



UNIVERSIDAD SIMON BOLIVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN.

**IMPLEMENTACIÓN DE APLICACIÓN DE GESTIÓN DE
FIDEICOMISOS PARA SEGUROS MERCANTIL.**

Por
Hetsy Milagros Rodríguez Viafara.

INFORME DE PASANTIA LARGA.
Presentado ante la ilustre Universidad Simón Bolívar
como requisito parcial para optar al título de
Ingeniero en Computación.

Sartenejas, junio de 2017.



UNIVERSIDAD SIMON BOLIVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN.

**IMPLEMENTACIÓN DE APLICACIÓN DE GESTIÓN DE
FIDEICOMISOS PARA SEGUROS MERCANTIL.**

Realizado con la asesoría de:

Tutor Académico: Prof. Ana María Borges.

Tutor Industrial: Ing. Abel Laguna.

INFORME DE PASANTIA LARGA.

Presentado ante la ilustre Universidad Simón Bolívar

como requisito parcial para optar al título de

Ingeniero en Computación.

Sartenejas, junio de 2017.

ACTA DE EVALUACIÓN

IMPLEMENTACIÓN DE APLICACIÓN DE GESTIÓN DE FIDEICOMISOS PARA SEGUROS MERCANTIL.

Realizado por Hetsy Milagros Rodríguez Viafara.

RESUMEN

La Corporación Bimodal Technology (CBT) es una empresa orientada a proporcionar tecnologías innovadoras a la medida de las necesidades de sus clientes, con el propósito de incrementar su competitividad y productividad. En este particular, para cumplir con tales objetivos, se desarrolló la aplicación basada en la información de la Base de Datos del Departamento de Fideicomisos del Banco Mercantil, a través de la migración del sistema actual BFE, agregando funcionalidades básicas que contemplan el módulo de emisión de reportes. En su imperativo orientado a la reducción de costos y hacer más expedito el proceso de gestión de fideicomisos que maneja Seguros Mercantil y como parte de un proceso global que opera en la actualidad con el sistema *Main Frame*, asociado a un lenguaje natural y base de datos ADABAS, se decidió migrar el aplicativo a una nueva plataforma de esquema cliente-servidor con el propósito de mejorarlo y cumplir con los estándares de vanguardia del mercado. Este informe tiene como objetivo académico, documentar el Proyecto de Pasantía efectuado en el período Enero-Mayo 2017, el cuál consistió en la implementación de nuevas funcionalidades al Módulo de Reportes, Manejo de Entradas y Salidas de Dinero, así como, la implementación funcional para la distribución a cada beneficiario de sus ganancias o pérdidas del período. En ese particular, se consideraron para su ejecución varias fases a saber: a) Adaptación, enfocada en el análisis del problema y funcionamiento de la empresa. b) Levantamiento de requerimientos centrados en la recolección y análisis de necesidades del sistema. c) Diseño. d) Implementación. e) Pase a producción, y la elaboración de entregables. En cuanto a la metodología, se apoyó en SCRUM para el proceso de desarrollo de software, la cual permitió como producto final la construcción de módulos, los resultados obtenidos de la aplicación cumplieron con los deseos y expectativas de la empresa, evidenciándose en las pruebas efectuadas un sencillo manejo y desempeño adecuado de los mismos por parte de los usuarios. No obstante, conviene aclarar que los cambios resultantes de las pruebas de regresión escapan al alcance del proyecto, y quedan de parte de la gerencia de CBT.

Palabras claves: aplicación cliente-servidor, desarrollo de software, fideicomiso, SCRUM.

DEDICATORIA

Dedico este logro enteramente a mi madre Isabel Cristina, quien me motivó no importa el lugar donde se encuentre. Me dejó en esta tierra muy rápido si se puede decir. Todos los días me hace falta y pienso en ella, pero no quiere decir que he dejado de sentir que está conmigo.

Su amor y sus enseñanzas me formaron en la persona que soy hoy en día, me educaste para luchar sola y gracias a eso he podido salir adelante, y para ella dedico uno de mis más grandes triunfos, para que donde se encuentre se pueda sentir orgullosa.

Aunque el tiempo jugó en contra nuestra, sentiré tu apoyo todos los días de mi vida, hasta que llegue la hora de encontrarnos nuevamente.

Te Amo mamá ahora y siempre.

AGRADECIMIENTOS

Primeramente a Dios por guiarme en este camino y no abandonarme en los momentos difíciles cuando se pudo haber perdido la esperanza y la fe.

A mi madre Isabel por haberme enseñado todo lo que pudo en el poco tiempo que pudimos compartir. Quizás en un principio pude haber pensado que me abandonó, pero no fue así, esta fue la prueba que nos tocó y poco a poco la hemos ido superando.

A Jonathan Moreno quien más que mi amigo es mi hermano por elección, gracias por adoptarme en tu vida y no dejarme caer esas veces que quise renunciar. Por brindarme una familia de la que cada día aprendo más.

A mi madre adoptiva Dionisia Herrera, quien me ha llevado por el camino de la racionalidad y del pensamiento a futuro para ser una mejor mujer cada día. Me enseña que no importa el tamaño del obstáculo, hay que trabajar con ánimos e insistencia.

A mi padres adoptivos Sandra Correa y Jhonathan Meléndez, de quien he aprendido a amar a quienes no llevan el mismo apellido o tipo de sangre y a buscar la perseverancia cuando se está perdido en el camino.

Mis mejores amigos que son parte de mi familia por elección Adriana González, Alejandra Preciado, Andel Núñez, Fernando Zerpa y Jhosbert Contreras, de quienes he aprendido que no importa lo diferentes que podamos ser, siempre nos vamos a apoyar en las buenas y las malas. No importa la distancia, siempre nos sentiremos cerca y siempre habrá consejos y colaboración entre nosotros. Agradezco saber el verdadero significado de una familia.

Los Profesores de la Universidad Simón Bolívar, quienes han dejado huella en mi camino más allá del ámbito académico, me demostraron la humildad de sus personalidades bañada de la experiencia en la trayectoria de una profesión.

Los Empleados que dan vida al campus Rosa Zambrano, Yurbi Ruscella, Betty Parra, Ninoska Cruz, Dianela Quiñones, mujeres que me demostraron un gran cariño y apoyo en muy poco tiempo y me mostraron sus versiones del día a día, de manera de yo poder aprender a construir la mía.

Agradezco a todos los que me hicieron ver el sentimiento del amor en todas sus expresiones, pues de verdad es lo que mueve al mundo más que cualquier otra cosa. Mi padre Said Jonás pudo demostrarme ese sentimiento en compensación a muchos aspectos, mi hermano Francisco Said me lo demuestra no importan los kilómetros que nos separan, y YC me enseñó a canalizar ese sentimiento para lograr mis metas y sueños.

La vida no me alcanzará para agradecerles.

LISTA DE ABREVIATURAS

CBT

Corporación Bimodal Technology, C.A.

BD

Base de Datos.

DAS

Documento de Arquitectura de Software.

IDE

Integrated Development Environment. Entorno de desarrollo integrado.

JAVA EE

Java Enterprise Edition.

SQL

Structured Query Language. Lenguaje de programación de consultas estructurado.

UML

Unified Modeling Language. Lenguaje unificado de modelaje.

HTTP

Hypertext Transfer Protocol. Protocolo de transferencia de archivos de hipertexto.

PDF

Portable Document Format. Formato de almacenamiento para documentos digitales independiente de plataformas de software o hardware.

HTML

Hypertext Transfer Protocol. Protocolo de transferencia de archivos de hipertexto.

XLS

Microsoft Excel file format. Extensión de archivo de los ficheros de la aplicación ofimática MS Excel (del inglés: MS Excel file extension)

CSV

Comma-Separated Values. Tipo de documento en formato abierto para representar datos en forma de tabla.

XML

Extensible Markup Language. Lenguaje de etiquetas.

API

Application Programming Interface. Interfaz de programación de aplicaciones.

FSOSEG

Aplicación de Fideicomiso Seguros Mercantil.

ADABAS

Adaptable Database System. Base de datos de listas invertidas de alto rendimiento.

MCU

Modelo de Casos de Uso.

ÍNDICE GENERAL

| | |
|--|-------|
| RESUMEN | iv |
| DEDICATORIA..... | v |
| AGRADECIMIENTOS..... | vi |
| LISTA DE ABREVIATURAS | vii |
| ÍNDICE DE FIGURAS | ix |
| INTRODUCCIÓN..... | 1 |
| CAPÍTULO 1..... | 6 |
| 1.1 Reseña Histórica de la Empresa | 6 |
| 1.2 Ubicación y Logo de la empresa..... | 7 |
| 1.3 Misión..... | 7 |
| 1.4 Visión | 7 |
| 1.5 Estructura Organizacional de la Empresa..... | 7 |
| CAPÍTULO 2..... | 9 |
| 2.1 Conceptos Básicos de Fideicomiso..... | 9 |
| 2.1.1 Fideicomiso | 9 |
| 2.1.2 Fideicomisos para Personas Jurídicas. | 10 |
| 2.1.3 Fideicomitente | 10 |
| 2.1.4 Fiduciario | 10 |
| 2.1.5 Beneficiario | 11 |
| 2.1.6 Rendimiento | 11 |
| 2.2 Conceptos relacionados con el desarrollo de software | 11 |
| 2.2.1 Arquitectura Tres Capas | 11 |
| 2.2.2 Arquitectura Cliente/Servidor | 12 |
| 2.2.3 Java | 13 |
| 2.2.4 JasperReport..... | 13 |
| 2.2.5 Servidor Apache HTTP | 14 |
| 2.3 Metodología para el desarrollo de Software | 15 |
| 2.3.1 Estructura de SCRUM | 15 |

| | |
|---|----|
| CAPÍTULO 3..... | 20 |
| 3.1 Naturaleza del Proyecto | 20 |
| 3.2 Fases de la metodología | 21 |
| 3.2.1 Fase 1: Adaptación. | 21 |
| 3.2.2 Fase 2: Levantamiento de Requerimientos. | 22 |
| 3.2.3 Fase 3: Diseño..... | 22 |
| 3.2.4 Fase 4: Implementación. | 23 |
| 3.2.5 Fase 5: Pase a Producción. | 23 |
| 3.2.6 Fase 6: Elaboración de Entregables..... | 24 |
| CAPÍTULO 4..... | 25 |
| 4.1 Fase de Adaptación. | 25 |
| 4.2 Fase de Levantamiento de Requerimientos..... | 26 |
| 4.3 Fase de Diseño..... | 27 |
| 4.4 Fase de Implementación. | 32 |
| 4.4.1 Primera Iteración: Login y Módulo de Transacciones. | 33 |
| 4.4.2 Segunda Iteración: Distribución de Rendimiento por Contrato y Beneficiario..... | 45 |
| 4.4.3 Tercera Iteración: Manejo de Reportes y Generación de Trazas de Auditoría. | 48 |
| 4.4.4 Resumen de Funcionalidades Desarrolladas..... | 53 |
| 4.5 Fase de Pase a Producción. | 53 |
| 4.5.1 Primera Iteración: Preparación de Ambiente de Implantación..... | 54 |
| 4.6 Fase de Elaboración de Entregables..... | 55 |
| CONCLUSIONES Y RECOMENDACIONES | 56 |
| REFERENCIAS BIBLIOGRÁFICAS | 60 |
| APÉNDICES..... | 62 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1.- Logo de CBT [1] | 7 |
| Figura2.- Estructura organizacional de CBT..... | 8 |
| Figura3.-Diagrama del ciclo de vida de SCRUM [15]..... | 16 |
| Figura4.- Modelo de Casos de Uso General..... | 29 |
| Figura 5.- Modelo de Casos de Uso. Paquete de Gestionar Módulo de Contabilidad..... | 30 |
| Figura 6.- Diagrama de Clases. | 31 |
| Figura 7.- Modelo Conceptual | 31 |
| Figura 8.- Diagrama de Implantación..... | 32 |
| Figura 9.- Diagrama de Implementación..... | 32 |
| Figura 10.- Insertar, Modificar, Anular Transacción..... | 35 |
| Figura 11.- Búsqueda de Transacciones..... | 36 |
| Figura 12- Gestor de Tipo de Transacciones..... | 38 |
| Figura 13.- Gestor de Plan Contable..... | 38 |
| Figura 14.- Gestor de Asientos Contables..... | 39 |
| Figura 15.- Gestor de Contabilidad..... | 39 |
| Figura 16- Gestor de Beneficiarios..... | 40 |
| Figura 17- Gestor de Fideicomisos..... | 40 |
| Figura 18- Ingreso a la Aplicación..... | 41 |
| Figura 19.- Ingreso fallido a la Aplicación..... | 42 |

| | |
|---|----|
| Figura 20- Ingreso exitoso a la Aplicación..... | 43 |
| Figura 21.- Gestión de Usuarios..... | 44 |
| Figura 22.- Gestión de Perfiles..... | 44 |
| Figura 23.- Gestión de Acciones de Usuario..... | 45 |
| Figura 24.- Ingreso de Asignación Diaria y Rendimiento..... | 47 |
| Figura 25.-Ventana de Cierre Diario..... | 47 |
| Figura 26.-Generar Reportes..... | 49 |
| Figura 27.-Generar Reportes- Balance de Comprobación..... | 49 |
| Figura 28.-Generar Reportes-Mov Mensual..... | 50 |
| Figura 29.-Visor de Log de Auditoría..... | 51 |
| Figura 30.-Vista de Instalación del Sistema..... | 52 |
| Figura 31.-Vista de Configuración del Sistema..... | 52 |
| Figura 32.- Resumen de Funcionalidades Desarrolladas | 53 |

INTRODUCCIÓN

Antecedentes

En la actualidad cualquier empresa, independientemente de su área de trabajo, tiene como imperativo ser competitiva y posicionarse en un buen lugar en el mercado. La necesidad de optimizar la producción, reducir los costos y poder cumplir con los requerimientos regulatorios, como la existencia de un sistema para la reducción de demoras en la realización de los trámites solicitados por un individuo, han hecho que las empresas se encuentren en una búsqueda continua de métodos más beneficiosos para sus negocios; todo esto aprovechando los recursos y avances tecnológicos que emergen a velocidad superlativa en todos los órdenes, aspectos que generan cambios que al ser asumidos por toda la esfera económica, comercial, financiera redundan en una mayor inserción para ocupar cada vez más espacios como en el sector bancario y afines. Por esta razón y dada la repercusión y el auge de la informática en la sociedad de hoy en día, son cada vez más las empresas que actualizan sus sistemas o aplicaciones actuales para aumentar su desempeño y productividad.

El sector financiero es uno de los pioneros en el uso de internet y las nuevas tecnologías, como recursos en el desarrollo y adecuación de sus procesos; lo cual ha facilitado considerablemente el intercambio físico de bienes y/o propiedades, pues el dinero hoy en día es referenciado como un bien electrónico.

El fideicomiso es un patrimonio relacionado con la transferencia de bienes en un dominio fiduciario, cuyo fin consiste en el establecimiento de un contrato, con el cual se pueden estructurar, eficazmente, financiamientos, proyectos de inversión y mecanismos para el fortalecimiento de garantías y acuerdos, entre otras aplicaciones [1]. El fideicomiso es entonces, un instrumento muy propio de las instituciones financieras.

