



UNIVERSIDAD CATÓLICA ANDRÉS BELLO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**“Portal Web de Registro de Empresas Ordenantes para Domiciliaciones y
Procesamiento de Operaciones de Créditos Directos ordenadas por
Organismos Oficiales a través de la Cámara de Compensación Electrónica”**

TRABAJO INSTRUMENTAL DE GRADO

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN INFORMÁTICA

REALIZADO POR

JAVIER ANGEL

TUTOR EMPRESARIAL

ANDRÉS SILVA

TUTOR ACADEMICO

RICARDO CASANOVA

FECHA

ENERO DE 2013



UNIVERSIDAD CATÓLICA ANDRÉS BELLO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**“Portal Web de Registro de Empresas Ordenantes para Domiciliaciones y
Procesamiento de Operaciones de Créditos Directos ordenadas por
Organismos Oficiales a través de la Cámara de Compensación Electrónica”**

**Este Jurado; una vez realizado el examen del presente trabajo ha evaluado su
contenido con el resultado: Quince (15) puntos**

JURADO EXAMINADOR:

Firma:

Nombre: Ricardo Casanova

Firma:

Nombre: Susana García

Firma:

Nombre: Rafael Andara

REALIZADO POR
TUTOR EMPRESARIAL
TUTOR ACADEMICO
FECHA

JAVIER ANGEL
ANDRÉS SILVA
RICARDO CASANOVA
ENERO DE 2013

DEDICATORIA

A mi madre querida por todo su cariño y comprensión, así como por todo su apoyo y esfuerzo para que lograra mis metas.

A mi padre y hermanos por quererme, creer en mí y siempre estar ahí. Por todo su apoyo y cariño.

En memoria de Rayden y Odín, los mejores compañeros que la vida me pudo haber regalado. Llenaron mi vida de luz y alegría, nunca los olvidaré, los amo y espero que estén juntos y bien donde quiera que estén.

Los quiero a todos

AGRADECIMIENTOS

Después de tanto esfuerzo es necesario agradecer a todas las personas que hicieron posible que llegara hasta aquí y los que me acompañaron en el camino.

Primero que nada a mi madre, mi padre, Will, Cheo y Katherine, por siempre estar ahí, apoyándome en cada paso que daba, espero haber logrado que se sientan orgullosos de mí, porque yo lo estoy de ustedes, gracias por todo a todos. Gracias por el esfuerzo, por la ayuda, por el apoyo y por la confianza en mí.

A mi tutor Andrés por toda la ayuda y la colaboración para que lograra el desarrollo del TIG en el tiempo justo, sin él todo se hubiese hecho mas cuesta arriba.

A los viejos amigos, a la gente de la plaza por ser mi entretenimiento durante tantos días, desperdiciábamos esos días pero nunca nos aburrimos, eso es seguro.

Gracias a mi compañero de proyectos Lucho que me acompañó casi toda la carrera, capaz si no era así no se graduaba ninguno nunca. A Nelson (aunque mas a su mamá que a él) que me adoptó por un semestre completo y me salvó varias veces de no dormir en una plaza. A la gente de la UCAB, a todo el que me dio asilo en algún momento (son muchos). A Guillelon, Gabyllis, Lisbé, Freddy y Gomita por su amistad, por andar pendientes de que saliera adelante con esto y en algunos casos hasta regañándome para que lo hiciera.

En fin: ¡Gracias totales!

Javier Angel

ÍNDICE DE CONTENIDOS

ÍNDICE DE CONTENIDOS

ÍNDICE DE CONTENIDOS	3
ÍNDICE DE FIGURAS.....	9
SINOPSIS	10
INTRODUCCIÓN	1
1. PLANTEAMIENTO DEL PROBLEMA	3
1.1 Necesidades de la Empresa:.....	3
1.2 Solución Propuesta.....	5
1.3. Objetivos	8
1.3.1 Objetivo General.....	8
1.3.2 Objetivos Especificos.....	8
1.4 Alcance y Limitaciones.....	10
1.5 Justificación	13
2. MARCO REFERENCIAL.....	14
2.1 Cámara de Compensación Electrónica (CCE).....	14
2.2 Sema Cleary System (SCS)	14
2.3 Plataforma de Acceso al Usuario (UAP)	15
2.4 Domiciliaciones	15
2.5 Créditos Directos	15
2.6 Java	16
2.7 Lightweight Directory Access Protocol (LDAP).....	16
2.8 ZK Framework.....	16
2.9 Apache Tomcat	17

ÍNDICE DE CONTENIDOS

2.10 XML.....	17
2.11 ZUL.....	18
2.12 Lenguaje SQL.....	18
2.13 Mapeo Objeto-Relacional.....	18
2.14iBatis.....	19
2.15 Inyección de Dependencias.....	19
2.16 Spring Framework.....	19
2.17 Spring Security.....	20
2.18 Quartz.....	21
2.19 Control de Versiones.....	21
2.20 Log4J.....	22
2.21 JasperReports.....	22
3. MARCO METODOLÓGICO.....	23
3.1 Metodología de Programación Extrema.....	23
3.1.1 Historias de Usuario.....	23
3.1.2 Prácticas.....	24
3.1.3 Proceso.....	27
4. DESARROLLO.....	30
4.1 Fase de Exploración.....	30
4.2 Fase de Planificación de Entregas.....	33
4.3. Iteraciones.....	34
4.3.1Primera Iteración: Estructuras de Datos.....	34
4.3.2 Segunda Iteración: Front End.....	39

ÍNDICE DE CONTENIDOS

4.3.3 Tercera Iteración: Módulo de Domiciliaciones	40
4.3.4 Cuarta Iteración: Módulo de Créditos Directos	46
4.3.5 Quinta Iteración: Módulo de Autenticación y Control de Acceso al Portal	52
4.3.6 Sexta Iteración: Módulo de Administración de Usuarios	55
5. RESULTADOS.....	58
6. CONCLUSIONES Y RECOMENDACIONES	64
6.1 Conclusiones	64
6.2 Recomendaciones	66
REFERENCIAS BIBLIOGRÁFICAS.....	67
APÉNDICES.....	69
Apéndice 1: Historias de Usuario	69
Apéndice 2: Plan de Entregas	70
Apéndice 3: Historias de Usuario de la Primera Iteración.....	71
Apéndice 4: Estructura del módulo de Créditos Directos.....	71
Apéndice 5: Ejemplo de formato para solicitud de creación de estructuras de datos	72
Apéndice 6: Pruebas de la Primera Iteración	73
Apéndice 7: Historias de Usuario de la Segunda Iteración.....	74
Apéndice 8: Pruebas de la Segunda Iteración.....	74
Apéndice 9: Historias de Usuario de la Tercera Iteración	75
Apéndice 10: Estructura de archivo de domiciliaciones.....	78
Apéndice 11: Pruebas de la Tercera Iteración	79

ÍNDICE DE CONTENIDOS

Apéndice 12: Archivos de Solicitud y Respuesta de Domiciliaciones generados	83
Apéndice 13: Historias de Usuario de la Cuarta Iteración.....	83
Apéndice 14: Formato de archivo para carga de Créditos Directos	84
Apéndice 15: Reglas de Validación y de Estructura para archivos de Créditos Directos	85
Apéndice 16: Diagrama de Clases del Módulo de Créditos Directos.....	86
Apéndice 17: Pruebas de la Cuarta Iteración.....	87
Apéndice 18: Archivo de Créditos Directos generado para las pruebas.....	88
Apéndice 19: Historias de Usuario de la Quinta Iteración	88
Apéndice 20: Pruebas de la Quinta Iteración.....	89
Apéndice 21: Historias de Usuario de la Sexta Iteración	91
Apéndice 22: Diagrama de Clases del Módulo de Administración de Usuarios y Reportes	92
Apéndice 23: Pruebas de la Sexta Iteración.....	92
Apéndice 24: Pantalla de inicio del portal	94
Apéndice 25: Funcionalidad Consulta General del módulo Gestión de Empresas	94
Apéndice 26: Reporte en formato Excel de la Gestión de Empresas	95
Apéndice 27: Reporte en formato PDF de la Gestión de Empresas	95
Apéndice 28: Detalle del formulario para Exclusión de Empresas	96
Apéndice 29: Detalle de la funcionalidad para realizar Solicitud de Afiliación..	96
Apéndice 30: Detalle de la funcionalidad para Solicitud de Desafiliación	96
Apéndice 31: Detalle del formulario para realizar la desafiliación	97

ÍNDICE DE CONTENIDOS

Apéndice 32: Detalle de funcionalidad de Consulta de Solicitudes	97
Apéndice 33: Detalle del formulario para la Actualización y Aprobación/ Denegación de Solicitud Pendiente por parte del usuario administrador.	98
Apéndice 34: Detalle de funcionalidad de Generación de Mensajes.....	98
Apéndice 35: Detalle del mensaje generado	98
Apéndice 36: Detalle del formulario estándar para la carga de archivos	99
Apéndice 37: Detalle de la funcionalidad de Consulta de Archivos de Solicitudes y Respuestas.....	99
Apéndice 38: Detalle de la funcionalidad de Consulta de Solicitudes y Respuestas.....	100
Apéndice 39: Reporte en formato PDF de solicitudes.....	100
Apéndice 40: Detalle de funcionalidad Configuración del Sistema (Institución Financiera)	101
Apéndice 41: Detalle de funcionalidad de Consulta de Archivos de Créditos Directos	101
Apéndice 42: Detalle de formulario para Anulación de Archivos.....	102
Apéndice 43: Consulta de detalle de archivo de créditos directos cargado	102
Apéndice 44: Consulta de detalle de los registros del archivo de créditos directos	103
Apéndice 45: Detalle de funcionalidad de parámetros de Créditos Directos ...	103
Apéndice 46: Reporte en formato PDF de carga de archivos de Créditos Directos	104
Apéndice 47: Detalle de funcionalidad Administración de Usuarios.....	104
Apéndice 48: Detalle de funcionalidad de edición de datos de usuario	105

ÍNDICE DE CONTENIDOS

Apéndice 49: Detalle de funcionalidad de otorgamiento o revocación de privilegios	105
---	-----

ÍNDICE DE FIGURAS

Figura 1 Esquema gráfica de una iteración de la metodología	28
Figura 2 Esquema del funcionamiento de la CCE y sus componentes.....	31
Figura 3 Modelo Entidad Relación para Módulo de Créditos Directos.....	35
Figura 4 Modelo Entidad Relación para nuevas estructuras para Modulo de Domiciliaciones	36
Figura 5 Procesos para solicitud de afiliación/desafiliación de empresas	41
Figura 6 Stored Procedures creados para las operaciones de Domiciliaciones	44
Figura 7 Representación gráfica del esquema de integración.....	47
Figura 8 Stored Procedures creados para las operaciones de Créditos Directos	49

SINOPSIS

El presente Trabajo Instrumental de Grado titulado "Portal Web de Registro de Empresas Ordenantes para Domiciliaciones y Procesamiento de Operaciones de Créditos Directos ordenadas por Organismos Oficiales a través de la Cámara de Compensación Electrónica", consiste en el diseño y desarrollo de cada uno de los módulos que componen el portal, para proporcionar a los usuarios representantes de las Instituciones Financieras Participantes en la Cámara de Compensación Electrónica, una vía para realizar estas operaciones de una manera más rápida y eficaz que el proceso utilizado actualmente.

En la actualidad, la Cámara de Compensación Electrónica carece de interfaces que permitan a los usuarios realizar el registro de empresas involucradas en el medio de Domiciliaciones, así como de realizar el procesamiento de pagos a través del instrumento de Créditos Directos. El proyecto desarrollado tiene la finalidad de proveer las interfaces para agilizar estas operaciones, reduciendo considerablemente el tiempo actual de respuesta de los procesos al ser automatizados. Todo esto dentro de un Portal que ofrece distintos niveles de acceso, dependiendo del perfil del usuario, ofreciendo una aplicación de uso sencillo, vistas amigables y el despliegue instantáneo de las funcionalidades solicitadas.

El proyecto se desarrolla basado en diferentes tecnologías que se complementan entre sí, teniendo como base el Framework Spring para realizar la inyección de dependencias de los objetos, así como de Spring Security para integrar la seguridad del portal al protocolo LDAP utilizado por el banco; iBatis, encargado de permitir la comunicación con la base de datos; ZK, para generar las vistas del portal, así como sus controladores, y distintas herramientas tales como Quartz para la programación de tareas, JasperReports para la generación de reportes, entre otros. Todo esto desarrollado dentro del marco de la metodología ágil XP que permite el desarrollo de los distintos módulos y funcionalidades de una manera completa y organizada.

INTRODUCCIÓN

El Banco Central de Venezuela (BCV) es una Institución Financiera Pública que vela por el equilibrio económico y el desarrollo ordenado de la economía venezolana, logrado esto por la convergencia funcional de sus unidades, entre ellas, la Cámara de Compensación Electrónica (CCE) con la cual garantiza la continuidad de los pagos ordenados por los clientes de las Instituciones Financieras Participantes (IFP) o en nombre propio desde una IFP a otra IFP.

A través de la Cámara de Compensación Electrónica se realiza la compensación de operaciones originadas por los medios de pagos: Créditos Directos y Domiciliaciones. Estas operaciones se realizan manualmente en la actualidad, con riesgos de pérdida de la información y con tiempos de respuesta bastante elevados que no satisfacen la creciente demanda de ambos servicios. Por lo tanto, resulta de gran importancia el desarrollo de una interfaz que permita a todas las Instituciones Financieras Participantes en la Cámara de Compensación Electrónica realizar el registro de estas operaciones de una manera rápida y eficaz, reduciendo los tiempos de respuesta y los riesgos de pérdida de información.

El Portal desarrollado ofrece a los usuarios, que ingresan en representación de las Instituciones Financieras Participantes, la oportunidad de realizar los procedimientos de registro de empresas ordenantes para las domiciliaciones y procesamiento de pagos de créditos directos desde una misma interfaz, de manera automatizada.

En el presente documento se realiza una descripción detallada del proceso de elaboración del Trabajo Instrumental de Grado. Se divide en los siguientes capítulos:

Capítulo I: Comprende el planteamiento del problema que dio origen al Trabajo Instrumental de Grado, así como la solución propuesta para resolver el mismo. Adicionalmente cuenta con los objetivos a cumplir durante el desarrollo del proyecto; su alcance, limitaciones y la justificación para la realización del proyecto.

