Instituto tecnológico Nacional de México

División de Estudios de Postgrado e Investigación

TESIS

<u>TÍTULO:</u> Aplicación Web de *eBanking* para realizar consultas, transferencias y pagos de los usuarios de una entidad financiera no bancaria cliente de la empresa SINC

PRESENTADO POR:

LI. NADIA GUZMAN HINOJOSA M01010288

PARA OBTENER EL GRADO DE:

Maestra en Sistemas Computacionales

DIRECTOR DE LA TESIS:

MCE. BEATRIZ ALEJANDRA OLIVARES ZEPAHUA

Orizaba, Veracruz, México

Autorización de impresión

Revisión de trabajo escrito

Agradecimientos

- A mi madre Gisela por apoyarme siempre y en cada una de mis decisiones
- A Rogelio por su apoyo incondicional para realizar mi proyecto de maestría, por su amor y paciencia
- A mis hermanos José Manuel, Graciela y Fátima por su cariño y apoyo
- A René por siempre ser como un verdadero padre para mí y por su apoyo y cariño
- A mis compañeros Yessica Thalia, Elayne, Sergio René, Franz, Manuel, Jesús y César por su amistad, cario y apoyo en todo momento
- A mi maestra Beatriz por su guía, enseñanza, confianza y sobre todo por su paciencia
- A todos los maestros que me impartieron clases en el trayecto de la maestría
- Al Consejo Nacional de Ciencia y Tecnología (CONACYT), por el apoyo económico otorgado para la manutención durante el tiempo de la investigación.
- A la empresa Servicios de Informática Colegiada y en especial al Lic. Juan Manuel Castro por las facilidades otorgadas para la realización del caso de estudio y estancias en sus instalaciones, mismas que permitieron enriquecer el presente trabajo

Contenido

Índice de figuras

Índice de tablas

Resumen

En la actualidad dentro del sector financiero existen diversos problemas, sobre todo en el segmento de las entidades financieras no bancarias (EFNB), las cuales se dice que son auxiliares dentro del sector debido a que su objetivo principal es brindar atención e inclusión financiera a los grupos vulnerables o de escasos recursos de la sociedad. Debido al objetivo que estas empresas persiguen, están creciendo día a día en número de usuarios por lo que se hace necesario contar con sistemas de información efectivos, confiables y seguros que brinden atención a los usuarios tanto en las diferentes sucursales u oficinas como desde Internet, de tal manera que llegue a más personas el acceso a los servicios financieros y los usuarios tengan mejor control de sus operaciones.

Existen pocos proveedores de software que cubren las necesidades mencionadas, entre ellos se encuentra la empresa Servicios de Informática Colegiada (SINC), con sede en la ciudad de Monterrey, Nuevo León, que provee OpenFIN (Operación de Entidades Financieras), un sistema informático de gestión integral dirigido a diversas entidades financieras no bancarias (EFNB) en diversos estados del país, entre las cuales se encuentran SOFIPOs (Sociedades Financieras Populares), SCAPs (Sociedades Cooperativas de Ahorros y Préstamo), SOFOMs (Sociedades Financieras de Objeto Múltiple) y SOFOLES (Sociedades Financieras de Objeto Limitado) entre otras.

OpenFIN es una aplicación de escritorio que se instala directamente en el servidor de aplicaciones de una EFNB, aunque la empresa SINC crea enlaces remotos para brindar soporte al sistema. SINC no cuenta con una solución basada en Web mediante la cual los usuarios finales gestionen sus operaciones. Para solucionar esta problemática la alternativa es un módulo de banca en línea que forme parte del sistema OpenFIN y se libere a las entidades financieras no bancarias que lo quieran

implementar para que a su vez sus usuarios accedan vía Internet a sus cuentas y realicen operaciones mediante una computadora desde su casa o lugar de trabajo con los mecanismos de seguridad adecuados para respaldar dichas operaciones.

Abstract

Nowadays, within the financial sector, there are diverse problems, mostly in the segment of non-bank financial institutions, which are auxiliaries within the sector because their main objective is to provide care and financial inclusion to vulnerable or low-income groups in society. Due to the objective these companies pursue, they are growing every day in number of users; that is why it is necessary to have effective, reliable and secure information systems to provide attention to the users in both different branches or offices as well as from the Internet, so that more people have access to financial services and the users have a better control of their operations.

There are few software providers that cover the mentioned needs. Among them is the company of Servicios de Informática Colegiada (SINC), with headquarters in the city of Monterrey, Nuevo Leon, which provides OpenFIN (Operation of Financial Institutions), a computer system of integral management addressed to various non-bank financial institutions (EFNBs) in various states of the country, among which are SOFIPOs (Popular Financial Societies), SCAPs (Cooperative Societies of Savings and Loans), SOFOMs (Multi-Purpose Financial Companies) and SOFOLES (Limited Financial Companies) among others.

