



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 1

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

Título: Sistema de Integración Bancaria, Desarrollo de los subsistemas
“Comunicación bancaria” y “Actualización de pagos”.

Autor: Rubén González Alfonso

Tutores: Ing. Ariel Ramírez Álvarez
Ing. Johann Rodríguez Hernández

La Habana, 2011

“Año 53 de la Revolución”

*"No hay en la tierra más vía, honrada,
que la que uno se abre con sus propios brazos".*

José Martí

Declaración de autoría

Declaro que soy el único autor del trabajo titulado: Sistema de Integración Bancaria, Desarrollo de los subsistemas Comunicación bancaria y Actualización de pagos, y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de ____ del 2011.

Rubén González Alfonso

(Autor)

Ing. Johann Rodríguez Hernández

(Tutor)

Ing. Ariel Ramírez Álvarez

(Tutor)

Agradecimientos

A todos los que, de una forma u otra, me apoyaron en la realización del Trabajo de Diploma.

A la Universidad, por poner a mi disposición todo lo necesario para formarme y ayudarme a crecer como persona, y por darme la oportunidad de hacer realidad mi sueño de estudiar y convertirme en un profesional.

A mis cuatro padres Senia (mami), Rubén (papi), Teresa (Mamita), Carlos (Papito) y a mi abuelo Pedro (Caruco) que son las personas que más alegran mi corazón, que cada vez que tropezaba en la vida pensar en ellos eran lo único que me daba fuerza para seguir adelante, le agradezco mucho a mis tías, tíos, primos, hermanos, por ayudarme tanto desde pequeños, por apoyarme incondicionalmente en todas mis decisiones, por guiarme siempre por el camino correcto y permitir que pudiera realizar mi sueño, por la fe que han depositado en mí, demostrando siempre que la familia es lo más importante del mundo, la unión y el amor que pueda existir en ella es lo que la hace linda.

A mi padrastro Roberto por estar al lado de mi madre cuidándola en todo este tiempo de estudio.

A mi novia Yuliet por su apoyo incondicional en todo momento.

A todas las personas que durante estos cinco años estuvieron a mi lado siempre compartiendo conmigo, haciendo que mi vida en la Universidad estuviera llena de buenos momentos. A Celia por ayudarme a mejorar el diseño del documento. En especial al clan2k: Yole, César, Yasma, Norkís y Ángel.

Agradecimientos

A Yole por ser mi amigo y mi hermano durante todo este tiempo, por entenderme y escucharme siempre, por aguantar todas mis pesadeces y apoyarme en todo, por hacerme saber que puedo contar con él para cualquier cosa.

A mis tutores Johann y Ariel, que desde el inicio han estado ahí para cada duda, inspirando apoyo y confianza.

A todos los integrantes del proyecto Sistema Integración Bancaria, que siempre fuimos muy unidos. A Fernando, Yanalía, Alexander, Juan Carlos, Isidro, Madiela, Marta, Yoandy y Ángel, especialmente a Fernando, que desde que llegué al proyecto me ha ayudado en todos los momentos difíciles sin decir nunca que no, demostrando ser un amigo en todo momento. A Yanalía que siempre estuvo dispuesta a brindarme su ayuda cuando la necesité y dejar que siempre la estuviera mortificando.

Dedicatoria

Dedico el presente trabajo al ser más dulce de mi vida, mi madre: Senia Alfonso Días por su amor, por guiarme siempre por un buen camino, por el cariño y la confianza que ha depositado en mí, por enseñándome siempre que lo más importante en el ser humano es hacer las cosas de corazón.

A mi padre Rubén por darme mis consejos en todo momento, por tener su apoyo, por enseñarme a ser hombre y saber valorar a los amigos, por ser una de las personas que me gustaría imitar.

A mi abuelo Pedro por ser la persona del carácter más lindo que he conocido, por ser un ejemplo de hombre y saber enfrentar las circunstancias de la vida.

A mamita y papito por quererme como su hijo más pequeño y luchar a mi lado desde niño.

A mi familia por tener su apoyo y ayuda en toda mi vida y la de mi madre.

A mis amigos, por ser mi familia y poder contar con ellos, dejando claro que haberlos conocidos ha sido uno de los grandes regalos que me ha dado la vida.

Resumen

La República de Cuba y la República Bolivariana de Venezuela, en aras de intensificar sus relaciones diplomáticas en los marcos del ALBA, mantienen en conjunto un acuerdo de colaboración, el cual comprende la concepción y ejecución de varios proyectos.

Uno ejemplo de estos proyectos es el Sistema Integración Bancaria, el cual cuenta con tres subsistemas para poder realizar el proceso de recaudación de pagos del Servicio Administrativo de Identificación, Migración y Extranjería (SAIME) con una mejor calidad. En el presente trabajo se realiza la implementación de dos subsistemas: Comunicación Bancaria y Actualización de pagos, que son los encargados de establecer comunicación con los sistemas de los bancos recaudadores que interactúan con el SAIME. Por esta razón se realizó el estudio de sistemas similares, se estudió la metodología y las tecnologías definidas en el proyecto, las cuales son libres de costo. Luego de realizar un profundo estudio se elaboró una propuesta de solución de acuerdo con la situación problemática que se detectó. Se realizó el análisis y diseño para llegar a cabo una mejor implementación, donde se generaron artefactos como: el diagrama de despliegue y el diagrama de componentes. Finalmente se validaron las aplicaciones obtenidas para demostrar que presentan la calidad que se requiere.

Índice

Índice

Introducción.....	1
Capítulo 1: Fundamentación Teórica.....	5
1.1 Introducción.....	5
1.2 Conceptos asociados al dominio del problema.....	5
1.2.1 Comunicación segura entre dos sistemas.....	5
1.2.2 Aplicación de escritorio.....	5
1.3 Análisis de sistemas similares.....	5
1.4 Estudio de la metodología y tecnologías.....	6
1.4.1 Metodología de desarrollo de software.....	6
1.4.1.1 El Proceso Unificado Racional (RUP).....	7
1.4.2 Lenguajes utilizados.....	9
1.4.2.1 Lenguaje unificado de modelado (UML).....	9
1.4.2.2 Lenguaje de programación (Java).....	10
1.4.2.3 XML.....	11
1.4.3 Herramienta de modelado (Visual Paradigm).....	12
1.4.4 Sistema gestor de bases de datos.....	12
1.4.4.1 PostgreSQL.....	13
1.4.5 Entorno de desarrollo (NetBeans).....	13
1.4.6 Mapeador de Objeto Relacional (Hibernate).....	14
1.4.7 Framework.....	14
1.4.7.1 Spring.....	14
1.4.8 Servicio web.....	15
1.4.8.1 Axis2.....	15
1.5 Conclusiones parciales.....	16
Capítulo 2: Propuesta de Solución.....	17
2.1 Introducción.....	17
2.2 Flujo de intercambio de ficheros con los bancos.....	17
2.3 Modelo de Dominio.....	18
2.4 Propuesta de solución.....	19
2.5 Especificación de los requisitos del sistema.....	20
2.5.1 Requisitos funcionales.....	20
2.5.2 Requisitos no funcionales.....	23
2.6 Modelo del sistema.....	24
2.6.1 Diagrama de Casos de Uso del Sistema.....	24
2.6.2 Descripción de los actores del sistema.....	26
2.6.3 Descripción de los casos de uso del sistema.....	27
2.7 Análisis.....	35

Índice

2.7.1	Diagrama de clases del análisis.	35
2.8	Diseño.	36
2.8.1	Descripción de la arquitectura.....	37
2.8.2	Patrones de diseño.	38
2.8.3	Diagrama de clases del diseño.	39
2.8.4	Diagramas de Interacción (Secuencia).	39
2.8.5	Modelo Físico de Datos.	39
2.9	Conclusiones parciales.....	41
Capítulo 3:	Implementación y Prueba.....	43
3.1	Introducción.....	43
3.2	Diagrama de Despliegue.....	43
3.3	Diagrama de Componentes.....	45
3.4	Descripción de las principales clases.	46
3.5	Estándares de código.	49
3.6	Modelo Prueba.....	50
3.6.1	Pruebas unitarias.	50
3.6.2	Pruebas funcionales. (Caja Negra).	51
3.7	Conclusiones parciales.....	52
Conclusiones	Generales.....	53
Recomendaciones.	54
Referencias	Bibliográficas	55
Bibliografía.....		56
Anexos		58
Anexo 1 :	Descripcion de casos de usos.	58
Anexo 2:	Casos de pruebas.	86
Anexo 3:	Diagrama de clases del análisis.	102
Anexo 4:	Diagrama de clases del Diseño.	108
Anexo 5:	Diagrama de secuencia del diseño.	116
Anexo 6:	Documento.	130

Índice Figuras

Figura. 1 Ciclo de vida de RUP.....	9
Figura. 2 Clase interfaz del análisis.	19
Figura. 3 Diagrama de casos de uso de la aplicación Gestión de comunicación bancaria.	25
Figura. 4 Diagrama de casos de uso de la aplicación Servicio de actualización de pagos.....	26
Figura. 5 Clase interfaz del análisis.	36
Figura. 6 Clase control del análisis.	36
Figura. 7 Clase entidad del análisis.	36
Figura. 8 Diagrama entidad relación.....	40
Figura. 9 Diagrama de despliegue.	44
Figura. 10 Diagrama de componentes Gestión de comunicación bancaria.....	45
Figura. 11 Diagrama de componentes Servicio de actualización de pagos.....	46
Figura. 12 DCA_Administrar lecturas de pagos.....	102
Figura. 13 DCA_Ejecutar lectura manual.....	103
Figura. 14 DCA_Ejecutar lectura automática.	104
Figura. 15 DCA_Autenticar usuario Gestión de comunicación bancaria.	104
Figura. 16 DCA_Autenticar usuario Servicio de actualización de pagos.....	105
Figura. 17 DCA_Enviar archivo seguridad pendiente.....	105
Figura. 18 DCA_Importar sucursales.....	106
Figura. 19 DCA_Configurar servidor SFTP.	106
Figura. 20 DCA_Generar archivo de seguridad.....	107
Figura. 21 DCD_Administrar lecturas de pago.....	108
Figura. 22 DCD_Ejecutar lectura manual.	109
Figura. 23 DCD_Ejecutar lectura automática.	110
Figura. 24 DCD_Autenticar usuario Gestión de comunicación bancaria.....	111
Figura. 25 DCD_Enviar archivo seguridad pendiente.....	112
Figura. 26 DCD_Importar sucursales.	113
Figura. 27 DCD_Configurar servidor SFTP.	114
Figura. 28 DCD_Generar archivo seguridad.	115
Figura. 29 DSD_Administrar lecturas de pago "Crear lectura.".....	116
Figura. 30 DSD_Administrar lecturas de pago "Modificar lectura.".....	117
Figura. 31 DSD_Administrar lecturas de pago "Habilitar lectura.".....	118
Figura. 32 DSD_Administrar lecturas de pago "Eliminar lectura.".....	119
Figura. 33 DSD_Ejecutar lectura automática "Iniciar.".....	120
Figura. 34 DSD_Ejecutar lectura automática "Pausa.".....	121
Figura. 35 DSD_Ejecutar lectura automática "Continuar.".....	122
Figura. 36 DSD_Ejecutar lectura automática "Detener.".....	123
Figura. 37 DSD_Ejecutar lectura manual.....	124
Figura. 38 DSD_Autenticar usuario Gestión de comunicación bancaria.....	125
Figura. 39 DSD_Enviar archivo seguridad pendiente.....	126
Figura. 40 DSD_Configurar servidor SFTP.	127
Figura. 41 DSD_Generar archivo seguridad.	128

Índice Figuras

Figura. 42 DSD_Importar sucursales.....	129
---	-----

Índice Tablas

Tabla. 1 Descripción de los actores del sistema.....	26
Tabla. 2 Descripción del caso de uso Administrar lecturas de pagos.	35
Tabla. 3 Clase interfaz del análisis.....	41
Tabla. 4 Descripción de la clase CLectura perteneciente a Servicio de actualización de pago.	47
Tabla. 5 Descripción de la clase CGenerarCodigoSeguridad perteneciente a Gestión de comunicación bancaria.....	49
Tabla. 6 Descripción del caso de uso Administrar lecturas de pagos.	66
Tabla. 7 Descripción del caso de uso Ejecutar lectura manual.	68
Tabla. 8 Descripción del caso de uso Ejecutar lectura automáticas.	73
Tabla. 9 Descripción del caso de uso Configurar servidor SFTP.	76
Tabla. 10 Descripción del caso de uso Generar archivo seguridad.	79
Tabla. 11 Descripción del caso de uso Enviar archivo de seguridad pendiente.	80
Tabla. 12 Descripción del caso de uso Importar sucursales.	82
Tabla. 13 Descripción del caso de uso Autenticar usuario.	85
Tabla. 14 Descripción del caso de uso Administrar lecturas de pagos Sección 1 “Crear Lectura”.	86
Tabla. 15 Descripción del caso de uso Administrar lecturas de pagos Sección 2 “Modificar Lectura”	88
Tabla. 16 Descripción del caso de uso Administrar lecturas de pagos Sección 3 “Habilitar Lectura”	88
Tabla. 17 Descripción del caso de uso Administrar lecturas de pagos Sección 4 “Eliminar Lectura”	89
Tabla. 18 Ejecutar lectura manual.....	90
Tabla. 19 Ejecutar lectura automática sección 1 “Iniciar”.....	91
Tabla. 20 Ejecutar lectura automática sección 2 “Pausa”.....	92
Tabla. 21 Ejecutar lectura automática sección 3 “Detener”.....	93
Tabla. 22 Ejecutar lectura automática sección 4 “Reiniciar”.....	94
Tabla. 23 Autenticar usuario sección 1 “Autenticar”.....	95
Tabla. 24 Autenticar usuario sección 2 “cambiar Usuario”.....	96
Tabla. 25 Importar Sucursales.....	97
Tabla. 26 Configurar servidor.....	99
Tabla. 27 generar archivo seguridad.	100
Tabla. 28 Enviar archivo seguridad pendiente.	101

Introducción

El surgimiento de la Alianza Bolivariana para la América (ALBA), iniciativa de integración para los países de América Latina y el Caribe, promovida por la República Bolivariana de Venezuela y a su vez liderada por su Presidente Hugo Rafael Chávez Frías, ha fortalecido las relaciones entre Cuba y Venezuela. Esto se evidencia mediante los contratos y acuerdos tomados para la realización de proyectos en busca del beneficio en diversos sectores de la sociedad de ambos países.

Venezuela ha prestado especial atención a la transformación tecnológica para la simplificación y seguridad de los procesos, donde una de las principales esferas que se priorizaron fue el servicio al ciudadano. Debido a estas causas surge la entidad Servicio Administrativo de Identificación, Migración y Extranjería (SAIME) cuyo objetivo fundamental consistió en renovar la tecnología del antiguo sistema en la Oficina Nacional de Identificación y Extranjería (ONIDEX), a partir de lo cual surge el Centro de Identificación y Seguridad Digital (CISED) en la Universidad de las Ciencias Informáticas (UCI), donde se ha jugado un papel importante en estas relaciones siendo el motor impulsor como empresa productora de software.

La dirección de gestión administrativa del SAIME, requiere de un sistema que garantice la fiabilidad y seguridad de la recaudación de pagos efectuados por los clientes en las sucursales de sus bancos recaudadores, el cual es llevado a cabo por el proyecto Identidad Venezuela perteneciente al CISED. Por todo ello, se propone la solución informática Sistema de Integración Bancaria (SIB) que cuenta con un subsistema principal llamado Gestión de Solicitudes y Pagos, el cual necesita una herramienta de apoyo para establecer comunicación digital con el Sistema en Bancos (SeB) para su completo funcionamiento, dando a la luz las siguientes **problemáticas**:

- ✓ No existe una comunicación digital entre SIB y SeB.
- ✓ Cada banco recaudador intercambia los datos en diferentes formatos con el proveedor de servicio a la hora de actualizar la información de los pagos efectuados en las sucursales.
- ✓ Es necesario generar los códigos de seguridad para sus entidades bancarias, notificando a los involucrados de los cambios producidos.

Introducción

De acuerdo a las razones explicadas anteriormente el **problema científico de la investigación** queda formulado de la siguiente manera:

¿Cómo garantizar la confiabilidad y seguridad de las notificaciones de pago y la comunicación entre el SIB y SeB?

Para dar respuesta al problema se define como **objetivo general**:

Desarrollar un sistema para la transferencia segura de los datos de los pagos entre los bancos de la República Bolivariana de Venezuela y el SAIME.

De donde se derivan los siguientes **objetivos específicos**:

- Desarrollar los subsistemas Comunicación bancaria y Actualización de pagos.
- Realizar las pruebas a los subsistemas Comunicación bancaria y Actualización de pagos.

Constituye el **objeto de estudio** el proceso de comunicación entre sistemas informáticos y como **campo de acción** el proceso de comunicación entre el SIB y el SeB.

La investigación se sustenta en la siguiente **idea a defender**:

El desarrollo de los subsistemas Comunicación bancaria y Actualización de pagos facilitará una comunicación segura y fiable en la transferencia de los datos de los pagos entre el SIB y el SeB.

Los **métodos científicos** empleados son:

Métodos Teóricos:

- **Analítico-sintético:** se utilizó para descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de la propuesta.
- **Inductivo-deductivo:** sirvió para llegar a un grupo de conclusiones sobre lo que se quiere lograr.
- **Histórico-lógico:** se utilizó en el estudio de trabajos hechos que están relacionados con el tema tratado en esta investigación, para utilizarlos como puntos de referencia.

Métodos Empíricos:

- **Observación:** método de la observación: permitió analizar cada fase del proceso e ir observando las tareas realizadas y tomar experiencia de estas para aplicarlas en todas las demás.

Introducción

Con el propósito de guiar y perfilar el trabajo hacia el alcance de los objetivos trazados, se definieron las siguientes **tareas de investigación**:

- Elaboración del diseño teórico de la investigación.
- Estudio y valoración de los documentos:
 - ✓ Especificación de requisitos de software.
 - ✓ Especificaciones técnicas sobre la solución de software de integración entre los Bancos recaudadores y el SAIME.
 - ✓ Proyecto técnico Sistema de integración con los bancos.
- Investigación sobre la metodología y las herramientas a utilizar para el modelado del sistema informático, así como el marco de trabajo propuesto por el grupo de arquitectura del proyecto.
- Elaboración del análisis de los subsistemas Comunicación bancaria y Actualización de pagos.
- Elaboración del diseño de los subsistemas Comunicación bancaria y Actualización de pagos.
- Implementación de los subsistemas Comunicación bancaria y Actualización de pagos.
- Realización y diseño de pruebas a los subsistemas Comunicación bancaria y Actualización de pagos.

Luego de haber realizado las tareas y haber dado cumplimiento a los objetivos que se han planteado, los **aportes prácticos esperados del trabajo** son los siguientes:

- Una comunicación segura y fiable entre el SIB y el SeB.
- Gestión de los ficheros que intercambian los bancos con el proveedor de servicios para actualizar la información de los pagos efectuados en las sucursales de los bancos.

Estructuración de capítulos.

Capítulo 1: Fundamentación Teórica, es el respaldo teórico de los temas tratados en el informe, está dedicado a exponer los fundamentos teóricos relacionados con el objeto de estudio. Incluye el análisis de la información existente acerca de las tendencias y tecnologías a nivel nacional e internacional, se estudian la metodología y tecnologías escogidas por el grupo de desarrollo del proyecto empleadas en el desarrollo de los subsistemas Comunicación bancaria y Actualización de pagos del proyecto SIB.

Capítulo 2: Propuesta de Solución, se describe el objeto de estudio en términos de negocio, se plantean los requerimientos funcionales y se definen los requisitos no funcionales, se identifican los casos de uso y

se realiza una propuesta de la solución de acuerdo a las características que deben cumplir los sistemas. Se desarrollan las fases de análisis y diseño definidas por la metodología seleccionada.

Capítulo 3: Implementación y Prueba, se describe la distribución física del sistema mediante el modelo de despliegue y el diagrama de componentes. Se lleva a cabo la implementación de los subsistemas basándose en el análisis y diseño, se realizan pruebas al producto final con el objetivo de garantizar la calidad del sistema.

Capítulo 1: Fundamentación Teórica.

1.1 Introducción.

En este capítulo se definen los conceptos básicos asociados al dominio del problema. Se justifican la metodología y tecnologías que se seleccionaron en el proyecto y que se utilizarán en el desarrollo de los subsistemas. Se realiza una investigación crítica del estado del arte de aplicaciones con características similares.

1.2 Conceptos asociados al dominio del problema.

1.2.1 Comunicación segura entre dos sistemas.

Se basa en las medidas de seguridad que se toman cuando se realizan transferencias de información entre dos sistemas informáticos para asegurar la integridad y confidencialidad de los datos.

1.2.2 Aplicación de escritorio.

Es la aplicación creada para ejecutarse en un ordenador de escritorio, sobre un sistema operativo de interfaz visual como *Windows* o *Linux*. Estas aplicaciones brindan una mayor capacidad gráfica visual y son rápidas, ya que tienen un menor tiempo de respuesta que las aplicaciones web. Las aplicaciones requieren de instalación y es necesario actualizar a cada cliente que esté usándolas en caso de que ocurran cambios.

1.3 Análisis de sistemas similares.

SAREN¹ es un organismo de la República Bolivariana de Venezuela que gestiona todo lo relacionado con registrar documentos de ventas de bienes muebles e inmuebles, compañías anónimas, firmas personales y registro de títulos universitarios entre otros servicios. También da garantía de seguridad jurídica en el país. SAREN se divide en cuatro tipos de oficinas, los Registros Principales, los Registros Públicos, los Registros Mercantiles y las Notarías Públicas.

¹ SAREN: Servicio Autónomo de Registros y Notarías.

Capítulo 1: Fundamentación Teórica

El proyecto de Automatización de los Registros y Notarías perteneciente a la Universidad de las Ciencias Informáticas, crea el sistema de pagos que utiliza SAREN para realizar su recaudación de pagos. Esta solución de pago integra el sistema de SAREN con el sistema de sus bancos recaudadores, para ello definieron datos estándar a transferirse. Teniendo en cuenta la seguridad de los datos se precisaron algoritmos de codificación y actualización de las cadenas de seguridad que se le generan a las sucursales de los bancos. Los cuatro bancos recaudadores de SAREN se encuentran dentro de los bancos que se integrarán con el SAIME. Debido a las razones explicadas anteriormente surge la idea de que el SIB reutilice el mecanismo de seguridad para validar la integridad de la información del sistema de pago seguro de SAREN ya desarrollada y puesta en explotación, cumpliendo de esta manera con los requisitos que se necesitan desarrollar.

1.4 Estudio de la metodología y tecnologías.

A continuación se realizará un estudio de la metodología y tecnologías que fueron seleccionadas por acuerdos del proyecto para realizar las dos aplicaciones.

1.4.1 Metodología de desarrollo de software.

Las metodologías de desarrollo de software surgieron a raíz de la necesidad de garantizar la calidad del desarrollo de un software. En la actualidad son muchas las metodologías de desarrollo de software que existen, el uso de una u otra dependen de las características de cada proyecto. Las metodologías de desarrollo de software pueden ser clasificadas en dos tipos: las tradicionales y las ágiles. Las metodologías tradicionales se centran especialmente en el control del proceso, estableciendo las actividades involucradas, los artefactos que se deben producir y las herramientas que se usarán. Por otra parte, las metodologías ágiles se centran en el factor humano o el producto software, dando mayor valor a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas.

1.4.1.1 El Proceso Unificado Racional (RUP).

El Proceso Unificado de Racional (RUP) es una metodología tradicional que hace uso del UML como lenguaje de modelado. Esta metodología describe detalladamente todas las actividades, roles, responsabilidades, productos de trabajo y herramientas para definir el quién, cómo, qué y cuándo en un proyecto de desarrollo de software. Su objetivo es transformar los requisitos de un usuario en un sistema de software dentro de un horario predecible y presupuesto. RUP posee tres características que lo identifican y lo hacen diferente a las demás metodologías, dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental. Divide el proceso de desarrollo en cuatro fases y nueve flujos de trabajo, seis de ingeniería y tres de soporte.

Las principales características de RUP son: (1)

- ✓ Centrado en la arquitectura: la arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas de software, sistemas operativos, manejadores de bases de datos, protocolos para la comunicación de red, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Una vez definida la arquitectura se puede decir que el sistema tiene forma.
- ✓ Dirigido por casos de uso: tiene a los casos de uso como el hilo conductor del proceso de desarrollo, los desarrolladores crean una serie de modelos de diseño e implementación que los llevan a cabo. Además, estos modelos se validan para que sean conformes a los casos de uso. Finalmente, los casos de uso también sirven para realizar las pruebas sobre los componentes desarrollados.
- ✓ Iterativo e incremental: esta característica propone dividir el proceso de desarrollo en partes, cada una de las cuales incluya las fases de: Requerimientos, Análisis, Diseño, Implementación y Pruebas, con el objetivo de acelerar el ritmo de desarrollo para que el producto salga al mercado en el menor tiempo posible y con mayor calidad.

Capítulo 1: Fundamentación Teórica

Fases

- ✓ Inicio: alcanzar un acuerdo entre todos los interesados respecto a los objetivos del ciclo de vida para el proyecto, generando el ámbito del proyecto, el caso de negocio, determinar la arquitectura candidata y el alcance del proyecto.
- ✓ Elaboración: cuyo principal propósito es el establecimiento de una línea base para la arquitectura del sistema y proporcionar una base estable para el diseño y el esfuerzo de implementación en la fase siguiente.
- ✓ Construcción: en esta fase se profundiza en el diseño de los componentes y de manera iterativa se van añadiendo las funcionalidades al software a medida que se construyen y prueban, permitiendo a la vez que se puedan ir incorporando cambios.
- ✓ Transición: esta fase se ocupa de la instalación del software a los usuarios finales que harán uso del sistema. Como consecuencia de esto pueden surgir nuevos requisitos.

Flujos de Trabajo

Flujos de ingeniería

- ✓ Modelación del negocio: describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- ✓ Requerimientos: define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- ✓ Análisis y diseño: describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- ✓ Implementación: define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- ✓ Prueba: busca los defectos a lo largo del ciclo de vida.
- ✓ Despliegue: produce un entregable del producto y realiza actividades como empaque, instalación, asistencia a usuarios, para entregar el software a los usuarios finales.

Capítulo 1: Fundamentación Teórica

Flujos de soporte:

- ✓ Gestión de configuración y cambios: describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos y control de versiones.
- ✓ Gestión del proyecto: involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- ✓ Gestión de entorno: contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

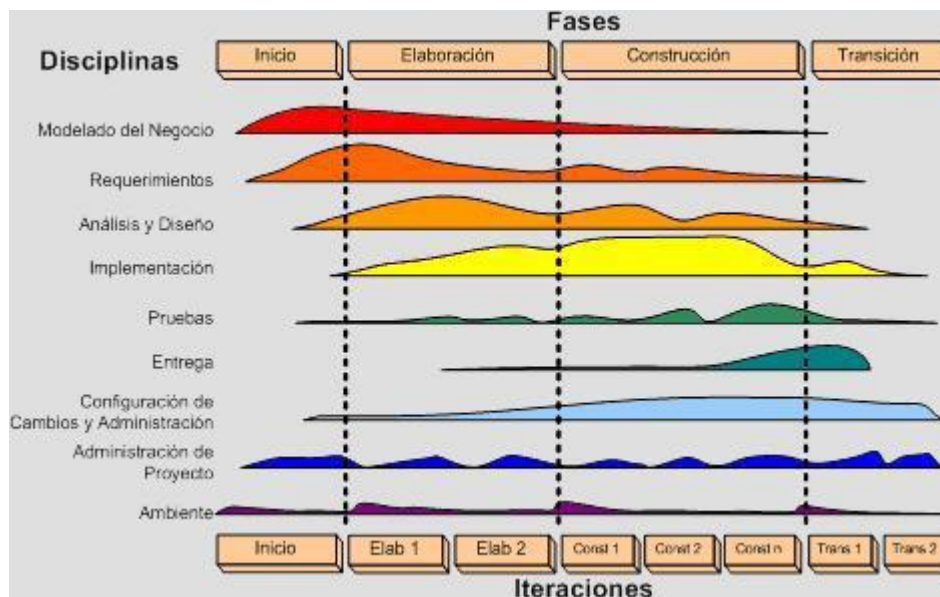


Figura. 1 Ciclo de vida de RUP.

1.4.2 Lenguajes utilizados.

1.4.2.1 Lenguaje unificado de modelado (UML).

El modelado es esencial para la construcción de software para comunicar la estructura de un sistema complejo, especificar el comportamiento deseado del sistema, comprender mejor lo que se está construyendo, descubrir oportunidades de simplificación y reutilización.

El Lenguaje Unificado de Modelado es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. Este lenguaje pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. (2)

UML tiene como objetivo permitir visualizar de una forma gráfica un sistema de tal manera que pueda ser entendido por todos, especificando cuáles son las características antes de que comience a ser construido.

1.4.2.2 Lenguaje de programación (Java).

Los lenguajes de programación son lenguajes creados por el ser humano para poder comunicarse con las computadoras. Está compuesto por un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.

Los lenguajes de programación según su nivel de abstracción se clasifican en bajo, medio y alto nivel. En los de bajo nivel la programación se realiza teniendo muy en cuenta las características del procesador, por ejemplo, lenguaje ensamblador y de máquina. Los de nivel medio se encuentran en un punto medio entre los lenguajes de nivel bajo y alto, ya que pueden acceder a los registros del sistema, trabajar con direcciones de memoria y a la vez realizar operaciones de alto nivel, por ejemplo C. Los de alto nivel son aquellos en los que las instrucciones son escritas con palabras similares a las del lenguaje humano, esto permite facilitar la escritura y la comprensión del código al programador, por ejemplo C++, C# y Java.

Java es un lenguaje que toma una estructura y sintaxis similar a C++, aunque tiene un modelo de objetos mucho más simple y elimina la manipulación de punteros. Proporciona un lenguaje independiente de la plataforma, es software libre, y evita la posibilidad de codificar virus sobre él.

Características de Java:

- ✓ Orientado a objetos: Java soporta las características esenciales del paradigma de la programación orientada a objetos: encapsulación, herencia y polimorfismo.
- ✓ Robusto: realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación de memoria.
- ✓ Multiplataforma: Java ofrece la posibilidad de que los archivos que son generados para una aplicación sean independientes de la plataforma, es decir, que se compilen una vez y se ejecuten en cualquier plataforma. Cuando se compila el programa se generan archivos conocidos como *bytecode*², los cuales van a ser interpretados en cualquier sistema operativo con un intérprete de Java.
- ✓ Alto rendimiento: permite el soporte de la sincronización de múltiples hilos de ejecución a nivel de lenguaje, llevando a cabo sus funciones de manera concurrente.

1.4.2.3 XML.

Lenguaje de Marcas Extensible (XML), es un metalenguaje usado para definir documentos que contienen datos estructurados, permite definir la gramática de lenguajes específicos. Uno de los usos de XML es que provee múltiples vistas para los datos. En efecto, los documentos XML definen el modelo de datos y luego se puede crear más de una vista basándose en las necesidades de la aplicación que se está desarrollando. Los datos se representan en un formato que es fácil de entender tanto por los seres humanos como por los computadores, específicamente por el software. Los documentos XML están bien estructurados, cada documento tiene un elemento raíz y todos los demás elementos deben estar anidados dentro de otros elementos. Otra aplicación de gran significado es que permite representar los datos que serán intercambiados o cedidos entre aplicaciones diferentes. Dado que XML describe los datos de manera fácil de analizar y que no está atado a un lenguaje de programación en específico, es posible

² El *bytecode* es un código intermedio más abstracto que el código máquina. Habitualmente es tratado como un fichero programa binario que contiene un ejecutable similar a un módulo objeto, que es un fichero binario producido por el compilador, cuyo contenido es el código objeto o código máquina. El *bytecode* recibe su nombre porque usualmente cada código de operación tiene una longitud de un byte.

transferir información entre aplicaciones fácilmente, incluso si estas aplicaciones residen en diferentes sistemas operativos o son escritos en diferentes lenguajes de programación. De hecho, como mismo se dice que Java provee una interoperabilidad entre plataformas, XML provee el mismo tipo de interoperabilidad para los datos.

1.4.3 Herramienta de modelado (Visual Paradigm).

Visual Paradigm es una herramienta UML profesional que ayuda a los equipos de desarrollo en la confección de los distintos modelos que van desde la construcción hasta el despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad y a un menor costo. Permite graficar varios tipos de diagramas de clases. Esta herramienta también proporciona una abundante documentación de UML. (3)

Está diseñado para una amplia gama de usuarios, incluidos los Ingenieros de Software, Analistas de Sistema, Analistas de Negocios, Sistema de Arquitectos, que estén interesados en la construcción de sistemas de software.

1.4.4 Sistema gestor de bases de datos.

Un Sistema Gestor de Bases de Datos (SGBD) es un conjunto de programas que permite definir, construir y manipular la Base de Datos (BD), garantizando la seguridad e integridad de los datos. Uno de sus principales objetivos es proporcionar a los usuarios una visión abstracta de los datos, es decir, el sistema esconde ciertos detalles desde cómo se almacenan y mantienen los datos, recuperándolos eficientemente.

Las características que deben cumplir los SGBD son: abstracción de la información, la independencia de los datos, redundancia mínima, consistencia en los datos, seguridad, integridad, respaldo y recuperación, tiempo de respuesta y control de concurrencia. (4)

Dentro de los SGBD se encuentran los comerciales y los libres, en los comerciales más reconocidos están SQL-Server y Oracle, y en los libres PostgreSQL y MySQL.

1.4.4.1 PostgreSQL.

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD³. El código fuente de PostgreSQL está disponible a cualquier persona libre de cargos directos, permitiendo a cualquiera colaborar con el desarrollo del proyecto o modificar el sistema para ajustarlo a sus necesidades. (5)

Entre sus principales características se encuentran: (6)

- ✓ Alta concurrencia, permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.
- ✓ Consistencia, es la propiedad que asegura que solo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
- ✓ Aislamiento, es la propiedad que asegura que una operación no puede afectar a otras. Esto garantiza que dos transacciones sobre la misma información nunca generarán ningún tipo de error.
- ✓ Durabilidad, es la propiedad que asegura que una vez realizada la operación, esta persistirá y no se podrá deshacer aunque falle el sistema.