En esas perspectivas, siguiendo con la frecuencia de avances tecnológicos y de adecuación de nuevos y novedosos sistemas, el Banco Mercantil desarrolló hace casi dos décadas un sistema de Gestión de Fideicomisos, el cual se encuentra operando en *Main Frame*, y fue desarrollado con lenguaje Natural y base de datos ADABAS (BD de listas invertidas). No obstante, a raíz de la constante evolución tecnológica surgió la necesidad de migrar el sistema a una plataforma con una arquitectura cliente-servidor, para mantenerse en vanguardia, sacar provecho de las bondades tecnológicas de hoy en día y disminuir costos de producción y mantenimiento.

Siguiendo los cambios tecnológicos y los retos que impone el mercado, esta institución, a los fines de ser más competitiva, productiva y eficiente dio inicio a la transformación de su estructura operativa de servicio, básicamente para racionalizar los costos, poniendo en funcionamiento y adaptándose a la aplicación FSOSEG (Aplicación Fideicomiso Seguros Mercantil), con lo cual se pretende añadirle valor tecnológico en función de tiempo.

Justificación e importancia

La esencia del proyecto radica en el hecho de que actualmente el costo de alojamiento en la infraestructura de *Main Frame* para el software actual es muy elevado para las operaciones que este despliega. Además que no cuenta con una parametrización para la adición de nuevos módulos y funcionalidades, lo que genera una deficiencia en la escalabilidad del mismo.

“En el mundo empresarial, el efectivo es visto como la parte más importante y delicada en cuanto a la manipulación, uso y preservación del mismo, es por ello que en toda empresa nunca deben faltar buenas directrices que garanticen el buen aprovechamiento de ese activo” [2].

En lo anterior se evidencia la importancia de hacer el uso adecuado del efectivo para las instituciones financieras, es por ello que obtener el saldo real y realizar transacciones entre las cuentas de manera eficaz y efectiva es fundamental para que éstas aprovechen al máximo dicho recurso.

Las empresas financieras, en especial la banca, poseen la necesidad de conocer el saldo real de sus cuentas luego de las transacciones que se realizan día a día entre sus productos, esto le permite a la institución la correcta colocación de los fondos y de esta manera realizar inversiones con el capital que realmente poseen.

Por el motivo expresado en el párrafo anterior, a la Corporación Bimodal Technology (CBT) se le solicitó el desarrollo de una herramienta eficiente que cumpliera con las funcionalidades de la aplicación actual, y que añadiera valor a la empresa permitiendo la creación de nuevos módulos y funcionalidades de forma parametrizable de manera de fortalecer la mantenibilidad y escalabilidad del software.

Esto permitirá a los actores involucrados maximizar su productividad, ya que, mediante el uso de esta tecnología, las labores diarias podrán realizarse de manera más eficiente con un costo de producción menor al actual. Adicionalmente, permite la posibilidad de contar con

reportes e informes que facilitan el entendimiento, estudio y verificación de los datos.

Con el desarrollo de este proyecto, quedará en evidencia que en algunos casos no es necesario incurrir en elevados costos de infraestructura y mantenibilidad para obtener un software de calidad y eficiente, pues mediante el levantamiento de requerimientos se podrá determinar cuál es la tecnología que mejor se adapta al desarrollo e implementarlo a través de un patrón de diseño adecuado permitirá su adaptación a futuros cambios o integraciones. Mientras que la empresa por otro lado, no tendrá que incurrir en altos costos para ejecutar dichas aplicaciones.

Planteamiento del Problema

Actualmente el sistema de Gestión de Fideicomisos de Seguros Mercantil se encuentra operando en *Main Frame*, el cual fue desarrollado con el lenguaje Natural y base de datos ADABAS, lo cual acarrea un elevado costo de mantenimiento en la infraestructura y disminución en la eficacia y eficiencia del software por falta de adición de nuevas tecnologías.

Una consecuencia radica en el hecho de que el costo de este mantenimiento no compensa la gestión de fideicomiso llevada a cabo actualmente, pues el volumen de transacciones que se maneja no requiere el alojamiento en una infraestructura tan costosa generando pérdidas en la inversión.

Sobre esta base se decidió migrar el sistema a una plataforma cliente-servidor, lo que reduciría el costo de alojamiento en infraestructura, proveerá mantenibilidad al sistema, además agilizaría y facilitaría el proceso de adicionar nuevas funcionalidades y parametrizar los procesos actuales.

Objetivo General

Implementar la aplicación y sus funcionalidades para la Gestión de Fideicomisos de Seguros Mercantil que incluyen reportes, obtención del rendimiento por título y distribución del mismo por beneficiario y manejo de transacciones a nivel administrativo y contable.

Objetivos Específicos

- Programar y desarrollar la funcionalidad de ingreso a la aplicación.
- Programar y desarrollar la funcionalidad del manejo de transacciones (administrativo y contable).
- Programar y desarrollar la funcionalidad de distribución del rendimiento por contrato y beneficiario.
- Programar y desarrollar la funcionalidad de reportes.
- Incluir trazas de auditoría.
- Realizar la documentación del sistema.
- Realizar pruebas integrales de la aplicación y pruebas de aceptación.
- Realizar entrega final y pase a producción.

Alcance

El alcance del proyecto fue establecido a través de la metodología de desarrollo ágil SCRUM, en consenso con la gerencia de CBT se acordó que se contará con seis *sprints* (iteraciones): uno de planificación, uno de diseño y arquitectura, tres de desarrollo y una de cierre. En la primera se definirán los requerimientos de los módulos a desarrollar, luego se diseñará la arquitectura de los mismos, tomando en cuenta únicamente los diagramas de dominio y clases, y los modelos de casos de uso y entidad relación.

Posteriormente, se tendrán tres *sprints* (iteraciones) de desarrollo, donde se implementarán todas las funcionalidades asociadas a los módulos: Login, Reportes de Balance de Comprobación, Movimientos Contables por Cuenta y Fideicomiso, Distribución de Rendimiento de las Inversiones, y Manejo de Transacciones de Entrada y Salida de Dinero (a nivel administrativo y contable).

Y finalmente se tendrá un *sprint* (iteración) de cierre, donde el sistema será alojado en un ambiente de pruebas para un cliente real. En la primera de estas, las pruebas serán realizadas por el personal de CBT.

Se entregará un prototipo funcional. Las pruebas que se realizarán serán pruebas internas, unitarias, de integración, desempeño, funcionalidad, regresión y aceptación con la gerencia de CBT. Así mismo, se abordará una primera instancia de la implantación del sistema en ambiente de producción, hasta llegar al punto de la disponibilidad del software para los usuarios finales y la capacitación de los mismos.

De igual forma se hará entrega de los siguientes documentos actualizados o elaborados a completitud de ser el caso:

- Propios de CBT: Requerimientos Técnicos (Apéndice A), Manual del Sistema (B), Manual del Usuario (Apéndice C).
- Tomados de RUP adoptados para este desarrollo de SCRUM: Documento de Arquitectura del Software (Apéndice D), Documento de Especificación de Requerimientos del Sistema (Apéndice E) y Documento de Casos de Pruebas (Apéndice F).

Finalmente, queda fuera del alcance del proyecto y bajo la responsabilidad de la gerencia de CBT la corrección de los errores encontrados luego de que sean efectuadas las pruebas de aceptación y regresión por parte de los usuarios finales.

En el siguiente capítulo se mostrará en detalle la información del marco organizacional, relacionado con la estructura y funciones de la empresa donde se desarrolló el proyecto de pasantía.

CAPÍTULO 1

MARCO ORGANIZACIONAL

En el presente capítulo se describe y reseña la sinopsis de la Corporación Bimodal Technology (CBT), con la finalidad de tener una idea de los aspectos más generales de la empresa donde fue realizado el proyecto de pasantía, además de exponer brevemente la trayectoria de la misma, misión y visión, seguido de la estructura organizacional, y finalmente la ubicación del pasante dentro de la organización.

1.1 Reseña Histórica de la Empresa

CBT fue fundada en Agosto de 2016 por un profesional de Informática proveniente del sector bancario. Desde sus inicios la empresa ha tenido un enfoque de servicios de consultoría y soluciones de forma integral. [3]

El origen de la empresa, está asociado a la necesidad del mercado nacional e internacional de un servicio de consultoría integrador de soluciones, con el objetivo de asesorar a los clientes sobre cómo optimizar sus procesos, permitiendo la reducción de costos sin sacrificar la calidad ni el desempeño eficiente de su Tecnología. [3]

CBT como socio estratégico de sus clientes, participa en el análisis y diagnóstico de sus procesos para identificar áreas de oportunidad y agregar valor a sus estrategias de negocio.

1.2 Ubicación y Logo de la empresa

La empresa se encuentra ubicada en la Avenida Santos Erminy, con calle los Mangos, Edificio Beatriz, piso 3, oficina 33, Sabana Grande, Caracas-Venezuela. Su dirección Web es <http://www.bimodalitil.com.ve>. El logo de la organización se muestra en la imagen de la Figura 1.



Figura 1.- Logo de CBT.

Fuente: Gerencia CBT.

1.3 Misión

“Proporcionar tecnologías innovadoras a la medida de las necesidades de nuestros clientes, con el propósito de incrementar su competitividad y productividad.” [3]

1.4 Visión

“Ser aliados de confianza de nuestros clientes, comprometidos con la solución eficaz y transparente de sus proyectos y requerimientos. Queremos ser la empresa de referencia, que camina con el cambio de la tecnología, dando a conocer las posibilidades de los estándares actuales.” [3]

1.5 Estructura Organizacional de la Empresa

La estructura organizacional de CBT se muestra en el organigrama de la Figura 2.

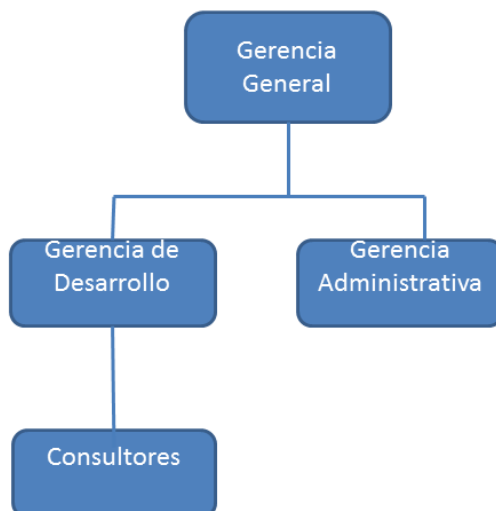


Figura 2.- Estructura organizacional de CBT

Fuente: Gerencia CBT

El proyecto de pasantía fue realizado en el área de Gerencia de Desarrollo bajo la supervisión del Gerente de Proyectos. A continuación, se describe la misión y algunas funciones de dicha sección.

- Contar con la infraestructura apropiada para el adiestramiento que forma parte del crecimiento en las nuevas tecnologías, tanto para sus profesionales como para los profesionales de los socios estratégicos.
- Poseer un modelo de operación en el desarrollo, esto incluye dentro y fuera del sitio para ofrecer servicios especializados que añadan valor a las organizaciones.
- Asesorar acerca del uso de metodologías y tecnologías de información a sus clientes, que les permita satisfacer sus necesidades.
- Realizar consultorías adaptadas a las necesidades del cliente.

El rol del pasante dentro de área de Gerencia de Desarrollo estuvo enfocado en el entendimiento de la lógica del negocio relacionada a la índole del sistema de gestión de fideicomisos, y el desarrollo de la aplicación escenificada en la gestión de transacciones realizadas en la fiduciaria de Seguros Mercantil.

CAPÍTULO 2

MARCO TEÓRICO

En este capítulo se presentan los fundamentos teóricos involucrados durante la elaboración del proyecto de pasantía, con los cuales se buscó una mejor comprensión para el desarrollo y aplicación, no sólo dentro del contexto, sino en la propia lógica de funcionamiento del FSOSEG (Aplicación de Fideicomiso de Seguros Mercantil). Dichos conceptos serán expuestos a continuación divididos en dos categorías. La primera hace referencia a los aspectos propios del área de negocio y la segunda refleja los factores técnicos que implicados en el desarrollo de la aplicación.

2.1 Conceptos Básicos de Fideicomiso

En esta sección se presentan además de las definiciones básicas asociadas a la gestión de fideicomisos, los elementos utilizados para la realización del mismo.

2.1.1 Fideicomiso

“La palabra proviene del latín fideicomisum, que significa comisión o encargo de confianza, palabra derivada de las raíces “fides” (fe, fidelidad y lealtad) y comisión (encargo).” [4]

Representa un convenio que permite el traspaso de bienes a una persona o entidad para que los invierta y/o administre según las instrucciones dadas, y se encargará de distribuir los bienes a favor de quien o quienes le señale como beneficiario.

El fin del fideicomiso es la realización de un negocio u operación. Estos negocios deben ser lícitos, preciso y determinados; posibles y realizables. [5]

2.1.2 Fideicomisos para Personas Jurídicas.

Son fideicomisos diseñados especialmente para empresas, instituciones y/o corporaciones que por su constante crecimiento necesita contar con uno o más tipos de fideicomisos en las áreas de prestaciones sociales, caja o fondos de ahorro, garantía, pensiones, desarrollo inmobiliario con financiamiento extranjero, para garantizar pagos en moneda extranjera, como es el caso de la institución de Seguros Mercantil que ofrece estos tipos de contratos.

2.1.3 Fideicomitente

Persona natural o jurídica (nacional o extranjera), que transfiere la administración de sus bienes a una institución o fiduciaria, quien bajo las restricciones un contrato realiza gerencias a favor de uno o varios beneficiarios.

“Cualquier persona natural o jurídica puede ser fideicomitente siempre y cuando posea la titularidad de un bien o derecho y que tenga la capacidad necesaria para transmitir estos bienes o derechos y constituir un fideicomiso.” [5]

2.1.4 Fiduciario

Se refiere a la institución financiera autorizada para actuar como administrador de los bienes del fideicomitente. “En Venezuela, sólo pueden ser fiduciarios, los bancos, empresas de seguros y las entidades financieras debidamente autorizadas por el Ejecutivo Nacional por resolución del Ministerio de Hacienda y Fomento (actualmente Ministerio del Poder Popular para la Economía y Finanzas), respectivamente.” [5]

2.1.5 Beneficiario

Es quien recibe los beneficios del cumplimiento del fideicomiso. Puede ser el mismo fideicomitente o la (s) persona (s) que éste designe. También pueden ser beneficiarios, las personas jurídicas o naturales, un incapaz mientras dure su incapacidad (liquidez), y pueden nombrarse uno o más beneficiarios. [5]

2.1.6 Rendimiento

“Es el nivel de beneficios producto de una inversión; es decir, la retribución por invertir.” [6]
En este caso, el rendimiento se obtiene por la venta de las inversiones a un precio mayor que su precio inicial de compra, siendo así una ganancia a liquidar mensual, trimestral o anual.

2.2 Conceptos relacionados con el desarrollo de software

En esta sección se presentan las definiciones de los términos y conceptos técnicos utilizados en la realización del proyecto, así como sus ventajas y la justificación de la selección de dichas herramientas.

2.2.1 Arquitectura Tres Capas

Este modelo aporta una flexibilidad adicional en la construcción de aplicaciones cuando éstas aumentan su complejidad. Influye tanto en el modelo de aplicación (lógicas de presentación, del negocio y de datos) como en la distribución de los servicios. El modelo conceptual de una aplicación establece sus definiciones, reglas y relaciones así como su estructura.