OpenFIN is a desktop application that is installed directly on the applications server of an EFNB, although the company SINC creates remote links to provide support to the system. SINC does not have a Web-based solution through which final users manage their operations. To solve this problem, the alternative is an online banking module that is part of the OpenFIN system and it sets free to the non-bank financial institutions that want to implement it, so that, in turn, its users can access to their accounts online and carry out operations through a computer from their home or workplace with the appropriate security mechanisms to support such operations

Introducción

Este trabajo presenta un panorama actual de las entidades financieras no bancarias en México, el objetivo que tienen dichas entidades dentro del sector financiero y las posibles limitantes que guardan para competir en el mercado financiero. Se aborda la solución que representa dentro del segmento de estas entidades, el hecho de contar con soluciones tecnológicas, principalmente las que son basadas en Web, para llegar a los diversos sectores y a los diversos usuarios de ellas dentro del mercado financiero y por lo tanto se reitera la conveniencia del desarrollo de una aplicación Web para efectuar transacciones básicas que faciliten la realización de las mismas, en este caso, se trata el caso de una aplicación de *eBanking* para llevar a cabo operaciones de consultas, transferencias y pagos en una entidad financiera no bancaria cliente de la empresa SINC. Así mismo se detalla el proceso de construcción de dicha solución.

El documento está formado por cinco capítulos: el capítulo uno tiene la finalidad de dar a conocer el marco teórico concerniente a la regulación vigente que rige a las entidades financieras y a la cual se sujeta la presente tesis, los conceptos relevantes del sector de las entidades financieras no bancarias y SINC, el proveedor del sistema OpenFIN, conceptos del área tecnológica que son necesarios conocer para introducir al desarrollo de la propuesta de solución; por otro lado se incluye también el objetivo general y los objetivos específicos de la tesis. En el capítulo dos se hace una revisión del estado del arte respecto al tema abordado, se analizan diferentes artículos de investigación en el área del sector financiero, el sector de comercio electrónico, las amenazas más comunes al navegar en Internet y sobre todo al realizar transacciones que implican el manejo de dinero y finalmente los mecanismos de seguridad informática. En el tercer capítulo se presenta la alternativa tecnológica elegida para el desarrollo de la aplicación de *eBanking* basándose en una arquitectura orientada a servicios, previendo que la aplicación sea responsiva, contemplando los mecanismos para asegurar la interoperabilidad con el sistema existente, los modelos para

implementar software como servicio para desarrollar el caso de estudio de la tesis. El capítulo cuatro presenta los resultados del caso de estudio y finalmente el capítulo cinco contiene las conclusiones y recomendaciones.

Capítulo 1. Antecedentes

Se presentan los antecedentes para abordar el proyecto de tesis, dentro del marco teórico se da a conocer diferente información con base en fuentes científicas.

1.1 Marco teórico

Presenta conceptos importantes sobre los cuales se fundamenta la orientación de la tesis.

1.1.1. Sistema financiero

La economía se divide en tres sectores: el primario o agropecuario, el secundario o de la transformación, terciario o de servicios, que comprende el sistema financiero. El sistema financiero es la base de la economía de un país. En México el sistema financiero abarca un conjunto de organismos e instituciones que autoriza el estado para captar, administrar, canalizar y regular los recursos económicos públicos o privados, nacionales e internacionales. Tiene leyes y reglamentos mediante los que busca gestionar los recursos monetarios de forma profesional [1].

1.1.2. Cooperativas de ahorro y préstamo y Financieras no bancarias.

Las cooperativas de ahorro y préstamo y las Financieras son organismos auxiliares del crédito dentro del sistema financiero. La Ley de Ahorro y Crédito Popular establece los principios normativos para las instituciones de la banca social. Las instituciones de banca social se dividen en dos tipos: Sociedades Cooperativas de Ahorro y Préstamo (SCAP) sin fines de lucro y Sociedades Financieras Populares (SOFIPO) con fines de lucro. Las SCAP se caracterizan por dedicarse al sector popular y a las microfinanzas. Las SCAP se supervisan por la

Comisión Nacional Bancaria y de Valores (CNBV). A su vez, todo el sistema financiero está bajo la tutela de la Secretaria de Hacienda y Crédito Público (SHCP). El objetivo de las Entidades Financieras no Bancarias es ofrecer servicios financieros a gran escala, dando inclusión financiera a los sectores menos favorecidos de la sociedad para contribuir con el crecimiento económico del país mediante el otorgamiento de créditos, principalmente los destinados al desarrollo de negocios los cuales son otorgados tanto a pequeñas y medianas empresas como a personas físicas [1] .

Hay diversas maneras en que la mayoría de la población de escasos recursos realiza sus ahorros para crear un patrimonio; como ejemplo las llamadas “tandas”, las “pirámides” o los patronatos de los lugares de trabajo, son la opción más cercana al no tener la posibilidad de acceso al sistema bancario. En este medio surgen las SCAP como otra manera de acceder a créditos, donde los plazos son cortos, los intereses moderados y los usos de los mismos son muy variados [2].

1.1.3. Organismos reguladores.

La Comisión Nacional Bancaria y de Valores es un organismo que representa a una autoridad encargada de supervisar y regular a las entidades que integran el sistema financiero mexicano, a fin de lograr una estabilidad y un correcto funcionamiento del mismo, manteniendo y fomentando el sano y equilibrado desarrollo de dicho sistema en su conjunto. Además se encarga de proteger los intereses de la población para tener más y mejores servicios financiero que coadyuven al crecimiento del país [3].

