementación de un sistema de gestión de procesos en la Casa Financiera de la Unión de Empresas Recuperadoras de Materias Primas contribuirá a mejorar la gestión de los procesos económicos.

Teniendo como **Variable independiente:**

- ✓ Sistema de gestión de procesos en la Casa Financiera de la Unión de Empresas Recuperadoras de Materias Primas.

Y como **Variable dependiente:**

- ✓ Mejorar la gestión de los procesos económicos.

Se definen como **Tareas a cumplir:**

- ✓ Estudiar el funcionamiento de la entidad.
- ✓ Seleccionar las herramientas a utilizar.
- ✓ Estudiar los requisitos.

Para el diseño de la investigación se utilizaron métodos científicos como:

Métodos empíricos:

- ✓ Entrevista, se realiza a los clientes, técnicos y especialistas de la Casa Financiera de la Unión de Empresas Recuperadoras de Materias Primas, para obtener información relacionada con la aplicación que se utiliza actualmente.
- ✓ Observación, se observa el flujo de actividades diarias de los clientes con el propósito de un mejor entendimiento de los procesos del negocio.

Métodos teóricos:

- ✓ 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 sintetizarlo, esto no es más que la unión de los elementos para formar un todo. Se utiliza para el análisis de las características de las tecnologías existentes, facilitando la selección de las que constituyen la propuesta de solución.
- ✓ Histórico-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 de Movimientos de Productos 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.

Método Matemático

- ✓ Medición, se utiliza para establecer comparaciones entre las aplicaciones anteriores con respecto a la presente, en la realización del cálculo de eficiencia y rapidez.

Distribución de capítulos

- ✓ **Capítulo 1: Fundamentación teórica**
En este capítulo se hace un análisis de las tendencias tecnológicas actuales, para realizar un sistema contable como el que se propone.
- ✓ **Capítulo 2: Características del Sistema**
En este capítulo se aborda la propuesta de solución elaborada por el equipo de trabajo, así como las primeras fases de la metodología Programación Extrema.
- ✓ **Capítulo 3: Implementación, prueba y validación del sistema propuesto**
En este capítulo se explica cómo se implementó la solución, también se muestran los métodos de pruebas que garantizan la calidad del sistema.

CAPÍTULO 1: Fundamentación teórica

En el presente capítulo se brinda una breve descripción de los principales conceptos asociados al proceso económico, proporcionando una visión general de los aspectos relacionados con dicho proceso, el cual se realiza en la CF de la UERMP. Se desarrolla un análisis de las tendencias tecnológicas actuales para realizar un sistema contable como el que se propone. Se caracterizan algunos de los sistemas que se utilizan tanto en el mundo como en Cuba y en la Universidad de las Ciencias Informáticas (UCI).

1.1. Conceptos asociados al dominio del problema

A continuación se muestra una serie de conceptos que están asociados al dominio del problema:

1.1.1. Contabilidad

Es una herramienta empresarial que permite el registro y control sistemático de todas las operaciones que se realizan en la empresa. (2)

"La contabilidad es el arte de registrar, clasificar y resumir en forma significativa y en términos de dinero, las operaciones y los hechos que son cuando menos de carácter financiero, así como el de interpretar sus resultados". (2)

"La contabilidad es el sistema que mide las actividades del negocio, procesa esa información convirtiéndola en informes y comunica estos hallazgos a los encargados de tomar las decisiones". (2)

"La contabilidad es el arte de interpretar, medir y describir la actividad económica". (2)

"La contabilidad es el lenguaje que utilizan los empresarios para poder medir y presentar los resultados obtenidos en el ejercicio económico, la situación financiera de las empresas, los cambios en la posición financiera y/o en el flujo de efectivo". (2)

"La contabilidad tiene diversas funciones, pero su principal objetivo es suministrar, cuando sea requerida o en fechas determinadas, información razonada, en base a registros técnicos, de las operaciones realizadas por un ente público o privado". (2)

Por lo que se puede concluir que la contabilidad es una técnica que se ocupa de registrar, clasificar y resumir las operaciones de un negocio con el objetivo de mostrar sus

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

resultados, para poder orientarse sobre el curso que siguen los negocios mediante datos contables; permitiendo así conocer la estabilidad, la solvencia de la compañía y la capacidad financiera de la empresa.

1.1.2. Sistema contable

El sistema de contabilidad es una estructura organizada mediante la cual se recogen las informaciones de una empresa como resultado de sus operaciones, valiéndose de recursos como formularios, reportes, libros entre otros que presentados a la gerencia, le permitirán a la misma tomar decisiones financieras. Un sistema de contabilidad no es más que normas, pautas, procedimientos entre otros para controlar las operaciones y suministrar información financiera de una empresa, por medio de la organización, clasificación y cuantificación de las informaciones administrativas y financieras que se suministren. (3)

1.1.3. Contabilidad de costos

Es un sistema de información que clasifica, asigna, acumula y controla los costos de actividades, procesos y productos que facilitan la toma de decisiones, la planeación y el control administrativo. (4)

En otras palabras, la contabilidad de costos es la encargada del cálculo de los costos de la producción y permite determinar si hubo ganancias o pérdidas.

1.1.4. Datos primarios

Los datos primarios son aquellos con los que se cuenta para comenzar un proyecto. Son los datos mínimos necesarios para realizar un análisis determinado.

1.1.5. Sistema informático:

Es la unión de diversos elementos, especialmente el hardware, el software y un soporte humano. El hardware incluye al menos una unidad central de procesamiento. El software incluye al sistema operativo y aplicaciones, siendo especialmente importante los sistemas de gestión de bases de datos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1.6. Gestión de procesos

Determina qué procesos necesitan ser mejorados o rediseñados, establece prioridades y provee de un contexto para iniciar y mantener planes de mejora, que permitan alcanzar objetivos establecidos

1.1.7. Gestión de información

Es todo lo relacionado con la obtención de la información adecuada, en la forma correcta, para la persona indicada, al costo adecuado, en el tiempo oportuno, en el lugar apropiado, para tomar la acción correcta, lo que constituye un proceso significativo en el desarrollo de cualquier organización. (18)

1.1.8. Herramientas CASE

Es un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

1.2. Funciones de la casa financiera

- ✓ Operar las cuentas bancarias de moneda libremente convertible (MLC), de la UERMP y su conciliación.
- ✓ Elaborar y emitir los estados de las sub-cuentas de las empresas.
- ✓ Emitir los cheques para gastos de la Oficina Central.
- ✓ Gestionar créditos a través de bancos o instituciones financieras no bancarias; tanto para capital de trabajo como para inversiones.
- ✓ Realizar los pagos por concepto de amortizaciones de créditos u otras deudas pactadas por la UERMP.
- ✓ Elaborar y firmar con las empresas, los convenios de pago que se originen por asignaciones de créditos u otras deudas. Controlar el pago y realizar los descuentos de la sub-cuenta correspondiente.
- ✓ Confeccionar los flujos de caja semanales, para el análisis en el *Consejo Financiero**.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- ✓ Proponer y realizar los traspasos de fondos a las cuentas de gastos de las empresas; a partir de lo acordado en el Consejo Financiero.
- ✓ Confeccionar y emitir mensualmente, el monto de pagos por obligaciones financieras de las empresas y controlar su pago.
- ✓ Emitir mensualmente el flujo de pagos de la UERMP.
- ✓ Aceptar y tramitar *letras de cambio**, tanto para pagos como para cobros.
- ✓ Cálculo, emisión y liquidación de dietas para viajes al extranjero.
- ✓ Elaborar y proponer variantes de operación financiera.
- ✓ Elaborar los documentos primarios para la contabilidad del movimiento de ingresos y gastos en MLC.
- ✓ Velar por el cumplimiento de la Resolución 92/2004 del Banco Central de Cuba (BCC).
- ✓ Mantener actualizado el registro de *cheques en tránsito** diariamente. Verificando que todas las operaciones de cobros estén registradas por nuestros controles.

1.3. Sistemas contables

A continuación se hace un estudio de los sistemas similares existentes a nivel internacional y nacional. Con el propósito de evaluar sus características fundamentales:

1.3.1. Sistemas contables utilizados en el mundo

Sería imposible determinar cuántos sistemas contables son utilizados en el mundo diariamente, desde las pequeñas y medianas empresas hasta las gigantescas abogan por un software para llevar a cabo sus finanzas de forma eficiente. A continuación se presentan algunos sistemas conocidos en el mundo utilizados y probados por muchos usuarios que dan fe de sus beneficios.

1.3.1.1. AgroWin - Sistema de gestión total para el Agro

Es un sistema especialmente diseñado para proveer información al agricultor sobre su finca y sus recursos. Una de sus principales características es que utiliza un esquema de registro de información muy sencillo, de tal forma que se convierta en un instrumento fácil de utilizar y ágil en el seguimiento y toma de decisiones basadas en la información

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

oportuna y actualizada de la finca. El registro sencillo de la información de la finca facilita que muchos agricultores con escasos conocimientos técnicos en áreas como contabilidad, inventarios, presupuestos y estadísticas, queden habilitados para manejar el sistema, debido a que es el sistema que se encarga de interpretar la información base que el usuario le suministra y automáticamente transformarla en información técnica y correctamente clasificada y codificada a la luz de las técnicas financieras y estadísticas.

(5)

Este sistema a pesar de ser un sistema de gestión contable, no es factible pues solo abarca el área agrícola y no está generalizado a otras empresas.

1.3.1.2. ContaPyme

Es un sistema totalmente integrado. Es un sencillo, amigable y completo sistema de contabilidad y gestión. Desarrollado con tecnología de punta, versátil y adaptable a cualquier empresa. Permite ejercer un control total de la información y el manejo de múltiples áreas de trabajo y múltiples empresas. Es el único con opciones especiales para el cálculo de costos de producción. (6)

A pesar de las grandes ventajas que presenta este sistema, es muy costoso por lo que el país no lo puede adquirir.

1.3.2. Sistemas contables utilizados en Cuba

En Cuba también se han desarrollado sistemas contables, ejemplo de ellos son el SABIC, utilizado por el BCC y el Versat-Sarasola, producto más general usado en todo el país.