Una de las características principales de este modelo reside en la desconexión total entre la lógica de presentación y la lógica de los datos. Las conexiones que se producen, se dan entre las

lógicas de presentación y del negocio, y las lógicas del negocio y la de datos. Particularmente en la capa de presentación se representa la interfaz gráfica del sistema a través de la cual se comunica y se captura información de los usuarios. En la capa de negocio se encuentran los programas que controlan la funcionalidad de la aplicación y los procesos característicos de cada sistema. Y por último, la capa de datos, está formada por uno o más servidores de bases de datos haciendo posible el almacenamiento y el acceso a la información requerida por la capa de negocio.

Este modelo hace que la aplicación final sea completamente independiente del origen de los datos que procesa, tarea que pasa a ser competencia directa del componente especializado. [7]

2.2.2 Arquitectura Cliente/Servidor

El esquema Cliente/Servidor se representa como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma. Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y el proceso cliente sólo se ocupa de la interacción con el usuario [8].

Elementos del esquema Cliente/Servidor:

Clientes: normalmente manejan todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de una red.

Servidores: proporcionan uno o más servicios al cliente y devuelven los resultados. En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, procesar requerimientos de base de datos, procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos, recibir su respuesta y enviarla al cliente. Por las razones anteriores la plataforma computacional asociada con los servidores es más poderosa que la de los clientes.

Uno de los aspectos que más ha promovido el uso de sistemas Cliente/Servidor, es la existencia de plataformas de hardware cada vez más baratas. Esta constituye a su vez una de

las más palpables ventajas de este esquema, la posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes. Además, se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones. [9]

2.2.3 Java

Es un lenguaje de programación orientado a objeto, de una plataforma independiente, fue desarrollado por la compañía Sun Microsystems, con la idea original de usarlo para la creación de páginas Web. [10]

Una de las utilidades de la programación en Java es el desarrollo de aplicaciones, que son programas que se ejecutan en forma independiente, es decir con la programación Java, se pueden realizar aplicaciones como un procesador de palabras, una hoja que sirva para cálculos, una aplicación gráfica, etc, en resumen cualquier tipo de aplicación se puede realizar con ella. Java permite la modularidad por lo que se pueden hacer rutinas individuales que sean usadas por más de una aplicación.

Está preservado por una cuantiosa comunidad de desarrolladores que lo utilizan, es un lenguaje que posee un API (Interfaz de programación de aplicaciones) bastante robusto, además de una documentación detallada y bastante extensa que se actualiza constantemente; por lo que numerosas aplicaciones en el sector financiero están elaboradas a base de este lenguaje. [10]

2.2.4 JasperReport

JasperReport es una biblioteca de creación de informes que tiene la habilidad de entregar contenido enriquecido al monitor, a la impresora o a ficheros PDF, HTML, XLS, CSV y XML.

Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones Web, para generar contenido dinámico. Aunque JasperReport se utiliza principalmente para capacidades de reporte a las aplicaciones basadas en

la Web a través de la API de Servlet, no tiene absolutamente ninguna dependencia de la API Servlet o cualquier otra biblioteca de Java EE. [11]

JasperReport permite separar los datos en secciones. Estas secciones incluyen:

- El título del informe, que aparecerá una vez en la parte superior del informe.
- Un encabezado de página, que aparecerá en la parte superior de cada página.
- Una sección de detalle, que normalmente contiene los datos de los informes primarios.
- A nivel del pie de página, que aparecerá en la parte inferior de cada página.
- Una sección de resumen, que aparecerá al final del informe.

JasperReports se usa comúnmente con iReport, un *front-end* gráfico de código abierto para la edición de informes. [11]

2.2.5 Servidor Apache HTTP

Un servidor Web o servidor HTTP (Protocolo de transferencia de archivos de hipertexto) es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador Web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se usa el protocolo HTTP para estas comunicaciones. [12]

El servidor HTTP Apache es un servidor Web de código abierto, multiplataforma, que implementa el protocolo HTTP y la noción de sitio virtual. El servidor Apache es desarrollado y mantenido por una comunidad de usuarios bajo la supervisión de la *Apache Software Foundation* dentro del proyecto HTTP Server (httpd). Apache es usado principalmente para enviar páginas Web estáticas y dinámicas en la World Wide Web.

Este producto tiene amplia aceptación en la red desde 1996, puesto que, es el servidor HTTP más usado. Así mismos, fungió un papel fundamental en el desarrollo fundamental de la *World Wide Web* y alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios Web en el mundo. En 2009 se convirtió en el primer servidor Web que alojó más de 100 millones de sitios Web [12].

La ventaja del uso de Apache para este proyecto es que proporciona un servidor Web seguro, eficiente y extensible que proporciona servicios de HTTP en sincronización con los estándares HTTP actuales.

2.3 Metodología para el desarrollo de Software

Las metodologías de desarrollo consisten en un conjunto de etapas, procedimientos, técnicas, herramientas y documentación que ayuda a los desarrolladores a crear un nuevo software de manera sistemática. Estas buscan satisfacer tres necesidades de relevancia, las cuales son: un proceso de desarrollo controlado, un proceso normalizado en una organización, es decir, no dependiente del personal, y por último, mejores aplicaciones que conduzcan a una mejor calidad [13].

Para el desarrollo de software involucrado en el proceso de la presente pasantía se ha seleccionado una metodología que abarca muchas de las prácticas de vanguardia en el desarrollo moderno de aplicaciones; dicha metodología es SCRUM, la cual es un producto muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto. SCRUM es una metodología ágil. Una de las ventajas de utilizarla es que es un modo de desarrollo de carácter adaptable más que predictivo, centrándose en las personas más que a los procesos, y emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones.

2.3.1 Estructura de SCRUM

Según [14], en la arquitectura de la metodología ágil SCRUM se comienza con la visión general del producto, especificando y dando detalle a las funcionalidades o partes que tienen mayor prioridad de desarrollo y que pueden llevarse a cabo en un periodo de tiempo breve. Cada uno de estos periodos de desarrollo es un *sprint* (iteración) que finaliza con la producción de un incremento operativo del producto.

En SCRUM, el equipo se focaliza en construir software de calidad. La gestión de un proyecto SCRUM se centra en definir cuáles son las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en vencer cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo.

El equipo SCRUM está formado por los siguientes roles:

- **SCRUM master:** Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el **Product Owner** para maximizar los resultados.
- **Product owner (PO):** Representante de los accionistas y clientes que usan el software. Se focaliza en la parte de negocio y el es responsable de entregar un valor superior al dinero invertido. Traslada la visión del proyecto al equipo, formaliza las prestaciones en funcionalidades a incorporar en el *backlog* y las reprioriza de forma regular.
- **Team:** Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a cabo las funcionalidades a las que se comprometen al inicio de cada sprint (iteración).

Estos *sprint* (iteraciones) son la base del desarrollo ágil, y SCRUM gestiona su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado el día anterior y el previsto para el día actual. A continuación se muestra en la figura 3, el diagrama del ciclo de vida:

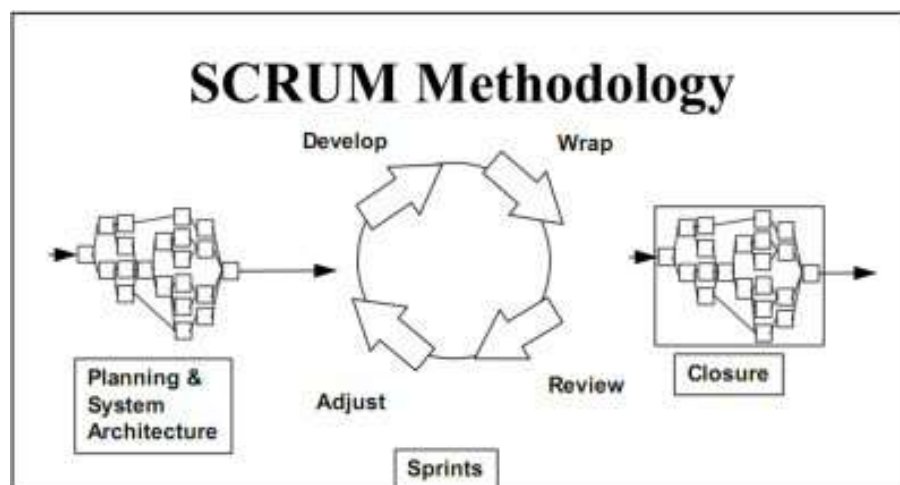


Figura 3.- Diagrama del ciclo de vida de SCRUM [15]

Etapas de planificación

- Desarrollo de un *backlog* completo.
- Determinación de la fecha de entrega y la funcionalidad de una o más versiones.
- Selección de la versión más adecuada para desarrollo inmediato.
- Trazado de los “paquetes del producto” (objetos) sobre los elementos del *backlog* de la versión elegida.
- Selección del equipo o equipos para desarrollar la nueva versión.
- Evaluación y control adecuado de los riesgos.
- Estimación del costo de la versión, incluyendo desarrollo, material, mercadeo, formación y despliegue.
- Conformidad de la dirección y financiación del proyecto.

Etapas de diseño y arquitectura

- Revisión de los elementos del *backlog* incluidos en la versión.
- Identificación de los cambios necesarios para implementar el *backlog*.
- Análisis del dominio para incluir los requisitos que incluye el desarrollo mejora o actualización.
- Delimitación de la arquitectura de sistema para apoyar el nuevo contexto y necesidades.
- Identificación de problemas de desarrollo o modificaciones.
- Reunión de revisión de diseño. Cada equipo desarrollador presenta los cambios para implementar los elementos del *backlog*, e identificar posibles reasignaciones.

Etapas de Desarrollo del *Sprint* (Iteración)

La fase de desarrollo implica un ciclo de trabajo repetitivo. La gestión determina el cumplimiento de los tiempos, funcionalidad y calidad, las características de esta etapa es conocido como ingeniería concurrente.

La lógica de lo antes planteado, apunta hacia aplicación de cambios, necesarios para la implementación de los requisitos del *backlog* en módulos, la apertura de los módulos, análisis del dominio, diseño, desarrollo, implementación, pruebas y documentación de los mismos. En tal

sentido, la etapa de desarrollo consiste en el micro proceso de descubrimiento, invención e implementación.

Cada *sprint* (iteración) es seguido de una revisión cuyas características son:

- Presencia y participación del equipo desarrollador.
- La revisión puede incluir a clientes, personal de ventas y otros.
- La revisión cubre los sistemas funcionales y ejecutables abarcados por el equipo e incluye los cambios que se han realizado para implementar los elementos del *backlog*.
- En la revisión se pueden evidenciar cambios en la forma en la que se han implementado los elementos del *backlog*.
- La revisión también puede introducir elementos nuevos en el *backlog*, cambiando de esta forma los contenidos y dirección de las versiones previstas.
- Se determina la fecha de la siguiente revisión en base al progreso y complejidad. La duración normal de los *sprints* (iteraciones) es de 1 a 4 semanas.

Cierre

Proceso según el cual, el equipo de gestión siente que las variables de tiempo, parte completada, requisitos, costo y calidad están alineadas para producir una nueva versión, declara cerrada la versión, dando, para dar paso a la siguiente fase. En la siguiente, se prepara el producto generado para producir una nueva versión. Entre las tareas de cierre se encuentran: integración, pruebas de sistema, documentación de usuario, preparación del material de formación y mercadeo.

Para efectos del proyecto de pasantía, el SCRUM Master estuvo representado por el Tutor Industrial, quien a su vez actuó como Gerente de Desarrollo del Proyecto. Mientras que el Product Owner estuvo representado por la Gerencia de Fideicomisos de Banco Mercantil, quien hará entrega a los usuarios de Seguros Mercantil el producto final FSOSEG. Finalmente el Team o equipo desarrollador fue representado por el pasante, quien realizó el desarrollo de los *sprints* (iteraciones) según las reglas de la metodología, y donde el *backlog* estuvo definido de acuerdo a las funcionalidades implementadas en cada caso.

En el siguiente capítulo se podrá visualizar en detalle la adaptación de la metodología SCRUM al desarrollo de la aplicación, la cual fue estructurada en seis fases que contemplan diversos *sprint* (iteraciones) para la consecución de los objetivos planteados.

CAPÍTULO 3

MARCO METODOLÓGICO

Con el fin de enmarcar un desarrollo orientado a mejorar la productividad y calidad del software, que involucre además una reducción de riesgos y se adapte a las necesidades del cliente, se ha seleccionado una metodología ágil que moldea la construcción de mejores características y funcionalidades a ofrecer por parte del software a través de las mejores prácticas de desarrollo. Dicha metodología es SCRUM la cual fue adaptada a las necesidades y exigencias tanto de la empresa como del proyecto.

3.1 Naturaleza del Proyecto

El trabajo de pasantía se encuentra enmarcado bajo la clasificación de proyecto especial, pues según [16], entran en esta modalidad todos aquellos trabajos que lleven creaciones tangibles, susceptibles de ser utilizadas como soluciones a problemas demostrados. Se incluyen en esta categoría los trabajos de elaboración de libros de texto, y de materiales de apoyo educativo, el desarrollo de software, prototipos y productos tecnológicos en general, así como también los de creación literaria y artística.

De acuerdo con lo expresado anteriormente, ya que, la actividad fundamental del proyecto de pasantía consistió en el desarrollo de un software para automatizar el proceso de conciliación bancaria, se evidencia el tipo de trabajo.

3.2 Fases de la metodología

Según [16], en el caso de los proyectos vinculados a actividades de pasantía profesional, la sección de metodología debe ser sustituida por una descripción de los roles y las funciones objetos de dicha experiencia, su sustentación en los objetivos, el perfil de las competencias y otros aspectos relevantes del diseño curricular del subprograma respectivo, y su justificación considerando la trayectoria y planes del desarrollo profesional del estudiante u otros argumentos relevantes.

En este sentido, se realizó la división de las experiencias del proyecto en seis fases, con la relevancia de que dentro de las antes mencionadas, a raíz de la naturaleza del mismo, fueron incluidas las etapas de la metodología SCRUM. Estas seis fases son:

1. Adaptación.
2. Levantamiento de requerimientos (*Etapas de Planificación de SCRUM – 1 Iteración*)
3. Diseño (*Etapas de Diseño y Arquitectura de SCRUM – 1 Iteración*)
4. Implementación (*Etapas de Desarrollo de SCRUM – 3 Iteraciones*)
5. Pase a producción (*Etapas de Cierre de SCRUM – 1 Iteración*)
6. Elaboración de entregables.

3.2.1 Fase 1: Adaptación.

Por ser la primera fase del proyecto, ésta estuvo enfocada en la familiarización con el ambiente laboral en la empresa, con lo que respecta a las normas, códigos de vestimenta, cumplimiento de horario, convivencia con los compañeros de trabajo y la comprensión del funcionamiento y estructura de la empresa, a través del análisis de la visión, misión y valores, así como el estudio de las funciones de las áreas que conforman la institución y su interacción.

Adicionalmente se asistió a una serie de presentaciones relacionadas a la lógica del negocio (Transacciones a través del sistema actual BFE – Proceso Contable de Fideicomiso), lo cual permitió expandir el panorama y facilitar la comprensión del proyecto.

