

# TRABAJO DE DIPLOMA

PARA OPTAR POR EL TÍTULO DE

# INGENIERO INFORMÁTICO

TÍTULO: SISTEMA PARA LA GESTIÓN DE DOCUMENTOS  
CONTABLES SOBRE LA CUENTA DE GASTOS DE LA ETE LIDIO  
RAMÓN PÉREZ

AUTOR: ORLANDO PRIETO SILVA

TUTORES: EXNEIDER PROENZA SILVA

RAMIRO ESCALONA PERDOMO

MOA, 2016

“AÑO 58 DE LA REVOLUCIÓN

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al departamento de informática del Instituto Superior Minero Metalúrgico Dr. Antonio Núñez Jiménez para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del \_\_\_\_\_.

Orlando Prieto Silva

---

Nombre completo del primer autor

Exneider Proenza Silva

---

Nombre completo del primer tutor

Ramiro Escalona Perdomo

---

Nombre completo del primer tutor

## *Dedicatoria*

*A mi madre Noira Silva Sánchez.*

*A mi segunda madre, mi abuela Rafaela Sánchez*

*A mi hermano mayor mi tío Osmani*

*A mi abuelo Arístides Urquía*

*A mis abuelas Oneida y Pastora*

*A la profesora Luisa*

*A mi padre Oxiris y mis hermanitos Orlenis y Orisbelis*

*A mis primas Yairi y Yaimé y mi primo Osmani*

*A todos esos amigos y amigas que tanto amor me han demostrado y que no alcanzaría esta cuartilla para mencionarlos*

## *Agradecimientos*

*A mi madre por tanto sacrificio para hacer de mí un hombre de bien, por poner a un lado su vida para vivir la mía, por enseñarme que el camino difícil es el que conduce a los mejores lugares, por ser mi amiga y consejera...*

*A mi abuela Rafaela por sentir tanto orgullo de un ser lleno de defectos por el llanto cuando la vida fue dura para mí, por ser más que una madre...*

*A mi tío Osmani por no tener nada que no sea también mío, por ser mi hermano siempre...*

*A mi abuelo Arístides por permitir que este hombre eligiera el camino del estudio y cargar con la responsabilidad económica familiar sin quejas...*

*A mis abuelas Oneida y Pastora por su ayuda incondicional cuando la elección de seguir se salía de mis manos...*

*A mi padre y hermanos por su preocupación, porque sé que en la distancia siempre me han tenido presente y sus buenos augurios me guían por el camino del éxito...*

*A la profesora Luisa por tanto amor y comprensión en un momento en que solo recibí maltrato...*

*A Luis (turi) Anache por ser hombre y persona más allá de las posesiones materiales...*

*A Osmani Verdecia, David, Elisa y Gelsi por tanto aprecio, porque sin su ayuda no hubiera estado la aplicación a tiempo...*

*Al resto de mi familia de amigos Yosbani, Mima, Toti, Mercei, Dianelis, Samuel, Saúl, Virgen, Yoima, Oswaldito, Saismelis, Ezequiel, mis hermanos Roger y Gabriel y mi siempre amada María Isabel gracias por quererme espero hacer honor y corresponder debidamente sus sentimientos...*

*A José Luis Fernández Guza por su ayuda desinteresada en todo momento, por enseñarme a programar...*

*A los profesores que a lo largo de mi vida limaron los defectos y alimentaron las virtudes que necesité para llegar al día de hoy...*

## *Pensamiento*

*La virtud, como el arte, se consagra constantemente a lo que es difícil de hacer, y cuanto más dura es la tarea, más brillante es el éxito.*

*Aristóteles*

## Resumen

Las Tecnologías de Información y las Telecomunicaciones no son más que un medio para transmitir y gestionar datos, información y conocimiento, el conocimiento es factor fundamental para la creación de riquezas. (1) La empresa termoeléctrica Lidio Ramón Pérez está considerada una entidad proa en el uso de estas Tecnologías en nuestro país, siendo prueba fehaciente de esto el alto control de sus recursos, así como el satisfactorio cumplimiento de su objeto social. El departamento de contabilidad tiene entre sus diversas responsabilidades el control de los gastos de dicha empresa, proceso éste que se realiza de manera manual suponiendo un enorme gasto de tiempo y recursos humanos y restando brillantes a la política vanguardista de la institución en el tema tecnología. Con esta investigación se propone la informatización del proceso de control de la cuenta de gastos en CUC con la implementación de una aplicación web más abierta, entendible, e independiente de la plataforma en que se trabaje, con el objetivo de adecuarse a los cambios de uso de esta cuenta, así como su supervivencia en caso de migraciones de plataforma o actualizaciones de la que se utiliza en la entidad. Para su realización se desarrolló un estudio sobre las aplicaciones web y las herramientas para la elaboración de las mismas. Se utilizó el entorno de desarrollo Dreamweaver, como servidor de aplicaciones web y servidor de bases de datos el paquete tecnológico XAMPP, así como los lenguajes PHP, HTML y CSS. En este documento se expone el estudio realizado y se presenta la metodología de Ingeniería de Software XP que se siguió para el desarrollo del software que se propone como solución de la problemática encontrada.

**Palabras clave:** gastos, acápites, informatización, proceso

## Abstract

Information's Technologies and Telecommunications are not more than a means to transmit and trying to obtain data, information and knowledge, knowledge it is fundamental factor for the creation of riches. (1) The thermoelectric company Lidio Ramón Pérez is once an entity was considered prow in the use of these Technologies at our country, being authentic proof of this the loud control of his resources as well as the satisfactory fulfillment of his corporate purpose. The accounting department is able to deal with between its various responsibilities the aforementioned company's expenses, I process this that comes true of manual way supposing time's enormous expense and human resources and subtracting diamonds to the avant-garde policy of the institution in the theme technology. More Web opened, understandable, proposes the computerization of the control process of the expense account in CUC with the implementation of an application itself with this investigation and independent of the platform in that it be worked up, for the sake of making suitable the changes of use of this account as well as his survival in case of migrations of platform or bringing ups to date of the one that is utilized at the entity. A study on applications developed Web and the tools for the elaboration of the same itself for his realization. Dreamweaver, like servant utilized the surroundings of development himself of applications Web and servant of bases of data the technological parcel XAMPP, as well as the languages PHP, HTML and CSS. The realized study is exposed in this document and XP that continued to for the development of the software that is offered like solution of the found problems encounters Ingeniería's methodology of Software.

**Key words:** Expenses, paragraph, computerization, process

## Contenido

Resumen.....	V
Abstract .....	VI
Introducción .....	1
Capítulo1: Fundamentación teórica.....	4
1.1    Estado del arte .....	4
1.1.1    Informatización del proceso de gestión de las transacciones bancarias de la cuenta de gastos	4
1.1.2    Flujo actual de los procesos .....	4
1.1.3    Análisis de la ejecución de los procesos.....	5
1.2    Antecedentes .....	5
1.2.1    Visual FoxPro .....	5
1.2.2    TimeTracker.....	6
1.2.3    Conta Hogar.....	7
1.2.4    TurboCash .....	<b>¡Error! Marcador no definido.</b>
1.2.5    Mis Cuentas Claras(Ciclope) .....	8
1.2.6    Mint.....	8
1.3    Propuesta de solución.....	8
1.3.1    Tendencias y tecnologías actuales .....	8
1.3.2    Lenguajes de programación .....	15
1.3.3    Metodologías de desarrollo .....	17
1.3.4    Metodología propuesta para el desarrollo de la aplicación. ....	26
1.3.5    Patrones Arquitectónicos.....	27
1.3.6    Herramientas a utilizar en la propuesta de solución .....	31
Capitulo2: Planeación y diseño .....	34
Introducción .....	34
2.1    Funcionalidades Generales .....	34
2.1.1    Personal relacionado con el sistema.....	34
2.1.2    Funcionalidades del sistema .....	35
2.1.3    Diagrama de casos de uso del sistema.....	37
2.1.4    Características del sistema .....	37
2.1.5    Historias de usuario.....	38
2.1.6    Tarjetas CRC .....	39
2.2    Conclusiones del Capítulo .....	39

Capítulo3: Desarrollo y pruebas.....	41
3.1    Modelo de datos .....	41
3.2    Diagramas de diseño (UML) .....	43
3.2.1    Diagramas de clases del diseño.....	43
3.2.2    Diagramas de secuencia .....	43
3.2.3    Diagrama de componentes .....	44
3.3    Interfaces de usuario.....	45
3.4    Desarrollo de las iteraciones .....	46
3.4.1    Tareas por historias de usuario .....	46
3.4.2    Tareas de ingeniería .....	48
3.5    Pruebas.....	48
3.5.1    Pruebas de aceptación .....	49
3.6    Conclusiones del Capítulo .....	50
Capítulo 4: Estudio de factibilidad .....	51
4.1    Introducción .....	51
4.2    Factibilidad técnica.....	51
4.2.1    Hardware.....	51
4.2.2    Software .....	52
4.3    Factibilidad Económica.....	53
4.3.1    Evaluación de Costo Beneficio .....	53
4.3.2    Costos en Moneda Libremente Convertible .....	56
4.3.3    Costos en Moneda Nacional: .....	57
4.4    Conclusiones del capítulo.....	60
Conclusiones Generales .....	61
Recomendaciones .....	61
Referencias Bibliográficas .....	62
Bibliografía .....	64
Glosario de Términos .....	66
Anexo 1: Historias de usuarios .....	71
Anexo 2: Tarjetas CRC .....	76
Anexo 3: Tareas de ingeniería .....	79
Anexo 4: Pruebas de Aceptación.....	89
Anexo 5: Diagramas de clases .....	94
Anexo 6: Diagramas de secuencia.....	98

Anexo 7: Diagramas de componentes .....	105
Anexo 8: Interfaces de usuario.....	109

# INTRODUCCIÓN

## Introducción

En la actualidad el uso de las nuevas tecnologías es un punto clave para la eficiencia empresarial en cualquier institución sea cual fuere el objeto social al que este destinada, la correcta implementación de políticas que tributen al desarrollo tecnológico de las empresas, así como la informatización y automatización de éstas puede ser crucial en el avance productivo y determinante a la hora de valorar la rentabilidad o sostenibilidad de cualquier centro productivo o de servicio. La empresa Termoeléctrica Lidio Ramón Pérez es una entidad clave en la estabilidad económica y social de nuestro país, cuya misión es la generación de energía eléctrica para la red nacional. Los gastos de esta entidad, así como el pago de estímulos y dieta alimenticia son realizados por el departamento de contabilidad desde la cuenta en divisas que habilita la UNE periódicamente por fondos del estado o por ingresos propios. La gestión de los cheques y transacciones que justifican el débito de esta cuenta se hace cada vez más ineficiente por la lentitud en que se procesan inversamente proporcional a la cantidad en que estos aparecen, por este motivo ha surgido la necesidad de desarrollar un Sistema de Gestión de la Información que agilice este proceso. Los Sistemas de Gestión de Información, permiten la gestión de los recursos de información tanto internos como externos. Su finalidad es generar servicios y productos que respondan a las necesidades y sobrepasen las expectativas de los usuarios, posibilitando que el sistema trabaje eficientemente y económicamente a la vez. El Sistema de Gestión de Información aprovecha al máximo sus recursos de información en función de la mejora continua y de la toma de decisiones organizacional a todos los niveles jerárquicos desde la cúspide estratégica hasta la base operativa. Mediante la gestión se proporcionan los recursos de información necesarios para una buena toma de decisiones, se desarrollan nuevos conocimientos que posibilitan calidad y eficiencia en los servicios y productos de las organizaciones. En el mencionado departamento se gestionan numerosos cheques y transferencias manipulando así la enorme cantidad de información que estos generan. La **situación problemática** que se identifica es que el proceso de gestión de la información sobre los cheques que emite la entidad justificados en la cuenta de gastos se vuelve tedioso y lento posibilitando la ocurrencia de pérdidas de información, además las consultas deben realizarse manualmente, en documentos Word y reportes realizados previamente lo que supone un enorme gasto de tiempo y

## INTRODUCCIÓN

recursos humanos y materiales para la entidad. A partir de esta situación se identifica el **problema científico** ¿Cómo favorecer la eficiencia en la gestión de la información de las transacciones bancarias de la cuenta de gastos de la CTE Lidio Ramón Pérez? En función de darle solución surge el presente proyecto que tiene como **objeto de estudio** La informatización de los procesos de gestión de la información de transacciones bancarias y abarca el **campo de acción** de la aplicación web que permita la gestión de la información de transacciones bancarias de la Empresa termoeléctrica Lidio Ramón Pérez. Para darle solución al problema se traza como **objetivo general** Implementar una aplicación web que permita la gestión de la información de las transacciones bancarias que se realizan en la cuenta de gastos de la empresa Termoeléctrica Lidio Ramón Pérez

Para darle solución al objetivo general se trazan las siguientes

### Tareas de investigación:

1. Realizar un levantamiento sobre los procesos que se lleva a cabo en el departamento de contabilidad de la Dirección Económica de la Empresa.
2. Estudiar el proceso de control y manipulación de la información referente a las transacciones que se realizan mediante la cuenta de gastos en la Empresa con la especialista encargada.
3. Seleccionar las tecnologías, herramientas y la metodología de desarrollo de software a utilizar.
4. Identificar los requerimientos del sistema.
5. Elaborar el diseño de la BD del sistema propuesto.
6. Implementar las funciones que deben tener según los requisitos que debe cumplir el software.
7. Realizar el estudio de factibilidad del sistema propuesto.
8. Confeccionar el Manual de Usuario.

De esta forma la **idea a defender** plantea que La aplicación web para la gestión de la información de las transacciones bancarias sobre la cuenta de gastos de la CTE Lidio Ramón Pérez, mejorará el acceso a la información, agilizará el proceso de trabajo y permitirá independencia de plataforma.

Los **métodos empíricos** utilizados son:

## INTRODUCCIÓN

1. Entrevista no estructurada a expertos: para recopilar información necesaria para valorar la situación actual del problema, lo cual permitió identificar los principales requisitos del sistema.
2. Revisión y análisis de documentos: para la recopilación de información y la realización las consultas bibliográficas.

Los **métodos teóricos** utilizados son:

1. Análisis y síntesis: este método se utiliza para desglosar el problema en partes o subproblemas, para de esta forma comprobar el funcionamiento de los mismos, luego integrarlo todo para corroborar las relaciones entre estas y su integración como un todo llegando así a una mejor solución, también para arribar a conclusiones de la investigación.
2. Histórico – Lógico: se puso de manifiesto durante el proceso de estudio de herramientas y metodologías de desarrollo de software a utilizar así como en el estudio de los antecedentes de la investigación.

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

## Capítulo1: Fundamentación teórica

En el presente capítulo se presenta una descripción general del escenario donde se desarrolla el objeto de estudio, así como el campo de acción en que se trabaja. Se ofrece una mejor visión sobre las herramientas a utilizar. Constituye la base teórica para la comprensión del trabajo que se desarrolla. Se abordarán conceptos asociados a la metodología seleccionada para el desarrollo del software.

### 1.1 Estado del arte

#### 1.1.1 Informatización del proceso de gestión de las transacciones bancarias de la cuenta de gastos

El perfeccionamiento del control de la información es una tarea fundamental en todas las instituciones, pues a través del análisis de los datos se puede llegar a conclusiones que ayuden a mejorar el desempeño de la empresa, así como a la toma de decisiones en determinados campos. Conocer y almacenar los datos de las transacciones bancarias es de gran importancia, pues así se puede entender cómo se está comportando la eficiencia económica de la institución y permite mantener los gastos por debajo de los ingresos además de hacer planeaciones para el futuro en base a lo pasado. En base al perfeccionamiento de este proceso se ha despertado en la Empresa Termoeléctrica Lidio Ramón Pérez la necesidad de informatizar dicho proceso mediante una aplicación web.

#### 1.1.2 Flujo actual de los procesos

En el departamento de economía se lleva el control de los gastos incurridos por la empresa, así como el estado actual de la cuenta de gastos, La Unión Eléctrica de Cuba debita la cuenta y hace llegar a la especialista el cheque emitido. En la mencionada recae la responsabilidad de almacenar la información referente a e este y a todos los demás cheques que sean emitidos por la empresa en debito de esta cuenta. Primeramente, debe guardar los datos dígame número de cheque, beneficiario, sdi, saldo entre otros y luego debe realizar con estos datos reportes como el estado financiero mensual, la declaración jurada, el libro del mayor entre otros los cuales se realizarán en documentos Word, además de realizar tantas

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

consultas como le solicite el personal de dirección del Dpto. de economía a los datos guardados.

### 1.1.3 Análisis de la ejecución de los procesos

Los factores o causas que afectan el proceso de gestión de la información de las transacciones bancarias han sido identificados mediante técnicas de observación directa y entrevistas a la económica que atiende este proceso.

#### Información

Causas identificadas

- Dificultad en el manejo de información.
- Incompatibilidad de Sistema Operativo.

## 1.2 Antecedentes

En la empresa existe un Software bastante primitivo, basado en una aplicación de escritorio sobre Visual FoxPro encargado de gestionar los datos de las transacciones bancarias que trabaja de una forma mecánica y hace bastante engorrosa la interacción software-cliente. Las ventajas, desventajas y la mayoría de las características que se pueden decir de dicha aplicación no van más allá de las que ofrece el lenguaje en que fue desarrollado, en el epígrafe 1.2.1 se mencionan algunas de ellas. También existen diversas aplicaciones cuyo objetivo es el control de los gastos de personas o pequeñas empresas que no cumplen con todas los requerimientos que necesita el Dpto. de contabilidad de la CTE Lidio Ramón Pérez, en los epígrafes del 1.2.2 al 1.2.6 se mencionan algunos de los más conocidos.

### 1.2.1 Visual FoxPro

Es un lenguaje de programación orientado a objetos y procedural, un Sistema Gestor de Bases de datos o Database Management System (DBMS), y desde la versión 7.0, un Sistema administrador de bases de datos relacionales, producido por Microsoft. (17)

#### **Características**

Visual FoxPro ofrece a los desarrolladores un conjunto de herramientas para crear aplicaciones de bases de datos para el escritorio, entornos cliente/servidor, tablet PC o para la Web. (17)

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Entre sus características se pueden enumerar:

- Capacidades poderosas y muy veloces para el manejo de datos nativos y remotos.
- Flexibilidad para crear todo tipo de soluciones de bases de datos.
- Lenguaje de programación Orientado a objetos.
- Utilización de sentencias SQL en forma nativa.
- Poderoso manejo de vistas y cursores y control completo de estructuras relacionales.
- Su propio gestor de base de datos incorporado. Sin embargo, también puede conectarse con servidores de base de datos, tales como Oracle, Microsoft SQL Server o MySQL.
- Cuenta con un motor de generación de informes renovado y muy flexible para soluciones más robustas.
- Desde la versión 9.0, amplio soporte de XML, tanto como fuente de datos (por ej., servicios Web basados en XML) como por generar reportes en formato XLM.

### **Ventajas:**

- Menor costo
- Entorno Windows (Es más familiar y predecible)

### **Desventajas:**

- Integridad
- Seguridad
- Menor capacidad de procesamiento

### **1.2.2 TimeTracker**

Es una herramienta de gestión de gastos que permite mantener el control de sus gastos. El sistema no se limita a una entrada de línea, pero integra todos los procesos de administración y análisis. Es una herramienta de administración de proyectos que le permite mantener el control de su negocio. La administración de sus proyectos se hace de manera intuitiva a través de una interfaz clara y ágil. Puede crear sus propios proyectos según sus propias especificaciones. Para cada proyecto, podrá establecer

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

tareas o actividades. Integra el proceso de aprobación de gastos. Se identifica un aprobador quien será responsable de revisar y aprobar los gastos de sus empleados. El aprobador tendrá acceso a toda la información necesaria que le ayudará a tomar una decisión. Puede aceptar o rechazar los gastos. (14)

### **TimeTracker puede:**

- Registrar gastos y asociarlos con sus proyectos y tareas
- Identificar el tipo de gasto (costos km, franqueo, viajes etc.)
- Envíe sus gastos para su aprobación
- Identificar un aprobador
- Establecer las escalas de los gastos y asociarlos con sus empleados
- Establecer presupuestos diarios, semanales y mensuales por tipo de gasto
- Definir el tipo de reembolso por tipo de gasto
- Enviar correos electrónicos y notificaciones automáticamente (configurable)
- Identificar otra moneda de pago y permite la conversión automática a la moneda de la empresa.
- Incluir un nivel de aprobación de reembolso
- Analizar e imprimir sus gastos (14)

### **1.2.3 Hogar Conta**

Es un software gratuito, en español, que puede ser utilizado en diversas plataformas de PC. Es un completo sistema contable con instrucciones sencillas, con gran adaptabilidad para diversas tareas hogareñas en relación a los gastos y su planificación. Es un programa muy sencillo, que cuenta con tutoriales específicos para todas sus aplicaciones. (15)

### **1.2.4 Turbo Cash**

posee licencia gratuita e ilimitada, se ofrece en español, y soporta plataformas de PC de todo tipo. Su funcionamiento, aunque sencillo, se orienta a funciones complejas, y resulta más apropiado para pequeños emprendimientos laborales, familiares o comerciales. También es el más adecuado para hogares con gastos complejos y

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

variables, incluyendo opciones de programación para acreedores, deudores y otros, además de gastos y presupuestos. (15)

### 1.2.5 Mis Cuentas Claras (Ciclope)

Es otro ejemplo de software de licencia gratuita e ilimitada, en español, y con plataforma para sistema operativo Windows XP, Vista y 7. Es uno de los softwares más sencillos en su uso, y ha sido diseñado pensando en los gastos cotidianos de cualquier hogar, incluyendo impuestos y servicios entre otros, contando con la posibilidad de determinar gastos fijos mensuales exhibidos en gráficos. (15)

### 1.2.6 Mint

Es una de las herramientas en línea más utilizadas por los usuarios para llevar la economía doméstica, con funciones muy similares a las de Ready For Zero. Este servicio también permite añadir las cuentas corrientes de los usuarios, siempre que sean de bancos internacionales o de Estados Unidos. (16)

## 1.3 Propuesta de solución

Después de analizada la problemática presentada se propone darle solución, mediante el desarrollo de una aplicación Web que garantice la eficiencia en la gestión de información, la integridad de los datos, el fácil manejo de estos, así como la seguridad de los mismos y agilice la emisión de reportes contables sobre las transacciones bancarias justificadas en la cuenta de gastos de la Empresa Termoeléctrica Lidio Ramón Pérez.

### 1.3.1 Tendencias y tecnologías actuales

#### 1.3.1.1 Aplicaciones Web

En la ingeniería de software se denomina **aplicación web** a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web (*HTML, JavaScript, Java, etc.*) en la que se confía la ejecución al navegador. (2)

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Aunque existen muchas variaciones posibles, una aplicación web está normalmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web ofrece la primera capa, un motor capaz de usar alguna tecnología web dinámica constituye la capa intermedia. Por último, una base de datos constituye la tercera y última capa. (2)

El navegador web manda peticiones a la capa intermedia que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario. (2)

### 1.3.1.1.1 Ventajas

- **Compatibilidad multiplataforma:** Las aplicaciones Web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo *Java*, *Flash*, *ASP* y *Ajax* permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales. (3)
- **Actualización:** Las aplicaciones basadas en Web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo, pues no se hace necesario iniciar nuevas descargas y/o procedimientos de instalación (algunas veces imposible cuando usted está trabajando dentro de grandes organizaciones). (3)
- **Inmediatez de acceso:** Las aplicaciones basadas en Web no necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online y están listas para trabajar sin importar cuál es su configuración o su hardware. (3)
- **Menos requerimientos de memoria:** Las aplicaciones basadas en Web tienen menos demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del proveedor, esas aplicaciones basadas en Web usan en muchos casos la memoria de las computadoras donde ellas corren, dejando más espacio para correr múltiples aplicaciones sin incurrir en frustrantes deterioros en el rendimiento. (3)
- **Menos Bugs (errores de programación de software):** Las aplicaciones basadas en Web son menos propensas a colgarse y crear problemas técnicos

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones basadas en Web, todos utilizan la misma versión, y todos los *bugs* pueden ser corregidos tan pronto como son descubiertos. (3)

- **Precio:** Las aplicaciones basadas en Web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional. Esto permite que las aplicaciones online cuesten una fracción de sus contrapartes descargables y no totalmente gratuitas, mientras que ofrecen componentes adicionales y servicios Premium como una opción. (3)
- **Múltiples usuarios concurrentes:** Las aplicaciones basadas en Web pueden ser utilizadas por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta. (3)
- **Los datos son más seguros:** Si bien la ruptura de discos no va a desaparecer, es probable que los usuarios escuchen mucho menos del tema. A medida que las compañías se hagan cargo del almacenamiento de los datos del usuario, granjas de almacenamiento de datos redundantes, altamente fiables, los usuarios van a tener mucho menos riesgo de perder sus datos debido a una ruptura de disco impredecible o a un virus de la computadora. Las compañías que proveen aplicaciones basadas en Web van a brindar amplios servicios de resguardo de datos ya sea como una parte integral del servicio básico o como una opción paga. (3)

### 1.3.1.1.2 ¿Por qué una aplicación web?

Las aplicaciones Web son de un desarrollo poco costoso, sencillo y rápido. Presentan acceso ubicuo, sin necesidad de distribución e idealmente con pocos requerimientos técnicos. Con datos centralizados y fácil integración de datos de múltiples fuentes. Por los aspectos antes expuestos se ha decidido usar la tecnología web para el desarrollo del sistema.

#### 1.3.1.2 Servidores web

Un servidor Web es un programa que implementa el protocolo HTTP (*Hypertext Transfer Protocol*). Este protocolo está diseñado para transferir lo que llamamos

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

hipertextos, páginas Web o paginas HTML (*Hypertext Markup Language*), textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproducciones de sonidos. (4)

Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y este le responde con el contenido que el cliente solicita. Entre los servidores Web más utilizados se encuentran *Microsoft Internet Information Server*, *Microsoft Personal Web Server*, *Apache HTTP Server*, *Netscape Enterprise Server* y *Sun ONE Web Server*. (4)

**Apache** es el servidor Web más difundido y utilizado en Internet debido a su alta estabilidad, seguridad, facilidad de expansión y por su costo ya que es software libre, lo que significa que se distribuye por medio de una licencia gratuita y de código abierto; además se encuentra disponible para una gran cantidad de sistemas operativos, tiene como fin crear un servidor HTTP (Web) robusto, de nivel comercial, con innumerables características y de distribución gratuita. (4)

**IIS** (*Internet Information Server*), es el servidor web estándar del sistema operativo Windows NT/2000/XP, ofrece varios servicios de Internet: servidor Web (o HTTP), servidor FTP y servidor SMTP. (4)

### 1.3.1.2.1 ¿Por qué utilizar Apache?

- 1) Corre sobre una multitud de plataformas y Sistemas Operativos.
- 2) Ofrece tecnología libre y de código abierto, otorgándole una transparencia y dando la posibilidad de conocer que es lo que realmente estamos instalando.
- 3) Es un servidor Web altamente configurable y de diseño modular, capaz de ampliar su funcionalidad y calidad de servicios.
- 4) Trabaja en conjunto con gran cantidad de Lenguajes de Programación interpretados como PHP (*PHP Hypertext Pre-processor*), Perl, soporte con CGI (*Common Gateway Interface*), Java, JSP(*Java Server Pages*) y otros lenguajes de script, el complemento ideal para los sitios web dinámicos que vemos en la actualidad.

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

- 5) Es posible configurar y personalizar cada uno de los mensajes de error que se pueden producir por la utilización del servidor. (4)

### 1.3.1.3 Sistema de gestión de bases de datos

Los sistemas de gestión de base de datos (SGBD); en inglés: *Database Management System*, abreviado DBMS son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. (5)

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 SGBD disponibles en el mercado se encuentran: PostgreSQL, MySQL, Firebird, SQLite, Oracle, Microsoft SQL Server, entre otros. (5)

#### 1.3.1.3.1 PostgreSQL

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. (5)

#### **Características**

Alta concurrencia.