1.3.2.1. SABIC

El Sistema automatizado para la Banca Internacional de Comercio (SABIC), es un sistema diseñado y desarrollado por la Dirección de Sistemas Automatizados del BCC para satisfacer las necesidades de procesamiento de datos de bancos e instituciones no bancarias, utilizando los medios técnicos de computación disponibles en el mercado. Este sistema ha sido adaptado a los requerimientos de las operaciones propias del BCC y ha sido desarrollado para que los empleados que hagan uso de él puedan tramitar sus

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

operaciones y realizar sus consultas sin necesidad de acudir a los archivos ni a la actividad manual. De esta forma se aumenta la seguridad, la eficiencia del trabajo y la productividad de los trabajadores. Entre sus principales características, está la contabilización en tiempo real (que permite mantener actualizados los ficheros contables); y contabilización *multimoneda**. Además, las operaciones contables se pueden realizar a través de transacciones tipificadas que generan los asientos contables de forma automática. (7)

Este sistema no resuelve el problema planteado en este trabajo, pues sólo está adaptado a las operaciones del BCC. Da soluciones muy certeras a algunas de las dificultades identificadas por el cliente, pero algunas de las problemáticas propias de las UERMP como conocer el estado de las sub-cuentas e imprimir reportes en tiempos específicos, son algunas de las tareas con las que este sistema no cumple.

1.3.2.2. Versat- Sarasola

Este es un software creado por el villaclareño, licenciado en Economía, Miguel Cabrera González. Es un sistema de contabilidad confiable que ofrece mayor organización, control y disciplina en cada gestión, posibilita enviar información eficaz de forma inmediata y desde lugares apartados.

Este proyecto es un sistema integrado, constituido por 12 módulos que incluyen: configuración y seguridad, contabilidad general y de gastos, costos y procesos, análisis económico empresarial y control de activos fijos, además de intervenir finanzas y cajas, planificación y presupuestos, control de inventarios, de productos terminados, pago de salario, paquete de gestión, contratación y facturación. Este producto le ha traído grandes beneficios al país, pues se ahorró un millón 186 mil dólares que costaban las licencias al evitarse la importación del sistema foráneo, más labor de consultaría, atenciones y otros gastos, además de que actualmente lo utilizan alrededor de 200 entidades de varias provincias y siguen aumentando los clientes. (8)

Este sistema no es aplicable a la CF de la UERMP por las especificidades de la institución. Al igual que el SABIC, a muchas de las funcionalidades especificadas por el cliente en el proceso del negocio, no se les da una solución.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.3.3. Sistemas contables utilizados en la UCI.

La UCI no está exenta de las bondades que traen consigo la implantación y explotación de los sistemas contables en los departamentos encargados de la gestión de los procesos económicos.

1.3.3.1. Assets NS

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, útiles y herramientas de su entidad. (9)

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. (9)

Asimismo, permite la obtención de los estados financieros, separado por parámetros, según el contenido y formato que se desee, realiza conciliaciones bancarias y consulta los saldos de sus cuentas, mayor y sub-mayor de las cuentas contables, prorrateo de gastos indirectos, entre otros. Cuenta con una amplia gama de análisis y consultas, plan del presupuesto y su ejecución, así como otros elementos que le permitirán conocer exactamente la situación actual de su empresa. (9)

Este producto es muy abarcador, pudiera resolver algunas problemáticas empresariales en la UERMP. No constituye una solución para la CF de la UERMP, porque es un producto muy abarcador que no está acorde con especificidades de este departamento que necesitan ser resueltas.

1.4. Fundamentación de la tecnología a utilizar

Basándose en el estudio realizado sobre las tendencias actuales referentes a las tecnologías informáticas, se procede a la selección de las que conformarán la propuesta de solución.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.4.1. Sistemas Gestores de Base de Datos

Los sistemas de gestión de base de datos son un tipo de software muy específico, dedicados a servir de interfaz entre la base de datos, el usuario y las aplicaciones utilizadas. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos. Entre los más utilizados están: Microsoft SQL Server, Oracle, PostgreSQL y MySQL.

1.4.1.1. Postgre SQL

Sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia Distribución de Software *Berkeley* (BSD), por sus siglas en inglés. Esta licencia pertenece al grupo de licencias de software libre y tiene menos restricciones que la Licencia Pública General (GPL), por sus siglas en inglés; estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre. Bajo esta licencia, el autor mantiene la protección de derecho de copia o de autor únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación. Puede argumentarse que esta licencia asegura “verdadero” software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre. (10)

PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el Grupo de Desarrollo Global de PostgreSQL (PGDG), por sus siglas en inglés. (10)

Ventajas de PostgreSQL:

Instalación ilimitada:

Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran esto, la principal

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

fuelle de incumplimiento de la licencia. Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software. (10)

Mejor soporte que los proveedores comerciales:

Además de las ofertas de soporte, existe una importante comunidad de profesionales y entusiastas de PostgreSQL de los que su compañía puede obtener beneficios y contribuir. (10)

Ejemplos de proyectos internacionales que hacen uso de PostgreSQL:

- ✓ Agencias gubernamentales de Estados Unidos como *City of Garden Grove* y el Ejército de Estados Unidos.
- ✓ Skype
- ✓ Yahoo
- ✓ Fujitsu
- ✓ Red Hat
- ✓ Sun Microsystems (11)

1.4.1.2. Oracle

Es un sistema de administración de bases de datos fabricado por la Corporación Oracle. Se considera como uno de los sistemas de bases de datos más completos, destacando su soporte de transacciones, estabilidad y escalabilidad. Además, es multiplataforma. Su mayor defecto es su enorme precio, que es de varios miles de euros. (11)

1.4.1.3. MySQL

MySQL es un sistema de gestión de base de datos relacional, *multihilo** y *multiusuario**. El servidor de base de datos MySQL es el gestor de datos de código fuente abierto más usado del mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación detallada para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. Permite el acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones. Garantiza la seguridad, en forma de

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

permisos y privilegios, determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas. Esto permite compartir datos sin que peligre la integridad de la base de datos o protegiendo determinados contenidos. Es un lenguaje muy potente para consultas a las bases de datos, usa un motor que ahorra una enorme cantidad de trabajo. (12)

1.4.1.4. Microsoft SQL Server 2005

El Servidor Microsoft para lenguaje estructurado de consultas (*Microsoft SQL Server*), por su nombre en inglés, es una plataforma de base de datos y análisis de datos que se utiliza en el procesamiento de transacciones en línea a gran escala, el almacenamiento de datos y las aplicaciones de comercio electrónico. Ofrece la tecnología y las funciones con las que pueden contar las organizaciones. Con avances significativos en áreas clave de la administración de datos empresariales, la productividad de los desarrolladores y la inteligencia empresarial, las ventajas de este gestor son considerables. Aprovechamiento de los activos de datos. Aumento de la productividad. Reducción de la complejidad de la tecnología de la información. Disminución del costo total de propiedad. Microsoft SQL Server, al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de *Microsoft*. (13)

1.4.1.5. Selección del sistema gestor de base de datos

Luego de un estudio de las tendencias mundiales sobre los gestores de base de datos y un análisis detallado de las características de los más usados, se procede a la selección del que más se ajusta a las condiciones del cliente. En la empresa existen aplicaciones desarrolladas con el gestor SQL Server 2005, y la solución debería también integrarse a estas, con el propósito de centralizar la información de todos los departamentos en un solo gestor. Es importante resaltar que el cliente influye en la decisión de la herramienta, debido a que tiene experiencia en el trabajo con la misma. Por lo que se decide desarrollar la aplicación sobre el gestor SQL Server 2005.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.4.2. Lenguajes de programación

Un lenguaje de programación es una construcción mental del ser humano para expresar programas. Está constituido por un grupo de reglas gramaticales, un grupo de símbolos utilizables, un grupo de términos *monosémicos** y una regla principal que resume las demás. Para que esta construcción mental sea operable en un computador debe existir otro programa que controle la validez o no de lo escrito. A este se le llama traductor. (14)

Los lenguajes pueden ser de alto o bajo nivel. En los de bajo nivel las instrucciones son simples y cercanas al funcionamiento de la máquina, como por ejemplo el código máquina y el ensamblador. En los lenguajes de alto nivel hay un alto grado de abstracción y el lenguaje es más próximo a los humanos, como por ejemplo Lexico, PASCAL, Cobol o Java. (14)

1.4.2.1. Lenguaje de programación C#

En la plataforma .NET es posible programar en casi todos los lenguaje, el C# es el lenguaje de propósito general diseñado por Microsoft para ser utilizado en ella. Por lo que se puede inferir en que si es el lenguaje nativo creado para esa plataforma, es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros. (15)

El código escrito en C# es autocontenido*, lo que significa que no necesita de ficheros adicionales a la propia fuente tales como ficheros de cabecera. (15)

El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile, lo que facilita la portabilidad del código.

Modernidad: C# incorpora en el propio lenguaje elementos que a través de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico decimal que permita realizar operaciones de alta precisión con reales de 128 bits (muy útil en el mundo financiero), la inclusión de una instrucción *foreach* que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, la inclusión de un tipo básico *string* para representar cadenas o la distinción de un tipo *bool* específico para representar valores lógicos. (15)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Orientación a objetos: Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos, aunque eso es más bien una característica del CTS que de C#. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código. (15)

Orientación a componentes: La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros) (15)

Seguridad de tipos: C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que evita que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura. Para ello se toman medidas del tipo:

Eficiente: En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. Para ello basta marcar regiones de código como inseguras y podrán usarse en ellas punteros de forma similar a cómo se hace en C++, lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad de procesamiento muy grande. (15)

1.4.2.2. Lenguaje de programación C++

Aunque en un principio C++ se plantea como una mejora de C, en la actualidad es un lenguaje independiente, aunque se conserva la compatibilidad con C. No obstante hay código que compila correctamente en C y no lo hace en C++. (16)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Las características de C++ como lenguaje orientado a objetos son:

- ✓ Abstracción
- ✓ Variables de instancia
- ✓ Métodos de instancia
- ✓ Variables de clase
- ✓ Métodos de clase

Encapsulación

- ✓ De variables: privada, pública y protegida
- ✓ De métodos: privados, públicos y protegidos

Herencia

- ✓ Sencilla y múltiple
- ✓ Unidades genéricas
- ✓ Polimorfismo

Además de los siguientes puntos, algunos heredados del lenguaje C:

- ✓ Es un lenguaje fuertemente tipado
- ✓ Soporta multitarea mediante clases
- ✓ Permite modularidad (16).