Capítulo II: Contiene el marco referencial o base teórica necesaria para la comprensión de los términos relacionados con el desarrollo del proyecto, así como la explicación de todas las herramientas tecnológicas utilizadas en el desarrollo del mismo.

Capítulo III: Presenta una explicación de la metodología utilizada para la elaboración del proyecto y la descripción de las distintas fases que la comprenden.

Capítulo IV: Comprende la descripción detallada de cada una de las actividades realizadas para alcanzar los objetivos propuestos, siguiendo las prácticas y fases propuestas en el marco metodológico seguido.

Capítulo V: Describe los resultados obtenidos a partir de del desarrollo del proyecto, en función de los objetivos completados.

Capítulo VI: Está conformado por las conclusiones y recomendaciones que surgieron a partir del desarrollo del Trabajo Instrumental de Grado.

En última instancia, se incluyen las referencias bibliográficas que sirvieron de apoyo durante la realización del Trabajo Instrumental de Grado; así como los apéndices que ofrecen información adicional para la comprensión del proyecto.

1. PLANTEAMIENTO DEL PROBLEMA

1.1 Necesidades de la Empresa:

El Banco Central de Venezuela (BCV) es una Institución Financiera Pública que vela por el equilibrio económico y el desarrollo ordenado de la economía venezolana, logrado esto por la convergencia funcional de sus unidades, entre ellas, la Cámara de Compensación Electrónica (CCE) con la cual garantiza la continuidad de los pagos ordenados por los clientes de las Instituciones Financieras Participantes (IFPs) o en nombre propio desde una IFP a otra IFP.

El Banco Central de Venezuela ejerce un doble papel en la Cámara de Compensación Electrónica. Por un lado el de ente compensador y por otro como Institución Financiera Participante.

A través de la CCE se compensan operaciones originadas por los medios de pagos: Cheques, Créditos Directos (transferencias) y Domiciliaciones. Esta compensación se realiza a través de las cuentas bancarias únicas que cada IFP posee en el BCV.

La domiciliación de pago es una facilidad que prestan las empresas de servicios (luz, agua, televisión por cable, entre otros), contando con la previa autorización del beneficiario, en la cual, la cuota por el servicio prestado es descontada automáticamente de la cuenta bancaria o tarjeta de crédito del mismo. En este proceso actúan dos IFP distintas, una quien solicita el pago (IFP beneficiaria u ordenante) y otra quien provee el pago (IFP pagadora).

Para que un pago domiciliado sea procesado a través de la CCE, es necesario que la empresa involucrada esté registrada en el sistema que realiza la compensación, así como la información del cliente debe ser validada por la IFP ordenante a través de la IFP pagadora utilizando la CCE como intermediario. Este registro se hace en forma manual y se inicia cuando una persona solicita a la empresa prestataria del servicio, domiciliar los pagos a su cuenta o tarjeta de la IFP A (banco donde el cliente posee cuenta), y la empresa prestataria del servicio, a su vez informa a su IFP B (banco donde la empresa posee cuenta). La IFP B solicita al departamento de CCE el registro

CAPÍTULO I: EL PROBLEMA

de la empresa y ésta informa a la IFP A que se tramitarán pagos contra la cuenta o tarjeta involucrada (Esto es para que se valide que la información existe y es correcta). La IFP A informa a la CCE la veracidad de los datos. La CCE crea de forma manual, los archivos planos necesarios, que son leídos por el Sistema de Compensación. Una vez realizado esto informa tanto a la IFP A como a la IFP B, que ya están listos para atender dichas operaciones.

Los Créditos Directos son transferencias ordenadas por los clientes, que involucra a dos IFPs diferentes, ejemplo de esto, es la transferencia que ordena una persona natural o jurídica desde una cuenta bancaria del Banco A, a un beneficiario natural o jurídico que tiene cuenta bancaria en un Banco B.

El BCV en su rol como IFP en la CCE, atiende las solicitudes de clientes internos (departamentos del BCV como RRHH, Habilitaduría, otros) y externos como ministerios y otros organismos públicos.

En la actualidad el BCV como IFP participante en la CCE, atiende a sus clientes internos a través de las interfaces de los sistemas existentes y desarrollados en casa, esto permite que los departamentos ordenen pagos, como por ejemplo la nómina de los empleados, entre otros, y obtengan la respuesta del éxito o no de la operación. En cuanto a la atención de los clientes externos no existe un interfaz que permita que éstos ordenen sus pagos de manera automatizada; el procedimiento actual es que el cliente envía un comunicado y un archivo en algún medio de almacenamiento, el cual es procesado por BCV-IFP y se emite vía telefónica la respuesta del éxito o no de la transacción; esta metodología, prácticamente manual conlleva a que los pagos no sean oportunos entre otros muchos otros riesgos.

1.2 Solución Propuesta

Se propone para la solución del problema mencionado en la sección anterior, el desarrollo de un portal Web que permita el registro y administración de empresas ordenantes para el medio de pago Domiciliaciones y por otro lado a modo de Internet Banking, el procesamiento de operaciones de Créditos Directos, ordenados por Organismos Oficiales a través de la Cámara de Compensación Electrónica, compuesto por los siguientes módulos:

1. Módulo de Autenticación Control de Acceso:

- Validación de usuario y contraseña usando el protocolo de acceso LDAP (Lightweight Directory Access Protocol) que utiliza el BCV.
- Validación de estructura de claves permitidas (números, letras, extensión).
- Accesos a los distintos módulos de acuerdo al tipo de usuario que ingresa.
- Cambio de clave.
- Cada usuario manipulará la información exclusivamente cargada por él, en representación de la entidad para la cual labore.

2. Módulo de Domiciliaciones:

- Gestión de Clientes:
 - Cargar Archivo de Solicitudes.
 - Cargar Archivo de Respuesta.
 - Carga Manual (solicitud y respuesta).
 - Consultar Archivos (Filtrar Solicitudes y Generar Archivos).
 - Consultar Afiliación – desafiliación.
 - Consultar Solicitudes Vencidas.

CAPÍTULO I: EL PROBLEMA

- Consultar Solicitudes Rechazadas.
 - Estadísticas.
 - Gestión de Empresas:
 - Registrar Solicitud de Afiliación.
 - Registrar Solicitud de Desafiliación.
 - Consultar Solicitudes.
 - Consultar Afiliaciones.
 - Generar Mensajes de Empresas.
 - Administración del módulo:
 - Administración de las IFP (consultar, agregar, editar, eliminar, inactivar banco, refrescar, exportar Excel, Exportar PDF).
 - Motivo desafiliación (consultar, agregar, editar, eliminar, refrescar, exportar Excel, Exportar PDF).
 - Motivo Rechazo (consultar, agregar, editar, eliminar, refrescar, exportar Excel, Exportar PDF).
 - Sector (consultar, agregar, editar, eliminar, refrescar, exportar Excel, Exportar PDF).
 - Tipo Transacción (consultar, agregar, editar, eliminar, refrescar, exportar Excel, Exportar PDF).
 - Código Resultado (consultar, agregar, editar, eliminar, refrescar, exportar Excel, Exportar PDF).
3. Módulo de Créditos Directos:
- Carga de Archivos:

CAPÍTULO I: EL PROBLEMA

- Validaciones de datos (estructura del nombre del archivo, estructura interna del archivo, dígitos de control código cuenta cliente, RIF, entre otros).
 - Despliegue en pantalla del resultado de la carga y confirmación de acuerdo a la cantidad de registros y monto total.
 - No se permitirá la carga parcial de archivos, es decir, cualquier error implica el rechazo total del mismo.
 - Los archivos podrán ser cargados con una fecha valor diferente al día en que sea realiza la carga. Esta fecha deberá ser indicada por el usuario que lleva a cabo la carga del archivo.
 - Consulta de cantidad de operaciones por archivos de acuerdo al estatus: Pendiente (fecha valor mayor a fecha actual), procesado (enviado a la CCE).
 - Posibilidad de anular el envío de un archivo pendiente (fecha valor mayor a fecha actual).
 - Reportes (selección de múltiples parámetros):
 - Operaciones rechazadas por fallas técnicas (UAP).
 - Operaciones transmitidas.
 - Operaciones devueltas.
 - Operaciones efectivamente compensadas.
 - Descarga de todos los reportes en formato PDF.
4. Módulo de Administración de Usuarios y Reportes:
- Administración de usuarios:
 - Posibilidad de cambiar clave.
 - Posibilidad de bloquear acceso a un usuario particular

- Reportes:
 - Archivos por estatus.
 - Operaciones rechazadas por fallas técnicas (UAP).
 - Operaciones transmitidas.
 - Operaciones devueltas.
 - Operaciones efectivamente compensadas.
 - Descarga de reportes en PDF.

1.3. Objetivos

1.3.1 Objetivo General

Desarrollar un portal Web que sea interfaz y procese el registro de empresas ordenantes de pagos domiciliados, así como el procesamiento de operaciones de créditos directos ordenados por organismos oficiales en el Sistema de Cámara de Compensación Electrónica del Banco Central de Venezuela.

1.3.2 Objetivos Específicos

- Definir los requerimientos del usuario de la funcionalidad de atención de las solicitudes de registro de empresas ordenantes; así como del usuario de la funcionalidad para la atención de clientes externos BCV-IFP y generar el documento de requerimientos.
- Diseñar e implementar el modelo de estructura de datos para que soporte la persistencia de información.
- Diseñar y desarrollar el Front End según el estándar del BCV.
- Diseñar y desarrollar el módulo de autenticación y control de acceso al portal contra el protocolo de seguridad LDAP del BCV, el cual permite el acceso a los diferentes módulos de acuerdo al perfil de usuario que ingresa, así como la manipulación de cuenta por el usuario.

CAPÍTULO I: EL PROBLEMA

- Diseñar y desarrollar la funcionalidad de gestión de clientes para el módulo de domiciliaciones que comprende la carga de archivos de solicitudes y respuestas; así como las consultas de archivos, afiliaciones, desafiliaciones, solicitudes y la sección de estadísticas.
- Diseñar y desarrollar la funcionalidad de gestión de empresas para el módulo de domiciliaciones que comprende el registro de solicitudes tanto de afiliación como de desafiliación, la generación de mensajes de las empresas y las consultas de afiliaciones y desafiliaciones.
- Diseñar y desarrollar la funcionalidad de administración del módulo de domiciliaciones, que comprende operaciones de ingreso, consulta, edición, eliminación para las IFP, motivos de desafiliación, motivos de rechazos, sector, tipo de transacción y código de resultado.
- Diseñar y desarrollar la funcionalidad de reportes que pueden ser exportados tanto en formato Excel como PDF para los distintos componentes de la administración de módulo de domiciliaciones.
- Diseñar y desarrollar la funcionalidad de carga de archivos para el módulo de créditos directos.
- Diseñar y desarrollar la funcionalidad de reportes por selección de múltiples parámetros exportables en formato PDF para el módulo de créditos directos.
- Diseñar y desarrollar el módulo de administración de usuarios disponible solo para el perfil de usuario interno que permite la administración de contraseñas y bloqueo de usuarios tanto internos como externos.
- Diseñar y desarrollar la funcionalidad de reportes para el módulo de administración de usuarios, la cual se muestra al usuario interno los detalles de los archivos cargados por los diferentes usuarios externos. Estos reportes se realizan por múltiples parámetros y pueden ser exportados en formato PDF.

1.4 Alcance y Limitaciones

La solución propuesta se enfoca en el desarrollo del portal Web para el registro de empresas ordenantes que participan en la domiciliación de pagos para el BCV como ente compensador; abarcando el cumplimiento del objetivo general en consecución de la ejecución satisfactoria de los objetivos específicos, que se encuentran enmarcados por la definición del requerimiento, diseño, presentación de solución y desarrollo.

Esta solución ofrece atención oportuna del BCV como ente compensador en unos de sus medios de pago como lo es la domiciliación.

Los módulos se desarrollaran considerando el diseño de interfaces, funciones y/o procedimientos bajo los requerimientos del Banco Central de Venezuela.

A continuación se detallan las funcionalidades para llevar a cabo el desarrollo de la aplicación:

- Diseño de la estructura de base de datos (tablas, campos, procesos etc.), presentación al área de arquitectura de datos de la institución y realizar las adecuaciones necesarias en caso de aplicar.
- Módulo de Autenticación y Control de Acceso: Permitirá al usuario tanto externo como interno autenticarse mediante nombre de usuario y contraseña contra el LDAP de la institución, garantizando el acceso legítimo al portal. Para todos los usuarios se permite el cambio de clave.
- Módulo de Domiciliaciones: Comprende las siguientes funcionalidades.
 - *Gestión de Empresas*: esta funcionalidad permitirá la generación de los archivos con el formato del mensaje respectivo que permite la comunicación con el SCS, ya sea para afiliación, desafiliación de empresas y actualización de empresas. Una vez procesada esta solicitud se emitirá respuesta de registro satisfactorio o no de la solicitud a la IFP que realizó la carga, así como también incluirá una Consulta General que permitirá filtrar por estatus (Activas, Inactivas,

CAPÍTULO I: EL PROBLEMA

- Suspendida, etc.); a partir de esta consulta se podrá editar, excluir y suspender una empresa. Consultar Solicitudes con las funcionalidades de filtro por período, de actualización de empresas.
- *Gestión de Clientes*: esta funcionalidad permitirá la generación de los archivos con el formato del mensaje respectivo que permite la comunicación con el SCS, para que una IFP presente a otra IFP la afiliación o desafiliación de un cliente y esta a su vez emita la respuesta de la validez o no de la información presentada; incluirá la carga de solicitudes, carga de respuesta, consulta de solicitudes por estatus, consulta de afiliaciones y estadísticas.
 - *Administración del módulo*: esta funcionalidad permitirá el usuario administrador del Módulo de Domiciliaciones agregar, editar, eliminar o inactivar una IFP, así como cargar los motivos para desafiliación (consultar, agregar, editar, eliminar), cargar el Sector (consultar, agregar, editar, eliminar), cargar el Tipo de Transacción (consultar, agregar, editar, eliminar) y cargar los Códigos de Resultados (consultar, agregar, editar, eliminar, refrescar). Todos los reportes del módulo de domiciliaciones deberán ser exportados a formato Excel y PDF.
 - **Módulo de Créditos Directos**: Comprende las siguientes funcionalidades.
 - *Carga de Archivos*: La carga de archivos permitirá ordenar de manera masiva las operaciones de créditos directos por cada cliente interno que accede al portal. Se valida el nombre del archivo, la estructura interna, la obligatoriedad de los campos, el formato de los campos, el dígito de control de la cuenta cliente, número de cédula o RIF y otros que se identifiquen durante el levantamiento de información.
 - *Reportes*: Muestran el detalle del resultado de la carga de archivos hecha por la institución que accede; los archivos pendientes por

CAPÍTULO I: EL PROBLEMA

procesar cuando la fecha valor sea mayor a la actual; los archivos y las operaciones válidas cargadas. Una vez llegada la fecha valor, muestra los archivos que se transmitieron exitosamente a la Cámara de Compensación y las operaciones que fueron devueltas tanto porque no pasaron las validaciones del Sistema de la CCE como aquellas devueltas por la IFP beneficiaria por alguna razón en específico. Todos los reportes se pueden guardar en formato PDF.