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos. (5) Alta variedad de tipos nativos.

PostgreSQL provee nativamente soporte para: (5)

1. Números de precisión arbitraria.
2. Texto de largo ilimitado.
3. Figuras geométricas (con una variedad de funciones asociadas).
4. Direcciones IP (IPv4 e IPv6).
5. Bloques de direcciones estilo CIDR.

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

6. Direcciones MAC.

7. Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. (8)

Otras características:(5)

8. Claves ajenas también denominadas Llaves ajenas o Claves Foráneas (foreign keys).

9. Disparadores (triggers): Un disparador o trigger se define como una acción específica que se realiza de acuerdo a un evento, cuando éste ocurra dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica.

10. Vistas.

11. Integridad transaccional.

12. Herencia de tablas.

13. Tipos de datos y operaciones geométricas.

14. Soporte para transacciones distribuidas. Permite a PostgreSQL integrarse en un sistema distribuido formado por varios recursos (p. ej, una base de datos PostgreSQL, otra Oracle, una cola de mensajes IBM MQ JMS y un ERP SAP) gestionado por un servidor de aplicaciones donde el éxito ("commit") de la transacción global es el resultado del éxito de las transacciones locales.

### **Desventajas**

Sin experiencia, configurar llega a ser un caos.

Es fácil de vulnerar sin protección adecuada.

El motor MyISAM es instalado por defecto y carece de capacidades de integridad relacional.

InnoDB genera mucho footprint en memoria al indizar.

El toolset empresarial tiene un costo adicional por suscripción anual.

Realizar revisiones llegar a ser una labor manual y tediosa para el Administrador de la Base de Datos (DBA).

#### 2.1.1.1.1 MySQL

Es una base de datos relacional de libre distribución con potencia, usada principalmente como soporte de portales web que necesitan manejar datos

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

dinámicamente, es una alternativa a los grandes servidores de bases de datos basados en Microsoft SQL Server y Oracle. MySQL se destaca especialmente por su velocidad al efectuar lecturas y búsquedas en la base de datos. (2)

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. (2)

Es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido. (2)

MySQL cuenta con muchas **ventajas**, entre las que se encuentran las siguientes: (2)

- Alto rendimiento
- Portabilidad
- MySQL software es Open Source.
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet.
- El software MySQL usa la licencia GPL (*General Public License*). (MySQL: Ventajas)

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

### Requisitos de hardware

MySQL usa menos recursos del hardware. Para instalar MySQL, se necesita 32 Mb de RAM y casi cerca de 60Mb de espacio en disco duro. MySQL no requiere los recursos del CPU adicionales. (2)

#### 1.3.1.3.2 ¿Por qué MySQL?

MySQL es muy rápido, fiable y fácil de usar. Es un sistema multiplataforma de base de datos relacionales, lo que da velocidad y flexibilidad, cuenta con un sistema de privilegios contraseñas, muy seguro que permite la autenticación básica para el acceso al servidor. MySQL es un sistema de administración de base de datos. Opera en una arquitectura cliente/servidor. Es el sistema gestor de bases de datos “*open source*” más popular, o sea que puede ser bajado de Internet y usarlo sin tener que pagar, además que cualquiera puede estudiar su código y adecuarlo a las necesidades que requiera.

Luego de analizadas las características y facilidades del SGBD presentado, se decide usar el MySQL como SGBD, por las siguientes razones:

- Alto rendimiento y velocidad al realizar las operaciones.
- El PHP maneja más fácil al MySQL que al SQL Server, debido a la gran cantidad de funciones que tiene explícitas.
- El MySQL es multiplataforma.
- El MySQL no tiene precio en el mercado, se adquiere libremente.

### 1.3.2 Lenguajes de programación

En el dominio de la red, los lenguajes del lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, PHP, entre otros. (9)

#### PHP

(*Hypertext Preprocessor*), es un lenguaje interpretado de alto nivel embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, con numerosas características de PHP específicas. La meta de este lenguaje es permitir escribir a los

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

creadores de páginas web, páginas dinámicas de una manera rápida y fácil. PHP (Hypertext Preprocessor), es un lenguaje interpretado de alto nivel embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, con numerosas características de PHP específicas. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil. ASP (Active Server Pages), desarrollado por Microsoft para funcionar junto a su servidor IIS, es un lenguaje poderoso sobre todo si se va a utilizar herramientas de Microsoft en conjunto como SQL Server. (9)

### PHP vs ASP

#### Velocidad

Uno de los principales problemas en los portales de Internet que soportan una gran carga de usuarios es la velocidad de respuesta de su servidor, por lo que este debería ser un punto crítico al elegir un lenguaje u otro. ASP, probablemente, nunca será tan rápido como PHP. ASP está construido en una arquitectura de objetos COM, y por lo tanto, cuando un programador de ASP usa VBScript realmente está corriendo un objeto COM. Si escribe un simple texto en la salida estándar, estaría llamando al objeto Response del método Write. Si accede a una base de datos, debe usar otro objeto, etc. Todo esto sobrecarga y hace más lento el sistema. En cambio, al usar PHP todas las operaciones se realizan en el espacio de memoria de PHP. Esto significa que el código PHP ejecutará más rápido ya que no debe cargar dinámicamente ningún otro objeto en memoria en la ejecución de un programa.

Costes Ocultos Una de las cosas que se suelen pasar por alto al usar un software determinado son los costes ocultos o derivados de la primera compra. ASP cumple con esta especificación, es decir, un desarrollador posiblemente no pueda realizar todas las tareas que necesita con las opciones que ofrece ASP por defecto; si necesita encriptación comprará ASPEncrypt, si necesita gestión de correo comprará Qmail de ServerObject, por ejemplo, si necesita subir ficheros (file uploading) al servidor comprará SA-FileUP de Artisans. Todo esto y más están disponibles en PHP gratuitamente (9)

**PHP vs ASP vs Perl** con respecto a su ejecución en sistemas operativos.

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

PHP es un lenguaje multiplataforma. Un script PHP normal puede ejecutarse sin cambiar ni una sola línea de código en cualquier servidor que interprete PHP, es decir, en servidores Windows, Linux etc. ASP, por contra, es un lenguaje propietario de Microsoft y solo puede ejecutarse en servidores Microsoft. Perl si puede ejecutarse bajo Windows y Unix, pero antes hay que realizar labores de adaptación de cada script al concreto sistema operativo. (9)

### 1.3.2.1.1 ¿Por qué usar PHP?

Es un lenguaje de programación usado generalmente en la creación de contenido para sitios o aplicaciones web. La versión 5 de PHP permite y facilita la reutilización de código entre otras facilidades.

Ventajas de trabajar con PHP comparado con otros lenguajes similares: (9)

- Es un lenguaje multiplataforma.
- Rapidez de ejecución.
- Mantiene un bajo consumo de recursos de máquina.
- Gran seguridad, muy poca probabilidad de corromper los datos.
- Capacidad de conexión con la mayoría de los manejadores de bases de datos que se emplean en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Posee una amplia documentación en internet, incluyendo una gran variedad de ejemplos y de ayudas.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de programación orientada a objetos.
- Permite crear formularios para la web.
- No requiere definición de tipos de variables ni manejo detallado de bajo nivel.

### 1.3.3 Metodologías de desarrollo

Una metodología para el desarrollo de un proceso de software es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas informáticos. Las

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

metodologías existentes en la actualidad se dividen en dos grandes grupos atendiendo a sus características: las metodologías tradicionales (RUP, MSF) y las metodologías ágiles (XP, SCRUM). Las primeras están pensadas para el uso exhaustivo de documentación durante todo el ciclo del proyecto mientras que las segundas ponen vital importancia en la capacidad de respuesta a los cambios, la confianza en las habilidades del equipo y al mantener una buena relación con el cliente. (2)

Este último enfoque ha mostrado su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, no obstante, mantiene una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales. (2)

### *1.3.3.1 Scrum*

Scrum es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software. Aunque Scrum estaba enfocado a la gestión de procesos de desarrollo de software, puede ser utilizado en equipos de mantenimiento de software, o en una aproximación de gestión de programas. (11)

Es un método adaptativo de gestión de proyectos que se basa en los principios ágiles: (11)

- Colaboración estrecha con el cliente. Predisposición y respuesta al cambio
- Prefiere el conocimiento tácito de las personas al explícito de los procesos
- Desarrollo incremental con entregas funcionales frecuentes
- Comunicación verbal directa entre los implicados en el proyecto
- Motivación y responsabilidad de los equipos por la auto-gestión, auto-organización y compromiso.
- Simplicidad.
- Supresión de artefactos innecesarios en la gestión del proyecto.

#### **Roles**

En Scrum se definen varios roles, estos están divididos en dos grupos: cerdos y

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

gallinas. El nombre de los grupos está inspirado en la siguiente frase: 1 De esta forma, los “cerdos” están comprometidos a desarrollar el software de forma regular y frecuente, mientras que todos los demás son 'gallinas' que sólo interesados en el proyecto, y si esta falla, ellos no son los cerdos, es decir, ellos no fueron los que se comprometieron a hacerlo. Las necesidades, deseos, ideas e influencias de los roles “gallina” se tienen en cuenta, pero no de forma que pueda afectar, distorsionar o entorpecer el proyecto Scrum. (11)

### Roles "Cerdo"

#### Propietario del Producto

El Propietario del Producto representa la voz del cliente. Se asegura de que el equipo Scrum trabaja de forma adecuada desde la perspectiva del negocio. Sus áreas de responsabilidad son: (11)

- Financiación del producto.
- Requisitos del sistema.
- Retorno de la inversión del proyecto.
- Lanzamiento del proyecto.

### Equipo

Responsable de transformar la pila del sprint (Sprint Backlog) en un incremento de la funcionalidad del software.

- Auto-gestionado.
- Auto-organizado.
- Multi-funcional.

### Scrum Manager

Responsable del proceso Scrum.

- Formación y entrenamiento del proceso.
- Incorporación de Scrum en la cultura de la empresa.
- Garantía de cumplimiento de roles y responsabilidad.

### Roles "Gallina"

Los roles gallina en realidad no son parte del proceso Scrum, pero deben tenerse en cuenta. Un aspecto importante de una aproximación ágil es la práctica de involucrar en el proceso a los usuarios, expertos del negocio y otros interesados (stakeholders).

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Es importante que esa gente participe y entregue retroalimentación con respecto a la salida del proceso a fin de revisar y planear cada sprint.

### **Análisis de la frase "Rol gallina":**

La gallina alimenta al proyecto "poniendo huevos", no se ve comprometida como el cerdo que va al matadero.

### **Usuarios**

Es el destinatario final del producto. Como bien lo dice la paradoja, El árbol cae en el bosque cuando no hay nadie ¿Hace ruido? Aquí la definición sería Si el software no es usado ¿fue alguna vez escrito?

### **Stakeholders (Clientes, Proveedores, Inversores)**

Se refiere a la gente que hace posible el proyecto y para quienes el proyecto producirá el beneficio acordado que lo justifica. Sólo participan directamente durante las revisiones del sprint.

### **Managers**

Es la gente que establece el ambiente para el desarrollo del producto.

### **Sprint**

Es el periodo de tiempo durante el que se desarrolla un incremento de funcionalidad. Constituye el núcleo de Scrum, que divide de esta forma el desarrollo de un proyecto en un conjunto de pequeñas "carreras".

- Duración máxima: 30 días.
- Durante el sprint no se puede modificar el trabajo que se ha acordado en el
- Backlog.

Sólo es posible cambiar el curso de un sprint, abortándolo, y sólo lo puede hacer el Scrum Master si decide que no es viable por alguna de las razones siguientes:

- ✓ La tecnología acordada no funciona.
- ✓ Las circunstancias del negocio han cambiado.
- ✓ El equipo ha tenido interferencias.

Artefactos

### **Pila de producto (Product Backlog)**

Listado con los requisitos del sistema

Es responsabilidad del dueño del producto

- ✓ Contenido

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

- ✓ Priorización
- ✓ Disponibilidad

Nunca llega a ser una lista completa y definitiva

El empleado para planificar el proyecto es sólo una estimación inicial de requisitos

Es un documento dinámico que incorpora constantemente las necesidades del sistema

Se mantiene durante todo el ciclo de vida (hasta la retirada del sistema)

### **Pila del producto**

Pila del Sprint (Sprint backlog)

Trabajo o tareas determinadas por el equipo para realizar en un sprint y lograr al final del mismo un incremento de la funcionalidad. Se recomienda que las tareas reflejadas tengan una duración comprendida entre las 4 y las 16 horas de trabajo. Las de mayor duración deben intentar descomponerse en sub-tareas de ese rango de tiempo. (11)

### ***1.3.3.2 Programación Extrema (Extreme Programming, XP)***

XP es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, que describe la filosofía de XP en el "Manifiesto Ágil", sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. (9)

#### **1.3.3.2.1 Valores que promueve XP**

Cuatro valores que promueven la metodología XP:

**Simplicidad:** XP propone el principio de hacer las cosas más simples que puedan funcionar, en relación al proceso y la codificación. Es mejor hacer hoy algo simple, que hacerlo complicado y probablemente nunca usarlo mañana. (10)

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

**Comunicación:** Algunos problemas en los proyectos tienen su origen en que alguien no dijo algo importante en algún momento. XP hace imposible la falta de comunicación. (10)

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

**Coraje:** El coraje (valor) existe en el contexto de los otros 3 valores. (10)

### 1.3.3.2.2 El ciclo de vida ideal de XP

El ciclo de vida ideal de XP consta de seis fases:

- Exploración.
- Planificación de la Entrega (Reléase)
- Iteraciones.
- Producción.
- Mantenimiento.
- Muerte del Proyecto.

### 1.3.3.2.3 Prácticas en las que se fundamenta XP

#### **Planificación incremental**

La programación extrema asume que la planificación nunca será perfecta, y que variará en función de cómo varíen las necesidades del negocio. Por tanto, el valor real reside en obtener rápidamente un plan inicial, y contar con mecanismos de retroalimentación que permitan conocer con precisión dónde se está. Como es lógico, la planificación es iterativa: un representante del negocio decide al inicio de cada iteración qué características concretas se van a implementar. (10)

El objetivo de XP es generar versiones de la aplicación tan pequeñas como sea posible, pero que proporcionen un valor adicional claro, desde el punto de vista del negocio. A estas versiones se les denomina *releases*. (10)

Un *release* cuenta con un cierto número de historias. La historia es la unidad de funcionalidad en un proyecto XP, y corresponde a la mínima funcionalidad posible que tiene valor desde el punto de vista del negocio. Durante cada iteración se cierran varias historias, lo que hace que toda iteración añada un valor tangible para el cliente. Gran parte de la eficacia de este modelo de planificación se deriva de una división clara de responsabilidades, que tiene en cuenta las necesidades del negocio en todo

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

momento. Dentro de esta división, el representante del cliente tiene las siguientes responsabilidades: (10)

- Decidir qué se implementa en cada release o iteración.
- Fijar las fechas de fin de un release, recortando unas características o añadiendo otras.
- Priorizar el orden de implementación, en función del valor de negocio.
- Las responsabilidades del equipo de desarrollo son las siguientes:
- Estimar cuánto tiempo llevará una historia de usuario, esto es fundamental para el cliente, y puede llevarle a reconsiderar qué historias se deben incluir en una iteración.
- Proporcionar información sobre el coste de utilizar distintas opciones tecnológicas.
- Organizar el equipo.
- Estimar el riesgo de cada historia.
- Decidir el orden de desarrollo de historias dentro de la iteración.

### Testing

La ejecución automatizada de tests es un elemento clave de la metodología XP. Existen tanto tests internos (o tests de unidad), para garantizar que el mismo es correcto, como tests de aceptación, para garantizar que el código hace lo que debe hacer. El cliente es el responsable de definir los tests de aceptación, no necesariamente de implementarlos. (10)

Él es la persona mejor calificada para decidir cuál es la funcionalidad más valiosa.

Un efecto lateral importante de los tests, es que dan una gran seguridad a los desarrolladores; es posible llegar a hacer cambios más o menos importantes sin miedo a problemas inesperados, dado que proporciona una red de seguridad. La existencia de tests hace al código muy maleable. (10)

### Diseño simple

Una práctica fundamental de la programación extrema es utilizar diseños tan simples como sea posible. El principio es "utilizar el diseño más sencillo que consiga que todo funcione". La metodología XP nos pide, que no se viva bajo la ilusión de que un diseño puede resolver todas o gran parte de las situaciones futuras. XP define un "diseño tan simple como sea posible" aquel que pasa todos los tests, no contiene código duplicado, deja clara la intención de los programadores (enfatisa el qué, no el

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

cómo) en cada línea de código y contiene el menor número posible de clases y métodos. (10)

### **Propiedad colectiva del código**

XP aboga por la propiedad colectiva del código. En otras palabras, todo el mundo tiene autoridad para hacer cambios a cualquier código, y es responsable de ellos. Esto permite no tener que estar esperando a otros cuando todo lo que hace falta es algún pequeño cambio. (10)

### **Integración continua.**

En muchos casos la integración de código produce efectos laterales imprevistos, y en ocasiones esta puede llegar a ser realmente traumática, cuando dejan de funcionar cosas por motivos desconocidos. La programación extrema hace que la integración sea permanente, con lo que todos los problemas se manifiestan de forma inmediata. (10)

### **Clientes en el equipo**

Algunos de los problemas más graves en el desarrollo son los que se originan cuando el equipo toma decisiones de negocio críticas. Esto no debería ocurrir, pero en el momento cumbre, con frecuencia no se obtiene retroalimentación del cliente con la fluidez necesaria. (10)

La metodología XP intenta resolver este tipo de problemas integrando un representante del negocio dentro del equipo de desarrollo. Esta persona siempre está disponible para resolver dudas y para decidir qué se hace en cada momento, en función de los intereses del negocio. (10)

### **Entregas pequeñas**

Siguiendo la política de la metodología XP, de dar el máximo valor posible en cada momento, se intenta liberar nuevas versiones de las aplicaciones con frecuencia. Estas deben ser tan pequeñas como sea posible, aunque deben añadir suficiente valor, para que resulten meritorias para el cliente. (10)

### **Semana de 40 horas**

La programación extrema lleva un modo de trabajo en el que el equipo siempre está al 100%. Una semana de 40 horas en las que se dedica la mayor parte del tiempo a tareas que suponen un avance puede dar mucho de sí, y hace innecesario recurrir a sobreesfuerzos, excepto en casos extremos. Además, el esfuerzo continuado pronto lleva a un rendimiento menor y a un deterioro de la moral de todo el equipo. (10)

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

### 1.3.3.2.4 Fases de la Metodología XP

#### **Fase I:** Planificación (10)

- Se escriben historias de usuario, cuya idea principal es describir un caso de uso en dos o tres líneas con terminología del cliente (de hecho, se supone que deben ser escritos por el mismo), de tal manera que se creen test de aceptación para historias de usuarios (user storie) y permita hacer una estimación de tiempo de desarrollo del mismo.
- Se crea un plan de lanzamiento (release planning), que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto. Se usa como base las historias de usuario, participando el cliente en la elección de las que se desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto.
- El desarrollo se divide en iteraciones, cada una de las cuales comienzan con un plan de iteración, para el que se eligen las historias de usuario a desarrollar y las tareas de desarrollo.
- Se cambia el proceso cuanto sea necesario, para adaptarlo al proyecto. (10)

#### **Fase II:** Diseño (10)

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

#### **Fase III:** Codificación (10)

- El cliente está siempre disponible, de ser posible, cara a cara. La idea es que forme parte del equipo de desarrollo, y esté presente en todas las fases de XP. La idea es usar el tiempo del cliente para estas tareas en lugar de crear una detallada especificación de requisitos, y evitar la entrega de un producto insuficiente, que le hará perder tiempo.
- El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código.
- Las pruebas unitarias se codifican antes que el código en sí, haciendo que la

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

codificación de este último sea más rápida, y que cuando se afronte la misma se tenga más claro, qué objetivos tiene que cumplir lo que se va a codificar.

- La programación del código se realiza en parejas, para aumentar la calidad del mismo. En cada momento, sólo habrá una pareja de programadores que integre código.
- Se integra código y se lanza dicha integración de manera frecuente, evitando divergencias en el desarrollo y permitiendo que todo el mundo trabaje con la última versión del desarrollo. De esta manera, se evitará pasar grandes períodos de tiempo integrando el código al final del desarrollo, ya que las incompatibilidades serán detectadas enseguida.
- Se usa la propiedad colectiva del código, lo que se traduce en que cualquier programador puede cambiar cualquier parte del código. El objetivo es fomentar la contribución de ideas por parte de todo el equipo de desarrollo.
- Se deja la optimización para el final.
- No se hacen horas extra de trabajo.

### **Fase IV: Pruebas (10)**

- Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado.
- Cuando se encuentra un error de codificación o bug, se desarrollan pruebas para evitar volver a caer en el mismo.
- Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las historias de usuarios (HU) elegidas para la iteración, y son "pruebas de caja negra", en las que el cliente verifica el correcto funcionamiento de lo que se está probando. Cuando se pasa la prueba de aceptación, se considera que la correspondiente historia de usuario se ha completado. (10)

### **1.3.4 Metodología propuesta para el desarrollo de la aplicación.**

En este trabajo, se decide utilizar la metodología XP ya que es ideal para grupos de desarrollo relativamente pequeños y donde el tiempo disponible para la entrega del proyecto es corto; como es el caso de la presente investigación. Además de ser de fácil realización y requiere poco trabajo documental por lo que se hace más cómoda

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

su utilización. Actualmente éste es el método ágil más documentado (hay una colección de libros con XP Series de Addison Wesley) y extendido. Existe una gran comunidad de desarrolladores XP. Otra de las ventajas de XP es que no es necesario adoptarlo en forma completa, sino que pueden utilizarse varias de sus prácticas en forma independiente. Esto hace que el costo de su implementación sea mucho más accesible que el de otras metodologías. (10)

### 1.3.4.1.1 Ventajas

#### Ventajas de XP (10)

- Apropriado para entornos volátiles.
- Puede ser implementado en forma parcial (elegir sólo algunas de las prácticas) o en forma gradual.
- Puede adaptarse a las necesidades de cualquier equipo de desarrollo.
- Estar preparados para el cambio, significa reducir su coste.
- Permite definir en cada iteración cuáles son los objetivos de la siguiente.
- Exige que se establezca una comunicación más fluida con el cliente y que este tenga mayor participación en el proceso de desarrollo. La consecuencia de esto es que el cliente se involucre más en el desarrollo del producto.
- Permite la retroalimentación.
- Actualmente es la metodología ágil más extendida y documentada.
- Se realizan pruebas constantemente del sistema

### 1.3.5 Patrones Arquitectónicos

Los patrones arquitectónicos, o patrones de arquitectura, son patrones de diseño de software que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen una escalera más grande. (12)

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Aunque un patrón arquitectónico comunica una imagen de un sistema, no es una arquitectura como tal. Un patrón arquitectónico es más un concepto que captura elementos esenciales de una arquitectura de software. Muchas arquitecturas diferentes pueden implementar el mismo patrón y por lo tanto compartir las mismas características. Además, los patrones son a menudo definidos como una cosa "estrictamente descrita y comúnmente disponible". Por ejemplo, la arquitectura en capas es un estilo de llamamiento-y-regreso, cuando define uno un estilo general para interactuar. Cuando esto es descrito estrictamente y comúnmente disponible, es un patrón. (12)

Uno de los aspectos más importantes de los patrones arquitectónicos es que encarnan diferentes atributos de calidad. Por ejemplo, algunos patrones representan soluciones a problemas de rendimiento y otros pueden ser utilizados con éxito en sistemas de alta disponibilidad. A primeros de la fase de diseño, un arquitecto de software escoge qué patrones arquitectónicos mejor ofrecen las calidades deseadas para el sistema. (12)

### *1.3.5.1 Modelo - Vista - Controlador*

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. (13)

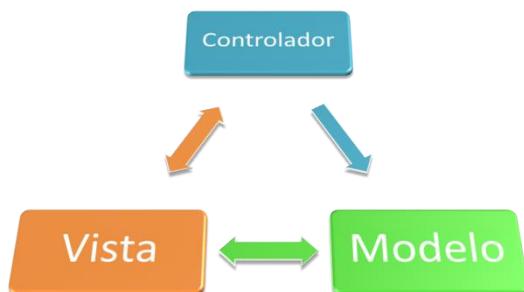
#### **Descripción**

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

- Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- Controlador: Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista. (13)



**Figura 1.5.1 Patrón Arquitectura Modelo Vista Controlador**

Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en líneas generales del MVC corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí. (13)

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente: (13)

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz- vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (*handler*) o *callback*.

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta relación entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. Este uso del patrón Observador no es posible en las aplicaciones Web puesto que las clases de la vista están desconectadas del modelo y del controlador. En general el controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista. Por ejemplo, en el MVC usado por Apple en su framework Cocoa. Suele citarse como Modelo-Interface-Control, una variación del MVC más puro

La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

### 1.3.5.1.1 ¿Por qué MVC?

Proponemos como patrón arquitectónico a utilizar en nuestro proyecto el MVC porque este separa Modelo de la Vista, es decir, separa los datos de la representación visual de los mismos, crea independencia de funcionamiento, facilita el mantenimiento en caso de errores y permite el escalamiento de la aplicación en caso de ser requerido.

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

### 1.3.6 Herramientas a utilizar en la propuesta de solución

Luego del estudio realizado se llega a la conclusión de que el sistema estará guiado por la metodología de desarrollo XP por sus facilidades, documentación y flexibilidad, además es la que mejor se adapta a las condiciones de desarrollo del sistema propuesto.

#### 1.3.6.1 XAMPP

Es un paquete formado por un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl. El nombre proviene de X (para cualquier sistema operativo), A (Apache), M (MySQL), P (PHP), P (Perl). El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X. (9)

XAMPP es regularmente actualizado para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como Open SSL, y PhpMyAdmin. (2) Para instalar XAMPP requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar programas por separado. (9)

Oficialmente, los diseñadores de XAMPP solo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios webs y programadores testear su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica, sin embargo, XAMPP es utilizado actualmente para servidor de sitios webs en WWW, y con algunas modificaciones es generalmente lo suficientemente seguro para serlo. Una herramienta especial es suministrada para proteger fácilmente las partes más importantes del paquete. (9)

#### 1.3.6.2 Embarcadero ER/Studio

Es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejos. (14)

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Ayuda a las organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia y alcanza en última instancia usos de más alta calidad que entreguen datos más eficientes y exactos a la empresa. ER/Studio está equipado para crear y manejar diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de bases de datos, documentación y fácil creación de reportes. (14)

### *1.3.6.3 Dreamweaver*

#### **Adobe Dreamweaver**

es una aplicación en forma de suite (basada en la forma de estudio de Adobe Flash) que está destinada a la construcción, diseño y edición de sitios y aplicaciones Web basados en estándares. Creado inicialmente por Macromedia (actualmente producido por Adobe Systems) es el programa más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium. Su principal competidor es Microsoft Expression Web y tiene soporte tanto para edición de imágenes como para animación a través de su integración MALVERDE'S Hasta la versión MX, fue duramente criticado por su escaso soporte de los estándares de la web, ya que el código que generaba era con frecuencia sólo válido para Internet Explorer y no validaba como HTML estándar. Esto se ha ido corrigiendo en las versiones recientes.

La gran ventaja de este editor sobre otros es su gran poder de ampliación y personalización del mismo, puesto que en este programa, sus rutinas (como la de insertar un hipervínculo, una imagen o añadir un comportamiento) están hechas en Javascript-C, lo que le ofrece una gran flexibilidad en estas materias. Esto hace que los archivos del programa no sean instrucciones de C++ sino rutinas de JavaScript que hace que sea un programa muy fluido, que todo ello hace, que programadores y editores web hagan extensiones para su programa y lo ponga a su gusto.

## CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Las versiones originales de la aplicación se utilizaban como simples editores WYSIWYG. Sin embargo, versiones más recientes soportan otras tecnologías web como CSS, JavaScript y algunos frameworks del lado servidor.

Dreamweaver permite al usuario utilizar la mayoría de los navegadores Web instalados en su ordenador para previsualizar las páginas web. También dispone de herramientas de administración de sitios dirigidas a principiantes como, por ejemplo, la habilidad de encontrar y reemplazar líneas de texto y código por cualquier tipo de parámetro especificado, hasta el sitio web completo. El panel de comportamientos también permite crear JavaScript básico sin conocimientos de código.

Con la llegada de la versión MX, Macromedia incorporó herramientas de creación de contenido dinámico en Dreamweaver. En lo fundamental de las herramientas HTML WYSIWYG, también permite la conexión a Bases de Datos como MySQL y Microsoft Access, para filtrar y mostrar el contenido utilizando tecnología de script como, por ejemplo, ASP (Active Server Pages), ASP.NET, ColdFusion, JSP (Java Server Pages) y PHP sin necesidad de tener experiencia previa en programación.

Un aspecto de alta consideración de Dreamweaver es su funcionalidad con extensiones. Es decir, permite el uso de "Extensiones". Las extensiones, tal y como se conocen, son pequeños programas, que cualquier desarrollador web puede escribir (normalmente en HTML y Javascript) y que cualquiera puede descargar e instalar, ofreciendo así funcionalidades añadidas a la aplicación. Dreamweaver goza del apoyo de una gran comunidad de desarrolladores de extensiones que hacen posible la disponibilidad de extensiones gratuitas y de pago para la mayoría de las tareas de desarrollo web, que van desde simple efectos rollover hasta completas cartas de compra.

También podría decirse, que para un diseño más rápido y a la vez fácil podría complementarse con fireworks en donde podría uno diseñar un menú o para otras creaciones de imágenes (gif web, gif websnap, gif adaptable, jpeg calidad superior, jpeg archivo más pequeño, gif animado websnap) para un sitio web y después exportar la imagen creada y así utilizarla como una sola, en donde ya llevara los vínculos a un dicho sitio en específico que uno le haya dado. (MRR - U.P.)

### Capitulo2: Planeación y diseño

#### Introducción

En el presente capítulo se realiza la fase de planeación y diseño, de la metodología de desarrollo de software XP, la cual se empleó para la realización del presente trabajo, donde se determinan las necesidades del cliente, y se describen las funcionalidades que se automatizarán mediante el empleo de las historias de usuarios (HU). Además, se presentará un plan de iteraciones donde se asignará cada una de las historias de usuario, aplicando una estimación de esfuerzo de las mismas. Se presentan las llamadas tarjetas CRC y sus principales, para hacer más detallado el resultado de esta fase serán utilizados artefactos de otra metodología ágil que en estrecha correspondencia con la elegida inicialmente la complementa en pos de facilitar el trabajo tal como son el diagrama de casos de uso del sistema y la descripción de los casos de uso del sistema.

#### 2.1 Funcionalidades Generales

La gestión de la información referida a las transacciones bancarias sobre la cuenta de gastos en divisas en la Empresa Termoeléctrica Lidio Ramón Pérez se realiza de forma manual ingresando los datos en documentos Word de los cuales se toman los datos para de manera trabajosa y sumamente tediosa realizar los reportes de estado de cuentas mensuales y anuales, así como otros que pida la dirección de economía de la empresa. De la descripción anterior se identifican como funcionales generales del sistema, guardar los datos de las transacciones bancarias así como generar los estados resultado y la declaración jurada de cada mes, así como los cierres mensuales.

##### 2.1.1 Personal relacionado con el sistema

<b>Personas relacionadas con el sistema</b>	<b>Justificación</b>
Administrador	Tiene bajo su responsabilidad la introducción de nuevos usuarios y la

	actualización de la lista de personas que interactúan con el software así como la verificación de los datos introducidos por el resto de los usuarios.
Esp. "C" en gestión Económica	Es la encargada del control y la gestión de los datos referentes a las transacciones bancarias.
Superadministrador	Es el encargado de la gestión de los administradores del sistema.
Desarrollador	Es la persona responsable de llevar a cabo la implementación del sistema.

Tabla 2.1.1: Personal relacionado con el sistema.

### 2.1.2 Funcionalidades del sistema

Posterior al conocimiento de las personas que interactúan con el sistema se analizarán las funcionalidades que este debe cumplir para realizar las expectativas de los mismos. De acuerdo a lo antes expuesto el sistema debe ser capaz de:

1. Autenticar usuario	
2. Gestionar usuario	<ul style="list-style-type: none"> <li>2.1. Insertar usuario</li> <li>2.2. Modificar usuario</li> <li>2.3. Eliminar usuario</li> <li>2.4. Mostrar usuario</li> </ul>
3. Gestionar persona	<ul style="list-style-type: none"> <li>3.1. Insertar persona</li> <li>3.2. Modificar persona</li> <li>3.3. Eliminar persona</li> <li>3.4. Mostrar persona</li> </ul>
4. Gestionar datos generales	<ul style="list-style-type: none"> <li>4.1. Insertar datos generales</li> <li>4.2. Modificar datos generales</li> <li>4.3. Eliminar datos generales</li> <li>4.4. Mostrar datos generales</li> </ul>
5. Gestionar cheque	<ul style="list-style-type: none"> <li>5.1. Insertar cheque</li> <li>5.2. Eliminar cheque</li> <li>5.3. Mostrar cheques</li> </ul>

6. Gestionar transferencia	6.1 Insertar transferencia 6.2 Eliminar transferencia 6.3 Mostrar transferencia
7. Gestionar comisión bancaria	7.1 Insertar comisión Bancaria 7.2 Eliminar comisión Bancaria 7.3 Mostrar comisión Bancaria
8. Gestionar crédito de la UNE	8.1 Insertar crédito 8.2 Eliminar crédito 8.3 Mostrar crédito
9. Gestionar devolución de tercero	9.1 Insertar devolución 9.2 Eliminar devolución 9.3 Mostrar devolución
10. Gestionar saldo mensual	10.1 Insertar saldo mensual 10.2 Modificar saldo mensual 10.3 Mostrar saldo mensual
11. Gestionar acápite	11.1 Insertar acápite 11.2 Modificar acápite 12.3 Eliminar acápite
12. Gestionar cargo	12.1 Insertar cargo 12.2 Modificar cargo 12.3 Eliminar cargo
13. Gestionar modificaciones	13.1 Insertar modificaciones 13.2 Eliminar modificaciones
14. Generar estado de cuenta mensual	
14. Generar declaración jurada	
15. Generar nomenclador	
16. Generar listado del maestro	

Tabla 1 Funcionalidades del sistema

## CAPÍTULO II: PLANEACIÓN Y DISEÑO

### 2.1.3 Diagrama de casos de uso del sistema

Este artefacto es un modelo de las funciones deseadas para el sistema y su entorno, y sirve como contrato entre el cliente y los desarrolladores. Se utiliza como entrada esencial para las actividades de planeación y diseño.

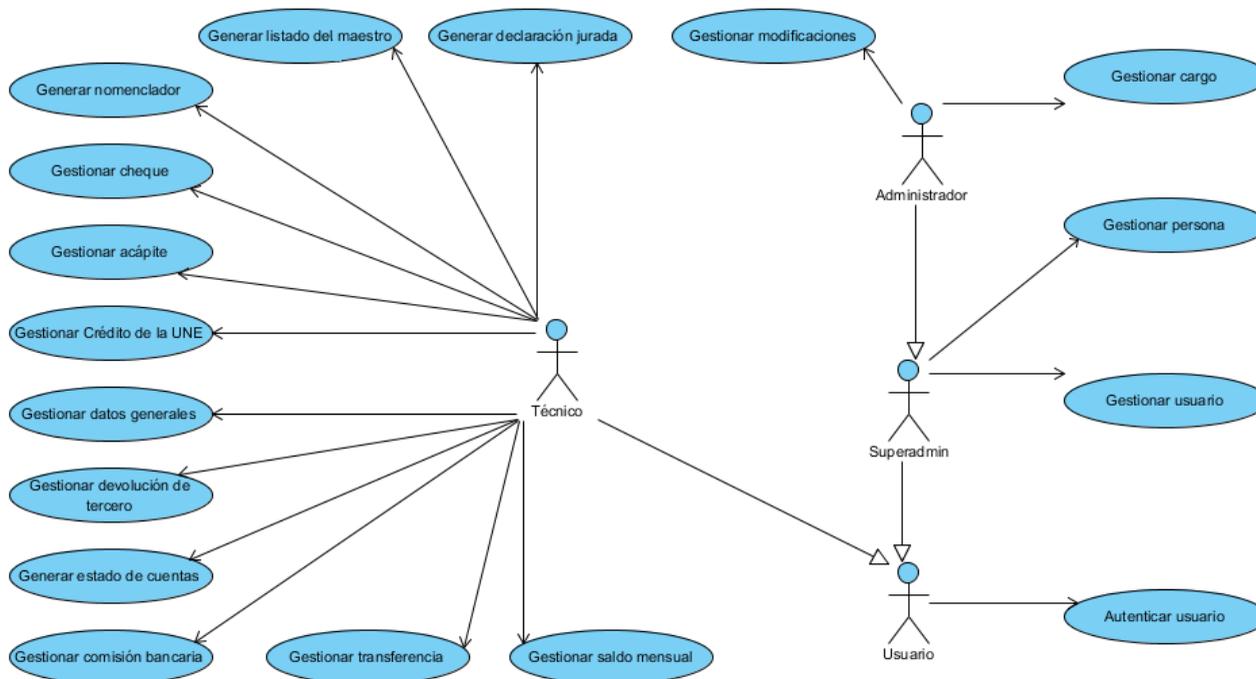


Figura 1 Diagrama de Clases del Diseño

### 2.1.4 Características del sistema

1. Requerimientos de software.
  - a. Navegador que soporte las características del lenguaje HTML5 y CSS3.
  - b. Servidor web Apache 2.x.
  - c. Servidores de bases de datos MySQL 5.x.
2. Requerimientos de Apariencia e interfaz externa.
  - a. Diseño gráfico no debe ser complejo.
  - b. Utilizar colores que proporcionen una interfaz amigable al usuario.
  - c. Utilizar un tamaño de letra legible y que no agote la visión.
3. Requerimientos de usabilidad
  - a. Mostrar en cada página un título con el nombre de la sección correspondiente.

## CAPÍTULO II: PLANEACIÓN Y DISEÑO

- b. Resaltar en todos los formularios los campos obligatorios con un asterisco o con un texto en rojo.
  - c. Agregar etiquetas a los campos de entrada de datos con una breve descripción del formato que admiten y un ejemplo.
4. Requerimientos de seguridad
- a. Se definieron dos roles: Administrador que tiene acceso a crear nuevas cuentas y a gestionar los usuarios, así como de ver las modificaciones y técnico que se encarga de la gestión de las transacciones bancarias.
  - b. Cada usuario diferente deberá acceder a la aplicación por una cuenta personal
  - c. Las modificaciones serán guardadas con la fecha en que se hicieron.
  - d. Verificar la información de todos los campos de entrada de datos, señalar aquellos con datos erróneos y mostrar mensajes que ayuden a corregirlos

### 2.1.5 Historias de usuario

Las historias de usuarios (HU), son la técnica utilizada en XP para detallar los requisitos del software. Son el resultado directo del intercambio entre los usuarios y desarrolladores a través de reuniones donde las conocidas *tormenta de ideas (brain storm)* arrojan no solo los requerimientos, sino también las posibles soluciones; representan una forma rápida de administrar las necesidades de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para gestionarlos, debido a que un requerimiento de software es descrito de forma concreta y sencilla utilizando el lenguaje común del usuario. Las HU permiten responder ágilmente a los requerimientos cambiantes y aunque se redactan desde las perspectivas de los clientes, también los desarrolladores pueden brindar ayuda en la identificación de las mismas. [\[Anexo 1: Historias de usuarios\]](#)

**Tabla 2.1.4.6 HU No. 6 Tramitar Transferencia**

Historia de usuario	
<b>Número: 6</b>	Usuario: Técnico
<b>Nombre:</b> Tramitar Transferencia	

## CAPÍTULO II: PLANEACIÓN Y DISEÑO

<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> Primera
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el usuario ingresa los datos de la Transferencia teniendo la posibilidad de listarlas y cancelarlas	
<b>Observaciones:</b> Confirmado por el cliente.	

### Diseño

En este epígrafe tiene lugar la realización de las tarjetas de clases, responsabilidades y colaboración, conocidas comúnmente como tarjetas CRC, estas se realizan con el objetivo de facilitar la comunicación y documentar los resultados. Además, las mismas permiten la total participación y contribución del equipo de desarrollo en la tarea de diseño.

#### 2.1.6 Tarjetas CRC

Una tarjeta CRC representa un objeto, por tanto, es una clase, cuyo nombre se ubica en forma de título en la parte superior de la tarjeta, los atributos y las responsabilidades más significativas se colocan a la izquierda y las clases implicadas con cada responsabilidad a la derecha, en la misma línea de su requerimiento correspondiente. Para mejor comprensión de las mismas, se determina agruparlas por HU.

A continuación se presenta la tarjeta CRC de la clase Usuario [[Ver Anexo 2: Tarjetas CRC](#)]

**Tabla 2.2.1.3 Tarjeta CRC No. 3**

<b>Nombre de la Clase:</b> Gestionar Cheque	
<b>Tipo de clase:</b>	
Lógica del negocio	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Inserta datos del cheque	insertar_ cheque
Muestra los datos del cheque	mostrar_ cheque
Modifica los datos del cheque	modificar_ cheque
Eliminar del cheque de la base de datos	eliminar_ cheque

## 2.2 Conclusiones del Capítulo

Como resultado de este capítulo se realizó la fase de planeación y diseño en las cuales se definieron las historias de usuarios (HU) con la participación del cliente, se efectuó la



## CAPÍTULO II: PLANEACIÓN Y DISEÑO

planificación de iteraciones de cada HU a partir de la estimación del esfuerzo necesario de las mismas. Se presentaron además las principales clases que se utilizarán en el desarrollo de la aplicación a través de las llamadas tarjetas CRC culminando así esta fase y se determina que el equipo de trabajo está listo para pasar a la siguiente etapa de desarrollo.

### Capítulo3: Desarrollo y pruebas

En este capítulo se siguen las fases de Desarrollo y Pruebas contenidas en la Metodología XP. Se muestra un modelo de datos correspondiente a la aplicación definitiva, así como las interfaces de usuarios diseñadas para la naciente aplicación. Se realiza a través del desarrollo de las iteraciones un desglose de las tareas a realizar para cada una de las HU expuestas en el capítulo anterior. Por último se describen las pruebas realizadas, mostrando las diferentes respuestas al aplicarse cada una de las funcionalidades. En estas se encuentran como encargado el programador en la parte de desarrollo y el cliente en la realización de las pruebas de aceptación.

#### 3.1 Modelo de datos

Después de analizados los requisitos que debe cumplir el sistema y los datos con los que opera, se realizó el modelo de datos, el cual cuenta con las tablas **acceso** y **usuario**, las que almacenarán la información de los usuarios de la aplicación. Entre las tablas más relevantes del modelo se encuentran **cheque** que es la encargada de almacenar los datos de las transacciones bancarias ya sean crédito de la UNE, devolución de tercero, cheque, transferencia o comisión bancaria. También se encuentra la tabla **fondo** que guarda la información del saldo de cada cuenta, la tabla **modificación** que guarda el usuario y la fecha de cada vez que se modifique alguna transacción y la tabla **saldo mensual**, que guarda los saldos de cada cuenta al iniciar cada mes del año, la tabla **persona** que guarda los datos de los trabajadores que interactúan con el sistema y la tabla **acápite** que guardara los acápites a utilizar. Así también utilizaremos la tabla **cargos** que guardara los cargos de las personas que interactúen con el software y **datos generales** que contendrá los datos de la empresa que se utilizaran para conformar los reportes del sistema.

A continuación, mostramos el modelo de datos utilizado en el sistema.

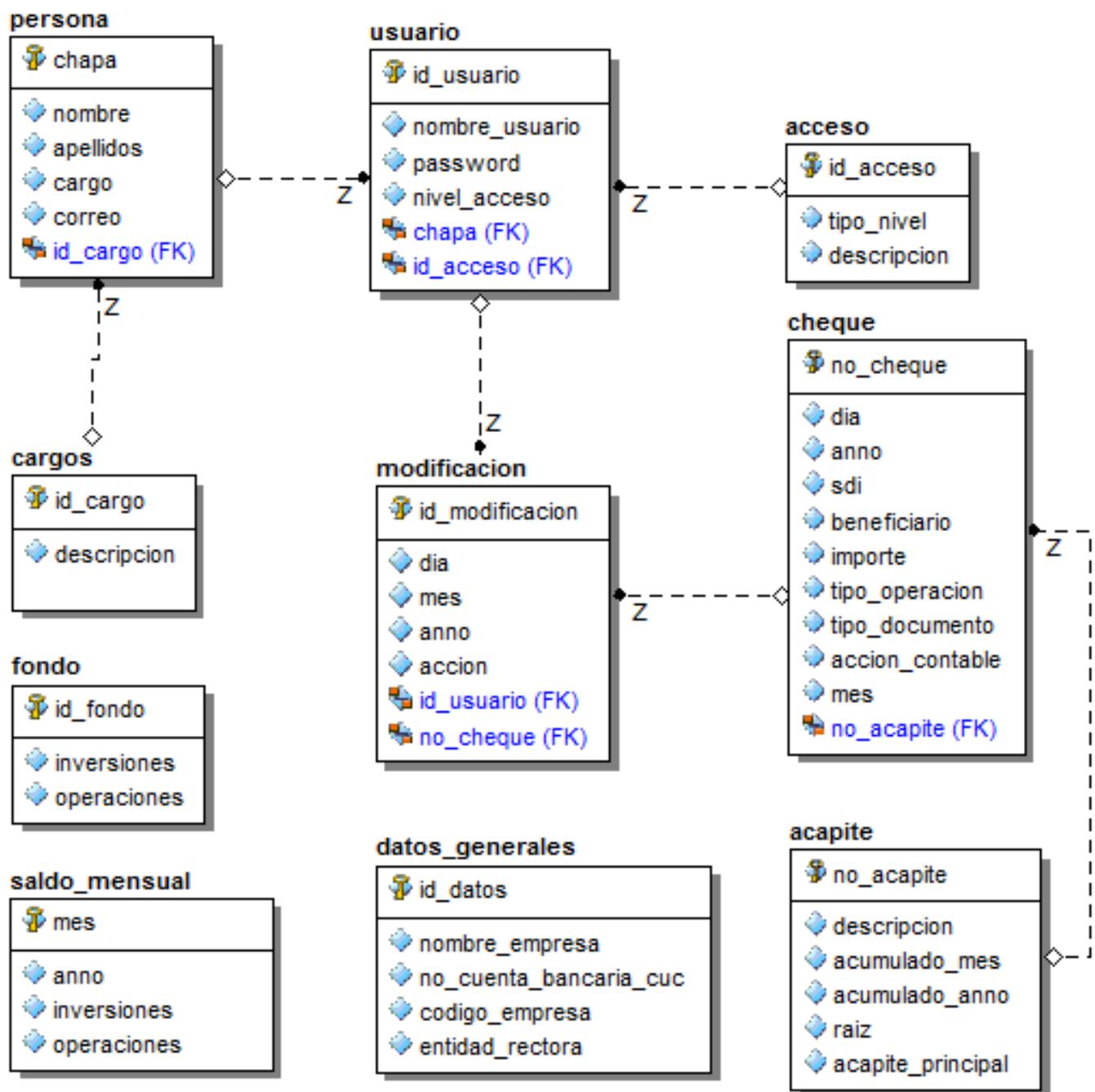


Figura 2 Modelo de datos

## 3.2 Diagramas de diseño (UML)

### 3.2.1 Diagramas de clases del diseño

Este artefacto es un modelo de las funciones deseadas para el sistema y su entorno, y sirve como contrato entre el cliente y los desarrolladores. Se utiliza como entrada esencial para las actividades de diseño, codificación y prueba. [\[Ver Anexo 5 Diagramas de clases\]](#)

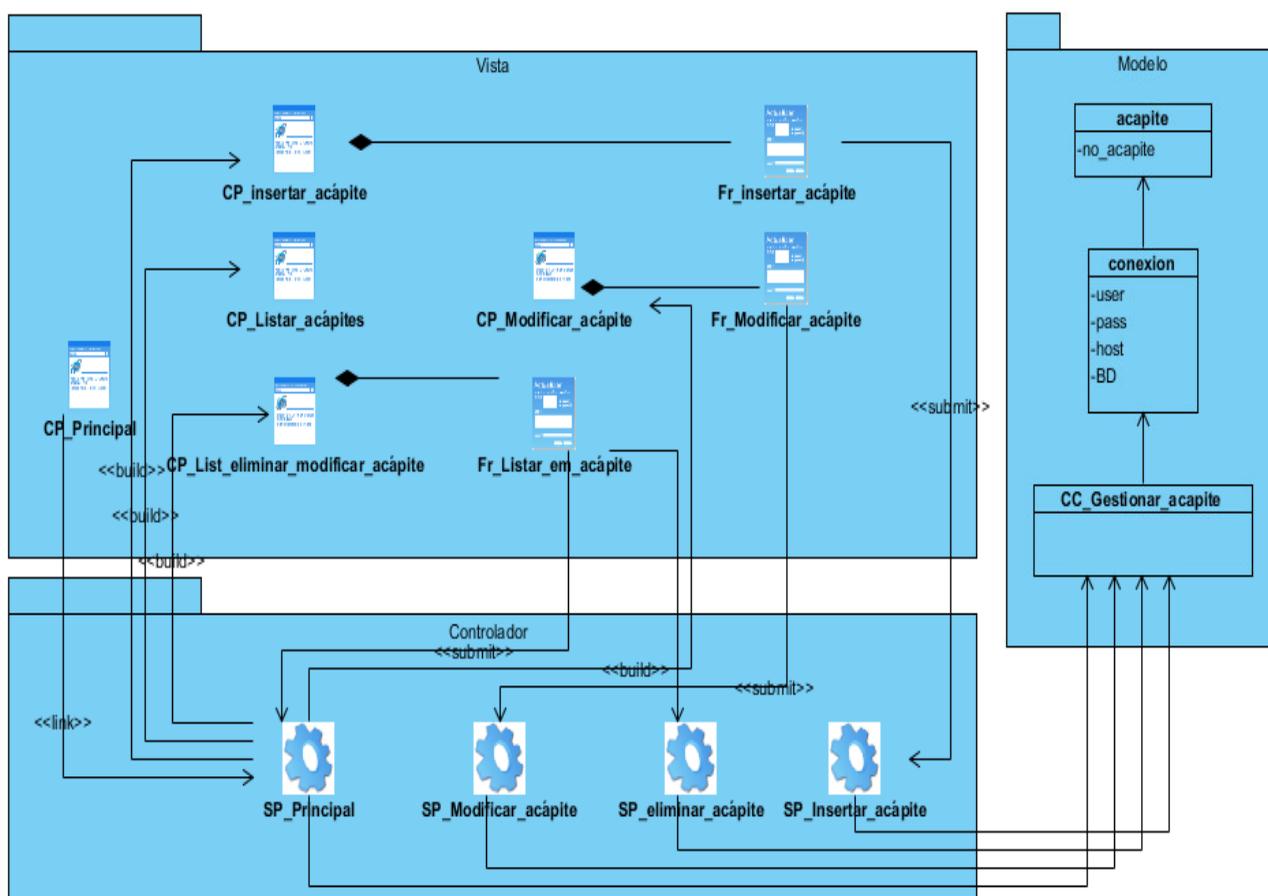


Figura 3 DCD Gestionar Acápites

### 3.2.2 Diagramas de secuencia

Un Diagrama de Secuencia (DS) muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela por escenarios para cada caso de uso. El diagrama de secuencia contiene detalles de implementación del escenario,

## CAPÍTULO III: DESARROLLO Y PRUEBAS

incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos. [\[ver Anexo 6 Diagramas de secuencia\]](#)

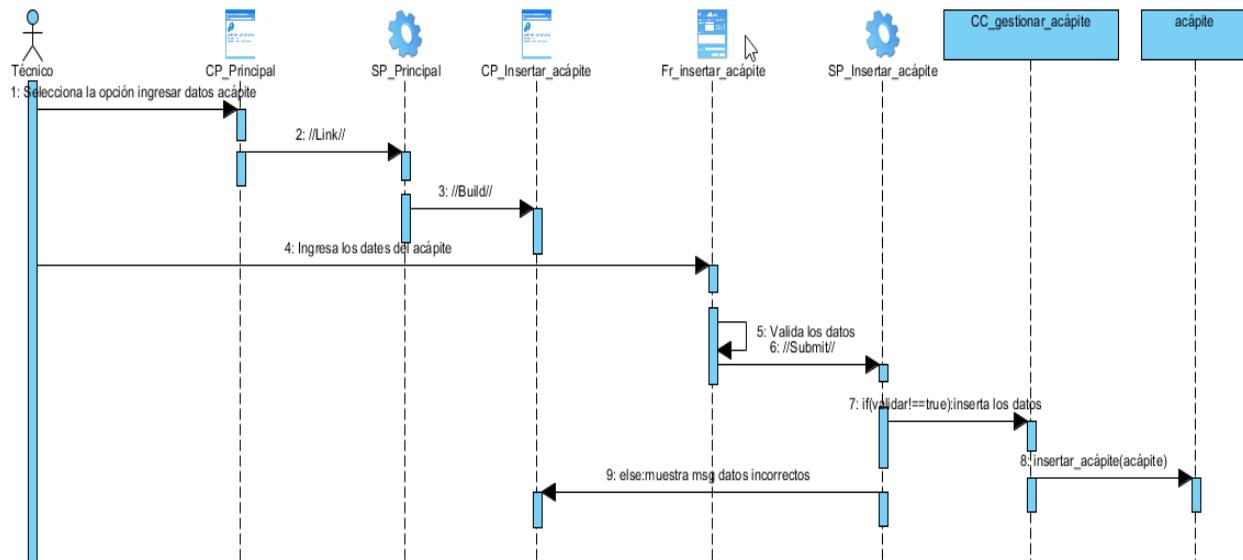


Figura 4 DS Insertar Acápite

### 3.2.3 Diagrama de componentes

Los diagramas de componentes conforman en esta disciplina el modelo de implementación, al describir los componentes a construir, su organización y dependencias. Un componente es una parte física y reemplazable de un sistema que se conforma con un conjunto de interfaces y proporciona la realización de dicho conjunto. Se usan para modelar los elementos físicos que pueden hallarse en un nodo por lo que empaquetan elementos como clases, colaboraciones e interfaces. [\[Ver anexo 7 Diagramas de componentes\]](#)

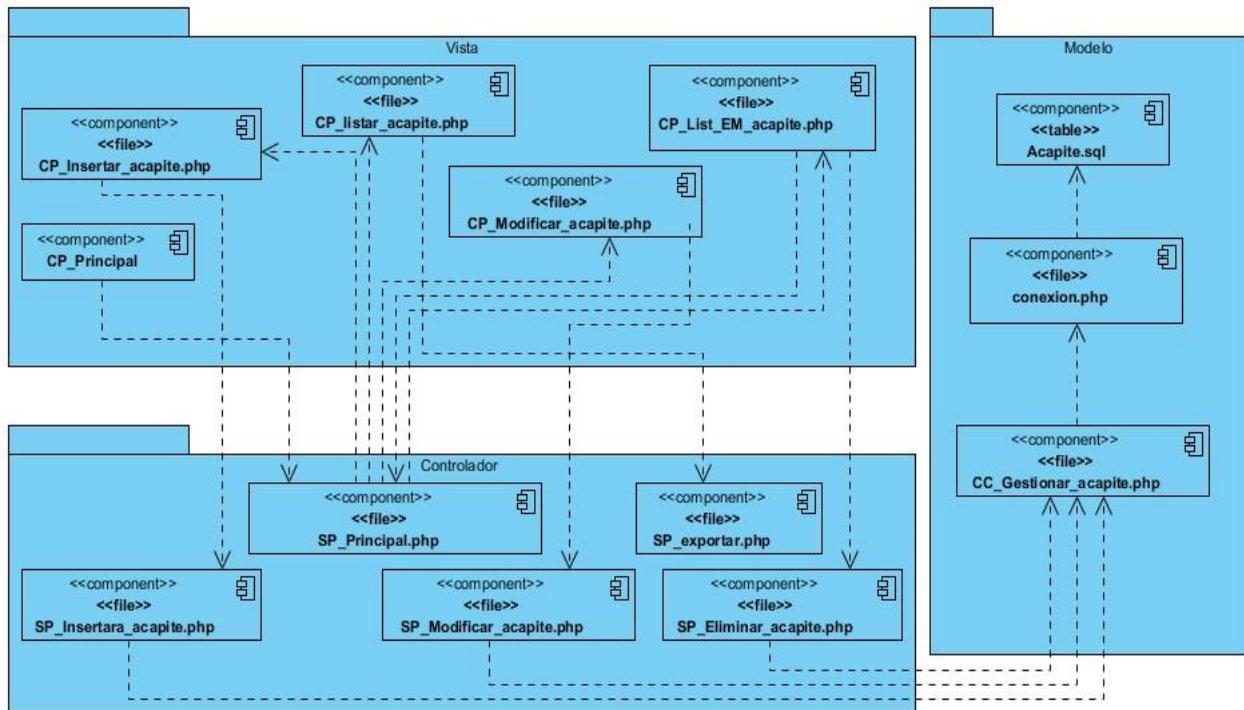


Figura 5 DC Gestionar acápite

### 3.3 Interfaces de usuario

En el capítulo se ejemplifica con la interfaz de usuario correspondiente a autenticar usuario del sistema mostrando un ejemplo de tratamiento de errores, las demás están reflejadas en el [\[Anexo 8: Interfaces de usuario\]](#)



Figura 6 Vista Autenticar usuario

### 3.4 Desarrollo de las iteraciones

En el capítulo anterior se detallaron todas las historias de usuarios de cada una de las iteraciones en la que se va a desarrollar el sistema, y para la realización de cada una de las HU se realizará una revisión al plan de iteraciones, para ver si pueden ocurrir cambios.