1.4.2.3. Lenguaje de programación Java

Las características principales que ofrece Java respecto a cualquier otro lenguaje de programación, son:

Es simple:

Ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de estos. C++ es un lenguaje que adolece de seguridad, pero C y C++ son lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje. (17)

Elimina muchas de las características de otros lenguajes como C++, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles. No es necesario preocuparse de liberar memoria, el reciclado se encarga de ello y como es un

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

hilo de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la memoria. (17)

Reduce en un 50% los errores más comunes de programación con lenguajes como C y C++ al eliminar muchas de las características de estos, entre las que se destacan:

- ✓ aritmética de punteros
- ✓ no existen referencias
- ✓ registros
- ✓ definición de tipos
- ✓ macros
- ✓ necesidad de liberar memoria (17)

Es Orientado a Objetos:

Implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, como en C++, clases y sus copias, instancia. Estas instancias, como en C++, necesitan ser construidas y destruidas en espacios de memoria.

Incorpora funcionalidades inexistentes en C++ como por ejemplo, la resolución dinámica de métodos. Esta característica deriva del lenguaje C, propietario del sistema operativo *Next*. En C++ se suele trabajar con librerías dinámicas que obligan a recompilar la aplicación cuando se retocan las funciones que se encuentran en su interior. Este inconveniente es resuelto por Java mediante una interfaz específica llamada Identificación de tipo de tiempo de ejecución (RTTI), por su nombre en inglés, que define la interacción entre objetos excluyendo variables de instancias o implementación de métodos. Las clases en Java tienen una representación en el tiempo de ejecución que permite a los programadores interrogar por el tipo de clase y enlazar dinámicamente la clase con el resultado de la búsqueda. (17)

Se ha construido con extensas capacidades de interconexión *TCP/IP**. Existen librerías de rutinas para acceder e interactuar con protocolos como *http** y *ftp**. Esto permite a los

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales. (17)

Realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria. También implementa los arreglos auténticos, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas. Estas características reducen drásticamente el tiempo de desarrollo de aplicaciones en Java. (17)

La seguridad en Java tiene dos facetas. En el lenguaje, características como los punteros o la prueba implícita que hacen los compiladores de C y C++ se eliminan para prevenir el acceso ilegal a la memoria. Cuando se usa Java para crear un navegador, se combinan las características del lenguaje con protecciones de sentido común aplicadas al propio navegador. (17)

El código Java pasa muchas pruebas antes de ejecutarse en una máquina. El código se pasa a través de un verificador “por códigos” que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal - código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo o clase de un objeto. (17)

Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Los enteros son siempre enteros y además, enteros de 32 bits en complemento a 2. Además, Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en entornos Unix, PC o Mac. (17)

Java también simplifica el uso de protocolos nuevos o actualizados. Si su sistema ejecuta una aplicación Java sobre la red y encuentra una pieza de la aplicación que no sabe manejar, tal como se ha explicado en párrafos anteriores, Java es capaz de traer automáticamente cualquiera de esas piezas que el sistema necesita para funcionar. (17)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Java, para evitar que los módulos de códigos en los objetos o nuevas clases, no se deben traer de la red cada vez que se necesiten, implementa las opciones de persistencia, para que no se eliminen cuando se limpie la caché de la máquina. (17)

1.4.2.4. Selección del lenguaje de programación

Se decide trabajar con el lenguaje de programación C# utilizando el *Visual Studio* en su versión correspondiente al año 2005, este ofrece muchas facilidades para el trabajo con el gestor de base de datos seleccionado. Para escoger este lenguaje se tuvieron en cuenta las limitadas características de hardware de las computadoras de la CF, valorando así que aplicaciones creadas en otros lenguajes de programación exigían de mayores requerimientos de hardware.

1.4.3. Herramientas para la migración de base de datos

En el trabajo con base de datos es importante hacer migraciones de una base de datos concebida inicialmente en un gestor para otro. Existen algunas herramientas que hacen que esta permuta de datos entre un gestor y otro no sea del todo compleja y hace que el trabajo sea más rápido.

1.4.3.1. SSMA 2005 para Access

El Asistente de Migración para SQL Server (SSMA), por sus siglas en inglés, es una herramienta muy amigable y fácil de usar que tiene la potencialidad de migrar para SQL Server 2005 los datos que están sobre base de datos en otros gestores. Particularmente la versión hecha para migrar datos de Access fue de gran utilidad para el equipo de trabajo pues al cliente le es necesario que los datos almacenados hasta entonces sean conservados y utilizados en las aplicaciones futuras. De esta forma la herramienta evita que el acumulado histórico de información se pierda. Esto permite la utilización o interacción con los mismos.

Es importante destacar que con esta herramienta se pudo también conocer la estructura real de la antigua base de datos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.5. Metodologías para la Ingeniería de Software

Las metodologías se desarrollan con el objetivo de dar solución a los problemas existentes en la producción de software, que cada vez son más complejos. Estas engloban procedimientos, técnicas, documentación y herramientas que se utilizan en la creación de un producto de software. Durante los últimos años se han desarrollado dos corrientes en lo referente a las metodologías de desarrollo de software, las llamadas “pesadas” y las llamadas “ligeras” o “ágiles”. Las primeras se basan en la idea de conseguir el objetivo común por medio de orden y documentación, mientras que las segundas tratan de lograrlo por medio de la comunicación directa e inmediata entre las personas que intervienen en el proceso. A continuación se analizan dos de las más conocidas.

1.5.1. Proceso Unificado de Desarrollo de Software

El Proceso Unificado de Desarrollo de Software (RUP), por sus siglas en inglés, es una metodología muy usada en el mundo. Está pensada para adaptarse a cualquier proyecto. Se basa en la orientación a objetos, el desarrollo iterativo y el modelado visual UML para describir un sistema, lo cual permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos y cambios. Posibilita la distribución del trabajo en diversos frentes de forma simultánea.

RUP se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición; en cada una de las cuales se llevan a cabo una o varias iteraciones, cada iteración resulta un incremento de la precedente, de ahí su carácter iterativo e incremental.

Esta metodología capacita a las organizaciones de muchas maneras, una de las más significativas es que proporciona la forma en la que el equipo de proyecto puede trabajar de manera conjunta con los clientes y demás implicados. Lo que favorece una mayor organización y entendimiento de lo que realmente el cliente necesita y una excelente proyección del proyecto.

1.5.2. Programación extrema

Mientras que RUP intenta reducir la complejidad del software por medio de estructura y la preparación de las tareas pendientes en función de los objetivos de la fase y actividad

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

actual, Programación Extrema (XP), por sus siglas en inglés, como toda metodología ágil, lo intenta por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción.

XP supone la disposición en todo momento, por parte del equipo de trabajo, de un representante competente del cliente, que debe estar en condiciones de dar una respuesta rápida y correcta, a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones.

Esta metodología apuesta por iteraciones cortas que generan software que el cliente puede ver. La codificación del software se realiza siempre en parejas (dos programadores, un ordenador), las cuales no son fijas sino que rotan a lo largo del proyecto, y el código que escriben no les pertenece sólo a ellos sino al equipo completo. Se programa sólo la funcionalidad requerida para la entrega en curso, se trabaja en función de las necesidades del momento, por lo que no se le da importancia al análisis como fase independiente.

1.5.3. Scrum

Es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nuevas funcionalidades. Las iteraciones en general tienen una duración entre 2 y 4 semanas. Scrum se utiliza como marco para otras prácticas de ingeniería de software como RUP o XP.

Scrum se focaliza en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. Está diseñado especialmente para adaptarse a los cambios en los requerimientos, por ejemplo en un mercado de alta competitividad. Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares. De esta forma se puede adaptar en tiempo real el producto que se está construyendo, a las necesidades del cliente. Se busca entregar un software que realmente resuelva las necesidades, aumentando la satisfacción del cliente.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.6. Lenguajes y herramientas para el modelado de software

1.6.1. Lenguaje UML

El Lenguaje Unificado de Modelado (UML), es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema. El UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

1.6.2. Visual Paradigm para UML

Es una herramienta *CASE* UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El lenguaje de modelado UML ayuda a una construcción más rápida de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso y generar código desde diagramas. Aunque la metodología seleccionada fue la metodología XP y la misma no establece el uso del UML como un paso fundamental si no una estrategia opcional, porque en su lugar esta metodología tiene las tarjetas CRC que facilitan mucho el trabajo.

1.7. Conclusiones del capítulo

Tras el estudio de los diferentes conceptos y las tecnologías relacionadas con los sistemas contables que se usan en el mundo, Cuba y la UCI. Se puede corroborar que:

- ✓ Los sistemas vistos presentan alguna dificultad que no le permite cumplir los requerimientos del cliente.
- ✓ Se desarrollará una aplicación de escritorio, utilizando el lenguaje de programación C#.
- ✓ Para el control de datos que requiere la misma se utilizará el gestor de base de datos SQL Server 2005.
- ✓ La aplicación se desarrollará con herramientas propietarias, en el caso de C# la decisión fue tomada puesto que el cliente requiere una aplicación de escritorio y se imposibilita hacerla en Java por los pobres recursos de hardware de las estaciones

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

de trabajo de la empresa y el gestor de base de datos, porque es el que se utiliza actualmente en la entidad, que dicho sea de paso no pretende migrar a software libre en breve espacio de tiempo.

- ✓ Se utilizará XP como metodología de desarrollo.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Capítulo 2: Características del Sistema

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. También se plasman las fases que dan inicio a la metodología XP: exploración-planificación y diseño.

2.1. Fase de Exploración-Planificación

Es la primera de las cuatro fases de la metodología XP y en ella se identifican las historias de usuario y 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 el pilar fundamental cuando se utiliza la mencionada metodología. Particularmente en este proyecto fue de gran importancia la misma, ya que el equipo de trabajo tuvo que aprender conceptos de economía y contabilidad para poder entender las necesidades del cliente.