- **Módulo de Administración de Usuarios y Reportes:** Este módulo permitirá al usuario interno del portal con el perfil respectivo, la posibilidad de administrar las contraseñas tanto de usuarios internos como externos; el bloqueo de acceso de un usuario en particular. Los reportes de este módulo muestran el detalle de todas los archivos cargados por los diferentes clientes externos, permitiendo elegir diferentes parámetros de consulta y diferenciados por el cliente que hizo la carga, el estatus del archivo, la o las operaciones devueltas tanto por el sistema de la CCE como las devueltas por las IFPs, así como las operaciones compensadas. Todos los reportes se pueden guardar en formato PDF.
- **La aplicación contará con cuatro perfiles de usuarios:** Un perfil para usuario externo domiciliaciones, el cual le permitirá acceder al portal, al cambio de clave y al módulo de domiciliaciones en las funcionalidades de Gestión de Empresas y Gestión de Clientes. Un perfil para usuario externo de créditos directos, el cual le permitirá acceder al portal, al cambio de clave y al módulo de carga de archivos y reportes. Un perfil para usuario interno, el cual le permitirá acceder al portal, al cambio de clave, a la funcionalidad de administración del módulo de domiciliaciones, y consultar los reportes para control y seguimiento de todos los archivos cargados por las IFP tanto para domiciliaciones como para créditos directos. Un perfil para usuario interno, cual le permitirá acceder al portar, al cambio de clave, consultar los reportes

para control y seguimiento de todos los archivos cargados por las IFP y la administración de usuarios externos.

Entre las posibles limitaciones durante el desarrollo de la solución se encuentran las siguientes:

- Se usarán las tecnologías como: Java 1.6, Eclipse¹ para entorno de desarrollo integrado (IDE), Subversion para el control de versiones, Quartz como frameworks para los Jobs o Tareas, Oracle como repositorio de datos, Ibatis para el manejo de base de datos, ZK para el modelo de vista controlador, Spring como base del proyecto para inyección de dependencias y Log4j para el manejo de Logs, otros.

1.5 Justificación

Mediante el sistema actual para el registro de empresas en la Cámara de Compensación Electrónica, el registro de una nueva empresa tarda entre 15 y 20 días hábiles, lo cual no satisface oportunamente la creciente demanda del servicio de domiciliación de pagos. De igual manera, el procedimiento utilizado para el pago de créditos directos a través de la CCE se lleva a cabo de una manera prácticamente manual, mediante el envío de información utilizando medios físicos, lo que genera riesgos tales como la pérdida de la información o que los pagos no se efectúen oportunamente.

Motivado a lo expuesto anteriormente, el Banco Central de Venezuela, se ve en la necesidad de proveer una interfaz (Portal Web) que se adecue a las necesidades del dinámico Sistema Financiero Venezolano; en su papel como ente compensador, debe agilizar el registro de empresas involucradas en el medio de pago Domiciliaciones y en su papel de Institución Financiera Participante, debe permitir a sus clientes externos como los ministerios, que hacen pago de la nómina pública y otros de gran importancia, cumplir con su obligación de una manera oportuna.

¹Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar Aplicaciones de Cliente Enriquecido

CAPÍTULO II: MARCO REFERENCIAL

2. MARCO REFERENCIAL

En el presente capítulo se presenta una explicación detallada de los diferentes términos y nociones utilizadas en la realización del Trabajo Instrumental de Grado. De igual manera, se realiza la descripción de los entes y procesos que intervienen en el funcionamiento del Portal Web.

2.1 Cámara de Compensación Electrónica (CCE)

Es una asociación voluntaria de carácter oficial integrada por instituciones financieras de una determinada plaza, que tiene como finalidad saldar las obligaciones recíprocas entre los entes compensadores.

En las cámaras de compensación se realizan las operaciones pertinentes de cargos y abonos a los bancos que resultaren deudores y acreedores, minimizándose los traslados de efectivo, debido a que las liquidaciones se realizan a través de transferencias entre cuentas.

La Cámara de Compensación Electrónica en Venezuela es un mecanismo creado por el Banco Central de Venezuela con la finalidad de facilitar las transacciones de pago y las operaciones interbancarias.

El sistema cuenta con cobertura nacional, y a través del mismo las instituciones financieras participantes compensan transacciones efectuadas con cheques y transferencias interbancarias; formando parte de éstas: pagos de nómina, pagos de proveedores, pagos de tarjetas de crédito y transferencias. (BCV, 2008)

2.2 Sema Cleary System (SCS)

El SCS en su versión venezolana está configurado como un sistema de información descentralizado en el que cada Institución Financiera Participante (IFP) cuenta en sus propias instalaciones con un módulo de acceso denominado UAP (User Access

CAPÍTULO II: MARCO REFERENCIAL

Platform) que le permite interactuar con el sistema central de compensación (SCS) desde sus sistemas de información internos (SIP).

De este modo, todos los intercambios tanto financieros como no financieros (flujos de archivos de transacciones monetarias, flujos de imágenes, flujos informativos, etc.) son canalizados a través de la UAP desde los sistemas de información internos de la IFP al núcleo central del SCS en la Cámara de Compensación Electrónica (CCE) del Banco Central de Venezuela.(BCV, 2006)

2.3 Plataforma de Acceso al Usuario (UAP)

La Plataforma de Acceso al Usuario (UAP) actúa como una interfaz técnica que permite a las IFPs conectarse al núcleo central de la CCE. La UAP está integrada con el sistema de información propio de la IFP a través de una interfaz que permite el intercambio de información entre el SIP y la UAP.

En cualquier caso, la UAP, instalada y configurada según las características de cada IFP, libera al sistema de información de la IFP (SIP) de la necesidad de tener que habilitar una conexión directa con el núcleo central de la CCE, tal conexión siempre se realiza por medio de la UAP. (BCV, 2006)

2.4 Domiciliaciones

La domiciliación de pago es un instrumento mediante el cual cualquier cliente puede autorizar la cancelación de facturas por servicios con cargo a una cuenta corriente o de ahorro, independientemente del banco en el cual el cliente o empresa de servicio posea su cuenta.(BCV, 2008)

2.5 Créditos Directos

Los Créditos Directos son transferencias de fondos interbancarios. Sin importar en que banco tengan sus cuentas los clientes, les permite realizar transferencias a otra persona en un banco diferente, quien dispondrá de esos fondos, a más tardar el día hábil siguiente de la institución. (BCV, 2008)

CAPÍTULO II: MARCO REFERENCIAL

2.6 Java

Java es un lenguaje de programación orientado a objetos, de propósito general, concurrente y basado en clases, que permite la ejecución de un programa en múltiples sistemas operativos y que incluye por defecto soporte para trabajo en red. La programación en Java permite ejecutar código en sistemas remotos de forma segura y el desarrollo de aplicaciones web dinámicas de servidor. (Gosling, Joy, Steele, Bracha, & Buckley)

2.7 Lightweight Directory Access Protocol (LDAP)

El protocolo LDAP (en español Protocolo Ligero de Acceso a Directorios) es un protocolo de comunicaciones diseñado para su uso en redes TCP/IP². LDAP define cómo puede tener acceso un cliente de directorio a un servidor de directorios y cómo el cliente puede realizar operaciones de directorio y compartir datos del directorio. Los grupos de trabajo correspondientes del Grupo de trabajo de ingeniería de Internet (IETF) establecen los estándares de LDAP.

Como implica su nombre, LDAP es un método eficaz que permite tener acceso a los servicios de directorio sin las complejidades asociadas a otros protocolos de servicios de directorio. Dado que LDAP define qué operaciones pueden realizarse para consultar y modificar información en un directorio y cómo se puede tener un acceso seguro a esa información, es posible utilizarlo para buscar o enumerar objetos de directorio. (Microsoft Corporation)

2.8 ZK Framework

ZK es un framework³ de código abierto, orientado a eventos y basado en componentes, que permite la creación de interfaces de usuario enriquecidas para

² TCP/IP: es un conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre computadoras.

³ Framework: es un marco de aplicación o conjunto de bibliotecas orientadas a la reutilización de componentes de software que sirve de base para el desarrollo de software.

CAPÍTULO II: MARCO REFERENCIAL

aplicaciones web. Permite el desarrollo de aplicaciones web basadas en AJAX⁴ sin la necesidad de utilizar otro tipo de lenguajes.

De igual manera, ZK permite desarrollar aplicaciones web en un ambiente similar a las aplicaciones de escritorio, ya que todos los componentes son creados del lado del servidor. En otras palabras, la relación entre los componentes en el lado del cliente y los componentes en el lado del servidor es simétrica. Cuando se modifican componentes en el lado del servidor, los componentes del lado del cliente son actualizados automáticamente en el explorador por el motor ZK.

Por lo tanto, los usuarios finales obtienen una interacción y respuesta similar a las de una aplicación de escritorio, mientras que la complejidad del desarrollo es similar a la que tendría la codificación de aplicaciones de escritorio. (Chen & Robbie, 2007)

2.9 Apache Tomcat

Apache Tomcat es un servidor Web⁵ de código libre desarrollado por la Fundación de Software Apache (ASF). Este servidor brinda un ambiente de servidor web que permite correr aplicaciones desarrolladas en lenguaje Java. (Apache Software Foundation)

2.10 XML

Son las siglas en inglés de Extensible Markup Language (Lenguaje de Etiquetado Extensible). XML es un lenguaje simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Su función principal es describir datos y su formato permite la lectura de datos a través de diferentes aplicaciones. (World Wide Web Consortium (W3C))

⁴AJAX: Es una técnica de desarrollo Web que se utiliza para crear aplicaciones Web interactivas con la combinación de distintas tecnologías.

⁵Servidor Web: Es un programa que procesa una aplicación del lado del servidor realizando conexiones con el cliente, generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente

CAPÍTULO II: MARCO REFERENCIAL

2.11 ZUL

ZUL es un lenguaje de descripción de interfaces de usuario que permite generar interfaces independientemente del lenguaje utilizado para el desarrollo de la aplicación, además de simplificar el proceso de diseñar aplicaciones web. También se puede describir como un subconjunto de XML para describir interfaces gráficas de usuario de una página web. Las etiquetas de documentos ZUL corresponden a controles gráficos en los formularios y están anidados de la misma forma en que los controles están insertados en la página. (Chen & Robbie, 2007)

2.12 Lenguaje SQL

SQL son las siglas en inglés de Structured Query Language (Lenguaje de Consulta Estructurado). Es un lenguaje de programación estándar para la generación, manipulación y la consulta de información contenida en una base de datos relacional. Maneja reglas de álgebra y cálculo relacional con el fin de realizar la manipulación de la información contenida en la base de datos por parte de los usuarios. (Beaulieu, 2005)

2.13 Mapeo Objeto-Relacional

El mapeo Objeto-Relacional es una técnica de programación utilizada para convertir datos entre lenguajes de programación orientados a objetos y bases de datos relacionales.

Desde el punto de vista del programador el sistema luce como un almacén de datos persistentes, se pueden crear objetos y trabajar con ellos de manera normal en un sistema orientado a objetos y todos los cambios que ocurran en el objeto van a ser reflejados en la base de datos. Una base de datos virtual es creada entre el sistema orientado a objetos y la base de datos relacional. (Lerman & Miller, 2012)

CAPÍTULO II: MARCO REFERENCIAL

2.14 iBatis

Es un framework de mapeo Objeto-Relacional de datos que facilita el uso de las bases de datos relacionales con aplicaciones orientadas objetos. Asocia objetos de modelo (JavaBeans) con sentencias SQL o procedimientos almacenados mediante ficheros descriptores XML, simplificando la utilización de bases de datos. (Apache Software Foundation).

2.15 Inyección de Dependencias

La inyección de dependencias es un patrón de diseño orientado a objetos, en el que se suministran objetos a una clase en lugar de ser la propia clase quien cree el objeto. Consiste en resolver las dependencias de cada clase (atributos) generando los objetos cuando se arranca la aplicación y luego inyectarlos en los demás objetos que los necesiten a través de métodos set o bien a través del constructor, pero estos objetos se instancian una vez, se guardan en una factoria y se comparten por todos los usuarios.

El objetivo es lograr un bajo acoplamiento entre los objetos de nuestra aplicación. Con este patrón de diseño, los objetos no crean o buscan sus dependencias (objetos con los cuales colabora) sino que éstas son dadas al objeto. El contenedor (la entidad que coordina cada objeto en el sistema) es el encargado de realizar este trabajo al momento de instanciar el objeto.

De esta manera, los objetos conocen sus dependencias por su interfaz. Así la dependencia puede ser intercambiada por distintas implementaciones a través del contenedor. Es una programación orientada a interfaces donde se inyectan las implementaciones a través del contenedor.(Prasanna, 2009)

2.16 Spring Framework

Spring Framework es una plataforma que proporciona una infraestructura que actúa de soporte para desarrollar aplicaciones Java. Promueve el bajo acoplamiento a partir de la inyección de dependencias flexible entre los objetos con configuración basada en anotaciones a través de archivos XML.

CAPÍTULO II: MARCO REFERENCIAL

Provee soporte para la programación orientada a aspectos, lo que permite el desarrollo cohesivo al separar la lógica de negocio de la aplicación de los servicios del sistema. Los objetos de la aplicación se encargan de realizar la lógica de negocios y no son responsables de otros aspectos del sistema tales como el registro o el soporte de transacciones.

Spring se encarga de contener y gestionar el ciclo de vida y la configuración de los objetos de la aplicación. Permite la configuración de la manera en que los Beans⁶ deben ser creados y como deben ser asociados entre ellos.