1.4.5 Entorno de desarrollo (NetBeans).

NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java; pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. (7)

Este entorno de desarrollo ofrece varias ventajas a los programadores tanto de Java como de otros lenguajes, ya que es un IDE de código abierto, soporta el desarrollo de todos los tipos de aplicación Java (escritorio, web y aplicaciones móviles), posee un editor de texto que permite el completamiento de código, aspecto importante tanto para principiantes como para expertos en el lenguaje.

³ BSD, *Berkeley Software Distribution*: licencia de software otorgada principalmente para los sistemas.

1.4.6 Mapeador de Objeto Relacional (Hibernate).

En el mundo hay muchos debates acerca de cómo persistir los objetos, y entre ellos el uso de las herramientas de persistencia como una buena solución para llevar a cabo la implementación del acceso a datos de un proyecto de software. Hibernate es ejemplo de estas herramientas desarrolladas para la plataforma Java. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, herencia y varios tipos de datos. Estos objetos son conocidos como POJOs (*Pain Old Java Objects*), los que establecen una relación directa entre el modelo de datos relacional a través de ficheros XML, ambos generados por el propio Hibernate. Esta herramienta es independiente del Sistema gestor de base de datos (SGBD) y posee un lenguaje propio de consultas llamado HQL (*Hibernate Query Language*). Hibernate posee un API para construir las consultas programáticamente conocidas como *criteria*, las cuales son más sencillas a la hora de emplear y reducen las fallas al crear código SQL. Además, este Mapeador de Objeto Relacional (ORM) tiene dos niveles de caché, guardando datos en memoria; lo que posibilita liberar la carga a la base de datos y permitir así multitud de consultas simultáneas sin colapsar la base de datos. Es software libre y con bastante documentación.

1.4.7 Framework.

Un *framework* puede ser definido como un conjunto de clases y las colaboraciones que establecen entre ellas, para proporcionar un diseño abstracto para las soluciones de un conjunto de problemas. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Un *framework* captura las decisiones de diseño comunes a un tipo de aplicación, estableciendo un modelo común a todas ellas, asignando responsabilidades y estableciendo colaboraciones entre las clases que forman el modelo.

1.4.7.1 Spring.

Spring es un *framework* de código abierto de desarrollo de aplicaciones para la plataforma Java, facilitando la integración de estas con otros componentes. Spring posee una potente gestión de la configuración, características fundamentales para manejar objetos de negocio, la Inversión de Control (IoC), esta parte se encarga de separar del código de las aplicaciones que se están desarrollando, los aspectos de configuración y las especificaciones de dependencia del *framework*. La IoC mediante la

Inyección de Dependencia (ID) permite inyectar las dependencias de un *bean* en el momento de su creación, utilizando un manejador externo, promoviendo el bajo acoplamiento de las clases.

Spring fomenta las buenas prácticas de programar orientado hacia interfaces más que a clases. (8) Su integración con el ORM Hibernate ofrece un manejo seguro y eficiente de sus sesiones, (9) posibilita definir estrategia para las transacciones de forma declarativa y programada y da soporte a la implementación de los DAOs.

1.4.8 Servicio web.

Los servicios web representan un conjunto de aplicaciones o tecnologías con el fin de intercambiar datos en la web. Estos proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones que interactúan entre sí para presentar información dinámica al usuario. Proporciona interoperabilidad y extensibilidad entre aplicaciones que pueden diferir desde el tipo de sistema operativo hasta el lenguaje donde fue desarrollado, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas.

1.4.8.1 Axis2.

Axis2 permite la creación y el despliegue de servicios web. Axis2 es orientada a XML, utiliza su propio modelo de objeto que analiza para mejorar la velocidad de procesamiento. Permite despliegue en caliente, es decir, los nuevos servicios se pueden agregar al sistema sin tener que cerrar el servidor. Provee un cierto número de módulos que se vienen desarrollando en relación a la seguridad.

Es un paquete Java libre que provee: (10)

- Un entorno que puede funcionar como un servidor SOAP⁴.
- Una API para desarrollar servicios web.
- El soporte para diferentes capas de transporte entre las que se encuentran: HTTP y FTP.
- La serialización/deserialización automática de objetos Java en mensajes SOAP.

⁴ SOAP (*Simple Object Access Protocol*): protocolo de intercambio entre aplicaciones ejecutadas sobre cualquier plataforma. El formato de intercambio está basado en el lenguaje XML. En otras palabras el llamado al servicio SOAP pone en marcha un flujo ASCII encerrado entre etiquetas XML y transportado en el protocolo HTTP.

- Herramientas para crear automáticamente WSDL⁵ correspondientes a clases Java o inversamente para crear clases Java sobre la base de un WSDL.
- Herramientas para desplegar, testear y monitorear servicios web.

1.5 Conclusiones parciales.

Durante la investigación se decidió reutilizar el mecanismo de seguridad para validar la integridad de la información del sistema de pago seguro de SAREN, cumpliendo con los requisitos necesarios que se necesitan desarrollar. Se analizó RUP como metodología de desarrollo del software, por las grandes iteraciones que esta posee así como la gran cantidad de plantillas que esta genera, para así lograr un mayor control sobre el producto. El framework seleccionado es Spring, el cual se integrará con el ORM Hibernate. Como lenguajes de programación se utilizará Java sobre el entorno de desarrollo Netbeans y como lenguaje de modelado UML en la herramienta Visual Paradigm.

⁵ WSDL (*Web Services Description Language*) da al formato XML la descripción de los servicios web precisando los métodos que pueden ser invocados, su firma y el punto de acceso (URL, puerto, etc.).

Capítulo 2: Propuesta de Solución.

2.1 Introducción.

En este capítulo se realiza la descripción de la solución propuesta, siendo necesario realizar un análisis del comportamiento del negocio para llegar a un entendimiento de las distintas acciones que el cliente desea. También se enumeran los requisitos funcionales y no funcionales esperados para el desarrollo del sistema propuesto, así como la identificación de los actores, casos de uso, descripciones de los mismos y las relaciones existentes entre ellos. Se desarrollará el flujo análisis y diseño de la metodología seleccionada.

2.2 Flujo de intercambio de ficheros con los bancos.

La notificación en línea es el proceso por el cual el banco publica la información de los pagos en ficheros XML ubicados en el servidor SFTP establecido para el correspondiente banco. Por este medio se le hace saber la información de los pagos de los vouchers ⁶que portan los ciudadanos a la Dirección de Gestión Administrativa de SAIME, quien reconocerá y validará dichas pagos, lo que representa una mejor garantía y confianza sobre la autenticidad de las transacciones de pagos.

La reversión de la notificación en línea es el proceso por el cual el banco emite un nuevo pago, sustituyendo a otro con información errónea, siguiendo el flujo de la notificación en línea.

El resumen diario es el proceso que ocurre después del cuadro que se realiza al final del día por parte del banco, una vez que esto ocurre es que se crea un fichero XML y se publica en el servidor SFTP del banco, con los datos de los pagos registrados por los clientes del SAIME. Dichos datos están compuestos por todas las notificaciones del día o los reversos correspondientes.

La fortaleza del mecanismo de seguridad establecido para garantizar la integridad de la información de los pagos que viaja a través de la red y en los vouchers que portan los ciudadanos recae sobre los códigos de seguridad que son asignados a cada sucursal bancaria, de conocimiento interno entre las partes

⁶ Comprobante que emiten los bancos recaudadores cuando un cliente efectúa un depósito en las cuentas recaudadoras de SAIME.

implicadas, para ello se genera un archivo XML con código de seguridad distinto para cada sucursal bancaria, que será publicado por el SAIME en el servidor del banco correspondiente.

Para más información del intercambio de ficheros del SIB con los bancos recaudadores ir al anexo 6.

2.3 Modelo de Dominio.

El modelo de dominio constituye un modelo conceptual que muestra gráficamente los conceptos, los atributos y las asociaciones más significativas dentro del contexto del problema, es decir, del mundo real y no de los componentes del software. De esta manera los usuarios, desarrolladores y clientes manejarán un vocabulario común, que permita compartir el conocimiento y evitar confusiones. El modelo de dominio es global y se representa a través de un diagrama UML.

Los conceptos del dominio.

- ✓ Administradores: personal encargado de ejecutar las funcionalidades de los sistemas.
- ✓ Bancos: se les denomina así a los bancos que recaudan dinero para SAIME.
- ✓ Datos de pagos: son los datos de los pagos que se realizan en las sucursales de cada banco recaudador diariamente, los cuales deberán ser almacenados en el sistema al ejecutarse las lecturas de pagos.
- ✓ Archivo de seguridad: es el archivo que se le genera a cada banco con las cadenas de seguridad para sus sucursales bancarias. Este archivo será colocado en los servidores de los bancos y a su vez es notificado de que puede usar el archivo.
- ✓ Servidor del banco: Es el servidor web que utiliza el banco recaudador para intercambiar información con el SIB.
- ✓ Lecturas de pagos: Se crean con el objetivo de almacenar los datos de los pagos que registra cada banco recaudador en un período de tiempo a una hora determinada.

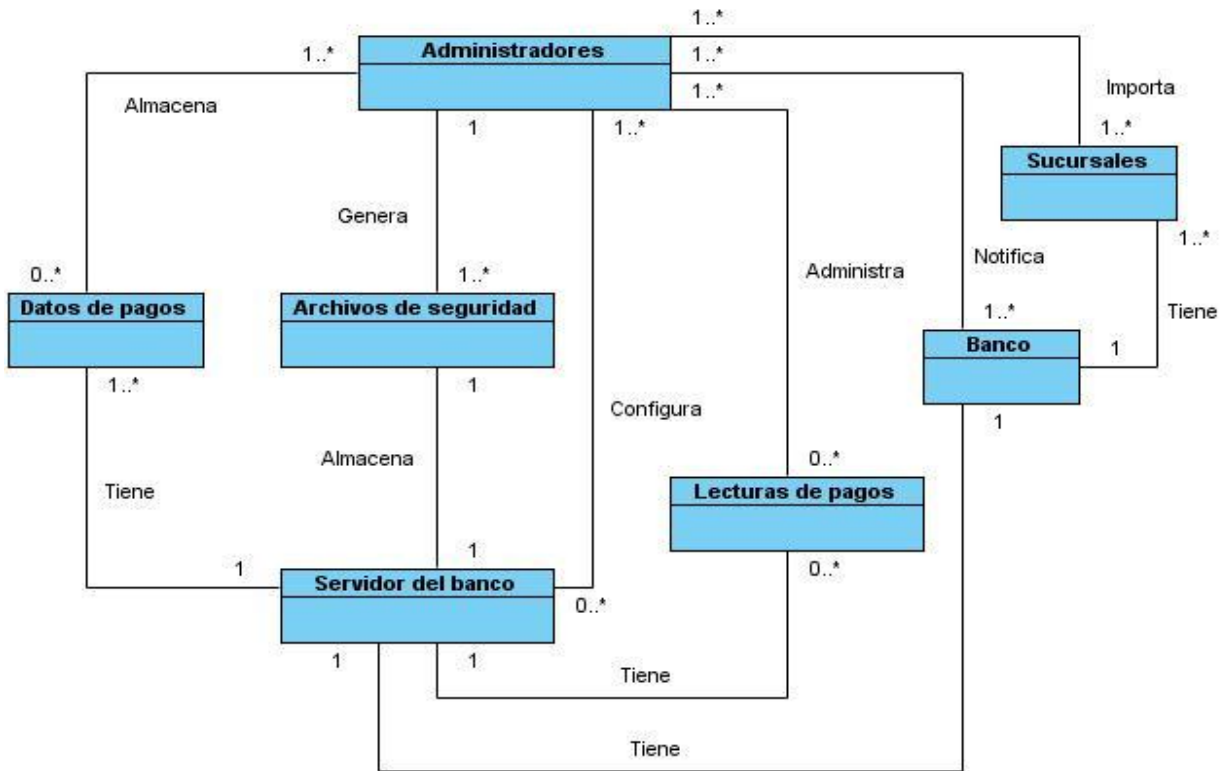


Figura. 2 Clase interfaz del análisis.

2.4 Propuesta de solución

Para lograr darle solución a la problemática planteada, se propone dos aplicaciones de escritorio que sirvan de apoyo a una aplicación web a la hora de gestionar la comunicación que existe entre los bancos recaudadores y las oficinas del SAIME, logrando la actualización de la información de los pagos. Cada aplicación está creada siguiendo la arquitectura de tres capas, diseñadas para reducir al mínimo el acoplamiento y aumentar la reutilización entre las mismas. Las aplicaciones deben permitir la autenticación de usuarios con acceso a la información requerida.

- ✓ **Gestión de comunicación bancaria:** esta aplicación es una herramienta de apoyo a la administración del proveedor de servicio, permitirá que se generen los códigos de seguridad para las sucursales de los bancos recaudadores notificando a los involucrados de los cambios producidos. Permitirá realizar la configuración de los servidores de los bancos recaudadores

para establecer la comunicación con los mismos y se importarán las sucursales de los bancos recaudadores.

- ✓ **Servicio de actualización de pagos:** esta aplicación es una herramienta configurada para gestionar los ficheros que intercambian los bancos recaudadores con el proveedor de servicios, actualiza la información de los pagos efectuados en las sucursales de los bancos de forma automática y manual, permitiendo chequear el estado de la recopilación de los datos.

2.5 Especificación de los requisitos del sistema.

Este proceso cumple un papel esencial en el proceso de desarrollo de un producto de software, se orienta a la definición de lo que se desea producir, describiendo con claridad, sin imprecisiones, en forma estable y compacta. Se define el comportamiento que deberá tener el sistema, minimizando la posibilidad de errores que puedan ocurrir relacionados al desarrollo del mismo, pues se tiene especificado de forma clara lo que el cliente desea. Los requisitos se dividen en dos grupos: requisitos funcionales y requisitos no funcionales.

2.5.1 Requisitos funcionales.

Los requisitos funcionales son capacidades o condiciones que el sistema debe ser capaz de cumplir. Establecen los comportamientos y describen las transformaciones que el mismo debe realizar para obtener los resultados deseados por los programadores. Los requisitos funcionales definidos se pueden ver a continuación:

RF - 1. Configurar dirección SFTP.

- 1.1. Mostrar el listado de los bancos y si tienen dirección configurada o no.
- 1.2. Seleccionar un banco para la configuración.
- 1.3. Entrar los datos:
 - IP o nombre del servidor
 - Puerto
 - Usuario
 - Contraseña
- 1.4. Probar la conexión bancaria.

Capítulo 2: Propuesta de solución

1.5. Informar al usuario de cualquier eventualidad.

1.6. Guardar los datos en la configuración del banco seleccionado.

RF - 2. Generar códigos de seguridad para los bancos.

2.1. Seleccionar fecha de comienzo de uso del código seguridad.

2.2. Se generan los códigos de seguridad para todos los bancos recaudadores existentes.

2.3 Se generan los códigos de seguridad para todas las oficinas del SAIME.

2.3.1. Escribir los códigos en el fichero del banco.

2.4. Guardar los códigos en el SFTP configurado para el banco.

2.4.1 Notificar al banco que tiene el fichero de código de seguridad en el SFTP.

2.4.2 Si no existe conexión con el SFTP se realiza una copia local para enviarlo en otro momento.

2.5 Se notifica al banco según lo sucedido en el punto anterior.

RF - 3. Administrar servicio de actualización de depósitos.

3.1. Iniciar servicio de actualización de depósitos.

3.2. Reiniciar servicio de actualización de depósitos.

3.3 Detener servicio de actualización de depósitos.

3.4. Conocer el estado del servicio de actualización de depósitos.

RF - 4. Administrar lecturas de depósito de pagos.

4.1. Crear nueva lectura.

4.1.1. Aceptar los datos de la nueva lectura:

- Nombre de la lectura
- Fecha inicial
- Fecha final
- Hora de la lectura
- Periodicidad

4.1.2. Guardar los datos de la lectura creada.

4.2. Modificar las lecturas existentes.

4.2.1. Listar los servidores configurados en el sistema.

Capítulo 2: Propuesta de solución

4.2.2. Seleccionar un servidor.

4.2.3. Listar las lecturas programadas para el servidor.

4.2.4. Seleccionar la lectura a modificar.

4.2.5. Validar los cambios efectuados.

4.2.5.1. Verificar que no se esté ejecutando en el horario de la modificación.

4.2.6. Guardar la lectura modificada.

4.3. Habilitar/Deshabilitar lecturas existentes.

4.3.1. Listar los servidores configurados en el sistema.

4.3.2. Seleccionar un servidor.

4.3.1. Listar las lecturas programadas para el servidor.

4.3.2. Deshabilitar/Habilitar una lectura.

4.3.3. Guardar los cambios.

4.4. Eliminar lecturas.

4.4.1. Listar los servidores configurados en el sistema.

4.4.2. Seleccionar un servidor.

4.4.1. Listar las lecturas programadas para el servidor.

4.4.2. Seleccionar lectura a eliminar.

4.4.3. Eliminar lectura seleccionada.

4.4.3.1. Verificar que de las que se intentan eliminar no haya alguna ejecutándose.

RF - 5. Ejecutar automáticamente lecturas configuradas.

5.1. Efectuar ejecución de lectura configurada en el horario y fecha previstos.

5.1.1. Verificar lecturas pendientes (que no se pudieron ejecutar) cuando se inicie la operación.

5.2. Notificar de cualquier eventualidad.

5.2.1. Ejecutar lectura sobre archivos pendientes (que no se pudieron ejecutar).

5.2.2. Ejecutar lectura diaria.

RF - 6. Ejecutar lectura manual.

6.1. Listar los servidores configurados en el sistema.

- 6.2. Seleccionar un servidor.
- 6.1. Listar lecturas existentes.
- 6.2. Seleccionar lectura.
- 6.3. Ejecutar lectura.

RF - 7. Importar sucursales al banco recaudador.

- 6.1. Listar los bancos recaudadores.
- 6.2. Seleccionar un servidor.
- 6.3. Importar el fichero con los datos de las sucursales.

2.5.2 Requisitos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son las características que hacen al producto atractivo, usable, rápido o confiable. (11) En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales.

Requisitos de software.

- La aplicación debe ser soportada en todas las plataformas de *Linux* y *Microsoft Windows*.
- Instalación de la máquina virtual de Java.

Requisitos de apariencia o interfaz externa.

- La aplicación debe cumplir con las pautas definidas por el centro de identidad.
- La corrección de errores en la introducción de datos será clara y fácil de realizar. La entrada de datos incorrecta será detectada y notificada al usuario.
- Todos los textos y mensajes en pantalla aparecerán en idioma español.
- La apariencia del producto será sencilla, cómoda, amigable y de fácil navegación para cualquier tipo de usuario.
- El sistema permitirá al usuario distinguir visualmente los datos en las ventanas, a través del uso de colores, tamaño de las fuentes e íconos.

Restricciones en el diseño y la implementación.

- El sistema debe implementarse usando el lenguaje Java.
- El sistema gestor de bases de datos será PostgreSQL 8.4.
- El sistema debe desarrollarse usando el IDE NetBeans 6.9.
- Visual Paradigm Enterprise Edition 6.9.

Requisitos de seguridad.

- El sistema garantizará el tratamiento de excepciones.
- La conexión a los servidores de bases de datos y de aplicaciones es requerida para el correcto funcionamiento.
- Encriptación de contraseña.
- Los usuarios se autenticarán para poder tener acceso a la aplicación. El sistema debe permitir solo el acceso a cada uno de sus servicios al personal requerido para esto.
- Integridad: la información manejada por el sistema será objeto de cuidadosa protección contra la corrupción de datos.

Soporte.

- Se requiere que el producto reciba mantenimiento y configuración ante cualquier fallo que ocurra durante el período de prueba.
- Es necesaria una correcta documentación para el uso o desarrollo de las funcionalidades de la herramienta.
- Integrar la solución desarrollada con sistemas paralelos desarrollados por los bancos recaudadores de SAIME.

2.6 Modelo del sistema.

2.6.1 Diagrama de Casos de Uso del Sistema.

Es un diagrama que permite visualizar de una forma sencilla las funcionalidades que deberá tener el sistema propuesto, en él se refleja cómo interactúan usuarios externos al sistema (actores) y las

Capítulo 2: Propuesta de solución

funcionalidades del sistema (casos de uso). El modelado de casos de uso es una técnica efectiva y a la vez simple para modelar los requisitos del sistema desde la perspectiva del usuario. El diagrama de casos de uso de cada aplicación que se quiere implementar se muestra en las figuras 3 y 4.

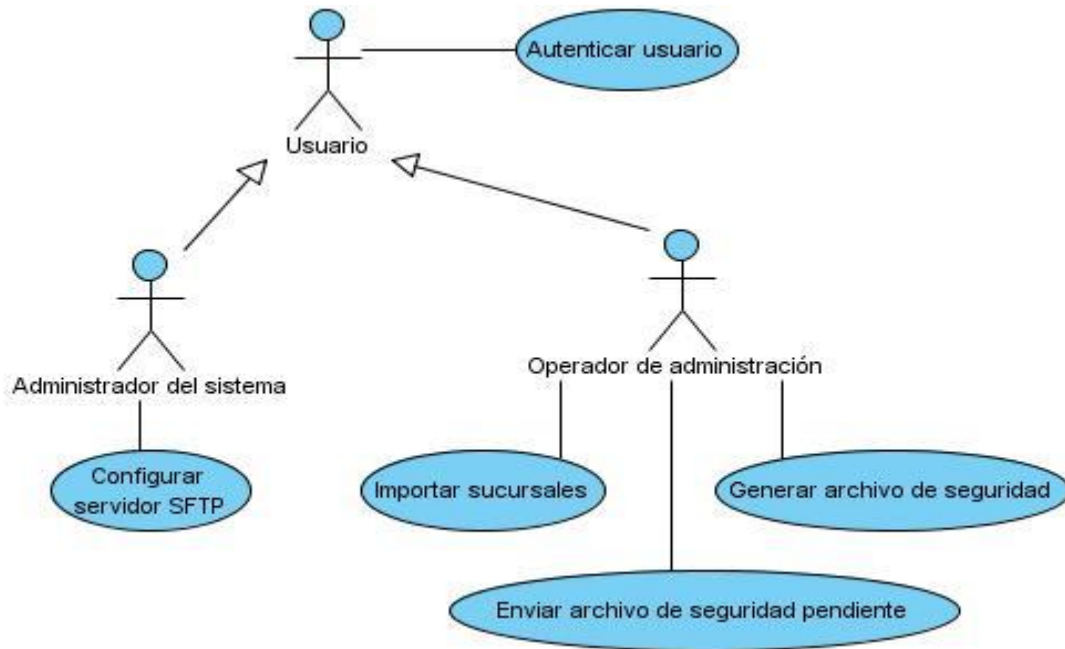


Figura. 3 Diagrama de casos de uso de la aplicación Gestión de comunicación bancaria.

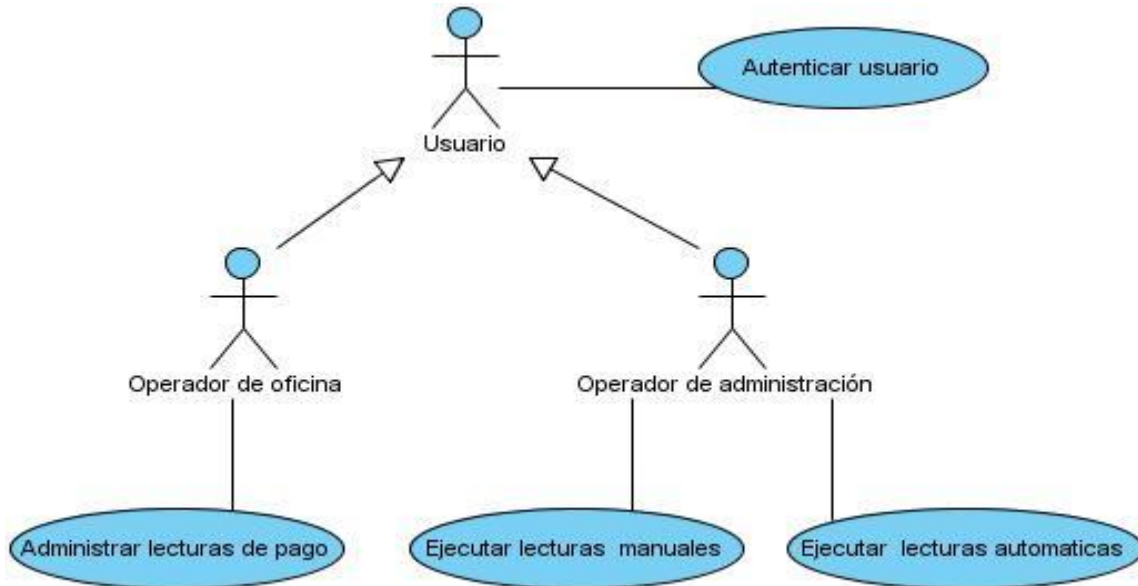


Figura. 4 Diagrama de casos de uso de la aplicación Servicio de actualización de pagos.

2.6.2 Descripción de los actores del sistema

Los actores de un sistema son agentes externos: personas u otros sistemas que interactúan con el sistema. En la tabla 1 se encuentran descritos los actores de los sistemas.

Actor	Descripción
Usuario	Cualquier usuario que desee registrarse en el sistema.
Operador de administración	Persona encargada de ejecutar las lecturas de pagos, generar y enviar los archivos de seguridad a los bancos recaudadores e importar las sucursales de los mismos.
Operador de oficina	Persona encargada de administrar las lecturas de pagos.
Administrador del sistema	Persona encargada de configurar la conexión con los servidores ftp de los bancos recaudadores.

Tabla. 1 Descripción de los actores del sistema.

2.6.3 Descripción de los casos de uso del sistema.

Para entender la funcionalidad asociada a los casos de uso, no es suficiente con la representación gráfica del Diagrama de casos de uso. La descripción de los casos de usos describe paso a paso, qué es lo que el sistema debe hacer, separando las responsabilidades del sistema y la de los actores. La descripción de los casos de usos de ambas aplicaciones se puede ver en el anexo 1, en la tabla 2 se muestra la descripción del caso de uso Administrar lecturas de pagos perteneciente a la aplicación Servicio de actualización de pagos.

Caso de Uso:	Administrar Lecturas de Depósito de Pagos
Actores:	Operador de oficina.
Resumen:	El caso de uso inicia cuando el Operador de oficina ejecuta la opción Administrar Lecturas de Pagos donde se crean, modifican, habilitan/deshabilitan y eliminan las lecturas, finalizando así el caso de uso.
Precondiciones:	El Operador de oficina debe estar autenticado en el sistema y el servidor del banco debe estar creado.
Referencias	RF30
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El Operador de oficina selecciona la opción Administrar Lectura de pagos.	2-El sistema muestra un menú con las opciones: <ul style="list-style-type: none">• Crear Lectura• Modificar Lectura.• Habilitar Lectura.• Eliminar Lectura.
3- El Operador de oficina selecciona la opción: 3a) Crear Lectura, ir a Sección 1 “Crear Lectura” . 3b) Modificar Lectura, ir a Sección 2 “Modificar Lectura” . 3c) Habilitar/ Deshabilitar Lectura, ir a Sección 3 “Habilitar Lectura” .	

Capítulo 2: Propuesta de solución

3e) Eliminar Lectura, ir a Sección 4 “Eliminar Lectura” .	
Sección 1 “Crear Lectura”	
	<p>1-El sistema permite:</p> <p>1a) Ver listado de servidores configurados en los bancos con los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre del banco. • Código del banco. • Ip del servidor. <p>1b) Seleccionar el servidor que desea crearle la lectura.</p>
2-El Operador de oficina selecciona al servidor que desea crearle la lectura.	<p>3-El sistema muestra los siguientes campos a llenar.</p> <ul style="list-style-type: none"> • Nombre lectura. • Fecha inicial. • Fecha final. • Hora lectura. • Periodicidad. • Habilitar lectura <p>Y las opciones:</p> <ul style="list-style-type: none"> • Crear • Cerrar
4-El Operador de oficina introduce los datos y selecciona la opción: 4a) Aceptar , continúa en la Sección 1 “Crear Lectura” parte 5. 4b) Cerrar , ir al Flujo Alternativo 1 “Cerrar Acción” .	5- El sistema valida que no haya campos vacíos sino ir al Flujo Alternativo 2 “Campos Vacíos Crear Lectura” .
	6- El sistema valida que los datos estén correctos, si no ir al Flujo Alternativo 3 “Datos Incorrectos Crear Lectura”
	7- El sistema guarda los datos en la base de datos y muestra un mensaje satisfactorio. En caso contrario, ver Flujo Alternativo 8 “Error de Conexión” .
Prototipo Interfaz	

Capítulo 2: Propuesta de solución

Seleccionar servidor			Crear Lectura	
Código	Nombre banco	IP servidor	Nombre lectura:	<input style="width: 80%;" type="text"/>
1100	Banco Exterior	10.13.35.120	Fecha inicial:	<input style="width: 60%;" type="text" value="4/06/11"/> <input style="width: 20%;" type="button" value="■"/>
1101	Banfoandes	10.13.35.120	Fecha final:	<input style="width: 60%;" type="text" value="4/06/11"/> <input style="width: 20%;" type="button" value="■"/>
1102	Banco del Tesoro	10.13.35.120	Hora lectura:	<input style="width: 60%;" type="text" value="03:26:59 PM"/> <input style="width: 20%;" type="button" value="▲▼"/>
1103	Banco de Venezuela	10.13.35.120	Periodicidad:	<input style="width: 80%;" type="text"/>
1105	Bancaribe	10.13.35.120	<input type="checkbox"/> Habilitar lectura	
1106	Banco Caroni	10.13.35.120	<input type="button" value="Crear"/> <input type="button" value="Cerrar"/>	
1107	Banco Provincial	10.13.35.120		
1108	CorpBanca	10.13.35.120		
1109	Banco Plaza	10.13.35.120		
1110	Banco Banesco	10.13.35.120		
1104	Banco Fondo Común	10.13.35.120		

Sección 2 “Modificar Lectura”

	<p>1-El sistema permite:</p> <p style="margin-left: 20px;">1a) Ver listado de servidores configurados en los bancos con los siguientes datos:</p> <ul style="list-style-type: none"> Nombre del banco. Código del banco. Ip del servidor. <p style="margin-left: 20px;">1b) Seleccionar el servidor que desea modificarle la lectura.</p>
2-El Operador de oficina selecciona al servidor que desea modificarle la lectura.	<p>3- El sistema permite:</p> <p style="margin-left: 20px;">3a) Ver listado de lecturas programadas del servidor seleccionado.</p> <p style="margin-left: 20px;">3b) Seleccionar la lectura programada a modificar del servidor seleccionado.</p>
4-El Operador de oficina selecciona la lectura que desea modificar.	<p>5-El sistema muestra una interfaz con los siguientes campos a modificar.</p> <ul style="list-style-type: none"> Nombre lectura. Fecha inicial. Fecha final. Hora lectura. Periodicidad. Habilitar lectura.

Capítulo 2: Propuesta de solución

	Y las opciones: <ul style="list-style-type: none">• Modificar.• Cerrar.
6-El Operador de oficina modifica los datos deseados y selecciona la opción: 6a) Modificar , continúa en la Sección 2“Modificar Lectura” parte 7. 6b) Cerrar , ir al Flujo Alterno 1 “Cerrar Acción” .	7- El sistema valida que no haya campos vacíos, si no ir al Flujo Alterno 4 “Campos Vacíos Modificar Lectura” .
	8- El sistema valida que los datos estén correctos, si no ir al Flujo Alterno 5 “Datos Incorrectos Modificar Lectura” .
	9 El sistema valido que no esté ejecutándose la lectura en ese momento de modificación, si no ir al Flujo Alterno 6 “Lectura Ejecutándose Modificar Lectura” .
	10 - El sistema guarda los cambios efectuados en la base de datos y muestra un mensaje satisfactorio. En caso contrario, ver Flujo Alterno 9 “Error de Conexión Modificar Lectura” .
Prototipo Interfaz	

Capítulo 2: Propuesta de solución

Seleccionar servidor

Código	Nombre banco	IP servidor
1100	Banco Exterior	10.13.35.120
1101	Banfoandes	10.13.35.120
1102	Banco del Tesoro	10.13.35.120
1103	Banco de Venezuela	10.13.35.120
1105	Bancaribe	10.13.35.120
1106	Banco Caroni	10.13.35.120
1107	Banco Provincial	10.13.35.120
1108	CorpBanca	10.13.35.120
1109	Banco Plaza	10.13.35.120
1110	Banco Banesco	10.13.35.120
1104	Banco Fondo Común	10.13.35.120

Seleccionar lectura a modificar

Nombre	Hora	Habilitada
Por la tarde	10:34:47	Habilitada
Lectura	23:32:02	Habilitada
De noche	10:34:47	Habilitada

Datos de la Lectura

Nombre lectura:

Fecha inicial:

Fecha final:

Hora de la lectura:

Periodicidad:

Habilitar lectura

Sección 3 “Habilitar Lectura”

	<p>1-El sistema permite:</p> <p style="margin-left: 20px;">1a) Ver listado de servidores configurados en los bancos con los siguientes datos:</p> <ul style="list-style-type: none"> Nombre del banco. Código del banco. Ip del servidor. <p style="margin-left: 20px;">1b) Seleccionar el servidor que desea cambiarle el estado de la lectura.</p>
<p>2-El Operador de oficina selecciona el servidor que desea cambiarle el estado de la lectura.</p>	<p>3-El sistema permite:</p> <p style="margin-left: 20px;">3a) Ver listado de lecturas programadas en el servidor seleccionado.</p>

Capítulo 2: Propuesta de solución

	3b) Seleccionar la lectura que desea cambiarle el estado. Y las opciones: <ul style="list-style-type: none">• Cambiar.• Cerrar.
4-El Operador de oficina selecciona la lectura deseada y selecciona la opción: 4a) Cambiar, continúa en la Sección 3 “Habilitar Lectura” parte 5. 4b) Cerrar, ver Flujo Alternativo 1 “Cerrar Acción” .	5- El sistema guarda los cambios efectuados en la base de datos y muestra un mensaje satisfactorio. En caso contrario, ver Flujo Alternativo 10 “Error de Conexión Habilitar Lectura” .

Prototipo Interfaz

El prototipo de interfaz de usuario se divide en dos paneles principales:

- Selección de servidor:** Una tabla con tres columnas: Código, Nombre banco y IP servidor. La fila con el código 1100 y el nombre 'Banco Exterior' está resaltada.
- Selección de lectura a habilitar o deshabilitar:** Una tabla con tres columnas: Nombre, Hora y Habilitada. Las filas muestran lecturas como 'Lectura', 'De noche' y 'Por la tarde' con sus respectivos horarios y estados de habilitación.