2.1.1. Historias de usuario

Las historias de usuario son los primeros artefactos que aparecen en este capítulo. Es una forma particular en la que se llaman a los requerimientos en esta metodología. Estas historias de usuario se redactan desde el punto de vista del cliente, aunque el criterio de los desarrolladores también se tiene en cuenta. En la fase de exploración se detectaron 17 historias de usuario. Las cuales se listan a continuación:

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 1	Usuario: Técnico
Nombre historia: Ingreso por cheque	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0,5	Iteración asignada: 3
Programador responsable: Darriel Osorio Plasencia y Alain Garcia Asencio	
Descripción: Maneja los ingresos que asume la UERMP por los cheques que llegan a la entidad por motivo de pago.	
Observaciones: Es una actividad sencilla no exige de alto nivel de complejidad.	

Tabla 1: Historia de Usuario: Ingreso por cheque.

Historia de Usuario	
Número: 2	Usuario: Técnico
Nombre historia: Egreso por cheque	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Darriel Osorio Plasencia y Alain Garcia Asencio	
Descripción: Se contabilizan todos los egresos por criterios de cheques hacia otras entidades ya sean nacionales o no. Todos los cheques válidos que se emiten desde la UERMP.	
Observaciones: Este tipo de actividad tiene mucha dependencia de otros componentes del Sistema. De ahí su riesgo en el desarrollo.	

Tabla 2: Historia de Usuario: Egreso por cheque.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 3	Usuario: Técnico
Nombre historia: Ingreso por traspaso	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 0,5	Iteración asignada: 2
Programador responsable: Dariel Osorio Plasencia y Alain Garcia Asencio	
Descripción: Controla las salidas por criterios de transacción hacia subcuentas. Por lo que hay que actualizar el estado de las subcuentas.	
Observaciones: Mucha dependencia de subcuentas, flujos así como otros componentes.	

Tabla 3: Historia de Usuario: Ingreso por traspaso.

Historia de Usuario	
Número: 4	Usuario: Técnico
Nombre historia: Egreso por traspaso	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Dariel Osorio Plasencia y Alain Garcia Asencio	
Descripción: Controla el estado de la cuenta principal haciendo un análisis de cada una de las subcuentas que forman parte de la misma.	
Observaciones: Es una de las actividades con más dependencias pero no requiere de alto nivel computacional para desarrollarla.	

Tabla 4: Historia de Usuario: Egreso por traspaso.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 5	Usuario: Especialista
Nombre historia: Estado de cuenta por subcuenta	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0,5	Iteración asignada: 3
Programador responsable: Dariel Osorio Plasencia y Alain Garcia Asencio	
Descripción: Es un consolidado general de las subcuentas, que dan un monto total que es el estado general financiero de la unidad.	
Observaciones: Depende de consultar todas las cuentas así como su saldo real.	

Tabla 5: Historia de Usuario: Estado de cuenta por subcuenta.

Historia de Usuario	
Número: 6	Usuario: Especialista
Nombre historia: Clasificador de epígrafe	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0,5	Iteración asignada: 1
Programador responsable: Dariel Osorio Plasencia	
Descripción: Número de serie mediante el cual se especifica los criterios de ingresos y egresos a la empresa. Es un número que clasifica un tipo de operación específica.	
Observaciones: Los números no deben cambiar ya que son de total dominio del cliente.	

Tabla 6: Historia de Usuario: Clasificador de epígrafe.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 7	Usuario: Técnico
Nombre historia: Ejecución de presupuesto	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Dariel Osorio Plasencia	
Descripción: Mediante esta operación se decide si se puede otorgar crédito a una empresa cualquiera o si se puede emplear el fondo para cualquier acción.	
Observaciones: Imprescindible de implementar.	

Tabla 7: Historia de Usuario: Ejecución de presupuesto.

Historia de Usuario	
Número: 8	Usuario: Informático
Nombre historia: Entrada de subcuenta	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 0,5	Iteración asignada: 3
Programador responsable: Dariel Osorio Plasencia	
Descripción: Crear nuevas cuentas.	
Observaciones: En caso de que se cree una nueva cuenta en cualquier sub-entidad perteneciente.	

Tabla 8: Historia de Usuario: Entrada de subcuenta.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 9	Usuario: Informático
Nombre historia: Funcionario autorizado	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 0,5	Iteración asignada: 3
Programador responsable: Alain Garcia Asencio	
Descripción: Gestiona los diferentes usuarios así como sus permisos.	
Observaciones: La ejecuta el informático.	

Tabla 9: Historia de Usuario: Funcionario autorizado.

Historia de Usuario	
Número: 10	Usuario: Jefe de Técnicos
Nombre historia: Ajuste y cancelación	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1,5	Iteración asignada: 1
Programador responsable: Alain Garcia Asencio	
Descripción: Es para realizar los cobros pendientes así como cancelar pagos pendientes.	
Observaciones: Necesita continuamente consultar la base de datos	

Tabla 10: Historia de Usuario: Ajuste y cancelación.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 11	Usuario: Especialista
Nombre historia: Chequeo de cuenta y presupuesto	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Alain Garcia Asencio	
Descripción: Revisar si es matemáticamente posible emitir pagos de las diferentes subcuentas.	
Observaciones: Deben acceder todos los técnicos, operadores y especialistas.	

Tabla 11: Historia de Usuario: Chequeo de cuenta y presupuesto.

Historia de Usuario	
Número: 12	Usuario: Especialista
Nombre historia: Hallar mejor cliente	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Dariel Osorio Plasencia	
Descripción: Buscar el cliente que más dinero ingresa en un período de tiempo establecido.	
Observaciones: El tiempo es ingresado al sistema por el especialista.	

Tabla 12: Historia de Usuario: Hallar mejor cliente.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 13	Usuario: Especialista
Nombre historia: Consolidar <i>flujo de caja</i> *	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1,5	Iteración asignada: 1
Programador responsable: Alain Garcia Asencio	
Descripción: Esquema de todas las actividades que se realizó durante un tiempo determinado.	
Observaciones: Es el resultado del uso de la aplicación con su acumulado.	

Tabla 13: Historia de Usuario: Consolidar flujo de caja.

Historia de Usuario	
Número: 14	Usuario: Jefe de Técnicos
Nombre historia: Cuadre diario-semanal	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 0,5	Iteración asignada: 1
Programador responsable: Dariel Osorio Plasencia	
Descripción: Parte que se hace antes de cerrar el día y la semana.	
Observaciones: Se utiliza en los consejos financieros.	

Tabla 14: Historia de Usuario: Cuadre diario-semanal.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 15	Usuario: Especialista
Nombre historia: Cuadre mensual-anual	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 0,5	Iteración asignada: 1
Programador responsable: Alain Garcia Asencio	
Descripción: Cierre que se hace para hacer una valoración resumen del mes y del año.	
Observaciones: Sirve para la toma de decisiones, es un balance completo.	

Tabla 15: Historia de Usuario: Cuadre mensual-anual.

Historia de Usuario	
Número: 16	Usuario: Especialista
Nombre historia: Conformación de reporte	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 0,5	Iteración asignada: 3
Programador responsable: Dariel Osorio Plasencia	
Descripción: Realizar los reportes que se archivarán.	
Observaciones: Es muy general y es estándar según el tipo de reporte que se desee.	

Tabla 16: Historia de Usuario: Conformación de reporte.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 17	Usuario: Especialista, Técnico, Jefe Técnico, Informático
Nombre historia: Autenticar usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Alain Garcia Asencio	
Descripción: Para registrar los usuarios antes de acceder directamente al sistema.	
Observaciones: Cuenta con las mínimas exigencias de seguridad en los datos transferidos.	

Tabla 17: Historia de Usuario: Autenticar usuario.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1.2. Personas relacionadas con el sistema

Existen diferentes tipos de usuarios que trabajan directamente con el sistema o sea interactúan con él, obteniendo resultados en los procesos desarrollados. A estos individuos se les denomina: Personas relacionadas con el sistema.

Personas relacionadas con el Sistema	Justificación
Técnico	Es el principal gestor de cada una de las actividades ordinarias que realiza el sistema, ya que interacciona con la mayor parte de él.
Jefe de Técnicos	Supervisa generalmente el trabajo de los técnicos. Emite documentos que no son más que cúmulos del trabajo hecho por los técnicos revisados por él.
Especialista	Tiene acceso a todo tipo de información del sistema, así como a los registros de cada uno de los usuarios de los diferentes roles. Puede modificar información siempre y cuando no afecte el proceso de auditoría.
Informático	Administrador del sistema es quien da alta y baja a los diferentes usuarios y es el encargado de dar mantenimiento y realizar las salvadas de la base de datos.

Tabla 18: Personas relacionadas con el sistema.

2.1.3. Requisitos no funcionales del sistema

Los requisitos no funcionales de este sistema tienen la salvedad que el cliente exige una solución sobreponiendo de antemano las características y prestaciones de cada una de

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

las máquinas que hay en existencia en la UERMP. Por lo que desde el mismo momento en que se ideó una solución reinó la interrogante:

¿Bajo qué condiciones se va a explotar este software?

2.1.3.1. Requisitos de usabilidad

- ✓ Los usuarios deben tener conocimientos informáticos elementales.
- ✓ Las interfaces de usuarios deben estar escritas en un lenguaje claro y ser lo más intuitivo posible.
- ✓ Las instrucciones de navegación para los usuarios deben ser precisas.

2.1.3.2. Requisitos de rendimiento

- ✓ La aplicación debe tener una velocidad de procesamiento óptimo y un buen aprovechamiento de los recursos del servidor.
- ✓ Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos (no más de 15 segundos), tanto para los accesos a la bases de datos como para la gestión de los procesos.
- ✓ Lograr un producto que cumpla con la gestión de la información que se necesita.
- ✓ Será una aplicación de escritorio, lo que le confiere una mayor rapidez y las conexiones de red deben hacer invariable esta característica, ya que el hardware no es el más idóneo.

2.1.3.3. Requisitos de seguridad

- ✓ Los usuarios pueden acceder a los módulos que les son asignados según su rol en el sistema.
- ✓ Las contraseñas se encriptan con el algoritmo MD5, lo que permite que los resultados sean irreversibles por lo que brinda gran seguridad. Un ejemplo de ello es que la notación MD5 estima que para hallar dos cadenas encriptadas iguales para diferentes cadenas de texto se supone una dificultad de 2^{64} intentos.
- ✓ Respetar las políticas de seguridad de la empresa.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1.3.4. Requisitos de software

Servidor:

- ✓ *Windows Service Pack 2* o cualquier versión superior de *Windows*.
- ✓ *SQL Server 2005* o cualquier versión superior de *SQL*.
- ✓ *SQL Server Management Studio Express*.
- ✓ Marco de trabajo .NET versión 2.0.