Brinda la posibilidad de configurar y formar aplicaciones complejas a partir de componentes más simples. En Spring, los objetos de aplicación son formados declarativamente, generalmente a través de un archivo XML. Spring también ofrece mucha de la funcionalidad de infraestructura (manejo de transacciones, integración con el framework de persistencia, etc.) permitiendo que el desarrollador se concentre más en el desarrollo de la lógica de la aplicación.

Estas características brindan la posibilidad de escribir un código más limpio, más manejable y con más facilidad para realizar pruebas. De igual manera, presenta un diseño modular, por lo que permite el uso de partes individuales del framework, así como la adaptación de distintos proyectos de la familia Spring, entre ellos Spring Security. Ofrece soporte para otros frameworks de uso común tales como Quartz, ZK e iBatis. (Walls & Ryan, 2005)

2.17 Spring Security

Spring Security es un subproyecto del framework Spring, que permite gestionar completamente la seguridad de las aplicaciones Java, y cuyas ventajas principales son las siguientes:

⁶ Los Beans son modelos de componentes que se usan para encapsular varios objetos en un único objeto (la vaina o Bean en inglés), para hacer uso de un solo objeto en lugar de varios más simples.

CAPÍTULO II: MARCO REFERENCIAL

Es capaz de gestionar seguridad en varios niveles: URLs⁷ que se solicitan al servidor, acceso a métodos y clases Java, y acceso a instancias concretas de las clases.

Permite separar la lógica de la aplicación del control de la seguridad, utilizando filtros para las peticiones al servidor de aplicaciones o aspectos para la seguridad en clases y métodos.

La configuración de la seguridad es portable de un servidor a otro, ya que se encuentra dentro del WAR o el EAR de las aplicaciones.

Soporta muchos modelos de identificación de los usuarios (HTTP BASIC, HTTP Digest, basada en formulario, LDAP, OpenID, JAAS y muchos más). Además permite ampliar estos mecanismos implementando clases propias que extiendan el modelo de Spring Security.(Spring Source)

2.18 Quartz

Quartz es un servicio de código abierto de planificación de tareas que puede ser integrado o utilizado junto con cualquier aplicación Java. Quartz puede ser utilizado para crear simples o complejas programaciones de muchas o pocas tareas; tareas que pueden ser definidas como componentes estándar de Java que puede ser ejecutadas virtualmente en cualquier momento en que sean programadas para ser realizadas. (Quartz Scheduler)

2.19 Control de Versiones

Control de versiones es la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra dicho producto en un momento dado de su desarrollo o modificación. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así

⁷ Un localizador de recursos uniforme, más comúnmente denominado URL (siglas en inglés de Uniform Resource Locator), es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación

CAPÍTULO II: MARCO REFERENCIAL

como las posibles especializaciones realizadas. (Collins-Sussman & Fitzpatrick, 2004)

2.20 Log4J

Es una librería adicional de Java que permite mostrar mensajes de información sobre lo que está sucediendo en la aplicación durante el tiempo de ejecución, lo que habitualmente se conoce como Log. Cuenta con un archivo de configuración donde es posible determinar el tipo de mensajes a mostrar y el tipo de salida que estos mensajes tendrán, bien sea por consola o añadiéndolos a un archivo que puede ser consultado posteriormente. (Apache Software Foundation)

2.21 JasperReports

JasperReports es una herramienta de código abierto que se compone de un conjunto de librerías Java para facilitar la generación de informes en aplicaciones tanto web como de escritorio.

Los informes se definen en un archivo XML el cual es compilado por las librerías JasperReport y genera un fichero .jasper que es utilizado para rellenar y mostrar el informe final. La salida de los informes pueden ser en formato PDF, Excel y algunos otros. La definición de los informes se puede realizar directamente sobre los XML descriptores o a través de herramientas gráficas. (JasperForge)

CAPÍTULO III: MARCO METODOLÓGICO

3. MARCO METODOLÓGICO

Para el desarrollo del Trabajo Instrumental de Grado fue implementada la metodología de Programación Extrema o eXtreme Programming (XP).

Esta metodología ágil fue seleccionada debido a que presenta características que se ajustan a las necesidades del proyecto realizado. Destacan principalmente la adaptabilidad a los cambios de requerimientos realizados por los usuarios, la simplicidad en el código, la reutilización del mismo y la retroalimentación con los usuarios para adecuar el producto en cualquier fase del desarrollo del proyecto.

A continuación se presenta la definición y características de la metodología utilizada. (Beck, 2000)

3.1 Metodología de Programación Extrema

La Programación Extrema o eXtreme Programming (XP) es la más destacada de las metodologías ágiles y fue formulada por el ingeniero de software estadounidense Kent Beck. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre.

Esta metodología se basa principalmente en la retroalimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y la audacia o valentía para enfrentar los cambios.

A continuación se presentan las características esenciales de la metodología XP organizadas en las secciones: historias de usuario, prácticas y proceso.

3.1.1 Historias de Usuario

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de descripciones breves realizadas por el cliente sobre las características que el sistema debe poseer, sean requisitos funcionales o no

CAPÍTULO III: MARCO METODOLÓGICO

funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento estas historias pueden ser eliminadas, remplazadas por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para poder ser implementada en unas semanas.

3.1.2 Prácticas

La mayoría de las prácticas propuestas por XP no son novedosas sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor en la práctica. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo.

La metodología XP se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Se considera que adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Esto se consigue gracias al uso de las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas que describiremos a continuación.

Planificación

La planificación se basa en mantener una comunicación frecuente entre los clientes y el equipo de programadores. El cliente establece la prioridad de cada historia de usuario, de acuerdo con el valor que aporta para el negocio. Los programadores realizan una estimación del tiempo y el esfuerzo necesario para el desarrollo de cada historia. Se ordenan las historias de usuario según prioridad y esfuerzo, y se define el contenido de la entrega o iteración. Esta actividad se realiza durante la planificación de la entrega, en la planificación de cada iteración y cuando sea necesario realizar algún cambio en el proyecto.

CAPÍTULO III: MARCO METODOLÓGICO

Entregas pequeñas

La idea es producir rápidamente versiones del sistema que sean operativas, aunque obviamente no cuenten con toda la funcionalidad pretendida para el sistema pero si que constituyan un resultado de valor para el negocio. Una entrega no debería tardar más de 3 meses.

Diseño simple

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. El creador de la metodología XP señala que el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos.

Pruebas

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

Refactorización (Refactoring)

La refactorización es una actividad constante de reestructuración del código realizada con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. La refactorización mejora la estructura interna del código sin alterar su comportamiento externo. Para mantener un diseño apropiado, es necesario realizar actividades de cuidado continuo durante el ciclo de vida del proyecto.

De hecho, este cuidado continuo sobre el diseño es incluso más importante que el diseño inicial. Un concepto pobre al inicio puede ser corregido con esta actividad continua, pero sin ella, un buen diseño inicial se degradará.

CAPÍTULO III: MARCO METODOLÓGICO

Integración continua

Cada vez que una pieza del código se encuentra lista es integrada al sistema. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente. El desarrollo de un proceso disciplinado y automatizado es esencial para un proyecto controlado, el equipo de desarrollo está más preparado para modificar el código cuando sea necesario, debido a la confianza en la identificación y corrección de los errores de integración.

40 horas por semana

Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo. Los proyectos que requieren trabajo extra para intentar cumplir con los plazos suelen ser entregados con retraso. En lugar de esto se puede realizar la actividad de planificación para cambiar el ámbito del proyecto o la fecha de entrega.

Disponibilidad del cliente

El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito de la metodología XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. En caso de que el cliente no cuente con la disponibilidad necesaria existen algunas recomendaciones: intentar conseguir un representante que pueda estar siempre disponible y que actúe como interlocutor del cliente, contar con el cliente al menos en las reuniones de planificación, establecer visitas frecuentes de los programadores al cliente para validar el sistema, anticiparse a los problemas asociados estableciendo llamadas telefónicas frecuentes y conferencias, reforzando el compromiso de trabajo en equipo.

CAPÍTULO III: MARCO METODOLÓGICO

Estándares de programación

Se enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios.

3.1.3 Proceso

Un proyecto elaborado con la metodología XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. A grandes rasgos, el ciclo de desarrollo consiste en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Se repite el primer paso.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

Fase I: Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo

CAPÍTULO III: MARCO METODOLÓGICO

se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Fase II: Planificación de la Entrega

En esta fase el cliente establece la prioridad de cada historia de usuario, y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Esta fase dura unos pocos días.

Fase III: Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En cada iteración se realiza un ciclo completo de análisis y planificación, diseño, desarrollo y pruebas. En la primera iteración se establece una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Al final de la última iteración el sistema estará listo para entrar en producción.

Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior.



Figura 1 Esquema gráfica de una iteración de la metodología

Fuente: Elaboración propia

CAPÍTULO III: MARCO METODOLÓGICO

Fase IV: Producción

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación.

Fase V: Muerte del Proyecto

Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

4. DESARROLLO

En el presente capítulo se presenta la descripción de los distintos pasos seguidos durante el desarrollo del Trabajo Instrumental de Grado.

Como se indica en el capítulo anterior, la metodología aplicada para el desarrollo fue la metodología XP. Es necesario destacar que las iteraciones de la metodología comienzan a partir de la implementación de las estructuras de datos y cada uno de los módulos del portal corresponde a una iteración distinta, por lo tanto se realizaron un total de 6 iteraciones, cada una de las cuales cuenta con las distintas fases de la metodología.

Es necesario destacar que durante el desarrollo del Trabajo Instrumental de Grado no se abarcaron todas las fases de la metodología. Se realizan las iteraciones necesarias para cumplir los objetivos y las 3 primeras fases generales, debido a que el sistema no será puesto en producción aún por el Banco Central de Venezuela no se llevan a cabo las fases de producción y fin del proyecto.

A continuación se presentan las distintas fases de la metodología aplicadas en el desarrollo del portal.

4.1 Fase de Exploración

Esta fase tuvo una duración de aproximadamente tres semanas. Durante la misma se llevaron a cabo reuniones con el equipo de trabajo para la familiarización con el entorno y la definición de los requerimientos generales del portal. Posteriormente se realizaron reuniones con el área usuaria y los miembros del equipo del proyecto para efectuar el levantamiento de información sobre los procesos que intervienen al atender las solicitudes de registro de empresas ordenantes en la domiciliación de pagos y las operaciones de crédito directo.

A partir de estas reuniones se recopilaron los requerimientos generales del portal y las historias de usuario a desarrollar (ver Apéndice 1).

CAPÍTULO IV: DESARROLLO

Igualmente, debido a que el Portal interactúa con la Cámara de Compensación Electrónica fue necesario realizar el levantamiento de información para conocer a fondo el funcionamiento de la misma, así como del Sema Cleary System (SCS) y la manera automatizada con la que este maneja el ingreso y salida de información desde las IFP. El levantamiento de información se realizó a partir del Dossier informativo entregado por la institución.

Este funcionamiento se encuentra reflejado gráficamente en la siguiente figura

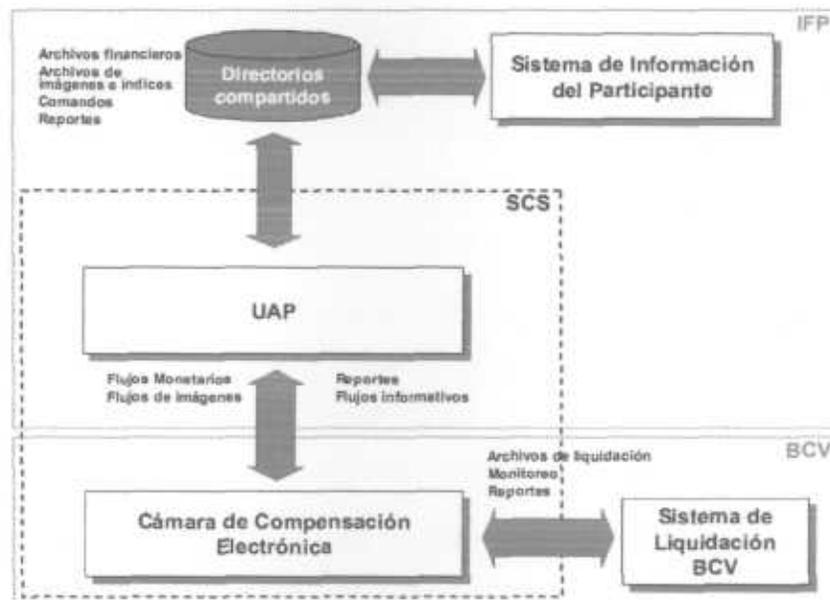


Figura 2 Esquema del funcionamiento de la CCE y sus componentes

Fuente: BCV, 2006

Las aplicaciones implementadas en el Sistema de Información de la Institución Financiera Participante envían y reciben información de la UAP a través de los directorios compartidos. La UAP es la encargada de realizar el envío de las distintas solicitudes y trámites a la Cámara de Compensación Electrónica que procesa las operaciones y envía las respuestas a través de la UAP.(BCV, 2006)

El Banco Central de Venezuela tiene como estándar en todas sus aplicaciones la implementación del protocolo de acceso LDAP para las validaciones de usuarios y las

CAPÍTULO IV: DESARROLLO

permisologías correspondientes, por lo cual durante la fase de exploración también se realizó la familiarización con el protocolo y las librerías suministradas por el banco para la implementación del mismo en el portal.

4.2 Fase de Planificación de Entregas

Basado en las historias de usuario recopiladas durante las reuniones realizadas en la fase de exploración, así como la información obtenida a partir del dossier informativo de la CCE, se realizó en conjunto con el área usuaria la identificación de la prioridad de cada uno de los módulos que componen el portal.

A partir de lo expresado por los usuarios, se creó un plan de proyecto que refleja la prioridad de cada módulo y el orden de desarrollo de cada uno de los componentes del portal. Cada componente identificado dentro del cronograma corresponde a una iteración en la metodología de desarrollo. Se acordó con el cliente que al finalizar el desarrollo de cada iteración se realizan pruebas y una revisión del componente desarrollado para, dependiendo del caso, realizar las adecuaciones necesarias o dar por completada la iteración y pasar a la siguiente.

Cabe destacar que dentro del plan de proyecto que se estableció un tiempo estimado del desarrollo de cada iteración, así como el orden de desarrollo y entrega de cada uno de los componentes del portal. Esto debido a que en cada una de las iteraciones se recopilan nuevas historias de usuario relativas al componente correspondiente.