En la parte inferior derecha del prototipo, se encuentran dos botones: 'Cambiar' y 'Cerrar'.

Sección 4 “Eliminar Lectura”

	1-El sistema permite: 1a) Ver listado de servidores configurados en los bancos con los siguientes datos: <ul style="list-style-type: none">• Nombre del banco.• Código del banco.• Ip del servidor. 1b) Seleccionar el servidor que desea eliminarle la lectura.
--	--

Capítulo 2: Propuesta de solución

<p>2-El Operador de oficina selecciona el servidor que desea eliminarle la lectura.</p>	<p>3-El sistema permite:</p> <p>3a) Ver listado de lecturas programadas en el servidor seleccionado.</p> <p>3b) Seleccionar la lectura que desea eliminar.</p> <p>Y las opciones:</p> <ul style="list-style-type: none"> • Eliminar. • Cerrar.
<p>4-El Operador de oficina selecciona la lectura a eliminar y selecciona la opción:</p> <p>4a) Eliminar, continúa en la Sección 4 “Eliminar Lectura” parte 5.</p> <p>4b) Cerrar, ver Flujo Alternativo 1 “Cerrar Acción”.</p>	<p>5-El sistema verifica que las lecturas seleccionadas no estén ejecutándose sino ir al Flujo Alternativo 7 “Lectura Ejecutándose Eliminar Lectura”.</p>
	<p>6-El sistema guarda los cambios efectuados en la base de datos y muestra un mensaje satisfactorio. En caso contrario, ver Flujo Alternativo 11 “Error de Conexión Eliminar Lectura”.</p>

Prototipo de Interfaz

El prototipo de interfaz muestra dos paneles de selección de datos:

Seleccionar servidor

Código	Nombre banco	IP servidor
1100	Banco Exterior	10.13.35.120
1101	Banfoandes	10.13.35.120
1102	Banco del Tesoro	10.13.35.120
1103	Banco de Venezuela	10.13.35.120
1105	Bancaribe	10.13.35.120
1106	Banco Caroni	10.13.35.120
1107	Banco Provincial	10.13.35.120
1108	CorpBanca	10.13.35.120
1109	Banco Plaza	10.13.35.120
1110	Banco Banesco	10.13.35.120
1104	Banco Fondo Común	10.13.35.120

Seleccionar lectura a Eliminar

Nombre	Hora	Habilitada
De noche	10:34:47	Habilitada
Lectura	23:32:02	Habilitada
Por la tarde	10:34:47	Habilitada

Flujos Alternos

Flujo Alternativo 1 “Cerrar Acción”

Capítulo 2: Propuesta de solución

Acción del Actor	Respuesta del Sistema
	1-Se reinician los valores de los campos y se muestra la interfaz principal.
Flujo Alternativo 2 “Campos Vacíos Crear Lectura”	
Acción del Actor	Respuesta del Sistema
	1-El sistema muestra un mensaje notificando al usuario que ha dejado campos vacíos, volviendo a la Sección 1 “Crear Lectura” parte 3.
Flujo Alternativo 3 “Datos Incorrectos Crear Lectura”	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje notificando al usuario que no ha llenado los campos correctamente, volviendo a la Sección 1 “Crear Lectura” parte 3.
Flujo Alternativo 4 “Campos Vacíos Modificar Lectura”	
Acción del Actor	Respuesta del Sistema
	1-El sistema muestra un mensaje notificando al usuario que ha dejado campos vacíos, volviendo a la Sección 2 “Modificar Lectura” parte 5.
Flujo Alternativo 5 “Datos Incorrectos Modificar Lectura”	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje notificando al usuario que no ha llenado los campos correctamente, volviendo a la Sección 2 “Modificar Lectura” parte 5.
Flujo Alternativo 6 “Lectura Ejecutándose Modificar Lectura”	
	1-El sistema muestra un mensaje que no se puede modificar la lectura ya que está ejecutándose en ese momento volviendo a la Sección 2 “Modificar Lectura” parte 5.
Flujo Alternativo 7 “Lectura Ejecutándose Eliminar Lectura”	
	1-El sistema muestra un mensaje que no se puede eliminar la lectura ya que está ejecutándose en ese momento, volviendo a la Sección 4 “Eliminar Lectura” parte 3.
Flujo Alternativo 8 “Error de Conexión Crear Lectura”	

Capítulo 2: Propuesta de solución

	1-El sistema muestra un mensaje diciendo que no existe conexión a la Base de datos, volviendo a la Sección 4 “Crear Lectura” parte 3.
Flujo Alterno 9 “Error de Conexión Modificar Lectura”	
	1-El sistema muestra un mensaje diciendo que no existe conexión a la Base de datos, volviendo a la Sección 4 “Modificar Lectura” parte 5.
Flujo Alterno 10 “Error de Conexión Habilitar Lectura”	
	1-El sistema muestra un mensaje diciendo que no existe conexión a la Base de datos, volviendo a la Sección 4 “Habilitar Lectura” parte 3.
Flujo Alterno 11 “Error de Conexión Eliminar Lectura”	
	1-El sistema muestra un mensaje diciendo que no existe conexión a la Base de datos en ese momento, volviendo a la Sección 4 “Eliminar Lectura” parte 3.
Poscondiciones	Quedan administradas las lecturas.

Tabla. 2 Descripción del caso de uso Administrar lecturas de pagos.

2.7 Análisis.

El análisis permite obtener una visión de qué es lo que debe hacer el software, para cumplir con las necesidades y expectativas de los clientes y usuarios finales, a través de los requisitos funcionales del mismo. Es utilizado fundamentalmente por los desarrolladores para comprender cómo debería darse forma al sistema, es decir, sirve como una abstracción del modelo de diseño y se centra en los requerimientos funcionales.

2.7.1 Diagrama de clases del análisis.

El diagrama de clases del análisis está estructurado por clases y paquetes estereotipados que proporcionan la estructura de la vista interna del sistema. Es utilizado fundamentalmente por los desarrolladores para comprender cómo debería darse forma al sistema, es decir, cómo debería ser diseñado e implementado. Para representar las clases en el análisis se utilizan los siguientes estereotipos:

Clase Interfaz: modela la interfaz del sistema y su interacción con el actor a través de ventanas, formularios, o la comunicación con otros sistemas.

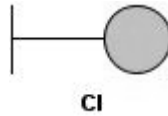


Figura. 5 Clase interfaz del análisis.

Clase Control: las clases de control representan coordinación, secuencia, transacciones, y control de otros objetos, coordinan los eventos necesarios para la realización o especificación del caso de uso, con otras palabras, son las que ejecutan el caso de uso. Usualmente son dependientes de la aplicación, además de tener un control sobre todas las acciones a realizar.



Figura. 6 Clase control del análisis.

Clase Entidad: modelan el comportamiento asociado a la información y los datos del sistema que tienen larga vida, por lo que son persistentes.

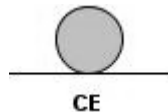


Figura. 7 Clase entidad del análisis.

Los diagramas de clases del análisis se encuentran en el anexo 3.

2.8 Diseño.

El diseño es el refinamiento del análisis, ocupándose de describir cómo el sistema debe cumplir con los requisitos funcionales establecidos, además de tener en cuenta los requisitos no funcionales, centrándose en los principales propósitos del diseño, transformar los requerimientos en un diseño de cómo el sistema

debe ser. El modelo de diseño está muy cercano al de implementación, por lo que es importante mantenerlo a través del ciclo de vida completo del software.

2.8.1 Descripción de la arquitectura.

Las aplicaciones están diseñadas mediante la implementación del patrón de arquitectura en tres capas. El principal objetivo que persigue la arquitectura en tres capas es reducir dependencias entre artefactos, situándolos en capas lógicas, donde cada capa depende del servicio prestado por la inferior y presta un servicio a la superior, proporcionando a los desarrolladores ventajas en cuanto al mantenimiento y reutilización de componentes, de manera tal que una capa puede ser totalmente remplazada sin afectar el resto del sistema. La aplicación está estructurada en:

Capa de presentación:

Esta es la capa encargada de la interacción con los diferentes tipos de usuarios y contiene las interfaces visuales de la aplicación, modela la forma en que se mostrarán y recogerán los datos de los usuarios, haciéndose la mayoría de las validaciones de entradas. Se comunica con la capa de lógica de negocio a la cual envía todas las solicitudes de los usuarios y recibe la respuesta después de que hayan sido procesadas cada una de estas solicitudes, siendo esta operación transparente para el usuario. Ninguna otra capa tiene acceso a ella.

Capa de negocio:

Es la capa encargada de modelar la lógica de negocio que dará solución a cada uno de los casos de usos de la aplicación, es donde se establecen las reglas o restricciones que debe cumplir la aplicación. La capa de negocio es el puente entre la capa de presentación y la capa de acceso a datos, por lo que ofrece los servicios que necesita la capa de presentación para interactuar con los usuarios y verifica que los datos estén correctos antes de relacionarse con la base de datos, logrando así una mayor integridad en la aplicación.

Capa de acceso a datos:

La capa de acceso a datos se encarga de acceder y manipular los datos persistentes de un sistema, considerándose la capa más crítica y sensible de la arquitectura. Recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

2.8.2 Patrones de diseño.

Los patrones de diseño ofrecen esquemas para definir estructuras de diseño con las que construye sistemas de software orientada a objetos. Permite formalizar un vocabulario común entre diseñadores y estandarizar el modo en que se realiza el diseño, así como la reiteración en la búsqueda de soluciones. Los patrones del diseño utilizados en el desarrollo de los componentes son los Patrones de Software para la Asignación General de Responsabilidad (GRASP) y patrones *Gang of Four* (GOF).

Patrones GRASP

Experto: permite asignar a una clase la información necesaria para cumplir su responsabilidad, de esta forma se asignan las mismas de forma adecuada y el módulo tiende a ser más fácil de entender, mantener y ampliar, pensando en una futura reutilización de componentes. Entre sus beneficios está el hecho de que permite conservar el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Un ejemplo sería a clase ASBCLecturaNegocio, la cual es responsable de realizar todas las operaciones acerca de las lecturas.

Creador: guía la asignación de responsabilidades de la creación de objetos. La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. Un ejemplo sería la clase CLectura que tiene una instancia de la clase ASBCLecturaNegocio.

Bajo acoplamiento: ayuda a aumentar la reutilización y disminuye la dependencia entre las clases. Uno de los principios para proteger al software frente al cambio es mantener bajo el acoplamiento entre clases, cuanto menor sea el acoplamiento entre clases, menor influencia tendrán los cambios. Como ejemplo se tiene la clase ASBCLecturaNegocio que recurre a las interfaces ASBILecturaDao y ASBIPagoDao, en las que se apoya al realizar sus funcionalidades.

Alta cohesión: al asignar responsabilidades en el diseño, se buscan soluciones que asignen la información que almacena una clase de forma coherente, completa y relacionada (relacionada con la clase). De esta forma, se obtienen clases cohesionadas. Este patrón se relaciona con el bajo acoplamiento debido a que si se tiene un diseño cohesionado existirá un bajo acoplamiento entre clases.

Se tiene como ejemplo la clase ASBILecturaDaolmp, la cual tendrá la implementación de todas las funcionalidades que se relacionen con las lecturas.

Controlador: ayuda a que la lógica del negocio esté separada de la capa de presentación, pues es el que recibe los datos de los usuarios y los envía a distintas clases en dependencia del método que fue llamado. Un ejemplo es la clase CGenerarCodigoSeguridad de la aplicación Gestión de comunicación bancaria.

Patron GoF.

Patrón Fachada (Facade): se empleó para proveer de una interfaz unificada y sencilla, que haga de intermediaria entre el cliente y la interfaz o grupo de interfaces más complejas, permitiendo así una mayor flexibilidad en el desarrollo del sistema. Permite reducir la complejidad y minimizar las dependencias.

2.8.3 Diagrama de clases del diseño.

Los diagramas de clases del diseño son los más útiles para ilustrar las relaciones entre las clases e interfaces en la implementación del sistema. El diagrama de clases del diseño se puede ver en el Anexo 4.

2.8.4 Diagramas de Interacción (Secuencia).

Los diagramas de interacción ilustran el flujo de interacciones entre objetos en un sistema. Estos diagramas de interacción son normalmente diagramas de secuencia y diagramas de colaboración, que permiten expresar fácilmente el comportamiento de los casos de uso en función de objetos que colaboran entre sí para realizar una operación. Estos diagramas contienen objetos, enlaces y mensajes. Los diagramas de secuencia del diseño se pueden ver en el anexo 5.

2.8.5 Modelo Físico de Datos.

El modelo físico de datos suele apoyarse en diagramas a través de los cuales se tratan de visualizar las diferentes entidades que intervienen, el tipo de datos y la forma en que se relacionan en una base de datos. A continuación el modelo físico de datos de los sistemas se encuentran en la figura 8.

Capítulo 2: Propuesta de Solución

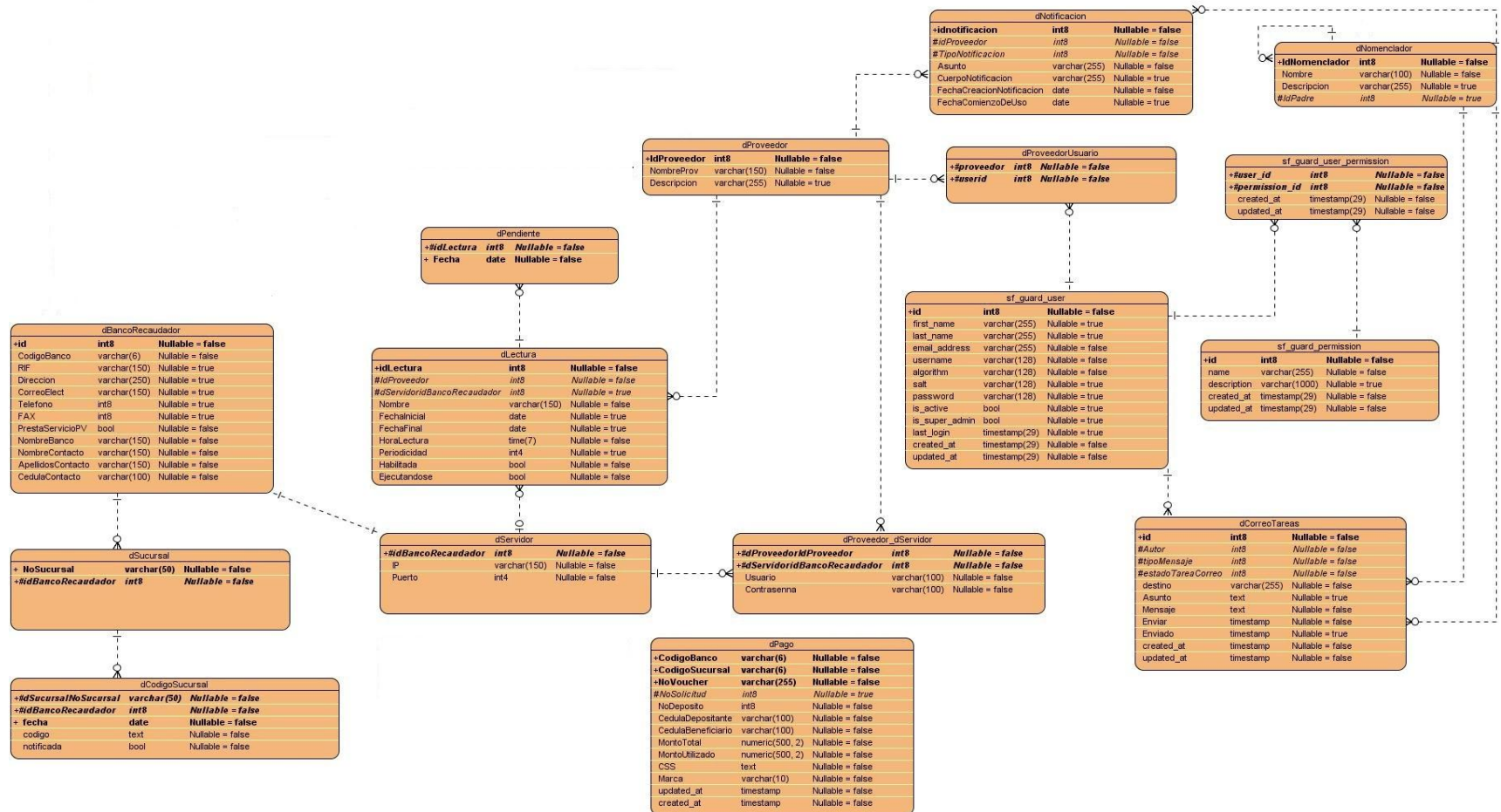


Figura. 8 Diagrama entidad relación.

Descripción del modelo de datos.

Tablas	Descripción
<i>sf_guard_user</i>	Guarda todos los usuarios del sistema.
<i>sf_guard_permission</i>	Guarda los permisos de que tendrán los usuarios.
<i>sf_guard_user_permission</i>	Guarda los datos de la relación de las tablas <i>Sf_guard_user</i> y <i>sf_guard_permission</i> producto de la relación de muchos a muchos que existe entre ellas.
<i>dCorreoTareas</i>	Guarda todos los correos que serán enviados.
<i>dNomenclador</i>	Guarda todos los campos multievaluados del sistema.
<i>dProveedorUsuario</i>	Guarda los datos de la relación de las tablas <i>Sf_guard_user</i> y <i>dProveedor</i> producto de la relación de muchos a muchos que existe entre ellas.
<i>dNotificacion</i>	Guarda los datos de las notificaciones que envía el sistema.
<i>dBancoRecaudador</i>	Guarda los datos de los bancos recaudadores.
<i>dPago</i>	Guarda los datos de los pagos realizados por los clientes.
<i>dSucursal</i>	Guarda los datos de las sucursales de los bancos.
<i>dCodigoSucursal</i>	Guarda los códigos que se generan para cada sucursal.
<i>dPendiente</i>	Guarda los datos de las lecturas que no se pudieron ejecutar.
<i>dLectura</i>	Guarda los datos de un grupo de lecturas asociada a un servidor.
<i>dServidor</i>	Guarda los datos de los servidores.
<i>dProveedor_dServidor</i>	Guarda los datos que permiten acceder a un proveedor a un servidor.
<i>dProveedor</i>	Guarda los datos de los proveedores.

Tabla. 3 Clase interfaz del análisis.

2.9 Conclusiones parciales.

En este capítulo se han definido los conceptos del modelo de dominio, se describen los requisitos funcionales y no funcionales del sistema propuesto, se elabora el diagrama de casos de uso del sistema, descripciones de los casos de usos y actores del sistema. De acuerdo a la realización del diagrama de clases del análisis y diseño, y los diagramas de secuencia del diseño se elaboró la propuesta de solución.

Capítulo 2: Propuesta de solución

Se describe la arquitectura en tres capas y se describen los patrones de diseño GRASP y GoF. Todo esto proporciona las bases para la implementación del sistema propuesto.

Capítulo 3: Implementación y Prueba.

3.1 Introducción.

En el presente capítulo se describe la implementación del software a través de los diagramas de componentes, así como definir los estándares de codificación a utilizar por el programador. Se necesita realizar la distribución de los nodos necesarios para el despliegue de la aplicación así como la realización de pruebas para obtener una mayor seguridad del sistema.

3.2 Diagrama de Despliegue.

El diagrama de despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos, donde cada hardware se representa como un nodo. Las conexiones o relaciones de los nodos se etiquetan con un estereotipo que identifica el protocolo de comunicación utilizado. Este modelo es utilizado como base para la realización de las actividades de diseño e implementación.

El despliegue del sistema estará ubicado en tres áreas, una en las oficinas del SAIME, otra en el Centro de Datos del SAIME y una auxiliar en los bancos recaudadores. En las oficinas se encuentran las aplicaciones, donde se controla el acceso y se gestiona la información. En el Centro de Datos del SAIME existirá un servidor de Base de Datos que almacenará la información de las aplicaciones, un servidor de aplicaciones que ofrecerá servicios web y un servidor SFTP. Debido a restricciones de algunos bancos que no puedan disponer de un servidor SFTP, el servidor SFTP ubicado en el Centro de Datos del SAIME prestará servicios a los mismos. Además existirá una VPN para la comunicación entre las áreas. En la figura 9 se encuentra el diagrama de despliegue de las aplicaciones.

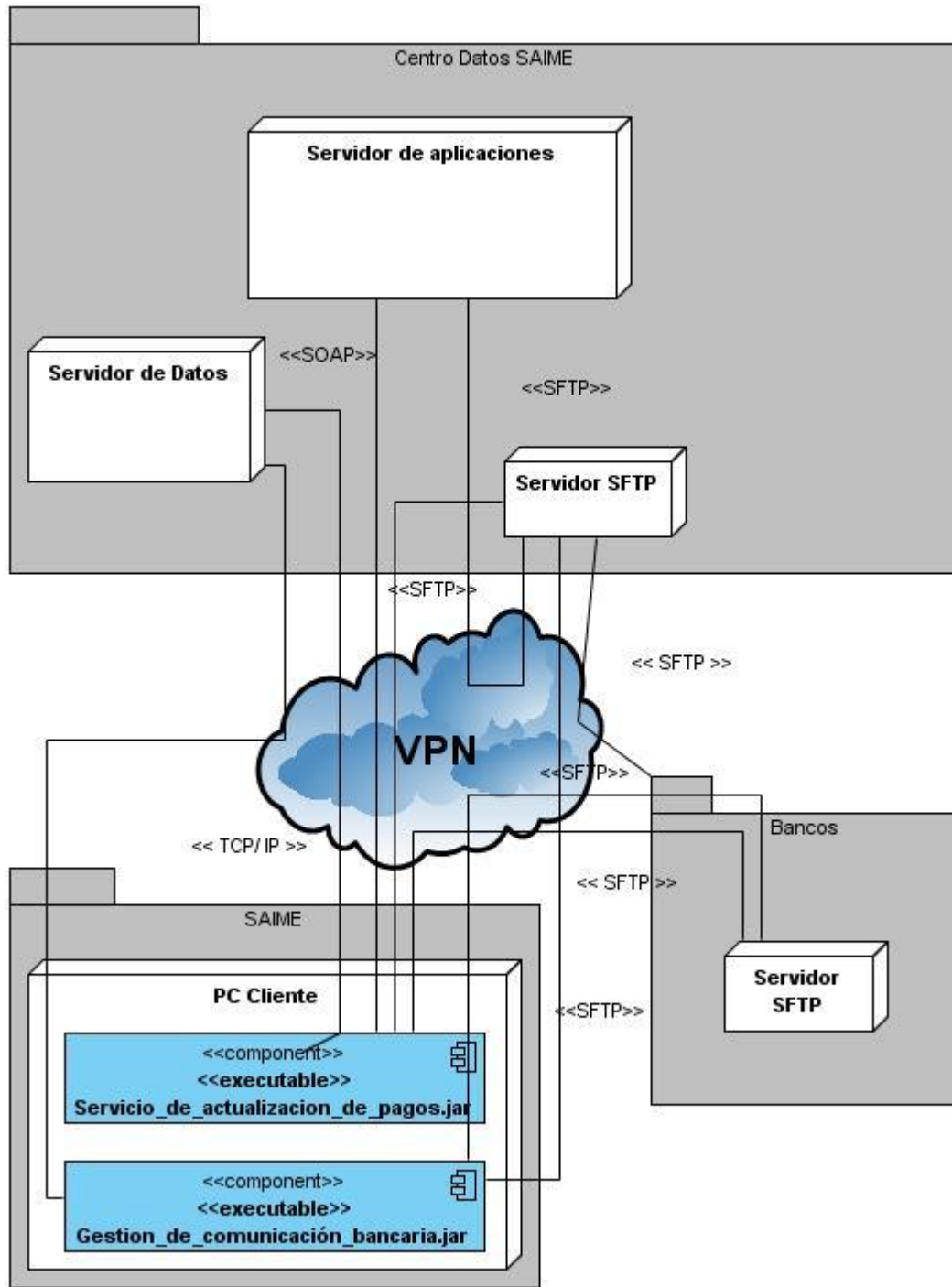


Figura. 9 Diagrama de despliegue.

3.3 Diagrama de Componentes.

Los componentes son partes modulares del sistema, que pueden desplegarse y reemplazarse. Por lo general los componentes contienen clases y pueden ser implementados por uno o más artefactos. El diagrama de componentes entonces, muestra un conjunto de componentes relacionados entre sí. Su uso fundamental es estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones de los elementos de implementación. A continuación se muestra cómo están estructurados los subsistemas de implementación de las aplicaciones Comunicación Bancaria y Actualización de Pagos en las figuras 10 y 11 respectivamente.

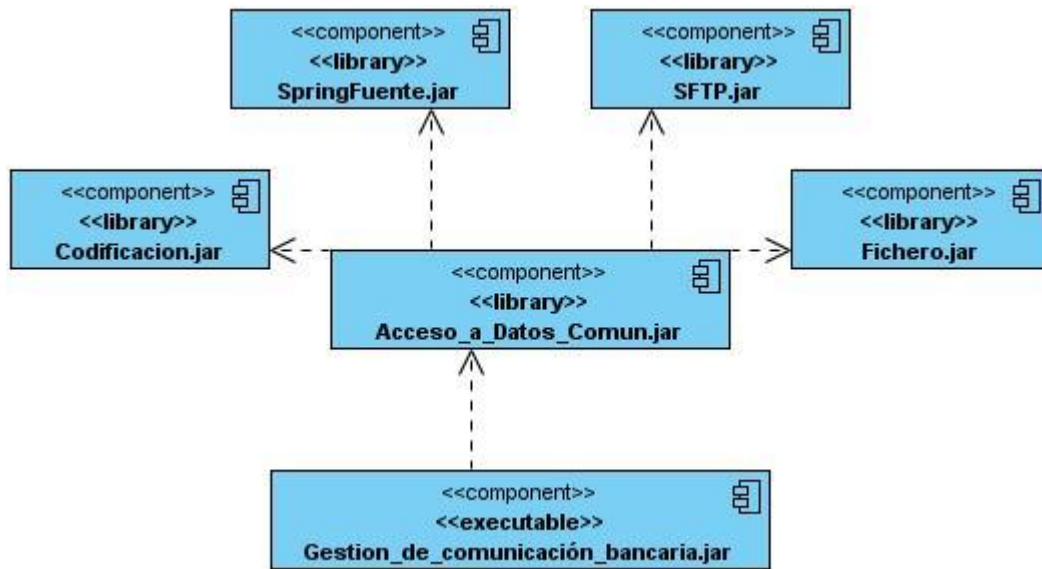


Figura. 10 Diagrama de componentes Gestión de comunicación bancaria.

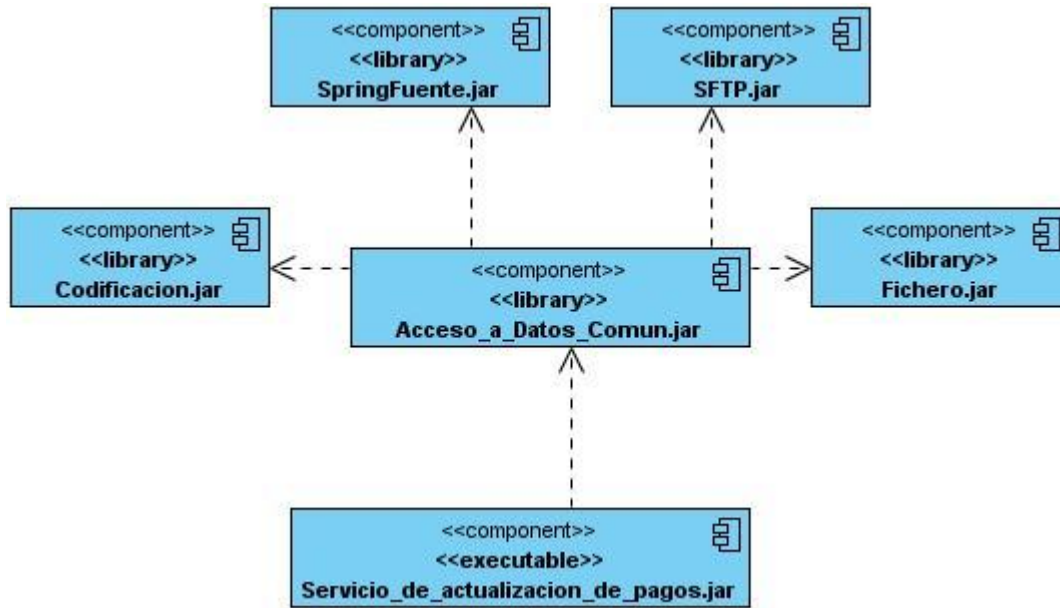


Figura. 11 Diagrama de componentes Servicio de actualización de pagos.

Gestion_de_comunicacion_bancaria.jar y Servicio_de_actualizacion_de_pagos.jar son los componentes principales de cada aplicación ya que son los encargados de interactuar con los usuarios. Estos hacen referencia al componente Acceso_a_Datos_Comun.jar el cual brinda funcionalidades comunes para las dos aplicaciones y es donde se encuentra la capa de acceso a datos, el componente SFTP.jar es el encargado de establecer comunicación con los servidores SFTP, Codificacion.jar representa los algoritmos usados para la codificación, SpringFuente.jar se encuentra el fichero de configuración de Spring y facilita el trabajo con las funciones básicas de los objetos del dominio y Fichero.jar posibilita el trabajo con los ficheros XML y CSV.

3.4 Descripción de las principales clases.

En las siguientes tablas se muestran las principales clases de implementación de ambas aplicaciones.

Nombre: *CGenerarCodigoSeguridad*

Tipo de clase: Controladora

Capítulo 3: Implementación y Prueba

Para cada funcionalidad	
Nombre:	Descripción:
generarCodigoSucursales(ComunDBancoRecaudador dBancoRecaudador, Date fecha): CSucursales	Genera los códigos de seguridad para las sucursales del banco.
generarCodigoSeguridad(long idProveedor, Date fecha): int	Genera el fichero XML con los códigos de los bancos, se almacenan en los servidores y se guarda en la base de datos.
salvarCodigoSeguridad(File fichero, boolean notificada): boolean	Guarda los códigos de seguridad que provienen del fichero en la base de datos.
subirfichero(ComunDproveedor proveedor, File fichero): boolean	Almacena el fichero XML al servidor de los bancos y si se pudo crea un correo.
notificarArchivoSeguridad(ComunDproveedor proveedor, File fichero): boolean	Sube el fichero XML al servidor de los bancos, lee el XML y guarda los datos en la base de datos y elimina el fichero local.
bancos(): List<ComunDBancoRecaudador>	Devuelve una lista de bancos.
servidorbanco(ComunDservidor servidor): boolean	Devuelve los servidores de cada banco.
importarSucursales(File f, ComunDBancoRecaudador banco): int	Registra las sucursales a cada banco importado desde un fichero csv.
conexión(String usuario, String contraseña, String ip int puerto): boolean	Prueba la conexión al servidor de un banco.

Tabla. 4 Descripción de la clase C Lectura perteneciente a Servicio de actualización de pago.

Capítulo 3: Implementación y Prueba

Nombre: <i>CLectura</i>	
Tipo de clase: Controladora	
Para cada funcionalidad	
Nombre:	Descripción:
ejecutar(long idLectura , Date dia) : int	Cuando la fecha que se le pasa es null ejecuta el método “ejecutarLecturaManual”, de lo contrario se ejecuta el método “ejecutarLecturaAutomatica”.
ejecutarLecturaAutomatica(ASBDlectura lectura, CSftp sftp, Atring fecha, String codigoBanco): int	Lee los resúmenes diarios de una fecha determinada en el servidor sftp.
ejecutarLecturaManual(ASBDlectura lectura, CSftp sftp, Atring fecha, String codigoBanco): int	Ejecuta una lectura sobre un servidor para leer los pagos de forma manual.
obtenerFicheroDirectorio(String destino, CSftp sftp , String fecha , String códigoBanco , char tipoFichero): List<String>	Obtiene los nombres de los ficheros ubicados de un directorio de un servidor.
descargar (CSftp sftp , String dirLocal , String dirdescarga , List<String> ficheros , String nombXml): List<String>	Descarga una lista de ficheros ubicados en un servidor en un directorio.
leerPagos(List<String> ficheros, char tipoFicheros): List<CPagoOrdinario>	Convierte los pagos de XML en objetos Java.
listPagosSinRepetir(List<CPagoOrdinario> reversos , List<CPagoOrdinario> notificaciones): List<CPagoOrdinario>	Verifica que no se repitan los pagos a la hora de insertarlos.
lecturasPendientes(): List<ASBDpendiente>	Devuelve el listado de lecturas pendientes.
lecturasProveedorDiaria(long idProveedor): List<ASBDlectura>	Devuelve el listado de lecturas diarias de un servidor.

Capítulo 3: Implementación y Prueba

crearLectura(ASBDlectura lectura):boolean	Crea una lectura.
modificarLectura(ASBDlectura lectura):int	Modifica una lectura.
eliminarLectura(ASBDlectura lectura):boolean	Elimina una lectura.
servidores(): List<ComunDservidor>	Devuelve un listado de servidores.

Tabla. 5 Descripción de la clase CGenerarCodigoSeguridad perteneciente a Gestión de comunicación bancaria.

3.5 Estándares de código.

Un estándar de codificación comprende los aspectos de la generación de código y repercute directamente en la legibilidad y la extensibilidad del código de cualquier proyecto de software, haciendo que los desarrolladores se acoplen rápidamente al proceso de desarrollo. A continuación se encuentran detallados todos los estándares de codificación que se definieron para la creación de este sistema teniendo en cuenta los lenguajes de programación.

Clases: los nombres de clases comienzan con mayúsculas y las palabras que la forman en minúsculas, en caso de que sea nombre compuesto siempre comenzarán con mayúscula. Algunas clases comienzan con un identificativo para ubicarlas lógicamente en el paquete.

Atributos de las clases: las variables miembros de clase se escriben con minúsculas, las variables con nombres compuestos, comienzan con la primera palabra enteramente en minúscula y el resto comenzando con mayúscula. Cuando se definan nombres de variables deben hacerse de forma representativa con el propósito de facilitar el entendimiento.

Métodos coherentes: cumple con las mismas restricciones que las variables. No debe añadirse demasiada complejidad a los procedimientos, pues si manejan muchas tareas resulta natural que sean difíciles de entender y tengan una alta probabilidad de ocurrencia de errores.