Cliente:

- ✓ *Windows Service Pack 2* o cualquier versión superior de *Windows*.
- ✓ Marco de trabajo .NET versión 2.0.

2.1.3.5. Requisitos de hardware

Servidor:

- ✓ Como mínimo 512 MB de Memoria de Acceso Aleatorio (RAM), por sus siglas en inglés.
- ✓ Como mínimo 60 GB de memoria física.

Cliente:

- ✓ Como mínimo 256 MB de RAM.
- ✓ Como mínimo 40 GB de memoria física.

2.1.3.6. Requisitos de confiabilidad

- ✓ Hacer un software robusto, en el cual cada usuario tenga acceso a los privilegios que le corresponden. Además todas las operaciones matemáticas que realice el software deben ser cuidadosamente programadas, para evitar errores que pueden traer consecuencias.
- ✓ La aplicación tiene que ser altamente confiable pues a través de ella se manipularán datos confidenciales, el buen uso de esta información es directamente proporcional a los buenos resultados obtenidos.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1.3.7. Requisitos de disponibilidad

- ✓ El sistema debe estar disponible durante todo el horario laboral, porque los usuarios necesitan trabajar con él directamente y así puedan estar haciéndole peticiones continuas y recibiendo la respuesta adecuada, en el momento adecuado.

2.1.4. Estimación de esfuerzo por historias de usuario

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

Historia de Usuario	Puntos Estimados
Estado de cuenta por subcuenta	1
Ejecución de presupuesto	0,5
Chequeo de cuenta y presupuesto	1,5
Hallar mejor cliente	1
Consolidar flujo de caja	1,5
Cuadre diario-semanal	0,5
Cuadre mensual-anual	0,5
Ingreso por cheque	0,5
Clasificador de epígrafe	0,5
Funcionario autorizado	0,5
Conformación de reporte	0,5
Egreso por cheque	0,5
Egreso por traspaso	1
Ingreso por traspaso	1
Entrada de subcuenta	1
Ajuste y cancelación	0,5
Autenticar usuario	0,5

Tabla 19: Estimación de esfuerzo por historia de Usuario.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1.5. Plan de iteraciones

Conocidas las historias de usuario y el esfuerzo que se requiere para desarrollarlas, se procede a dividir el trabajo en iteraciones, para poder tener un desarrollo 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 hizo un balance, para que cada iteración fuera escenario de conjuntos de historias de usuario que demanden un desempeño aproximado del equipo de trabajo.

Iteración 1

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

Iteración 2

En esta iteración se 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

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 él.

2.1.6. Plan 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 este equipo de trabajo cuenta con dos integrantes, se tomó la decisión de que algunas de las historias de usuario serían realizadas en conjunto, pero en otras no era necesario.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

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 los miembros.

Iteración	Historia de usuario	Puntos estimados	Puntos de iteración
1	Ingreso por cheque	0,5	2
	Clasificador de epígrafe	0,5	
	Funcionario autorizado	0,5	
	Ingreso por traspaso		
	Conformación de reporte	0,5	
2	Estado de cuenta por subcuenta	1	6
	Hallar mejor cliente	1	
	Ejecución de presupuesto	0,5	
	Chequeo de cuenta y presupuesto	1,5	
	Consolidar flujo de caja	1,5	
	Cuadre diario-semanal	0,5	
	Cuadre mensual-anual	0,5	
3	Egreso por cheque	1	2,5
	Egreso por traspaso	0,5	
	Entrada de subcuenta	1	
	Ajuste y cancelación	0,5	
	Autenticar usuario	0,5	

Tabla 20: Planificación de duración de iteraciones.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1.7. Tareas de ingeniería

Tarea de Ingeniería	
Número de Tarea: 1	HU*#1: Chequeo de Ingresos
Nombre de Tarea: Establecer Período	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0,2
Fecha de Inicio: 7 febrero de 2010	Fecha Fin: 8 febrero de 2010
Programador Responsable: Dariel Osorio Plasencia	
Descripción: Se debe desarrollar un ambiente en el que el usuario seleccione inicialmente el tiempo para el cual quiere saber los ingresos.	

Tabla 21: Tarea de Ingeniería: Establecer Período.

Tarea de Ingeniería	
Número de Tarea: 2	HU#1: Chequeo de Ingresos
Nombre de Tarea: Realizar Monto	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0,2
Fecha de Inicio: 8 febrero de 2010	Fecha Fin: 9 febrero de 2010
Programador Responsable: Alain Garcia Asencio	
Descripción: Hacer un conglomerado de los egresos en ese período ubicándolos por destinatarios de esos ingresos.	

Tabla 22: Tarea de Ingeniería: Realizar Monto.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tarea de Ingeniería	
Número de Tarea: 3	HU#1: Chequeo de Ingresos
Nombre de Tarea: Revisar Chequeo	
Tipo de Tarea: Mejora	Puntos Estimados: 0,1
Fecha de Inicio: 8 febrero de 2010	Fecha Fin: 9 febrero de 2010
Programador Responsable: Dariel Osorio Plasencia	
Descripción: Revisar la lógica de la programación, así como la correcta ejecución del código generado.	

Tabla 23: Tarea de Ingeniería: Revisar Chequeo.

Tarea de Ingeniería	
Número de Tarea: 4	HU# 2: Egresos por Emisión de Cheques
Nombre de Tarea: Seleccionar Egresos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha de inicio: 8 febrero de 2010	Fecha Fin: 9 febrero de 2010
Programador Responsable: Alain Garcia Asencio	
Descripción: Seleccionar todos los egresos que se han producido en el sistema bajo el criterio de cheque en un intervalo de tiempo establecido.	

Tabla 24: Tarea de Ingeniería: Seleccionar Egresos.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tarea de Ingeniería	
Número de Tarea: 5	HU# 2: Egresos por Emisión de Cheques
Nombre de Tarea: Chequera	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.8
Fecha de Inicio: 8 febrero de 2010	Fecha Fin: 12 febrero de 2010
Programador Responsable: Alain Garcia Asencio	
Descripción: Establecer una chequera digital con los talonarios reales otorgados a la empresa por el Banco Central de Cuba.	

Tabla 25: Tarea de Ingeniería: Chequera.

Tarea de Ingeniería	
Número de Tarea: 6	HU# 2: Egresos por Emisión de Cheques
Nombre de Tarea: Revisar Chequera	
Tipo de Tarea: Corrección	Puntos Estimados: 0.1
Fecha de Inicio: 13 febrero de 2010	Fecha Fin: 13 febrero de 2010
Programador Responsable: Alain Garcia Asencio	
Descripción: Velar por la correcta lógica de esta funcionalidad teniendo en cuenta que se deben encontrar principalmente errores en el código.	

Tabla 26: Tarea de Ingeniería: Revisar Chequera.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tarea de Ingeniería	
Número de Tarea: 7	HU# 3: Egresos por Traspaso hacia Subcuentas
Nombre de Tarea: Traspasos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Fecha de Inicio: 15 febrero de 2010	Fecha Fin: 18 febrero de 2010
Programador Responsable: Alain García Asencio	
Descripción: Establecer una transacción de una cuenta a otra por compromisos adquiridos mediante préstamos facturas o cheques.	

Tabla 27: Tarea de Ingeniería: Traspasos.

Tarea de Ingeniería	
Número de Tarea: 8	HU# 4: Estado de Cuentas por Subcuentas
Nombre de Tarea: Llevar Estado Cuenta	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.6
Fecha de Inicio: 16 febrero de 2010	Fecha Fin: 19 febrero de 2010
Programador Responsable: Dariel Osorio Plasencia	
Descripción: Hacer una sumatoria de todas las subcuentas y ese monto no es más que el estado de cuenta de la empresa.	

Tabla 28: Tarea de Ingeniería: Llevar Estado de Cuenta.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tarea de Ingeniería	
Número de Tarea: 9	HU# 5: Presupuesto Clasificador de Epígrafes
Nombre de Tarea: Crear Epígrafe	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.6
Fecha de Inicio: 19 febrero de 2010	Fecha Fin: 23 febrero de 2010
Programador Responsable: Dariel Osorio Plasencia	
Descripción: Crear los diferentes criterios mediante los cuales se pueden o no hacer operaciones dentro del sistema.	

Tabla 29: Tarea de Ingeniería: Crear Epígrafe.

Tarea de Ingeniería	
Número de Tarea: 10	HU# 5: Presupuesto Clasificador de Epígrafes
Nombre de Tarea: Diseñar Reportes por Epígrafes	
Tipo de Tarea: Mejora	Puntos Estimados: 0.2
Fecha de Inicio: 18 febrero de 2010	Fecha Fin: 20 febrero de 2010
Programador Responsable: Alain García Asencio	
Descripción:	

Tabla 30: Tarea de Ingeniería: Diseñar Reportes por Epígrafes.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tarea de Ingeniería	
Número de Tarea: 11	HU# 6: Ejecución de Presupuestos
Nombre de Tarea: Verificar Presupuesto	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Fecha de Inicio: 20 febrero de 2010	Fecha Fin: 21 febrero de 2010
Programador Responsable: Alain Garcia Asencio	
Descripción: Implementar que cada petición de presupuesto se le confirme si económicamente está en condiciones de recibir un presupuesto.	

Tabla 31: Tarea de Ingeniería: Verificar Presupuesto.

Tarea de Ingeniería	
Número de Tarea: 12	HU# 6: Ejecución de Presupuestos
Nombre de Tarea: Asignar Presupuesto	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.6
Fecha de Inicio: 23 febrero de 2010	Fecha Fin: 26 febrero de 2010
Programador Responsable: Dariel Osorio Plasencia	
Descripción: Asignar o denegar el presupuesto al solicitante.	

Tabla 32: Tarea de Ingeniería: Asignar Presupuesto.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tarea de Ingeniería	
Número de Tarea: 13	HU# 7: Entrada de Subcuentas
Nombre de Tarea: Entrada por subcuentas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha de Inicio: 21 febrero de 2010	Fecha Fin: 23 febrero de 2010
Programador Responsable: Alain Garcia Asencio	
Descripción: Ingresos que tienen lugar en las subcuentas de la Entidad.	