Es importante resaltar que en las etapas posteriores (referentes al proceso de desarrollo del software) se utilizó, por las razones expresadas en la sección anterior, la metodología SCRUM

como guía.

3.2.2 Fase 2: Levantamiento de Requerimientos.

Ya que, a partir de esta etapa las actividades de la pasantía comprendieron un proceso de desarrollo de software, se decidió junto con la gerencia de la Corporación Bimodal Technology (CBT) emplear como marco referencial la metodología SCRUM, aprovechando que la misma permite ser flexible en cuanto a la cantidad de tareas, procedimientos y entregables. Así, las siguientes etapas estuvieron orientadas en función de una planificación sencilla cada uno de los *sprints* (iteraciones) correspondientes. De igual forma cabe destacar que ésta es la metodología usada en la empresa para todos sus proyectos que involucran el desarrollo de software.

Así pues, esta fase inició con el entendimiento de las labores relacionadas al proceso de fideicomiso más a fondo, y las actividades correspondientes que de ahí se derivan. Así mismo se procedió con la planificación de reuniones en Banco Mercantil con el asesoramiento del tutor industrial, para ejecutar la versión anterior de Fideicomiso y la instalación posterior de las herramientas para continuar con las labores de desarrollo de la versión cliente servidor.

En esta sección fue actualizado el artefacto: Documento de Especificación de Requerimientos del Software (Apéndice E).

Así mismo, durante esta etapa se comenzó con la primera iteración estipulada en SCRUM a través de su etapa de Planificación.

3.2.3 Fase 3: Diseño.

En esta fase se realizó la primera revisión y actualización de los diagramas del sistema, como por ejemplo:

- Modelo de Casos de Uso (Ver “Documento de Arquitectura de Software”, Apéndice D)
- Diagrama Entidad-Relación (Ver “Documento de Arquitectura de Software”, Apéndice D)
- Diagrama de Clases (Ver “Documento de Arquitectura de Software”, Apéndice D)

Finalmente, la idea fundamental de esta etapa es definir una arquitectura base sobre la cual

implementar los módulos del sistema; con la presente culminó la iteración correspondiente a la Etapa de Diseño y Arquitectura de SCRUM.

3.2.4 Fase 4: Implementación.

Esta etapa estuvo dividida en tres *sprints* (iteraciones) en función de la metodología SCRUM, a lo largo de las cuales se llevaron a cabo la construcción y mejora de los siguientes módulos:

- Login.
- Manejo de Transacciones (Nivel Administrativo y Contable).
- Distribución de Rendimiento.
- Reportes: Balance de Comprobación, Movimientos Contables por Cuenta y Fideicomiso.

Todas las anteriormente mencionadas iteraciones corresponden a la etapa de Desarrollo de SCRUM.

3.2.5 Fase 5: Pase a Producción.

La primera actividad de esta fase consistió en una investigación para la selección de la tecnología necesaria para la implantación del sistema, y poder habilitar el acceso a los mismos para los usuarios finales.

Una vez seleccionada la herramienta se estableció en conjunto con la Gerencia de CBT un cronograma para el pase a producción del sistema. De forma paralela se ejecutaron pruebas unitarias, de desempeño, funcionalidad y regresión del nuevo sistema, junto con pruebas en paralelo versus el sistema anterior. Fueron ajustados los manuales y documentación necesaria:

- Documento de Casos de Prueba (Apéndice F).
- Manual de Usuario (Apéndice C).

Adicionalmente, se preparó el ambiente para el pase a producción en los equipos de Banco Mercantil y se culminó una serie de presentaciones y cursos para la capacitación de los usuarios finales.

Todo lo anterior transcurrió a lo largo de dos *sprints* (iteraciones) contempladas en la Etapa de

Cierre de SCRUM.

3.2.6 Fase 6: Elaboración de Entregables.

En lo correspondiente a la etapa final del proyecto de pasantía, la misma estuvo enfocada hacia la revisión y actualización de todos los documentos relacionados con el sistema, tanto los artefactos solicitados por la gerencia, como los propios de la empresa, adicionalmente en esta etapa se redactó el informe final para entregar a la universidad.

En el siguiente capítulo se podrá detallar cómo fue la implementación de cada módulo de la aplicación, a través del seguimiento de la realización de las actividades.

CAPÍTULO 4

RESULTADOS

El presente capítulo tiene como finalidad hacer un recorrido, ahondar en los detalles y ordenar todo el proceso de desarrollo del proyecto de pasantía. Para ello se describen las actividades realizadas y los resultados obtenidos en las fases correspondientes al antes mencionado.

4.1 Fase de Adaptación.

La fase de adaptación tiene como finalidad la inclusión del pasante en el ambiente laboral de la empresa, el conocimiento de las normas, integración al equipo de trabajo y conocimiento de la lógica del negocio. Por lo que, en principio se realizó el recorrido de las instalaciones de la compañía y la introducción al personal laboral, se le fueron expuestas las tareas de las áreas dentro de la organización. Luego el Tutor Industrial realizó una serie de presentaciones donde se introdujeron los conceptos y características primordiales en lo que envuelve el sistema actual de Fideicomiso, y explicando a detalle el proceso de transacciones; además éstas demostraciones emplearon el uso de datos reales, lo que representó un agregado de importancia.

Al finalizar la etapa se logró colocación adecuada del pasante dentro del ambiente de la organización, además se contaba con las instrucciones necesarias para la inclusión en la lógica de negocio y los conocimientos básicos necesarios para comprender las transacciones y cómo se lleva a cabo el proceso de fideicomiso.

4.2 Fase de Levantamiento de Requerimientos.

Dentro de esta fase se realizó un estudio un poco más detallado de cada una de las funcionalidades comprendidas en el sistema de Fideicomiso, teniendo la primera interacción con el mismo, a través de la navegación en la versión actual ejecutada en los equipos de Banco Mercantil. El pasante pudo estudiar más a fondo el comportamiento de la plataforma, las posibles acciones ejecutadas por un usuario y las respuestas de la aplicación.

Por otra parte, para esta etapa del proyecto se hizo uso de las directrices establecidas en la etapa de Planificación de la metodología SCRUM, la cual comprende el análisis y diseño de los requerimientos funcionales y técnicos más significativos de los módulos a incluir y funciones a revisar y mejorar en la aplicación, con la finalidad de definir una arquitectura adecuada para los mismos que permita de manera sencilla su integración. Es así como mediante una primera iteración se realizaron un conjunto de reuniones y sesiones de orientación con el personal de la empresa involucrado en el área, en estas se discutió la metodología de trabajo, los estándares de desarrollo y las herramientas utilizadas para las versiones anteriores de la aplicación. En consecuencia se aclararon dudas y se establecieron los requerimientos de los nuevos módulos, destacándose los siguientes:

- Mantener al día el proceso de transacciones (a nivel administrativo y contable).
- Mantener al día los asientos contables y distribución de rendimiento.
- Generar los reportes para conocer los movimientos en los contratos.
- Crear un log de auditoría que registre los eventos que sean prioritarios en el sistema.

Se realizó una investigación acerca de las herramientas tecnológicas que mejor se adaptaba a los requerimientos funcionales y técnicos expuestos, donde se evidenció que el uso de Java posee la característica de incrementar el rendimiento gracias a su concepción de orientación a objeto lo que facilita el desarrollo modular, y particularmente en este caso un modelo cliente-servidor que cuenta con las ventajas descritas en los capítulos anteriores; entre ellas una capa de abstracción a la base de datos que permite ejecutar cualquier tipo de consultas al servidor de base de datos que se utilice sin necesidad de implementar código extra e independientemente del manejador que se emplee.

Para la implementación de las interfaces gráficas de usuario se utilizó el API (Interfaz de programación de aplicaciones) Swing de Java adaptado al IDE (Entorno de desarrollo integrado)

de Netbeans, para interactuar con el usuario a través de su modelo de eventos. JasperReports para el diseño y desarrollo de los reportes y MySQL como manejador de BD.

Seguidamente se ejecutó la instalación de todas las herramientas necesarias para la puesta en marcha del desarrollo de software, para ello la empresa CBT facilitó una computadora con los requerimientos de hardware para tal labor.

Siguiendo los estándares de los artefactos de la metodología RUP, se realizó parte de la documentación correspondiente, con el fin de contar con un claro panorama en relación a las necesidades y características de la empresa y de los módulos y funcionalidades adicionales que son requeridas por la organización; este documento es:

- Especificación de Requerimientos de Software (Este documento detalla los requerimientos de software para el Sistema FSOSEG (Aplicación de Fideicomiso de Seguros Mercantil), describiendo todo lo referente a las decisiones arquitectónicas del sistema. Para más detalle ver Apéndice E).

4.3 Fase de Diseño.

Se diseñaron los nuevos módulos y se realizó el Modelo de Casos de Uso (MCU), según el lenguaje UML [17] y [18] haciendo uso de la herramienta StarUML. El MCU toma vital relevancia, ya que, cada caso de uso representa una función específica dentro de la aplicación; este artefacto funge como puente de enlace y herramienta que facilita el entendimiento entre el usuario, quien mediante él visualiza un objetivo determinado dentro del sistema, y el desarrollador. Además, cada caso de uso puede representar una funcionalidad muy simple o compleja. A raíz de la reestructuración del MCU, se definieron un total de sesenta y seis Casos de Uso, de los cuales se implementaron sesenta y cinco a completitud, de los cuales dos de ellos no se encontraban en el alcance del plan de pasantía.

En reuniones previas al desarrollo, se había determinado que el ingreso y salida de la aplicación estaría conectado al directorio activo de Seguros Mercantil. Sin embargo por causas internas no expuestas en la Gerencia de Banco Mercantil, se solicitó incluir el desarrollo a completitud del ingreso y salida a FSOSEG de forma independiente; en base a esto la funcionalidad de modificar contraseña, no fue contemplada como una prioridad en este desarrollo, pues la información de los usuarios sería migrada y estaría alojada en la nueva BD. Por lo que en caso de ser necesaria una modificación en la contraseña, se solicitaría al actor

Superusuario la gestión de dicha actividad a través de la BD directamente.

En las siguientes figuras 4 y 5 se puede observar la nueva estructura del MCU general con todas las funcionalidades de la aplicación:

Leyenda:



Casos de Uso implementados a completitud



Casos de Uso implementados a completitud, no contemplados en el proyecto de pasantía.

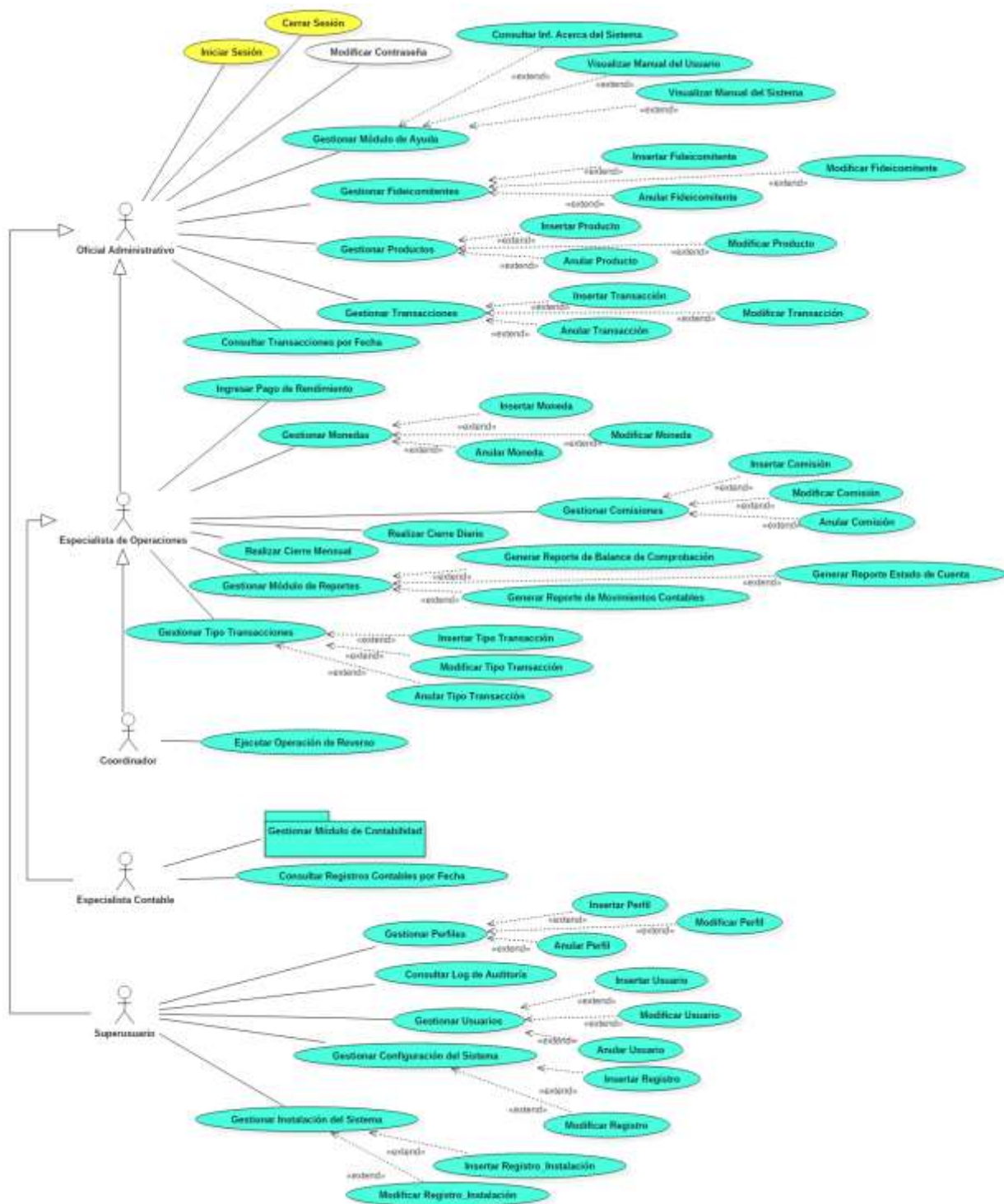


Figura 4.- Modelo de Casos de Uso General.

Fuente: Elaboración Propia.

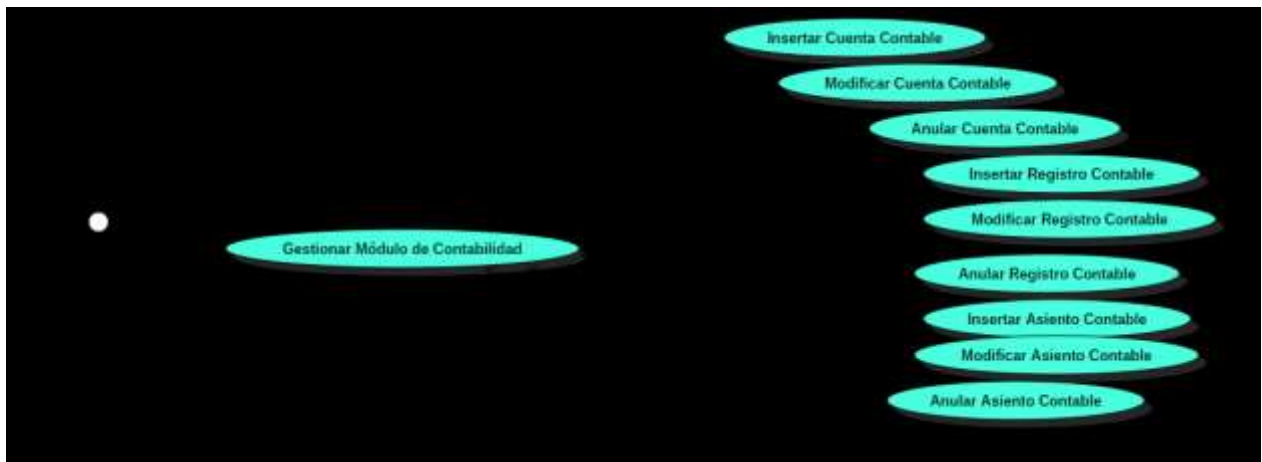


Figura 5.- Modelo de Casos de Uso. Paquete de Gestionar Módulo de Contabilidad.