#### 3.4.1 Tareas por historias de usuario

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

Historia de usuario	Tareas
Autenticar usuario	Autenticar usuario
Gestionar usuario	Insertar usuario. Mostrar usuario. Modificar usuario. Eliminar usuario.
Gestionar cheque	Insertar cheque. Mostrar cheque. Eliminar cheque.
Gestionar transferencia	Insertar transacción. Mostrar transacción Eliminar transacción
Gestionar saldo mensual	Mostrar saldo mensual. Modificar saldo mensual
Gestionar comisión bancaria	Insertar comisión bancaria. Mostrar comisión bancaria. Eliminar comisión bancaria.
Gestionar devolución de tercero	Insertar devolución Mostrar devolución Eliminar devolución
Gestionar Acápite	Insertar acápite. Mostrar acápite. Modificar acápite. Eliminar acápite.

Gestionar Persona	Insertar Persona. Mostrar Persona. Modificar Persona. Eliminar Persona.
Gestionar Modificación	Mostrar Modificación. Eliminar Modificación.
Gestionar datos generales	Mostrar datos generales. Modificar datos generales
Gestionar cargo	Insertar Cargo. Mostrar Cargo. Modificar Cargo. Eliminar Cargo.
Gestionar crédito de la UNE	Insertar Crédito Mostrar Crédito Eliminar Crédito
Mostrar el estado de saldos mensual	Mostrar el estado de saldos mensual
Mostrar declaración jurada	Mostrar declaración jurada
Mostrar nomenclador	Mostrar nomenclador
Mostrar listado del maestro	

Tabla 3.3 Distribución de tareas por historias de usuario.

### Historias de usuario abordadas en la Iteración No. 1

Historia de usuario	Tiempo de estimación(días)
Autenticar usuario	1
Gestionar usuario	4
Gestionar Cheque	4
Gestionar Saldo mensual	4
Gestionar Acápite	4
Gestionar Persona	4
Gestionar Cargo	4
Gestionar Crédito de la UNE	4

Tabla 3.4 Historias de usuario abordadas en la primera iteración.

### Historias de usuario abordadas en la Iteración No. 2

Historia de usuario	Tiempo de estimación(días)
---------------------	----------------------------

## CAPÍTULO III: DESARROLLO Y PRUEBAS

Gestionar Modificación	3
Gestionar Devolución de tercero	3
Gestionar Transferencia	3
Gestionar Comisión Bancaria	3
Gestionar Datos Generales	3
Generar el estado de cuenta mensual	2
Generar Declaración jurada	1
Generar Nomenclador	1
Generar Listado del Maestro	3

Tabla 3.5 Historias de usuario abordadas en la segunda iteración.

### 3.4.2 Tareas de ingeniería

En XP generalmente cada historia se divide en tareas de ingeniería o tareas de programación. Las tareas no tienen que ser entendidas necesariamente por el cliente, pues las mismas, sólo son utilizadas por los miembros del equipo de desarrollo, por lo que pueden ser escritas en lenguaje técnico. Estas se crean para obtener una mejor planificación y cumplir con las funcionalidades básicas que luego conformaran las funcionalidades generales de cada historia de usuario [[Anexo 3: Tareas de ingeniería](#)].

Tarea de Programación	
No. Tarea: 1	No. Historia: 1
Nombre de la tarea: Autenticar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 22/01/2014	Fecha fin: 22/01/2014
Programador responsable: Orlando Prieto Silva	
Descripción: Permite autenticar los usuarios al sistema.	

Tabla 3.3.2.1 Tarea de programación Autenticar usuario.

### 3.5 Pruebas

En la metodología XP las pruebas juegan un papel fundamental, pues esta permite la comprobación continua del código. El desarrollo constante de las pruebas da lugar a que se desarrolle un software con mayor calidad dando una mayor seguridad de lo que se está haciendo.

En la metodología escogida hay dos tipos de prueba; las unitarias o desarrollo dirigido por pruebas, desarrolladas por los programadores verificando su código de forma automática, y las pruebas de aceptación, las cuáles son evaluadas luego de culminar una iteración verificando así que se cumplió la funcionalidad requerida por el cliente.

## CAPÍTULO III: DESARROLLO Y PRUEBAS

### 3.5.1 Pruebas de aceptación

Las pruebas de aceptación en XP, se pueden asociar con las pruebas de caja negra que se aplican en otras metodologías de desarrollo, sólo que se crean a partir de las historias de usuario y no por un listado de requerimientos. Durante las iteraciones, las HU se traducen a pruebas de aceptación. En ellas se especifican desde la perspectiva del cliente, los escenarios para probar que la HU ha sido implementada correctamente. La misma puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo que persiguen estas pruebas, es garantizar que las funcionalidades solicitadas por el cliente han sido realizadas. Una HU no se considera completa hasta que no ha transitado por sus pruebas de aceptación. Luego de ver los arquetipos anteriores empleados para la realización de las pruebas y reunirse con el cliente para su análisis, el mismo decidió que se lleve a cabo el proceso mediante las pruebas de aceptación.

La planilla utilizada para plasmar el contenido de las pruebas de aceptación se muestra a continuación. [[Anexo 4: Pruebas de Aceptación](#)].

Prueba de aceptación
Historia de usuario # 1: Autenticar usuarios
Nombre: Prueba para comprobar la entrada de un usuario al sistema.
Descripción: Validación de entrada de los datos del usuario.
Condiciones de ejecución: El usuario debe introducir su nombre de usuario y contraseña.
Entrada/Pasos de ejecución: El usuario escribe su nombre de usuario y contraseña y luego pulsa el botón “Entrar”.
Resultado esperado: <ul style="list-style-type: none"> <li>➤ Si el usuario tiene acceso para entrar a la aplicación e inserta sus datos correctamente entrara sin problemas al sistema.</li> </ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"> <li>➤ Se inserten los datos de un usuario no valido para el sistema o datos incorrectos.</li> <li>➤ Se dé clic en el botón Entrar sin insertar nada en los campos de texto.</li> </ul>
Evaluación de la prueba: Aceptada.

Tabla 3.4.1.1. Prueba de aceptación HU #:1 Autenticar usuarios.

### 3.6 Conclusiones del Capítulo

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

### Capítulo 4: Estudio de factibilidad

#### 4.1 Introducción

El presente capítulo propone identificar las causas que ameritan la informatización de estos procesos, a través de un estudio de factibilidad para determinar la infraestructura tecnológica y la capacidad técnica que implica la implantación del sistema en cuestión, así como los costos, beneficios y el grado de aceptación que la propuesta genera en la empresa.

Para que toda estrategia de solución sea aprobada, independientemente del contenido de la misma, debe cumplir con dos componentes de los requerimientos de factibilidad que son: factibilidad técnica y económica.

#### 4.2 Factibilidad técnica.

La Factibilidad Técnica consiste en realizar una evaluación de la tecnología existente en la organización, este estudio está destinado a recolectar información sobre los componentes técnicos que posee la organización y la posibilidad de hacer uso de los mismos en el desarrollo e implementación del sistema propuesto y de ser necesario, los requerimientos tecnológicos que deben ser adquiridos para el desarrollo y puesta en marcha del sistema en cuestión. De acuerdo a la tecnología necesaria para la implantación de la Aplicación Web para el control de la información referente a los cheques emitidos en la CTE Lidio Ramón Pérez, se evaluó bajo los enfoques: Hardware y Software.

##### 4.2.1 Hardware

Según la evaluación del hardware existente, no se requirió realizar inversión inicial para la adquisición de nuevos equipos, ni tampoco para mejorar o actualizar los existentes, ya que los mismos satisfacen los requerimientos establecidos para la puesta en funcionamiento del sistema propuesto. A continuación, se muestran las características de los medios de cómputo con los que cuenta la CTE Lidio Ramón Pérez.

Tabla 2: Características de los medios de cómputo de la CTE Lidio Ramón Pérez

Características de los medios de cómputo	
Servidor	Procesador: Intel Core(TM) i3 CPU 2.67GHz
	Memoria RAM: 4 GB
	Disco Duro: 1 TB
	Unidad de Protección UPS
Estaciones de trabajo	Procesador : Intel Core(TM) i3 CPU 2.67GHz
	Memoria RAM: 4 GB
	Disco Duro: 1 TB
	Unidad de Protección UPS

Todas las estaciones de trabajo están conectadas al servidor a través de una red utilizando cable par trenzado. Esta configuración permite que los equipos instalados puedan interactuar con la Aplicación Web para el control de la información sobre cheques emitidos en la CTE.

#### 4.2.2 Software

La Empresa cuenta con las aplicaciones necesarias para la implantación del sistema, lo cual no requiere inversión alguna para la adquisición de las licencias de las mismas. Las estaciones de trabajo operan en ambiente Windows y tienen instalado un navegador web, única herramienta necesaria para acceder al sistema. El servidor se encuentra instalado sobre una plataforma Windows.

Tabla 3 : Aplicaciones instaladas en las computadoras de la CTE Lidio Ramón Pérez.



Aplicaciones Instaladas	
Servidores	Sistema Operativo Windows 8.1
	Paquete tecnológico XAMPP
Estaciones de trabajo	Sistema Operativo Windows 8.1
	Navegadores Mozilla Firefox

Como resultado de este estudio técnico se determinó que la institución posee la infraestructura tecnológica (Hardware, Software) necesaria para la puesta en funcionamiento de la aplicación desarrollada.

### 4.3 Factibilidad Económica

A continuación, se presenta un estudio que dio como resultado la factibilidad económica en el desarrollo del nuevo sistema de control de dieta. Se determinaron los recursos para desarrollar, implantar y mantener en operación el sistema programado, haciendo una evaluación donde se puso de manifiesto el equilibrio existente entre los costos del sistema y los beneficios que se liberaron de este, lo cual permitió observar de una manera más precisa las bondades del sistema propuesto.

#### 4.3.1 Evaluación de Costo Beneficio

La técnica de análisis de Costo-Beneficio, tiene como objetivo fundamental proporcionar una medida de la rentabilidad de un proyecto, mediante la comparación de los costos previstos con los beneficios (efectividad) esperados en la realización del mismo:

- El costo involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema y los costos de operación asociados.
- La efectividad se entiende como la capacidad o facultad para lograr un objetivo o fin deseado, que se han definido previamente, y para el cual se han desplegado acciones estratégicas para llegar a él, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad de cumplimiento del objetivo). Este puede estar justificado por los beneficios tanto tangibles como

## CAPÍTULO IV: ESTUDIO DE FACTIBILIDAD

intangibles que origina el mismo. En este proceso, se necesita de una selección adecuada de los elementos más convenientes para su evaluación.

### **Efectos Económicos**

Pueden clasificarse como:

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

### **Efectos directos:**

#### Positivos

- Se cuenta con una herramienta capaz de mantener la seguridad e integridad de los datos que se procesan.
- El administrador y los usuarios del sistema tendrán la posibilidad de realizar de forma integrada las actividades asociadas al proceso de control de los gastos.
- Se obtiene información actualizada, resumida y detallada, del proceso de control de los gastos.
- Se generan reportes en un tiempo muy bajo y con un alto nivel de fidelidad.

#### Negativos

- Para usar la aplicación es vital el uso de un ordenador conectado a la red, aparejado a los gastos de consumo de energía eléctrica que trae consigo.

### Efectos Indirectos:

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

### Efectos externos:

- Se contará con una herramienta disponible que facilitará el procesamiento de la Información de los cheques emitidos en gastos en la CTE, optimizando el tiempo de realización de las actividades del mismo.

## CAPÍTULO IV: ESTUDIO DE FACTIBILIDAD

### Intangibles:

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

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

### **Situación sin el producto**

Actualmente el proceso de control de la información referente a los cheques emitidos en gastos presenta la siguiente situación:

1. El proceso de recolección de los datos es muy lento y tedioso.
2. La técnica C en gestión económica debe hacer los reportes de manera mecánica.
3. La realización de las búsquedas o consultas se torna engorroso, debido al gran volumen de información que esta maneja.

### **Situación con el producto**

Con el producto informático desarrollado se tiene la siguiente situación:

1. Los usuarios que accedan al sistema deberán estar previamente autenticados; según el tipo de usuario (Administrador, Especialista, Director o Jefe de Área) podrá realizar sus operaciones específicas en el Sistema.
2. El proceso de entrada de datos es muy rápido.
3. Los reportes serán elaborados fácilmente y los cálculos quedaran del lado del sistema disminuyendo de gran manera la probabilidad de error.
4. Las búsquedas o consultas de los datos referentes a los cheques emitidos en gastos serán más rápidas.

### **Beneficios y Costos Intangibles en el proyecto**

Costos:

- Resistencia al cambio de las personas a utilizar el software.

Beneficios:

## CAPÍTULO IV: ESTUDIO DE FACTIBILIDAD

- Disminución del tiempo y esfuerzo que se invierte en esta tarea realizada hasta el momento de forma manual.
- Mayor comodidad para los usuarios.
- Mayor información visual sobre los datos arrojados por el sistema.
- Posibilidad de iniciar el proceso control de cheques emitidos en gastos desde cualquier PC de la entidad que esté conectada a la red.
- Ahorro de tiempo en la búsqueda de información sobre los cheques emitidos en gastos.
- Muestra los reportes rápido y con un alto grado de actualidad.

### Ficha de costo

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

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

#### 4.3.2 Costos en Moneda Libremente Convertible

Tabla 4: Ficha de Costo en Moneda Libremente Convertible.

<b>Ficha de Costo.</b>	
<b>Costos Moneda Libremente Convertible</b>	<b>Precio(s)</b>
<b>Costos Directos</b>	
Compra de equipos de cómputo	0,00
Alquiler de equipos de cómputo	0,00
Compra de licencia de Software	0,00
Depreciación de equipos	25,00
Materiales directos	0,00
<b>Subtotal</b>	<b>25,00</b>

## CAPÍTULO IV: ESTUDIO DE FACTIBILIDAD

<b>Costos Indirectos</b>	
Formación del personal que elabora el proyecto	0,00
Gastos en llamadas telefónicas	0,00
Gastos para el mantenimiento del centro	0,00
KnowHow	0,00
Gastos en representación	0,00
<b>Subtotal</b>	0,00
<b>Gastos de Distribución y Venta</b>	
Participación en ferias o exposiciones	0,00
Gastos en transportación	0,00
Compra de materiales de propagandas	0,00
<b>Subtotal</b>	0,00
<b>Total</b>	25,00

### 4.3.3 Costos en Moneda Nacional:

Tabla 5: Ficha de Costo Moneda Nacional.

<b>Ficha de Costo.</b>	
<b>Costos Moneda Nacional</b>	<b>Precio(s)</b>
<b>Costos Directos</b>	
Salario del personal que laborará en el proyecto	400,00
12,5% del total de gastos por salarios se dedica a la seguridad social	0,00

## CAPÍTULO IV: ESTUDIO DE FACTIBILIDAD

9.09% de salario total, por concepto de vacaciones a acumular	0,00
Gasto por consumo de energía eléctrica	170,70
Gastos en llamadas telefónicas	0,00
Gastos administrativos	0,00
<b>Subtotal</b>	570,70
<b>Costos Indirectos</b>	
KnowHow	0,00
<b>Subtotal</b>	
<b>Total</b>	570,70

La evaluación económica se efectúa conjuntamente con la evaluación técnica del proyecto, que consiste en cerciorarse de la factibilidad técnica del mismo. En el análisis de la factibilidad técnica del proyecto, se pudo apreciar que se cuenta con la disponibilidad de *hardware/software* por lo que se puede inferir que el proyecto es factible técnicamente y no necesita de inversión alguna para su realización, por tanto la decisión de inversión recae en la evaluación económica. Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la Metodología Costo-Beneficio. Dentro de esta metodología la técnica de punto de equilibrio aplicable a proyectos donde los beneficios tangibles no son evidentes el análisis se basa exclusivamente en los costos. Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es poco significativo, se tomará como costo el tiempo en minutos empleado para realizar la gestión de la información del sistema de control de la información referente a los cheques emitidos en gastos.

### Valores de la variable (Solución sin el software)

1. Generar declaración jurada (480 min)
2. Generar estado de cuentas mensual (240 min)

## CAPÍTULO IV: ESTUDIO DE FACTIBILIDAD

3. Generar listado del maestro (360 min)
4. Generar nomenclador. (60 min)

### Valores de la variable (Solución con el sistema)

1. Generar declaración jurada (20 min)
2. Generar estado de cuentas mensual (10 min)
3. Generar listado del maestro (15 min)
4. Generar nomenclador. (8 min)

El gráfico que a continuación se describe, muestra el comportamiento de estas variables teniendo en cuenta las formas de realización de las actividades que componen el proceso.

Tiempo (min)

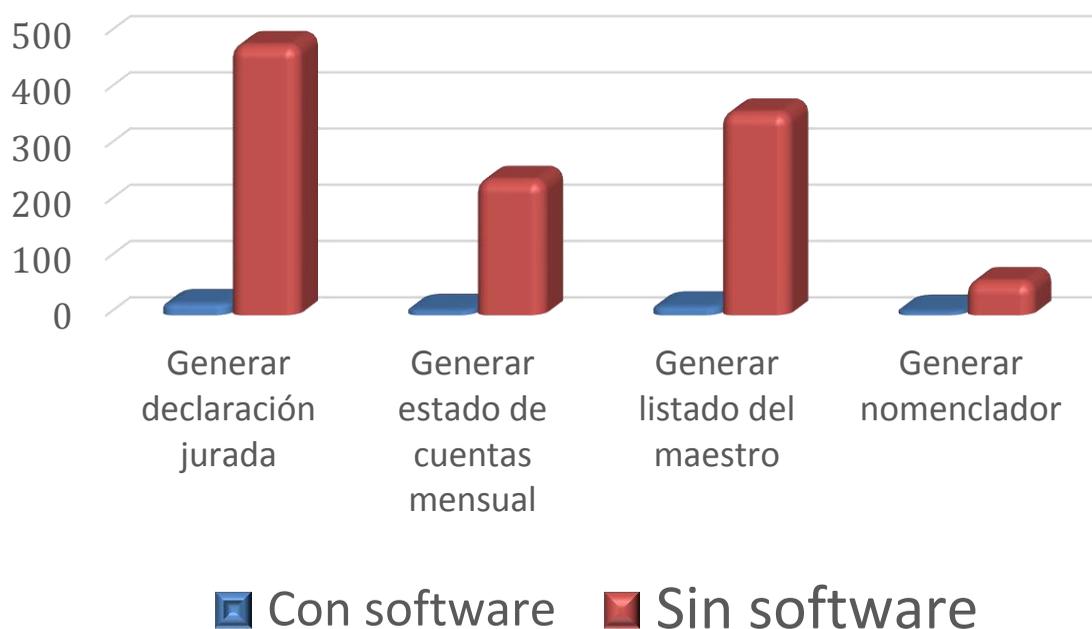


Figura 7: Comparación de la solución manual y con el sistema.

Teniendo en cuenta los resultados reflejados en la gráfica queda demostrada la factibilidad del sistema, basándose en el tiempo que demora la realización de las actividades fundamentales de forma manual e informatizada.

## CAPÍTULO IV: ESTUDIO DE FACTIBILIDAD

### 4.4 Conclusiones del capítulo

En este capítulo se realizó el estudio de factibilidad mediante La Metodología Costo-Beneficio. Se analizó la factibilidad técnica demostrándose que la institución cuenta con la infraestructura necesaria. Para la factibilidad económica, se llevó a cabo un análisis de costo beneficio, considerando los efectos económicos, los beneficios y costos intangibles, así como el cálculo del costo de ejecución del proyecto mediante la ficha de costo, arrojando como resultado \$ 25.00 CUC y \$ 570.70 MN, demostrándose la factibilidad del proyecto.

## CONCLUSIONES GENERALES Y RECOMENDACIONES

### Conclusiones Generales

Con el desarrollo de la aplicación web para la gestión de la información de las transacciones bancarias sobre la cuenta de gastos de la CTE Lidio Ramón Pérez se dio cumplimiento a los objetivos trazados en esta investigación, pues se obtuvo como resultado una herramienta informática en la que se aplican los resultados de la investigación realizada arribándose a las siguientes conclusiones:

- Se elaboró el marco teórico metodológico que fundamenta la investigación, esto permitió que quedara identificada la situación problemática existente y las bases para comenzar con el diseño e implementación de la aplicación.
- Se realizó un levantamiento sobre los procesos que se llevan a cabo en el departamento de contabilidad de la Central Termoeléctrica Lidio Ramón Pérez.
- Durante el desarrollo de la metodología XP se generaron 16 requisitos funcionales, 16 historias de usuario, 16 tarjetas CRC, 5 pruebas de aceptación y 41 tareas de ingeniería.
- Se efectuó un estudio de la factibilidad técnica, económica, y operativa del software en cuestión; arrojando resultados satisfactorios para el proyecto.

Por lo antes expuesto se concluye que los objetivos propuestos fueron cumplidos satisfactoriamente.

### Recomendaciones

El autor de la presente investigación ha considerado pertinente ofrecer las siguientes recomendaciones:

- Realizar la implantación del software en la empresa termoeléctrica Lidio Ramón Pérez
- Ejecutar la aplicación preferentemente en los navegadores Mozilla Firefox ya que traduce los estilos de diseño con mayor claridad.
- Realizar un estudio más profundo de este sistema en vista a perfeccionarlo en versiones futuras.

## REFERENCIAS BIBLIOGRÁFICAS

### Referencias Bibliográficas

1. Robles, R. García. El nuevo paradigma de la gestión del conocimiento y su aplicación en el ámbito educativo. [Online] [Cited: 06 02, 2014.]  
<http://tecnologiedu.us.es/edutec/paginas/125.htm>.  
[Online] Casa de Software. [Cited: 05 28, 2014.] <http://www.matcom.uh.cu>.
2. Lecusay, Pablo. Sistema informático para favorecer el proceso de selección de personal en el instituto superior minero metalúrgico de Moa. Moa: ISMMM, 2012.
3. Silva, Ing. Exneider Proenza. Sistema de Costos de Calidad para la Empresa. 2010.
4. Ferrer, J. [Online] 02 25, 2003. [Cited: 02 15, 2014.]  
<http://libresoft.es/downloads/ferrer-20030312.pdf>.
5. [www.postgresql.org/support/professional\\_support](http://www.postgresql.org/support/professional_support). [Online] [Cited: 02 18, 2014.]  
[http://www.postgresql.org/support/professional\\_support](http://www.postgresql.org/support/professional_support).
8. MySQL: Ventajas. [Online] [Cited: 06 8, 2014.]  
<http://sistemaspyt.blogspot.com/2008/09/ventajas.html>.
9. Meriño, Pricido Fidel Rusiaux. Herramienta Informática para la gestión de información sobre Incidentes y Accidentes Ambientales en la Empresa Che Guevara. Moa: ISMMM, 2013.
10. Jeffries, R. What is Extreme Programming? [Online] [Cited: 02 16, 2014.]  
<http://www.xpprogramming.com/xpmag/whatisxp.htm>.
11. Bañeres, Juan Palacio. Gestión ágil de proyectos: Scrum. Navegapolis.net. [Online] <http://www.navegapolis.net>.
12. Pressman, Roger S. Ingeniería de Softwar, un enfoque práctico. s.l.: McGraw-Hill, 2011.
12. Gabriel Duarte. Preguntale al Gabo. [Online] [Cited: 04 14, 2014.]  
<http://gabrielduarte77.wordpress.com/category/yii-framework>.
13. EmbarcaderoER/Studio. [Online] [Cited: 02 25, 2014.]  
[http://bureaudeprensa.com/es/view.php?bn=bureaudeprensa\\_software&key=1153755](http://bureaudeprensa.com/es/view.php?bn=bureaudeprensa_software&key=1153755)
14. [https://timetrackerprojects.com/es/Inicio/gestion\\_en\\_linea\\_de\\_cuenta\\_de\\_gastos\\_y\\_gestion\\_de\\_proyectos](https://timetrackerprojects.com/es/Inicio/gestion_en_linea_de_cuenta_de_gastos_y_gestion_de_proyectos)
15. <http://ahorroyhogar.com/n/2137/programas-para-controlar-tus-gastos>
16. <http://www.consumer.es/web/es/tecnologia/software/2012/02/21/206849>



## REFERENCIAS BIBLIOGRÁFICAS

17. *Manual del programador, Parte 1: Programación en Visual FoxPro* (mayo del 2000). 83. <http://www.itzitacuaro.edu.mx/residencias/archivos/2009.pdf>

## BIBLIOGRAFÍA

### Bibliografía

1. [BECK, 1999]. Kent Beck. *Extreme Programming Explained: Embrace Change*, Pearson Education 1999. Addison-Wesley, 2000.
2. [BEIGHLEY; MORRISON, 2005]. Beighley, Lynn, Morrison, Michael. *Head First PHP & MySQL*. O'Reilly, 2005.
3. [CANOS; LETELIER; PENADES, 2005]. H. Canós, José, Letelier, Patricio, Penadés, M. Carmen. *Metodologías Ágiles en el desarrollo de software*. Universidad Politécnica de Valencia, 2005.
4. Calvo Burke, Aurelio. *Sistema de Gestión de Información sobre Derrames de Hidrocarburos en el puerto de Moa*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Minero Metalúrgico de Moa "Dr. Antonio Núñez Jiménez", Moa, julio 2013.
5. [DUARTE, 2011]. Gabriel Duarte. *Guía básica de Yii Framework*. España, 2011.
6. [FERNANDEZ, 2002]. Gerardo Fernández Escribano. *Introducción a Extreme Programming*. Ingeniería de Software II. 9-12-2002.
7. [FREEMAN, 2005]. Elisabeth Freeman. *A Brain-Friendly Guide to HTML & CSS*. *Head First Html with Css and Xhtml*. O'Reilly, 2005.
8. GDP. *Manual de PHP*. Grupo de Documentación de PHP, 2002.
9. [HERNANDEZ, 2006]. Edelsys Hernández Meléndrez. *Como escribir una tesis*. Escuela Nacional de Salud Pública, 2006.
10. Hernández Luque, Eyllin. *Aplicación Web para la gestión de información de instrumentos meteorológicos del Instituto de Meteorología*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Universidad de las Ciencias Informáticas, Ciudad de la Habana, mayo 2007.
11. [LETELIER; PENADES, 2005]. Letelier, Patricio; Penadés, M<sup>a</sup> Carmen. *Metodologías ágiles para el desarrollo de software: Extreme Programming (XP)*. Universidad Politécnica de Valencia, 2005.
12. *Manual de SQL*. WebExperto.COM, 2001.
13. [MARTEENS, 1999]. Ian Marteens. *La cara oculta de C++ Builder*. Madrid, 1999.
14. Mesa Martínez, Jorge I. *Modelos de formularios con Yii*. <http://blog.jorgeivanmeza.com/category/desarrollo-de-software> (20/02/14).
15. Peñalosa B, Leisewitz T, Bastías G, Zárata V, Depaux R, Villarroel L, et al. *Metodología para la evaluación de la relación costo-efectividad en centros de atención primaria de Chile*. *Revista Panam. Salud Pública*. 2010.
16. Rodríguez Pérez, Agustín. *SIGEP: Sistema para La Gestión del Presupuesto en el ISMMM*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Minero Metalúrgico de Moa "Dr. Antonio Núñez Jiménez", Moa, junio 2012.