Comentarios: se comentarán los métodos al principio para facilitar la comprensión y entendimiento del código.

Saltos de línea: añadir un salto de línea después del cierre de los paréntesis de los parámetros y después de un punto y coma cuando termina la sentencia.

Longitud de la línea: evitar las líneas de más de 80 caracteres, cuando se supere esta cifra se debe reorganizar el código usando algún principio descrito anteriormente.

3.6 Modelo Prueba.

Desarrollar un software implica una serie de actividades de producción, las cuales son realizadas por personas y estas cometen errores y cambian de ideas. Las pruebas constituyen la actividad en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos especificados, los resultados son observados y registrados y una evaluación es realizada de algún aspecto del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. (1)

Para escoger los tipos de pruebas que se utilizaron se tuvo en cuenta el lenguaje de programación que se utilizó, el tipo de funcionalidad que se implementa, si la aplicación es de escritorio o web, si realiza conexiones a bases de datos y la metodología que se definió.

3.6.1 Pruebas unitarias.

El objetivo de las pruebas unitarias es el aislamiento de las partes del código y la demostración de que estas partes no contienen errores. Es por ello que se consideran a las pruebas unitarias como uno de los tipos de pruebas más importantes que se le aplican a los software, logrando como resultado que disminuya en un gran por ciento el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad. Las pruebas unitarias se le realizaron a los servicios del sistema para validar que las salidas son las correctas y asegurarle al desarrollador que su solución no presenta errores en la lógica de programación.

Cuando se prueba un programa, este se ejecuta con unos datos de entrada (casos de prueba) para verificar que el funcionamiento cumple los requisitos esperados. En los últimos años se han desarrollado un conjunto de herramientas que facilitan la elaboración de pruebas unitarias en diferentes lenguajes.

Dicho conjunto se denomina XUnit. Dentro de este, JUnit es la herramienta utilizada para realizar pruebas unitarias en Java. Existe una gran variedad de razones por las cuales utilizar JUnit, ellas son:

- ✓ Es gratis.
- ✓ Está integrado al entorno de desarrollo escogido.
- ✓ Se pueden escribir rápidamente los test que ejerciten el código e incrementalmente añadir test según va creciendo el software. Una vez que se han escrito algunos test, se pueden ejecutar rápido y frecuentemente sin interrumpir el diseño creativo ni el proceso de desarrollo.
- ✓ Los test JUnit se pueden ejecutar automáticamente y chequean sus propios resultados. Cuando se ejecutan los test, se obtiene una retroalimentación visual de inmediato, indicando si se ha pasado o fallado el test. No existe la necesidad de leer un informe para comparar los resultados.

3.6.2 Pruebas funcionales. (Caja Negra).

Las pruebas funcionales, son pruebas usadas por las personas encargadas de la calidad del software, ya que tienen como prioridad verificar el completo y buen funcionamiento del software. A este tipo de pruebas se les denomina también pruebas de Caja Negra.

Las pruebas de Caja Negra deben su nombre a los elementos que estas revisan y las condiciones en que se hace la revisión. Estas se basan en los requerimientos funcionales del sistema y se llevan a cabo desde el exterior de la aplicación. Este tipo de prueba es importante a la hora de medir el grado de cumplimiento de los requerimientos solicitados por el cliente y se aplican sobre la interfaz de la aplicación, observando las respuestas del sistema ante determinadas acciones y los datos de salida para determinados datos de entrada. (11)

Con el diseño de casos de prueba de Caja Negra se obtienen un conjunto de pruebas, cuya misión es encontrar defectos y errores en el sistema. Cada caso de prueba brinda un número de valores de entradas en las pruebas, condiciones de ejecución y resultados esperados de las mismas para verificar una determinada funcionalidad del sistema. Para cada caso de prueba se determinarán tantas secciones como las que tenga el caso de uso correspondiente. Se diseñaron casos de prueba por cada una de las funcionalidades del sistema, ver anexo 2.

3.7 Conclusiones parciales.

En este capítulo se definió el diagrama de componentes para cada aplicación, el diagrama de despliegue, los estándares de codificación por los que se rigió la implementación y se definieron las principales clases de la implementación. El resultado obtenido durante las pruebas realizadas fue la mejora del funcionamiento y la calidad de lo implementado, logrando mediante su ejecución continua, un código limpio de errores; listo para integrarse con las demás partes del sistema y ser probado como un todo por cada una de las funcionalidades que debe cumplir.

Conclusiones Generales

Conclusiones Generales.

Una vez terminado todo el proceso de trabajo se pudo llegar a la conclusión de que el desarrollo de los subsistemas es de vital importancia a la hora de recaudar los pagos del SAIME, con la afirmación de que este sistema da solución a la situación problemática que lo originó y se logra el cumplimiento del objetivo general trazado inicialmente así como los objetivos específicos. Además de que el costo por parte del software es mínimo, ya que fue construido con tecnologías gratuitas sin descuidar la calidad del producto. A continuación se mencionan los principales logros obtenidos.

- ✓ Se fundamentaron la metodología y las tecnologías utilizadas en el desarrollo de las aplicaciones, empleando como metodología de desarrollo RUP, el *framework* seleccionado fue Spring, el cual se integrará con el ORM Hibernate. Como lenguajes se utilizarán Java y XML en el entorno de desarrollo Netbeans, como lenguaje de modelado UML en la herramienta Visual Paradigm y como SGBD PostgreSQL.
- ✓ Se realizó un estudio de sistemas similares donde se decidió reutilizar el mecanismo de seguridad para validar la integridad de la información del sistema de pago seguro de SAREN.
- ✓ Se definieron los requerimientos funcionales y no funcionales que el sistema debe cumplir.
- ✓ Se desarrolló el análisis y diseño de las aplicaciones facilitando las bases de la implementación.
- ✓ Se describe la arquitectura en tres capas y se describen los patrones de diseño GRASP y GoF.
- ✓ La implementación de los subsistemas cumple con los requerimientos definidos.
- ✓ Las pruebas (unitarias y funcionales) realizadas a la solución permitieron corregir errores en la implantación, logrando así que los resultados finales sean satisfactorios.

Recomendaciones

Recomendaciones.

Durante la realización de la investigación surgieron ideas que pueden servir como recomendación para el perfeccionamiento del sistema, ellas son:

- ✓ Desarrollar la ayuda y el manual de usuarios de las aplicaciones.
- ✓ Realizar pruebas de estrés a las aplicaciones.
- ✓ Realizar el despliegue de la solución en los bancos recaudadores, oficinas y centro de datos del SAIME.

Referencias Bibliográficas

Referencias Bibliográficas

1. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *“El Proceso Unificado de Desarrollo de Software”*. 2000. : s.n.
2. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** El Lenguaje Unificado de Modelado. 2000.
3. Visual Paradigm for UML. [En línea] [Citado el: 12 de enero de 2011.] <http://www.visual-paradigm.com>.
4. CAVSI. *Sistema Gestor de base de datos*. [En línea] [Citado el: 6 de enero de 2011.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
5. **Denzer, Patricio.** PsotgreSQL. [En línea] 23 de octubre de 2002. [Citado el: enero de 13 de 2011.] <http://profesores.elo.utfsm.cl/~agv/elo330/2s02/projects/denzer/informe.pdf>.
6. **Quiñones, Ernesto.** Introducción a Postgresql. [En línea] [Citado el: 13 de enero de 2011.]
7. NetBeans. [En línea] [Citado el: 7 de enero de 2011.] http://netbeans.org/index_es.html.
8. **Walls, Craig y Breidenbach, Ryan.** *Spring in Action*. 2005.
9. La Capa de Lógica de Negocio y el Marco de Trabajo Spring. [En línea] [Citado el: 18 de enero de 2011.] http://www.programacion.net/tutorial/jap_jsfwork/3/.
10. Axis 2. *Kioskea*. [En línea] [Citado el: 12 de junio de 2011.] <http://es.kioskea.net/faq/3176-axis-2-parte-1>.
11. **Pressman, Roger.** *Ingeniería de software. Un enfoque práctico*. Interamericana de España : s.l. : McGraw.Hill, 2002.

Bibliografía

1. **Jacobson, I, Booch, G y Rumbaugh, J.** *El Proceso Unificado de Desarrollo de software.* 2000.
2. **Booch, G, Rumbaugh, J y Jacobson, I.** *El Lenguaje Unificado de Modelado.* 2000.
3. **Grau, Xavier Ferré.** Tutorial UML, Desarrollo Orientado a Objetos con UML. [En línea] 2004. [Citado el: 5 de 12 de 2010.] <http://www.clikear.com/manuales/uml/index.aspx>.
4. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* México : s.n., 1999.
5. Visual Paradigm for UML. *sitio Web de Visual Paradigm for UML.* [En línea] [Citado el: 12 de 1 de 2011.] <http://www.visual-paradigm.com..>
6. Sitio oficial e Postgres. [En línea] [Citado el: 13 de 1 de 2011.] <http://www.postgresql.org>.
7. [En línea] [Citado el: 16 de 1 de 2011.] <http://bibliodoc.uci.cu/pdf/reg03438.pdf>.
8. **Ángel, Miguel.** Descripción y características de este potente y moderno lenguaje de programación. [En línea] [Citado el: 14 de 1 de 2011.] <http://www.desarrolloweb.com/articulos/497.php>.
9. **Mercer, Dave.** *Fundamentos de programacih en XML Colombia .* 2001.
10. Sitio oficial de XML. [En línea] [Citado el: 15 de 1 de 2011.] <http://www.w3.org/XML/>.
11. NetBeans. [En línea] [Citado el: 17 de 1 de 2011.] http://netbeans.org/index_es.html.
12. Sitio oficial de hibernate. [En línea] [Citado el: 18 de 1 de 2001.] <http://www.hibernate.org>.
13. **Walls, Craig y Breidenbach, Ryan.** *Spring in Action.* 2005 : s.n.
14. La Capa de Lógica de Negocio y el Marco de Trabajo Spring. [En línea] [Citado el: 18 de 1 de 2011.] [http://www.programacion.net/tutorial/jap_jsfwork/3/..](http://www.programacion.net/tutorial/jap_jsfwork/3/)
15. **Fatima, Doval Garcia.** [En línea] [Citado el: 5 de febrero de 2011.] www.um.es/glosasdidacticas/GD14/10.pdf.
16. **Presman, Roger S.** *Ingeniería de Software. Un enfoque práctico.* 2002.
17. **Jayasinghe, Deepal.** *Quickstart Apache Axis2.* s.l. : Packt Publishing Ltd, 2008.
18. **Flanagan, David.** *Java en pocas palabras.* México: : McGraw-Hill: , 1999. U0003822.
19. Lenguaje de programación Java. *Ubuntu Arte.* [En línea] 31 de 3 de 2008. [Citado el: 11 de 6 de 2011.] <http://ubuntuarte.com/wordpress/?p=429>.
20. Introducción a Java. *HTMIPOINT.* [En línea] [Citado el: 11 de 6 de 2011.] http://www.htmlpoint.com/guidajava/java_01.htm.
21. Axis 2. *Kioskea.* [En línea] [Citado el: 12 de 6 de 2011.] <http://es.kioskea.net/faq/3176-axis-2-parte-1>.

Bibliografía

22. Programming4. *Apache Axis2 (Java)*. [En línea] [Citado el: 12 de 6 de 2011.]
<http://mscerts.programming4.us/es/840660.aspx>.
23. **Quiñones, Ernesto**. Introducción a Postgresql. [En línea] [Citado el: 13 de 1 de 2011.]
http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf.
24. **Denzer, Patricio**. PostgreSQL. [En línea] 23 de 10 de 2002. [Citado el: 13 de 1 de 2011.]
<http://profesores.elo.utfsm.cl/~agv/elo330/2s02/projects/denzer/informe.pdf>.
25. CAVSI. *Sistema gestor de base de datos*. [En línea] [Citado el: 16 de 1 de 2011.]
<http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
26. **Gómez Ballester, Eva y Martínez Barco, Patricio**. Bases de Datos 1. [En línea] [Citado el: 16 de 1 de 2011.] <http://www.alu.ua.es/~jmr36/Conectate/Base%20Datos/Apuntes2006.pdf>.
27. Clasificación de lenguajes de programación. [En línea] [Citado el: 14 de 1 de 2011.]
<http://qbitacora.wordpress.com/2007/09/21/clasificacion-de-lenguajes-de-programacion/>.
28. Programación Java. [En línea] [Citado el: 14 de 1 de 2011.] <http://www.lenguajes-de-programacion.com/programacion-java.shtml>.

Anexos

Anexos

Anexo 1 : Descripción de casos de usos.

Caso de Uso:	Administrar Lecturas de Depósito de Pagos
Actores:	Operador de oficina.
Resumen:	El caso de uso inicia cuando el Operador de oficina ejecuta la opción Administrar Lecturas de Pagos donde se crean, modifican, habilitan/deshabilitan y eliminan las lecturas, finalizando así el caso de uso.
Precondiciones:	El Operador de oficina debe estar autenticado en el sistema y el servidor debe estar creado.
Referencias	RF4
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El Operador de oficina selecciona la opción Administrar Lectura de pagos.	2-El sistema muestra un menú con las opciones: <ul style="list-style-type: none">• Crear Lectura• Modificar Lectura.• Habilitar Lectura.• Eliminar Lectura.
3- El Operador de oficina selecciona la opción: 3a) Crear Lectura, ir a Sección 1 “Crear Lectura” . 3b) Modificar Lectura, ir a Sección 2 “Modificar Lectura” . 3c) Habilitar/ Deshabilitar Lectura, ir a Sección 3 “Habilitar Lectura” . 3e) Eliminar Lectura, ir a Sección 4 “Eliminar Lectura” .	
Sección 1 “Crear Lectura”	
	1-El sistema permite: 1a) Ver listado de servidores configurados en los bancos con los siguientes datos:

Anexos

	<ul style="list-style-type: none">• Nombre del banco.• Código del banco.• Ip del servidor. 1b) Seleccionar el servidor que desea crearle la lectura.
2-El Operador de oficina selecciona al servidor que desea crearle la lectura.	3-El sistema muestra los siguientes campos a llenar. <ul style="list-style-type: none">• Nombre lectura.• Fecha inicial.• Fecha final.• Hora lectura.• Periodicidad.• Habilitar lectura Y las opciones: <ul style="list-style-type: none">• Crear• Cerrar
4-El Operador de oficina introduce los datos y selecciona la opción: 4a) Aceptar , continúa en la Sección 1 “Crear Lectura” parte 5. 4b) Cerrar , ir al Flujo Alterno 1 “Cerrar Acción” .	5- El sistema valida que no haya campos vacíos sino ir al Flujo Alterno 2 “Campos Vacíos Crear Lectura” .
	6- El sistema valida que los datos estén correctos sino ir al Flujo Alterno 3 “Datos Incorrectos Crear Lectura”
	7- El sistema guarda los datos en la base de datos y muestra un mensaje satisfactorio. En caso contrario, ver Flujo Alterno 8 “Error de Conexión” .
Prototipo Interfaz	

Seleccionar servidor			Crear Lectura	
Código	Nombre banco	IP servidor	Nombre lectura:	<input type="text"/>
1100	Banco Exterior	10.13.35.120	Fecha inicial:	4/06/11 <input type="button" value="■"/>
1101	Banfoandes	10.13.35.120	Fecha final:	4/06/11 <input type="button" value="■"/>
1102	Banco del Tesoro	10.13.35.120	Hora lectura:	03:26:59 PM <input type="button" value="▲▼"/>
1103	Banco de Venezuela	10.13.35.120	Periodicidad:	<input type="text"/>
1105	Bancaribe	10.13.35.120	<input type="checkbox"/> Habilitar lectura	
1106	Banco Caroni	10.13.35.120		
1107	Banco Provincial	10.13.35.120		
1108	CorpBanca	10.13.35.120		
1109	Banco Plaza	10.13.35.120		
1110	Banco Banesco	10.13.35.120		
1104	Banco Fondo Común	10.13.35.120		
			<input type="button" value="Crear"/>	<input type="button" value="Cerrar"/>

Sección 2 “Modificar Lectura”

	<p>1-El sistema permite:</p> <p>1a) Ver listado de servidores configurados en los bancos con los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre del banco. • Código del banco. • Ip del servidor. <p>1b) Seleccionar el servidor que desea modificarle la lectura.</p>
<p>2-El Operador de oficina selecciona al servidor que desea modificarle la lectura.</p>	<p>3- El sistema permite:</p> <p>3a) Ver listado de lecturas programadas del servidor seleccionado.</p> <p>3b) Seleccionar la lectura programada a modificar del servidor seleccionado.</p>
<p>4-El Operador de oficina selecciona la lectura que desea modificar.</p>	<p>5-El sistema muestra una interfaz con los siguientes campos a modificar.</p> <ul style="list-style-type: none"> • Nombre lectura. • Fecha inicial. • Fecha final. • Hora lectura. • Periodicidad. • Habilitar lectura.

Anexos

	Y las opciones: <ul style="list-style-type: none">• Modificar.• Cerrar.
6-El Operador de oficina modifica los datos deseados y selecciona la opción: 6a) Modificar , continúa en la Sección 2“Modificar Lectura” parte 7. 6b) Cerrar , ir al Flujo Alterno 1 “Cerrar Acción” .	7- El sistema valida que no haya campos vacíos sino ir al Flujo Alterno 4 “Campos Vacíos Modificar Lectura” .
	8- El sistema valida que los datos estén correctos sino ir al Flujo Alterno 5 “Datos Incorrectos Modificar Lectura” .
	9 - El sistema valida que no se esté ejecutándose la lectura en ese momento de modificación sino ir al Flujo Alterno 6 “Lectura Ejecutándose Modificar Lectura” .
	10 - El sistema guarda los cambios efectuados en la base de datos y muestra un mensaje satisfactorio. En caso contrario, ver Flujo Alterno 9 “Error de Conexión Modificar Lectura” .
Prototipo Interfaz	

Seleccionar servidor

Código	Nombre banco	IP servidor
1100	Banco Exterior	10.13.35.120
1101	Banfoandes	10.13.35.120
1102	Banco del Tesoro	10.13.35.120
1103	Banco de Venezuela	10.13.35.120
1105	Bancaribe	10.13.35.120
1106	Banco Caroni	10.13.35.120
1107	Banco Provincial	10.13.35.120
1108	CorpBanca	10.13.35.120
1109	Banco Plaza	10.13.35.120
1110	Banco Banesco	10.13.35.120
1104	Banco Fondo Común	10.13.35.120

Seleccionar lectura a modificar

Nombre	Hora	Habilitada
Por la tarde	10:34:47	Habilitada
Lectura	23:32:02	Habilitada
De noche	10:34:47	Habilitada

Datos de la Lectura

Nombre lectura:

Fecha inicial:

Fecha final:

Hora de la lectura:

Periodicidad:

Habilitar lectura

Sección 3 “Habilitar Lectura”

	<p>1-El sistema permite:</p> <p>1a) Ver listado de servidores configurados en los bancos con los siguientes datos:</p> <ul style="list-style-type: none"> Nombre del banco. Código del banco. Ip del servidor. <p>1b) Seleccionar el servidor que desea cambiarle el estado de la lectura.</p>
2-El Operador de oficina selecciona el servidor que desea cambiarle el estado de la lectura.	<p>3-El sistema permite:</p> <p>3a) Ver listado de lecturas programadas en el servidor seleccionado.</p>

	<p>3b) Seleccionar la lectura que desea cambiarle el estado. Y las opciones:</p> <ul style="list-style-type: none"> • Cambiar. • Cerrar.
<p>4-El Operador de oficina selecciona la lectura deseada y selecciona la opción: 4a) Cambiar, continúa en la Sección 3 “Habilitar Lectura” parte 5. 4b) Cerrar, ver Flujo Alterno 1 “Cerrar Acción”.</p>	<p>5- El sistema guarda los cambios efectuados en la base de datos y muestra un mensaje satisfactorio. En caso contrario, ver Flujo Alterno 10 “Error de Conexión Habilitar Lectura”.</p>

Prototipo Interfaz

El prototipo de interfaz muestra dos paneles de selección:

Seleccionar servidor

Código	Nombre banco	IP servidor
1100	Banco Exterior	10.13.35.120
1101	Banfoandes	10.13.35.120
1102	Banco del Tesoro	10.13.35.120
1103	Banco de Venezuela	10.13.35.120
1105	Bancaribe	10.13.35.120
1106	Banco Caroni	10.13.35.120
1107	Banco Provincial	10.13.35.120
1108	CorpBanca	10.13.35.120
1109	Banco Plaza	10.13.35.120
1110	Banco Banesco	10.13.35.120
1104	Banco Fondo Común	10.13.35.120

Seleccionar lectura a habilitar o deshabilitar

Nombre	Hora	Habilitada
Lectura	23:32:02	Habilitada
De noche	10:34:47	Habilitada
Por la tarde	10:34:47	Deshabilitada

Sección 4 “Eliminar Lectura”

	<p>1-El sistema permite:</p> <p>1a) Ver listado de servidores configurados en los bancos con los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre del banco. • Código del banco. • Ip del servidor. <p>1b) Seleccionar el servidor que desea eliminarle la lectura.</p>
--	--

Anexos

<p>2-El Operador de oficina selecciona el servidor que desea eliminarle la lectura.</p>	<p>3-El sistema permite:</p> <p>3a) Ver listado de lecturas programadas en el servidor seleccionado.</p> <p>3b) Seleccionar la lectura que desea eliminar.</p> <p>Y las opciones:</p> <ul style="list-style-type: none"> • Eliminar. • Cerrar.
<p>4-El Operador de oficina selecciona la lectura a eliminar y selecciona la opción:</p> <p>4a) Eliminar, continúa en la Sección 4 “Eliminar Lectura” parte 5.</p> <p>4b) Cerrar, ver Flujo Alternativo 1 “Cerrar Acción”.</p>	<p>5-El sistema verifica que las lecturas seleccionadas no estén ejecutándose sino ir al Flujo Alternativo 7 “Lectura Ejecutándose Eliminar Lectura”.</p>
	<p>6-El sistema guarda los cambios efectuados en la base de datos y muestra un mensaje satisfactorio. En caso contrario, ver Flujo Alternativo 11 “Error de Conexión Eliminar Lectura”.</p>

Prototipo de Interfaz

Seleccionar servidor

Código	Nombre banco	IP servidor
1100	Banco Exterior	10.13.35.120
1101	Banfoandes	10.13.35.120
1102	Banco del Tesoro	10.13.35.120
1103	Banco de Venezuela	10.13.35.120
1105	Bancaribe	10.13.35.120
1106	Banco Caroni	10.13.35.120
1107	Banco Provincial	10.13.35.120
1108	CorpBanca	10.13.35.120
1109	Banco Plaza	10.13.35.120
1110	Banco Banesco	10.13.35.120
1104	Banco Fondo Común	10.13.35.120

Seleccionar lectura a Eliminar

Nombre	Hora	Habilitada
De noche	10:34:47	Habilitada
Lectura	23:32:02	Habilitada
Por la tarde	10:34:47	Habilitada

Flujos Alternos

Anexos

Flujo Alternativo 1 "Cerrar Acción"	
Acción del Actor	Respuesta del Sistema
	1-Se reinician los valores de los campos y se muestra la interfaz principal.
Flujo Alternativo 2 "Campos Vacíos Crear Lectura"	
Acción del Actor	Respuesta del Sistema
	1-El sistema muestra un mensaje notificando al usuario que ha dejado campos vacíos, volviendo a la Sección 1 "Crear Lectura" parte 3.
Flujo Alternativo 3 "Datos Incorrectos Crear Lectura"	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje notificando al usuario que no ha llenado los campos correctamente, volviendo a la Sección 1 "Crear Lectura" parte 3.
Flujo Alternativo 4 "Campos Vacíos Modificar Lectura"	
Acción del Actor	Respuesta del Sistema
	1-El sistema muestra un mensaje notificando al usuario que ha dejado campos vacíos, volviendo a la Sección 2 "Modificar Lectura" parte 5.
Flujo Alternativo 5 "Datos Incorrectos Modificar Lectura"	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje notificando al usuario que no ha llenado los campos correctamente, volviendo a la Sección 2 "Modificar Lectura" parte 5.
Flujo Alternativo 6 "Lectura Ejecutándose Modificar Lectura"	
	1-El sistema muestra un mensaje que no se puede modificar la lectura ya que está ejecutándose en ese momento volviendo a la Sección 2 "Modificar Lectura" parte 5.
Flujo Alternativo 7 "Lectura Ejecutándose Eliminar Lectura"	
	1-El sistema muestra un mensaje que no se puede eliminar la lectura ya que está ejecutándose en ese momento, volviendo a la Sección 4 "Eliminar Lectura" parte 3.

Anexos

Flujo Alternativo 8 “Error de Conexión Crear Lectura”	
	1-El sistema muestra un mensaje diciendo que no existe conexión a la Base de datos, volviendo a la Sección 4 “Crear Lectura” parte 3.
Flujo Alternativo 9 “Error de Conexión Modificar Lectura”	
	1-El sistema muestra un mensaje diciendo que no existe conexión a la Base de datos, volviendo a la Sección 4 “Modificar Lectura” parte 5.
Flujo Alternativo 10 “Error de Conexión Habilitar Lectura”	
	1-El sistema muestra un mensaje diciendo que no existe conexión a la Base de datos, volviendo a la Sección 4 “Habilitar Lectura” parte 3.
Flujo Alternativo 11 “Error de Conexión Eliminar Lectura”	
	1-El sistema muestra un mensaje diciendo que no existe conexión a la Base de datos en ese momento, volviendo a la Sección 4 “Eliminar Lectura” parte 3.
Poscondiciones	Quedan administradas las lecturas.

Tabla. 6 Descripción del caso de uso Administrar lecturas de pagos.

Caso de Uso:	Ejecutar Lectura Manual.
Actores:	Operador de Administración
Resumen:	El caso de uso inicia cuando el Operador de Administración selecciona la opción: Ejecutar Lectura Manual , se muestra un listado de los servidores de los bancos, se escoge unos y se muestra las lecturas que tiene, tanto las actuales como las pendientes y se escoge una de ellas para ejecutarla, finalizando así el caso de uso.
Precondiciones:	El Operador de Administración debe estar autenticado en el sistema y los servidores deben de estar creados.
Referencias	RF5
Prioridad	Crítico
Flujo Normal de Eventos	

Anexos

Acción del Actor	Respuesta del Sistema
1- El Operador de Administración selecciona la opción: Ejecutar Lectura Manual .	2-El sistema permite: 2a) Ver listado de servidores configurados en los bancos con los siguientes datos: <ul style="list-style-type: none"> • Nombre del banco. • Código del banco. • Ip del servidor. 2b) Seleccionar el servidor que desea ejecutarle la lectura.
3-El Operador de Administración selecciona al servidor que desea ejecutarle la lectura.	4- El sistema permite: 4a) Ver listado de lecturas programadas pendientes y actuales del servidor seleccionado con los siguientes datos: <ul style="list-style-type: none"> • Nombre de la lectura • Hora de la lectura • Habilitada 4b) Seleccionar la lectura programada pendiente o actual a ejecutar del servidor seleccionado. Y las opciones: <ul style="list-style-type: none"> • Ejecutar • Cerrar
5-El Operador de Administración selecciona la lectura que desea ejecutar y selecciona la opción: 6a) Ejecutar , continua en la Sección Flujo Normal de Eventos parte 6. 6b) Cerrar , ir al Flujo Alternativo 1 “Cerrar Acción” .	6-El sistema verifica que la lectura se haya ejecutado correctamente sino ir al Flujo Alternativo 2 “Error de Ejecución” .
	7- El sistema muestra un mensaje satisfactorio indicando que se ejecutó la lectura seleccionada.
Flujos Alternos	
Flujo Alternativo 1 “Cerrar Acción”	
Acción del Actor	Respuesta del Sistema

Anexos

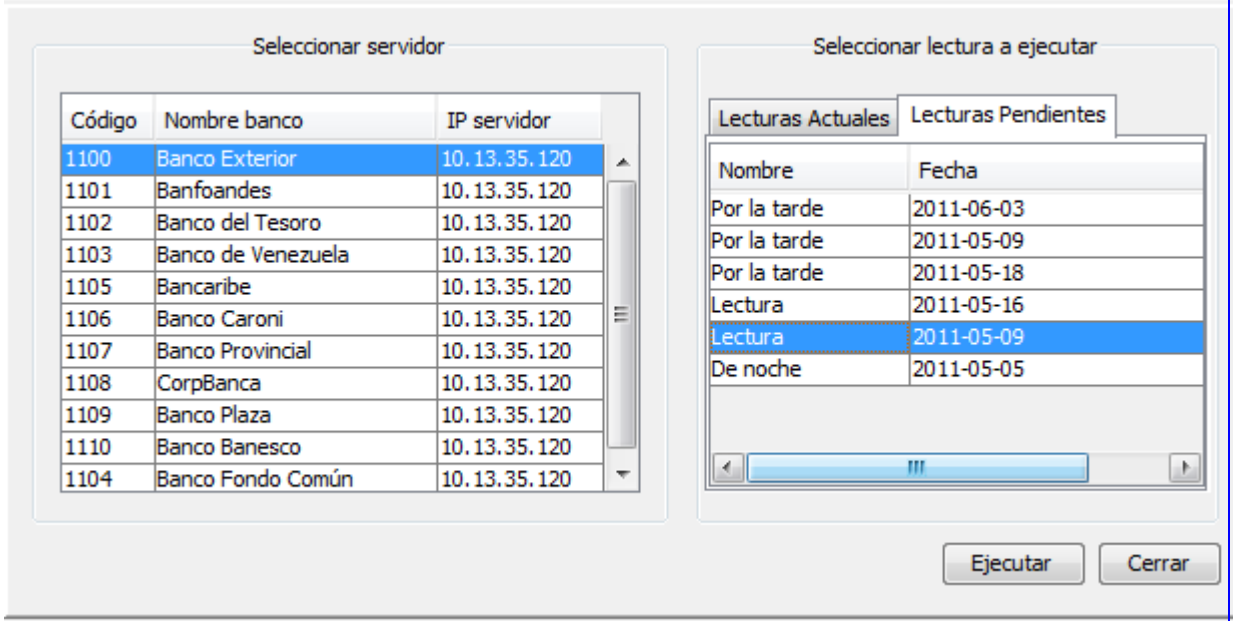
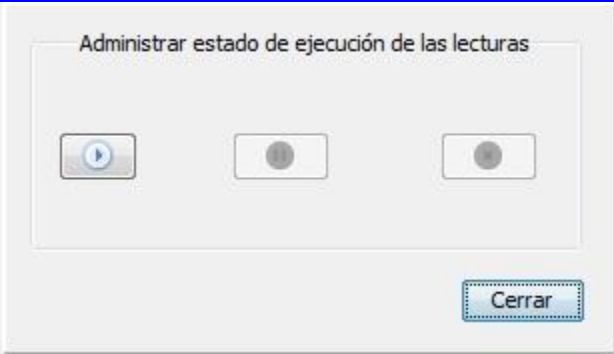
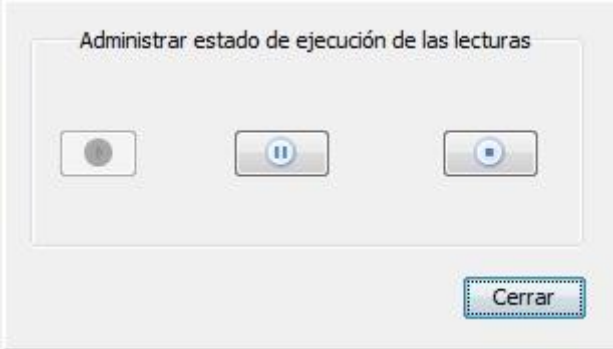
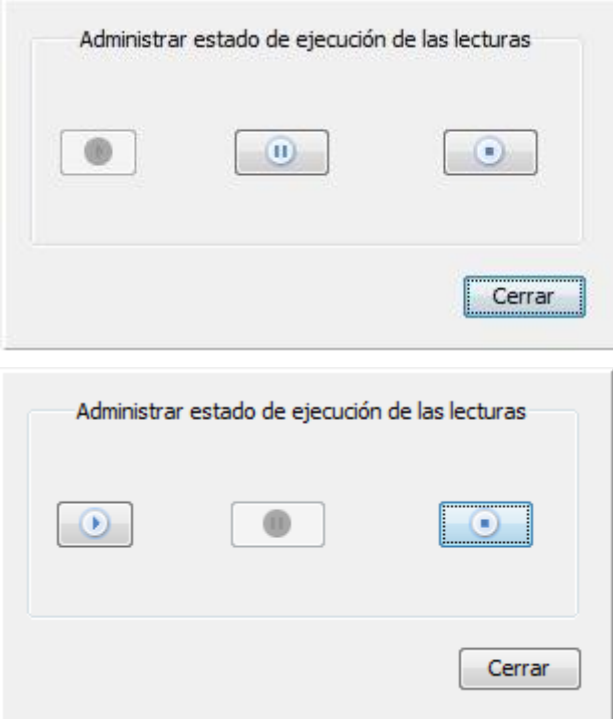
	1- Se reinician los valores de los campos y se muestra la interfaz principal.
Flujo Alterno 2 “Error de Ejecución”	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje indicando que existió un error en la ejecución de la lectura volviendo al flujo normal de eventos parte 4.
Prototipo de Interfaz	
	
Poscondiciones	Quedan ejecutadas las lecturas de forma manual.

Tabla. 7 Descripción del caso de uso Ejecutar lectura manual.