La capacidad de los bancos de elevar los márgenes de capital depende en gran parte de la adaptación a la tecnología. Un modelo de banca tradicional se ve amenazado si no se adapta a los avances tecnológicos. La competencia entre las empresas del sector financiero está siempre latente. Existen otros competidores

independientes que entran al mercado usando los nuevos avances tecnológicos, en el sector del comercio electrónico e intermediarios que ofrecen sus propias formas de dinero electrónico. Las instituciones que no apuestan por la tecnología, siguen asumiendo costos mayores al brindar sus servicios [4].

1.1.4. EBanking

Es un término bastante amplio que agrupa a todas las maneras de interactuar con un banco o entidad financiera de manera electrónica y en línea. Se asocia el concepto *eBanking* con los sitios Web de un banco, ya que así fue cómo comenzó a hacerse conocido el término, pero actualmente los clientes se relacionan con un banco a través de muchos más dispositivos como teléfonos celulares, televisores, autoservicios y aplicaciones de escritorio entre otras, por lo que la relación con un banco se ha extendido fuera del sitio Web de éste y agrega la opción de pagar sus cuentas de servicios en línea.[5].

1.1.5. Seguridad y privacidad de los datos.

La privacidad de la información se entiende como la decisión de una persona o un grupo de personas respecto a cuándo, cómo y en qué forma comunicar cierta información a los demás, manteniendo un control sobre su información y el uso que se le da a ésta. El término “privacidad de los datos en Internet” se refiere a que un negocio en la Web debe responder a la confianza que un individuo o un grupo de individuos ponen en él, al compartirle cierta información personal o confidencial. La seguridad de la información y datos consiste de tres partes: la integridad, la confidencialidad y la disponibilidad. Las tres partes se afectan por fallas técnicas, fallas naturales o por errores humanos, ya sean accidentales o deliberados. Actualmente la tecnología permite el diseño seguro de las aplicaciones y es tarea del desarrollador y del usuario considerarlas al hacer de uso de transacciones en línea [7].

1.1.6. Interoperabilidad

El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) define interoperabilidad como la habilidad de dos o más sistemas o componentes para intercambiar información y luego utilizar esa información. Otra definición más amplia la proporciona la Comisión Europea, como "la habilidad que tienen las organizaciones con sistemas diferentes para intercambiar información y conocimiento mediante sus sistemas informáticos, de forma consensuada, con el objetivo de obtener beneficios". La interoperabilidad pasa por el desarrollo de estándares de tecnología abiertos que hagan posible que productos y plataformas heterogéneas se comuniquen entre sí. Microsoft comparte información sobre su tecnología con cientos de miles de entidades y empresas colaboradoras, y con organismos independientes, para desarrollar estándares abiertos. La "*Interoperability Initiative*", es una iniciativa que consta de cuatro áreas principales: asegurar las conexiones abiertas; promover la portabilidad de los datos; fomentar la interacción con la industria, incluida la comunidad de software de código abierto; y mejorar el soporte a los estándares del sector [8].

1.1.7. JavaServer Faces

JavaServer Faces es un marco de trabajo del lado del servidor para construir aplicaciones Web con tecnología basada en Java. Consiste en una API (*Application Programming Interface*, Interfaz de Programación de Aplicaciones) para representar componentes y administrar su estado, manejo de eventos, validación del lado del servidor y conversión de datos, definición de la navegación de páginas Web, soportar internacionalización y accesibilidad, y proporcionar extensibilidad para estas características. Proporciona bibliotecas de etiquetas para agregar componentes de interfaz de usuario a las páginas Web y para conectar componentes del lado del servidor, grabar y restaurar el estado de los componentes de interfaz de usuario[9].

JavaServer Faces es parte de la plataforma Java EE y proporciona un modelo de programación bien definido y varias bibliotecas de etiquetas. Las bibliotecas contienen controladores de etiquetas que implementan componentes de etiquetas. Con un esfuerzo mínimo se realizan las siguientes tareas:

- Crear una página Web dinámica.
- Mapear componentes UI a una página de datos del servidor
- Restaurar y guardar el estado de la aplicación más allá de la vida de las peticiones al servidor.
- Reutilizar y extender componentes a través de la personalización.

Una aplicación con JavaServer Faces se compone de un conjunto de páginas Web en las que se disponen los componentes, etiquetas para agregar componentes, un conjunto de beans administrados (*managed beans*), un descriptor de despliegue (web.xml), uno o más archivos de configuración (faces-config.xml), opcionalmente un conjunto de objetos personalizados, que permite incluir componentes personalizados, validadores, convertidores, entre otros, creado por el desarrollador de aplicaciones, opcionalmente un conjunto de etiquetas personalizadas para representar objetos personalizados en la página.

1.1.8. PrimeFaces

PrimeFaces es un marco de trabajo para aplicaciones enriquecidas que funciona directamente con JSF y permite, además de una mejor experiencia de usuario, validaciones del lado del cliente [10].