## BIBLIOGRAFÍA

17. Salazar Pupo, Ricardo. Sistema de gestión de capacitación ECRIN. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, Moa, julio 2013.
18. Schenone Marcelo Hernán *Diseño de una Metodología Ágil de Desarrollo de Software*. Tesis de Grado en Ingeniería en Informática. Universidad de Buenos Aires, argentino, abril 2004.
19. Suárez Falcón, Yuniel. Sistema de gestión de información para la estimación y mitigación de riesgos de desastres en la Provincia Holguín. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, Moa, julio 2013.
20. [WELLING; THOMSON, 2004]. Welling Luke, Thomson, Laura. Desarrollo Web con PHP y MySQL. Anaya Multimedia, 2004.
21. *Manual del programador, Parte 1: Programación en Visual FoxPro* (mayo del 2000). 83. <http://www.itzitacuaro.edu.mx/residencias/archivos/2009.pdf>

## GLOSARIO DE TÉRMINOS

### Glosario de Términos

**Actualización:** Cambios que se aplican a un programa, para corregir errores, agregarle funcionalidades, etc.

**Análisis costo beneficio:** El coste-beneficio es una lógica o razonamiento basado en el principio de obtener los mayores y mejores resultados al menor esfuerzo invertido, tanto por eficiencia técnica como por motivación humana. Se supone que todos los hechos y actos pueden evaluarse bajo esta lógica, aquellos dónde los beneficios superan el coste son exitosos, caso contrario fracasan.

**Aplicación:** Programa preparado para una utilización específica. Existen muchos programas de ordenador que pueden clasificarse como aplicación. Generalmente se les conoce como Software.

**Aplicación web:** Una aplicación Web es un sistema informático que los usuarios usan accediendo a un servidor Web a través de los protocolos de Internet. Las aplicaciones Web son populares por su practicidad del navegador Web como cliente ligero.

**API:** Una API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**ASP:** Active Server Pages, también conocido como ASP clásico, es una tecnología de Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services (IIS).

**Ajax:** acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

**Beneficio:** Ganancia económica que se obtiene de un negocio, inversión u otra actividad mercantil.

**Bugs:** Errores que se presentan en la aplicación que atentan contra su correcto funcionamiento.

**Cliente:** Persona, organización o grupo de personas que solicita la construcción de un sistema, ya sea empezando desde cero, o mediante el refinamiento de versiones sucesivas.

**Costo:** En economía el coste o costo es el valor monetario de los consumos de factores que supone el ejercicio de una actividad económica destinada a la producción de un bien o servicio.

## GLOSARIO DE TÉRMINOS

**Compatibilidad:** Condición que hace que un programa y un sistema, arquitectura o aplicación logren comprenderse correctamente tanto directamente o indirectamente mediante un algoritmo.

**Concurrencia:** Coincidencia, concurso simultáneo de varias circunstancias. Es la simultaneidad en la ejecución de múltiples tareas interactivas. Estas tareas pueden ser un conjunto de procesos o hilos de ejecución creados por un único programa.

**Firebird:** Sistema de administración de base de datos relacional (o RDBMS) (Lenguaje consultas: SQL) de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000. Su código fue reescrito de C a C++.

**Gestión de información:** es un conjunto de procesos por los cuales se controla el ciclo de vida de la información, desde su obtención - por creación o captura, hasta su disposición final - archivada o eliminada. Los procesos también comprenden la extracción, combinación, depuración y distribución de la información a los interesados.

**Herramientas:** Son los ambientes de apoyo necesario para automatizar las prácticas de Ingeniería de Software.

**Historia de usuario:** es una representación de un requisito de software escrito en una o dos frases utilizando el lenguaje común del usuario.

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

**IDE:** Integrated Development Environment / Entorno de Desarrollo Integrado. Entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

**Informatización:** Aplicar los conjuntos de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores.

**Interfaz:** Conjunto de representaciones de operaciones públicas.

**Iteraciones:** En el contexto de un proyecto se refieren a la técnica de desarrollar y entregar componentes incrementales de funcionalidades de un negocio. Una iteración resulta en uno o más paquetes atómicos y completos del trabajo del proyecto que pueda realizar alguna función tangible del negocio.

Múltiples iteraciones contribuyen a crear un producto completamente integrado.

**JavaScript:** Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

**Java:** Es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Con respecto a la memoria, su gestión

## GLOSARIO DE TÉRMINOS

no es un problema ya que ésta es gestionada por el propio lenguaje y no por el programador.

**Lógica del negocio:** Parte de un sistema que se encarga de las tareas relacionadas con los procesos de un negocio, tales como ventas, control de inventario, contabilidad, etc.

**Multiplataforma:** Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas o sistemas operativos.

**Metodología Ágil:** Constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de proyectos que las metodologías convencionales debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

**Metodología de desarrollo:** Se define como un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información.

**Metodologías tradicionales:** Metodologías basadas en procesos.

**Navegador web:** Un navegador o navegador web (del inglés, web browser) es una aplicación que opera a través de Internet, interpretando la información de archivos y sitios web para que podamos ser capaces de leerla, (ya se encuentre ésta alojada en un servidor dentro de la World Wide Web o en un servidor local).

**PHP:** PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje interpretado de alto nivel embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, con solamente un par de características PHP específicas. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil.

**Proceso:** Conjunto de las fases sucesivas de un fenómeno natural o de una operación artificial. *INF.* Programa en ejecución.

**Proyecto:** Esfuerzo de desarrollo para llevar un sistema a lo largo de un ciclo de vida.

**Programación extrema:** Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

**Requisitos:** Son las funciones, servicios y restricciones operativas del sistema.

**Release:** Versión candidata definitiva de un producto de software y se refiere a un producto final, preparado para lanzarse como versión definitiva a menos que aparezcan errores que lo impidan.

**Script:** En informática un guión, archivo de órdenes o archivo de procesamiento por lotes, es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. Los guiones son casi siempre interpretados, pero no todo programa interpretado es considerado un guión.

**Sistema:** Delimita el mundo sobre el cual se está construyendo el modelo

## GLOSARIO DE TÉRMINOS

**SQLite:** es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 kB) biblioteca en C. SQLite es un proyecto de dominio público

**Servidor Web:** Un programa que corre sobre el servidor que escucha las peticiones HTTP (Hyper Text Markup Language) que le llegan y las satisface. Dependiendo del tipo de petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición

**Software:** Se refiere a los programas y datos almacenados en un ordenador. Es la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo.

**Testing:** Proceso de pruebas usado para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador o videojuego. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

**Usuario:** Persona que interactúa con el sistema



# GLOSARIO DE TÉRMINOS

## Anexo 1: Historias de usuarios

Tabla 2.1.4.1 HU No. 1 Autenticar usuario

Historia de usuario	
<b>Número: 1</b>	Usuario: todos
<b>Nombre:</b> Autenticar usuarios.	
<b>Prioridad en el negocio:</b> Alta	Riesgo en desarrollo: media
<b>Puntos estimados:</b> 1	Iteración asignada: Primera
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> Los usuarios del sistema ingresan sus datos para entrar al sistema (nombre de usuario y contraseña). El sistema verifica que los datos estén correctos, en caso de que nos sean correctos la aplicación muestra un mensaje de error.	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.2 HU No. 2 Gestionar Usuario

Historia de usuario	
<b>Número: 2</b>	Usuario: Administrador
<b>Nombre:</b> Gestionar usuarios	
<b>Prioridad en el negocio:</b> Alta	Riesgo en desarrollo: Alta
<b>Puntos estimados:</b> 1	Iteración asignada: Primera
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el administrador ingresara los datos de un usuario que se insertara en la tabla usuarios teniendo posibilidad de listarlo modificarlo y eliminarlo.	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.3 HU No. 3 Tramitar crédito de la UNE

Historia de usuario	
<b>Número: 3</b>	Usuario: Técnico
<b>Nombre:</b> Tramitar crédito de la UNE	
<b>Prioridad en el negocio:</b> Alta	Riesgo en desarrollo: Alta
<b>Puntos estimados:</b> 1	Iteración asignada: Primera
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el usuario ingresa los datos del crédito teniendo la posibilidad de listarlos	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.4 HU No. 4 Tramitar Devolución de Tercero

Historia de usuario	
<b>Número: 4</b>	Usuario: Técnico
<b>Nombre:</b> Tramitar Devolución de Tercero	
<b>Prioridad en el negocio:</b> Alta	Riesgo en desarrollo: Alta
<b>Puntos estimados:</b> 1	Iteración asignada: Primera
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el usuario ingresa los datos de la evolución de Tercero teniendo la posibilidad de listarlas	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.5 HU No. 5 Tramitar Cheque

Historia de usuario	
<b>Número: 5</b>	Usuario: Técnico
<b>Nombre:</b> Tramitar Cheque	
<b>Prioridad en el negocio:</b> Alta	Riesgo en desarrollo: Alta
<b>Puntos estimados:</b> 1	Iteración asignada: Primera
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el usuario ingresa los datos del Cheque teniendo la posibilidad de listarlos y cancelarlos	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.7 HU No. 7 Tramitar Comisión Bancaria

Historia de usuario	
<b>Número: 7</b>	Usuario: Técnico
<b>Nombre:</b> Tramitar Comisión Bancaria	
<b>Prioridad en el negocio:</b> Alta	Riesgo en desarrollo: Alta
<b>Puntos estimados:</b> 1	Iteración asignada: Primera
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el usuario ingresa los datos de la Comisión bancaria teniendo la posibilidad de listarlas y cancelarlas	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.8 HU No. 8 Eliminar Transacción

Historia de usuario	
<b>Número: 8</b>	Usuario: Administrador
<b>Nombre:</b> Eliminar Transacción	
<b>Prioridad en el negocio:</b> Alta	Riesgo en desarrollo: Alta
<b>Puntos estimados:</b> 1	Iteración asignada: Primera
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el administrador selecciona la opción Transacciones(Cheque, transferencia, Crédito de la UNE, Devolución de Tercero, Comisión Bancaria) y el sistema le muestra una lista donde las puede eliminar	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.9 HU No. 9 Gestionar Cierre del mes

Historia de usuario	
<b>Número: 9</b>	Usuario: Técnico
<b>Nombre:</b> Gestionar Cierre del mes	
<b>Prioridad en el negocio:</b> Alta	Riesgo en desarrollo: Alta
<b>Puntos estimados:</b> 1	Iteración asignada: Primera
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el usuario selecciona la opción actualizar en y el sistema muestra un formulario con el saldo de la cuenta para inversiones y operaciones con que se iniciara el mes próximo y la opción de actualizar.	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.10 HU No. 10 Gestionar Acápite

Historia de usuario	
<b>Número: 10</b>	Usuario: Técnico
<b>Nombre:</b> Gestionar acápite	
<b>Prioridad en el negocio:</b> Alta	Riesgo en desarrollo: Alta
<b>Puntos estimados:</b> 1	Iteración asignada: Primera
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el usuario selecciona la opción cierre de mes y el sistema inserta un saldo para el mes entrante.	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.11 HU No. 11 Gestionar Persona

Historia de usuario	
<b>Número: 11</b>	Usuario: Administrador
<b>Nombre:</b> Gestionar persona	
<b>Prioridad en el negocio:</b> Alta	Riesgo en desarrollo: Alta
<b>Puntos estimados:</b> 1	Iteración asignada: Primera
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el administrador inserta modifica y elimina los datos de las personas que pueden convertirse en usuarios del sistema.	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.12 HU No. 12 Gestionar Modificación

Historia de usuario	
<b>Número: 12</b>	Usuario: Técnico
<b>Nombre:</b> listar modificación	
<b>Prioridad en el negocio:</b> Alta	Riesgo en desarrollo: Alta
<b>Puntos estimados:</b> 1	Iteración asignada: segunda
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> al realizarse cualquier acción del usuario se inserta una modificación con la fecha el usuario y el tipo	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.13 HU No. 13 Generar estado de cuentas mensual

Historia de usuario	
<b>Número: 13</b>	Usuario: Técnico
<b>Nombre:</b> Generar estado de cuenta mensual	
<b>Prioridad en el negocio:</b> Alta	Riesgo en desarrollo: Alta
<b>Puntos estimados:</b> 1	Iteración asignada: segunda
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el usuario selecciona la opción de saldo y se le muestra el estado de las cuentas con que trabaja teniendo en cuenta los gastos y los ingresos del mes	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.14 HU No. 14 Generar declaración jurada

Historia de usuario	
<b>Número: 14</b>	Usuario: Técnico
<b>Nombre:</b> Generar declaración jurada	
<b>Prioridad en el negocio:</b> Alta	Riesgo en desarrollo: Alta
<b>Puntos estimados:</b> 1	Iteración asignada: segunda
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el usuario selecciona la opción declaración jurada y el sistema muestra un listado con los gastos del mes y los acápites que autorizaron los mismos	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.15 HU No. 15 Generar nomenclador

Historia de usuario	
<b>Número: 15</b>	Usuario: Técnico
<b>Nombre:</b> Generar nomenclador	
<b>Prioridad en el negocio:</b> Baja	Riesgo en desarrollo: Bajo
<b>Puntos estimados:</b> 1	Iteración asignada: segunda
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el usuario selecciona la opción nomenclador y el sistema muestra un listado de los acápites y los gastos que autoriza cada uno	
<b>Observaciones:</b> Confirmado por el cliente.	

Tabla 2.1.4.16 HU No. 16 Gestionar datos generales

Historia de usuario	
<b>Número: 16</b>	Usuario: Técnico
<b>Nombre:</b> Gestionar datos generales	
<b>Prioridad en el negocio:</b> Baja	Riesgo en desarrollo: Bajo
<b>Puntos estimados:</b> 1	Iteración asignada: segunda
<b>Programador responsable:</b> Orlando Prieto Silva	
<b>Descripción:</b> el usuario selecciona la opción datos generales y el sistema muestra un formulario con los datos actuales y la opción entrar datos	
<b>Observaciones:</b> Confirmado por el cliente.	

## Anexo 2: Tarjetas CRC

### Tabla 2.2.1.1 Tarjeta CRC No. 1

<b>Nombre de la Clase:</b> Autenticar Usuario	
<b>Tipo de clase:</b>	
Lógica del negocio	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Autentica al usuario en el sistema.	login

### Tabla 2.2.1.2 Tarjeta CRC No. 2

<b>Nombre de la Clase:</b> Gestionar Usuario	
<b>Tipo de clase:</b>	
Lógica del negocio	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Inserta datos del usuario	insertar_usuario
Muestra los datos del usuario	mostrar_usuario
Modifica los datos del usuario	modificar_usuario
Eliminar el usuario de la base de datos	eliminar_usuario

### Tabla 2.2.1.4 Tarjeta CRC No. 4

<b>Nombre de la Clase:</b> Gestionar Saldo mensual	
<b>Tipo de clase:</b>	
Lógica del negocio	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Inserta datos del saldo del mes	insertar_saldo_mensual
Muestra los datos del saldo del mes	mostrar_saldo_mensual
Modifica los datos del saldo del mes	modificar_saldo_mensual

### Tabla 2.2.1.5 Tarjeta CRC No. 5

<b>Nombre de la Clase:</b> Gestionar Acápito	
<b>Tipo de clase:</b>	
Lógica del negocio	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Inserta datos del acápito	Insertar_acápito
Muestra los datos del acápito	mostrar_acápito
Modifica los datos del acápito	modificar_acápito
Eliminar el acápito de la base de datos	Eliminar_acápito

Tabla 2.2.1.6 Tarjeta CRC No. 6

<b>Nombre de la Clase:</b> Gestionar Persona	
<b>Tipo de clase:</b>	
Lógica del negocio	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Inserta datos de la persona	Insertar_persona
Muestra los datos de la persona	mostrar_persona
Modifica los datos de la persona	modificar_persona
Eliminar la persona de la base de datos	eliminar_persona

Tabla 2.2.1.7 Tarjeta CRC No. 7

<b>Nombre de la Clase:</b> Gestionar Modificación	
<b>Tipo de clase:</b>	
Lógica del negocio	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Inserta datos de la modificación	Insertar_modificación
Muestra los datos de la modificación	mostrar_modificación
Eliminar la persona de la base de datos	eliminar_modificación

Tabla 2.2.1.8 Tarjeta CRC No. 8

<b>Nombre de la Clase:</b> Gestionar Crédito de la UNE	
<b>Tipo de clase:</b>	
Lógica del negocio	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Inserta datos del crédito de la UNE	insertar_cheque
Muestra los datos del crédito de la UNE	mostrar crédito
Modifica los datos del crédito de la UNE	insertar_cheque
Eliminar cheque de la BD	eliminar cheque

Tabla 2.2.1.9 Tarjeta CRC No. 9

<b>Nombre de la Clase:</b> Gestionar Datos Generales	
<b>Tipo de clase:</b>	
Lógica del negocio	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Inserta datos de la empresa	Insertar_datos_generales
Muestra los datos de la empresa	mostrar_datos_generales
Modifica los datos de la empresa	modificar_datos_generales

Tabla 2.2.1.10 Tarjeta CRC No. 10

<b>Nombre de la Clase:</b> Gestionar Devolución de tercero	
<b>Tipo de clase:</b>	
Lógica del negocio	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Inserta datos de la devolución	insertar_cheque
Muestra los datos de la devolución	mostrar_devolucion
Modifica los datos de la devolución	insertar_cheque
Eliminar la devolución de la BD	eliminar_cheque

Tabla 2.2.1.11 Tarjeta CRC No. 11

<b>Nombre de la Clase:</b> Gestionar Transferencia	
<b>Tipo de clase:</b>	
Lógica del negocio	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Inserta datos de la transferencia	insertar_cheque
Muestra los datos de la transferencia	mostrar_transferencia
Modifica los datos de la transferencia	insertar_cheque
Eliminar la transferencia de la BD	eliminar_cheque

Tabla 2.2.1.12 Tarjeta CRC No. 12

<b>Nombre de la Clase:</b> Gestionar Comisión bancaria	
<b>Tipo de clase:</b>	
Lógica del negocio	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Inserta datos de la comisión bancaria	insertar_cheque
Muestra los datos de la comisión bancaria	mostrar_comisión_bancaria
Modifica los datos de la comisión bancaria	insertar_cheque
Eliminar la comisión bancaria de la BD	eliminar_cheque

Tabla 2.2.1.13 Tarjeta CRC No. 13

<b>Nombre de la Clase:</b> Generar Saldo Mensual	
<b>Tipo de clase:</b>	
Lógica del negocio	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Genera el estado de la cuenta en el mes	Generar_saldo

### Anexo 3: Tareas de ingeniería

**Tabla 3.3.2.1 Tarea de Programación autenticar usuario**

<b>Tarea de Programación</b>	
No. Tarea: 1	No. Historia: 1
Nombre de la tarea: Autenticar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 20/05/2015	Fecha fin: 24/05/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite autenticar los usuarios al sistema.	

**Tabla 3.3.2.2 Tarea de Programación Insertar usuario**

<b>Tarea de Programación</b>	
No. Tarea: 2	No. Historia: 2
Nombre de la tarea: Insertar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 25/05/2015	Fecha fin: 26/05/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite insertar los usuarios que van a interactuar con el sistema.	

**Tabla 3.3.2.3 Tarea de Programación Mostrar usuario**

<b>Tarea de Programación</b>	
No. Tarea: 3	No. Historia: 2
Nombre de la tarea: Mostrar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 26/05/2015	Fecha fin: 27/05/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite mostrar los datos de los usuarios del sistema.	

**Tabla 3.3.2.4 Tarea de Programación Modificar usuario**

<b>Tarea de Programación</b>	
No. Tarea: 4	No. Historia: 2
Nombre de la tarea: Modificar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 28/05/2015	Fecha fin: 28/05/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite modificar los datos de los usuarios del sistema.	

**Tabla 3.3.2.5 Tarea de Programación Eliminar usuario**

<b>Tarea de Programación</b>	
No. Tarea: 5	No. Historia: 2
Nombre de la tarea: Eliminar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 29/05/2015	Fecha fin: 29/05/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite eliminar el usuario seleccionado de la base de datos.	

**Tabla 3.3.2.6 Tarea de Programación Insertar Crédito de la UNE**

<b>Tarea de Programación</b>	
No. Tarea: 6	No. Historia: 3
Nombre de la tarea: Insertar Crédito de la UNE	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 30/05/2015	Fecha fin: 01/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite insertar un Crédito de la UNE al sistema	

**Tabla 3.3.2.7 Tarea de Programación Listar Créditos de la UNE**

<b>Tarea de Programación</b>	
No. Tarea: 7	No. Historia: 3
Nombre de la tarea: Listar Crédito de la UNE	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 02/06/2015	Fecha fin: 02/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Muestra una lista de los Créditos de la UNE existentes	

**Tabla 3.3.2.8 Tarea de Programación Insertar Devolución de Tercero**

<b>Tarea de Programación</b>	
No. Tarea: 8	No. Historia: 4
Nombre de la tarea: Insertar Devolución de Tercero	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 03/06/2015	Fecha fin: 03/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite insertar una Devolución de tercero al sistema	

**Tabla 3.3.2.8 Tarea de Programación Listar Devolución de tercero**

<b>Tarea de Programación</b>	
No. Tarea: 8	No. Historia: 4
Nombre de la tarea: Listar Devolución de Tercero	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 04/06/2015	Fecha fin: 05/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite listar las Devoluciones de Tercero que existan en el sistema	

**Tabla 3.3.2.9 Tarea de Programación Insertar cheque**

<b>Tarea de Programación</b>	
No. Tarea: 9	No. Historia: 5
Nombre de la tarea: Insertar cheque	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 05/06/2015	Fecha fin: 05/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite Insertar un cheque en el sistema	

**Tabla 3.3.2.10 Tarea de Programación Listar cheques**

<b>Tarea de Programación</b>	
No. Tarea: 10	No. Historia: 5
Nombre de la tarea: Listar Cheques	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 06/06/2015	Fecha fin: 06/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite ver una lista con todos los cheques que existen en el sistema	

**Tabla 3.3.2.11 Tarea de Programación Cancelar Cheque**

<b>Tarea de Programación</b>	
No. Tarea: 11	No. Historia: 5
Nombre de la tarea: Cancelar Cheque	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 07/06/2015	Fecha fin: 07/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite cancelar un cheque	

**Tabla 3.3.2.12 Tarea de Programación Insertar Transferencia**

<b>Tarea de Programación</b>	
No. Tarea: 12	No. Historia: 6
Nombre de la tarea: Insertar Transferencia	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 08/06/2016	Fecha fin: 08/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite Insertar una transferencia en el sistema	

**Tabla 3.3.2.13 Tarea de Programación Listar Transferencias**

<b>Tarea de Programación</b>	
No. Tarea: 13	No. Historia: 6
Nombre de la tarea: Listar Transferencias	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 09/06/2016	Fecha fin: 09/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite Listar las Transferencias existentes en el sistema	

**Tabla 3.3.2.14 Tarea de Programación cancelar Transferencia**

<b>Tarea de Programación</b>	
No. Tarea: 14	No. Historia: 6
Nombre de la tarea: Cancelar Transferencias	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 10/06/2016	Fecha fin: 10/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite cancelar una Transferencia de las existentes en el sistema	

**Tabla 3.3.2.15 Tarea de Programación insertar Comisión Bancaria**

<b>Tarea de Programación</b>	
No. Tarea: 15	No. Historia: 7
Nombre de la tarea: Insertar Comisión Bancaria	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 11/06/2016	Fecha fin: 11/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite insertar una nueva Comisión Bancaria en el sistema	

**Tabla 3.3.2.16 Tarea de Programación Listar Comisiones Bancarias**

<b>Tarea de Programación</b>	
No. Tarea: 16	No. Historia: 7
Nombre de la tarea: Insertar Comisión Bancaria	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 12/06/2016	Fecha fin: 12/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite listar las Comisiones Bancarias existentes en el sistema	

**Tabla 3.3.2.17 Tarea de Programación Eliminar Transacción**

<b>Tarea de Programación</b>	
No. Tarea: 17	No. Historia: 8
Nombre de la tarea: Eliminar Transacción	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 13/06/2016	Fecha fin: 13/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite eliminar una transacción de las existentes en el sistema	

**Tabla 3.3.2.18 Tarea de Programación Insertar cierre mensual**

<b>Tarea de Programación</b>	
No. Tarea: 18	No. Historia: 9
Nombre de la tarea: Insertar cierre del mes	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 14/06/2015	Fecha fin: 14/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite Insertar un saldo para el mes	

**Tabla 3.3.2.19 Tarea de Programación Modificar Cierre Mensual**

<b>Tarea de Programación</b>	
No. Tarea: 19	No. Historia: 9
Nombre de la tarea: Modificar saldo mensual	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 15/06/2015	Fecha fin: 15/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite modificar el saldo del mes	

**Tabla 3.3.2.20 Tarea de Programación Mostrar Cierre Mensual**

<b>Tarea de Programación</b>	
No. Tarea: 20	No. Historia: 9
Nombre de la tarea: Mostrar cierre mensual	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 16/06/2015	Fecha fin: 16/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Muestra el saldo del mes	

**Tabla 3.3.2.21 Tarea de Programación insertar acápite**

<b>Tarea de Programación</b>	
No. Tarea: 21	No. Historia: 10
Nombre de la tarea: Insertar acápite	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 16/06/2015	Fecha fin: 16/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite insertar un acápite	

**Tabla 3.3.2.22 Tarea de Programación modificar acápite**

<b>Tarea de Programación</b>	
No. Tarea: 22	No. Historia: 10
Nombre de la tarea: Modificar acápite	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 17/06/2015	Fecha fin: 17/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite modificar un acápite	

**Tabla 3.3.2.23 Tarea de Programación mostrar acápite**

<b>Tarea de Programación</b>	
No. Tarea: 23	No. Historia: 10
Nombre de la tarea: Mostrar acápite	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 17/06/2015	Fecha fin: 17/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Muestra los acápites que se utilicen	