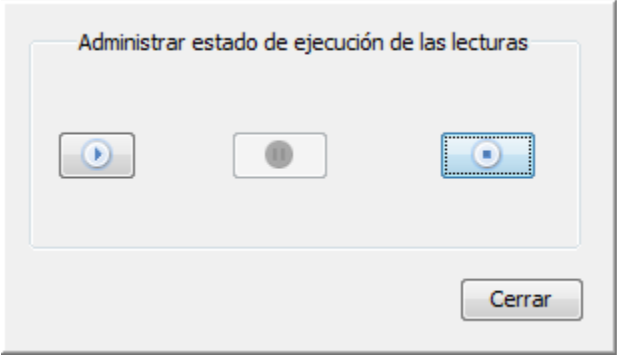
Caso de Uso:	Ejecutar Lecturas Automáticas.
Actores:	Operador de Administración
Resumen:	El caso de uso inicia cuando el Operador de Administración selecciona la opción Ejecutar Lecturas Automáticas e inicia la ejecución, la misma puede ser detenida, reiniciada y cerrada, finalizando así el caso de uso.
Precondiciones:	El Operador de Administración debe estar autenticado en el sistema.
Referencias	RF6

Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
2- El Operador de Administración selecciona la opción: Ejecutar Lecturas Automáticas .	3- El sistema permite seleccionar la opción: <ul style="list-style-type: none"> • Iniciar • Cerrar 	
4- El Operador de Administración selecciona la opción: 3a) Iniciar , continua en la Sección Flujo Normal de Eventos parte 4. 3b) Cerrar , ir al Flujo Alterno 1 “Cerrar Acción” .	5- El sistema verifica que exista conexión con el servidor web y que el servicio solicitado esté corriendo mostrando un mensaje satisfactorio diciendo que se inició correctamente la ejecución de la lectura. En caso contrario, ver Flujo Alterno 2 “Error de Conexión” .	
	6- El sistema permite seleccionar las siguientes opciones: <ul style="list-style-type: none"> • Pausa • Detener • Cerrar 	
6- El Operador de Administración selecciona la opción: 6a) Pausa, ir a Sección 1 “Pausar Servicio” . 6b) Detener Lectura, ir a Sección 2 “Detener Servicio” . 6c) Cerrar, ir al Flujo Alterno 1 “Cerrar Ventana” .		
		

Sección 1 “Pausar Servicio”	
	1-El sistema verifica que exista conexión con el servidor web y que el servicio solicitado esté corriendo mostrando un mensaje satisfactorio diciendo que se para correctamente la ejecución de la lectura. En caso contrario, ver Flujo Alterno 3 “Error de Conexión” .
	2-El sistema permite seleccionar las siguientes opciones: <ul style="list-style-type: none">• Reiniciar• Detener• Cerrar
3- El Operador de Administración selecciona la opción: 3a) Continuar , ir a Sección 3 “Continuar Servicio . 3b) Detener Lectura , ir a Sección 2 “Detener Servicio” . 3c) Cerrar , ir al Flujo Alterno 1 “Cerrar Ventana” .	
	
Sección 2 “Detener Servicio”	
	1- El sistema verifica que exista conexión con el servidor web y que el servicio solicitado esté corriendo mostrando un mensaje satisfactorio

	<p>diciendo que se detuvo correctamente la ejecución de la lectura. En caso contrario, ver Flujo Alternativo 5 “Error de Conexión”.</p>
	<p>2-El sistema permite seleccionar las siguientes opciones:</p> <ul style="list-style-type: none">• Iniciar• Cerrar
<p>3- El Operador de Administración selecciona la opción:</p> <p>3a) Iniciar ir a Sección Flujo Normal de Eventos parte 4.</p> <p>3c) Cerrar, ir al Flujo Alternativo 1 “Cerrar Ventana”.</p>	
 <p>The image displays two screenshots of a software interface titled "Administrar estado de ejecución de las lecturas". The top screenshot shows three buttons: a grey circle, a pause icon, and a square with a circle. The bottom screenshot shows three buttons: a play icon, a grey circle, and a square with a circle. Both screenshots have a "Cerrar" button at the bottom right.</p>	
<p>Sección 2 “Continuar Servicio”</p>	

Anexos

	<p>1- El sistema verifica que exista conexión con el servidor web y que el servicio solicitado esté corriendo mostrando un mensaje satisfactorio diciendo que se Continua correctamente la ejecución de la lectura. En caso contrario, ver Flujo Alterno 5 “Error de Conexión”.</p>
	<p>2-El sistema permite seleccionar las siguientes opciones:</p> <ul style="list-style-type: none"> • Pausa • Detener • Cerrar
<p>3- El Operador de Administración selecciona la opción:</p> <p>3a) Iniciar ir a Sección Flujo Normal de Eventos parte 4.</p> <p>3b) Detener Lectura, ir a Sección 2 “Detener Servicio”.</p> <p>3c) Cerrar, ir al Flujo Alterno 1 “Cerrar Ventana”.</p>	
	
<p>Prototipo de Interfaz</p>	
<p>Flujos Alternos</p>	
<p>Flujo Alterno 1 “Cerrar Acción”</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>

Anexos

	1- El sistema permite que el servicio se siga realizando y se muestra la interfaz principal.
Flujo Alternativo 2 “Error de Conexión iniciar”	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje diciendo que no existe conexión con el servidor web volviendo al flujo normal de eventos paso 2.
Flujo Alternativo 3 “Error de Conexión Pausar”	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje diciendo que no existe conexión con el servidor web volviendo al flujo normal de eventos paso 1.
Flujo Alternativo 4 “Error de Conexión Continuar”	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje diciendo que no existe conexión con el servidor web volviendo al flujo normal de eventos paso 1.
Flujo Alternativo 5 “Error de Conexión Detener”	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje diciendo que no existe conexión con el servidor web volviendo al flujo normal de eventos paso 1.
Poscondiciones	Se realiza la ejecución de las lecturas de forma automática.

Tabla. 8 Descripción del caso de uso Ejecutar lectura automáticas.

Caso de Uso:	Configurar servidor SFTP.
Actores:	Administrador del sistema.
Resumen:	El caso de uso inicia cuando el Administrador del sistema selecciona la opción de Configurar servidor SFTP , se crea la conexión bancaria y se prueba la misma, finalizando así el caso de uso.

Anexos

Precondiciones:	El administrador del sistema debe estar autenticado en el sistema y debe de existir el banco al cual se le configurará el servidor.
Referencias	RF1
Prioridad	Crítica
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El Administrador de sistema selecciona la opción de “Configurar servidor FTP” .	2- El sistema permite: 2a) Ver listado de bancos recaudadores con los siguientes datos: <ul style="list-style-type: none"> • Código del banco. • Nombre del banco. 2b) Seleccionar el banco que desea configurarle la conexión del servidor.
3- El Administrador de sistema selecciona el banco que desea configurarle la conexión del servidor.	4- El sistema permite configurar la conexión por primera vez o modificarla en caso de que ya se halla creado. Los datos que se mostraran son los siguiente: <ul style="list-style-type: none"> • IP • Puerto • Usuario • Contraseña Y las opciones: <ul style="list-style-type: none"> • Comprobar Conexión • Guardar • Cerrar
5-El Administrador del Sistema introduce los datos en caso de que la configuración se realizaba por primera vez o necesite modificar los datos existentes y selecciona la opción: 5a) Comprobar Conexión , continúa en la Sección 1 “Comprobar conexión” . 5b) Guardar , continúa en la Sección Flujo Normal de Eventos parte 6.	6- El sistema valida que no haya campos vacíos. En caso contrario, ver Flujo Alterno 2 “Campos Vacíos” .

Anexos

5c) Cerrar , ver Flujo Alternativo 1 “Cerrar Acción” .	
	7- El sistema valida que los datos entrados sean correctos. En caso contrario, ver Flujo Alternativo 3 “Datos Incorrectos” .
	8- El sistema guarda la configuración creada o modificada mostrando un mensaje satisfactorio. En caso contrario, ver Flujo Alternativo 5 “Error de Configuración” .
Sección 1 “Comprobar conexión”	
	1-El sistema valida que no haya campos vacíos. En caso contrario, ver Flujo Alternativo 2 “Campos Vacíos” .
	2- El sistema valida que los datos entrados sean correctos. En caso contrario, ver Flujo Alternativo 3 “Datos Incorrectos” .
	3- El sistema muestra un mensaje satisfactorio diciendo que existe conexión al servidor del banco seleccionado, en caso contrario ver Flujo Alternativo 4 “Error de Conexión” .
Prototipo Interfaz	
Flujos Alternos	
Flujo Alternativo 1 “Cerrar Acción”	
Acción del Actor	Respuesta del Sistema
	1- Se reinician los valores de los campos y se muestra la interfaz principal
Flujo Alternativo 2 “Campos Vacíos”	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje notificando al usuario que ha dejado campos sin llenar volviendo al flujo normal de eventos parte 4.

Anexos

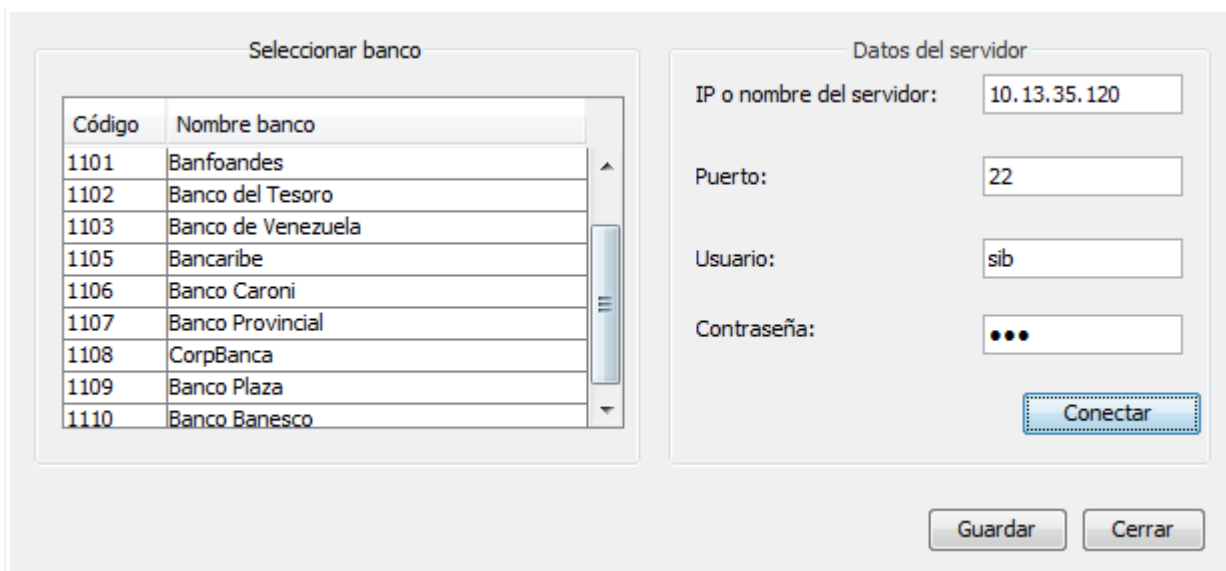
Flujo Alterno 3 “Datos Incorrectos”	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje notificando al usuario que no ha llenado los campos correctamente volviendo al flujo normal de eventos parte 4.
Flujo Alterno 4 “Error de Conexión”	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje indicando que no existe conexión al servidor del banco seleccionado volviendo al flujo normal de eventos parte 4.
Flujo Alterno 5 “Error de Configuración”	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje indicando que existe error a al guardar los datos en la base de datos volviendo al flujo normal de eventos parte 4.
Prototipo Interfaz	
	
Poscondiciones	EL servidor queda configurado o modificado.

Tabla. 9 Descripción del caso de uso Configurar servidor SFTP.

Anexos

Caso de Uso:	Generar archivo de seguridad.
Actores:	Operador de Administración
Resumen:	El caso de uso inicia cuando el Operador de Administración ejecuta la opción Generar archivo de seguridad, selecciona la fecha de comienzo de uso de las cadenas, envía un fichero XML con las cadenas generadas a los bancos y estos serán notificados, finalizando así el caso de uso.
Precondiciones:	El Operador de Administración debe estar autenticado en el sistema.
Referencias	RF2
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El caso de uso se inicializa cuando el Operador de Administración selecciona la opción Generar archivo seguridad.	2-El sistema permite seleccionar: <ul style="list-style-type: none"> • Fecha de comienzo de uso Y las opciones: <ul style="list-style-type: none"> • Generar • Cerrar
3- El Operador de Administración selecciona la fecha y la opción: 3a) Generar: continúa en la Sección Flujo Normal de Eventos parte 4. 3b) Cerrar, ver Flujo Alterno 1 “Cerrar Acción”.	4-El sistema genera el archivo de seguridad para las oficinas y las sucursales por bancos, guardándolas en la base de datos y generando un fichero XML por cada banco. En caso contrario, ver Flujo Alterno 2 “Error de conexión base datos”.
	5- El sistema envía los ficheros generados por vía sftp en el servidor de cada uno de los banco y se notifican para que sepan que se generó la cadena de seguridad, mostrándose un mensaje satisfactorio. Y muestra la interfaz principal. En caso contrario, ver Flujo Alterno 3 “Ficheros no Enviados”.
Prototipo de Interfaz	
Flujos Alternos	
Flujo Alterno 1 “Cerrar Acción”	

Anexos

Acción del Actor	Respuesta del Sistema
	1- Se reinician los valores de los campos y se muestra la interfaz principal.
Flujo Alternativo 2 "Error de conexión base datos"	
Acción del Actor	Respuesta del Sistema
	El sistema muestra un mensaje indicando que existe error al guardar los datos en la base de datos volviendo al flujo normal de eventos parte 2.
Flujo Alternativo 3 "Ficheros no Guardados"	
Acción del Actor	Respuesta del Sistema
	1- El sistema hace una copia local del fichero que no se pudo enviar al banco, se le muestra un mensaje al usuario y se le notifica por vía correo de lo sucedido para enviarlo posteriormente. Y se muestra la interfaz principal.

Prototipo de Interfaz

Seleccione la fecha de comienzo de uso de la cadena de seguridad

mayo 2011

lun mar mié jue vie sáb dom

25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Generar Cerrar

Anexos

Poscondiciones	Queda generado el código de seguridad para el banco.
-----------------------	--

Tabla. 10 Descripción del caso de uso Generar archivo seguridad.

Caso de Uso:	Enviar archivo de seguridad pendiente.	
Actores:	Operador de Administración	
Resumen:	El caso de uso inicia cuando el Operador de Administración ejecuta la opción Enviar archivo de seguridad pendiente , selecciona el archivo a enviar, envía un fichero con las cadenas generadas, al banco y este es notificado, finalizando así el caso de uso.	
Precondiciones:	El Operador de Administración debe estar autenticado en el sistema y deben a ver archivos de seguridad sin enviar.	
Referencias	RF2	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1-El caso de uso se inicializa cuando el Operador de Administración selecciona la opción Enviar archivo de seguridad pendiente .	2-El sistema muestra un listado con los archivos que no pudieron ser enviados y permite seleccionar: <ul style="list-style-type: none"> • Archivo de seguridad sin enviar Y las opciones: <ul style="list-style-type: none"> • Enviar • Cerrar 	
3- El Operador de Administración selecciona el archivo y la opción: 3a) Enviar: continúa en la Sección Flujo Normal de Eventos parte 4. 3b) Cerrar, ver Flujo Alterno 1 “Cerrar Acción” .	4-El sistema envía el archivo de seguridad al banco, registrándola como enviado en la base de datos y se elimina la lista de pendientes mostrándose un mensaje satisfactorio. En caso contrario, ver Flujo Alterno 2 “Ficheros no Enviados” .	
Prototipo de Interfaz		
Flujos Alternos		
Flujo Alterno 1 “Cerrar Acción”		


Acción del Actor	Respuesta del Sistema
	1- Se reinician los valores de los campos y se muestra la interfaz principal.
Flujo Alternativo 2 "Ficheros no Guardados"	
Acción del Actor	Respuesta del Sistema
	2- El sistema muestra un mensaje al usuario que no se pudo enviar el archivo.
Prototipo de Interfaz	
	
Poscondiciones	Queda generado el código de seguridad para el banco.

Tabla. 11 Descripción del caso de uso Enviar archivo de seguridad pendiente.

Caso de Uso:	Importar Sucursal
Actores:	Operador de Administración.
Resumen:	El caso de uso inicia cuando el Operador de Administración selecciona la opción Importar Sucursales , el sistema muestra un listado de bancos, el Operador de Administración escoge el banco deseado, busca un fichero con los datos de las sucursales y selecciona la opción "Importar", finalizando así el caso de uso.
Precondiciones:	El Operador de Administración debe estar autenticado en el sistema y deben estar creados los bancos recaudadores.

Anexos

Referencias	RF 7
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1-El Operador de Administración selecciona la opción: “Importar Sucursales” .	2-El sistema muestra una interfaz con el listado de bancos a seleccionar y permite buscar el fichero con los datos de las sucursales y las opciones: <ul style="list-style-type: none"> • Importar • Cerrar
3-El Operador de Administración selecciona el banco, busca el fichero con los datos de las sucursales y selecciona la opción: 3 a) Importar , continúa en la Sección Flujo Normal de Eventos parte 4. 3 b) Cerrar , ver Flujo Alternativo “Cerrar Acción” .	4-El sistema verifica que el banco este seleccionado. En caso contrario ver Flujo Alternativo: “Seleccionar banco” .
	5-El sistema verifica que le fichero haya sido buscado y escogido. En caso contrario ver Flujo Alternativo: “Fichero no escogido” .
	6-El sistema le notifica al Operador de Administración la cantidad de sucursales que se importaron.
Flujos Alternos	
Flujo Alternativo 1 “Cerrar Acción”	
Acción del Actor	Respuesta del Sistema
	1- Se reinician los valores de los campos y se muestra la interfaz principal.
Flujo Alternativo 2:”Seleccionar banco” .	
Acción del Actor	Respuesta del Sistema
	1- El sistema notifica al usuario que debe de seleccionar un banco, para importar las sucursales.

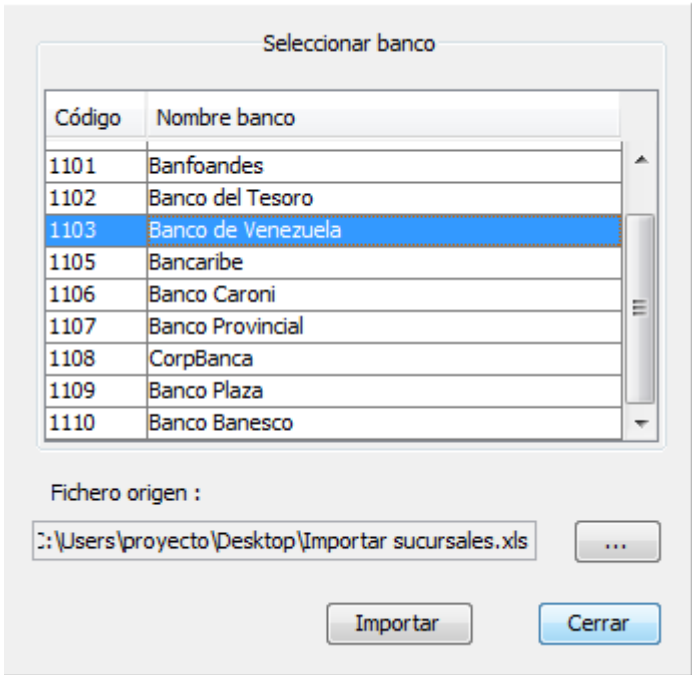
Flujo Alterno 3: "Fichero no escogido".	
Acción del Actor	Respuesta del Sistema
	1- El sistema notifica al usuario que debe buscar y escoger un fichero con los datos de las sucursales, para importar las mismas.
Prototipo de Interfaz	
	
Poscondiciones	Quedan importadas las sucursales a los bancos recaudadores.

Tabla. 12 Descripción del caso de uso Importar sucursales.

Caso de Uso:	Autenticar Usuario
Actores:	Usuario
Resumen:	El caso de uso inicia cuando el sistema se inicia pidiéndole al usuario que inserte su usuario y contraseña, posteriormente el sistema valida los datos y permite que el usuario entre al sistema ya autenticado. Una vez dentro del sistema, el mismo permite también cambiar de usuario, finalizando así el caso de uso.
Precondiciones:	El usuario debe estar creado antes.

Referencias	
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	<p>1- El sistema muestra interfaz con los siguientes campos a llenar:</p> <ul style="list-style-type: none"> • Usuario • Contraseña <p>Y las opciones:</p> <ul style="list-style-type: none"> • Autenticarse • Cerrar
<p>2- El usuario inserta los datos y escoge la opción: Autenticarse, continua en la Sección Flujo Normal de Eventos parte 2.1.</p> <ul style="list-style-type: none"> • 2b) Cerrar, ver Flujo Alterno “Cerrar Acción”. 	<p>3- El sistema valida que los campos no estén vacíos. En caso contrario, ver Flujo Alterno “Campos Vacíos”</p>
	<p>4- El sistema valida también que no existan datos incorrectos. En caso contrario ver Flujo Alterno “Datos Incorrectos”.</p>
	<p>5- El sistema verifica que haya conexión con la base de datos. En caso contrario ver Flujo Alterno “Conexión fallida”.</p>
	<p>6-El sistema muestra la interfaz principal.</p>
Sección 1.1 “Cambiar de Usuario”	
<p>1- El usuario selecciona la opción: “Cambiar de usuario”.</p>	<p>2- El sistema muestra interfaz con los campos:</p> <ul style="list-style-type: none"> • Usuario • Contraseña <p>Y las opciones:</p> <ul style="list-style-type: none"> • Autenticarse

Anexos

	<ul style="list-style-type: none"> • Cerrar
<p>3- El usuario inserta los datos y escoge la opción: Autenticarse, continua en la Sección Flujo Normal de Eventos parte 2.1. 2b) Cerrar, ver Flujo Alterno 1 “Cerrar Acción”.</p>	
Flujos Alternos	
Flujo Alterno 1 “Cerrar Acción”	
Acción del Actor	Respuesta del Sistema
	1- El sistema se cierra.
Flujo Alterno 2 “Campos Vacíos”	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje notificando al usuario que ha dejado campos vacíos.
Flujo Alterno 3 “Datos Incorrectos”	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra un mensaje notificando al usuario que no ha llenado los campos correctamente, ya que los datos solo deben ser letras y números, y deben estar registrados en la base de datos.
Flujo Alterno 4 “Conexión fallida”	
Acción del Actor	Respuesta del Sistema
	1-El sistema muestra un mensaje notificando al usuario que no hay conexión con la base de datos.
Flujo Alterno 5 “Cerrar Acción_1.1”	
Acción del Actor	Respuesta del Sistema
	1- Se reinician los valores de los campos y se muestra la interfaz principal.
Prototipo de Interfaz	

<p>Usuario: <input type="text" value="prueba"/></p> <p>Contraseña: <input type="password" value="••••••"/></p> <p><input type="button" value="Autenticarse"/> <input type="button" value="Cerrar"/></p>	
Poscondiciones	Queda autenticado el usuario.

Tabla. 13 Descripción del caso de uso Autenticar usuario.

Anexos

Anexo 2: Casos de pruebas.

Id del escenario	Escenario	Variable1 Listado de servidores	Variable 2 Nombre Lectura	Variable 3 Fecha Inicial	Variable 4 Fecha final	Variable 5 Hora Lectura	Variable 6 Periodicidad	Variable 7 Habilitada	Variable 8 Crear	Variable 9 Cancelar	Respuesta del Sistema	Resultado de la Prueba
EC 1.2	Flujo Normal de Eventos	Uno de los servidores.	Lectura	12/02/2011	31/02/2011	12am	1 Día	Si	Si	___	Se registra una nueva lectura en el sistema.	Satisfactorio
EC 1.3	Flujo alterno	Uno de los servidores	Lectura	12/02/2011	31/02/2011	12am	1 Día	Si	___	SI	El sistema indica que será cancelada la acción.	Cancelado
EC 1.4	Flujo alterno	Uno de los servidores	Lectura	12/02/2011	31/02/2011	En blanco	1 Día	Si	Si	___	El sistema muestra un mensaje indicando que existen campos vacíos.	Insatisfactorio
EC 1.5	Flujo alterno	Uno de los servidores	Lectura	12/02/2011	31/02/2011	12r	1 Día	Si	Si	___	El sistema muestra un mensaje indicando que existen datos incorrectos.	Insatisfactorio
EC 1.6	Flujo alterno	Uno de los servidores	Lectura	12/02/2011	31/02/2011	12am	1 Día	Si	Si	___	El sistema muestra un mensaje indicando que no existe conexión a la base de datos.	Insatisfactorio

Tabla. 14 Descripción del caso de uso Administrar lecturas de pagos Sección 1 “Crear Lectura”.

Anexos

Id del escenario	Escenario	Variable 1 Listado de servidores	Variable 2 Listado de lecturas	Variable 3 Nombre Lectura	Variable 4 Fecha Inicio	Variable 5 Fecha final	Variable 6 Hora Lectura	Variable 7 Periodicidad	Variable 8 Habilitada	Variable 9 Modificar	Variable 10 Cancelar	Respuesta del Sistema	Resultado de la Prueba
EC 2.1	Flujo Normal de Eventos	Uno de los servidores.	Una de las lecturas.	Lectura	12/02/2011	31/02/2011	12am	3Dias	Si	Si	-----	Se modifica la lectura.	Satisfactorio.
EC 2.2	Flujo alterno	Uno de los servidores.	Una de las lecturas.	Lectura	12/02/2011	31/02/2011	12am	3Dias	Si	---	Si	El sistema indica que será cancelada la acción.	Cancelado
EC 2.3	Flujo alterno	Uno de los servidores.	Una de las lecturas.	Lectura	12/02/2011	En Blanco	12am	3Dias	Si	Si	-----	El sistema muestra un mensaje indicando que existen campos vacíos.	Insatisfactorio
EC .2.4	Flujo alterno	Uno de los servidores.	Una de las lecturas.	Lectura	12/02/2011	31/02/2011	12r	3Dias	Si	Si	-----	El sistema muestra un mensaje indicando que existen datos incorrectos.	Insatisfactorio
EC .2.5	Flujo alterno	Uno de los servidores.	Una de las lecturas.	Lectura	12/02/2011	31/02/2011	12am	3Dias	Si	Si	-----	El sistema muestra un mensaje indicando que se está ejecutando la lectura en estos momentos.	Insatisfactorio

Anexos

EC 2.6	Flujo alterno	Uno de los servidores.	Una de las lecturas.	Lectura	12/02/2011	31/02/2011	12am	1dia	Si	Si	-----	El sistema muestra un mensaje indicando que no existe conexión a la base de datos.	Insatisfactorio
--------	----------------------	------------------------	----------------------	---------	------------	------------	------	------	----	----	-------	--	-----------------

Tabla. 15 Descripción del caso de uso Administrar lecturas de pagos Sección 2 “Modificar Lectura”.

Id del escenario	Escenario	Variable1 Listado de servidores	Variable 2 Listado de lecturas	Variable 4 Cambiar	Variable 5 Cancelar	Respuesta del Sistema	Resultado de la Prueba
EC 3.1	Flujo normal de eventos	Uno de los servidores	Una de las lecturas.	Si	---	Se cambia el estado de la lectura.	Satisfactorio
EC 3.2	Flujo alterno	Uno de los servidores	Una de las lecturas.	---	Si	El sistema muestra un mensaje indicado que será cancelada la acción	Cancelado
EC 3.3	Flujo alterno	Uno de los servidores	Una de las lecturas.	Si	-----	El sistema muestra un mensaje indicando que no existe conexión a la base de datos	Insatisfactorio
EC 3.4	Flujo alterno	Uno de los servidores	Una de las lecturas.	Si	-----	El sistema muestra un mensaje indicando que se está ejecutando la lectura en ese momento	Insatisfactorio

Tabla. 16 Descripción del caso de uso Administrar lecturas de pagos Sección 3 “Habilitar Lectura”.

Anexos

Id del escenario	Escenario	Variable1 Listado de servidores	Variable 1 Listado de lecturas	Variable 2 Eliminar	Variable 3 Cancelar	Respuesta del Sistema	Resultado de la Prueba
EC 4.1	Flujo normal de eventos	Uno de los servidores.	Una de las lecturas.	Si	_____	El sistema elimina los datos de las lecturas seleccionadas.	Satisfactorio
EC 4.2	Flujo alterno	Uno de los servidores	Una de las lecturas.	_____	Si	Se muestra la interfaz principal.	Cancelado
EC 4.3	Flujo alterno	Uno de los servidores	Una de las lecturas.	Si	_____	EL sistema muestra un mensaje indicando que se está ejecutando la lectura en estos momentos.	Insatisfactorio
EC 4.4	Flujo alterno	Uno de los servidores	Una de las lecturas.	Si	_____	El sistema muestra un mensaje indicando que no existe conexión a la base de datos.	Insatisfactorio

Tabla. 17 Descripción del caso de uso Administrar lecturas de pagos Sección 4 “Eliminar Lectura”.

Anexos

Id del escenario	Escenario	Variable 1 Listado de servidores	Variable 2 Listado de lecturas	Variable 3 Ejecutar	Variable 4 Cerrar	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Flujo Normal de Eventos	Uno de los servidores.	Una de las lecturas.	Si	_____	-El sistema ejecuta la lectura seleccionada correctamente.	Satisfactorio
EC 1.2	Flujo Alternativo de Eventos.	Uno de los servidores.	Una de las lecturas.	_____	Si	Se reinician los valores de los campos y se muestra la interfaz principal.	Cancelado
EC 1.2	Flujo Alternativo de Eventos.	Uno de los servidores.	Una de las lecturas.	Si	_____	- El sistema muestra un mensaje indicando que existió un error en la ejecución de la lectura	Insatisfactorio

Tabla. 18 Ejecutar lectura manual.

Anexos

Id del escenario	Escenario	Variable 1 Iniciar	Variable 2 Cerrar	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Flujo Normal de Eventos	SI	-----	<ul style="list-style-type: none">- El sistema verifica que exista conexión con el servidor web y que el servicio solicitado esté corriendo mostrando un mensaje satisfactorio diciendo que se inició correctamente la ejecución de la lectura.- El sistema permite seleccionar las siguientes opciones:<ul style="list-style-type: none">• Pausa• Detener• Cerrar- El Operador de Administración selecciona la opción deseada.	Satisfactorio
EC 1.2	Flujo Alternativo de Eventos.	-----	SI	<ul style="list-style-type: none">- El sistema permite que el servicio se siga realizando y se muestra la interfaz principal.	Cancelado
EC 1.3	Flujo Alternativo de Eventos.	SI	-----	<ul style="list-style-type: none">- El sistema muestra un mensaje diciendo que no existe conexión con el servidor web o que el servicio solicitado no se está corriendo.	Insatisfactorio

Tabla. 19 Ejecutar lectura automática sección 1 “Iniciar”.

Anexos

Id del escenario	Escenario	Variable 1 Pausa	Variable 2 Reiniciar	Variable 4 Cerrar	Respuesta del Sistema	Resultado de la Prueba
EC 2.1	Flujo Normal de Eventos	SI	-----	-----	<p>El sistema verifica que exista conexión con el servidor web y que el servicio solicitado esté corriendo mostrando un mensaje satisfactorio diciendo que se para correctamente la ejecución de la lectura</p> <p>- El sistema permite seleccionar las siguientes opciones:</p> <ul style="list-style-type: none"> • Reiniciar • Detener • Cerrar <p>- El Operador de Administración selecciona la opción deseada.</p>	Satisfactorio
EC 2.2	Flujo Alternativo de Eventos.	-----	-----	Si	- El sistema permite que el servicio se siga realizando y se muestra la interfaz principal.	Cancelado
EC 2.3	Flujo Alternativo de Eventos.	. Si	-----	---	- El sistema muestra un mensaje diciendo que no existe conexión con el servidor web o que el servicio solicitado no se está corriendo.	Insatisfactorio

Tabla. 20 Ejecutar lectura automática sección 2 “Pausa”.

Anexos

Id del escenario	Escenario	Variable 1 Detener	Variable 2 Reiniciar	Variable4 Cerrar	Respuesta del Sistema	Resultado de la Prueba
EC 3.1	Flujo Normal de Eventos	Si	-----	-----	<p>-El sistema verifica que exista conexión con el servidor web y que el servicio solicitado esté corriendo mostrando un mensaje satisfactorio diciendo que se detuvo correctamente la ejecución de la lectura.</p> <p>- El sistema permite seleccionar las siguientes opciones:</p> <ul style="list-style-type: none"> • Iniciar • Cerrar <p>- El Operador de Administración selecciona la opción deseada.</p>	Satisfactorio
EC 3.2	Flujo Alterno de Eventos.	-----	-----	Si	- El sistema permite que el servicio se siga realizando y se muestra la interfaz principal.	Cancelar
EC 3.3	Flujo Alterno de Eventos.	. Si	-----	---	- El sistema muestra un mensaje diciendo que no existe conexión con el servidor web o que el servicio solicitado no se está corriendo.	Insatisfactorio

Tabla. 21 Ejecutar lectura automática sección 3 “Detener”.

Anexos

Id del escenario	Escenario	Variable 1 Reiniciar	Variable 2 Pausa	Variable 4 Cerrar	Respuesta del Sistema	Resultado de la Prueba
EC 2.1	Flujo Normal de Eventos	Si	-----	-----	<p>- El sistema verifica que exista conexión con el servidor web y que el servicio solicitado esté corriendo mostrando un mensaje satisfactorio diciendo que se Reinició correctamente la ejecución de la lectura.</p> <p>- El sistema permite seleccionar las siguientes opciones:</p> <ul style="list-style-type: none"> • Pausa • Detener • Cerrar <p>- El Operador de Administración selecciona la opción deseada.</p>	Satisfactorio
EC 3.2	Flujo Alternativo de Eventos.	-----	-----	Si	- El sistema permite que el servicio se siga realizando y se muestra la interfaz principal.	Cancelado
EC 4.3	Flujo Alternativo de Eventos.	. Si	-----	---	- El sistema muestra un mensaje diciendo que no existe conexión con el servidor web o que el servicio solicitado no se está corriendo.	Insatisfactorio

Tabla. 22 Ejecutar lectura automática sección 4 “Reiniciar”.

Anexos

Id del escenario	Escenario	Variable 1 Usuario	Variable 2 Contraseña	Variable 3 Autenticarse	Variable 4 Cerrar	Respuesta del sistema.	Resultado de la Prueba
EC 1.1	Flujo Normal de Eventos	Juan	bancos123	Si	-----	-El usuario de auténtica y el sistema muestra la interfaz principal.	Satisfactorio
EC 1.2	Flujo Alternativo de Eventos.	Juan	bancos123	-----	Si	-El sistema se cierra.	Cancelado
EC 1.3	Flujo Alternativo de Eventos.	Juan	En blanco	Si	-----	-- El sistema muestra un mensaje notificando al usuario que ha dejado campos vacíos.	Insatisfactorio
EC 1.4	Flujo Alternativo de Eventos.	*/**/5665*//	bancos123	Si	-----	- El sistema muestra un mensaje notificando al usuario que no ha llenado los campos correctamente, ya que los usuarios solo deben tener letras y números, y deben estar registrados en la base de datos.	Insatisfactorio
EC 1.5	Flujo Alternativo de Eventos.	Juan	bancos123	Si	-----	- El sistema muestra un mensaje notificando al usuario que no hay conexión con la base de datos	Insatisfactorio

Tabla. 23 Autenticar usuario sección 1 “Autenticar”.