Dentro de este plan de proyecto y de acuerdo con las prioridades expresadas por los usuarios se define el siguiente orden en las iteraciones que quede ser apreciado detalladamente en el Apéndice 2:

- Estructuras de datos.
- Front End.
- Módulo de Domiciliaciones.
- Módulo de Créditos Directos.
- Módulo de Autenticación y Control de Acceso al Portal.
- Módulo de Administración de Usuarios.

A continuación se especifica cada una de las iteraciones realizadas divididas por etapas.

4.3. Iteraciones

4.3.1 Primera Iteración: Estructuras de Datos

4.3.1.1 Análisis y Planificación

En esta primera iteración se realizaron una serie de reuniones con los usuarios para determinar las necesidades y requerimientos básicos del Portal en cuanto a estructuras de datos, tanto para el módulo de Domiciliaciones como para Créditos Directos.

Durante esta fase se pudieron recopilar las historias de usuario:

- HU-1-01
- HU-1-02

El detalle correspondiente a estas historias de usuario se encuentra reflejado en el Apéndice 3.

4.3.1.2 Diseño

Luego de analizar los requerimientos de datos suministrados por el área usuaria en las historias de usuario, se procedió a realizar el diseño de las estructuras de datos necesarias para la implementación de los requerimientos.

Es necesario recordar que previo al desarrollo del portal, la Cámara de Compensación Electrónica ya contaba con un sistema para realizar el registro de empresas ordenantes y para permitir el ordenamiento de pagos a través de créditos directos. Debido a esto fue necesario realizar una revisión a las estructuras de datos ya creadas en el sistema de información de la empresa, para así acoplar las nuevas estructuras de datos a las ya existentes. Una vez que se tuvo claro cuáles de las estructuras existentes podrían relacionarse con las nuevas, se procedió a realizar el diseño de las nuevas estructuras de acuerdo a los requerimientos del usuario.

Para el caso de créditos directos se diseñaron 4 nuevas tablas en la base de datos, que pueden ser observadas en el siguiente modelo entidad relación.

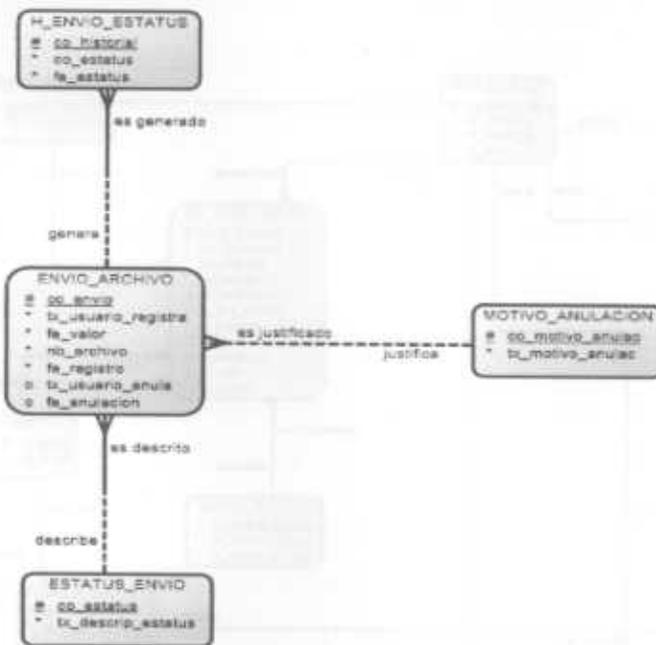


Figura 3 Modelo Entidad Relación para Módulo de Créditos Directos

Fuente: Elaboración propia

En cuanto al módulo de domiciliaciones fueron 14 las nuevas tablas necesarias para almacenar toda la información que el cliente solicitó, esta estructura se encuentra reflejada en el modelo entidad relación presentado a continuación.

CAPÍTULO IV: DESARROLLO

En el paquete UI se encuentran ubicados los diferentes controladores para los eventos ocurridos en las vistas a ser creadas mediante lenguaje ZUL, estas vistas no se encuentran representadas en el diagrama, pero se relacionan solo con el paquete de controladores.

El paquete Service se encarga de manejar la lógica de la aplicación y provee a los controladores de los métodos requeridos a partir de los eventos registrados en las vistas.

El paquete DAO contiene los diferentes métodos que interactúan con los mapas de iBatis para enviar o recibir información de la base de datos.

El paquete Bean, como su nombre lo indica contiene los Beans que encapsulan los objetos obtenidos en la base datos, con todos sus métodos Set y Get.

Cada una de las clases contenidas en los DAO y Service cuenta con interfaces mediante la cual se realiza la comunicación con el resto de clases de la aplicación.

4.3.1.3 Implementación

Para realizar la creación o modificación de cualquier estructura de datos en la empresa es necesaria la aprobación de las mismas por parte del área de arquitectura de datos. Por lo tanto, para la implementación de las estructuras de datos diseñadas previamente, fue necesario completar los formatos correspondientes y enviarlos al área de arquitectura de datos para su evaluación. Estos formatos cumplen un estándar de la empresa y contienen la descripción detallada de las estructuras de datos a crear (tablas, campos, procesos etc.). Un ejemplo del formato utilizado puede ser apreciado en el apéndice 5.

Una vez evaluados por el área de estructura de datos y corregidos los detalles necesarios, se procedió a la creación de las nuevas estructuras en la base de datos, así

CAPÍTULO IV: DESARROLLO

como de los constraints⁸ y las secuencias⁹ necesarias; esta tarea fue realizada por el departamento de bases de datos de la institución a partir de los formatos entregados.

En esta fase también se realizó la configuración necesaria para los diferentes Frameworks a utilizar durante el desarrollo del proyecto. Se importaron las librerías correspondientes a iBatis, Spring y ZK; realizando todas las configuraciones necesarias para integrarlos exitosamente, el proceso fue llevado a cabo sin inconvenientes. Fue necesario crear un archivo donde se configuran las inyecciones de dependencias de Spring, este archivo es un archivo XML llamado applicationContext y es modificado a lo largo del proyecto para llevar a cabo la inyección de dependencias, en él se determinan los Beans que contiene el archivo y la inyección de las dependencias de ese Bean a realizarse al momento de su creación.

Igualmente se realizó la importación de la librería log4J y en el archivo de configuración del mismo se configuraron los mensajes a mostrar durante tiempo de ejecución, en este caso se configuró para que las excepciones y mensajes de error fueran mostrados por consola.

4.3.1.4 Pruebas

Una vez implementadas las nuevas estructuras en la base de datos, fueron realizadas una serie de inserciones, actualizaciones y eliminaciones mediante sentencias SQL a través de la herramienta Toad¹⁰ para comprobar el correcto desarrollo de las historias de usuario. Estas pruebas se encuentran reflejadas en el Apéndice 6.

Al cumplirse satisfactoriamente todas las pruebas necesarias, se procedió a dar por culminada esta iteración.

⁸Constraints: Son restricciones o condiciones que obligan el cumplimiento de ciertas condiciones en la base de datos.

⁹Secuencias: Son tablas con un campo numérico en el cual se almacena un valor y cada vez que se consulta, se incrementa tal valor para la próxima consulta. Se emplea para generar valores enteros secuenciales únicos y asignárselos a campos numéricos.

¹⁰Toad: Es una aplicación informática de desarrollo SQL y administración de base de datos.

CAPÍTULO IV: DESARROLLO

4.3.2 Segunda Iteración: Front End

4.3.2.1 Análisis y Planificación

Durante esta fase se realizaron reuniones con el equipo de trabajo para determinar los requerimientos básicos en cuanto a interfaz gráfica para el portal. Las historias de usuario surgidas durante esta fase fueron:

- HU-2-01
- HU-2-02

El portal debía cumplir con las características de una RIA¹¹ (Rich Internet Application), utilizando el Framework ZK por requerimientos de la institución.

4.3.2.2 Diseño

A partir de la información suministrada en el manual de estilo del BCV, así como en los requerimientos recopilados, se procedió a realizar el diseño de la estructura principal del Front End. Este diseño se realizó basado en la estructura que presentan los portales desarrollados para la institución, cumpliendo los estándares establecidos por la misma. Es decir, en la parte izquierda de la pantalla se encontraría un menú desplegable con las distintas opciones a las cuales podrá acceder el usuario y en el centro de la pantalla se cargaría el contenido relacionado con el vínculo elegido, sin necesidad de recargar la página completa cada vez que se accede a un vínculo.

4.3.2.3 Implementación

Para la implementación de la interfaz gráfica, primero se creó el archivo ZUL correspondientes a la vista de la página principal del portal, que contiene una breve descripción del portal y links de interés para los usuarios.

Posteriormente se generó el archivo correspondiente a la vista Index que será mostrado una vez el usuario sea autenticado. Esta vista comprende el menú desplegable donde se muestran las opciones de Domiciliación, Créditos Directos,

¹¹RIA: Las RIA surgen como una combinación de las ventajas que ofrecen las aplicaciones web y las aplicaciones tradicionales. Buscan mejorar la experiencia del usuario.

CAPÍTULO IV: DESARROLLO

Usuarios y Documentación, junto con la opción Documentación que no se desarrollará durante el TIG, pero fue requerida sin inclusión en el portal para desarrollarla en un futuro; de igual manera la vista contiene el panel principal donde se cargan cada una de las opciones elegidas en el menú.

Una vez generadas las vistas de la interfaz gráfica base utilizando el lenguaje ZUL que es el formato estándar para las vistas en el Framework ZK, se procedió a generar las clases en java correspondientes a los controladores que en el Framework ZK reciben el nombre de Composer, todos incluidas en el paquete UI correspondiente a los controladores, estos son: el controlador MainComposer que se encarga de los eventos ocurridos en la ventana principal y LayoutTree que controla los eventos ocurridos en el menú desplegable, tales como desplegar el menú y mostrar en el panel principal la opción seleccionada.

Igualmente se creó el archivo CSS¹² que contiene los estilos base a utilizar durante todas las etapas del proyecto.

4.3.2.4 Pruebas

Una vez implementada la interfaz gráfica base, esta se presentó al área usuaria para realizar las pruebas correspondientes y dar por terminada la iteración. Las pruebas realizadas se encuentran reflejadas en el Apéndice 8.

4.3.3 Tercera Iteración: Módulo de Domiciliaciones

4.3.3.1 Análisis y Planificación

En el comienzo de esta iteración se realizó el levantamiento de información y requerimientos correspondientes al módulo de domiciliaciones.

Este módulo debe encargarse de la administración de empresas ordenantes y permitir a los usuarios responsables de las instituciones financieras, llevar a cabo los procesos de afiliación y desafiliación de empresas con las que se hayan establecido acuerdos de

¹² CSS: (Hoja de Estilo en Cascada) del inglés Cascading Style Sheets es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

CAPÍTULO IV: DESARROLLO

domiciliación interbancaria, para el procesamiento de transacciones de cobro a través de la Cámara de Compensación Electrónica (CCE). Para realizar las solicitudes de afiliación o desafiliación de empresas el portal debe utilizar la interfaz que provee el BCV hacia la Cámara de Compensación Electrónica, la cual recibe las solicitudes y las envía a la CCE que se encarga de procesar las solicitudes y devolver una respuesta a la interfaz que será tomada por el Portal y suministrada al usuario. Este proceso se encuentra reflejado en la siguiente figura:

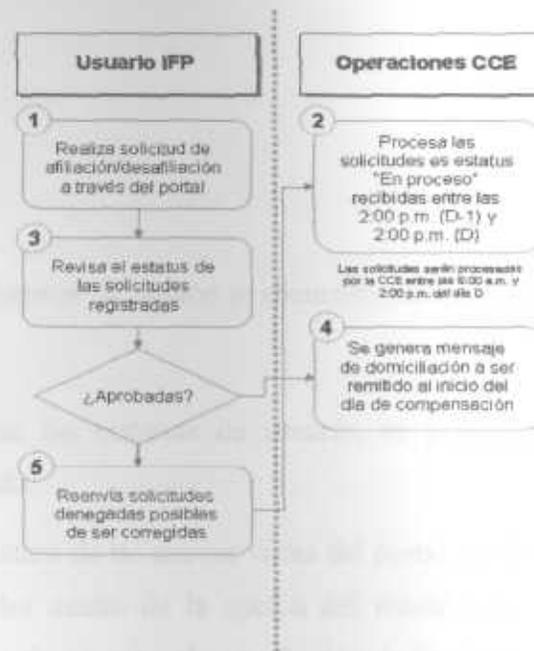


Figura 5 Procesos para solicitud de afiliación/desafiliación de empresas

Fuente: Dossier informativo para IFPs.

El módulo se divide en diferentes funcionalidades, una encargada de la gestión de empresas, otra de la gestión de clientes y por último la funcionalidad de administración del módulo. Las historias de usuario recopiladas durante esta fase son las siguientes:

- HU-3-01
- HU-3-02
- HU-3-03
- HU-3-04

CAPÍTULO IV: DESARROLLO

- HU-3-05
- HU-3-06
- HU-3-07
- HU-3-08
- HU-3-09
- HU-3-10
- HU-3-11
- HU-3-12
- HU-3-13
- HU-3-14
- HU-3-15
- HU-3-16

El detalle de estas historias de usuario se encuentra reflejado en el Apéndice 9.

4.3.3.2 Diseño

Una vez recopiladas las historias de usuario, se procedió a realizar el diseño estructural del módulo.

En cuanto a la estructura de las nuevas vistas del portal, se determinó la adición de 3 opciones desplegadas dentro de la opción del módulo de Domiciliaciones. Estas corresponderían a las funcionalidades de Gestión de Empresas, Gestión de Clientes y Configuración del Sistema.

A la vez, cada una de ofrecerá otras opciones en el menú desplegable para dividir los requerimientos del usuario en secciones que se muestran a continuación:

Gestión de Empresas: comprende las secciones de Consulta General, Registrar Solicitud Afiliación, Registrar Solicitud Desafiliación, Consulta Solicitudes, Consulta Afiliaciones, Generar Mensajes Empresas.

Gestión de Clientes: formado por las secciones Carga Archivos Solicitudes, Carga Archivos Respuestas, Consulta Archivos, Consulta Solicitudes/Respuestas, Consulta Solicitudes Vencidas, Consulta Solicitudes Rechazadas, Estadísticas.