Tabla 33: Tarea de Ingeniería: Entrada por subcuentas.

Tarea de Ingeniería	
Número de Tarea: 14	HU# 8: Funcionarios Autorizados
Nombre de Tarea: Funcionarios	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,5
Fecha de Inicio: 26 febrero de 2010	Fecha Fin: 1 marzo de 2010
Programador Responsable: Dariel Osorio Plasencia	
Descripción: Conformar los permisos en la aplicación para los usuarios, según su rol.	

Tabla 34: Tarea de Ingeniería: Funcionarios.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tarea de Ingeniería	
Número de Tarea: 15	HU# 9: Ajuste y Cancelación
Nombre de Tarea: Ajuste y Cancelación	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0,5
Fecha de Inicio: 25 febrero de 2010	Fecha Fin: 28 febrero de 2010
Programador Responsable: Alain Garcia Asencio	
Descripción: Crear la funcionalidad para modificar errores en la base de datos.	

Tabla 35: Tarea de Ingeniería: Ajustes Cancelaciones.

Tarea de Ingeniería	
Número de Tarea: 16	HU# 10: Chequeo de Cuentas y Presupuestos
Nombre de Tarea: Cuentas y Presupuestos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1,5
Fecha de Inicio: 1 marzo de 2010	Fecha Fin: 12 marzo de 2010
Programador Responsable: Dariel Osorio Plasencia	
Descripción: Mostrar los presupuestos según los estados de cuenta.	

Tabla 36: Tarea de Ingeniería: Cuentas y Presupuesto.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tarea de Ingeniería	
Número Tarea: 17	HU# 11: Flujo de Caja Diario-Semanal
Nombre de Tarea: Caja Semanal	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha de Inicio: 28 febrero de 2010	Fecha Fin: 6 marzo de 2010
Programador Responsable: Alain Garcia Asencio	
Descripción: Hacer los Flujos de Caja por semana según indica el cliente.	

Tabla 37: Tarea de Ingeniería: Caja Semanal.

Tarea de Ingeniería	
Número Tarea: 18	HU# 12: Flujo de Caja Mensual-Anual
Nombre Tarea: Caja Anual	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha de Inicio: 12 marzo de 2010	Fecha Fin: 19 marzo de 2010
Programador Responsable: Dariel Osorio Plasencia	
Descripción: Hacer los Flujos de Caja por semana según indica el cliente.	

Tabla 38: Tarea de Ingeniería: Caja Anual.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Tarea de Ingeniería	
Número de Tarea: 19	HU# 13: Consolidar Flujo de Caja
Nombre de Tarea: Consolidar	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha de Inicio: 6 marzo de 2010	Fecha Fin: 12 marzo de 2010
Programador Responsable: Alain Garcia Asencio	
Descripción: Dejar las condiciones en subcuentas y cuentas creadas para el cierre del día.	

Tabla 39: Tarea de Ingeniería: Consolidar.

2.1.8. Plan de entregas

Artefacto	Hito	Entrega
Sistema gestor de procesos económicos de la casa financiera versión 0.1	Final de la primera iteración.	15-18 febrero 2010
Sistema gestor de procesos económicos de la casa financiera versión 0.2	Final de la segunda iteración.	30 marzo-1 abril 2010
Sistema gestor de procesos económicos de la casa financiera versión 1.0	Final de la tercera iteración.	16-19 abril 2010
Tesorería UERMP Beta (Nombre de la aplicación)	Final de la fase de implementación.	12-14 mayo de 2010

Tabla 40: Planificación de entrega.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.2. Fase de Diseño

En XP se trabaja con las tarjetas Clase-Responsabilidad-Colaboración (CRC) las cuales sustituyen el UML y hacen el trabajo mucho más comunicativo. Aunque no hacen que la metodología esté exenta de este lenguaje, ya que en los casos en que el manejo de las clases y la dependencia entre las mismas es muy grande, entonces es recomendable utilizarlo ya que mejora el entendimiento del problema y ayuda a trazarse estrategias mucho más certeras.

2.2.1. Metáfora para el sistema

A la UERMP le es necesario tener una aplicación que agilice los procesos económicos, mantenga correctamente actualizados los estados de cuenta de la casa financiera de la casa matriz así como los de las cuentas de las sedes diseminadas a lo largo y ancho de la isla. Esa aplicación debe ser capaz de emitir reportes de períodos de trabajos especificados por los usuarios para ayudar al colectivo de trabajadores en la toma de decisiones. Esta aplicación debe cumplir con todas las funciones requeridas por el negocio de forma diáfana y transparente debido a la información que se maneja.

2.2.2. Diseño de la base de datos

Para diseñar la base de datos hubo que respetar, como en muchos de los casos, los criterios que tenía el cliente sobre el funcionamiento. Existe gran acumulación de datos, producto a los doce años de explotación del software existente y estos comprenden facturas económicas, registros de préstamos, así como otros compromisos comerciales adquiridos durante este extenso período. Esto es una muestra fehaciente, de que los datos de la aplicación antigua debían estar guardados en la nueva aplicación. Por lo que se migra la salva de la base de datos del día 21 de enero del año en curso, para el gestor *SQL Server 2005* con la herramienta *SSMA*.

Se obtiene una base de datos que cuenta con 54 tablas, las cuales estaban saturadas de datos en muchos casos y en otros existían tablas vacías a pesar del tiempo de explotación. Lo que demostraba que este software no fue concebido desde un inicio para esa entidad y el primer trabajo del equipo de desarrollo a partir de ese momento fue entonces eliminar las tablas vacías y con las restantes reestructurar un diseño de base de

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

datos en el ER Studio 8.0 sin violar las estructuras iniciales, ni las relaciones entre tablas. Obteniendo un diseño mejorado de la base de datos inicial se logra migrar al SQL, que sería entonces el diseño original de la aplicación y a partir del mismo se desarrollan todas las funcionalidades, debido a que las clases persistentes de cada uno de los diferentes módulos estarían reflejadas en el diseño final.

Se obtiene una base de datos en segunda forma normal que no pudo ser transformada producto a que si se modifican los datos de las tablas entonces sería muy engorroso o hasta imposible migrar los datos de la aplicación que se está utilizando.

2.3. Tarjetas CRC

Clase: Ajustes	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Realizar los ajustes de cobros pendientes que se llevan a cabo en la empresa. - Cancelar emisiones de pago o de cobro que no se han realizado. 	<ul style="list-style-type: none"> - Egresos - Ejecución - Estado - Ingresos - Reporte Cancelación - Solicitud Egreso - Solicitud Ingreso - Solicitud Traspaso

Tabla 41: Ajustes.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Clase: Auditor	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Acceder a los reportes acumulativos de los registros diarios. 	<ul style="list-style-type: none"> - Cambiar contraseña - Egresos - Ejecución - Estado - Mostrar Cancelaciones - Mostrar Estado - Reporte Cancelación - Traspasos

Tabla 42: Auditor.

Clase: Cambiar Contraseña	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Cambiar la contraseña de los diferentes usuarios. 	<ul style="list-style-type: none"> - Auditor - Usuario - Informático - Operador

Tabla 43: Cambiar Contraseña.

Clase: Chequeo de Presupuesto	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Verificar la disponibilidad de presupuesto para realizar operaciones que demanden egreso de las diferentes cuentas. 	<ul style="list-style-type: none"> - Solicitud Egreso - Solicitud Traspaso - Sub-Cuentas - Traspasos

Tabla 44: Chequeo de Presupuesto.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Clase: Clasificador	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Clasificar los epígrafes para valorar los criterios de egreso e ingreso según estos. 	<ul style="list-style-type: none"> - Egresos - Ingresos - Traspasos

Tabla 45: Clasificador.

Clase: Usuario	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Gestionar los diferentes usuarios de la aplicación así como los datos que contienen o generan. 	<ul style="list-style-type: none"> - Auditor - Cambiar contraseña - Informático - Operador

Tabla 46: Usuario.

Clase: Egresos	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Registrar y realizar las salidas de presupuestos de las cuentas. - Registrar los egresos. 	<ul style="list-style-type: none"> - Chequeo de Presupuesto - Ejecución - Estado - Operador - Solicitud Egreso

Tabla 47: Egresos.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Clase: Ejecución	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Ejecutar el presupuesto siempre que sea matemáticamente posible. 	<ul style="list-style-type: none"> - Chequeo de Presupuesto - Egresos - Estado - Solicitud Egreso - Sub-Cuentas

Tabla 48: Ejecución.

Clase: Entrada de Sub-Cuentas	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Crear nuevas subcuentas y registrarlas. 	<ul style="list-style-type: none"> - Chequeo de Presupuesto - Sub-Cuentas

Tabla 49: Entrada de Sub Cuentas.

Clase: Estado	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Conocer el estado matemático de las diferentes cuentas. 	<ul style="list-style-type: none"> - Chequeo de Presupuesto - Sub-Cuentas

Tabla 50: Estado.

Clase: Informático	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Gestionar los datos de este usuario. 	<ul style="list-style-type: none"> - Cambiar contraseña - Estado - Mostrar Cancelaciones - Mostrar Estado - Usuario

Tabla 51: Informático.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Clase: Ingresos	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Gestionar todo el presupuesto que ingresa a la entidad bajo cualquier criterio. 	<ul style="list-style-type: none"> - Chequeo de Presupuesto - Ejecución - Estado - Operador - Solicitud Egreso

Tabla 52: Ingresos.

Clase: Mostrar Cancelaciones	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Consultar todas las cancelaciones que se han realizado. 	<ul style="list-style-type: none"> - Ajustes

Tabla 53: Mostrar Cancelaciones.

Clase: Mostrar Estado	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Conocer el estado de determinadas cuentas. - Conocer el estado de las operaciones que se han realizado. 	<ul style="list-style-type: none"> - Chequeo de Presupuesto - Estado - Sub-Cuentas - Traspasos

Tabla 54: Mostrar Estado.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Clase: Operador	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Gestionar este tipo de usuario. - Realizar operaciones básicas en la aplicación. 	<ul style="list-style-type: none"> - Ajustes - Cambiar contraseña - Egresos - Ejecución - Entrada de Sub-Cuentas - Estado - Ingresos - Mostrar Estado - Reporte Cancelación - Solicitud Egreso - Solicitud Ingreso - Solicitud Traspaso - Sub-Cuentas - Traspasos

Tabla 55: Operador.