**Tabla 3.3.2.24 Tarea de Programación eliminar acápite**

<b>Tarea de Programación</b>	
No. Tarea: 24	No. Historia: 10
Nombre de la tarea: Eliminar acápite	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 17/06/2015	Fecha fin: 18/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite eliminar un acápite	

**Tabla 3.3.2.25 Tarea de Programación insertar persona**

<b>Tarea de Programación</b>	
No. Tarea: 25	No. Historia: 11
Nombre de la tarea: Insertar persona	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 18/06/2016	Fecha fin: 18/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite eliminar una persona	

**Tabla 3.3.2.26 Tarea de Programación modificar persona**

<b>Tarea de Programación</b>	
No. Tarea: 26	No. Historia: 11
Nombre de la tarea: Modificar persona	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 17/06/2016	Fecha fin: 18/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite modificar los datos de una persona	

**Tabla 3.3.2.18 Tarea de Programación listar personas**

<b>Tarea de Programación</b>	
No. Tarea: 26	No. Historia: 11
Nombre de la tarea: Listar persona	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 18/06/2016	Fecha fin: 18/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite listar los datos de las personas	

**Tabla 3.3.2.27 Tarea de Programación eliminar persona**

<b>Tarea de Programación</b>	
No. Tarea: 27	No. Historia: 11
Nombre de la tarea: Eliminar persona	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 18/06/2016	Fecha fin: 19/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite eliminar los datos de una persona	

**Tabla 3.3.2.28 Tarea de Programación insertar modificación**

<b>Tarea de Programación</b>	
No. Tarea: 28	No. Historia: 12
Nombre de la tarea: Eliminar persona	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 19/06/2016	Fecha fin: 19/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite insertar una modificación	

**Tabla 3.3.2.29 Tarea de Programación listar modificaciones**

<b>Tarea de Programación</b>	
No. Tarea: 29	No. Historia: 12
Nombre de la tarea: Listar modificaciones	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 19/06/2016	Fecha fin: 19/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite insertar una modificación	

**Tabla 3.3.2.30 Tarea de Programación eliminar modificaciones**

<b>Tarea de Programación</b>	
No. Tarea: 30	No. Historia: 12
Nombre de la tarea: Eliminar modificaciones	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 20/06/2016	Fecha fin: 20/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite eliminar una modificación	

**Tabla 3.3.2.23 Tarea de Programación eliminar modificaciones**

<b>Tarea de Programación</b>	
No. Tarea: 31	No. Historia: 12
Nombre de la tarea: Eliminar modificaciones	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 20/06/2016	Fecha fin: 21/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite eliminar una modificación	

**Tabla 3.3.2.32 Tarea de Programación mostrar estado de saldo mensual**

<b>Tarea de Programación</b>	
No. Tarea: 32	No. Historia: 13
Nombre de la tarea: Mostrar estado de saldo mensual	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 21/06/2016	Fecha fin: 21/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Muestra el estado de las cuentas	

**Tabla 3.3.2.25 Tarea de Programación mostrar declaración jurada**

<b>Tarea de Programación</b>	
No. Tarea: 33	No. Historia: 14
Nombre de la tarea: Mostrar declaración jurada	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 22/06/2016	Fecha fin: 22/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Muestra la declaración jurada	

**Tabla 3.3.2.34 Tarea de Programación mostrar nomenclador**

<b>Tarea de Programación</b>	
No. Tarea: 34	No. Historia: 15
Nombre de la tarea: Mostrar nomenclador	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 23/06/2015	Fecha fin: 23/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Muestra el nomenclador	

**Tabla 3.3.2.35 Tarea de Insertar Datos Generales**

<b>Tarea de Programación</b>	
No. Tarea: 35	No. Historia: 16
Nombre de la tarea: Insertar datos generales	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 23/06/2015	Fecha fin: 24/06/2015
Programador responsable: Orlando Prieto Silva	
Descripción: Permite insertar los datos de la empresa	

**Tabla 3.3.2.36 Tarea de modificar Datos Generales**

<b>Tarea de Programación</b>	
No. Tarea: 36	No. Historia: 16
Nombre de la tarea: Modificar datos generales	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 24/06/2016	Fecha fin: 24/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite modificar los datos de la empresa	

**Tabla 3.3.2.37 Tarea de Insertar Datos Generales**

<b>Tarea de Programación</b>	
No. Tarea: 37	No. Historia: 16
Nombre de la tarea: Ver datos generales	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 25/06/2016	Fecha fin: 25/06/2016
Programador responsable: Orlando Prieto Silva	
Descripción: Permite ver los datos de la empresa	

## Anexo 4: Pruebas de Aceptación

Tabla 3.4.1.1PA: Prueba para comprobar la entrada de un usuario al sistema.

Prueba de aceptación
Historia de usuario # 1: Autenticar usuarios
Nombre: Prueba para comprobar la entrada de un usuario al sistema.
Descripción: Validación de entrada de los datos del usuario.
Condiciones de ejecución: El usuario debe introducir su nombre de usuario y contraseña.
Entrada/Pasos de ejecución: El usuario escribe su nombre de usuario y contraseña y luego pulsa el botón "Entrar".
<p>Resultado esperado:</p> <ul style="list-style-type: none"> <li>➤ Si el usuario tiene acceso para entrar a la aplicación e inserta sus datos correctamente entrara sin problemas al sistema.</li> </ul> <p>Se emite un mensaje de error en caso de que:</p> <ul style="list-style-type: none"> <li>➤ Se inserten los datos de un usuario no valido para el sistema o datos incorrectos.</li> <li>➤ Se dé clic en el botón Entrar sin insertar nada en los campos de texto.</li> </ul>
Evaluación de la prueba: Aceptada.

**Tabla 3.4.1.2 PA: Prueba para comprobar la gestión de usuarios.**

<b>Prueba de aceptación</b>
Historia de usuario # 2: Gestionar usuarios
Nombre: Prueba para comprobar la gestión de usuarios.
Descripción: Validación de gestión de usuarios.
Condiciones de ejecución: El usuario debe entrar a la aplicación y tener el rol de Administrador para poder crear, modificar, listar y eliminar usuarios.
Entrada/Pasos de ejecución: El administrador escribe los datos del usuario que se piden para crear el usuario luego presiona el botón Crear. Después que el usuario esté insertado se podrá eliminar, modificar y acceder a sus datos en el listado de usuarios.
Resultado esperado: <ul style="list-style-type: none"> <li>➤ Si se introducen los datos correctamente se creará el nuevo usuario.</li> <li>➤ Si se desea eliminar el usuario se muestra un mensaje de confirmación</li> </ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"> <li>➤ Se inserten los datos de un usuario ya creado.</li> <li>➤ Se inserte en el campo Email un correo no válido.</li> <li>➤ Se dejen campos obligatorios vacíos.</li> </ul>
Evaluación de la prueba: Aceptada.

**Tabla 3.4.1.3 PA: Prueba para comprobar la gestión de transacciones.**

Prueba de aceptación
Historia de usuario # 3: Gestionar transacción
Nombre: Prueba para comprobar la gestión de transacciones.
Descripción: Validación de entrada de los datos de la transacción.
Condiciones de ejecución: El usuario debe estar autenticado y tener el rol de técnico o administrador.
Entrada/Pasos de ejecución: El Administrador escribe los datos que se piden de la transacción y pulsa el botón enviar. Después que la transacción esté insertada se podrá modificar y acceder a ella en la lista de transacciones.
<p>Resultado esperado:</p> <ul style="list-style-type: none"> <li>➤ Si los datos son correctos se inserta los datos de la transacción</li> </ul> <p>Se emite un mensaje de error en caso de que:</p> <ul style="list-style-type: none"> <li>➤ Se introduzcan datos inválidos.</li> <li>➤ Se dé clic en el botón Enviar sin introducir nada en los campos de texto.</li> <li>➤ Se introduzca un número de cheque que ya existe.</li> <li>➤ Se dejen campos obligatorios vacíos.</li> </ul>
Evaluación de la prueba: Aceptada.

**Tabla 3.4.1.4 PA: Prueba para comprobar la gestión de acápites.**

Prueba de aceptación
<b>Historia de usuario # 5 Gestionar Acápite</b>
Nombre: Prueba para comprobar la gestión de acápites.
Descripción: Validación de entrada de los datos de un acápite.
Condiciones de ejecución: El usuario debe estar autenticado y tener el rol de técnico o administrador.
Entrada/Pasos de ejecución: El usuario escribe los datos que se piden del acápite y pulsa el botón enviar. Después que el acápite esté insertado se podrá eliminar, modificar y acceder a él en el nomenclador.
Resultado esperado: <ul style="list-style-type: none"> <li>➤ Si los datos son correctos se inserta los datos del acápite.</li> </ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"> <li>➤ Se introduzcan datos inválidos.</li> <li>➤ Se dé clic en el botón Enviar sin introducir nada en los campos de texto.</li> <li>➤ Se introduzca un número de acápite que ya existe.</li> <li>➤ Se dejen campos obligatorios vacíos.</li> </ul>
Evaluación de la prueba: Aceptada.

**Tabla 3.4.1.5 PA: Prueba para comprobar la gestión de personas.**

Prueba de aceptación
<b>Historia de usuario # 6 Gestionar Persona</b>
Nombre: Prueba para comprobar la gestión de personas.
Descripción: Validación de entrada de los datos de una persona.
Condiciones de ejecución: El usuario debe estar autenticado y tener el rol de administrador.
Entrada/Pasos de ejecución El Administrador escribe los datos que se piden de la persona y pulsa el botón enviar. Después que la persona esté insertada se podrá eliminar, modificar y acceder a ella en la lista de personas.
Resultado esperado: <ul style="list-style-type: none"> <li>➤ Si los datos son correctos se inserta los datos del acápite.</li> </ul> Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"> <li>➤ Se introduzcan datos inválidos.</li> <li>➤ Se dé clic en el botón Enviar sin introducir nada en los campos de texto.</li> <li>➤ Se introduzca un número de chapa que ya existe.</li> <li>➤ Se dejen campos obligatorios vacíos.</li> </ul>
Evaluación de la prueba: Aceptada.