1.1.9. Servicios Web

Un servicio Web es una tecnología que consiste de hardware y software y sirve para comunicar diferentes aplicaciones que a su vez se desarrollaron en lenguajes de programación diferentes, esta tecnología utiliza ciertos estándares y protocolos que permiten la interoperabilidad. La WS-I (*Web Services Interoperability Organization*) es una que establece mejores prácticas para la interoperabilidad de los servicios Web, para grupos seleccionados de estándares de servicios Web, a

través de plataformas, sistemas operativos y lenguajes de programación. WS-I comprende una comunidad diversa de líderes de servicios Web a partir de una amplia gama de empresas y organizaciones de normalización. Se anima a las empresas interesadas en ayudar a establecer las mejores prácticas para los servicios Web para unirse a WS-I [11] [12].

1.1.10. SOAP

SOAP (*Simple Object Access Protocol*, protocolo de acceso simple a objetos) es un formato de mensaje XML utilizado en interacciones de servicios Web. Los mensajes SOAP habitualmente se envían sobre HTTP ya que es compatible con todos los servidores de Internet. SOAP proporciona una forma de comunicarse entre aplicaciones que se ejecutan en diferentes sistemas operativos, con diferentes tecnologías y lenguajes de programación. Un mensaje SOAP viaja de un emisor a un receptor pasando diferentes puntos a lo largo de la ruta del mensaje [13].

1.1.11. WS-Security

Seguridad de Servicios Web (WS-Security) describe mejoras para la mensajería SOAP para proporcionar calidad de protección a través de la integridad de mensajes, confidencialidad de mensajes y una sola autenticación de mensajes. Los mecanismos de WS-Security se utilizan para dar cabida a una amplia variedad de tecnologías de cifrado y modelos de seguridad.

WS-Security es un estándar a nivel de mensajes que tiene su base en la protección de mensajes SOAP a través de la firma digital XML, la confidencialidad a través del cifrado XML y la propagación de credenciales a través de señales de seguridad [14].

1.1.12. Mecanismos de seguridad al utilizar servicios Web

Por su naturaleza de acceso abierto y bajo acoplamiento los servicios Web contemplan un conjunto de requerimientos en el aspecto de la seguridad: A) Autenticación, verificando que el usuario es quien dice ser, el usuario se identifica mediante una credencial, por ejemplo 1) una credencial emitida por una autoridad, 2) una contraseña secreta compartida, 3) algo que uno es, por ejemplo, información biométrica; B) Autorización o control de acceso, otorgar acceso a ciertos recursos, con base en los derechos del usuario autenticado, éstos derechos están definidos mediante ciertos atributos del usuario; C) Confidencialidad o privacidad, mantener la información en secreto, por ejemplo, un correo electrónico tanto como las partes del emisor y el receptor de forma confidencial cifrando el contenido del mensaje; D) Integridad, asegurar que el mensaje no sea alterado durante su envío, mediante la firma del remitente [15].

1.1.13. Arquitectura Orientada a Servicios

La Arquitectura Orientada a Servicios (*Service Oriented Architecture*) es un paradigma para diseñar y desarrollar sistemas distribuidos, se construye a partir de un conjunto de servicios que se comunican entre sí. La comunicación implica desde el paso sencillo de datos o hasta dos o más elementos coordinando alguna actividad. Se requiere un medio de conexión de servicios y esta conexión son los Servicios Web [11].

1.1.14. Computación en la nube

Computación en la nube o "*Cloud computing*", es una forma especializada de computación distribuida que introduce modelos de utilización para el aprovisionamiento de recursos escalables de forma remota y medida. De acuerdo al Instituto Nacional de Estándares y Tecnología norteamericano (NSIT), las directrices para la computación en nube tienen cuatro formas de implementación: privada, para la comunidad, públicos e híbridos, así como tres diferentes modelos

de entrega: el SaaS (Software como Servicio), PaaS (plataforma como servicio) e IaaS (Infraestructura como Servicio). El Software como Servicio (SaaS) es un modelo de entrega de servicios en el que se alojan las aplicaciones y su gestión se hace en el centro de datos del proveedor de servicios, el pago es por suscripción y se accede a través de un visualizador a través de una conexión a Internet. Básicamente se trata de la concesión de licencias de una aplicación a los clientes para su uso como un servicio bajo demanda. [16].

1.1.15. OpenFIN

OpenFIN es la solución informática para las EACP (Empresas de Ahorro y Crédito Popular), resultado de un amplio estudio de las necesidades reales de información tanto internas como externas, cumpliendo con las exigencias que impone la ley. Contiene un módulo de administración para los administradores del sistema, un apartado de manejo de catálogos contables y generación de reportes contables; un módulo de gestión de clientes que es donde se registran todas las operaciones diarias en la EACP; un módulo de gestión de cobranza. La aplicación es de escritorio y se instala sobre plataforma Linux, Ubuntu 12.04. Actualmente OpenFIN se distribuye bajo licencia de la empresa SINC (Servicios de Informática Colegiada S.A. de C.V); quien a su vez brinda soporte remoto a sus clientes desde las instalaciones centrales de SINC.