CAPÍTULO IV: DESARROLLO

Configuración del Sistema: Incluye las secciones Institución Financiera, Motivo Desafiliación, Motivo Rechazo, Sector, Tipo Transacción, Código Resultado y Parámetros del Sistema.

En total se necesitan 36 archivos ZUL para generar los componentes de cada una de las vistas.

Las estructuras de datos necesarias para el módulo son las generadas en la primera iteración del proyecto, en conjunto con otras tablas que servirían de apoyo para suministrar la información necesaria y ya se encontraban implementadas en el sistema del BCV. Para la entrega de datos a la interfaz por parte del portal, es necesario diseñar una serie de procedimientos SQL (Stored Procedures) que serán activados según el proceso a ejecutar y facilitan la integración. En este caso los procedimientos se encargarán de la inserción, actualización y consulta de las empresas existentes en la interfaz, así como de las validaciones correspondientes.

En cuanto a la estructura lógica del módulo, fueron definidas las clases que se codificarían posteriormente en la fase de implementación: un paquete que contendría todos los controladores de las vistas creadas; paquetes para los Beans que encapsulan los atributos correspondientes a cada una de las tablas a utilizar; de igual manera otro paquete donde se encontrarían las distintas clases DAO y sus interfaces para la comunicación con la base de datos a través de iBatis. Por último el paquete de servicios, donde se encontraría toda la lógica correspondiente al módulo, así como su interfaz.

4.3.3.3 Implementación

A continuación se divide la implementación del módulo de Domiciliaciones en diferentes etapas:

- Primera Etapa: Fueron codificados en Oracle, mediante lenguaje SQL los Stored Procedures diseñados para interactuar con la interfaz del BCV. Cada uno de ellos realiza validaciones sobre los parámetros recibidos y a continuación realizan las inserciones correspondientes en la base de datos para

CAPÍTULO IV: DESARROLLO

que la interfaz se encargue de enviar las transferencias a la CCE. A continuación se muestra una lista de los procedimientos generados:

PROCESO	PROCEDIMIENTO	TABLAS AFECTADAS
Insertar RIF y razón social de la empresa	SICO.PKG_EMPRESA.INSERTAR	ONDC.EMPRESA
Actualizar la razón social de la empresa	SICO.PKG_EMPRESA.ACTUALIZA	ONDC.EMPRESA
Búsqueda de la razón social de la empresa por RIF	SICO.PKG_EMPRESA.LEER_RAZON_SOCIAL	ONDC.EMPRESA

Figura 6 Stored Procedures creados para las operaciones de Domiciliaciones

Fuente: Elaboración Propia

- Segunda Etapa: Se procedió a crear las vistas diseñadas previamente, utilizando el lenguaje ZUL. Posteriormente se crearon las estructuras generales de los controladores para cada una de estas vistas, que pueden ser identificados en el diagrama de clases por el sufijo Composer que acompaña a cada uno de ellos. Estos controladores gestionan los eventos que pueden ocurrir en cada una de las vistas.
- Tercera Etapa: Fueron creadas las clases de los Beans donde se encapsularían cada uno de los objetos de datos utilizados en el módulo. De igual manera fueron creadas las clases DAO y sus respectivas interfaces que permiten la interacción con las tablas de bases de datos utilizadas.
- Cuarta Etapa: Se crearon los archivos XML correspondientes al mapeo necesario para que iBatis realice todas las sentencias SQL necesarias y de ser necesario devuelva objetos java encapsulados en Beans, a través de los DAO correspondientes para su utilización en las demás capas de la aplicación. En este caso estas sentencias comprenden el llamado a los Stored Procedures implementados; así como las inserciones, eliminaciones, actualizaciones y diferentes consultas posibles correspondientes a las tablas implementadas para el módulo de Domiciliaciones.

CAPÍTULO IV: DESARROLLO

- Quinta Etapa: Una vez desarrolladas las clases correspondientes al acceso a datos, las vistas y sus controladores; se procedió a crear las clases e interfaces que permiten acceder vía servicios a las estructuras de datos correspondientes. Estas clases se encargan de realizar la interacción entre los controladores y la capa de acceso a datos; así como de implementar todas las validaciones necesarias a los archivos antes de realizar las inserciones o actualizaciones correspondientes. En general se generan servicios que son consumidos por los controladores para responder a las peticiones realizadas de las vistas, tales como consultas, eliminaciones, inserciones y validaciones tales como la estructura del RIF de las empresas. Igualmente fueron agregados los Beans necesarios al applicationContext para generar la inyección de dependencias en cada uno de los objetos generados.

Dentro de esta etapa también se creó un Job o tarea mediante el uso de Quartz que se encarga de verificar el vencimiento de las solicitudes enviadas o recibidas que no han sido respondidas. Para esto fue necesaria la importación de la librería y mediante la integración de Quartz con Spring, sólo fue necesaria la creación de una serie de Beans en el applicationContext que se encarga de llamar diariamente a las 7 horas al método aplicarVencimientoSolicitudCliente() el cual revisa las solicitudes de afiliación y verifica si la fecha de vencimiento, si igual a la fecha actual, en ese caso cambio el estatus de la solicitud a vencida.

- Sexta Etapa: Después de haber generado todos los servicios necesarios, se realizó una modificación a los controladores para, desde allí, realizar los llamados correspondientes a los servicios que realizan las validaciones, consultas, inserciones y modificaciones correspondientes según lo que sea requerido desde las vistas.
- Séptima Etapa: Se procedió a adecuar las vistas de consultas: general, de solicitudes/respuestas y de archivos; para implementar los nuevos métodos creados para filtrar los archivos por los parámetros de estatus, empresa, rangos de fecha, tipo de solicitud, búsquedas por RIF y demás parámetros necesarios.

- Octava Etapa: En esta última etapa se procedió a generar los reportes solicitados por los usuarios. Estos reportes fueron creados utilizando la herramienta JasperReports. El formato de los reportes fue creado a partir de formatos estándar existentes en otras aplicaciones del BCV. Para la generación del formato fue necesario utilizar la interfaz gráfica de la herramienta y diseñar el modelo de los reportes a generar tanto en Excel como en PDF, lo cual produce un archivo con extensión `.jrxml`¹³. Luego en la clase de servicios del módulo, utilizando la librería de JasperReports, se creó un método que inyecta al archivo del reporte la información que debe ser reflejada de acuerdo con la opción escogida en la vista.

4.3.3.4 Pruebas

Una vez realizada la implementación del módulo de Domiciliaciones, este fue presentado al área usuaria para que realizaran la revisión del flujo de los procesos que se producen al momento de realizar las solicitudes de afiliación o desafiliación de empresas ordenantes. De igual manera se realizaron las pruebas reflejadas en el Apéndice 11.

Al culminar la realización de las pruebas se dio por finalizada la iteración.

4.3.4 Cuarta Iteración: Módulo de Créditos Directos

4.3.4.1 Análisis y Planificación

Al inicio a esta iteración, fue necesario realizar el levantamiento de información para conocer el funcionamiento actual de la funcionalidad de envío de transferencias de tipo Crédito Directo hacia la interfaz BCV – Cámara de Compensación Electrónica. Para realizar el envío se requiere la utilización de la mencionada interfaz, la cual recibe las transacciones enviadas por los sistemas internos del BCV, las valida y las

¹³JRXML: Es el formato estándar de generado por JasperReports, es un archivo XML común con otra extensión para su reconocimiento por JasperReports.

CAPÍTULO IV: DESARROLLO

convierte al formato que recibe la CCE para luego enviarlas a la misma para su compensación y posterior liquidación financiera. En este caso, el portal enviará a la interfaz un lote de instrucciones de pago. La interfaz ordenará y enviará las transacciones a la CCE. Esa última realizará su procesamiento, que podría generar devoluciones o en su defecto aceptación. Luego se procede a ejecutar el proceso de liquidación en las cuentas corrientes de los Bancos Participantes en el BCV y finalmente, los bancos acreditan los fondos en las cuentas de los beneficiarios.

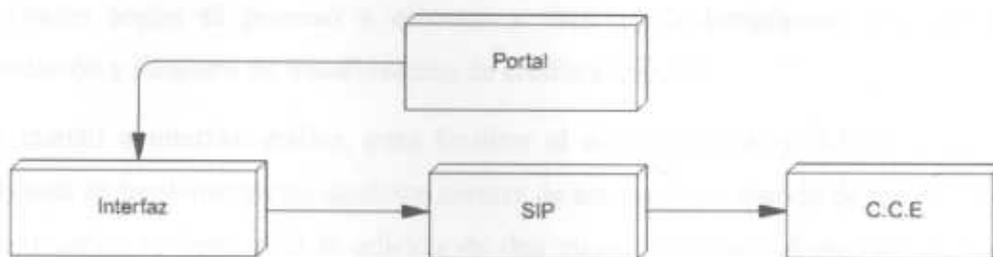


Figura 7 Representación gráfica del esquema de integración

Fuente: Elaboración Propia

La interfaz se encarga de ordenar las transferencias recibidas, generar los archivos de intercambio y enviarlos de forma automática a la CCE para su procesamiento.

Posteriormente, se realizaron una serie de reuniones con el área usuaria y equipo de proyecto donde se determinaron los requerimientos y necesidades para el módulo de créditos directos.

A partir de estas reuniones se recopilaron las siguientes historias de usuario:

- HU-4-01
- HU-4-02
- HU-4-03
- HU-4-04
- HU-4-05

Estas historias de usuario se encuentran detalladas en el Apéndice 13.

4.3.4.2 Diseño

Una vez determinadas las historias de usuario a implementar durante esta fase y a partir de las estructuras de datos creadas en la primera iteración; se procedió a realizar el diseño de la estructura del módulo.

Para la entrega de datos a la interfaz por parte de las aplicaciones del BCV, es necesario diseñar una serie de procedimientos Oracle (Stored Procedures) que serán activados según el proceso a ejecutar y facilitan la integración. En este caso Anulación y Registro de transferencias de créditos directos.

En cuanto a interfaz gráfica, para facilitar el uso del portal y debido a que más adelante se implementarían distintos niveles de acceso dependiendo de los privilegios del usuario; se determinó la adición de dos nuevas opciones dentro de la pestaña “Créditos Directos” en el menú principal. Una de ellas comprendería un listado donde se refleja toda la información disponible respecto a los archivos cargados, así como los reportes y la opción de cargar un nuevo archivo. La otra es una opción de parámetros, donde se permite al usuario la creación, modificación o eliminación de los diferentes estatus en que se puede encontrar un archivo, así como de los distintos motivos por los cuales se pueden anular. En total se crearon 6 páginas ZUL para las vistas que corresponderían al listado, 3 correspondientes a la vista de motivos de anulación y 3 más para los estatus del archivo.

Para la estructura lógica del módulo, se definieron las clases a utilizar durante la implementación: un paquete que contendría todos los controladores de las vistas creadas; paquetes para los Beans que encapsulan los atributos correspondientes al encabezado, nombre y registros de un archivo y los correspondientes a cada una de las tablas a utilizar; de igual manera otro paquete donde se encontrarían las distintas clases DAO y sus interfaces para la comunicación con la base de datos a través de iBatis. Por último el paquete de servicios, donde se encontraría toda la lógica correspondiente al módulo, así como su interfaz. En el Apéndice 16 se puede apreciar el diagrama de clases que surgió a partir del diseño.

4.3.4.3 Implementación

La implementación del módulo de Créditos Directos se divide en las siguientes etapas:

- Primera Etapa: Fueron codificados en Oracle, mediante lenguaje SQL los Stored Procedures diseñados para interactuar con la interfaz del BCV. Cada uno de ellos realiza validaciones sobre los parámetros recibidos y a continuación realizan las inserciones correspondientes en la base de datos para que la interfaz se encargue de enviar las transferencias a la CCE. A continuación se muestra una lista de los procedimientos generados:

PROCESO	PROCEDIMIENTO	TABLAS AFECTADAS
Registro de transferencias de crédito para su envío a la CCE.	ONLC.FN_NU_PROCESO_PAGO_CCE ONLC.SP_REGISTRO_DETALLE_CCE ONLC.SP_REGISTRO_ENCABEZADO_CCE	SEQ_NU_PROC_PAGO DET_PROCESO_PAGO DATO_PROCESO_PAGO
Anulación de transferencia de crédito	ONLC.SP_ANULA_PROCESO_CCE ONLC.SP_ANULA_DET_PROCESO_CCE	DATO_PROCESO_PAGO DET_PROCESO_PAGO

Figura 8 Stored Procedures creados para las operaciones de Créditos Directos

Fuente: Elaboración Propia

- Segunda Etapa: Se procedió a crear las vistas diseñadas previamente, utilizando el lenguaje ZUL. Posteriormente se crearon las estructuras generales de los controladores para cada una de estas vistas, que pueden ser identificados en el diagrama de clases por el sufijo Composer que acompaña a cada uno de ellos. Estos controladores gestionan los eventos que pueden ocurrir en cada una de las vistas.
- Tercera Etapa: Fueron creadas las clases de los Beans donde se encapsularían cada uno de los objetos de datos utilizados en el módulo. De igual manera fueron creadas las clases DAO y sus respectivas interfaces que permiten la interacción con las tablas de bases de datos utilizadas.

CAPÍTULO IV: DESARROLLO

- Cuarta Etapa: Se crearon los archivos XML correspondientes al mapeo necesario para que iBatis realice todas las sentencias SQL necesarias y de ser necesario devuelva objetos java encapsulados en Beans, a través de los DAO correspondientes para su utilización en las demás capas de la aplicación. En este caso estas sentencias comprenden el llamado a los Stored Procedures implementados; así como las inserciones, eliminaciones, actualizaciones y diferentes consultas posibles correspondientes a las tablas envio_archivo, h_envio_estatus, estatus_envio y motivo_anulacion.
- Quinta Etapa: Una vez desarrolladas las clases correspondientes al acceso a datos, las vistas y sus controladores; se procedió a crear las clases e interfaces que permiten acceder vía servicios a los estatus, motivos de anulación, historial y envío de archivos. Estas clases se encargan de realizar la interacción entre los controladores y la capa de acceso a datos; así como de implementar todas las validaciones necesarias a los archivos antes de realizar las inserciones o actualizaciones correspondientes. En general se generan servicios que son consumidos por los controladores para responder a las peticiones realizadas de las vistas, tales como consultas, eliminaciones, inserciones y validaciones. A través de Quartz se creó un Job o Tarea que se encarga de verificar diariamente a las 7 am la fecha valor de los archivos registrados, y en caso de llegar a la fecha correspondiente cambiar su estatus a "procesado".
- Sexta Etapa: Después de haber generado todos los servicios necesarios, se realizó una modificación a los controladores para, desde allí, realizar los llamados correspondientes a los servicios que realizan las validaciones, consultas, inserciones y modificaciones correspondientes según lo que sea requerido desde las vistas.
- Séptima Etapa: Se procedió a adecuar las vistas de envío de archivo para implementar los nuevos métodos creados para filtrar los archivos por los parámetros de fechas valor, de registro y estatus de los mismos.