Anexos

Id del escenario	Escenario	Variable 1 Usuario	Variable 2 Contraseña	Variable3 Autenticarse	Variable4 Cerrar	Respuesta del Sistema	Resultado de la Prueba
EC 1.1.1	Flujo Normal de Eventos	Pedro	bancos123	15- Si	-----	-El usuario de auténtica y el sistema muestra la interfaz principal.	Satisfactorio
EC1.1.2	Flujo Alternativo de Eventos.	Pedro	bancos123	-----	Si	Se reinician los valores de los campos y se muestra la interfaz principal.	Cancelado
EC 1.1.3	Flujo Alternativo de Eventos.	Pedro	En blanco	17- Si	-----	-- El sistema muestra un mensaje notificando al usuario que ha dejado campos vacíos.	Insatisfactorio
EC1.1.4	Flujo Alternativo de Eventos.	*/**/5665*//	bancos123	19- Si	-----	- El sistema muestra un mensaje notificando al usuario que no ha llenado los campos correctamente, ya que los datos solo deben ser letras y números, y deben estar registrados en la base de datos.	Insatisfactorio
EC 1.1.5	Flujo Alternativo de Eventos.	Pedro	bancos123	21- Si	-----	- El sistema muestra un mensaje notificando al usuario que no hay conexión con la base de datos.	Insatisfactorio

Tabla. 24 Autenticar usuario sección 2 “cambiar Usuario”.

Anexos

Id del escenario	Escenario	Variable 1 Listado de Bancos	Variable 2 Escoger Fichero	Variable 3 Importar	Variable 4 Cerrar	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Flujo Normal de Eventos	Si	Si	Si	----	-Se importan las sucursales, en dependencia del banco seleccionado y el fichero escogido, y se muestra un mensaje que notifica al Operador de Administración de las sucursales que se importaron.	Satisfactorio
EC 1.2	Flujo Alternativo de Eventos	Si	Si	-----	Si	- Se reinician los valores de los campos y se muestra la interfaz principal.	Cancelado
EC 1.3	Flujo Alternativo de Eventos	En blanco	Si	Si	-----	-Se muestra un mensaje notificando al usuario que debe de seleccionar un banco, para importar las sucursales.	Insatisfactorio
EC 1.4	Flujo Alternativo de Eventos	Si	En blanco	Si	-----	-Se muestra un mensaje que notifica al usuario que debe buscar y escoger un fichero con los datos de las sucursales.	Insatisfactorio

Tabla. 25 Importar Sucursales.

Anexos

<i>Id del escenario</i>	<i>Escenario</i>	<i>Variable 1 Listado de bancos</i>	<i>Variable 2 IP</i>	<i>Variable 3 usuario</i>	<i>Variable 4 Contraseña</i>	<i>Variable 5 puerto</i>	<i>Variable 6 Conectar</i>	<i>Variable 7 Guardar</i>	<i>Variable 8 Cerrar</i>	<i>Respuesta del Sistema</i>	<i>Resultado de la Prueba</i>
EC 1.1	Flujo Normal de Eventos	Uno de los bancos	10.8.150.13	sib	sib	22	Si	-----	_____	-El sistema muestra un mensaje satisfactorio al conectarse al servidor del banco seleccionado.	Satisfactorio
	Flujo Normal de Eventos	Uno de los bancos	10.8.150.13	sib	sib	22	_____	Si	_____	El sistema muestra un mensaje satisfactorio al guardar los datos del servidor en la base de datos.	Insatisfactorio
EC 1.2	Flujo Alternativo de Eventos.	Uno de los bancos	10.8.150.13	sib	sib	22	_____	_____	Si	Se reinician los valores de los campos y se muestra la interfaz principal.	Cancelado
EC 1.3	Flujo Alternativo de Eventos.	Uno de los bancos	En blanco	sib	sib	22	Si	Si	_____	El sistema muestra un mensaje notificando al usuario que ha dejado campos sin llenar.	Insatisfactorio

Anexos

EC 1.4	Flujo Alternativo de Eventos.	Uno de los bancos	10.8.150.13	sib	sib	22rr	Si	Si	_____	- El sistema muestra un mensaje notificando al usuario que no ha llenado los campos correctamente.	Insatisfactorio.
EC 1.5	Flujo Alternativo de Eventos.	Uno de los bancos	10.8.150.14	sib	sib	22	Si	_____	_____	- El sistema muestra un mensaje indicando que no existe conexión al servidor del banco seleccionado.	Insatisfactorio
EC 1.6	Flujo Alternativo de Eventos.	Uno de los bancos	10.8.150.13	sib	sib	22	_____	Si	_____	-El sistema muestra un mensaje indicando que existe error al guardar los datos.	Insatisfactorio

Tabla. 26 Configurar servidor.

Anexos

Id del escenario	Escenario	Variable 1 Fecha de comienzo de uso	Variable 2 Generar	Variable 3 Cerrar	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Flujo Normal de Eventos	09/03/2011	Si	_____	-El sistema genera el fichero en formato XML - El sistema guarda estos códigos de seguridad como no notificados. - El sistema guarda los ficheros generados en el sftp para cada uno de los bancos.	Satisfactorio
EC 1.2	Flujo Alternativo de Eventos.	09/03/2011	_____	Si	- Se reinician los valores de los campos y se muestra la interfaz principal.	Cancelado.
EC 1.3	Flujo Alternativo de Eventos.	09/03/2011	Si	-----	El sistema hace una copia local del fichero XML que no se pudo enviar al banco y se le notifica por vía correo de lo sucedido para enviarlo posteriormente, mostrándose la interfaz principal.	Insatisfactorio

Tabla. 27 generar archivo seguridad.

Anexos

Id del escenario	Escenario	Variable 1 Listado de ficheros sin enviar	Variable 2 Enviar	Variable 3 Cerrar	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Flujo Normal de Eventos	Un fichero	Si	_____	<ul style="list-style-type: none"> - El sistema envía el archivo de seguridad al banco. - El sistema registra el archivo como enviado en la base de datos y se elimina la lista de pendientes. - El sistema muestra un mensaje satisfactorio. 	Satisfactoria
EC 1.2	Flujo Alternativo de Eventos.	Un fichero	_____	Si	- Se reinician los valores de los campos y se muestra la interfaz principal.	Insatisfactorio.
EC 1.3	Flujo Alternativo de Eventos.	Un fichero	Si	-----	El sistema le muestra un mensaje al usuario que no se pudo enviar el archivo.	

Tabla. 28 Enviar archivo seguridad pendiente.

Anexos

Anexo 3: Diagrama de clases del análisis.

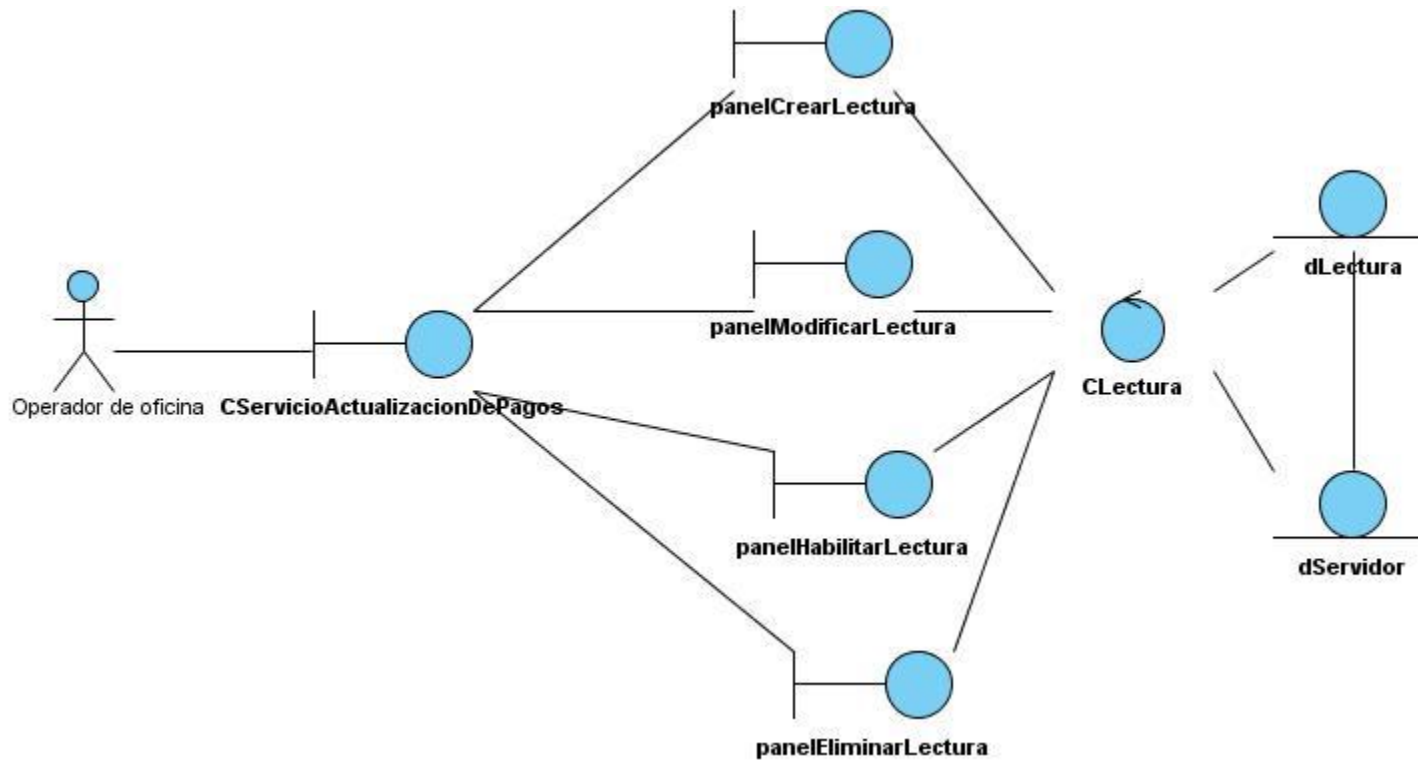


Figura. 12 DCA_Administrar lecturas de pagos.

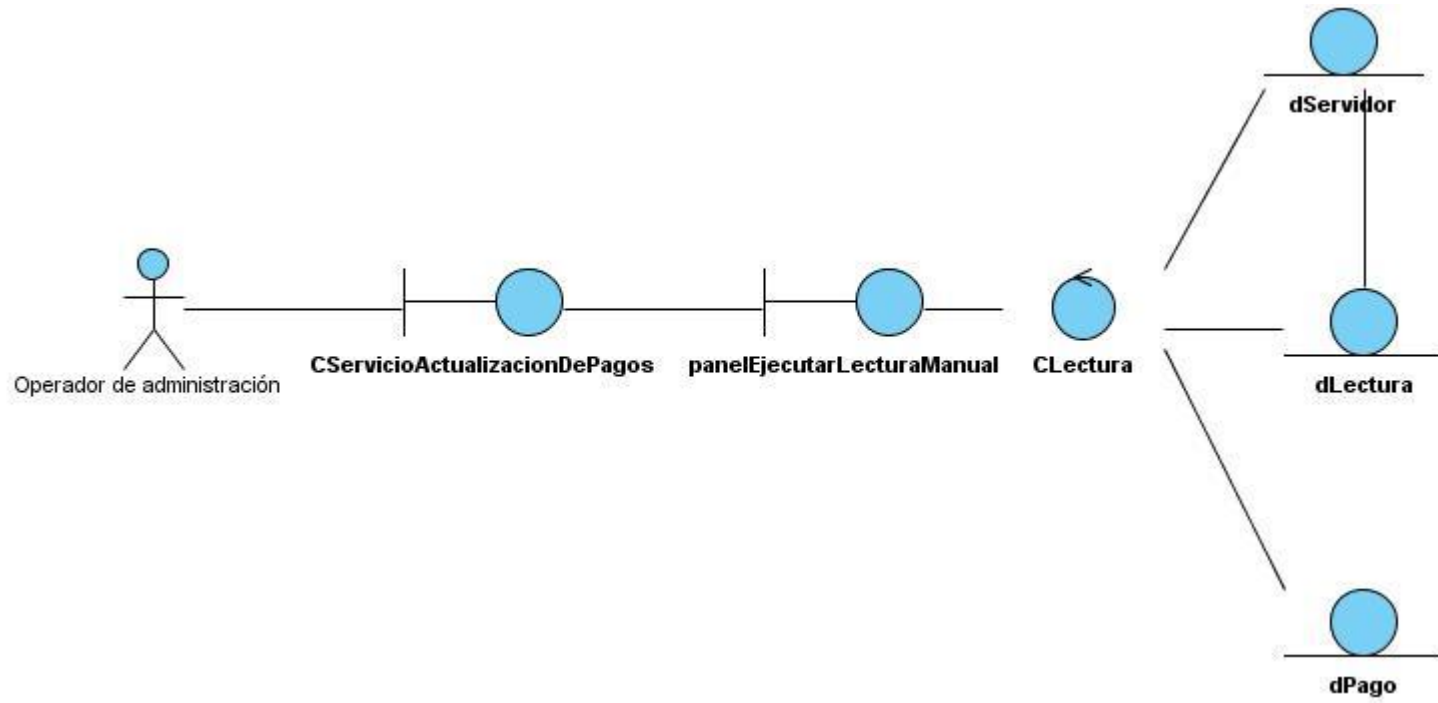


Figura. 13 DCA_Ejecutar lectura manual.



Figura. 14 DCA_Ejecutar lectura automática.

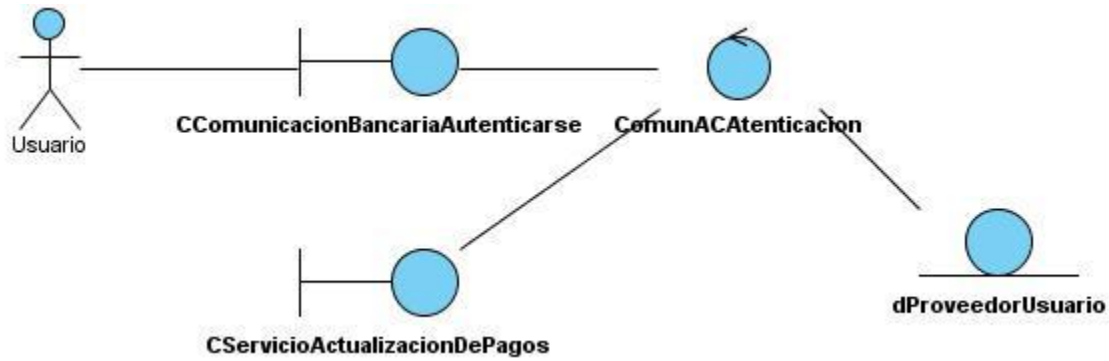


Figura. 15 DCA_Autenticar usuario Gestión de comunicación bancaria.

Anexos

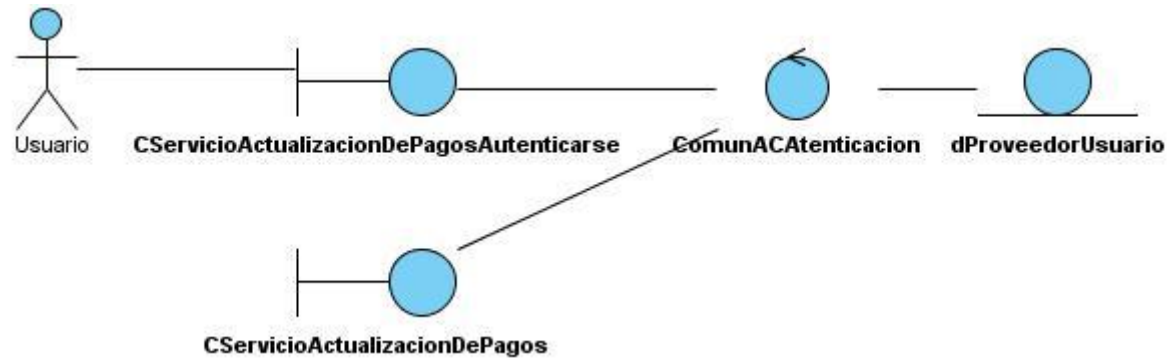


Figura. 16 DCA_Autenticar usuario Servicio de actualización de pagos.

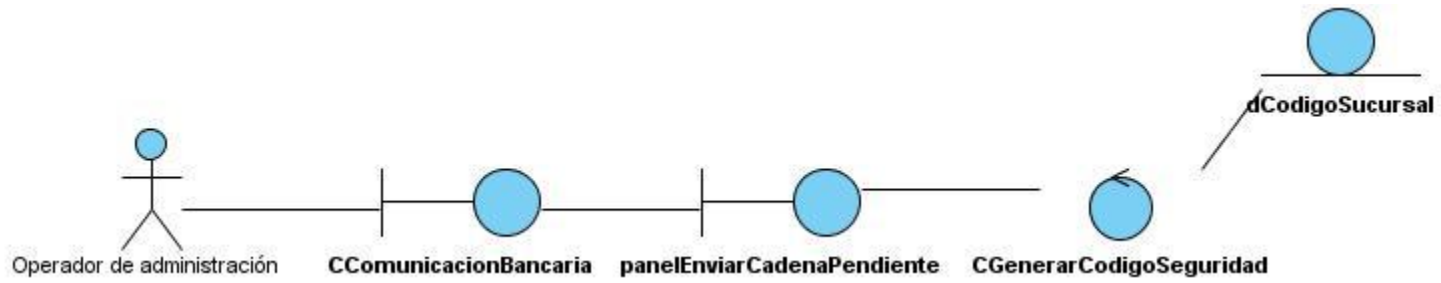


Figura. 17 DCA_Enviar archivo seguridad pendiente.

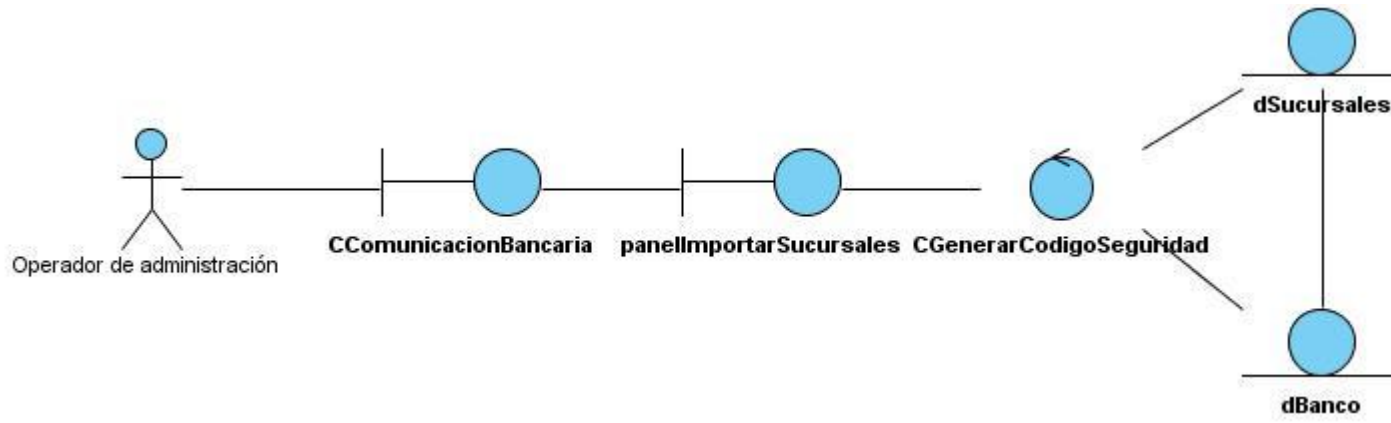


Figura. 18 DCA_Importar sucursales.

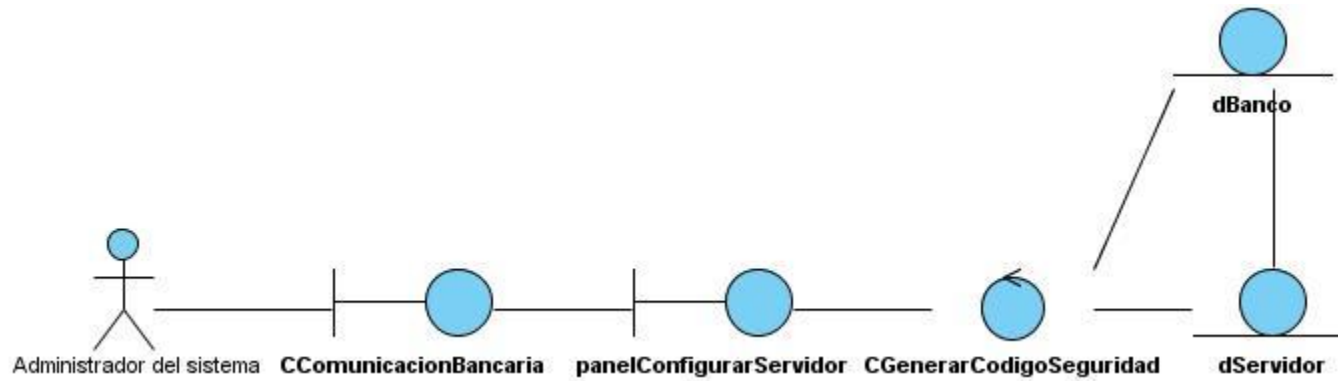


Figura. 19 DCA_ Configurar servidor SFTP.

Anexos

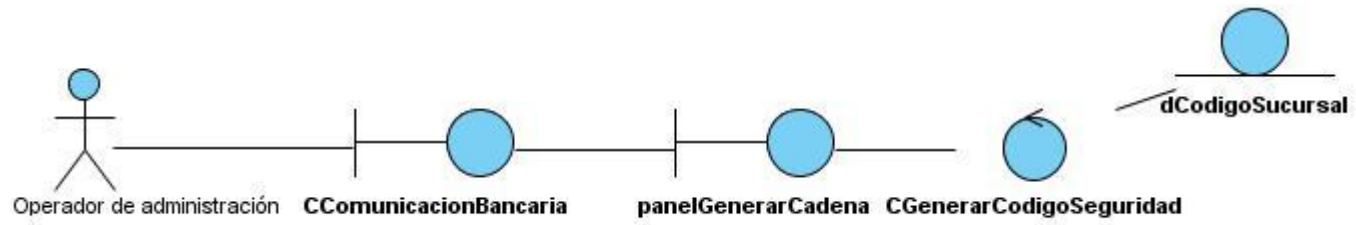


Figura. 20 DCA_Generar archivo de seguridad.

Anexo 4: Diagrama de clases del Diseño.

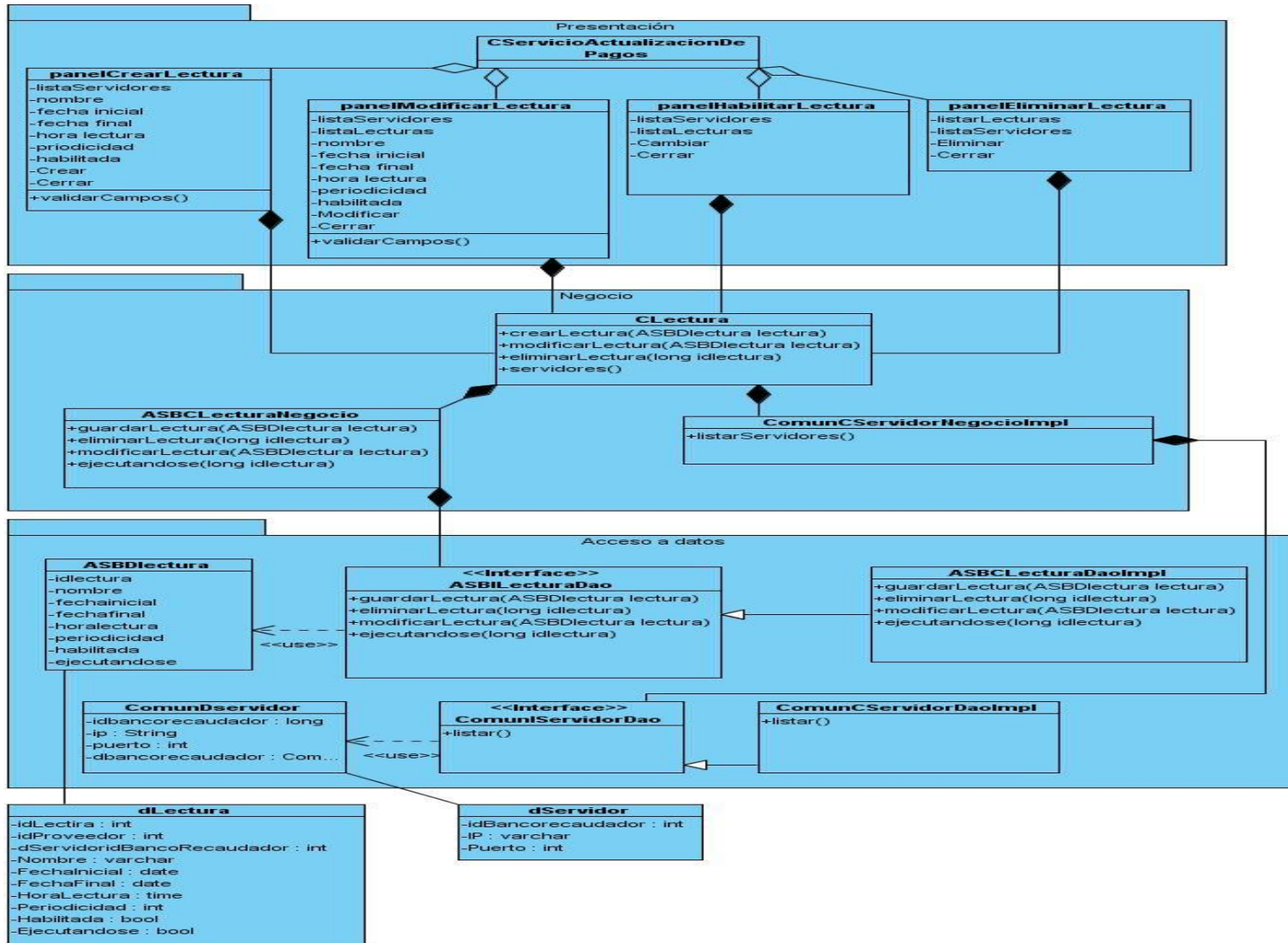


Figura. 21 DCD_Administrar lecturas de pago.

Anexos

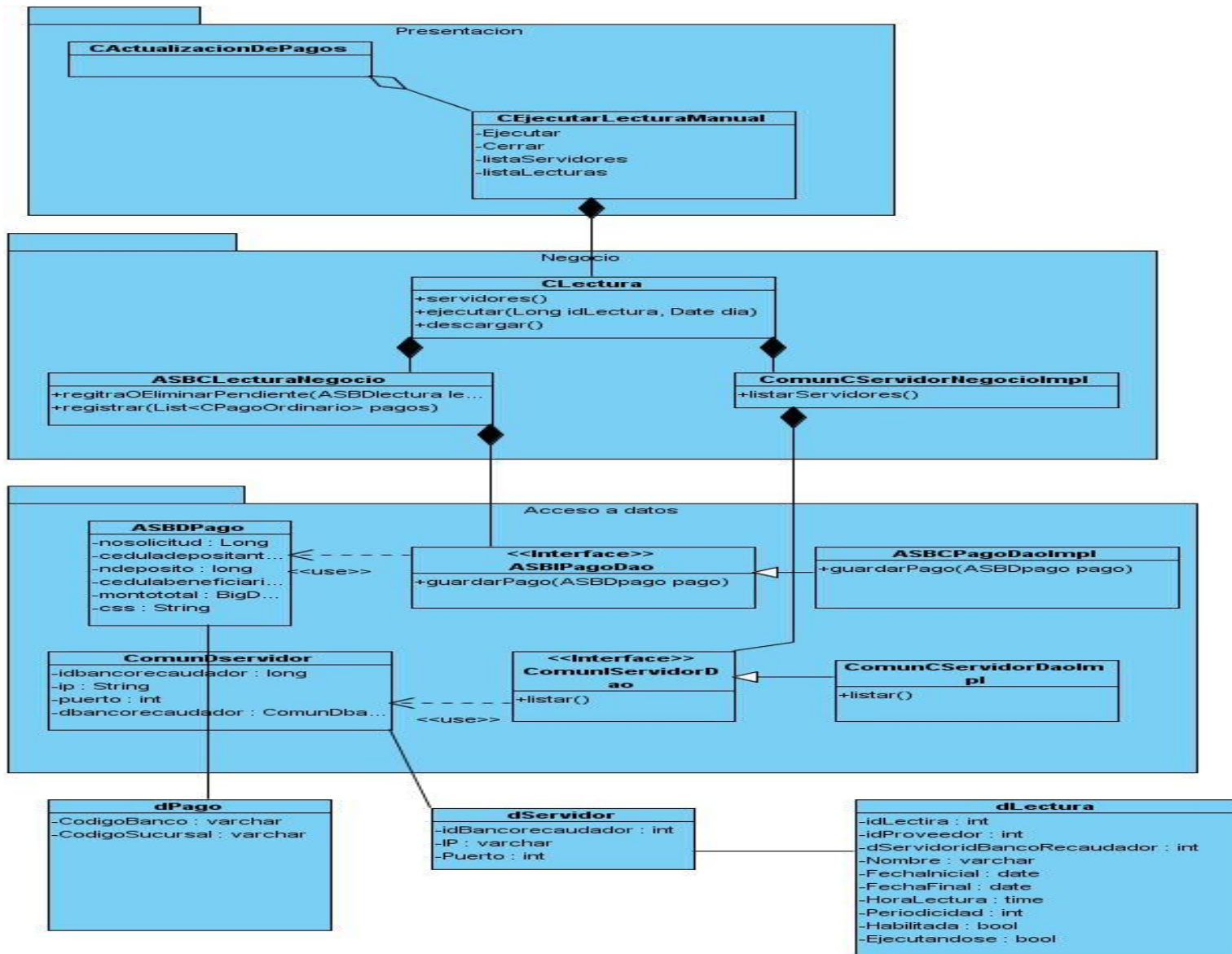


Figura. 22 DCD_Ejecutar lectura manual.

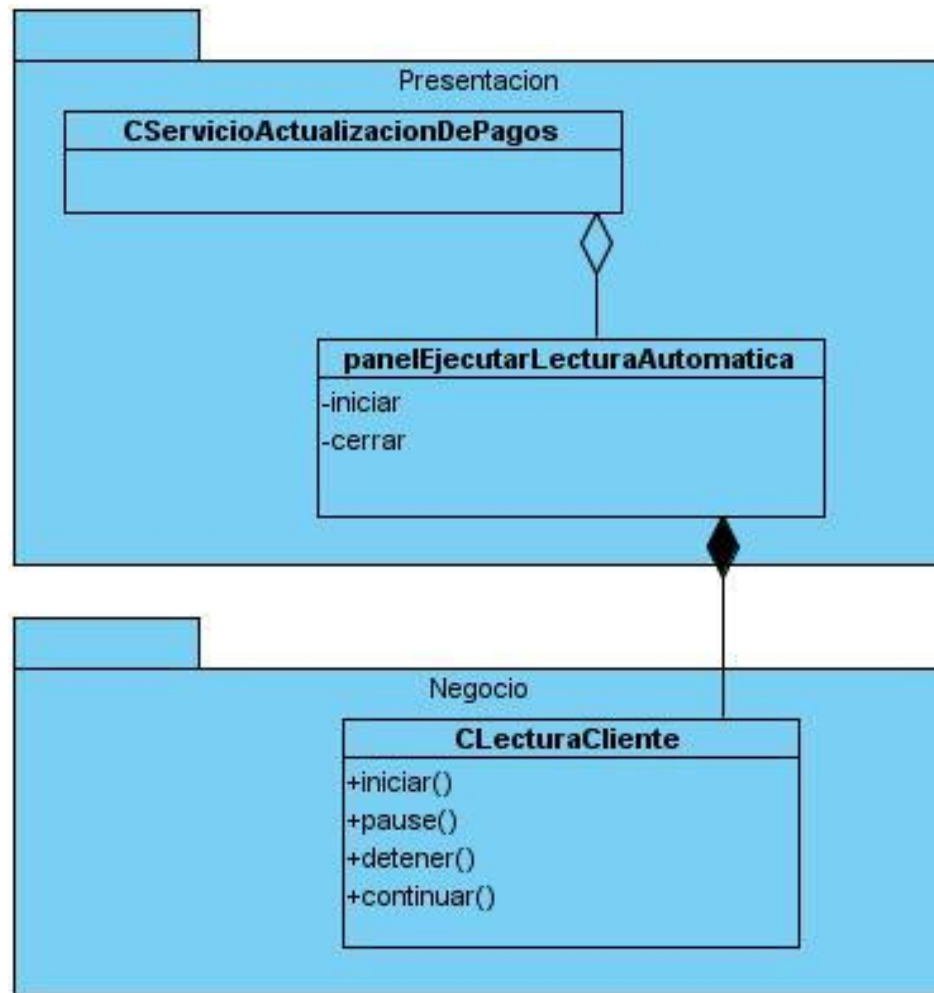


Figura. 23 DCD_Ejecutar lectura automática.