Clase: Reporte Cancelación	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Consultar las cancelaciones realizadas en la empresa. 	<ul style="list-style-type: none"> - Ajustes

Tabla 56: Reporte Cancelación.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Clase: Solicitud Egreso	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Solicitar mediante un modelo un egreso. 	<ul style="list-style-type: none"> - Chequeo de Presupuesto - Estado - Sub-Cuentas

Tabla 57: Solicitud Egreso.

Clase: Solicitud Ingreso	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Solicitar mediante un modelo un ingreso. 	<ul style="list-style-type: none"> - Chequeo de Presupuesto - Estado - Sub-Cuentas

Tabla 58: Solicitud Ingreso.

Clase: Solicitud Traspaso	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Solicitar mediante un modelo un traspaso. 	<ul style="list-style-type: none"> - Chequeo de Presupuesto - Estado - Sub-Cuentas

Tabla 59: Solicitud Traspaso.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Clase: Sub-Cuentas	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Gestionar las diferentes cuentas que pertenecen a la empresa. 	<ul style="list-style-type: none"> - Ajustes - Chequeo de Presupuesto - Egresos - Entrada de Sub-Cuentas - Estado - Mostrar Estado

Tabla 60: Sub-Cuentas.

Clase: Traspasos	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> - Realizar traspasos de una cuenta a otra. 	<ul style="list-style-type: none"> - Egresos - Ingresos - Sub-Cuentas

Tabla 61: Traspasos.

2.4. Estimación de esfuerzo por el COCOMO II

Desarrollado y propuesto por el estadounidense *Barry Boehm* el Modelo Constructivo de Costos (COCOMO), por su acrónimo del inglés, consiste en un modelo de estimación de costos muy documentado y figura entre los más utilizados. Permite estimar el costo, esfuerzo y tiempo cuando se planifican proyectos de desarrollo de software. Compuesto por tres sub modelos que se adaptan tanto a las necesidades de los diferentes sectores descritos, como al tipo y cantidad de información. Estos tres sub modelos son: básico, intermedio y detallado.

Cada uno de estos sub modelos se divide en modos teniendo en cuenta el tipo de proyectos al que se va a aplicar la estimación. Y estos modos pueden clasificarse como:

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

- ✓ **Orgánico:** para proyectos en los cuales se reúnen un grupo pequeño de programadores experimentados y lo desarrollan en un entorno familiar. El tamaño del software logrado oscila entre unos pocos de miles de líneas de códigos (software pequeños) y unas decenas de miles (software tamaño medio).
- ✓ **Rígido o Empotrado:** para proyectos con fuertes restricciones, dentro de sus funcionalidades y/o pueden ser técnicas. El problema que se resuelve es único y es difícil basarse en la experiencia, debido a que puede no haberla.
- ✓ **Semilibre o Semiencajado*:** Es un modo intermedio entre el orgánico y el rígido aplicable a los proyectos en los cuales el equipo de desarrollo puede estar compuesto por personas experimentadas y no experimentadas.

Conocida la clasificación de los diferentes proyectos se procede a calcular la estimación de esfuerzo. Conocidas las características de la investigación se clasifica en el modo rígido dentro del sub modelo básico. La fórmula general del esfuerzo de desarrollo en el COCOMO es:

$$E = a_i S_k^{b_i} m(X)$$

Para calcular la estimación de esfuerzo, tiempo y costo a los clasificados en este grupo se calcula de la siguiente forma:

- ✓ **Costo:** $K_m = 3.6 S_k^{1.20}$
- ✓ **Tiempo:** $T = 2.5 K_m^{0.32}$

La tabla del COCOMO para los coeficientes de los diferentes modos es la siguiente:

Modo	Básico		Intermedio	
	<i>a</i>	<i>b</i>	<i>a</i>	<i>b</i>
Orgánico	2.4	1.05	3.2	1.05
Semiencajado	3.0	1.12	3.0	1.12
Empotrado	3.6	1.2	2.8	1.2

Tabla 62: Valores de los coeficientes para los modos del modelo de estimación y costos.

Como parte de la investigación es necesario saber el tiempo de desarrollo, es preciso utilizar la fórmula de tiempo.

S_k = número de líneas de código 21 799

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Quedando:

$$K_m = 3.6 (160740)^{1.20}$$

$$K_m = 578664$$

Teniendo K_m entonces se procede a calcular la segunda fórmula.

$$T = 2.5 (50570.5)^{0.32}$$

$$T = 2.5 (69,8)$$

$$T = 174$$

Arrojando como resultado que se necesitan de 174 días estimados de trabajo para la construcción de la aplicación.

2.5. Conclusiones del capítulo

- ✓ El capítulo que recién finaliza, muestra la facilidad que ofrece a equipos de desarrollos pequeños trabajar bajo la metodología XP.
- ✓ Se expone inicialmente la propuesta de solución.
- ✓ Se desarrollan las dos primeras fases de la metodología XP: exploración-planificación y diseño.
- ✓ Se obtienen los diferentes artefactos de estas fases.

CAPÍTULO 3: IMPLEMENTACIÓN, PRUEBA Y VALIDACIÓN DEL SISTEMA PROPUESTO.

Capítulo 3: Implementación, prueba y validación del sistema propuesto

En el siguiente capítulo se verá una representación física de cómo se implementó la solución, a través de representaciones gráficas, enfocándose en el diagrama de componente y diagrama de despliegue de la solución desarrollada. Además de que se validará y verificará el cumplimiento de los requerimientos estipulados, mediante la aplicación de métodos de pruebas que garanticen la calidad del sistema.

3.1. Técnicas de programación en el lenguaje C#

Este lenguaje proporciona características útiles, como los indizadores, atributos y delegados que hacen posible emplear en él técnicas de programación avanzada.

Los indizadores proporcionan una forma de tener acceso a clases o estructuras de la misma manera que a una matriz. Por ejemplo, se puede utilizar una clase que representa un departamento único en una compañía. La clase podría contener los nombres de todos los empleados del departamento y los indizadores podrían permitir el acceso a estos nombres.

Los atributos en C# proporciona un mecanismo, para agregar información declarativa sobre los tipos. Los atributos son de algún modo similares al concepto de anotaciones en Java. La información adicional sobre un tipo se coloca dentro de etiquetas declarativas que preceden la definición de tipo.

El lenguaje seleccionado admite el concepto de punteros a función como unas de las características que “heredó” de lenguajes antecesores, que le permiten elegir a qué función desea llamar en tiempo de ejecución. A través del uso de la clase Delegado, una instancia de delegado encapsula un método que es una entidad invocable.

En el caso de los métodos de instancia, el delegado consta de una instancia de la clase contenedora y un método en la instancia. En los métodos estáticos, una entidad invocable consta de una clase y un método estático en la clase. Así, se puede utilizar un delegado para invocar una función de cualquier objeto y los delegados se orientan a objetos, tienen seguridad de tipos y son seguros.

CAPÍTULO 3: IMPLEMENTACIÓN, PRUEBA Y VALIDACIÓN DEL SISTEMA PROPUESTO.

Existen tres pasos para definir y utilizar los delegados:

- ✓ Declaración
- ✓ Creación de instancias
- ✓ Invocación

3.1.1. Descripción del algoritmo más complejo en el desarrollo.

Dentro de las funcionalidades que más esfuerzo necesitaron en la aplicación una que es muy puntual destacar es la llamada Traspasos. La cual necesitó de mucha lógica y comprensión del equipo de trabajo así como un esfuerzo adicional en comparación con otras historias de usuario. Para que se pueda realizar este método debían cumplirse varias condiciones: no dejar espacios en blanco; que existan ambas cuentas, la que ingresa y la que egresa; que existan los epígrafes correspondientes a las dos cuentas antes mencionadas, de cumplirse las mismas se realizan un conjunto de operaciones para guardar los datos o actualizarlos, según corresponda. A continuación se brinda una explicación del método.

Se comienza verificando que todos los campos estén llenos, si hay alguno que no cumpla con la condición se levanta un error diciéndole al usuario que no puede dejar campos vacíos; sino, se sigue con el método. Luego, se comprueba que existan ambas cuentas, en caso positivo se guardan los valores originales de las mismas para luego actualizarlas, de no existir alguna de ellas se lanza un error y se limpia el campo correspondiente en el formulario permitiéndole al usuario entrar una cuenta válida. Si existen ambas cuentas, después de guardados los datos originales, se procede a analizar la existencia de los epígrafes, de no existir uno de ellos, o ambos, se lanzará un error y se limpiará el campo correspondiente para que el operador entre un epígrafe real; por otra parte, si existen se guardará en variables sus descripciones para luego guardarlas en las operaciones.

Después se actualizan las cuentas y se buscan en la tabla “Mostrar” que estén las mismas, si están se actualizan, sino, se insertan. Posteriormente, solo queda guardar las operaciones realizadas en las cuentas en la tabla “Operaciones” y guardar los datos en la tabla “Traspasos” para luego limpiar todos los campos.

CAPÍTULO 3: IMPLEMENTACIÓN, PRUEBA Y VALIDACIÓN DEL SISTEMA PROPUESTO.

3.2. Beneficios tangibles e intangibles

La implantación del sistema “Tesorería-UERMP” traerá a la empresa un cúmulo de beneficios. Estos pueden clasificarse en tangibles o intangibles según su significado económico, en eficiencia, rapidez así como otras características que hacen el trabajo del cliente mucho más placentero.

Como beneficios intangibles vale destacar que el cliente cuenta con un programa nuevo, con tecnología de punta, que puede generar reportes con mayor rapidez y así simplifica el contenido de trabajo de algunos de los empleados que ahora puede utilizarlo en otras labores de sus contenidos de trabajo. Se ha reducido considerablemente la posibilidad de entrada de errores en los datos primarios con los que trabaja la aplicación. La impresión de reportes se hace de manera automática, sin necesidad que los técnicos tengan que desplazarse de lugar físico. Los usuarios pueden hacer consultas a la base de datos de la aplicación referentes a los estados de cuenta, sin necesidad de esperar al cierre diario. Se pueden conocer los mejores clientes de la semana, del mes o de algún otro período deseado, teniendo en cuenta el número de operaciones realizadas en esta etapa; así como otras informaciones que con el sistema antiguo no se tenía acceso y en la mayoría de los casos no se podía ejecutar.