1.1.16. Situación tecnológica, económica y operativa de la empresa

La empresa SINC Servicios de Informática Colegiada S.A. de C.V brinda el software informático de gestión integral para entidades financieras no bancarias, cuenta con la siguiente infraestructura tecnológica: un una red VPN (*Virtual Private Network*, Red Privada Virtual) hacia algunas de las entidades que atiende, mediante la cual brinda soporte remoto a su sistema informático OpenFIN, el cual se instala sobre un servidor Linux distribución Ubuntu 15.04, con una base de datos en PostgreSQL 9.3.

1.1.17. Planteamiento del problema

Actualmente el problema en el sector de las entidades financieras no bancarias es que hacen falta mecanismos tecnológicos que fortalezcan la competitividad de éstas frente a las instituciones de la banca tradicional y otros intermediarios no bancarios, debido a que no existen reportes de que haya aplicaciones de banca en línea para este tipo de entidades y son pocos los proveedores de software que cumplen con los requisitos que impone la ley. Por otro lado, tanto las instituciones de banca no tradicional como sus clientes, incurren en grandes gastos al tener que dar atención a los clientes ya sea en campo o en sucursales físicas, a su vez los clientes gastan dinero al trasladarse o dejan de acudir y caen en atraso de sus créditos. Es posible mejorar tal situación al tener un acceso en línea, disponible desde su propia computadora para mantener un mejor control de las operaciones en sus cuentas.

1.1.18. Objetivo general y específicos

A continuación, se describen tanto el objetivo general como los objetivos específicos del proyecto de tesis.

1.1.19. Objetivo general

Desarrollar una aplicación Web de eBanking para facilitar la gestión de sus pagos y manejos de cuenta y seguimiento de saldos de ahorro de los socios en la empresa, utilizando la seguridad de compartición de mensajes que implementa WSI (*Web Services Interoperability,* Interoperabilidad de los Servicios Web).

1.1.20. Objetivos específicos

- Analizar la estructura existente de la empresa

- Analizar las tecnologías de desarrollo Web existentes más adecuadas para una implementación con la seguridad informática debida
- Analizar los estándares y las tecnologías de seguridad informática
- Diseñar una arquitectura para la interoperabilidad con el sistema existente
- Desarrollar la aplicación Web responsiva de banca en línea para la gestión de las cuentas de los usuarios.

1.1.21. Justificación

Las entidades financieras no bancarias tienen presencia en más de 150 países alrededor del mundo y tienen como prioridad la atención y la inclusión de los servicios financieros a los sectores relegados de la población. La banca accesible a través de las tecnologías Web actualmente aún no está muy difundida. En México, un estudio de BBVA Research revela que, del total de las transacciones bancarias que se realizan, sólo 16% son a través de Internet y apenas 6.4% por telefonía móvil. La banca digital en México apenas empieza. No hay un banco que tenga todavía una buena propuesta de banca digital, dice Hugo Nájera, director de Banca Digital de BBVA Bancomer [17].La exigua presencia de los intermediarios financieros formales en los municipios rurales mexicanos impulsa el uso de medios informales de financiamiento. Las cajas de ahorro constituyen una alternativa importante y aprovechan el capital social comunitario con el fin de brindar servicios accesibles y al alcance de todos sus usuarios.

La creación de un sistema de eBanking, que brinde la posibilidad de gestionar las cuentas de ahorro de los socios, así como sus préstamos contraídos con la EFNB desde una computadora, proveerá más control sobre los saldos al permitir realizar pagos desde Internet, consultar las operaciones realizadas, favoreciendo el pago oportuno y ayudando a evitar el atraso de los pagos sin necesidad de acudir directamente a alguna sucursal. La banca es un sector históricamente ligado a las Tecnologías de la Información y de la Comunicaciones, se apuesta por la tecnología y la innovación para así lograr rentabilizar sus inversiones e

incrementar su presencia en el mercado y aumentar en número de clientes. Para contar con una plataforma segura de banca en línea o banca electrónica, es necesario contemplar la interoperabilidad con el sistema principal y aspectos de seguridad en la transferencia de datos a través de Internet. Contar con herramientas de seguridad y defender el sistema de forma rápida contra ataques maliciosos, contar con la autenticación, autorización y gestión de sesiones, así como envíos seguros desde el visualizador hasta la base de datos, seguridad de archivos, detección de vulnerabilidades y desarrollo seguro.

Capítulo 2 Estado de la práctica

En este capítulo se presenta una recopilación y análisis de trabajos relacionados con el proyecto de tesis, lo que permite conocer hasta dónde se ha investigado en el campo de las tecnologías que existen al utilizar la banca en línea y en cuanto a las amenazas a la seguridad en la Web.

2.1 Trabajos relacionados

En [18] se estableció que el problema de la detección de fraudes es bien conocido en entornos bancarios y se magnifica por el crecimiento de las transacciones económicas, empujando a las instituciones bancarias hacia la regulación del uso de la banca electrónica. A medida que más gente está expuesta, la privacidad de la información y la seguridad es crucial para las transacciones electrónicas. El “Acuerdo de Capital de Basilea”, también conocido como "Basilea 2" se refiere a la definición de los elementos que componen el riesgo de una transacción y define un estándar internacional que establece los requerimientos de capital para proteger a una entidad frente a los riesgos. Un primer paso es la construcción de un grupo de pruebas de temas y situaciones de fraude, bajo el Modelo de Markov, en el que se aprende de las operaciones del "mundo real", analizando procesos por lotes en tiempo real, las operaciones de los bancos. De esta manera se calcula un valor monetario de los riesgos asociados a los fraudes.