Fuente: Elaboración Propia

Posteriormente se realizó el Documento de Arquitectura del Software (Apéndice D), donde se mantuvo el diagrama de casos de uso del sistema, el diagrama de Clases y Entidad-Relación. De igual manera, es importante señalar que todos los diagramas y modelos del sistema fueron actualizados, refinados y corregidos constantemente, para así poder garantizar que los mismos se ajustasen correctamente a los requerimientos del sistema.

Por último, para explicar y describir el contexto del problema se realizó un diagrama conceptual del mundo real y no de los componentes del sistema. Este modelo presenta los diversos conceptos que intervienen en el entorno del problema y plantea la relación e interacción que existe entre estos. Además se incluye el diagrama de clases del sistema, el diagrama de implantación que describe la topología del sistema, la estructura de los elementos de software y hardware empleados, y el diagrama de implementación donde se resume la interacción de los componentes del sistema (Figuras 6,7, 8 y 9).

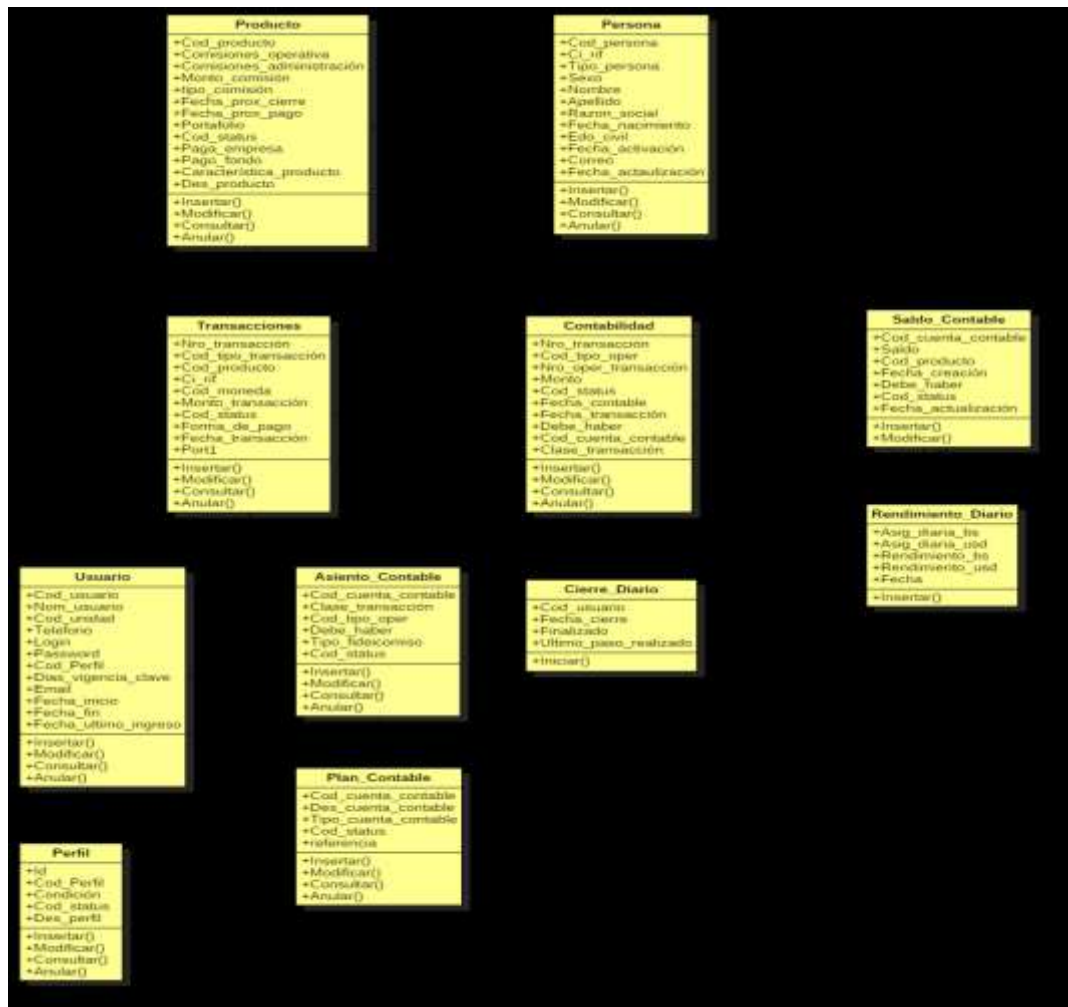


Figura 6.- Diagrama de Clases.

Fuente: Elaboración Propia.

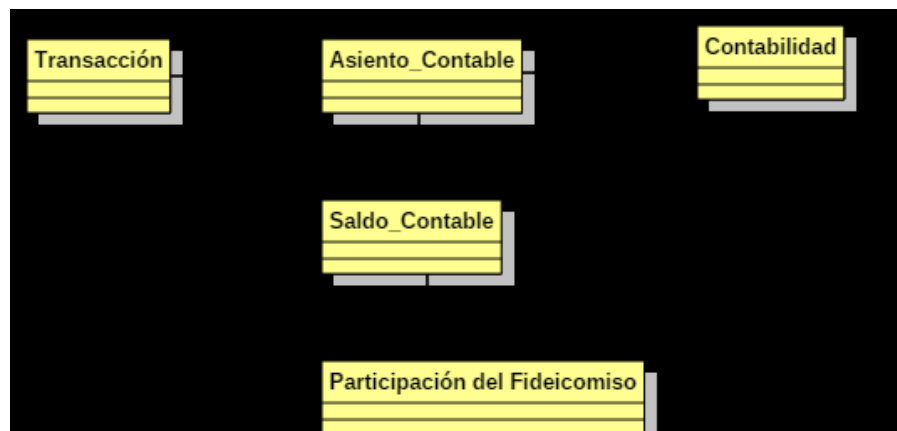


Figura 7.- Modelo Conceptual.

Fuente: Gerencia CBT.

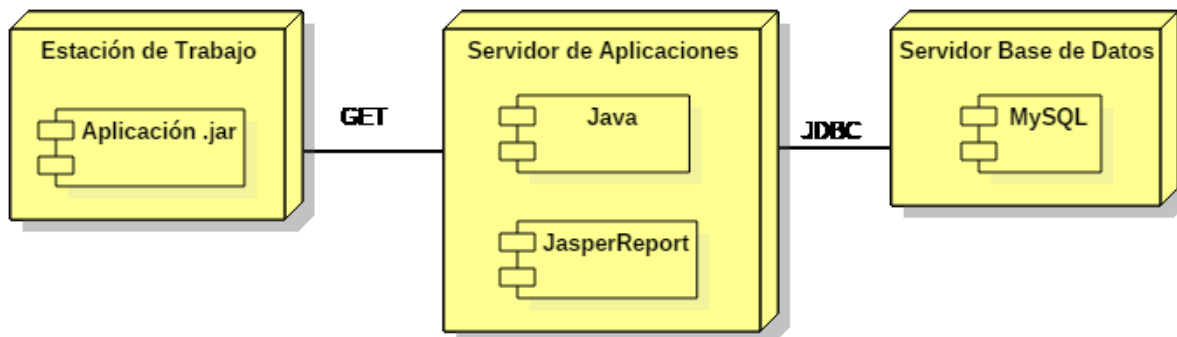


Figura 8.- Diagrama de Implantación.

Fuente: Elaboración Propia.

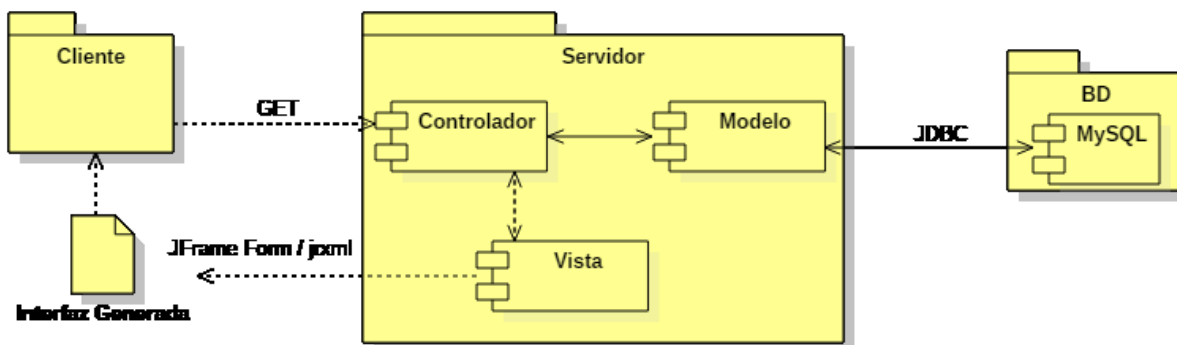


Figura 9.- Diagrama de Implementación.

Fuente: Elaboración Propia.

4.4 Fase de Implementación.

La fase de implementación estuvo enfocada en la construcción de los nuevos módulos que serían integrados al sistema junto con las nuevas funcionalidades. En esta etapa también se llevó a cabo el proceso de revisión de los módulos y funcionalidades de la versión previa de FSOSEG para su evaluación, mejoras y ajustes de las mismas. Además, a lo largo de esta fase fueron ejecutados todas las pruebas y ajustes pertinentes para garantizar un producto estable y adecuado a lo esperado por el cliente.

Es importante resaltar que a lo largo del proceso de implementación, mejora e integración de los módulos y funcionalidades que se contemplaron para la fase se mantuvo la consistencia en

todos los componentes de interfaz. Esto a través del empleo de técnicas consistentes, como el mismo tamaño y estilo de fuente, tipos, ubicación, color y consistencia de botones, íconos y formularios, garantizando así el cumplimiento de las mejores prácticas para el diseño de interfaces gráficas de usuario.

Haciendo uso de SCRUM, esta etapa fue dividida en tres iteraciones correspondientes la fase de desarrollo de dicha metodología. A continuación se describen detalladamente las actividades realizadas durante cada una de ellas.

4.4.1 Primera Iteración: Login y Módulo de Transacciones.

Esta iteración estuvo dedicada en su totalidad al desarrollo de la funcionalidad de Login y del módulo manejo de transacciones de la aplicación para cumplir con los objetivos de “Programar y desarrollar la funcionalidad de ingreso a la aplicación” y “Programar y Desarrollar el manejo de transacciones (a nivel administrativo y contable)”.

Inicialmente se realizó un proceso de investigación sobre las herramientas de codificación, se consultaron tutoriales y documentación de sistemas realizados con la misma tecnología. Así mismo, se realizaron sesiones en Banco Mercantil para entender la ejecución de los diversos tipos de transacciones en el sistema actual y cómo se generan los asientos contables una vez generado un cierre diario o mensual. Posterior a ese proceso se decidió comenzar el desarrollo programando las transacciones administrativas donde destacan aportes, retiro, pagos de impuestos y sus correspondientes reversos.

Las transacciones representan los tipos de movimientos que se pueden hacer sobre un fideicomiso. Una vez se ejecute un cierre diario o mensual, se deben generar los asientos contables correspondientes a cada tipo de transacción, de esta manera se realiza de forma automática la contabilidad en la aplicación la cual es verificada por los especialistas contables.

El proceso de elaboración de las funcionalidades relacionadas a las transacciones comenzó con el diseño e implementación de la interfaz, aquí se emplearon botones para hacer la división de las opciones de búsqueda, inserción, modificación y anulación de los registros. Inicialmente se desarrolló el proceso de inserción, que consiste en el ingreso de datos en un formulario por parte del usuario, que serán usados para crear un registro en el sistema. Allí se debe ingresar el documento que identifica a la empresa previamente registrada, escoger el contrato, la moneda, el tipo de transacción y las fechas de acuerdo al registro y la transacción. Luego se tiene otra

sección de datos contables la cual es opcional, y se tiene por si se quiere agregar información adicional a la transacción. Aquí se hizo uso de formularios con listas de opciones desplegables para la introducción de los datos por parte del usuario, calendarios dinámicos para el ingreso de fechas, y botones para confirmar la ejecución de las acciones.

Seguidamente se procedió con el proceso de validaciones de los campos, haciendo énfasis en el formato de aquellos campos que usan solo texto o solo números. Así mismo se construyeron mensajes de errores para cada tipo de acción indebida o datos mal suministrados, donde se le indica al usuario el motivo del error.

Posteriormente se implementó el *back-end* de la funcionalidad, dónde a través del desarrollo de los procedimientos en el controlador se realizan invocaciones en el modelo de *store procedure* al servidor para ingresar el registro en la base de datos y se envía una respuesta de confirmación al cliente.

Acto seguido se prosiguió con el desarrollo de la funcionalidad de modificación de una transacción, la cual consiste en el ingreso del número de transacción por parte del usuario, que será usado para extraer del sistema la transacción correspondiente que coincida con ese valor y pueda ser editada. Para esto solo fue necesario el uso de un campo de texto para obtener el valor. De igual manera, se diseñó e implementó la interfaz siguiendo el estándar de la aplicación, utilizando formularios con el mismo estilo que los que ya se tienen en el sistema. Además se construyeron mensajes de errores para cada tipo de acción indebida o datos mal suministrados, donde se le indica al usuario el motivo del error.

La siguiente funcionalidad desarrollada fue la anulación de una transacción. Esto se trata de cambiar el estatus de la transacción a nivel de base de datos, pero que el registro no quede eliminado por completo. En este caso, se utilizó la misma interfaz donde el usuario debe ingresar un número de transacción existente.

De igual forma, en la capa del controlador se implementaron los procedimientos que realizan invocaciones en el modelo para hacer las modificaciones en la base de datos.

Cabe destacar que para efectos de anular algún registro en este módulo o en cualquier otro el concepto es el mismo, ya que por requerimientos de la empresa no se deben eliminar registros desde la aplicación sino anulados.

La última de las funcionalidades relacionadas con el manejo de transacciones es la de búsqueda, donde el usuario debe ingresar un rango de fechas para que el sistema pueda extraer los registros que coincidan con esos valores. Para ello se diseñó e implementó la interfaz que

posee calendarios dinámicos para el manejo de fechas, y un botón para confirmar la ejecución de la acción seguido de una tabla donde se muestra la información resultante de manera adecuada. Así mismo se construyeron mensajes de errores para cada tipo de acción indebida o datos mal suministrados, donde se le indica al usuario el motivo del error.

De igual forma se construyeron en el controlador los procedimientos necesarios que hacen las llamadas correspondientes al modelo para poder acceder a la base de datos y consultar la información requerida.