## Anexo 5: Diagramas de clases

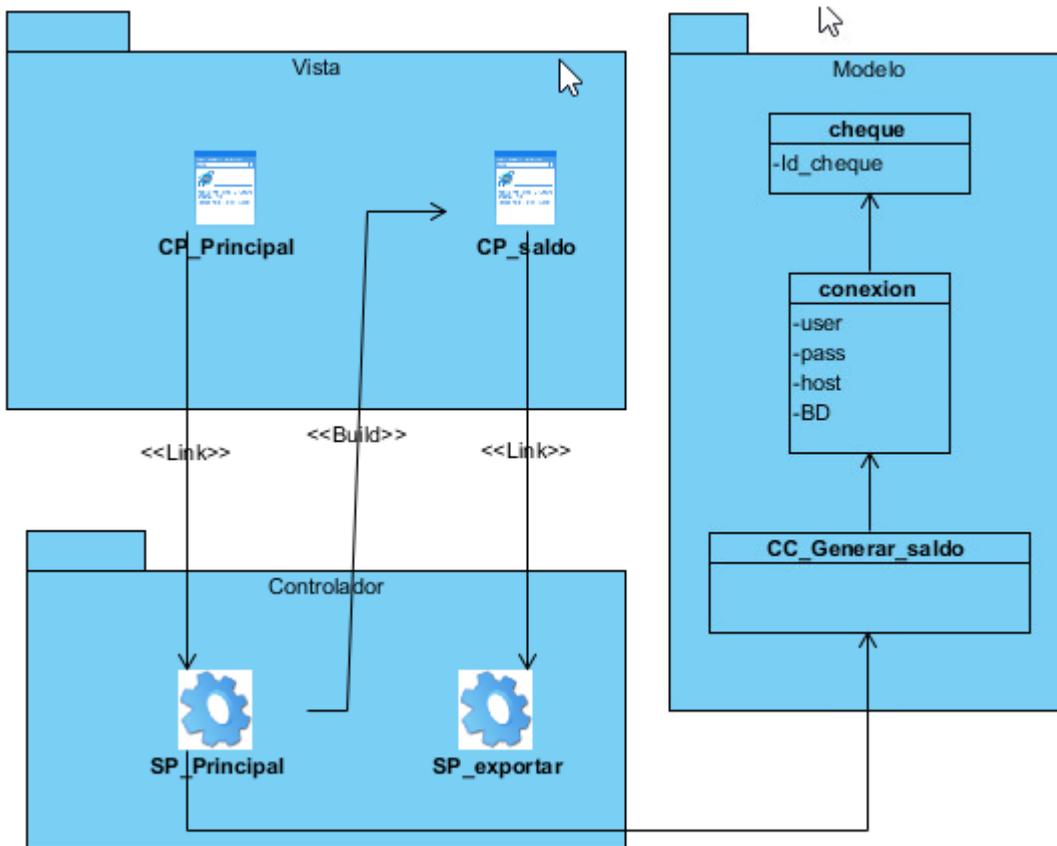


Figura 8 DCD Generar Saldo

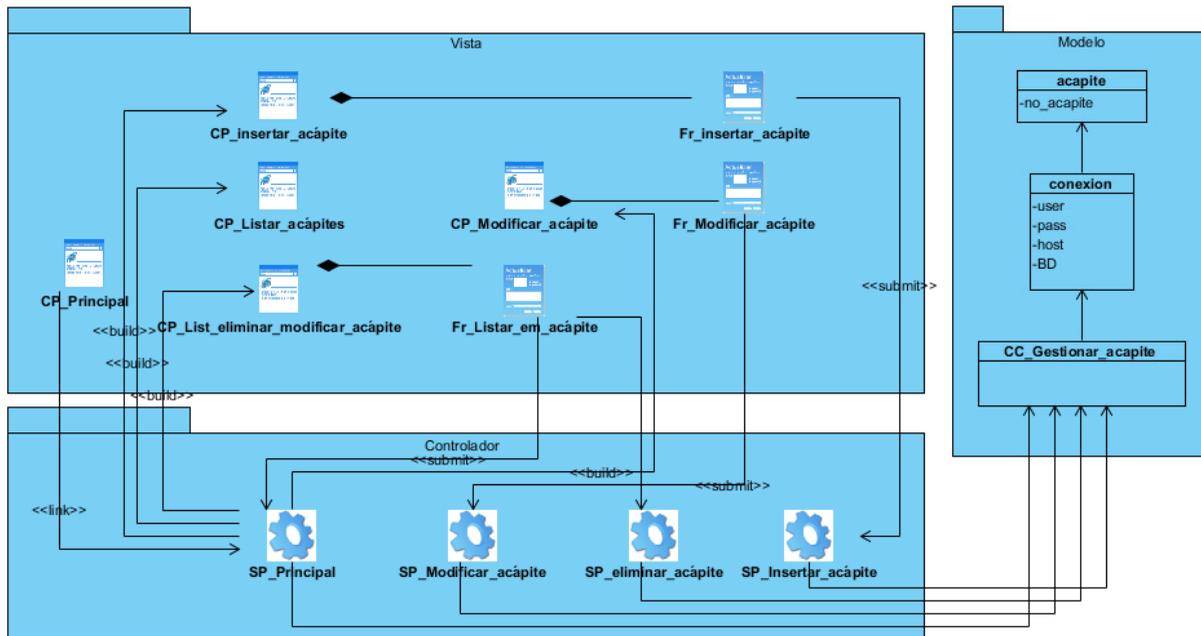


Figura 9 DCD Gestionar acápite

a

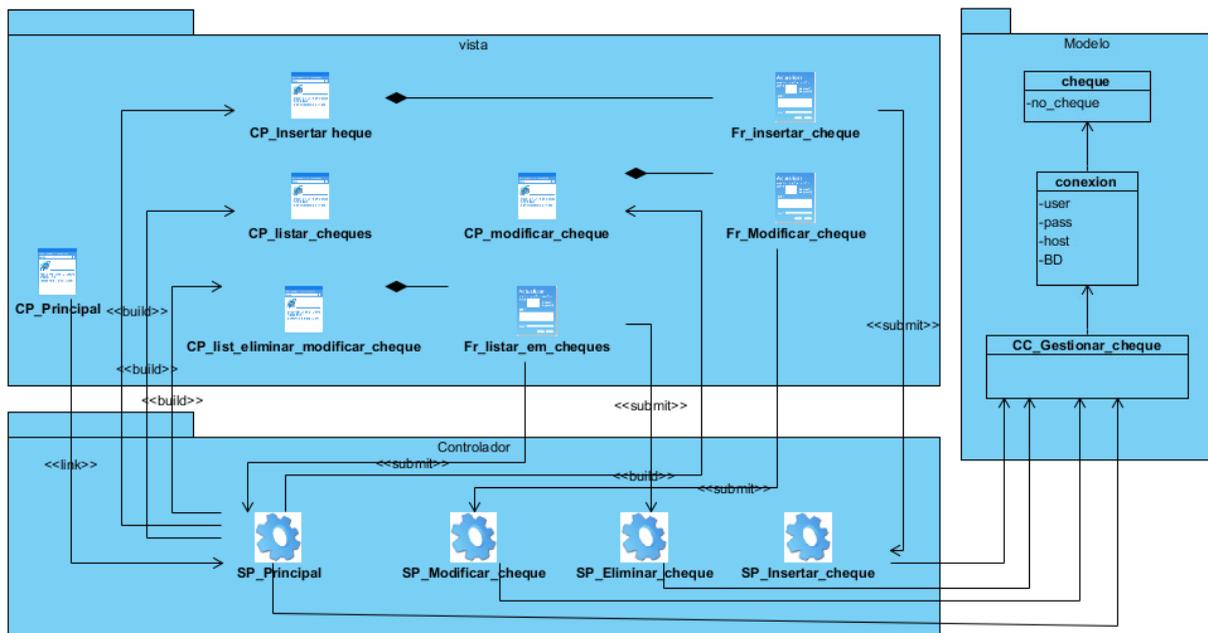


Figura 10 DCD Generar Cheque

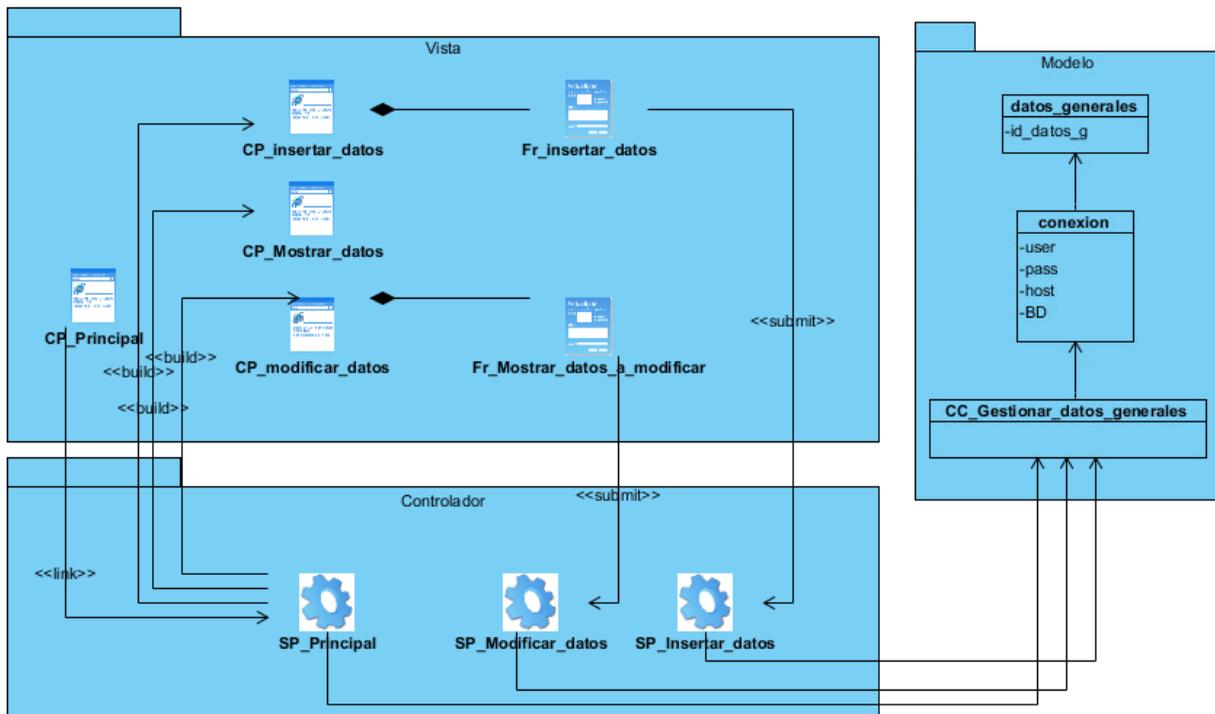


Figura 11 DCD Gestionar Datos Generales

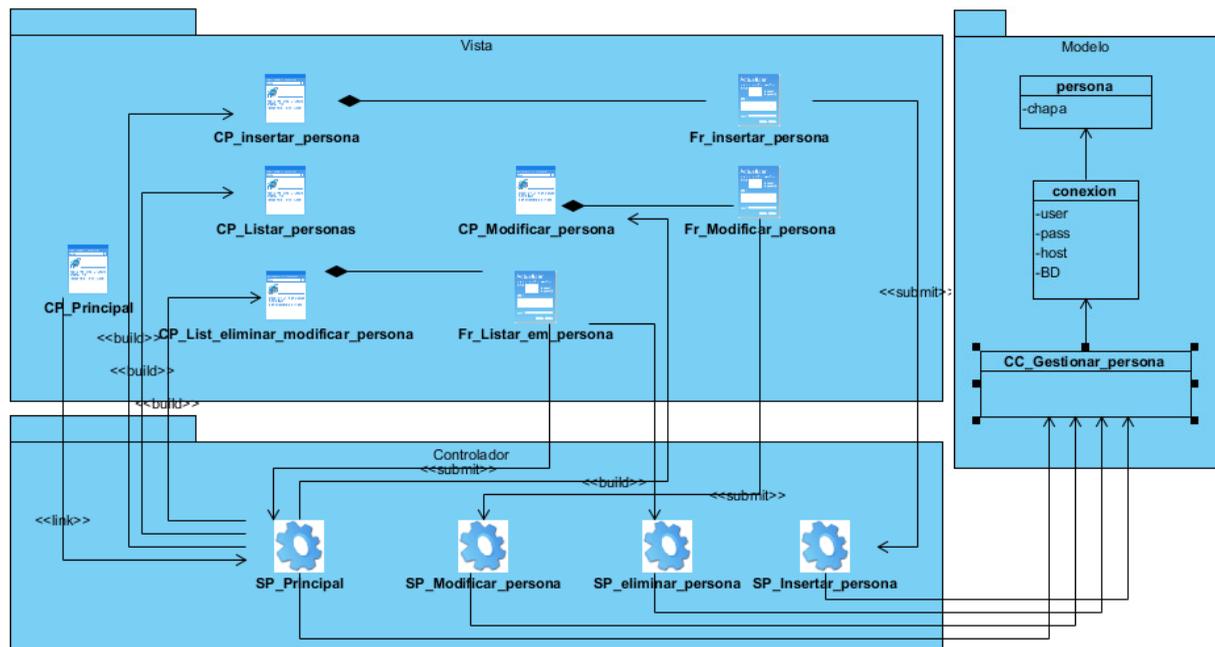


Figura 12 DCD Gestionar Persona

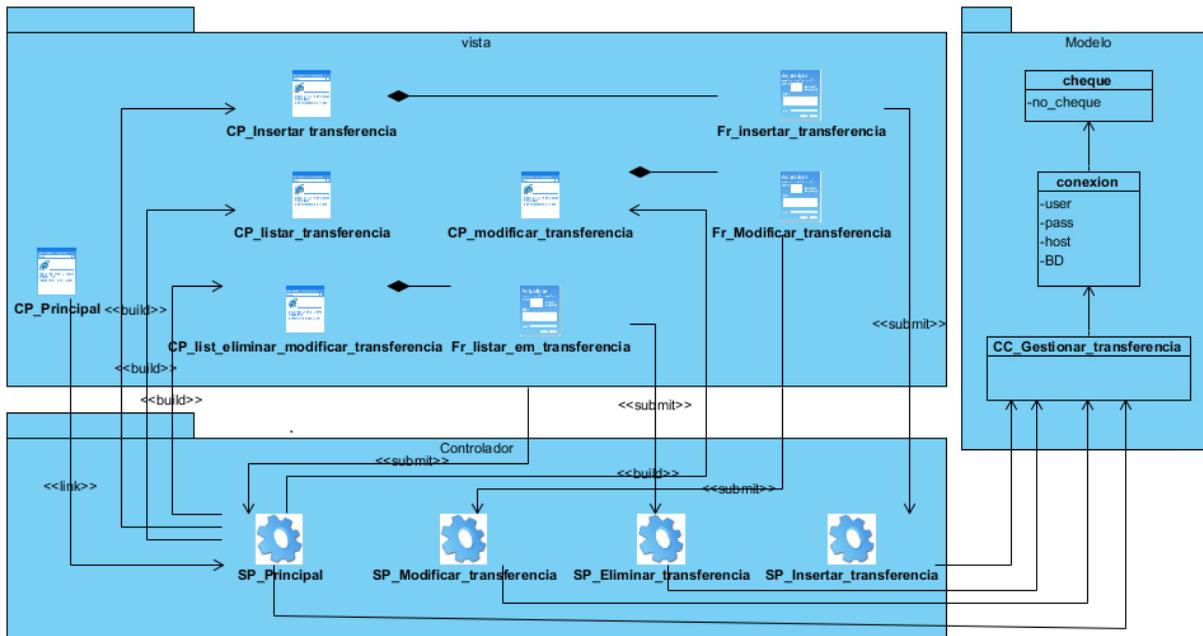


Figura 13 DCD Gestionar Transferencia

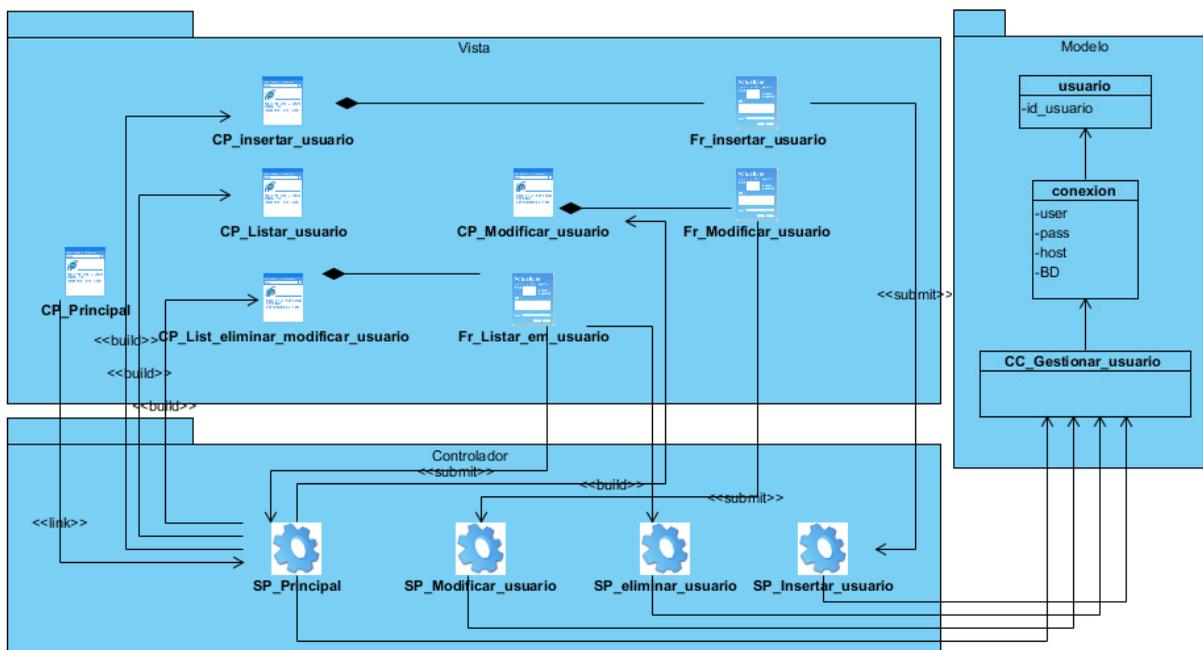


Figura 14 DCD Gestionar Usuario

## Anexo 6: Diagramas de secuencia

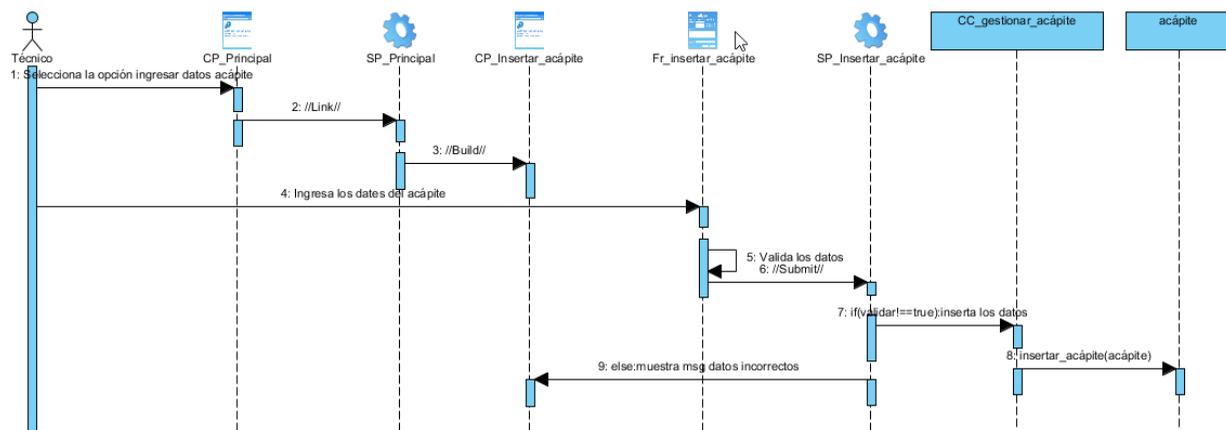


Figura 15 DS Insertar acápite

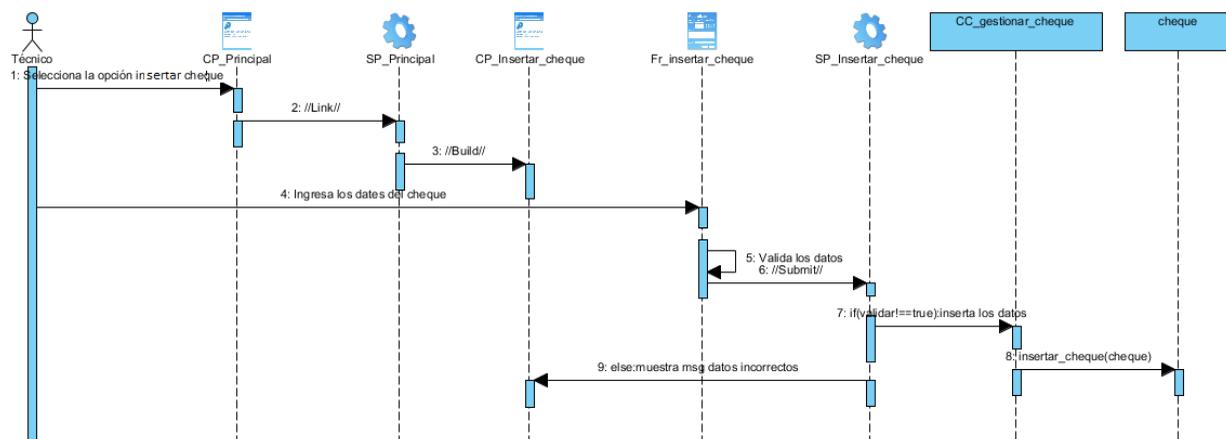


Figura 16 DS Insertar Cheque

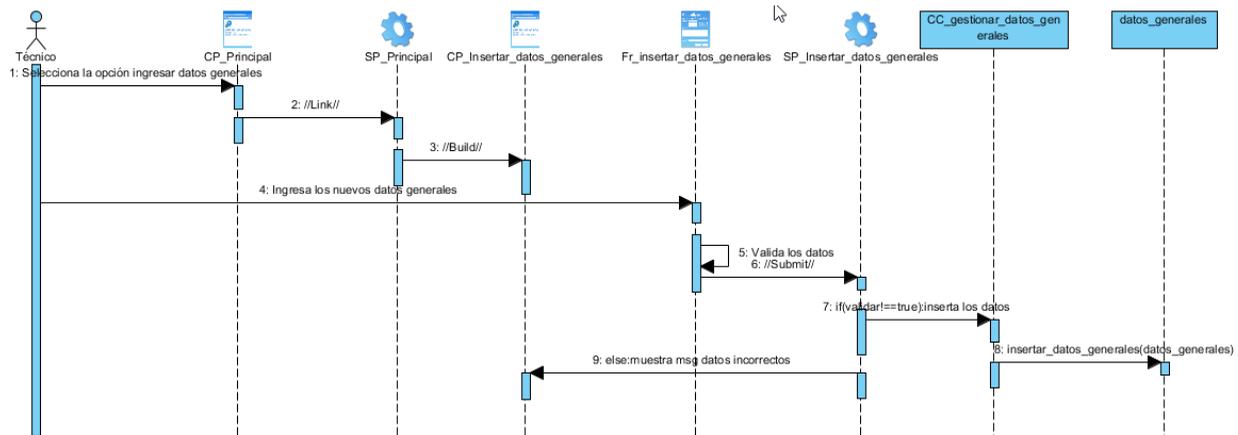


Figura 17 DS Insertar Datos generales

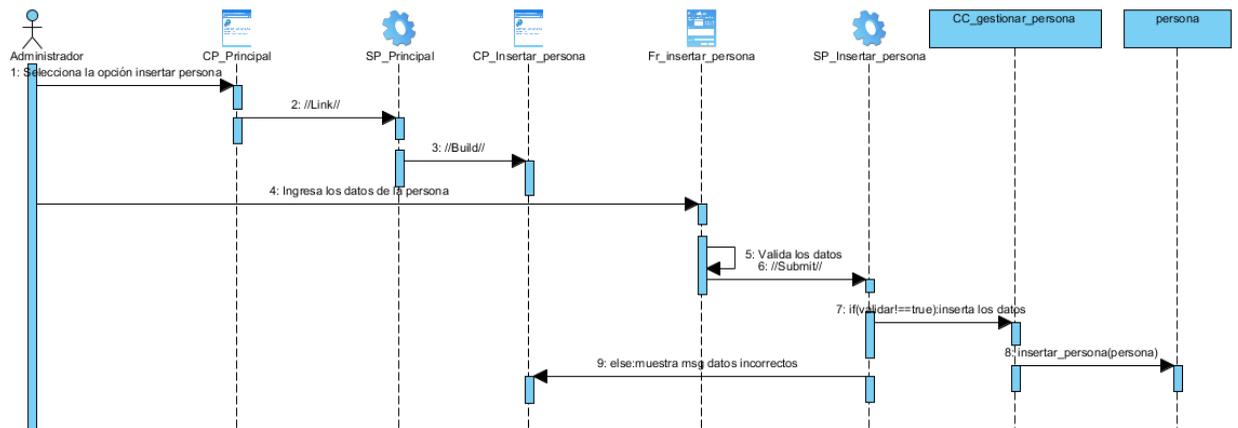


Figura 18 DS Insertar Persona

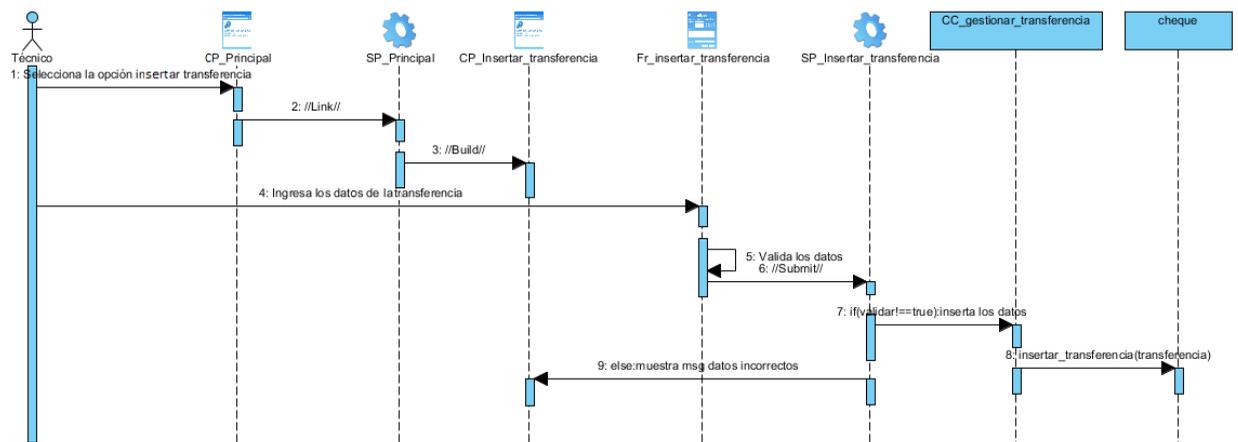


Figura 19 DS Insertar Cheque

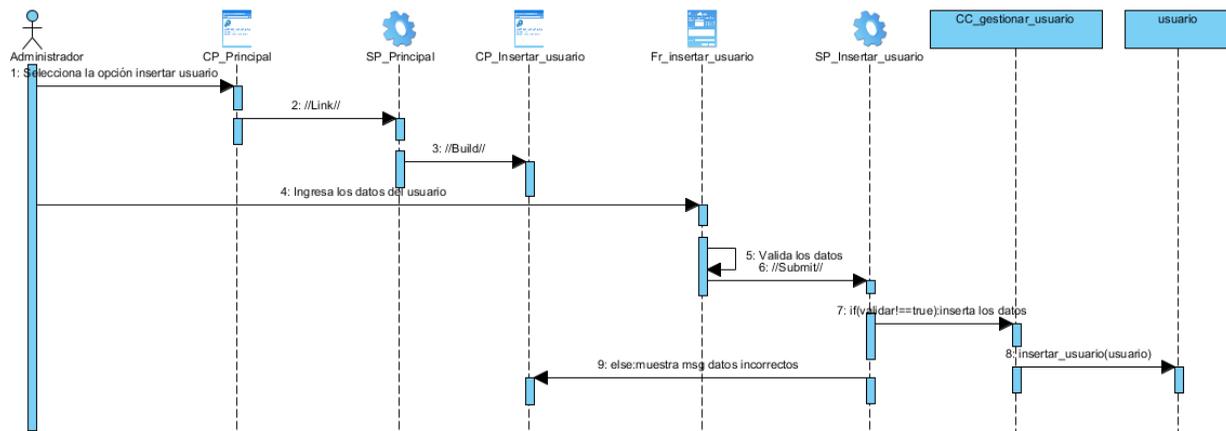


Figura 20 DS Insertar Usuario

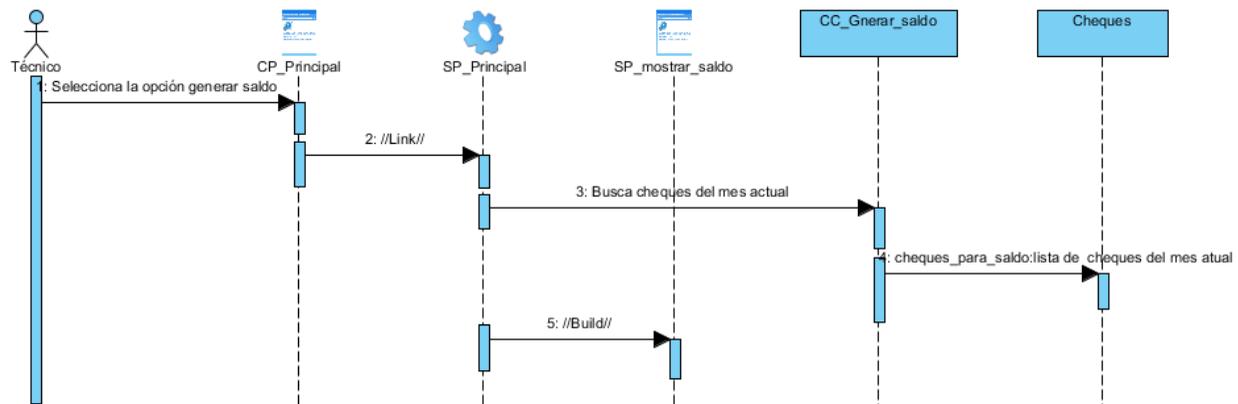


Figura 21 DS Generar Saldo Mensual

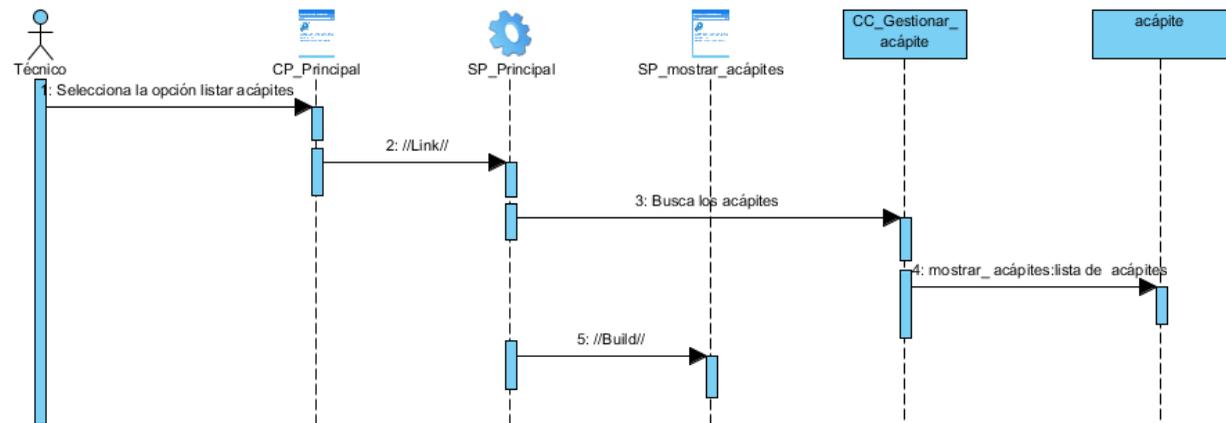


Figura 22 DS Listar Acapites

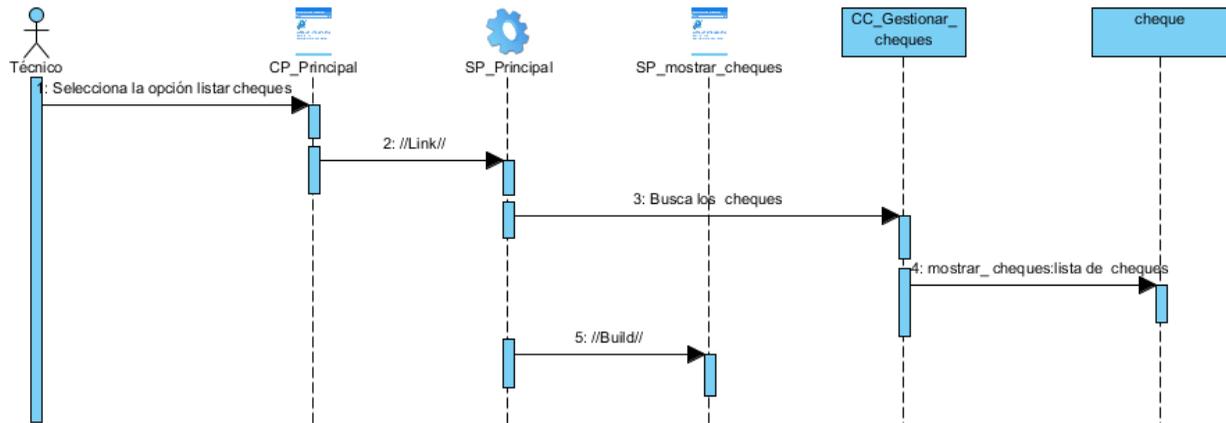


Figura 23 DS Listar Cheques

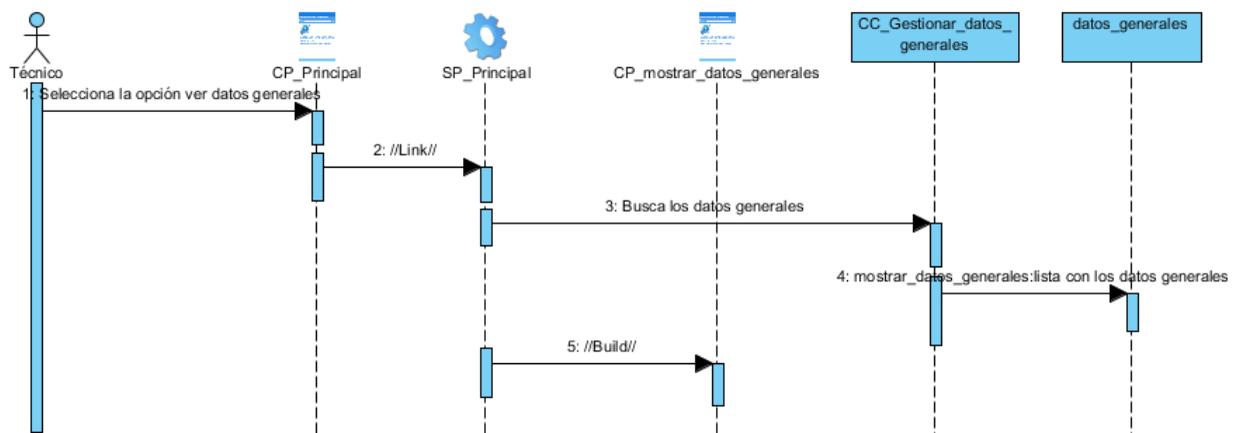


Figura 24 DS Listar Datos Generales

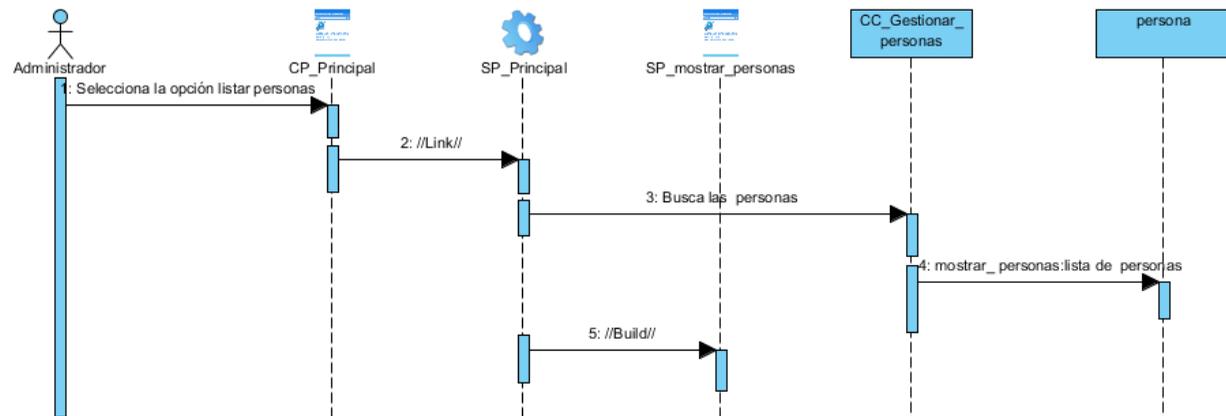


Figura 25 DS Listar Personas

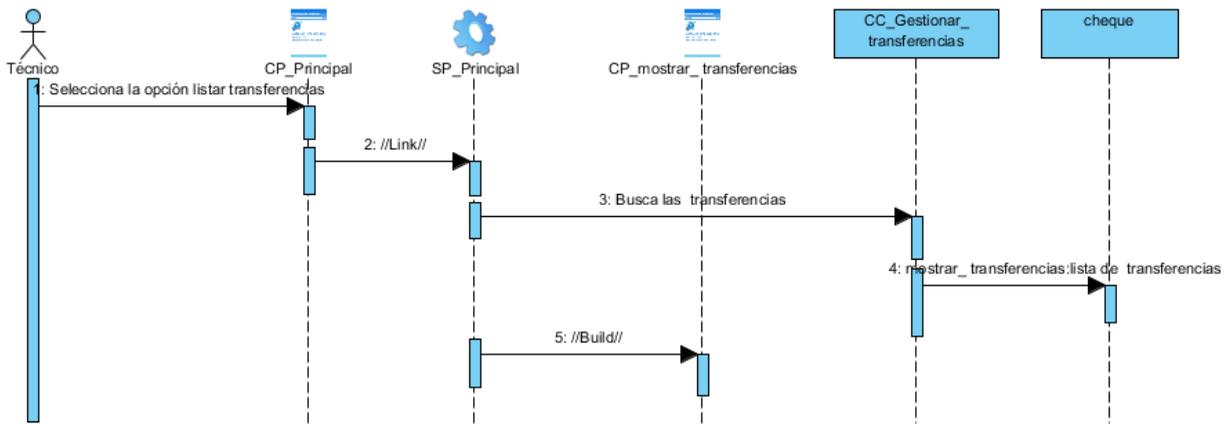


Figura 26 DS Listar Cheques

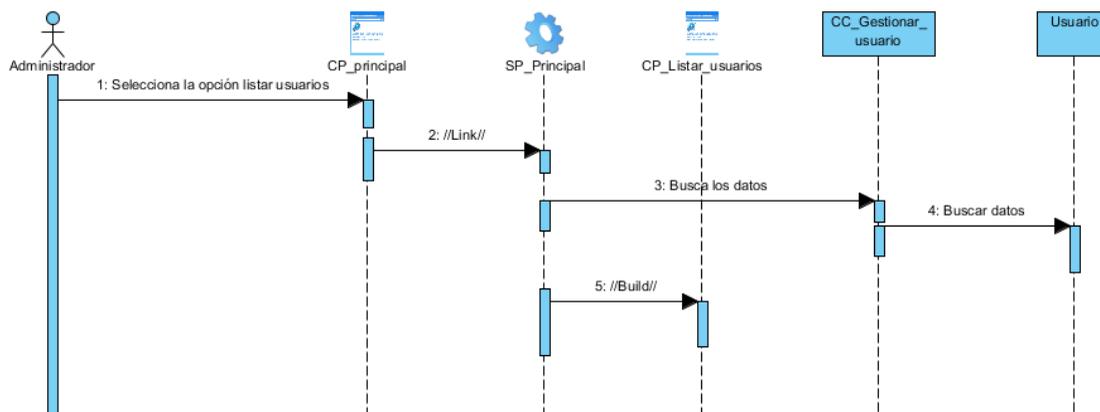


Figura 27 DS Listar Usuarios

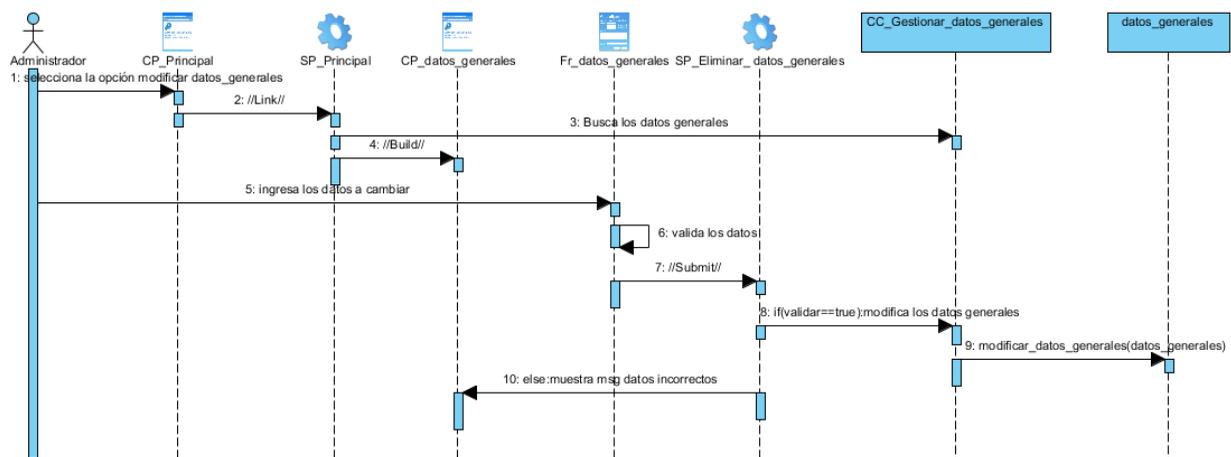


Figura 28 DS Modificar Datos Generales

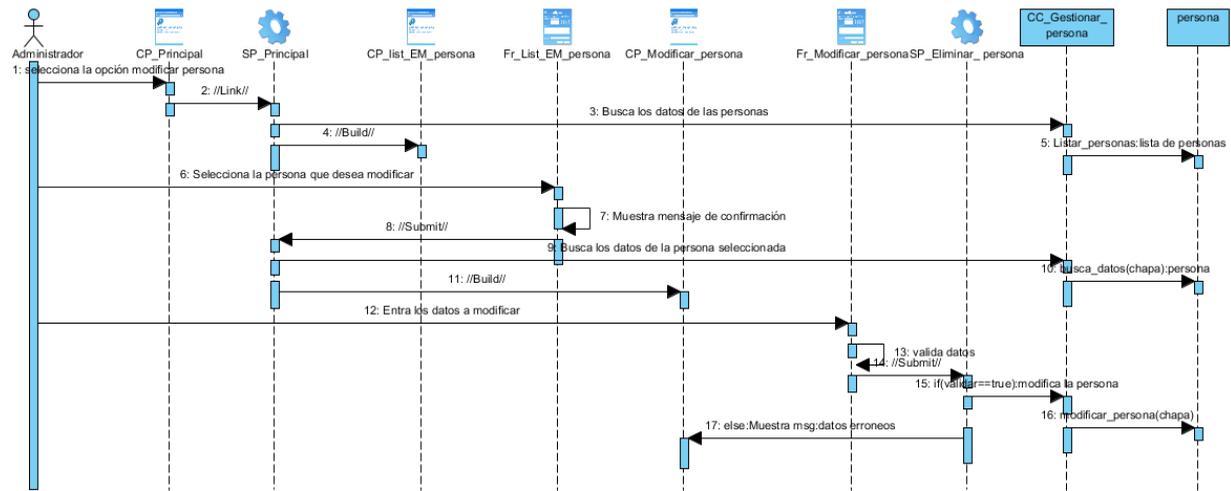


Figura 29 DS Modificar Persona

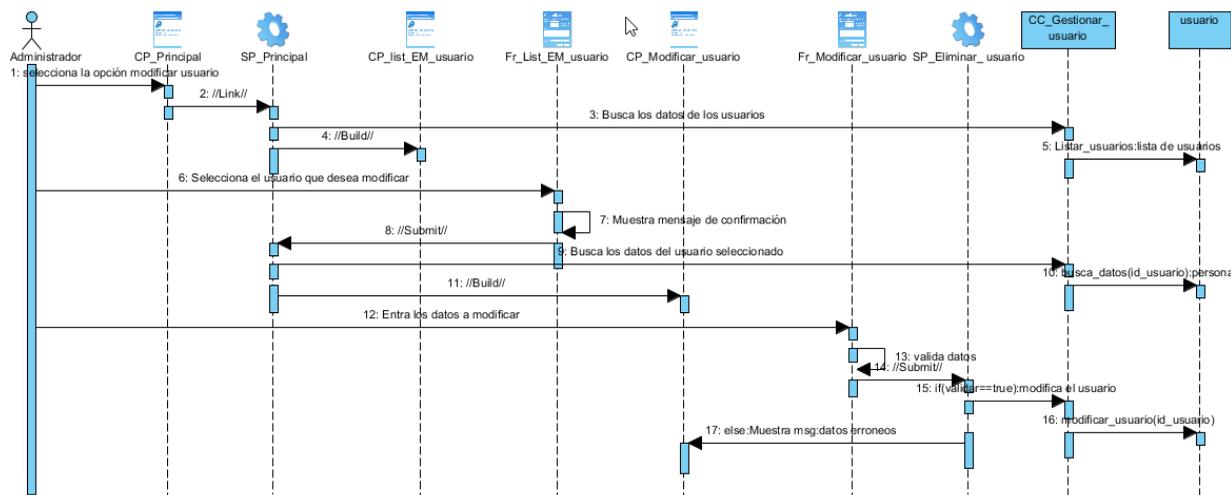


Figura 30 DS Modificar Usuario

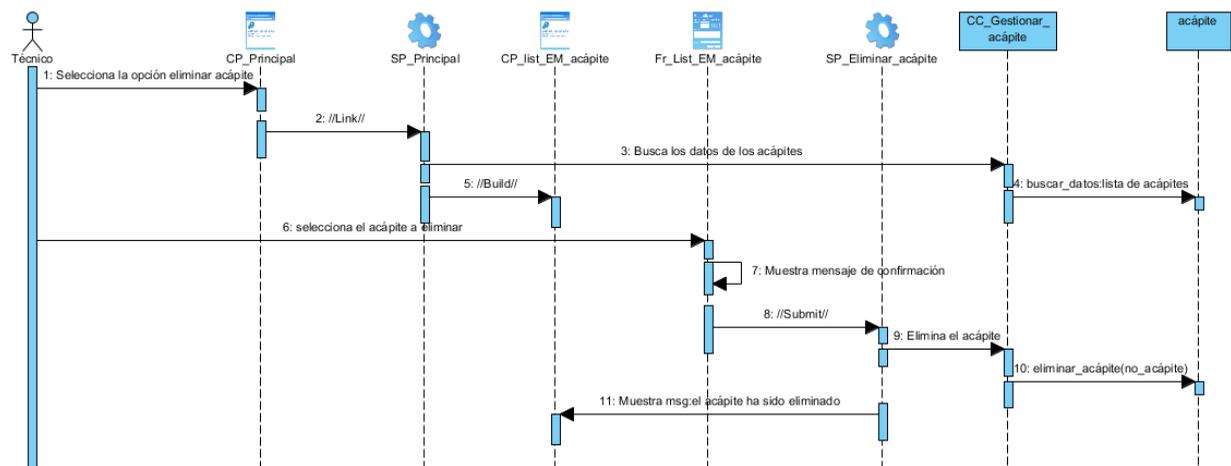


Figura 31 DS Eliminar Acápite

# ANEXOS

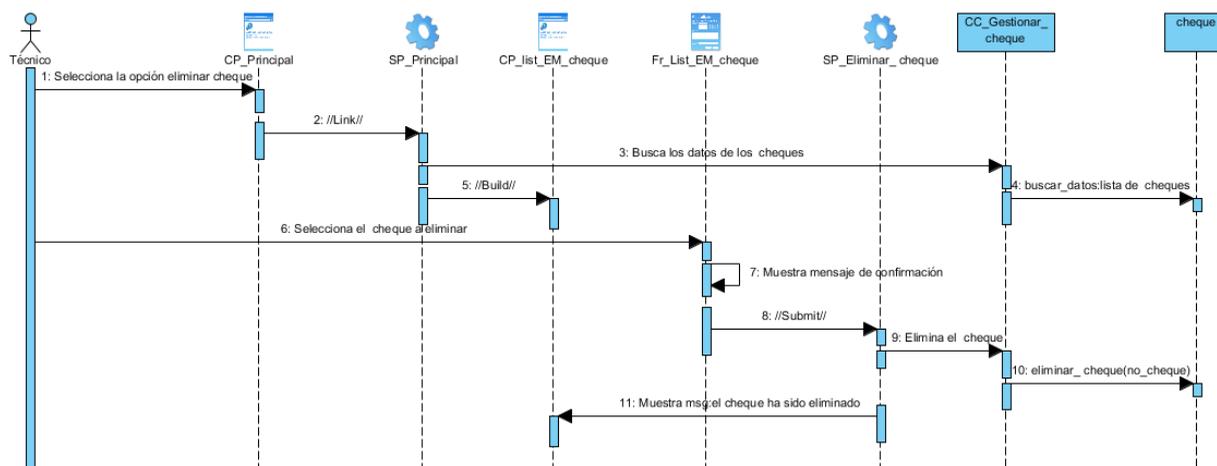


Figura 32 DS Eliminar Cheque

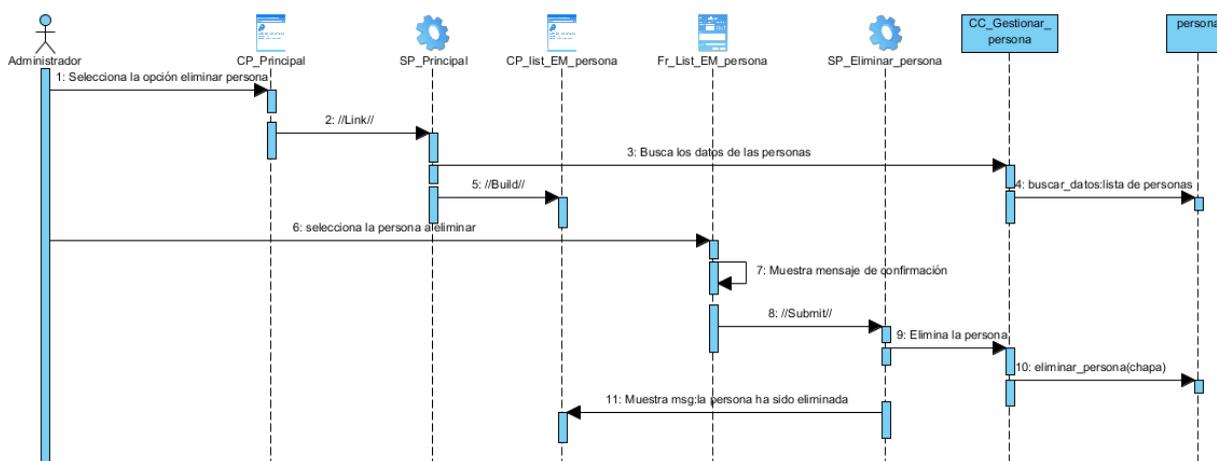


Figura 33 DS Eliminar Persona

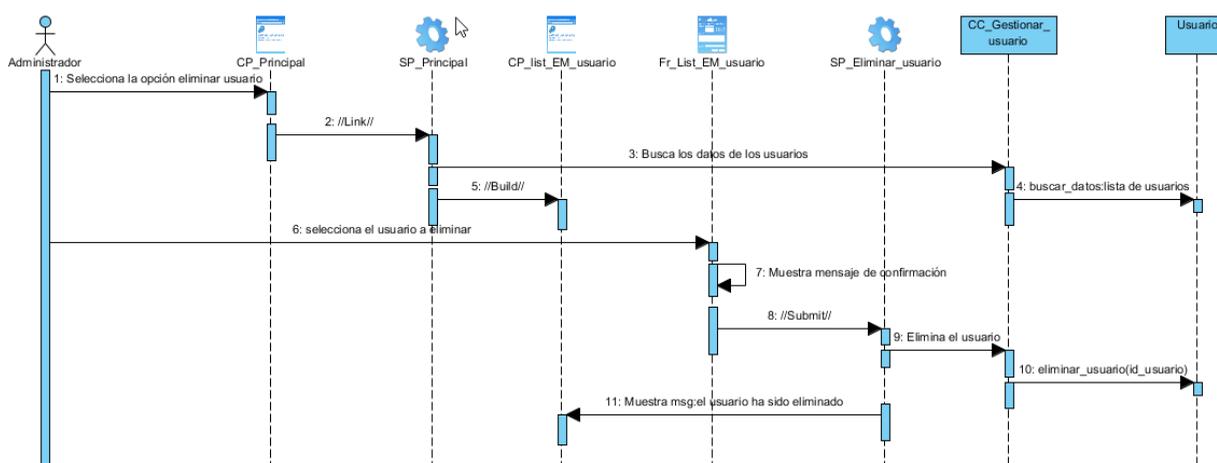


Figura 34 DS Eliminar Usuario

### Anexo 7: Diagramas de componentes

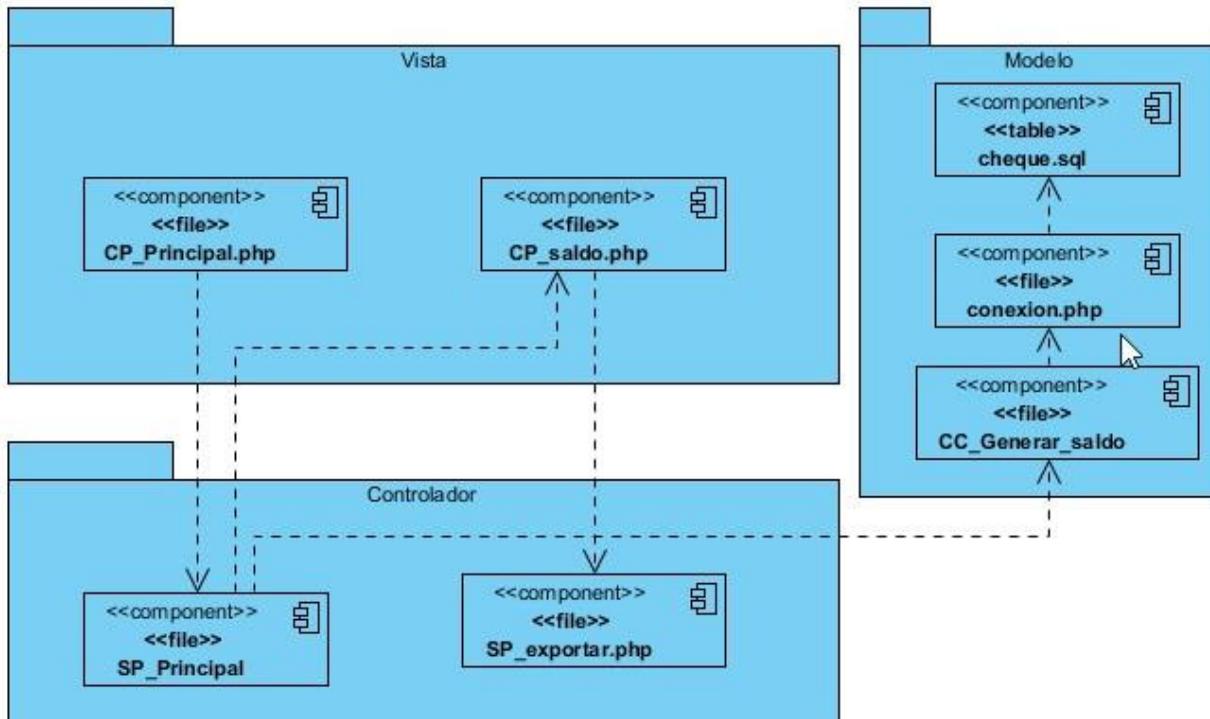


Figura 35 DC Generar Saldo Mensual

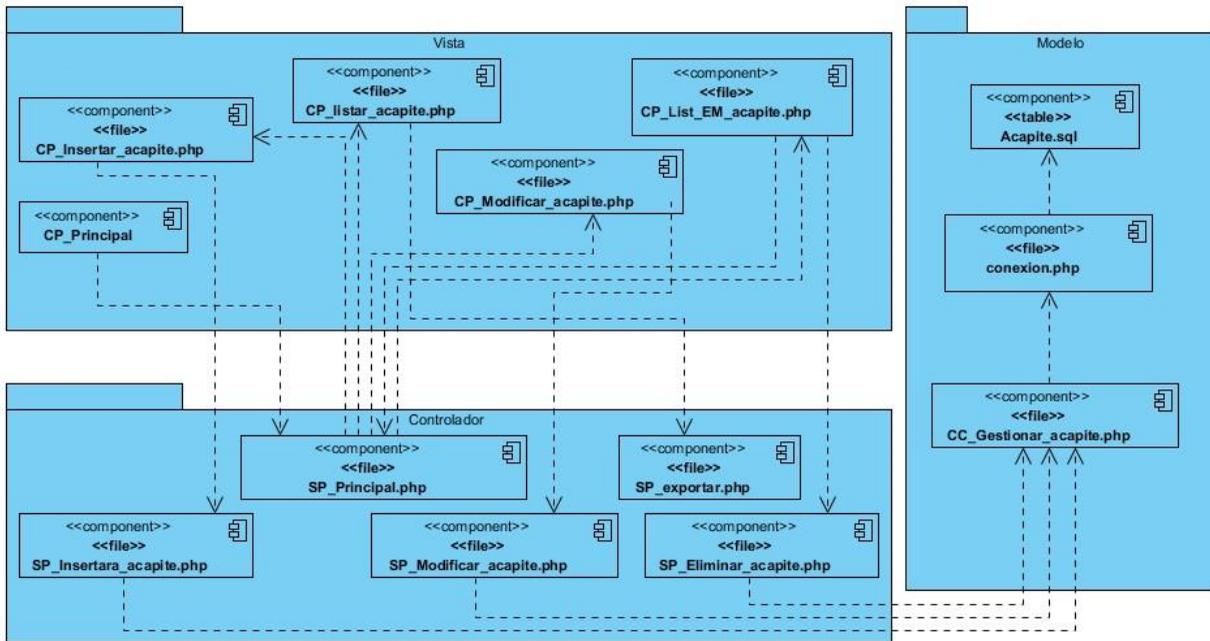


Figura 36 DC Gestionar Acapite

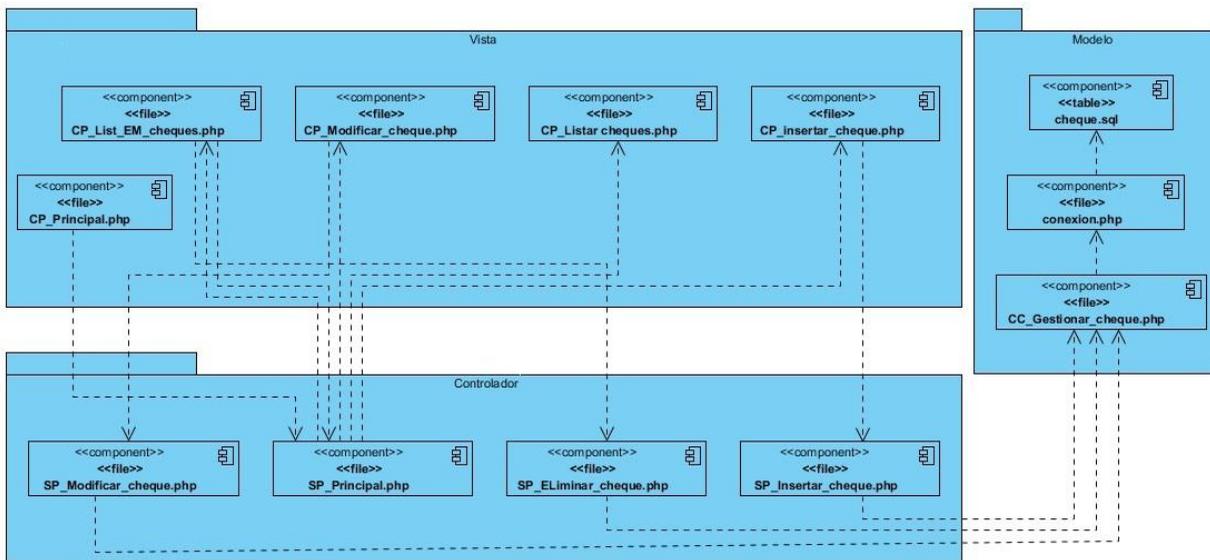


Figura 37 DC Gestionar Cheque

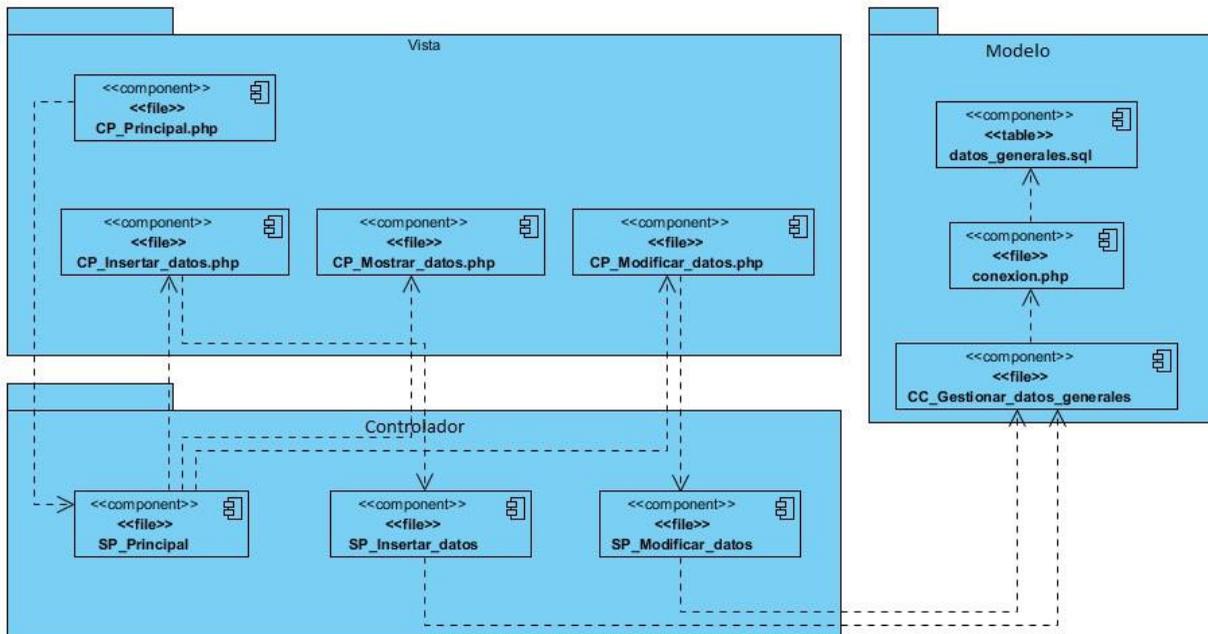


Figura 38 DC Gestionar Datos Generales

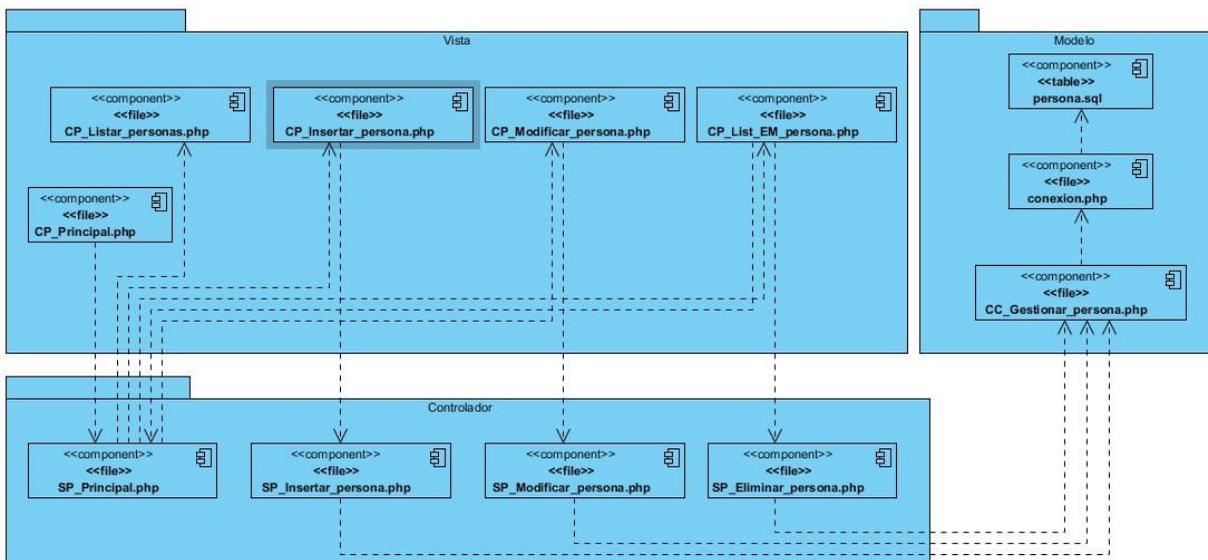


Figura 39 DC Gestionar Persona

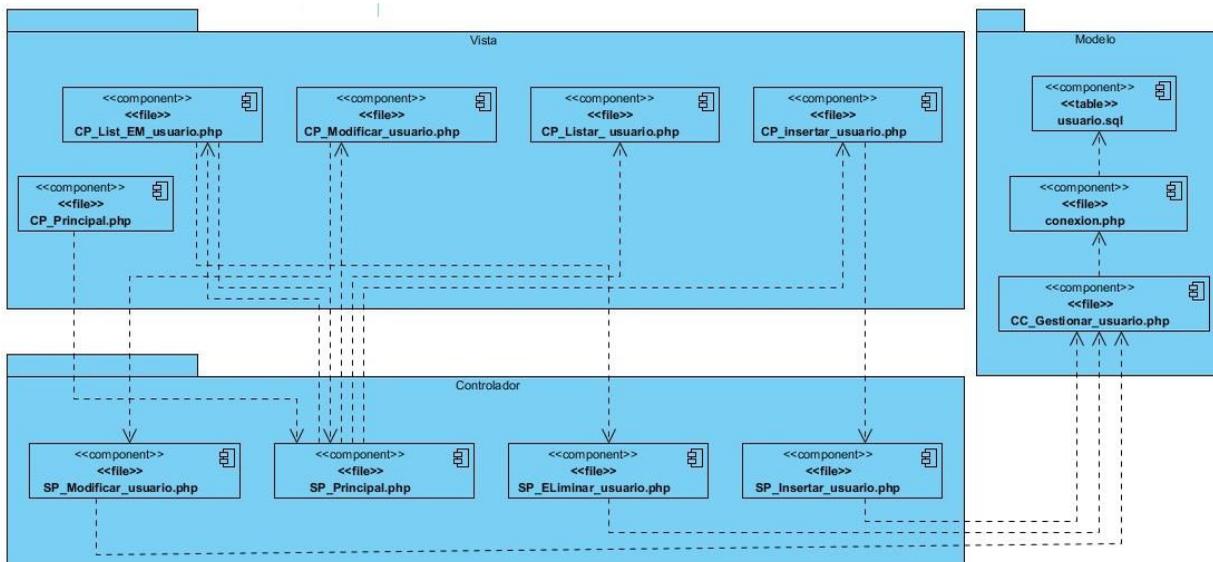


Figura 40 DC Gestionar Usuario

## Anexo 8: Interfaces de usuario

**Sistema para la Gestion de Documentos Contables**

Perfil

**UNE**  
 Ministerio de Energía y Minas  
 DECLARACIÓN JURADA SOBRE LA UTILIZACIÓN DE LOS RECURSOS FINANCIEROS  
 ENTIDAD: CTE Lidio Ramón Pérez CÓDIGO: 104-0-12648  
 POR MEDIO DE LA PRESENTE DECLARAMOS QUE LOS RECURSOS FINANCIEROS DE NUESTRA:  
 CUENTA BANCARIA EN CUC: 1271325060500020  
 EN EL MES DE: junio DEL AÑO: 2016  
 ASCIENDE A UN MONTO DE: \$ 7663.25  
 LOS MISMOS HAN SIDO UTILIZADOS DE LA SIGUIENTE FORMA:

Nºacápite	Descripción	Gastos del mes		Acumulado	
20500	Inversiones	\$	0	\$	1482
20503	Resto de equipos	\$	0	\$	634
2050319	Resto de equipos Cta 494 Prov	\$	0	\$	256
20504	Construcción y montaje	\$	0	\$	592
20800	Pago de deudas	\$	0	\$	1203
20800001	Pago de deudas vencidas al financiamiento	\$	0	\$	674
20800002	Financiamiento año plan (interes)	\$	0	\$	311
20800003	Crédito anticipado año plan (int)	\$	0	\$	218

Figura 41 Vista Generar declaración jurada

**Sistema para la Gestion de Documentos Contables**

Perfil

**ENTRE LOS DATOS DE LA TRANSFERENCIA**

\* Por Favor: llene este campo

No de transferencia:

Beneficiario: EMCE

Aápite:

Importe:

Fecha: 24/junio/2016

Consecutivo: 8

Tipo de operación:

Enviar

Figura 42 Vista Insertar transferencia

**Sistema para la Gestion de Documentos Contables**

SC

Persona Usuarios Cargos Modificaciones Perfil

**ENTRE LOS DATOS DEL USUARIO**

Nombre de usuario:  \* Por Favor llene este campo

Chapa:

Contraseña:

Repetir contraseña:

Privilegio:

Figura 43 Vista Insertar Usuario

**Sistema para la Gestion de Documentos Contables**

SC

Persona Usuarios Cargos Modificaciones Perfil

De este mes  
De este año

Nºde cheque	Acción	Importe	Acápite	Beneficiario	Fecha	Usuario
5876243	debitar	56	20800001	EMCE	19/junio/2016	misabel
63214	debitar	32	20504	ECO16	19/junio/2016	misabel
620	debitar	24	20800001	EMCE	19/junio/2016	misabel
458731	debitar	23	20800003	EMCE	19/junio/2016	misabel
23694	debitar	56	2050319	EMCE	19/junio/2016	misabel
25879	debitar	36	20800001	NICAROTEC	19/junio/2016	misabel
756342	debitar	45	20800003	ECO16	19/junio/2016	misabel
1245678	debitar	45	20503	EMCE	19/junio/2016	misabel
458756	debitar	23	20800003	EMCE	19/junio/2016	misabel
789572	debitar	75	20800002	ECO16	19/junio/2016	misabel
127456	debitar	45	2050319	ECO16	19/junio/2016	misabel
258463	debitar	75	20800001	EMCE	19/junio/2016	misabel
4237412	debitar	58	20800003	EMCE	19/junio/2016	misabel
124537	debitar	12	20800001	ECO16	19/junio/2016	misabel

1

Figura 44 Vista Modificaciones

**Sistema para la Gestion de Documentos Contables**

Perfil

**Credito** Debito **Generar** Acápites Datos generales

**UNE**  
 Unión Eléctrica de Cuba  
 CTE Lidio Ramón Pérez  
 Nomenclador

Nºacápites	Descripción
<b>20500</b>	<b>Inversiones</b>
20503	Resto de equipos
2050319	Resto de equipos Cta 494 Prov
20504	Construcción y montaje
20506	Otros
<b>20800</b>	<b>Pago de deudas</b>
20800001	Pago de deudas vencidas al financiamiento
20800002	Financiamiento año plan (interes)
20800003	Crédito anticipado año plan (int)
<b>20918000</b>	<b>Consumo productivo</b>
209180021	Materiales auxiliares generales
209180022	Laminados y tuberías
209180023	Pinturas
209180024	Gases industriales
209180027	Productos de alimentación animal
2091801	Materias primas y materiales
20918210	Pago por gasto de agua
<b>20987005</b>	<b>Servicios de comunicaciones</b>
209870051	Correo
209870052	Telefono y fax
209870053	Trunking y celulares
<b>20998063</b>	<b>Gastos de viajes nacionales</b>
209980631	Pasajes nacionales
209980632	alquiler de omnibus, microbus, taxis

Figura 45 Vista Generar Nomenclador