- Octava Etapa: En esta última etapa se procedió a generar los reportes solicitados por los usuarios. Estos reportes fueron creados utilizando la herramienta JasperReports. El formato de los reportes fue creado a partir de formatos estándar existentes en otras aplicaciones del BCV. Para la generación del formato fue necesario utilizar la interfaz gráfica de la herramienta y diseñar el modelo de los reportes a generar, lo cual produce un archivo con extensión .jrxml. Luego en la clase de servicios del módulo, utilizando la librería de JasperReports, se creó un método que inyecta al archivo del reporte la información que debe ser reflejada de acuerdo con la opción escogida en la vista.

4.3.4.4 Pruebas

Una vez realizada la implementación del módulo, fue presentado al área usuaria para la revisión del flujo de los procesos al momento de realizar la carga de un archivo, su anulación y las diferentes consultas por parámetros que se pueden hacer sobre los archivos. De igual manera que el flujo para agregar o eliminar un motivo de anulación o estatus de archivo.

Las pruebas realizadas para esta iteración pueden apreciarse en el Apéndice 17. Para realizar algunas de las pruebas se desarrolló una función encargada de generar archivos de prueba a partir de la data que se encuentra registrada en la base de datos, y mediante esta función se generó una serie de archivos de operaciones de Créditos Directos para verificar el correcto funcionamiento del módulo y las validaciones correspondientes. Estos archivos fueron utilizados para la realización de varias de las pruebas. En el Apéndice 18 se refleja uno de los archivos generados para realizar las pruebas.

Al culminar exitosamente las pruebas se dio por finalizada la iteración.

4.3.5 Quinta Iteración: Módulo de Autenticación y Control de Acceso al Portal

4.3.5.1 Análisis y Planificación

Durante esta fase se determinaron los requerimientos específicos para el desarrollo del módulo de Autenticación y Control de Acceso al Portal. Estos requerimientos se encuentran contenidos en las siguientes historias de usuario:

- HU-5-01
- HU-5-02
- HU-5-03
- HU-5-04
- HU-5-05

En el Apéndice 19 se encuentra reflejado el detalle de cada una de estas historias de usuario.

4.3.5.2 Diseño

A partir de los requerimientos recopilados en la fase anterior de la iteración, se concluyó que para su implementación no es necesaria la creación de nuevas estructuras de datos, solo la utilización de las ya existentes en el sistema y correspondientes al protocolo LDAP implementado por la institución.

Para cumplir con los requerimientos de esta iteración en cuanto a las necesidades de interfaz gráfica, se optó por modificar la vista de pantalla principal desarrollada previamente. En el nuevo diseño de la vista se incluye un nuevo panel donde deben ser introducidos los datos para el inicio de sesión. Por lo tanto, es necesaria la creación del controlador correspondiente a esta vista, responsable de los eventos ocurridos en la misma.

Debido a que las funcionalidades de los módulos desarrollados fueron divididas en secciones, las delimitaciones de los accesos correspondientes a cada perfil se realizan a través del archivo de configuración de LDAP. De igual manera, la visualización de la información de la entidad propia para usuarios externos, y de todas las entidades

para usuarios internos, se validaría directamente en los servicios correspondientes a cada módulo, por lo que se necesitaría realizar una modificación de los mismos.

Se debe realizar la integración de la librería de seguridad suministrada por la institución para realizar la autenticación correspondiente al ingresar al portal. De igual manera, es necesario realizar la configuración correspondiente para la integración de Spring Security con el protocolo LDAP del banco. Sin embargo, no se requiere la creación de nuevas clases java en el proyecto.

4.3.5.3 Implementación

En primera instancia, se crearon 4 nuevos perfiles o grupos LDAP correspondientes a los tipos de usuario que manejará el portal. Esto fue realizado siguiendo los estándares de la institución y realizando la petición al departamento de seguridad informática de la institución, nuevamente llenando un formato como el que se puede apreciar en el apéndice 5.

Posteriormente, se realizó la integración a la aplicación de las librerías estándar suministradas por la institución para la utilización del protocolo LDAP y la seguridad en el portal; el framework ZK también provee librerías que se complementan con Spring Security, por lo que estas fueron incluidas en la aplicación. Se realizaron las modificaciones pertinentes en el contexto de la aplicación para incluir la seguridad a través de Spring Security utilizando el protocolo LDAP de la institución; así como las distintas configuraciones necesarias para su funcionamiento en la aplicación. Se crearon 2 nuevos archivos de configuración para integrar Spring Security con el protocolo: "ldap.properties" que contiene las propiedades necesarias para realizar la conexión al directorio; y el archivo xml "applicationContext-security" correspondiente a la configuración de Spring Security que se encarga de permitir o denegar el acceso a las distintas vistas de las funcionalidades de acuerdo al grupo LDAP al que pertenezca el usuario. Spring Security se encarga de realizar la autenticación de usuario y contraseña contra el directorio una vez se realiza el inicio de sesión; de igual manera permite consultar los datos relativos al usuario contenidos

CAPÍTULO IV: DESARROLLO

en el directorio, entre ellos la institución a la que pertenece y el grupo LDAP al que está asociado.

Luego, se procedió a realizar la modificación de la vista correspondiente a la pantalla principal, agregando los elementos necesarios para el login o inicio de sesión; se creó el controlador de la vista, que se encarga de realizar las validaciones en las estructuras tanto del nombre de usuario como de la contraseña y retorna a la interfaz el mensaje correspondiente en caso de ocurrir alguna excepción al ingresar al portal. Estas excepciones son generadas por Spring Security al realizar la validación contra el directorio.

A continuación, se realizó la modificación del archivo Index correspondiente a la vista del menú desplegable, para permitir la visualización de las distintas opciones dependiendo del grupo LDAP al que pertenezca el usuario, siendo esta información suministrada por Spring Security.

Por último, se modificaron las vistas y controladores correspondientes a todas las funcionalidades de consulta para permitir que sólo los usuarios internos puedan tener la opción de filtrar información por entidades bancarias. Los usuarios externos no tendrán esta opción y solo pueden consultar la información correspondiente a su institución. Esto fue logrado mediante el uso de condicionales que verifican a través de la información suministrada por Spring Security si el grupo asociado al usuario es interno o externo; de esta manera se permite ocultar o mostrar las opciones de filtrado correspondientes o llenar las tablas visualizadas con información de una institución o todas, dependiendo del grupo al que pertenezca el usuario.

4.3.5.4 Pruebas

Una vez implementada la autenticación y control de acceso al portal, se realizó la solicitud al departamento de seguridad y estructuras de datos para la creación de cuatro nuevos usuarios, cada uno perteneciente a un grupo LDAP distinto. Al ser creados los usuarios se procedió a realizar las pruebas correspondientes para comprobar los niveles de accesos que cada uno debía tener. Estas pruebas se pueden apreciar en el Apéndice 20. Posteriormente se dio por finalizada la iteración.

4.3.6 Sexta Iteración: Módulo de Administración de Usuarios

4.3.6.1 Análisis y Planificación

Para dar inicio a esta iteración, se realizaron reuniones con el área usuario y el líder del proyecto con el fin de aclarar algunos requerimientos y características del portal. A partir de estas reuniones se determinaron las siguientes historias de usuario para la iteración:

- HU-6-01
- HU-6-02
- HU-6-03

Las historias de usuario se encuentran detalladas en el Apéndice 21.

Al comienzo del planteamiento del proyecto, una de los requerimientos para el módulo de administración de usuarios era una sección de reportes disponible exclusivamente para el usuario interno administrador, que refleje los archivos cargados por entidad bancaria y usuarios. Estas funcionalidades se implementaron previamente en los módulos de Créditos Directos y Domiciliaciones, que permiten al usuario ingresado como interno administrador realizar consultas filtrando por entidad bancaria y obtener los reportes correspondientes. Por lo tanto, se acordó con el área usuaria que no se requiere incluir estas características dentro del módulo de administración ya que se encuentran cubiertos en los dos módulos previamente desarrollados.

4.3.6.2 Diseño

Para esta fase no fue necesario el diseño de nuevas estructuras de datos, ya que las tablas de usuarios y grupos LDAP a utilizar en el módulo se encontraban creadas previamente en el sistema de la institución.

Se realizó el diseño de las vistas a integrar en el portal. Para esto se dividirá el módulo en 3 funcionalidades: “Cambiar Password” que puede ser accedida por todos los tipos de usuarios; “Administración de Usuarios” y “Grupos LDAP” para el uso exclusivo del usuario interno administrador. Para añadir estas opciones a la interfaz

gráfica del portal se estimó la creación de 6 nuevos archivos ZUL para las vistas correspondientes.

En cuanto al diseño lógico es necesaria la creación de un archivo XML para la comunicación con la base de datos a través de iBatis y así administrar los grupos LDAP en la tabla correspondiente. Adicionalmente se requiere la creación de los controladores para las vistas a crear, así como los Beans correspondientes tanto a los grupos LDAP como a los usuarios y el archivo DAO correspondiente a grupos LDAP. La administración de usuarios se debe realizar a través de la librería suministrada por la institución, por lo que las llamadas a las funciones contenidas en las librerías se realiza directamente desde el archivo de servicios de LDAPy no es necesaria la creación del archivo XML para administrar usuarios.

Adicionalmente se requiere la creación de los archivos correspondientes a los servicios que proveen a los controladores los métodos necesarios para la administración de usuarios y grupos, así como las validaciones.

El diseño lógico creado para el módulo de administración de usuarios se ve reflejado en el diagrama de clases ubicado mostrado en el Apéndice 22.

4.3.6.3 Implementación

Para la implementación del módulo, inicialmente se procedió a crear los archivos ZUL correspondientes a las nuevas vistas y modificar la vista Index para incluir en el menú desplegable la opción de ingresar al módulo de administración y sus distintas funcionalidades.

Posteriormente, se procedió a crear el archivo XML, donde se crearon las sentencias SQL que permiten, a través de iBatis, realizar las consultas, inserciones y actualizaciones en la tabla "grupos_ldap" del sistema de la institución, para permitir la administración de los grupos a través del portal.

Una vez creado el archivo XML se desarrollaron los Beans encargados de encapsular los atributos de los grupos LDAP y de los usuarios. De igual manera, se implementaron los métodos en el archivo DAO de LDAP que se encargan de acceder

CAPÍTULO IV: DESARROLLO

a la base de datos mediante las llamadas a los procedimientos creados en el XML por medio de iBatis.

Para el manejo de usuarios, se creó el archivo de servicios “LdapService”, que se encarga de realizar llamados a la librería suministrada por la institución donde se encuentran los métodos necesarios para consultar usuarios, crear nuevo usuario, cambiar contraseña, restablecer contraseña, buscar usuario; también se encarga de encontrar los grupos a los que pertenece un usuario y asignar o remover grupos al mismo. Para el manejo de los grupos se creó el archivo de servicios “GrupoLdapService” que se encarga de realizar la interacción entre el controlador y el archivo DAO creado previamente.

4.3.6.4 Pruebas

Se realizaron reuniones con el equipo de trabajo para llevar a cabo las pruebas correspondientes al módulo. En el Apéndice 23 se muestran las pruebas realizadas.

Al completar todas las pruebas necesarias, se dio por finalizada la iteración.

Posteriormente fueron establecidas reuniones con el área usuaria para realizar una revisión de todos los módulos desarrollados. Para varios miembros del departamento de Créditos Directos se crearon distintos usuarios con diferentes niveles de acceso que se encargaron de probar cada una de las funcionalidades. Realizando una revisión del flujo de trabajo necesario para utilizar cada una de las funcionalidades.

Se crearon archivos de Domiciliaciones y Créditos Directos, para probar todas y cada una de las validaciones necesarias en los archivos. Igualmente se revisaron los reportes generados tanto en Excel como en PDF de cada una de las funcionalidades; el perfil Usuario Interno se encargó de asignar y revocar privilegios a los usuarios para comprobar el buen funcionamiento del módulo de administración de usuario y la seguridad.

Al finalizar exitosamente esta serie de pruebas, se cumplieron todas las responsabilidades y se lograron los objetivos establecidos en la propuesta del Trabajo Instrumental de Grado.

5. RESULTADOS

En el presente capítulo se muestran los resultados obtenidos a partir de la aplicación de la metodología XP en el desarrollo del Trabajo Instrumental de Grado.

Al finalizar el ciclo de vida del proyecto, a través del desarrollo de las diferentes fases e iteraciones dictadas por la metodología XP, se completaron los objetivos planteados en la propuesta. A continuación se muestran los resultados logrados en cada uno de los objetivos planteados:

- **Definir los requerimientos del usuario de la funcionalidad de atención de las solicitudes de registro de empresas ordenantes; así como del usuario de la funcionalidad para la atención de clientes externos BCV-IFP y generar el documento de requerimientos.**

A partir de las reuniones mantenidas tanto con el área usuaria como con el líder del proyecto, surgieron las primeras historias de usuario donde se establecían los requerimientos generales que debía cumplir el portal. Como resultado final de este objetivo se obtuvo la documentación de requerimientos del portal generada a partir de las historias de usuario, lo que permitió definir el diseño inicial del portal y establecer el plan de trabajo para el desarrollo del proyecto.

- **Diseñar e implementar el modelo de estructura de datos para que soporte la persistencia de información.**

En base a los requerimientos suministrados por los usuarios se realizó el análisis y diseño de las estructuras de datos necesarias para el almacenamiento de la información requerida para el portal de una manera eficiente y manteniendo la integridad de los datos. Se obtuvo como resultado una estructura de datos que satisface los requerimientos del cliente, garantizando la consistencia y unicidad de los datos. Esta estructura de datos se encuentra reflejada en los modelos entidad – relación (Figuras 3 y 4) que posteriormente fueron implementados en la base de datos de la institución por el departamento correspondiente.