En las figuras 10 y 11 se pueden apreciar la interfaz de la creación, búsqueda, modificación y anulación de transacciones.

Figura 10 .- Insertar, Modificar, Anular Transacción.

Fuente: Elaboración Propia.



Figura 11.- Búsqueda de Transacciones.

Fuente: Elaboración Propia.

La codificación relacionada a esta funcionalidad implicó el desarrollo de otras áreas relacionadas con el manejo de transacciones, puesto que a nivel de *front-end* sólo se añadieron mensajes estáticos y emergentes en las interfaces ya presentadas. Entre las implementaciones más importantes se tienen:

- **Gestión de Tipo de Transacción:** Se implementó una interfaz para la gestión del tipo de transacciones a utilizar en la aplicación con las mismas opciones que en la interfaz de transacciones (insertar, modificar y anular), las cuales permiten generar una transacción correspondiente. Su acceso queda restringido para aquellos usuarios sin permisología.
- **Gestión de Asiento Contable:** Esta interfaz permite gestionar los asientos contables que se pueden generar una vez ejecutado el cierre diario o mensual. Cuenta con las mismas opciones de insertar, modificar y anular. Su acceso queda restringido para aquellos usuarios sin permisología.
- **Gestión de Plan de Cuentas:** Permite gestionar el plan de cuentas contables requerido para realizar la contabilidad en el sistema una vez generado el cierre. Su interfaz provee las opciones de insertar, modificar y anular. Mientras que su acceso queda restringido para aquellos usuarios sin permisología.

- **Gestión de Contabilidad:** Permite gestionar los asientos contables generados al ejecutar el cierre diario o mensual en el sistema. Por lo general el llenado de esta tabla es automático, sin embargo se implementó la interfaz de esta funcionalidad en el caso de que sea necesario alguna inserción manual, modificación o anulación de algún registro. Su acceso queda restringido para aquellos usuarios sin permisología.
- **Ejecución del proceso de Reverso:** Esta funcionalidad se realiza cuando ha ocurrido algún error en el proceso de cierre diario o mensual, como por ejemplo fallas de luz que impida la consecución del cierre, errores en el cálculo de los saldos, entre otros. Al ejecutarla se eliminan los asientos contables que hayan sido generados en la contabilidad, y permite consultar si hubo un error en el manejo de transacciones que deba ser corregido antes de generar nuevamente el cierre. Su acceso queda restringido para aquellos usuarios sin permisología.
- **Gestión de Beneficiarios:** Permite gestionar el registro e información de las personas que forman parte de un fideicomiso. Tiene las opciones de insertar, modificar y anular.
- **Gestión de Fideicomisos:** Permite gestionar los productos o fideicomisos dentro de la aplicación, a los cuales se les pueden atribuir diversos tipos de transacciones. Su interfaz provee las opciones de insertar, modificar y anular.

En las figuras 12, 13, 14, 15, 16 y 17 se pueden visualizar la interfaz correspondiente a cada funcionalidad.

Tipo de Transacciones

21/03/2017 18:57

Datos Administrativos

Código Único

Cuenta

Descripción Tipo de Transacción

Referencia

Debe Haber

Clase de Transacción

Refrescar

| Código Único | Cuenta | Descripción Tipo de Transacción | Referencia | Debe Haber | Clase de Transacción |
|--------------|--------|---------------------------------|------------|------------|----------------------|
| | | | | | |

Insertar

Guardar

Anular

Modificar

Actualizar

Consultar

Salir

Figura 12- Gestor de Tipo de Transacciones.
Fuente: Elaboración Propia.

Plan Contable

27/06/2017

28

Datos Contables

Código de la Cuenta Contable

Descripción de la Cuenta Contable

Tipo de Cuenta Contable

Referencia

Actualizar

| Código de Cuenta Contable | Descripción Cuenta Contable | Tipo de Cuenta Contable | Referencia |
|---------------------------|---|-------------------------|------------|
| 411 | Disponibilidades | Efectivo | N/A |
| 41102 | Depositos en la Institución | Efectivo | N/A |
| 411021 | Depositos en la Institución-Efectivo | Efectivo | N/A |
| 41102106 | Depositos en la Institución-Efectivo | Efectivo | N/A |
| 4110210601 | Cuenta Operativa Fideicomiso | Catálogo | N/A |
| 41102106010 | Cuenta Operativa Fideicomiso 1077-677 | Efectivo | N/A |
| 41102106010 | Cuenta Invs. Fideicomiso 1079-67614-6 | Efectivo | N/A |
| 41102106010 | Cta Invs. Pol Especializada 1079-6514-6 | Efectivo | N/A |
| 41102106010 | Cta Invs. Pol FAPSH 1699-67620-6 | Efectivo | N/A |
| 41102106010 | Cta Invs. Pol FAPSH 1699-67621-6 | Efectivo | N/A |
| 41102106010 | Cta Invs. Pol BCO 1009-60418-2 | Efectivo | N/A |
| 4110220001 | Cuenta Inversión Fideicomiso | Catálogo | N/A |
| 41103 | Otras Disponibilidades | Efectivo | N/A |
| 411031 | Otras Disponibilidades | Efectivo | N/A |
| 41103103 | Otras Disponibilidades | Efectivo | N/A |
| 41103103010 | Otras Disponibilidades | Efectivo | N/A |
| 41103103010 | Provincial 0108-0182-18-0103179-18 | Efectivo | N/A |
| 41103103010 | Exterior 0115-0010-24-00001695-18 | Efectivo | N/A |
| 41103103010 | Banco 0124-0850-51-5803004909 | Efectivo | N/A |
| 411032 | Otras Disponibilidades | Efectivo | N/A |
| 41103206 | Otras Disponibilidades | Efectivo | N/A |
| 41103206010 | Cuenta Operativa Fideicomiso ME 83000 | Efectivo | N/A |
| 41103206010 | Cta Invs. Pol ME HERLI 116431-621-67246 | Efectivo | N/A |
| 41103206010 | Cta Invs. COMMERCEBANK 630-151212 FEM | Efectivo | N/A |
| 41103206010 | Cta Invs. COMMERCEBANK 630-151212 A. | Efectivo | N/A |
| 41103206010 | Cta Invs. UBS 110501076101400 | Efectivo | N/A |
| 41103206010 | Cta Invs. Standar Charter 1105040801 | Efectivo | N/A |

Actualizar

Insertar

Quitar

Anular

Modificar

Actualizar

Consultar

Salir

Figura 13- Gestor de Plan Contable.
Fuente: Elaboración Propia.

Asiento Contable 27/05/2017 3

Tipo de Asiento:

Código Tipo Cuenta de Trans. Clase de la Transacción Código de Cuenta Contable

Debe Haber Debe

Campo de la Transacción Cálculo

Referencia de la Transacción

| Tipo Oper... | Clase Trans... | Cuenta Con... | Debe/Haber | Campo Trans. | Cálculo | Estatus |
|--------------|-----------------|---------------|------------|----------------|---------|---------|
| AOA | Aporte | 41102100001 | D | administracion | | A |
| AOA | Aporte | 41102199001 | H | administracion | | A |
| ROA | Reverso Apto... | 41102199001 | D | administracion | | A |
| ROA | Reverso Apto... | 41102100001 | H | administracion | | A |
| ROR | Reverso Retiro | 41102100001 | D | administracion | | A |
| ROR | Reverso Retiro | 41102199001 | H | administracion | | A |
| RUD | Reverso Retiro | 41102100001 | D | administracion | | A |
| RUD | Reverso Retiro | 41102199001 | H | administracion | | A |
| OA | Aporte | 41700100001 | D | administracion | | A |
| OA | Aporte | 41102199001 | H | administracion | | A |
| RUZ | Reverso Apto... | 41102199001 | D | administracion | | A |
| RUZ | Reverso Apto... | 41700100001 | H | administracion | | A |
| OA | Retiro | 41102199001 | D | administracion | | A |
| OA | Retiro | 41700100001 | H | administracion | | A |
| ROR | Reverso Retiro | 41700100001 | D | administracion | | A |
| ROR | Reverso Retiro | 41102199001 | H | administracion | | A |
| OAC | Aporte | 41200100006 | D | administracion | | A |
| OAC | Aporte | 41102199001 | H | administracion | | A |
| OAC | Reverso Apto... | 41102199001 | D | administracion | | A |
| OAC | Reverso Apto... | 41200100006 | H | administracion | | A |

Insertar Guardar Anular Modificar Actualizar Consultar Salir

Figura 14- Gestor de Asientos Contables.

Fuente: Elaboración Propia.

Contabilidad 27/05/2017 18:53

Datos Contables:

Número de Transacción Código Tipo de la Transacción Clase de la Transacción

Cl. Tipo de la Operación Número Operación Transacción Cod. Cuenta Contable

Moneda Debe Haber Fecha de la Transacción

Fecha Contable Cod. del Status

Figura 15- Gestor de Contabilidad.

Fuente: Elaboración Propia.

Figura 16- Gestor de Beneficiarios.

Fuente: Elaboración Propia.

Figura 17- Gestor de Fideicomisos.

Fuente: Elaboración Propia.

Para el desarrollo de la funcionalidad de Login se realizó el diseño e implementación de una interfaz más atractiva que la actual, donde el cliente ingresará su nombre de usuario y contraseña

previamente registrados en el sistema.

Inicialmente la aplicación debió tener su autenticación con el directorio activo de Banco Mercantil por lo que la funcionalidad de Login no iba a ser implementada. Sin embargo, en vista de que la información a migrar será utilizada solamente por la fiduciaria de Seguros Mercantil que cuenta con un personal administrativo limitado, la empresa Banco Mercantil tomó la decisión que para esta primera versión no se realizará dicha autenticación, sino que FSOSEG contara con su propio ingreso.

Las validaciones se realizaron en base a la correspondencia del usuario y su clave, en donde en caso de ser inválidas, el cliente visualiza un mensaje de error. Por otro lado el *back-end* de la funcionalidad estuvo orientado a implementar en el controlador los procedimientos correspondientes para la conexión con la base de datos y la llamada a las funciones del modelo que hiciera las validaciones necesarias para el correcto ingreso a la aplicación.

En la figura 18 se puede apreciar la interfaz de la funcionalidad de Login.

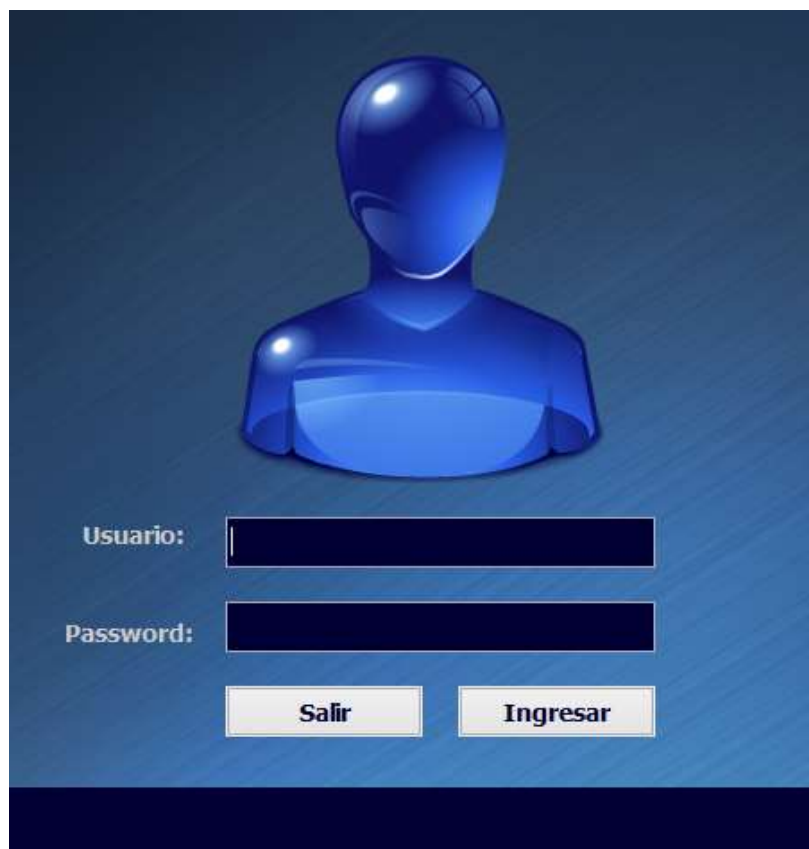
La imagen muestra una interfaz de usuario para el login. En el centro superior hay un icono de un busto humano azul brillante. Debajo del icono, a la izquierda, están las etiquetas 'Usuario:' y 'Password:'. A la derecha de cada etiqueta hay un campo de entrada rectangular. En la parte inferior, hay dos botones rectangulares: 'Salir' a la izquierda y 'Ingresar' a la derecha. El fondo de la interfaz es azul oscuro con un patrón sutil de líneas diagonales.

Figura 18- Ingreso a la Aplicación.

Fuente: Elaboración Propia.

En caso de tener un ingreso exitoso, la aplicación desplegará la vista del menú principal. En

caso de tener un ingreso fallido, la aplicación mostrará un mensaje de error.

A continuación se muestra la interfaz en el caso de éxito o fracaso del inicio de sesión en las figuras 19 y 20.

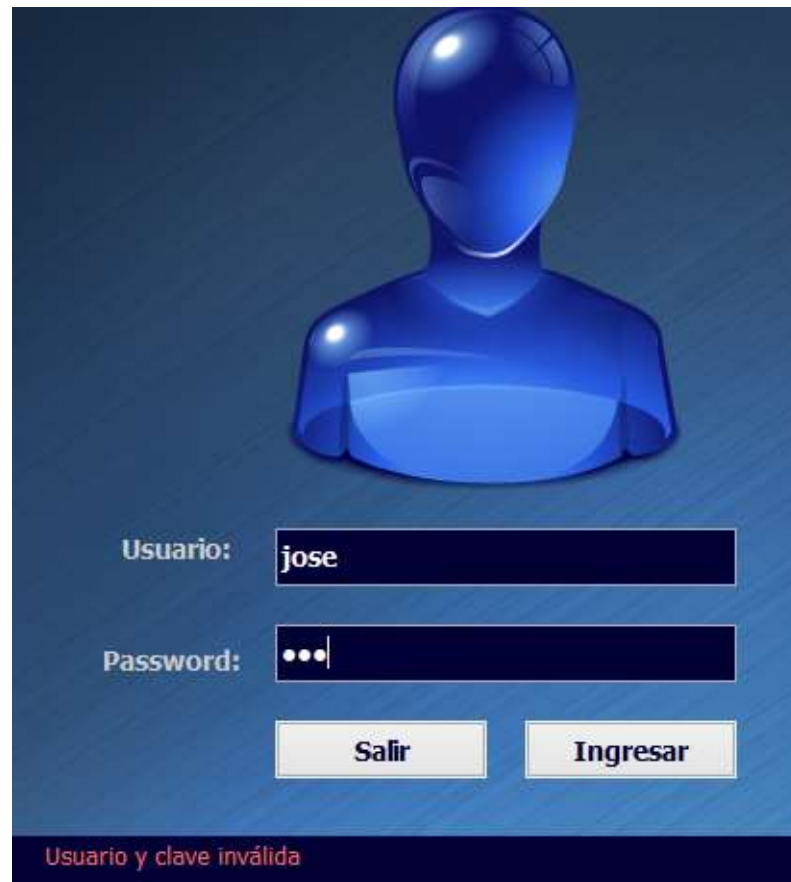


Figura 19.- Ingreso fallido a la Aplicación.