Como beneficio tangible se tiene que los trabajadores de la empresa desarrollarán más actividades en menos tiempo, lo que aumentará visiblemente la productividad de la UERMP traduciéndose en mayores ingresos.

3.3. Pruebas

En la metodología XP se propone la comprobación como un fundamento principal en el desarrollo del software, donde cada programador realiza pruebas a medida que produce su código. Las pruebas se integran en el proceso de integración continua y construcción, lo que rinde una plataforma altamente estable para el desarrollo futuro. La programación extrema define dos tipos de pruebas las cuales son: pruebas unitarias y pruebas de aceptación. El proceso de desarrollo de las pruebas ayuda al cliente a clarificar y concretar la funcionalidad de la historia de usuario y favorece la comunicación entre el cliente y el equipo de desarrollo. Además, ayuda a identificar y corregir fallos u omisiones en las historias de usuario. También permite corregir errores en las ideas del cliente, por

CAPÍTULO 3: IMPLEMENTACIÓN, PRUEBA Y VALIDACIÓN DEL SISTEMA PROPUESTO.

ejemplo encontrar resultados que el cliente espera en la implementación, pero para los que no existe ningún camino de ejecución que conduzca a ello. Permite identificar historias adicionales que no fueran obvias para el cliente o en las que cliente no hubiese pensado de no enfrentarse a dicha situación.

3.3.1. Pruebas unitarias

Una prueba unitaria es un modo de probar el correcto funcionamiento de un módulo o segmento de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. En XP la producción de código está dirigida por las pruebas unitarias. Estas pruebas son establecidas aconsejablemente antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Se hacen pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. 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.

3.3.2. Pruebas de aceptación

Las pruebas de aceptación hacen posible confirmar que las historias de usuario hayan sido implementadas correctamente al final de cada iteración. Este período de prueba se conoce también como período de caja negra donde se definirán las entradas al sistema y los resultados esperados de estas entradas. Una historia de usuario, puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de éstas, es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable. La prueba de aceptación es realizada por un grupo de usuarios finales o los clientes del sistema, para asegurarse que el sistema desarrollado cumple sus requisitos.

3.4. Conclusiones del capítulo

Durante este capítulo se hizo referencia a las etapas de implementación y prueba del software. Se hizo énfasis en:

CAPÍTULO 3: IMPLEMENTACIÓN, PRUEBA Y VALIDACIÓN DEL SISTEMA PROPUESTO.

- ✓ La simplicidad y la reutilización de código como factor indispensable para lograr buenos hábitos de programación.
- ✓ Los problemas para el mantenimiento del sistema y defectos que demeritan la calidad del producto.

También:

- ✓ Se demostró que los desarrolladores aprenden mucho de las funcionalidades cuando la implementan.
- ✓ Quedó demostrado que es mucho más eficaz que sea el cliente quien elabore las pruebas.

Conclusiones

Se arriba a las siguientes conclusiones:

- ✓ El sistema “Tesorería-UERMP” que automatiza la gestión de los procesos económicos en la Casa Financiero de la UERMP favorece el desarrollo de los procesos económicos que se realizan en este departamento, garantiza la validez de los datos, reduce el tiempo que duraba una operación, permitiendo realizar más operaciones en igual período de tiempo, posee las principales características de los sistemas contables, cuenta con mecanismos de seguridad para manipular información de forma segura, brinda así confidencialidad e integridad a los servicios que brinda la Casa Financiera.
- ✓ El sistema cumple con los estándares definidos para el trabajo con sistemas contables y a la vez con las normas cubanas de economía. Demostrando que está a la altura de software creados con igual propósito para pequeñas y medianas empresas.
- ✓ Se cumplió el objetivo trazado en el diseño teórico de la investigación, comprobándose la hipótesis elaborada, tributando directamente a la gestión de los procesos económicos de la Casa Financiera de la UERMP.
- ✓ El empleo de los métodos teóricos y empíricos proporcionó conocer el estado de los procesos de gestión de información que se desarrollan en la UERMP.
- ✓ Las tareas investigativas organizaron la investigación adecuadamente para lograr cumplir con los objetivos planificados en tiempo y con calidad.

Recomendaciones

- ✓ Hacer extensible la aplicación a otros gestores de base de datos.
- ✓ En otras versiones del sistema, agregar nuevas funcionalidades, y a largo plazo se puede optar por la opción de utilizar herramientas de software libre.
- ✓ Ampliar los contenidos de ayuda al usuario.

Referencias Bibliográficas

1. **UERMP.** *Historia de la UERMP.* Ciudad Habana : s.n., 2008.
2. **Rolle, Hans y Zietzke, Julian.** Monografías.com. *Monografías.com.* [En línea] [Citado el: 15 de Diciembre de 2009.] <http://www.monografias.com/trabajos10/sistcont/sistcont.shtml#concepto>.
3. **Álvarez, Félix y Cristina, Isabel.** Monografías.com. *Monografías.com.* [En línea] [Citado el: 15 de Diciembre de 2009.] <http://www.monografias.com/trabajos2/sistdecont.shtml>.
4. Monografías.com. *Monografías.com.* [En línea] [Citado el: 16 de Diciembre de 2009.] <http://www.monografias.com/trabajos39/contabilidad-de-costos/contabilidad-de-costos.shtml?monosearch>.
5. **InSoft.** InSoft. *InSoft.* [En línea] [Citado el: 13 de Diciembre de 2009.] <http://www.agrowin.com>.
6. —. InSoft. *InSoft.* [En línea] [Citado el: 26 de Noviembre de 2009.] <http://www.infofweb.com/contapyme/default.htm..>
7. *Enriquecimiento del sistema contable del Banco Central de Cuba.* **Tamargo Cerezal, Lourdes.** Ciudad Habana : s.n., 2008.
8. *El Versat-Sarasola: Sistema cubano de Gestión Contable-Financiera.* **Porteiro Sosa, Marisel.** Ciudad Habana : s.n., 2009.
9. **UCI.** *Manual de usuario: Assets NS.* Ciudad Habana : s.n., 2007.
10. **PostgreSQL.** PostgreSQL. *PostgreSQL.* [En línea] [Citado el: 12 de Diciembre de 2009.] <http://www.postgresql.org>.
11. **Oracle® E-Delivery.** Sitio Web de Oracle® E-Delivery. *Sitio Web de Oracle® E-Delivery.* [En línea] 2008. [Citado el: 12 de Diciembre de 2009.] http://edelivery.oracle.com/EPD/WelcomePage/get_form?ARU_LANG=E..
12. **MySQL.** MySQL Hispano. *MySQL Hispano.* [En línea] Octubre de 2007. [Citado el: 12 de Diciembre de 2009.] [http://www.mysql-hispano.org/..](http://www.mysql-hispano.org/)
13. **Saavedra, Jorge.** Mundo Informático. *Mundo Informático.* [En línea] 2009. [Citado el: 11 de Diciembre de 2009.] [http://jorgesaavedra.wordpress.com/category/sql-server-2005/.](http://jorgesaavedra.wordpress.com/category/sql-server-2005/)

REFERENCIAS BIBLIOGRÁFICAS

14. **Enciclopedia.** Enciclopedia libre en Español. *Enciclopedia libre en Español*. [En línea] [Citado el: 13 de Diciembre de 2009.] http://enciclopedia.us.es/index.php/Lenguaje_de_programaci%C3%B3n.
15. **Pozo, Pedro.** Clikear.com. *Clikear.com*. [En línea] [Citado el: 10 de Diciembre de 2009.] <http://www.clikear.com/manuales/csharp/c10.aspx>.
16. **Microsoft.** Inicio de Visual C++. *Inicio de Visual C++*. [En línea] [Citado el: 14 de Diciembre de 2009.] <http://msdn.microsoft.com/es-es/visualc/default.aspx>.
17. **Java.** Java Hispano. *Java Hispano*. [En línea] [Citado el: 15 de Diciembre de 2009.] <http://www.javahispano.org/>.
18. **Empresariales, D. D. C. D. C.** *Los subsistemas funcionales de la empresa*. 2009.

Glosario de términos

Cheques en tránsito:

Se refiere a los cheques que se están trabajando.

Consejo Financiero:

Es la reunión que se lleva a cabo en la UERMP por parte de los directivos de la empresa y el personal de la CF para trazar estrategias en cuanto a la economía concierne.

Control físico y auditable:

Se refiere a los datos de la base de datos plasmados en papel y archivados para posterior consulta de los auditores.

Flujo de caja:

Diferencia entre los cobros y pagos de una empresa en un período de tiempo determinado.

HU:

Historias de usuario.

Letras de cambio:

Es un título de crédito a la orden, abstracto, formal y completo, que contiene una promesa incondicionada de hacer pagar o, en su defecto, de pagar una suma determinada de dinero a su portador legitimado, vinculando solidariamente a todos sus firmantes.

Materia prima:

Elemento o conjunto de elementos que pueden transformarse por la acción de otros elementos que actúan sobre él

Monosémico:

Según la Real Academia de la Lengua Española, se aplica a la palabra que tiene un solo significado.

Multihilo:

Varios caminos, en el ambiente informático, varias vías de ejecución.

Multimoneda:

GLOSARIO DE TÉRMINOS

Se le llama a los sistemas contables cuando permiten, pasada la moneda de origen, generar los equivalentes en otras monedas.

Multiusuario:

Refiere más de un usuario

Procesos económicos:

Son todos aquellos procesos que se llevan a cabo que refieren una finalidad económica.

TCP/IP:

Protocolo de control de transmisión/Protocolo de Internet. Este protocolo representa todas las reglas de comunicación para Internet y se basa en la noción de dirección IP.

Http:

Protocolo de transferencia de hipertexto (HTTP, por sus siglas en inglés) es el protocolo usado en cada transacción de la Web (WWW)

Ftp:

Protocolo de Transferencia de Archivos (FTP, por sus siglas en inglés) es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor.

Semilibre o Semiencajado:

Modo en que se clasifica a los submodelos del Modelo Constructivo de Costos. Y su nombre se refiere a una clasificación intermedia referente a la libertad y las restricciones que guían un proyecto de software y la experiencia de los desarrolladores.