En [19] se presentó que la arquitectura básica del sistema de banca en línea consta de tres componentes principales, el cliente, la aplicación y el servidor que almacena los datos del cliente y el banco. Hay dos cuestiones de tecnología a vigilar: 1) Seguridad: y 2) Autenticación. Existen muchos algoritmos disponibles basados en software para implementar el cifrado de claves, por ejemplo Rivest-Shamir-Adleman (RSA), *Advanced Encryption Standard* (AES) y *Data Encryption Standard* (DES). El departamento de Información Tecnológica de la Universidad

de Alejandría optó por combinar el cifrado con IB-MRSA (*Simple Identity-Based Cryptography with Mediated RSA*, Cifrado simple basado en identidad mediada con RSA) el cual tarda aproximadamente 4-5 veces menos que el RSA. Divide las claves privadas entre el cliente y el servidor, cada clave es usada una sola vez y cada firma involucra a ambas partes. Hay tres tipos de ataques de denegación de servicio: contra el servidor, la memoria, y el CPU. Para manejar el problema de ataque de CPU, la idea de emplear una sola vez el identificador se sugiere con el fin de hacer que los atacantes no reutilicen las solicitudes generadas por el usuario legal.

En [20] se presentaron los ataques en línea que tratan de capturar credenciales, información de intercepción y desviar fondos, sin la posibilidad de que sus propietarios obtengan el reembolso de las instituciones bancarias; así mismo se describe que las instituciones bancarias son incapaces de tomar las medidas legales contra sus atacantes. La manera en que un cliente realiza el proceso de autenticación a su banco o cooperativa de crédito debe, sin excepción, asegurar que la confirmación y verificación de su identidad se lleva a cabo de manera segura a través de Internet. Ambas partes de una transacción no negarán más tarde que la transacción ocurrió de verdad. Los riesgos y métodos de ataque cómo los ataques de "*malware*" (*Malicious software*, software malicioso) buscan patrones de números de cuentas bancarias, patrones de números de tarjetas de crédito, información de la cuenta y luego envían esta información al atacante. Para solucionar esto se recomienda a los usuarios a elegir bien el sistema operativo, el software antivirus, un *firewall*, así como algunos otros programas anti "*malware*", restringir el uso de los visualizadores Web y contratar servicios de alerta de los bancos.

En [21] se presentó que una transacción de comercio electrónico típicamente se protege usando SSL (*Secure Socket Layer*, Capa de Conexión Segura) y TSL (*Transport Layer Security*, Seguridad de la Capa de Transporte). Sin embargo,

persisten algunos riesgos de tal utilización: la información que se almacena en la computadora del cliente y la autenticación de usuario. Aunque SSL / TLS sí ofrece este último, el servicio de seguridad es opcional y por lo general se omite y no obliga a la autenticación del cliente. Se propuso un protocolo de pago en el que el riesgo de tener datos de tarjetas de débito o crédito almacenados en un servidor se elimina. Esto se logra mediante la utilización de la confidencialidad de los datos del servicio GSM para cifrar la información sensible. La seguridad de GSM (*Global System for Mobil Communications*, Sistema global para comunicaciones móviles) proporciona la autenticación de la identidad del usuario. Si la autenticación de cliente la proporciona correctamente SSL / TLS, a continuación, el usuario establece un par de claves, una de ellas pública. También se necesita lugar para almacenar la parte privada de la clave. Dado que un gran número de usuarios de todo el mundo poseen un teléfono móvil GSM, se propone un protocolo de pago en el que un teléfono móvil GSM se utiliza para proporcionar el titular de la tarjeta.

En [22] se presentó que los ataques por inyección SQL tienen por objeto buscar las vulnerabilidades de una aplicación Web, para insertar código SQL malicioso. El etiquetado SQLPIL (*SQL Injection Prevention by Input Labeling*, prevención de inyección SQL mediante etiquetado de entrada) una solución para plataformas Java contra este tipo de amenaza, ya que los datos que llegan se tratan siempre como datos y no como instrucciones SQL. Después de comprobar la autenticidad de la solicitud de un cliente, se transforma dinámicamente la cadena de datos en instrucciones seguras antes de su ejecución.

En [23] se dijo que debido a la naturaleza de acceso abierto que tiene la Web, los servicios Web son vulnerables al XSS (*Cross Site Scripting*, Secuencias de órdenes en sitios cruzados) que funciona inyectando código JavaScript o lenguaje similar en las páginas que el utiliza el usuario. WS-Security (Seguridad en servicios Web) es un protocolo que proporciona mecanismos de seguridad a los servicios Web; *Security Tokens* es una especificación de seguridad para autenticar

la identidad del usuario y autenticar la autorización de un Servicio Web. Ambas herramientas disminuyen significativamente el ataque por XSS en los servicios Web.