Fuente: Elaboración Propia.



Figura 20- Ingreso exitoso a la Aplicación.

Fuente: Elaboración Propia.

Un desarrollo que se derivó de la implementación de Login fue la gestión de usuarios y sus perfiles. Dado a que la información sería migrada, según las especificaciones del cliente la aplicación no contará con una interfaz de registro de usuarios previo al ingreso, sino que los nuevos ingresos deberán ser incluidos una vez se ha ingresado a la aplicación, por lo que era necesario la gestión de esos registros a través de una interfaz intuitiva y amigable.

Luego de realizar el diseño e implementación de las vistas, se implementaron las validaciones de los campos que conforman el formulario, seguido del desarrollo a nivel de *back-end* en el controlador de los procedimientos correspondientes que realizan las invocaciones en el modelo para hacer las inserciones, modificaciones y/o anulaciones en la aplicación. Además se mantuvo el uso de mensajes para informar al usuario en caso de hacer una acción indebida o de confirmación de la misma.

A continuación se presentan en las figuras 21, 22 y 23 el diseño de las interfaces de gestión de usuarios y sus perfiles.

Usuarios 26/05/2017 11:09

Detalle General Detalle de Conexión

Codigo de Usuario Codigo de Perfil Nombre de Usuario

Codigo Unidad Telefono de Usuario Estatus de Usuario

Insertar Guardar Anular Modificar Actualizar Consultar Salir

Figura 21.- Gestión de Usuarios.

Fuente: Elaboración Propia.

Perfil 26/05/2017 11:09

Detalle Perfil

Codigo de Perfil Descripción del Perfil

Perfiles

| Titulo 1 | Titulo 2 | Titulo 3 |
|----------|----------|----------|
| | | |
| | | |
| | | |

Insertar Guardar Activar Anular Asignar Consultar Salir

Figura 22.- Gestión de Perfiles.

Fuente: Elaboración Propia.

| Condición del Perfil | Estado de Condición | Seleccionar |
|----------------------|---------------------|--------------------------|
| C001 | Activo | <input type="checkbox"/> |
| C002 | Activo | <input type="checkbox"/> |
| C003 | Activo | <input type="checkbox"/> |
| C004 | Activo | <input type="checkbox"/> |
| C005 | Activo | <input type="checkbox"/> |
| C006 | Inactivo | <input type="checkbox"/> |

Figura 23.- Gestión de Acciones de Usuario.

Fuente: Elaboración Propia.

4.4.2 Segunda Iteración: Distribución de Rendimiento por Contrato y Beneficiario

Esta parte del proyecto estuvo dedicada a cumplir con el objetivo “Programar y Desarrollar la funcionalidad de distribución de rendimiento por contrato y beneficiario”. La etapa inicial de esta iteración estuvo enfocada en el diseño e implementación de la interfaz, donde sólo se ingresará la información correspondiente al rendimiento y asignación de intereses. En vista de que la aplicación a desarrollar no contará con el módulo de manejo de inversiones, se debió desarrollar un proceso de captura de datos que almacenara estos valores, para realizar el proceso de distribución por contrato y beneficiario.

Se desarrolló el proceso de inserción de los datos por parte del usuario, esto comprende los montos a distribuir y la fecha a la que corresponde la información. Para ello se hizo uso de un formulario con campos de texto y un campo de calendario dinámico para el ingreso de la fecha,

además de los botones para confirmar la ejecución de las acciones.

Seguidamente se procedió con el proceso de validaciones de los campos, haciendo énfasis en el formato de la fecha introducida. Así mismo se construyeron mensajes de errores para cada tipo de acción indebida o datos mal suministrados, donde se le indica al usuario el motivo del error.

Posteriormente se implementó el *back-end* de la funcionalidad, dónde a través de la invocación de *store procedure* al servidor se ingresa el registro en la base de datos y se envía una respuesta de confirmación al cliente.

Una vez generado el cierre diario o mensual se debe consultar si en esa fecha se debe generar el pago de rendimiento de algún fideicomiso. Para ello se debe hacer el cálculo de la participación de los contratos, que permite determinar la proporción que le corresponde a cada uno y sus beneficiarios. Una vez realizado esto, se deben generar las transacciones a nivel administrativo y contable que correspondan según cada fideicomiso, además de actualizar los saldos por contrato y beneficiario.

Para implementar esto, se desarrollaron los procedimientos correspondientes en el controlador para verificar si en esa fecha de cierre se debería efectuar algún pago, y para hacer el cálculo de la participación y distribución según afecte.

En las figuras 24 y 25 se puede observar la inserción de los valores por concepto de distribución de rendimiento, y la interfaz de cierre diario y mensual.

Pago de Rendimiento

Por favor ingrese los valores correspondientes, obtenidos en su informe de LA generado

Asignación Diaria en Bs.F:

Asignación Diaria en USD:

Rendimiento en Bs.F:

Rendimiento en USD:

Fecha del Proceso: 

Figura 24.- Ingreso de Asignación Diaria y Rendimiento.

Fuente: Elaboración Propia.

Cierre Diario

Cierre a la fecha: **2017/05/27** Primer Cierre

1.0 - Chequeo de Caja
2.0 - Inicio Cierre
3.0 - Contabilidad
4.0 - Recalculo saldo beneficiario
5.0 - Recalculo saldo Remba

Figura 25.-Ventana de Cierre Diario.

Fuente: Elaboración Propia.

4.4.3 Tercera Iteración: Manejo de Reportes y Generación de Trazas de Auditoría.

Esta etapa estuvo orientada en base a la consecución de los objetivos “Programar y desarrollar la funcionalidad de reportes” e “Incluir trazas de auditoría”. El módulo de reportes en la aplicación actual en *Main Frame* se lleva a cabo en otra herramienta, por lo que se decidió que en FSOSEG se pudieran generar los reportes básicos para la consulta del personal de Seguros Mercantil. Mientras que las trazas de auditoría resultan muy importantes para determinar qué acciones realizan los usuarios en el sistema.

Para el módulo de reportes se diseñó e implementó una interfaz que permitía filtrar por fecha, fideicomiso y/o beneficiario los diversos reportes que se podían ejecutar en la aplicación. Entre ellos se tiene el Balance de Comprobación el cual consiste en un resumen del saldo contable de las cuentas de tipo Activo, Pasivo y Capital. Luego se tiene el reporte de Movimiento Contable por Cuenta y Fideicomiso donde se detalla los movimientos por cada cuenta perteneciente a un fideicomiso y su saldo actual. Finalmente se tiene el Estado de Cuenta por Fideicomiso y Beneficiario en los cuales se detallan los saldos de las cuentas de acuerdo los movimientos de débito y crédito realizados en una fecha determinada.

Por cada reporte se implementó un controlador y un modelo con la intención de modularizar el desarrollo y disminuir el acoplamiento entre dependencias, permitiendo crear una mejora en la mantenibilidad de los procesos. Cada controlador hace la verificación de los datos ingresados y luego realiza la invocación al modelo correspondiente para extraer de la base de datos la información solicitada.

A continuación se muestran en las figuras 26, 27 y 28 la interfaz para generar los reportes y un modelo del formato de cada uno.

Generar Reporte

Parámetros

Inicio Al Final

Periodo: CA

Fideicomiso:

Persona:

Tipo de Reporte:

Generar Cancelar

Figura 26.- Generar Reportes.

Fuente: Elaboración Propia.

Port. Seg. Mercantil
Administración de Fideicomiso

Balance de Comprobación

2017-05-01 Al 2017-05-31

| Cuenta | Descripción de la cuenta Contable | Tipo | Monto |
|-------------|------------------------------------|---------|-----------|
| 42200100001 | Impuesto por pagar | Pasivo | -12767386 |
| 43101100001 | Fideicomisos de inversion | Capital | 79957280 |
| 41801102001 | Garantias mobiliarias | Activo | -10000 |
| 41801103001 | Garantia de titulos valores | Activo | -6006 |
| 41801104001 | Otros fideicomisos de garantias | Activo | 2000 |
| 43103101001 | Garantias de creditos inmobiliaria | Capital | 2000 |

Figura 27.-Generar Reportes-Balance de Comprobación.

Fuente: Elaboración Propia.

Moneda Bs. Tasa de Cambio 0,00

Información Confidencial

FE2035B

PORT. SEG. MERCANTIL

Fecha: 27/05/17

Página: 1

Movimiento Mensual de Contabilidad por Cuenta y por Fideicomiso

| NUMFSO | DEBITOS | CREDITOS | SALDO ACTUAL |
|--------------|---------|----------|--------------|
| 41400100050 | 0.0 | 0.0 | 0.0 |
| 451001000050 | 0.0 | 0.0 | 0.0 |

Figura 28.- Generar Reportes-Mov Mensual.

Fuente: Elaboración Propia.

Es importante resaltar que a través de todo el proceso de implementación de los módulos y funcionalidades mencionados se agregaron las trazas de auditorías correspondientes para registrar en el log del sistema todas las acciones ejecutadas por los usuarios, garantizando el objetivo de “Incluir Trazas de Auditoría” mencionado anteriormente.

En la figura 29 se podrá visualizar la consulta del log de auditoria según la permisología correspondiente.



| Usuario | Módulo | Opción | Perfil | Fecha del Ev... | Hora del Evento | Descripción |
|---------|--------|--------|--------|-----------------|-----------------|--|
| jose | 04 | 03 | 1 | 20/04/2017 | 15:00:03 | insert into t_persona values('jose', 'jose', 'jose', 'cod_estado_nacimiento=masa', 'cod_persona... |
| jose | 04 | 03 | 1 | 20/04/2017 | 15:00:37 | insert into t_persona values('jose', 'jose', 'jose', 'cod_estado_nacimiento=masa', 'cod_persona... |

Figura 29.-Visor de Log de Auditoría.

Fuente: Elaboración Propia.

En esta iteración también se desarrolló la interfaz de opciones e instalación del sistema como complemento a las trazas de auditoría. De esta manera se podía tener una información más detallada una vez se haya hecho la implantación en una estación de trabajo y poder gestionar los datos a raíz de algún cambio.

Se diseñó una interfaz para capturar la información de la instalación y otra para capturar la información de las opciones del sistema. Seguido de esto se implementaron las validaciones de los campos y en el controlador los procedimientos necesarios para invocar en el modelo los métodos necesarios que realizan la inserción, modificación y anulación de registros en base de datos.

A continuación en las figuras 30 y 31 se puede apreciar el diseño de las interfaces correspondientes.

Instalación del Sistema 28/5

Datos Básicos | Hardware | Software | Base de Datos

| | | |
|--------------------------|-------------------------------------|------------------------------------|
| Código del Sistema | Descripción del Sistema | Versión del Sistema |
| Fecha de Instalación | Fecha de Actualización | Manejador de Doc. del Sistema |
| Nombre del Servidor | Autenticación del Directorio Activo | Manejo Multi-User |
| Manejo Chat Ayuda Online | Dirección IP | Inst. Soft. Manejador de Contenido |
| Puerto de Conexión | | |

Insertar | Guardar | Anular | Modificar | Actualizar | Consultar | Salir

Figura 30.-Vista de Instalación del Sistema.

Fuente: Elaboración Propia.

Configuración 28/5/2017 12:40

Datos Básicos | Datos Contables | Sistema

| | | |
|----------------------|---------------------------|----------------------------|
| Código de la Empresa | Cédula de Identidad o RIF | Razón Social de la Persona |
| Código Personal | Código País | Código Postal |
| Formato de Fecha | Formato de la Hora | Edición |
| Dirección | Teléfono | Código |
| Persona de Contacto | | |

Insertar | Guardar | Anular | Modificar | Actualizar | Consultar | Salir

Figura 31.-Vista de Configuración del Sistema.

Fuente: Elaboración Propia.

4.4.4 Resumen de Funcionalidades Desarrolladas.

En la figura 32 se observa un resumen del trabajo de desarrollo de software realizado, en el cual se puede observar por módulo el listado de casos de uso implementados:

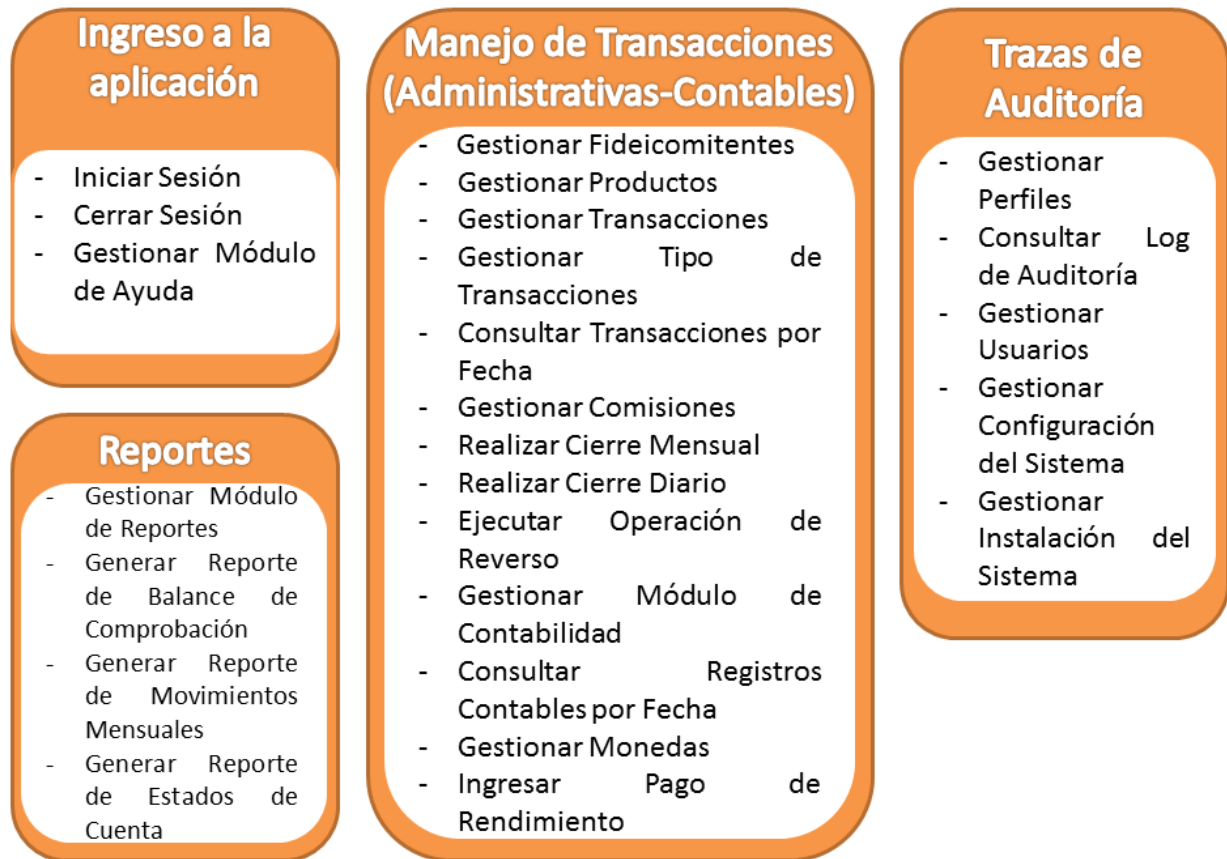


Figura 32.-Resumen de Funcionalidades Desarrolladas.