Anexos

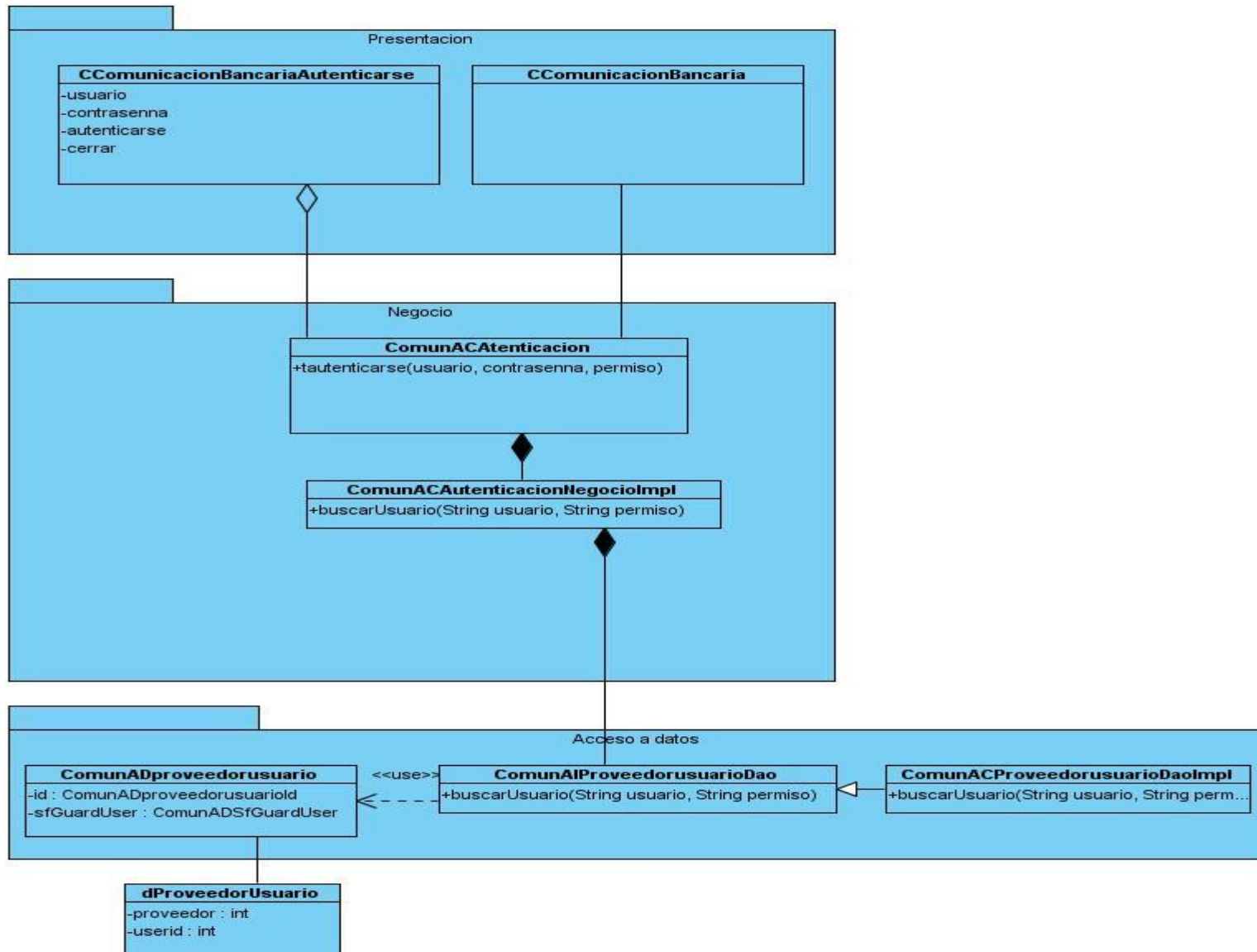


Figura. 24 DCD_Autenticar usuario Gestión de comunicación bancaria.

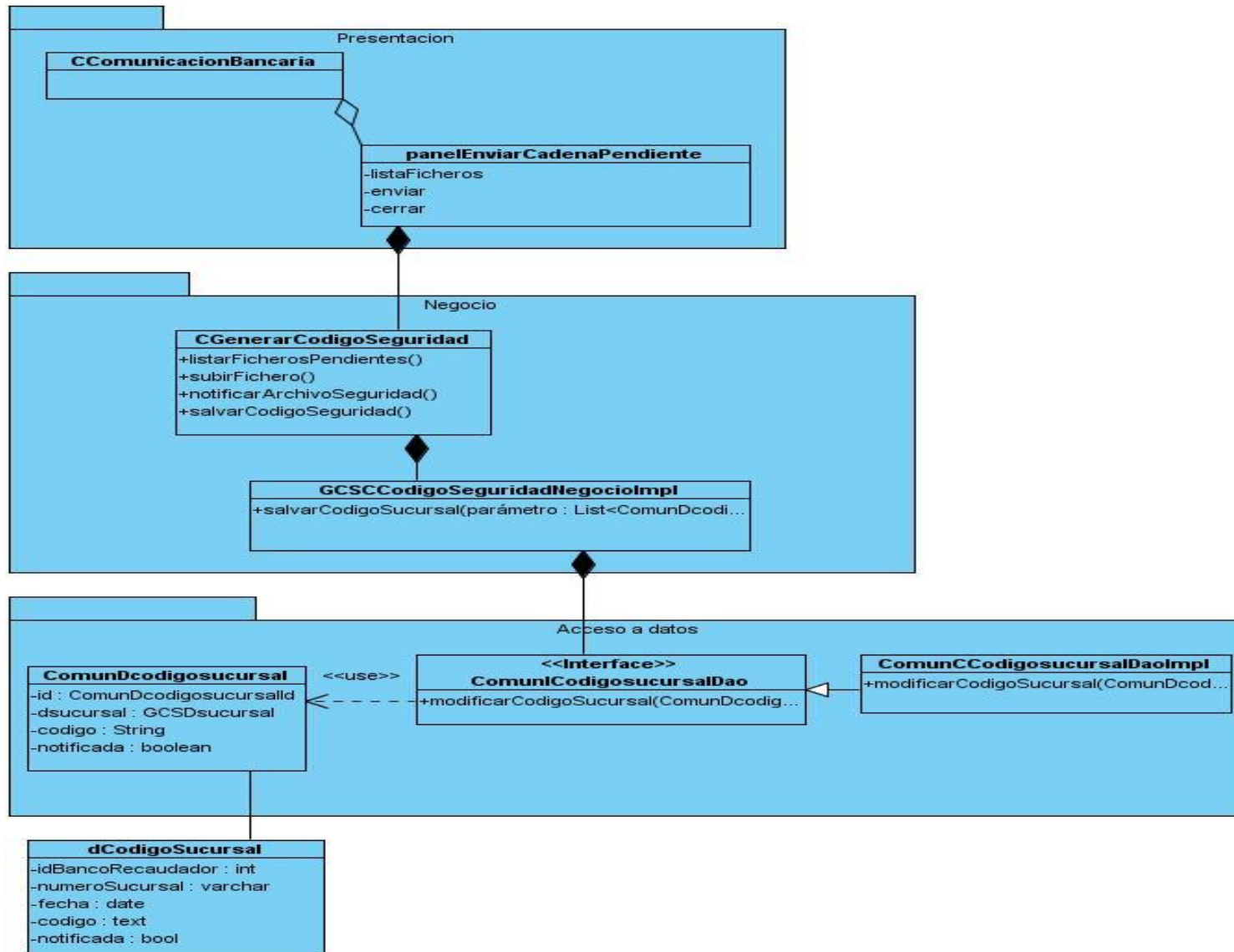


Figura. 25 DCD_Enviar archivo seguridad pendiente.

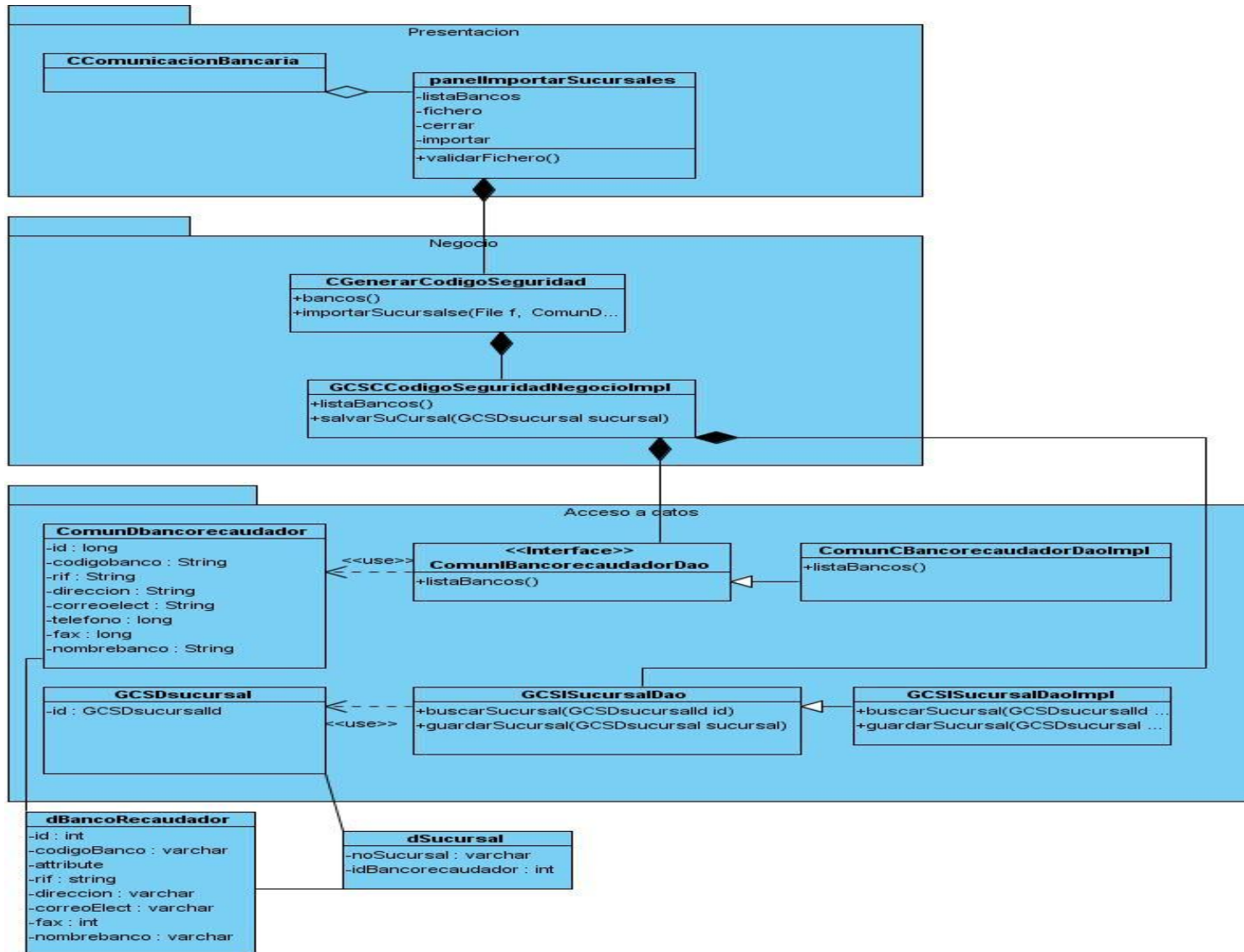


Figura. 26 DCD_Importar sucursales.

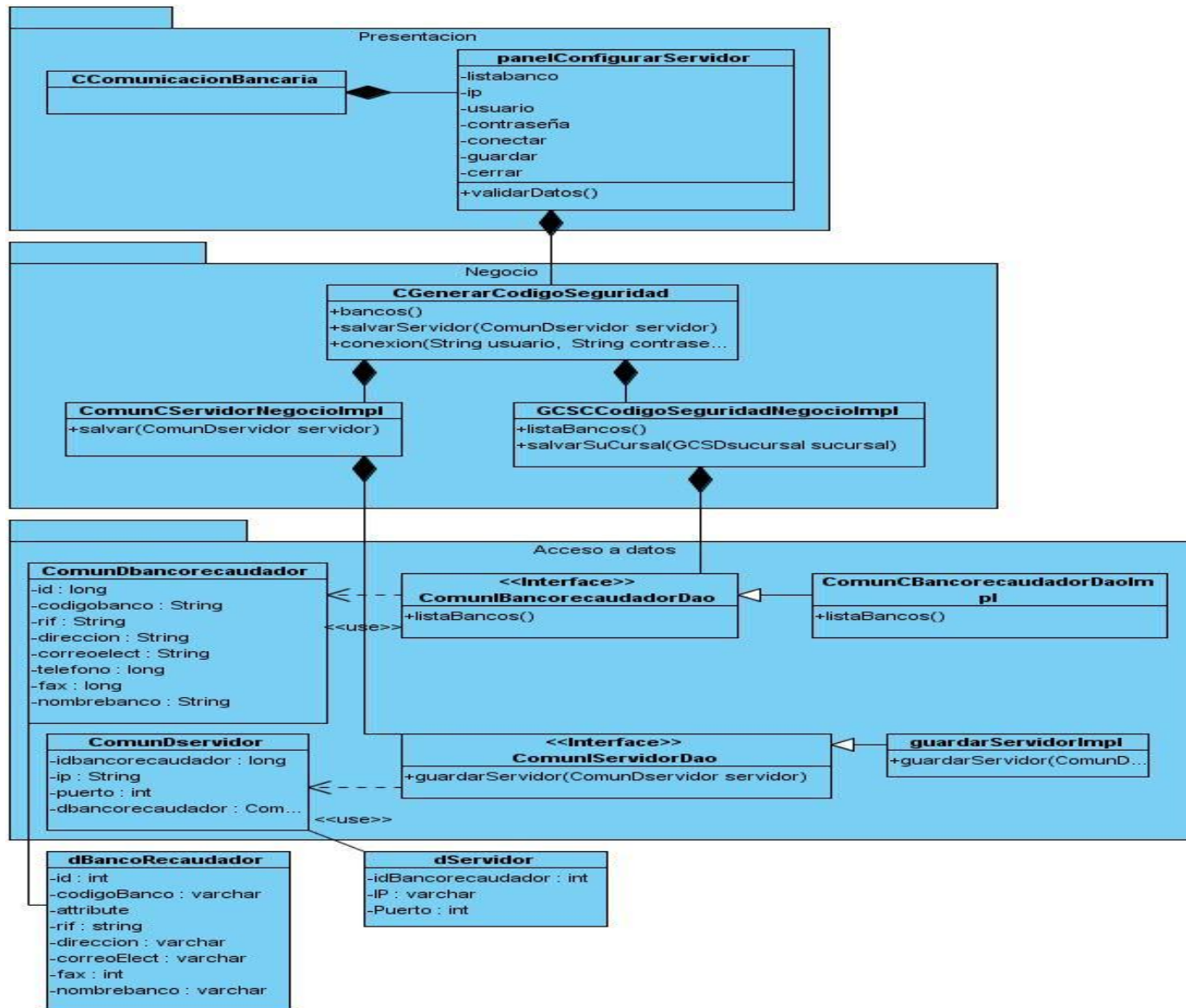


Figura. 27 DCD_Configurar servidor SFTP.

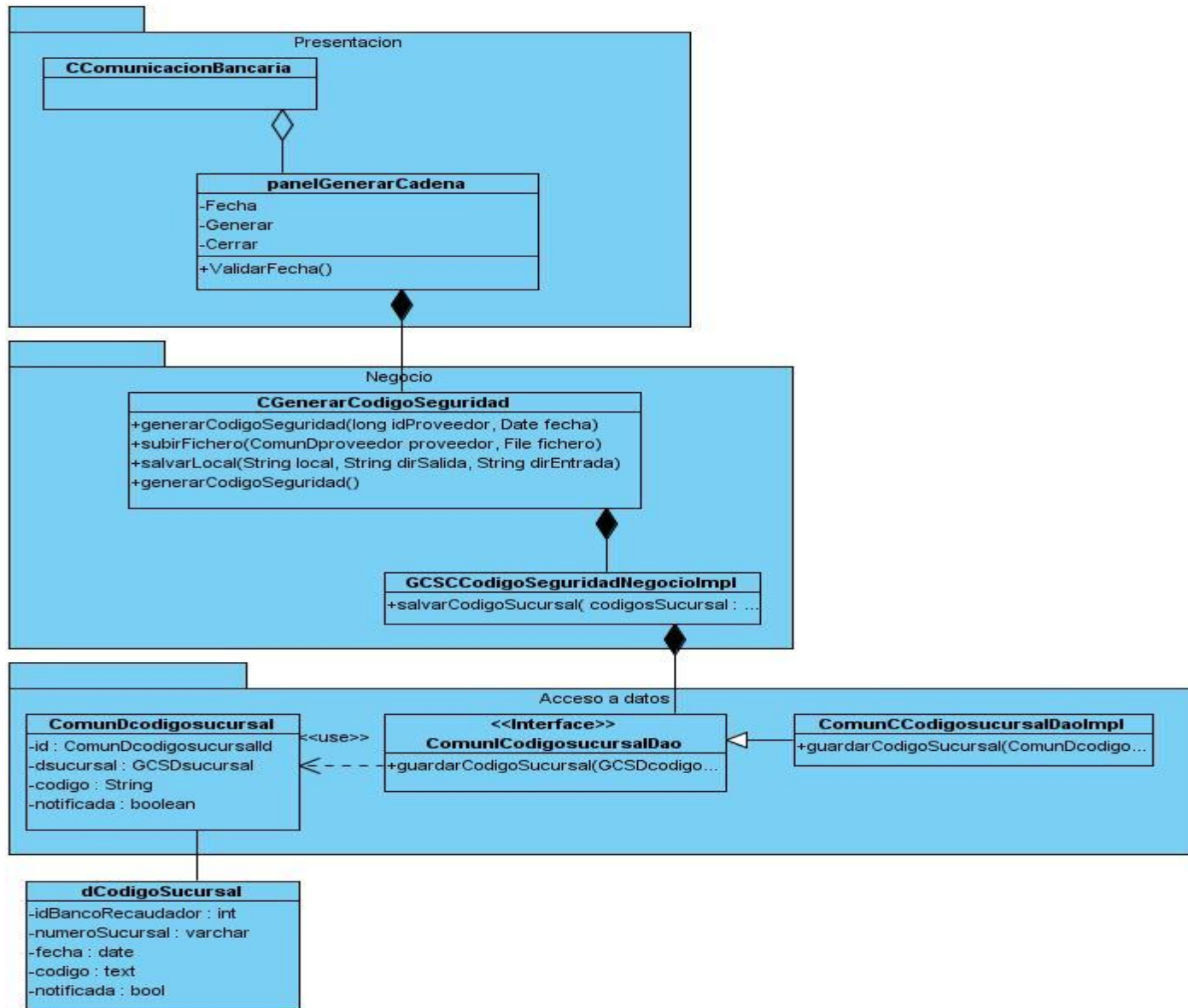


Figura. 28 DCD_Generar archivo seguridad.

Anexo 5: Diagrama de secuencia del diseño.

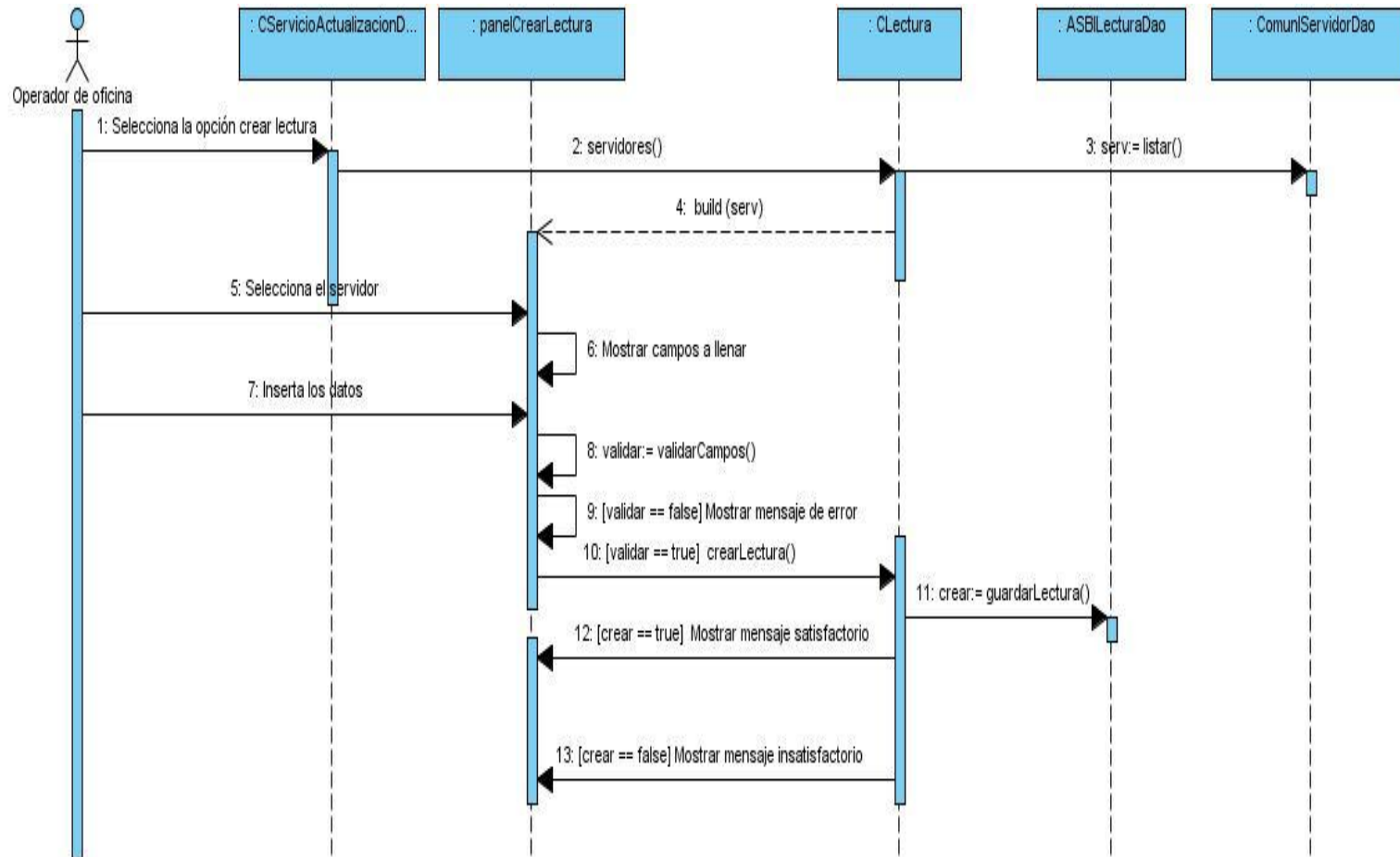


Figura. 29 DSD_Administrar lecturas de pago "Crear lectura."

Anexos

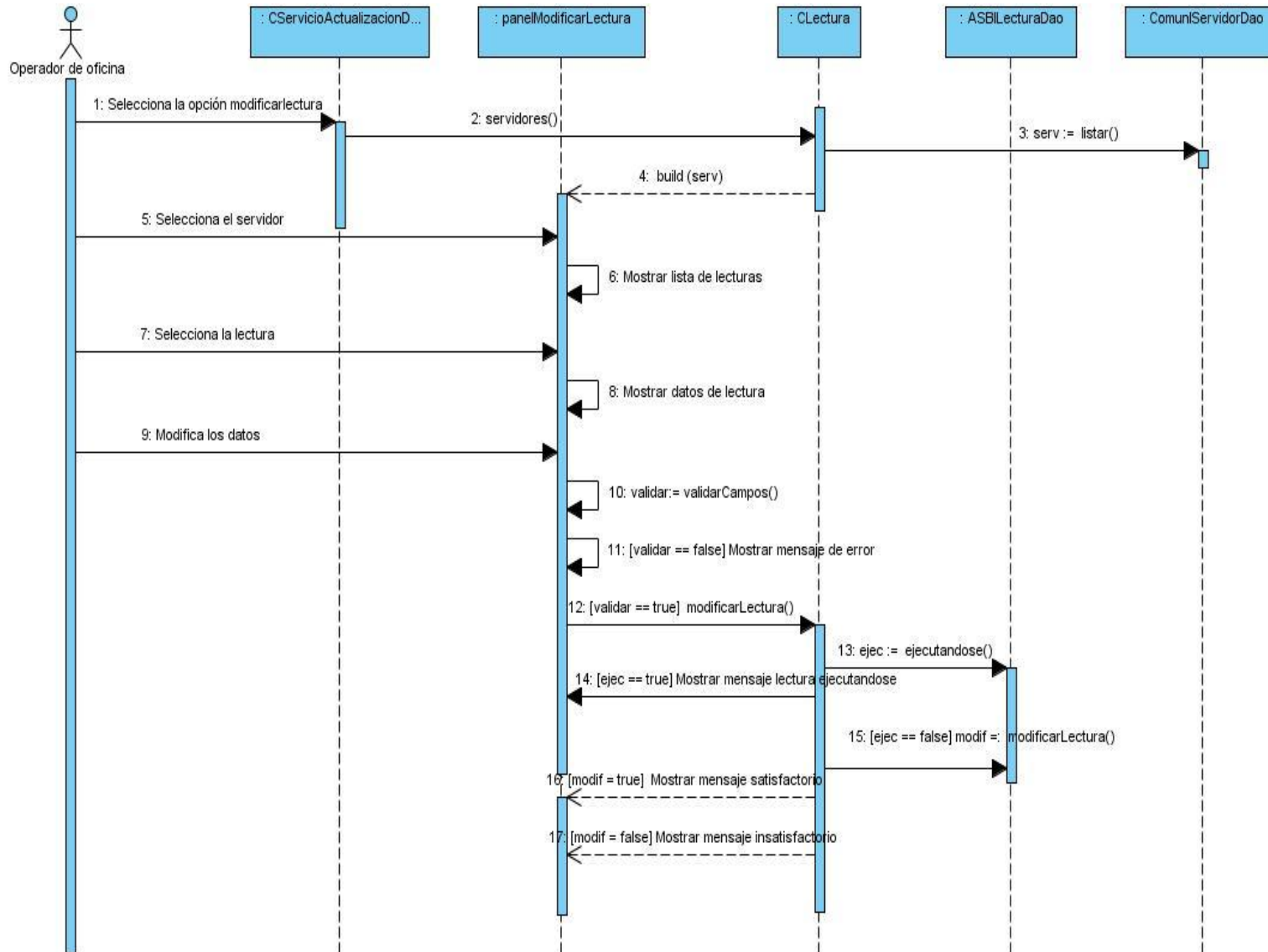


Figura. 30 DSD_Administrar lecturas de pago "Modificar lectura."

Anexos

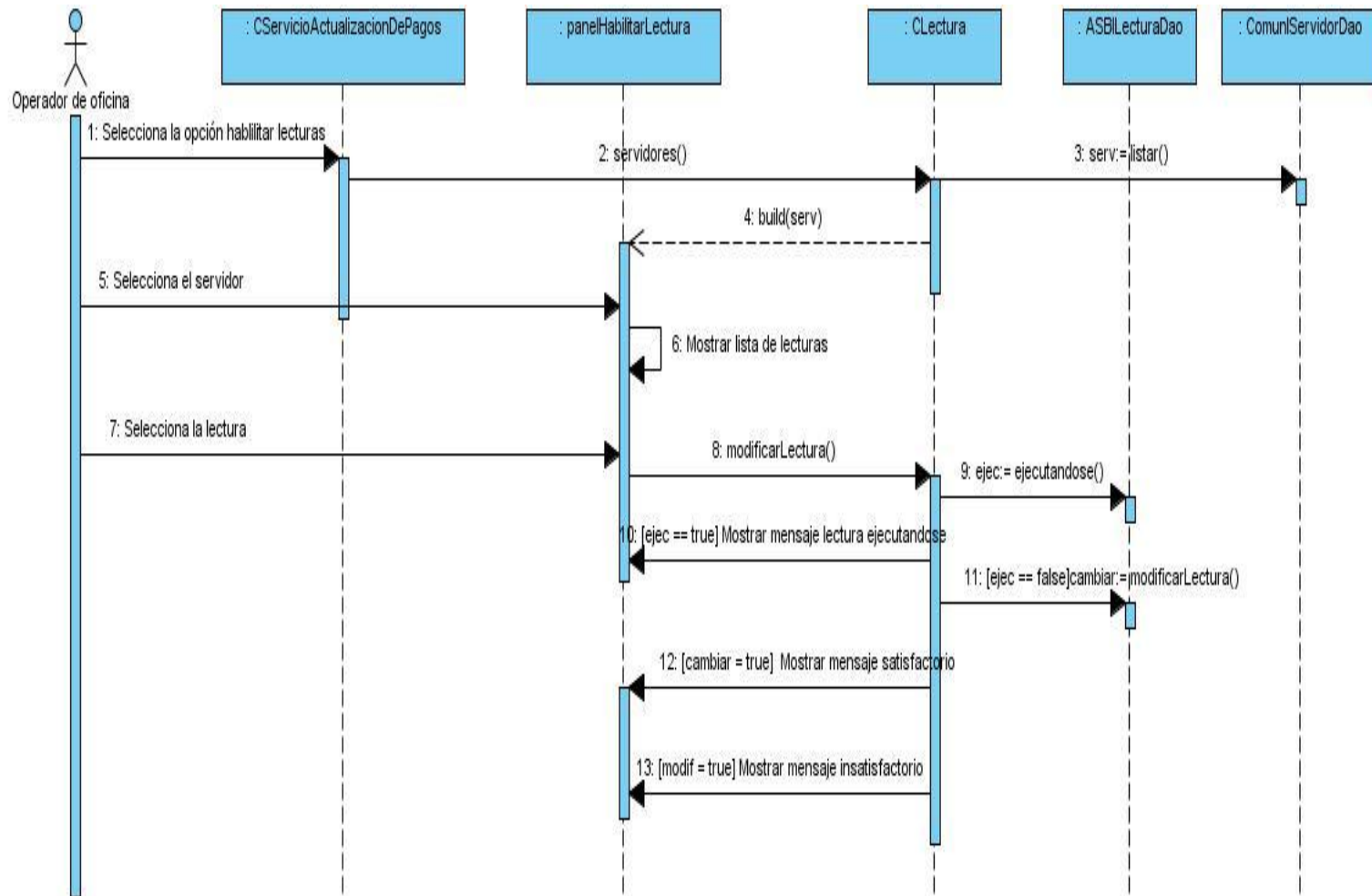


Figura. 31 DSD_Administrar lecturas de pago "Habilitar lectura."

Anexos

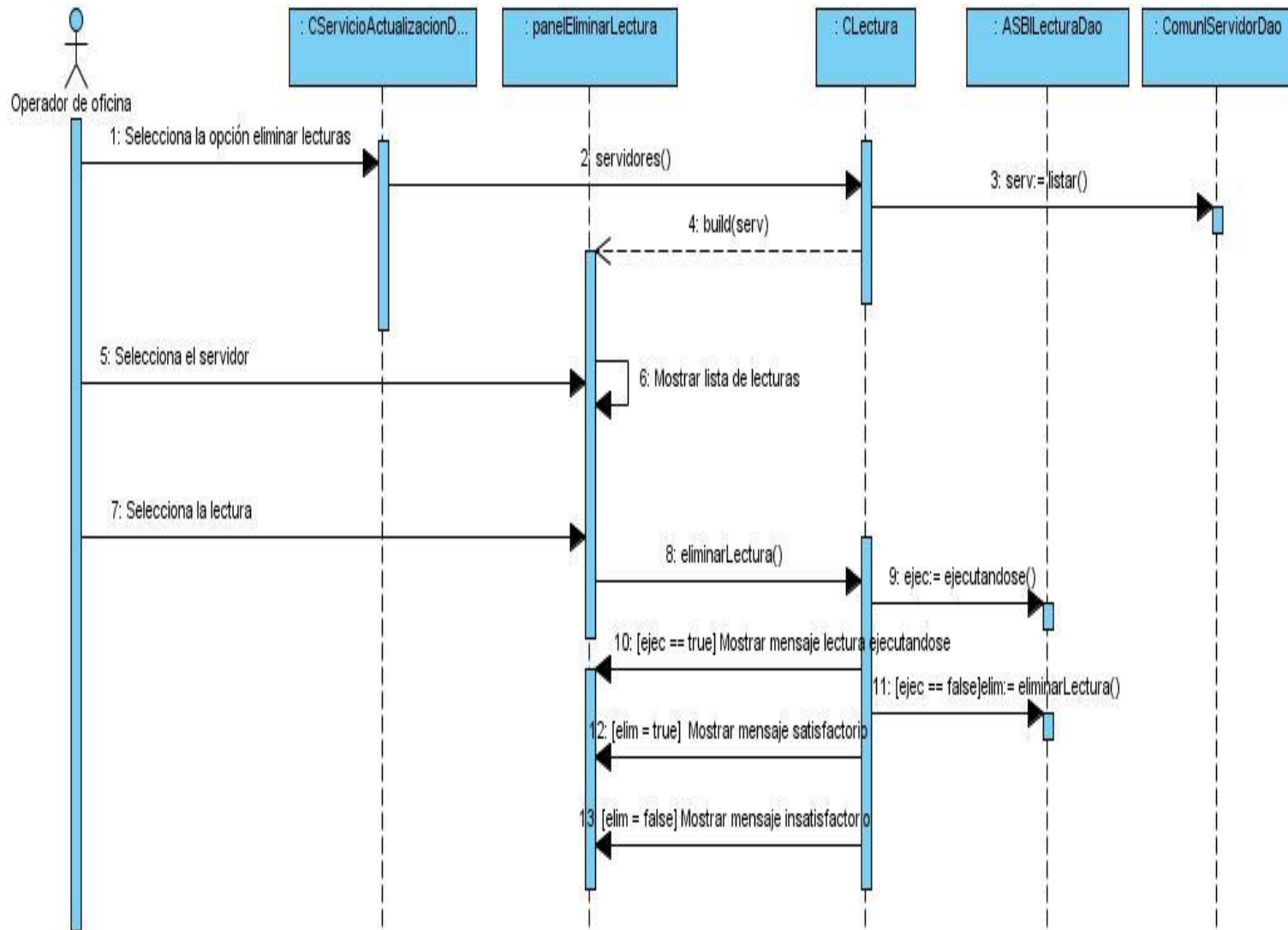


Figura. 32 DSD_Administrar lecturas de pago "Eliminar lectura."