En [24] se estableció que la aparición de la computación en la nube alteró drásticamente la percepción acerca de la infraestructura de la arquitectura, entrega de software y modelos de desarrollo. Abarca elementos de computación en malla (*grid computing*), suministro de recursos computacionales (*utility computing*) y computación autónoma, todo esto en una arquitectura innovadora. Surgen entonces los riesgos y la deficiencia en la seguridad de los mecanismos tradicionales. Los autores proponen una infraestructura de clave pública que opera en conjunto con SSO (*Single Sign On*, Sistema de Autenticación Reducida) y LDAP (*Lightweight Directory Access Protocol*, Protocolo Ligero de Acceso a Directorios) para garantizar la autenticación, la integridad y la confidencialidad de los datos involucrados y las comunicaciones. Una combinación de PKI, LDAP y SSO hace frente a la mayoría de las amenazas identificadas en la computación en la nube.

En [11] se definió SOA (*Service Oriented Architecture*, Arquitectura Orientada a Servicios) como una forma de diseñar, implementar y estructurar servicios para soportar o automatizar funciones de negocios. SOA se construye a través de un conjunto de servicios. Los WS (*Web Services*, Servicios Web) son una tecnología que utiliza un conjunto de protocolos y estándares para conectar componentes o aplicaciones en distintas formas. Cuando se ponen estos servicios a través de Internet se tiene la base de la computación en la nube.

En [25] la computación en la nube se describe como un medio para obtener servicios de sistemas de información sin ser experto en la tecnología que hay detrás de ella; ofrece servicios para las necesidades de los consumidores y de los negocios, brinda escalabilidad ilimitada y servicios diferenciados. En el modelo

SaaS, el proveedor despliega el software en su propio servidor o en la nube a través de un servicio de terceros, este modelo sobre la nube y el sistema de “pago por uso” ayuda al proveedor a reducir la inversión en infraestructura pero el cliente depende del proveedor respecto a que el proveedor es quien se asegura de cubrir la seguridad de que el resto de los clientes no vean los datos de los demás medidas de seguridad En la tabla 2.1 se muestran las soluciones actuales a los problemas comunes de seguridad que se presentan en la computación en la nube.

Tabla 2. 1 Las soluciones actuales disponibles para asegurar el Software como servicio

No	**Área de seguridad**	**Solución**
1	Autenticación y autorización	• Open Authorization • 2FA, Two Factor Authentication • OAuth
2	Disponibilidad	Dispersión de datos
3	Confidencialidad de los datos	Encriptación proxy basada en atributos
4	Seguridad de máquina virtual	Máquina virtual distribuida reconfigurable
5	Seguridad de la información	Framework de administración de seguridad de la información
6	Estándares de la nube	• IEEE estándar de computación en la nube • Alianza de seguridad en la nube (CSA)
8	Acceso a datos	• Políticas de seguridad multiusuario • Administración de acceso a datos
9	Seguridad de aplicación Web	Escáner de aplicación Web
11	Back up	Método sin agente de copia de seguridad y recuperación de datos
12	Administración de identidad y proceso de autenticación	Guía de acceso y Gestión de identidad

2.2 Análisis comparativo

A continuación, se presenta el análisis comparativo del estado del arte en la tabla 2.2.

Tabla 2. 2 Análisis comparativo del estado del arte

Autores	Artículo	Problema	Objetivo	Estado actual y futuro
Fedrizzi, Mario, Andrea Molinari, and Viviana Ventre [10].	A Model for Evaluating the Transaction Risk in EBanking.	Incremento de fraudes en el sector de banca electrónica debido al crecimiento de las transacciones	Construir un modelo de Markov que cumple con el acuerdo de capital “Basilea 2” para evaluar el riesgo y generar protección a las entidades	En fase de pruebas en diferentes bancos.
Darwish, Saad M., and Ahmed M. Hassan [11].	A Model to Authenticate Requests for Online Banking Transactions.	Mantener la confianza de los usuarios y la seguridad de los servicios de banca en línea	Autenticar al usuario una sola vez, incluyendo una respuesta única del servidor con el algoritmo IB-mRSA	Terminado.
Hosburgh, Matthew [12].	Protecting small business banking.	Ataques a las máquinas de los usuarios.	Recomendar mejores prácticas para los usuarios	Terminado.