Fuente: Elaboración Propia

4.5 Fase de Pase a Producción.

Esta fase fue constituida por todas las tareas relacionadas con el pase a producción de la nueva versión de FSOSEG, incluyendo la configuración, pruebas, preparación e instalación del software necesario para el ambiente, implantación del sistema y revisión final de la documentación.

Aunque por razones de fuerza mayor en las dependencias de Banco Mercantil, ajenas

completamente a CBT y el desarrollo de FSOSEG, no se pudo completar el pase a producción en las instalaciones de Banco Mercantil ni la capacitación de usuarios finales. Sin embargo la aplicación se implantó en las estaciones de trabajo de CBT para realizar las pruebas correspondientes. Así, mediante SCRUM se llevaron a cabo la iteración correspondiente a su fase de Pase a Producción.

4.5.1 Primera Iteración: Preparación de Ambiente de Implantación

Luego de finalizado el proceso de implementación de todas las funcionalidades y módulos previos, se procedió con la ejecución de pruebas unitarias y de integración, con la finalidad de garantizar el correcto funcionamiento del sistema en cada una de sus partes y en su totalidad. Cumpliendo así con el objetivo “Realizar pruebas integrales de la aplicación y pruebas de aceptación”.

Para esto se creó el documento de casos de pruebas, el cual consta de distintos escenarios para evaluar el comportamiento de la aplicación. Tales pruebas fueron efectuadas por el personal de CBT arrojando como resultado un porcentaje de satisfacción del 100% (Apéndice F).

Por otro lado, se elaboraron de igual forma los siguientes documentos, que además se pueden consultar desde el sistema como herramienta de ayuda:

- Manual de usuario. (Apéndice C).
- Manual del sistema. (Apéndice B).

Posteriormente, en esta iteración se realizó un análisis referente a las características de los servidores disponibles para alojar el sistema, así como los requisitos mínimos necesarios para el proceso. (Ver Apéndice A, “*Documento de Requerimientos Técnicos*”).

Seguidamente se procedió con la instalación de los componentes en uno de los ambientes de prueba de la empresa CBT, incluyendo sistema operativo, manejador de base de datos, herramientas, paquetes y librerías, cumpliendo con el objetivo de “Entrega final y pase a producción”.

Una vez instaladas todas las dependencias, hechas todas las configuraciones y desplegado a cabalidad el sistema se realizaron una serie de pruebas de aceptación por parte de la gerencia de CBT con el fin de contar con una etapa de verificaciones extras.

Durante la ejecución de las pruebas y comparaciones la gerencia de CBT se percató de

requisitos y funcionalidades faltantes a la aplicación, y aunque no estaban contempladas en el plan de trabajo del proyecto, debido a la importancia de las mismas y ya que el tiempo lo permitía se procedió a realizar las actividades y tareas necesarias para garantizar la completitud y calidad del sistema. Entre ellas destacan la gestión de instalación y configuración del sistema que complementa el módulo de trazas de auditoría, de esta manera se agrega mayor seguridad a la aplicación, pues se almacena información relevante acerca de hardware donde se está alojando y del usuario que está realizando dichas acciones.

Dichas labores se llevaron a cabo a tiempo y fueron aprobadas por la gerencia de la empresa con un porcentaje de aceptación del 100%.

4.6 Fase de Elaboración de Entregables.

Esta fase corresponde a la etapa final de proyecto de pasantía, la cual se enfocó en cumplir con los objetivos: “Realizar la documentación del sistema”. Por lo que, durante la misma se revisó y actualizó en última instancia toda la documentación propia de sistema, los artefactos a entregar en la empresa, a saber: Requerimientos Técnicos, Manual del Sistema, Manual del Usuario, y aquellos tomados de la metodología RUP, a saber: Documento de Arquitectura del Software, Documento de Especificación de Requerimientos del Sistema y Documento de Casos de Pruebas. Finalmente se redactó el informe final de pasantía.

A continuación se presenta la lista de documentos elaborados o actualizados, los cuales fueron entregados en su versión definitiva a la Gerencia de CBT al final de esta etapa:

- ✓ Requerimientos Técnicos (Apéndice A).
- ✓ Manual del Sistema (Apéndice B).
- ✓ Manual del Usuario (Apéndice C).
- ✓ Documento de Arquitectura del Software (Apéndice D).
- ✓ Documento de Especificación de Requerimientos del Sistema (Apéndice E).
- ✓ Documento de Casos de Pruebas (Apéndice F).

CONCLUSIONES Y RECOMENDACIONES

El presente proyecto ha tenido como objetivo general “Implementar la aplicación y sus funcionalidades para la gestión de fideicomisos de Seguros Mercantil que incluyen reportes, obtención del rendimiento por título y distribución del mismo por beneficiario y manejo de transacciones a nivel administrativo y contable” el cual se ha logrado exitosamente. Para tal logro se trazó una ruta de objetivos específicos debidamente concatenados e interrelacionados, que facilitaron la labor. En tal sentido, seguidamente se presentan las conclusiones alcanzadas mediante el desarrollo de cada uno de ellos.

En principio es importante mencionar que aunque la mayoría de las necesidades y requerimientos de la organización no estuvieron claros desde un principio, se realizó la planificación y diseño necesarios para no generar demoras en la entrega del producto final. Además se contó con una buena base para el diseño y desarrollo del software, ya que, se hizo una buena escogencia de las herramientas durante la primera etapa del sistema, por lo que, se facilitó la integración de los nuevos módulos desarrollados. En gran parte el hecho se estableció en la simpleza del manejo entre los *frames* a través de Netbeans y en las características de Java como lenguaje de programación amigable para este tipo de proyectos.

Ahora bien, en cuanto al primer objetivo “Programar y Desarrollar las funcionalidades de ingreso a la aplicación” se puede decir que:

- A pesar de que no fue una funcionalidad planificada desde el inicio del proyecto, fue desarrollada en su totalidad, se realizaron las pruebas unitarias, integrales, de aceptación y de comparación con el mismo módulo en la versión anterior del sistema y se obtuvo un resultado del 100% de agrado por parte de la gerencia de CBT.

En lo que respecta al segundo objetivo “Programar y desarrollar la funcionalidad del manejo de transacciones (administrativo y contable)” se concluye que:

- El proceso se desarrolló a totalidad, con todas las funciones que allí estaban contempladas 100% operativas e inclusive con la función de ejecutar el reverso la cual fue añadida para facilitar la modificación de las transacciones en caso de que ocurriera un error en el cierre diario o mensual.

➤ Esta funcionalidad representa lo más importante en el desarrollo de la aplicación, pues a través de este módulo se generan de forma manual y automática todas las entradas y salidas de dinero a nivel administrativo y contable. Por lo que era necesario realizar una interfaz intuitiva y amigable que permitirá gestionar el proceso.

El tercer objetivo estuvo constituido por la “Programar y desarrollar la funcionalidad de distribución del rendimiento por contrato y beneficiario”, al respecto es importante mencionar:

➤ No se migró como tal dicha funcionalidad de la aplicación actual, pues el manejo de portafolio y cartera de inversiones se ejecuta en otra aplicación que no es la de BFE. En vista de eso se diseñó una interfaz que permitiera capturar los datos que deberían ser distribuidos por contrato y beneficiario, y se logró implementar a nivel de back end la ejecución de dicho proceso.

Con lo que al cuarto objetivo “Programar y desarrollar la funcionalidad de reportes” concierne, se puede afirmar que:

- El proceso se desarrolló a totalidad, con todas las funciones que allí estaban contempladas 100% operativas e inclusive con la función de ejecutar el reverso la cual fue añadida para facilitar la modificación de las transacciones en caso de que ocurriera un error en el cierre diario o mensual.
- Se diseñó una interfaz amigable que permitiera la generación de los diversos reportes del sistema filtrando por fecha, fideicomiso y/o beneficiarios según sea el caso.

El quinto objetivo del proyecto se basó en “Incluir Trazas de Auditoría”, lo cual se realizó con éxito, puesto que en cada una de las tareas desarrolladas se incluyó el log correspondiente.

En cuanto al sexto objetivo de “Realizar la documentación adecuada del sistema”, se tiene:

- Fueron elaborados todos los documentos solicitados por la empresa CBT, los cuales son: Manual de Usuario, Manual del Sistema, Requerimientos Técnicos, Documento de Arquitectura del Software, Documentos de Especificación de Requerimientos del Sistema, Casos de Prueba.

A lo que al séptimo objetivo de “Realizar pruebas integrales de la aplicación y pruebas de aceptación” se refiere:

- Cada uno de los módulos y funcionalidades implementadas se probó en conjunto con el personal y la gerencia de CBT, tanto unitariamente como a integralmente. En este

caso se obtuvo un porcentaje de aceptación del 100% por parte de la gerencia.

Y en con respecto al octavo objetivo “Entrega Final y Pase a Producción” cabe destacar que:

- Se realizaron las investigaciones referentes a los recursos y tecnologías necesarias para la implantación del sistema en un ambiente similar al de los equipos en las estaciones de trabajo finales.
- No se realizó como tal el pase a producción a un cliente en vivo por falta de acuerdos internos en Banco Mercantil. Sin embargo la aplicación fue implantada en algunos equipos provistos por CBT donde se realizaron las pruebas ya mencionadas y se entregó todo el desarrollo con la documentación.

Es importante también mencionar que la metodología SCRUM fue un pilar fundamental para lograr la consecución de los objetivos, puesto que gracias a su flexibilidad y enfoque se pudo facilitar la adaptación a las exigencias del cliente, además de haber fomentado una evaluación y retroalimentación continua a lo largo del proceso de desarrollo de software.

Por la parte de elaboración de artefactos, la metodología RUP fue de gran ayuda pues al tener un enfoque más disciplinado se pudo realizar la documentación requerida por el cliente y la adecuada para este tipo de desarrollo.

La aplicación FSOSEG cuenta con un alto nivel de parametrización en su diseño de interfaz y modularización del código gracias a su bajo nivel de acoplamiento, lo permite la adaptación a nuevos requerimientos favoreciendo la escalabilidad y mantenibilidad del producto.

Para finalizar se hacen las siguientes recomendaciones para mejorar aún más la aplicación a través de futuros desarrollos:

- Implementar una herramienta para la colaboración y el control de versiones, lo que facilitaría el trabajo en equipo para los desarrolladores. Así como también permitiese tener un mejor registro de las versiones del sistema. Por ejemplo pudiese considerarse el empleo de tecnología como la que ofrece GitHub o BitBucket.
- Finalizar la implementación de la modificación de la contraseña para que sea adaptada una vez haya expirado la vigencia de la misma, de esta manera se contribuye a tener un software completo y de calidad.
- Evaluar la documentación entregada y en el código a la hora de adaptar nuevas funcionalidades a la aplicación, con la intención de no desmejorar la modularidad lograda y aceptada con este desarrollo.

- Determinar si es factible implementar la eliminación de aquellos registros que sean insertados en base de datos con datos erróneos, siempre y cuando este proceso sea el mismo día de la inserción. De esta manera se puede evitar contaminar la base de datos con información que no sea correcta.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Nación Fideicomisos, S.A. (2015) *Fideicomiso*. Recuperado de: <http://www.nacionfides.com.ar/fideicomiso.html>. Consultado el 24 de abril de 2017.
- [2] Miller, F. (1999). *Curso de Contabilidad*. Mc Graw Hill.
- [3] Bimodal Technology, C.A. (2016). *Bimodalit-com*. Recuperado de: <http://www.bomodalitil.com> .Consultado el 24 de abril de 2017
- [4] Solustrust. (2012). *Origen de la palabra Fideicomiso*. Recuperado de: <http://www.solustrust.com/preguntasFrecuentes/origen>. Consultado el 24 de abril de 2017.
- [5] FIFONAFE. (2014). *Concepto y Partes de un Fideicomiso*. Recuperado de: <http://www.fifonafe.gob.mx/acercade/concepto> .Consultado el 24 de abril de 2017.
- [6] Garrido, E. (2015). *Riesgo y Rendimiento by Everardo Garrido on Prezi*. Recuperado de: <https://prezi.com/tqirmzrds0ut/riesgo-y-rendimiento/> . Consultado el 24 de abril de 2017.
- [7] Molina, L (2013). *Aplicaciones de 2, 3 y n Capas | Desarrollo de Aplicaciones para ambientes distribuidos*. Recuperado de: <https://laurmolina7821.wordpress.com/1-1-3-aplicaciones-de-2-3-y-n-capas/> Consultado el 24 de abril de 2017.
- [8] Reenskaug, T. Coplien, J. (2009). *The DCI Architecture: A New Vision of Object-Oriented Programming*. Recuperado de: http://www.artima.com/articles/dci_vision.html. Consultado el 24 de abril de 2017.
- [9] Younes, S. (2009) *The Client/Server Model*. Recuperado de: http://www.samuraii.net/writings/notes_networksintro.htm , Consultado el 24 de abril de 2017.

- [10] Lenguajes de Programación. (2016). *Lenguajes de Programación, programación java*. Recuperado de: <http://www.lenguajes-de-programacion.com/programacion-java.shtml>. Consultado el 24 de abril de 2017.

- [11] Sistemas Informática & Electrónica. (2014). *JasperReport & IReport en Netbeans | Sistemas Informática & Electrónica*. Recuperado de: <https://jossjack.wordpress.com/2014/06/15/jasperreport-ireport-en-netbeans/>. Consultado el 24 de Abril de 2017.

- [12] The Apache Software Foundation. (2017). *Apache HTTP Server Project*. Recuperado de: <https://httpd.apache.org/>, Consultado el 24 de Abril de 2017.

- [13] Chipia, J. (2010). *Análisis del uso de metodologías de desarrollo y métricas de calidad de software en aplicaciones educativas*. Recuperado de: <http://revisionreflexivaeducativa.blogspot.com/2010/03/analisis-de-la-importancia-del-usode.html> , Consultado el 24 de Abril de 2017.

- [14] Scrum Manager, Body of Knowledge. (2013). *Modelo original de Scrum para desarrollo de software*. Recuperado de: https://www.scrummanager.net/bok/index.php?title=Modelo_original_de_Scrum_para_desarrollo_de_software , Consultado el 24 de Abril de 2017.

- [15] Gabay, A. (2014). *Métodos ágiles de gestión de proyectos*. Recuperado de: <https://es.slideshare.net/Alejandroslide/metodologias-agiles-de-gestion-de-proyectoagilevspmiby-a-gabaypmibaagosto2014>. Consultado el 24 de Abril de 2017.

- [16] Universidad Pedagógica Experimental Libertador. (2006). *Manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales*. Recuperado de: <http://unefapostgradoanzoategui.jimdo.com/textos-manuales-y-normativas/manual-de-normas-upel/> , Consultado el 24 de Abril de 2017.

- [17] Quatrani, T. (1999). *Visual Modeling with Rational Rose 2000 and UML*. Addison Wesley.

- [18] Rumbaugh, J., Jacobson, I., Booch, G. (1999). *The Unified Modeling Language Reference Manual*. Addison Wesley.