CAPÍTULO V: RESULTADOS

- **Diseñar y desarrollar el Front End según el estándar del BCV.**

Para cumplir con este objetivo se realizó el diseño de la interfaz gráfica a implementar en el portal, siguiendo los lineamientos establecidos para las aplicaciones desarrolladas en la institución. Se obtuvo una interfaz gráfica que cumple con los estándares establecidos en el manual de estilos del BCV, ofreciendo a los usuarios un portal con una interfaz gráfica de uso sencillo y amigable. En los apéndices 24 y 25 se pueden apreciar respectivamente la pantalla de inicio sobre el diseño base del portal y el menú desplegable que permite el acceso a los módulos correspondientes.

- **Diseñar y desarrollar el módulo de autenticación y control de acceso al portal contra el protocolo de seguridad LDAP del BCV, el cual permite el acceso a los diferentes módulos de acuerdo al perfil de usuario que ingresa, así como la manipulación de cuenta por el usuario.**

Durante el levantamiento de requerimientos para este módulo se establecieron los cuatro tipos de usuario de los que dispondría el portal. A partir de ellos, se obtuvo un módulo que realiza la autenticación del usuario que ingresa al Portal y permite el acceso a las diferentes opciones y módulos dependiendo del grupo de usuario al que pertenezca. Esto se realiza autenticando los datos del usuario a través del protocolo LDAP utilizado por la institución.

- **Diseñar y desarrollar la funcionalidad de gestión de empresas para el módulo de domiciliaciones que comprende el registro de solicitudes tanto de afiliación como de desafiliación, la generación de mensajes de las empresas y las consultas de afiliaciones y desafiliaciones.**

Se obtuvo una sección dentro del módulo de Domiciliaciones llamado Gestión de Empresas, que permite a los usuarios representantes de las instituciones financieras participantes realizar las solicitudes de afiliación y desafiliación de empresas con las que se hayan establecido acuerdos de domiciliación interbancaria. De igual manera, permite al usuario descargar los mensajes de

CAPÍTULO V: RESULTADOS

registro de empresas ordenantes generados por la CCE. Por último brinda la opción de realizar consultas que pueden ser filtradas por múltiples parámetros de las afiliaciones y desafiliaciones realizadas por la institución financiera correspondiente; así como la descarga en archivos de formato PDF o Excel de cada uno de los reportes generados en las consultas. Estos resultados pueden ser apreciados en los apéndices desde el 25 al 35.

- **Diseñar y desarrollar la funcionalidad de gestión de clientes para el módulo de domiciliaciones que comprende la carga de archivos de solicitudes y respuestas; así como las consultas de archivos, afiliaciones, desafiliaciones, solicitudes y la sección de estadísticas.**

Se diseñó y desarrolló esta funcionalidad dentro del módulo de Domiciliaciones a partir de los requerimientos suministrados por los usuarios. Esta funcionalidad se encuentra dentro de la sección Gestión de Clientes y permite al usuario, en representación de una institución financiera, realizar la carga de archivos de solicitudes y respuestas relativos a la domiciliación de pagos de una empresa. De igual manera, la funcionalidad desarrollada permite realizar la consulta de los archivos cargados al sistema, así como las consultas y exportación a archivos formato PDF y Excel de los listados de afiliaciones, desafiliaciones y solicitudes de los clientes de la institución financiera, filtrándolas por múltiples parámetros para comodidad del usuario. Algunas de las funcionalidades pueden ser apreciadas en los apéndices del 36 al 39.

- **Diseñar y desarrollar la funcionalidad de administración del módulo de domiciliaciones, que comprende operaciones de ingreso, consulta, edición, eliminación para las IFP, motivos de desafiliación, motivos de rechazos, sector, tipo de transacción y código de resultado.**

La funcionalidad de administración desarrollada, se encuentra en la sección Configuración del Sistema dentro del módulo de Domiciliaciones. Esta opción se encuentra disponible solo para los usuarios internos del portal y les permite

CAPÍTULO V: RESULTADOS

realizar las operaciones de inserción, actualización, eliminación y visualización de las Instituciones Financieras Participantes en la Cámara de Compensación Electrónica, así como de las variables utilizadas en el portal como motivos de desafiliación, motivos de rechazos, sector, tipo de transacción y código de resultado. En el apéndice 40 se puede apreciar el detalle de una de las opciones de la funcionalidad, las demás opciones cumplen con el mismo formato.

- **Diseñar y desarrollar la funcionalidad de reportes que pueden ser exportados tanto en formato Excel como PDF para los distintos componentes de la administración de módulo de domiciliaciones.**

Al completar el objetivo se obtuvo como resultado la funcionalidad que permite exportar, tanto en formato PDF como Excel, los listados mostrados en todos los componentes incluidos en la sección de Configuración del Sistema, para permitir al usuario llevar un control de los distintos parámetros del sistema. En los apéndices 26, 27 y 39 se puede apreciar el formato de algunos de los reportes generados tanto en formato PDF como Excel.

- **Diseñar y desarrollar la funcionalidad de carga de archivos para el módulo de créditos directos.**

Como resultado de este objetivo, se obtuvo una funcionalidad que permite a los usuarios, en representación de la IFP correspondiente, realizar la carga de archivos para el ordenamiento de operaciones de créditos directos. Esta funcionalidad se encarga de realizar la validación de la estructura del archivo, así como del nombre, encabezado y cada uno de los registros del archivo y enviarlo posteriormente a la Cámara de Compensación Electrónica para su procesamiento. Permite consultar los datos incluidos en cada uno de los archivos, así como la anulación de un archivo cargado. En los apéndices comprendidos entre el 41 y el 44 se puede apreciar el detalle de algunos componentes de esta funcionalidad.

- **Diseñar y desarrollar la funcionalidad de reportes por selección de múltiples parámetros exportables en formato PDF para el módulo de créditos directos.**

CAPÍTULO V: RESULTADOS

La funcionalidad desarrollada permite realizar la consulta para la visualización de los datos correspondientes a los archivos de créditos directos cargados al sistema. Este listado de archivos puede ser filtrados por múltiples parámetros: rango de fechas de registro, rango de fechas valor y estatus del archivo. También permite a los usuarios externos visualizar los archivos cargados por la institución que representan. En el apéndice 46 se muestra un reporte de esta funcionalidad generado en PDF.

- **Diseñar y desarrollar el módulo de administración de usuarios disponible solo para el perfil de usuario interno que permite la administración de contraseñas y bloqueo de usuarios tanto internos como externos.**

A partir del desarrollo de este objetivo, se obtuvo como resultado un módulo que cuenta con diferentes opciones de administración. Una de ellas está disponible para todo tipo de usuario y permite realizar el cambio de contraseña propia.

El resto de las funcionalidades del módulo se encuentran disponibles solo para usuarios internos y les permite realizar la administración de todos los usuarios del portal, realizando búsquedas mediante nombre, apellido o código de identificación.

Dentro del módulo de administración un usuario interno puede modificar los datos de los demás usuarios, así como otorgar o revocar privilegios dentro del portal y realizar el cambio de clave. Permite también la administración de los grupos LDAP en caso de que sea necesario modificar la descripción de un grupo existente o la creación o deshabilitación de un grupo de ser requerido en algún momento. En los apéndices del 47 al 49 se muestra el detalle de algunas funcionalidades del módulo

- **Diseñar y desarrollar la funcionalidad de reportes para el módulo de administración de usuarios, la cual se muestra al usuario interno los detalles de los archivos cargados por los diferentes usuarios externos. Estos reportes se realizan por múltiples parámetros y pueden ser exportados en formato PDF.**

CAPÍTULO V: RESULTADOS

Esta funcionalidad fue implementada dentro de las secciones de Domiciliaciones y Créditos Directos. Al momento de ingresar como usuario interno a alguno de los módulos, se le brinda la posibilidad al usuario de filtrar a través de las instituciones los archivos cargados por los usuarios externos en representación de cada una de ellas. De esta manera el usuario interno administrador de la Cámara de Compensación Electrónica puede llevar un control de los archivos cargados por los usuarios de cada una de las instituciones participantes. Además de filtrar por institución, también cuenta con los filtros de múltiples parámetros y la posibilidad de descargar los listados generados en formato PDF. Unos ejemplos de archivos generados se pueden apreciar en los apéndices correspondientes a los módulos de Créditos Directos y Domiciliaciones.

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

A partir del cumplimiento de los objetivos establecidos en el Trabajo Instrumental de Grado y una vez realizado el análisis de los resultados obtenidos, se llegó a las siguientes conclusiones:

- El uso de la metodología XP es de gran utilidad, ya que la programación por iteraciones permitió que se realizaran entregas de funcionalidades en tiempos cortos, manteniendo un contacto constante con el usuario para la revisión de las funcionalidades entregadas. Esta integración del usuario con el desarrollo del proyecto, facilitó la realización de las adecuaciones necesarias al momento de desarrollar el módulo y permitió el cumplimiento de los requerimientos del usuario.
- La utilización de Spring Framework fue de gran ayuda para el desarrollo del proyecto. Debido a la gran cantidad de documentación que posee el Framework, facilitó la configuración necesaria para la implementación del patrón de diseño inyección de dependencias sobre la aplicación; igualmente su componente Spring Security proporcionó las funcionalidades necesarias para integrar el protocolo LDAP utilizado por la institución. Por último, permitió una integración óptima con el resto de los Frameworks utilizados durante el desarrollo.
- El uso del Framework ZK permitió la separación de las vistas y controladores de la lógica de la aplicación y el acceso a datos. Permitted realizar el diseño e implementación de una interfaz gráfica que cumple las necesidades de los usuarios y visualmente se ajusta a los requerimientos de un portal moderno, permitiendo una mayor capacidad de respuesta ya que el usuario interactúa con el servidor sin necesidad de recargar la página. Todo esto mientras facilita

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

la labor del desarrollador, permitiendo la implementación de AJAX sin necesidad de implementar código en JavaScript, ya que solo se requiere el diseño las páginas a través de componentes en el lenguaje ZUL que poseen características propias. Los eventos ocurridos en la interfaz se manejan directamente a través del lenguaje Java, que fue el utilizado para el resto de la aplicación.

- El módulo “Créditos Directos” representa un gran avance en cuanto al procesamiento de operaciones de créditos directos para los usuarios representantes de Instituciones Financieras Participantes de la Cámara de Compensación Electrónica. Permite a los usuarios realizar el procesamiento de una manera automatizada sin necesidad de utilizar medios físicos para el ordenamiento y reduciendo los tiempos de respuesta.
- El módulo “Domiciliaciones” automatiza el proceso de registro de empresas ordenantes para domiciliaciones, siendo de gran utilidad para los usuarios que anteriormente debían esperar entre 15 y 20 días para realizar esta operación. Igualmente, proporciona una interfaz de fácil uso para la realización de las operaciones correspondientes al módulo.
- El desarrollo de módulo “Administración de Usuarios” ofrece a los usuarios internos de la Cámara de Compensación Electrónica una herramienta útil para el monitoreo de las actividades llevadas a cabo a través del Portal. Proporcionando una serie de reportes sobre las actividades realizadas por cada uno de los usuarios de cada institución participante, igualmente brinda la oportunidad de gestionar los usuarios registrados en el sistema y revocar los privilegios en caso de ser necesario.

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.2 Recomendaciones

A lo largo del desarrollo del proyecto y una vez culminados los objetivos planteados, surgieron algunas recomendaciones para ser aplicadas en un futuro y así mejorar las funcionalidades del portal desarrollado:

- Implementar un mecanismo que permita realizar la carga masiva de varios archivos de Créditos Directos al mismo tiempo, para de esta manera reducir el tiempo invertido por los usuarios si requieren cargar una gran cantidad de archivos a la vez.
- Se recomienda la implementación de firmas digitales para una mayor seguridad a la hora de autenticar a los usuarios que realizan las operaciones en el portal.
- Implementación de mensajes para notificar a los usuarios al momento en que una solicitud de afiliación o desafiliación en el módulo de Domiciliaciones sea aprobada. De igual manera para el módulo de Créditos Directos cuando una operación ha sido procesada por la Cámara de Compensación Electrónica.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

- Apache Software Foundation. (s.f.). *Apache log4j*. Recuperado el Febrero de 2012, de <http://logging.apache.org/log4j/>
- Apache Software Foundation. (s.f.). *Apache Tomcat*. Recuperado el Enero de 2012, de <http://tomcat.apache.org/>
- Apache Software Foundation. (s.f.). *mybatis*. Recuperado el Enero de 2012, de <http://www.mybatis.org/>
- BCV. (2006). Documento informativo para IFPs. *Uso Interno*. Atos Origin S.A.E.
- BCV. (13 de 5 de 2008). BCVoz Económico. *Especial 2*.
- Beaulieu, A. (2005). *Learning SQL*. O'Reilly Media.
- Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional.
- Chen, H., & Robbie, C. (2007). *ZK Ajax without Script Framework*. Nueva York: FirstPress.
- Collins-Sussman, B., & Fitzpatrick, B. (2004). *Version Control with Subversion*. O'Reilly Media.
- Gosling, J., Joy, B., Steele, G., Bracha, G., & Buckley, A. (s.f.). *The Java Language Specification*. Recuperado el Febrero de 2012, de Java SE 7 Edition: <http://docs.oracle.com/javase/specs/>
- JasperForge. (s.f.). *JasperReports*. Recuperado el Febrero de 2012, de <http://jasperforge.org/projects/jasperreports>
- Lerman, J., & Miller, R. (2012). *Programming Entity Framework: DbContext*. O'Reilly Media.

REFERENCIAS BIBLIOGRÁFICAS

Microsoft Corporation. (s.f.). <http://technet.microsoft.com>. Recuperado el Febrero de 2012, de <http://technet.microsoft.com/es-es/library/cc737317%28v=ws.10%29.aspx>

Prasanna, D. (2009). *Dependency Injection*. Manning Publications.

Quartz Scheduler. (s.f.). Recuperado el Febrero de 2012, de <http://quartz-scheduler.org/>

Spring Source. (s.f.). *Spring Security*. Recuperado el Enero de 2012, de <http://static.springsource.org/spring-security/site/>

Walls, C., & Ryan, B. (2005). *Spring in Action*. Manning Publications.

World Wide Web Consortium (W3C). (s.f.). <http://www.w3c.es/>. Recuperado el Febrero de 2012, de <http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>