Anexos

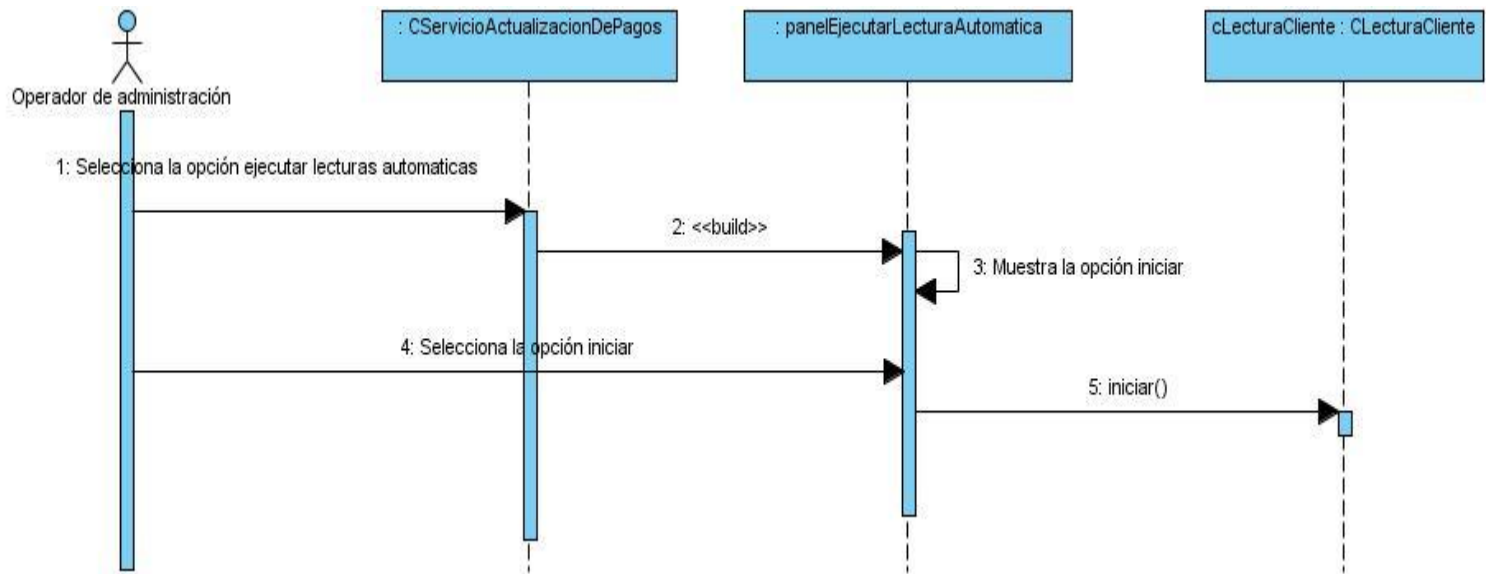


Figura. 33 DSD_Ejecutar lectura automática "Iniciar."

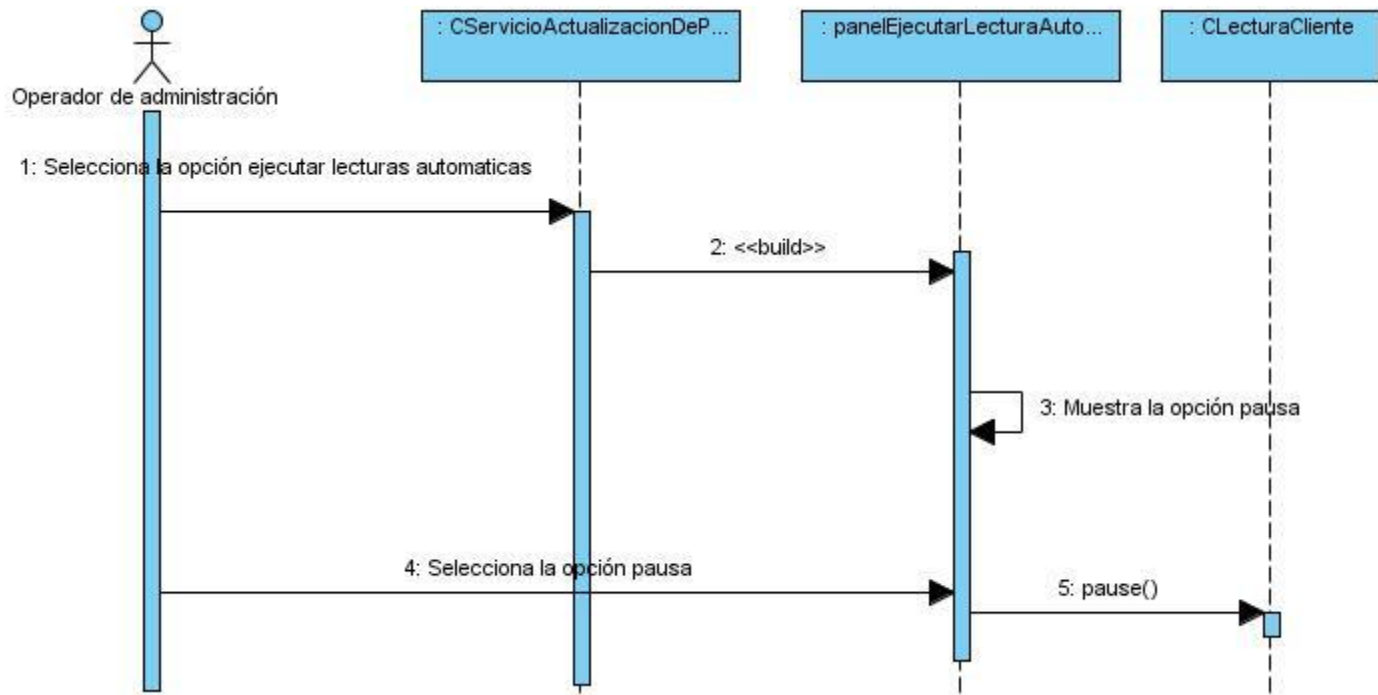


Figura. 34 DSD_Ejecutar lectura automática "Pausa."

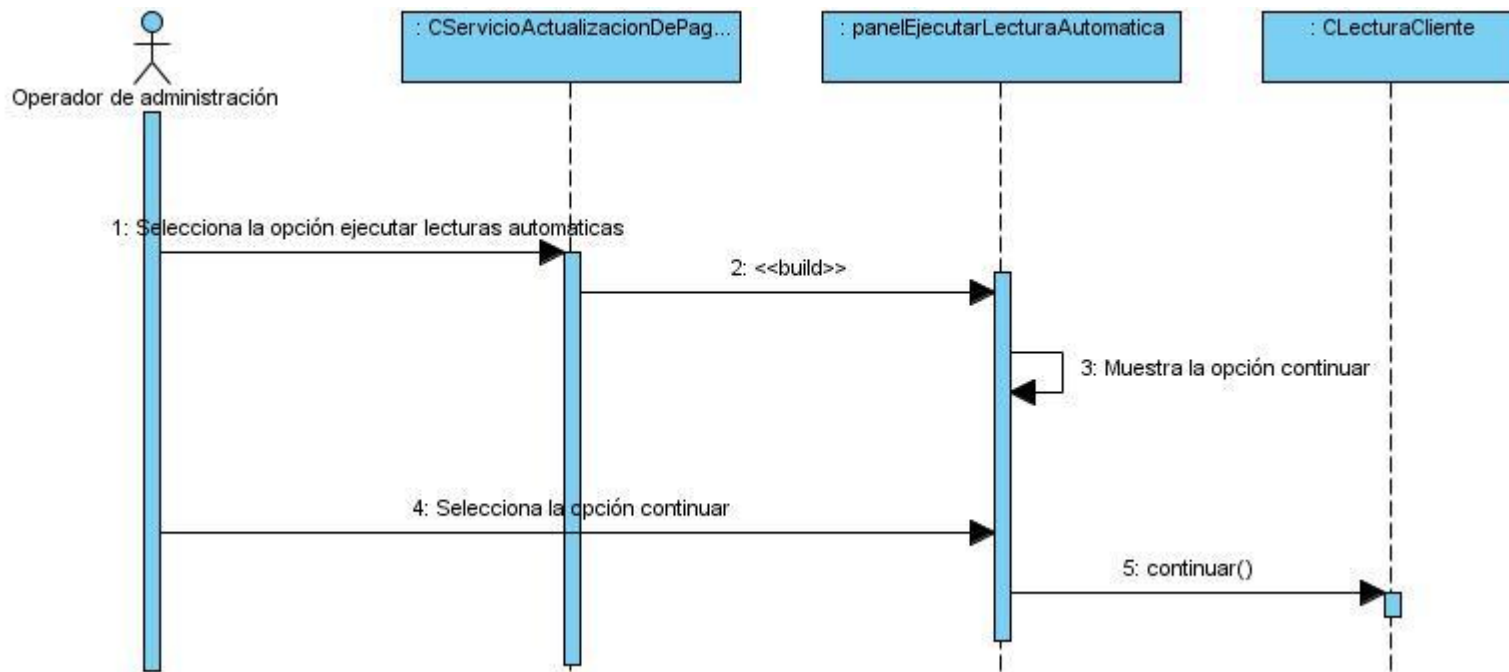


Figura. 35 DSD_Ejecutar lectura automática "Continuar."

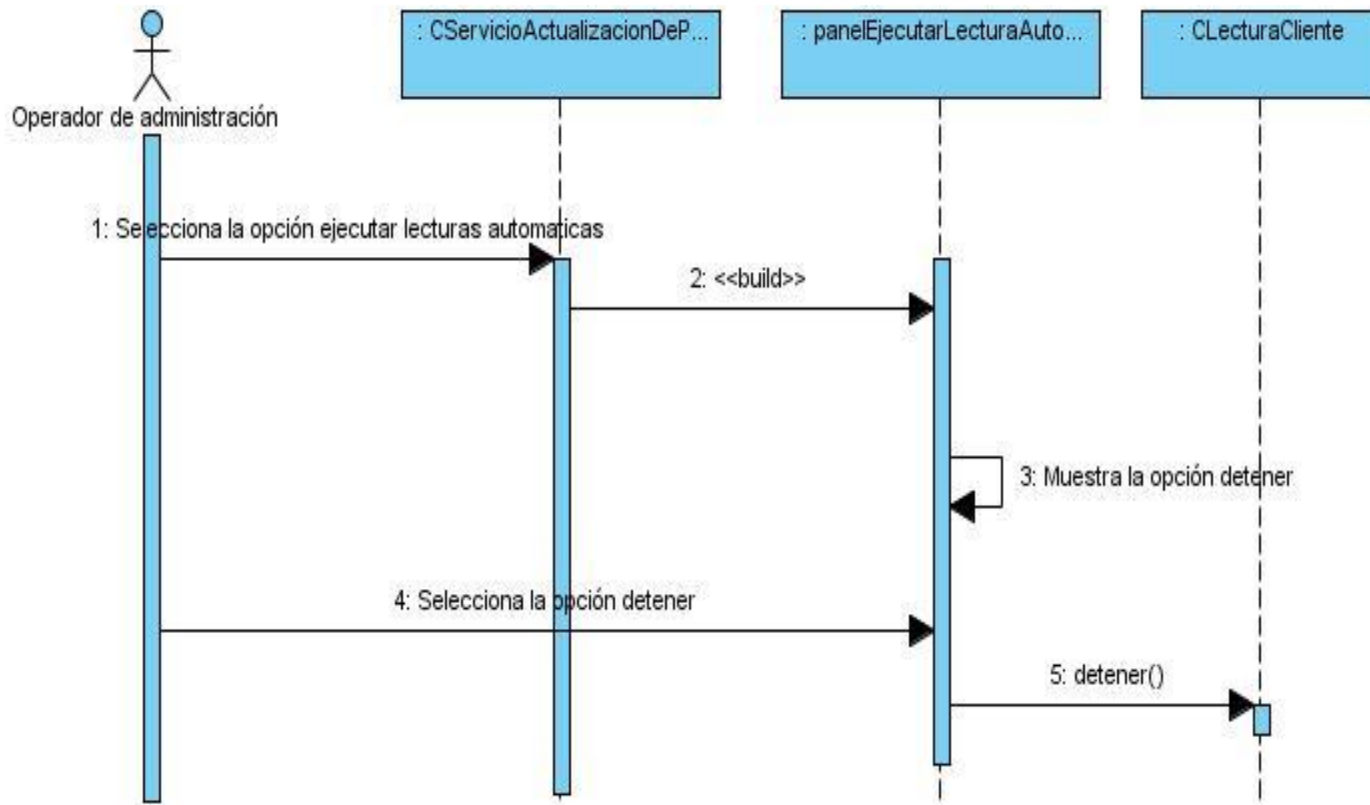


Figura. 36 DSD_Ejecutar lectura automática "Detener."

Anexos

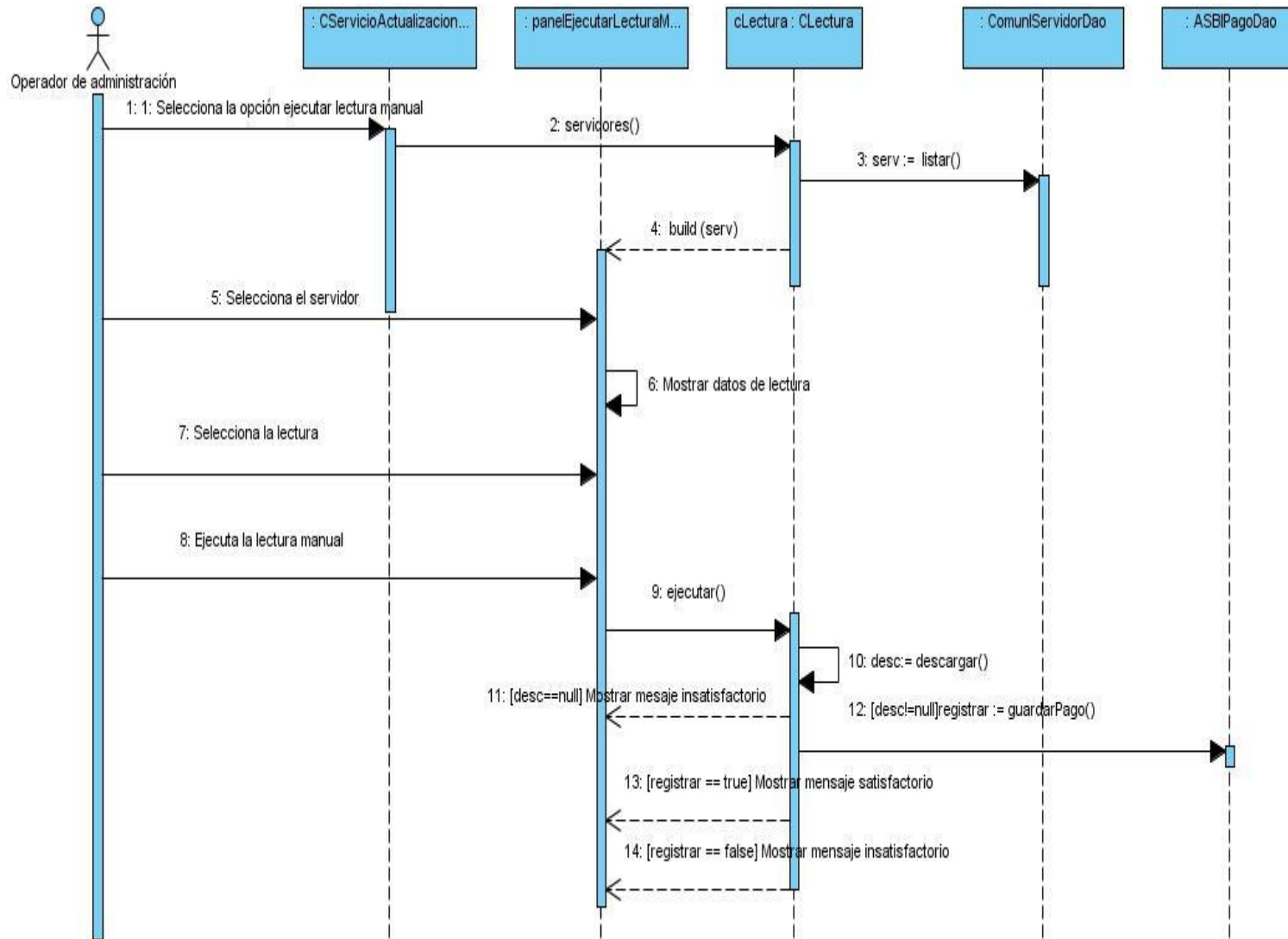


Figura. 37 DSD_Ejecutar lectura manual.

Anexos

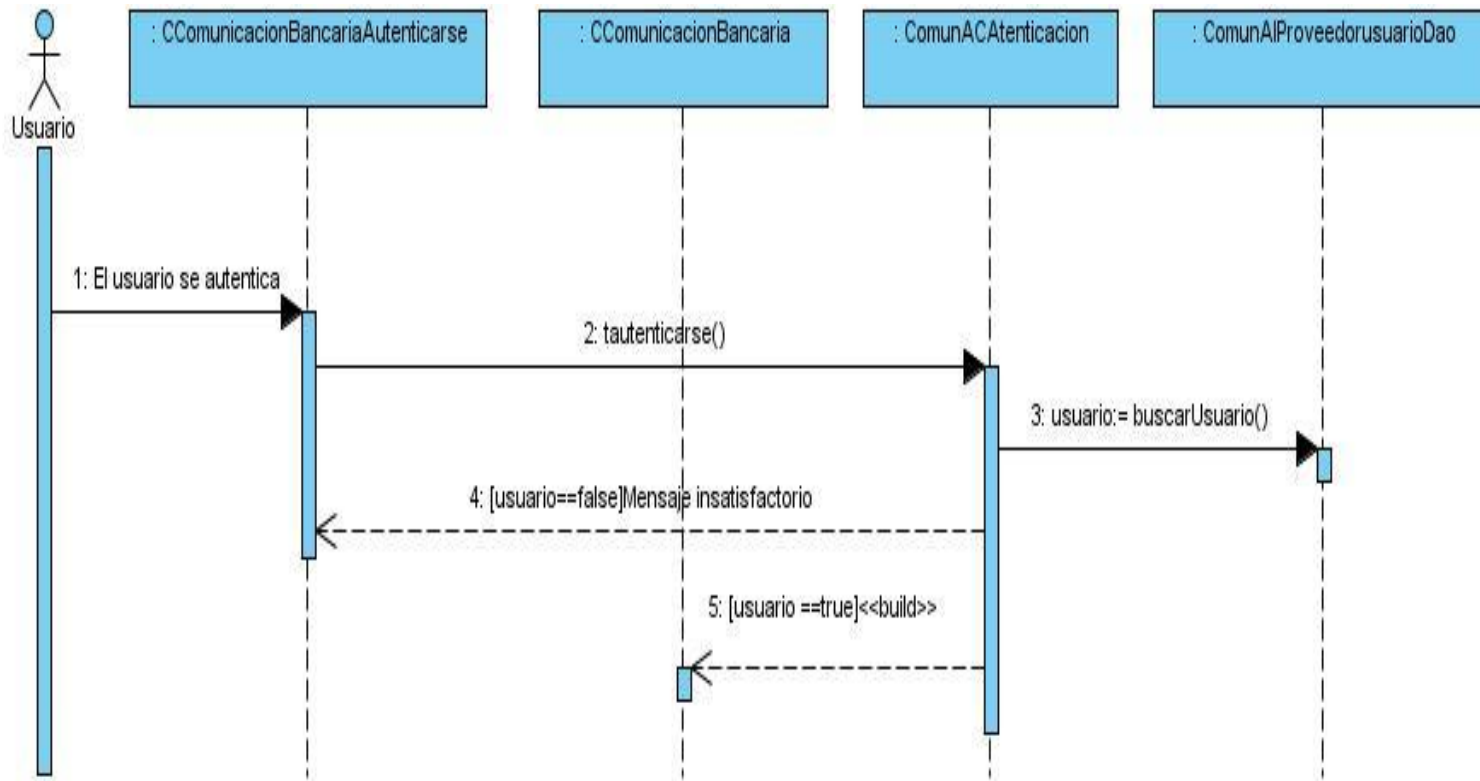


Figura. 38 DSD_Autenticar usuario Gestión de comunicación bancaria.

Anexos

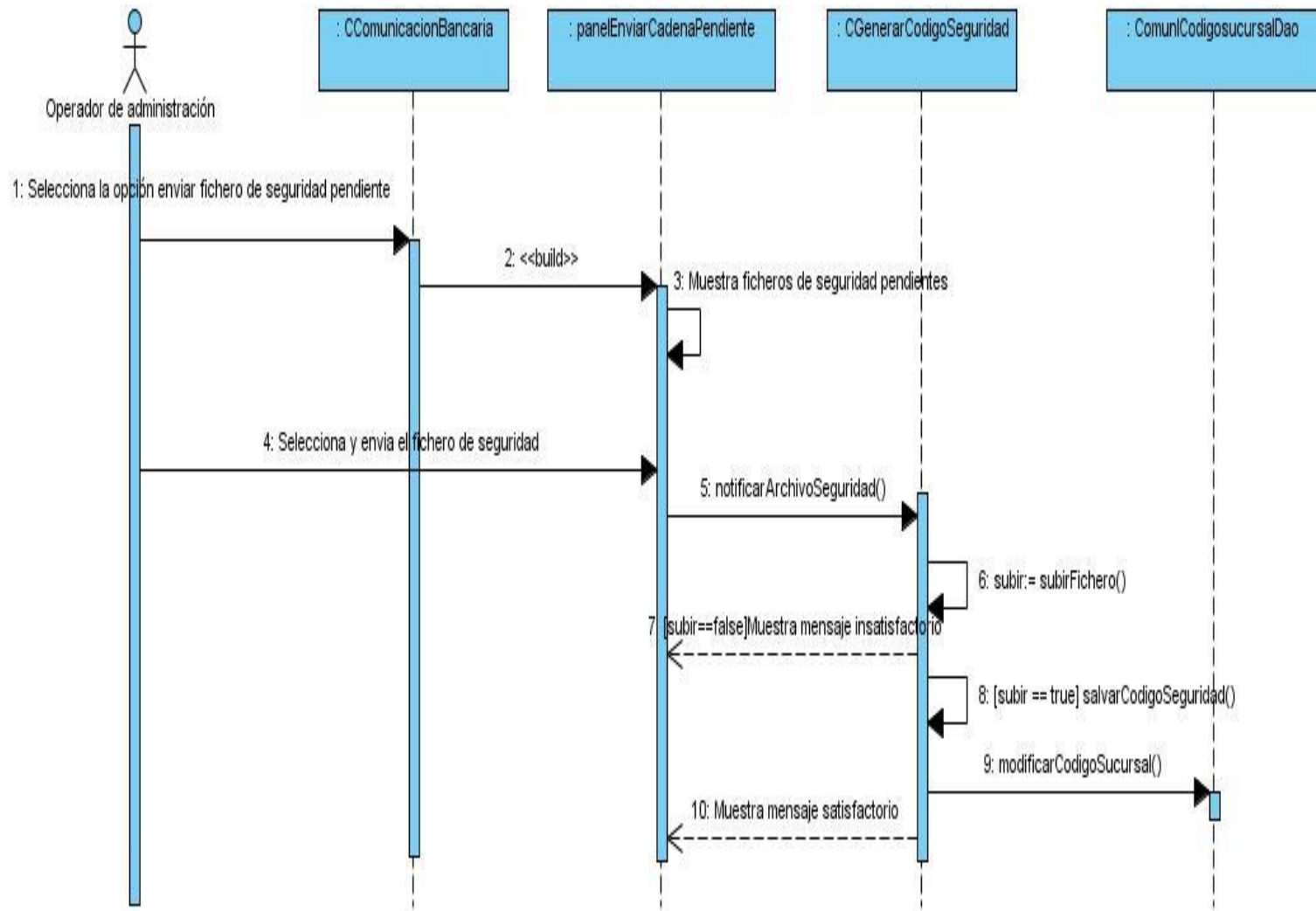


Figura. 39 DSD_Enviar archivo seguridad pendiente.

Anexos

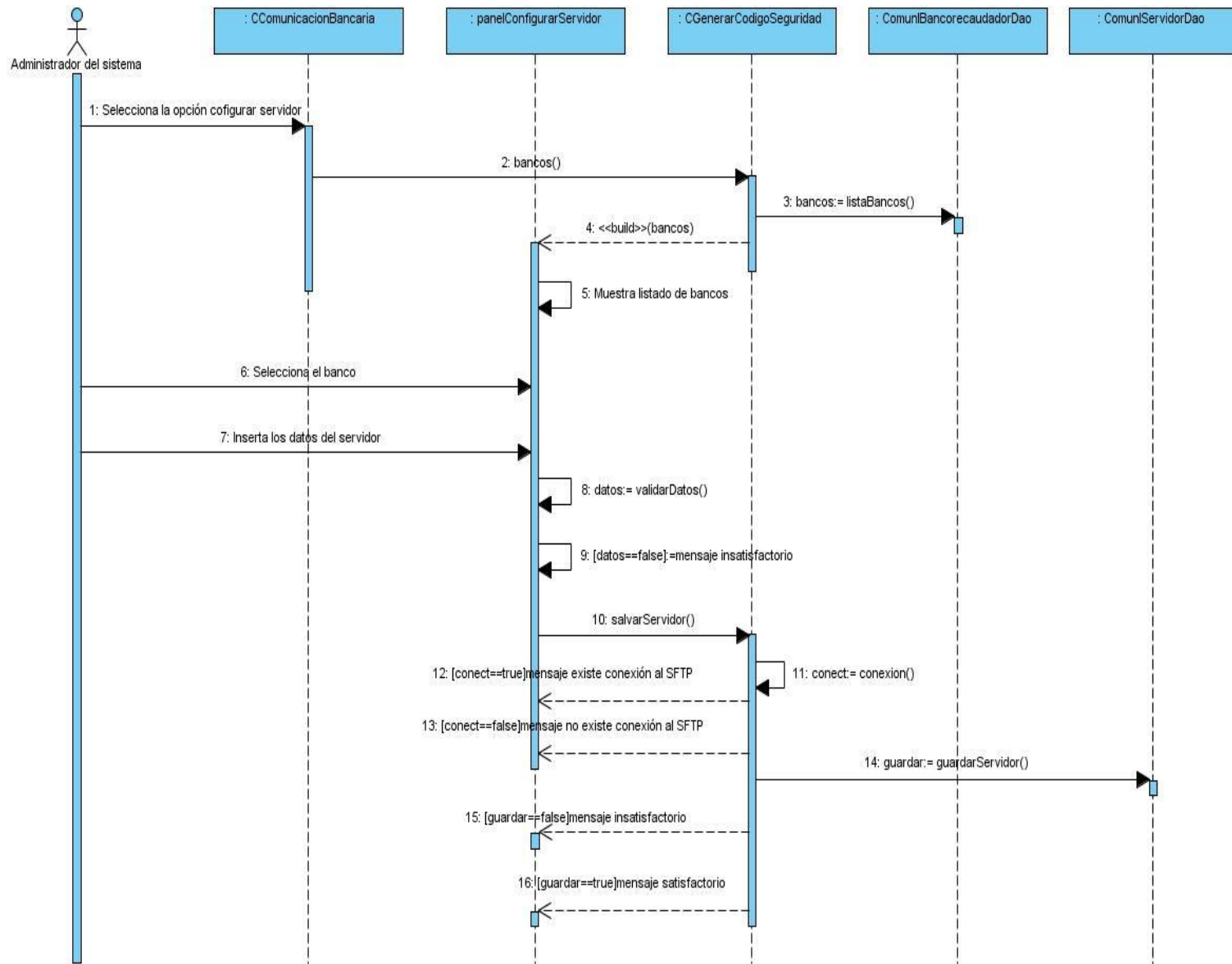


Figura. 40 DSD_Configurar servidor SFTP.

Anexos

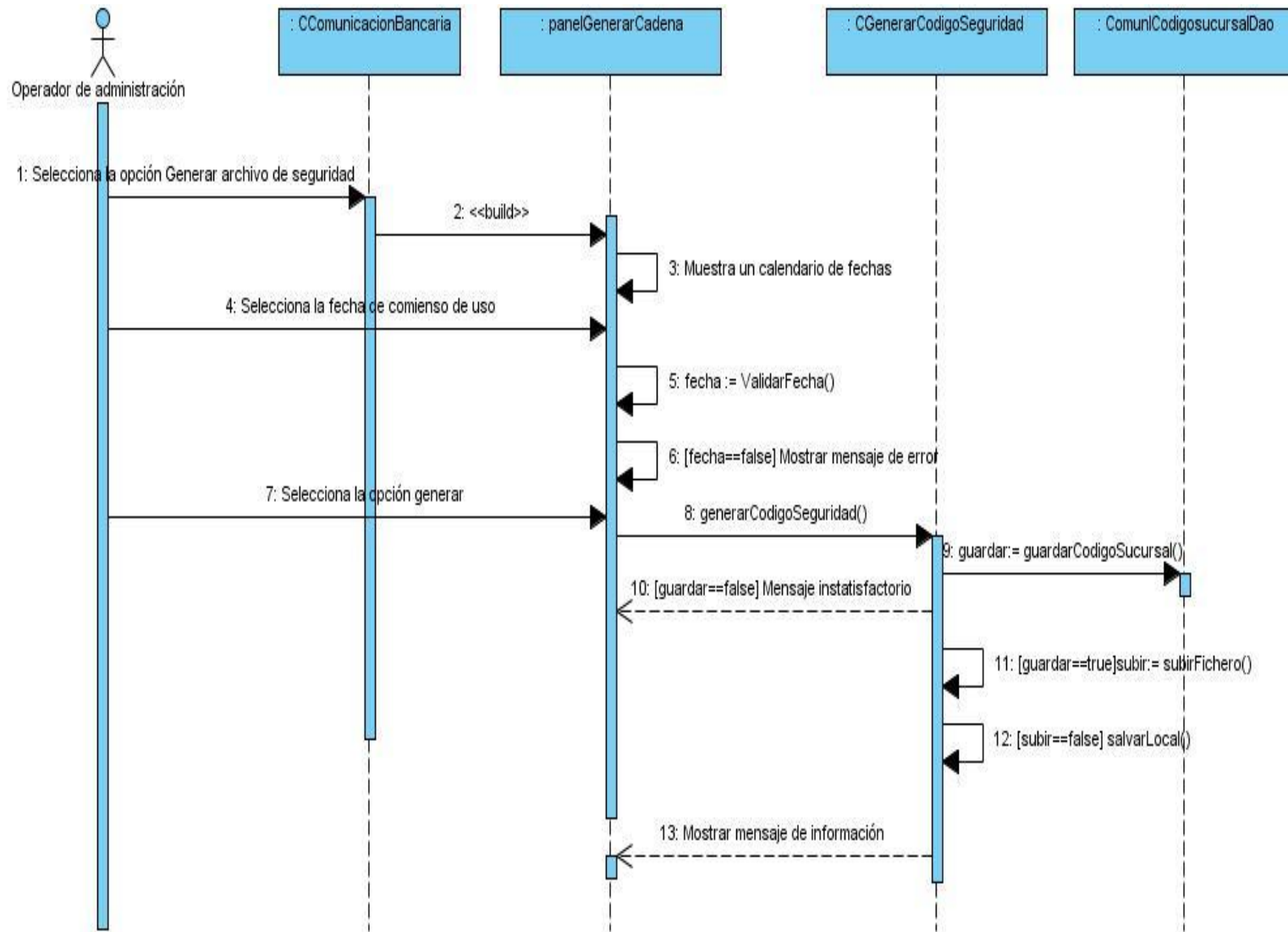


Figura. 41 DSD_Generar archivo seguridad.

Anexos

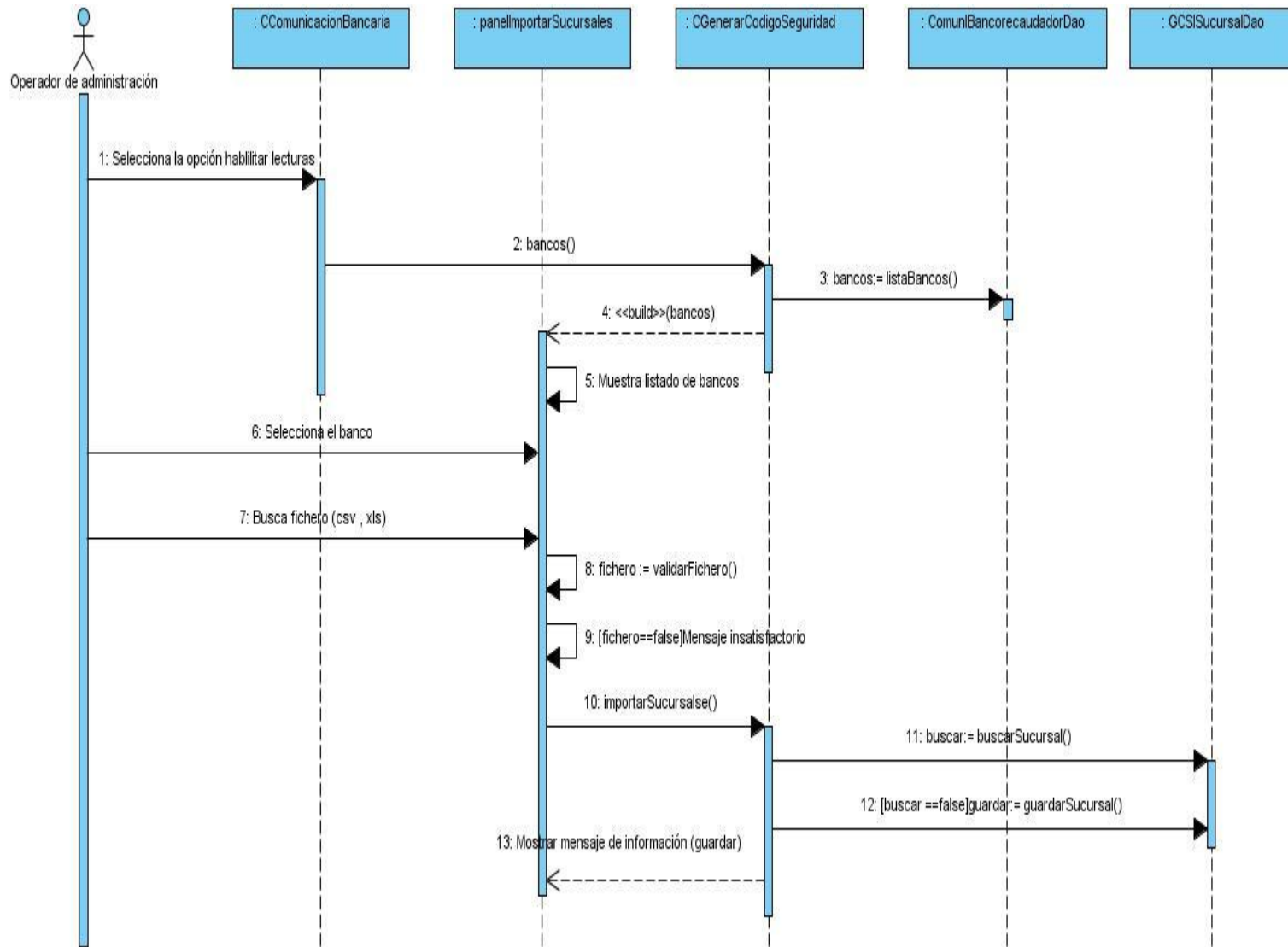


Figura. 42 DSD_Importar sucursales.

Anexo 6: Documento.

Documento: **ID-IB-CC-013 Especificaciones técnicas sobre la solución de Software de integración entre los Bancos recaudadores y el SAIME v1.4.**