Autores	Artículo	Problema	Objetivo	Estado actual y futuro
			finales en el uso adecuado de las herramientas anti fraudes.	
Khu-smith, Vorapranee, and Chris J Mitchell [13].	Using GSM to Enhance E-Commerce Security	Transacciones de comercio electrónico desprotegidas	Usar del servicio GSM, para proporcionar la autenticación del usuario.	Terminado.
Wes Maseri, Sam Sleiman [14].	SQLPIL: Sql Injection prevention by input labeling	Ataque *SQL Injection* en la Web.	Usar de SQLPIL para prevenir este tipo de ataque	En un futuro se estudiará el caso sobre otras plataformas.
M.I.P Salas, E. Martins [15].	Security Testing Methodology for vulnerbilities detection of XSS in Web Services and WS-Security.	Vulnerabilidades en los Servicios Web.	Usar del protocolo WS-Security y la especificación *Security Tokens*, para validar peticiones de SW.	Probar otro tipo de vulnerabilidades sobre Servicios Web.
Dimitrios Zissis. Dimitrios Lekkas[16].	Addressing cloud computing security issues.	Deficiencia en los mecanismos de seguridad tradicionales, en la Web.	Implementar mecanismos para controlar las vulnerabilidades	Terminado

Autores	Artículo	Problema	Objetivo	Estado actual y futuro
			de seguridad	
Douglas k. Barry[17].	Web Services, Service-Oriented Architectures and Cloud Computing. The Savvy Manager´s Guide.	Comunicación entre diferentes sistemas en la nube.	Conocer la arquitectura orientada a servicios y los protocolos que utiliza	Terminado

Con base a la comparativa del estado del arte se observa que existe investigación en el campo de las aplicaciones basadas en Web, respecto a seguridad en las transacciones, pero poca aplicación en el área de las entidades financieras no bancarias, por lo cual la propuesta del desarrollo de una aplicación Web de *eBanking* contemplando estándares de seguridad y de distribución de software es factible y contribuye a la mejora de la calidad de los servicios que se ofrecen a la población.

2.3 Propuesta de solución

La solución propuesta es utilizar JavaServer Faces para el desarrollo de la aplicación, PrimeFaces para los componentes visuales, en cuanto a la comunicación con la aplicación OpenFIN se utilizarán Servicios Web basados en SOAP asegurando los estándares de *WS-Security* que apliquen, trabajando bajo la metodología de desarrollo SCRUM adaptada para una persona. Ver la tabla 2.3.

Tabla 2. 3 Propuesta de solución

Framework de Trabajo	**Servicios Web**	**Seguridad**	**Metodología de desarrollo**
JavaServer Faces + Prime Faces	SOAP	*WS-Security*	SCRUM

Esta solución se propone tomando en cuenta que se basa en estándares internacionales abiertos para cubrir la seguridad que se requiere en los entornos de la banca en línea. La metodología indicada es multidisciplinaria y permitirá cubrir los roles para el desarrollo del proyecto en todas las fases. La figura 2.1 esquematiza la arquitectura planteada.

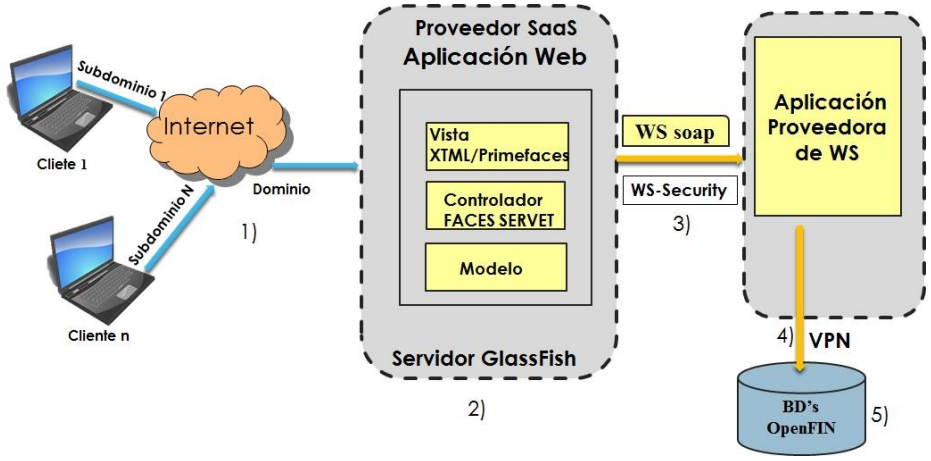


Figura 2. 1 Esquema de arquitectura

.

Se pretende que la aplicación de medios electrónicos de pago se provea a los clientes de SINC como un servicio contratado extra al uso de la aplicación de escritorio adquirida, por lo cual se llega a un modelo de Software como Servicio (SaaS, *Software as a Service*); siguiendo la arquitectura de aplicación que se describe en [9] los componentes son:

1. Los clientes o socios de cualquier EFNB (entidad financiera no bancaria), acceden a la aplicación a través de una dirección Web identificada dependiendo de la entidad a la que están afiliados, ésta dirección es un subdominio que se redirige al dominio principal.
2. Un servidor de aplicaciones Web GlassFish, en el cual estará alojada la aplicación de medios electrónicos de pago, se llega a ella mediante una conexión segura HTTPS.
3. Un servidor de aplicaciones Web GlassFish que aloja la aplicación proveedora de los servicios Web y que se conecta de forma segura dentro de la Intranet.
4. Enlaces VPN (*Virtual Private Network*, Red Privada Virtual) hacia las bases de datos de las diferentes entidades financieras.
5. Bases de datos que contienen la información financiera de cada socio de cualquier entidad financiera.

Justificación de la solución indicada

Considerando que la aplicación servirá para diferentes empresas, se contempla un gran número de transacciones por lo cual se seleccionó el estándar de programación de aplicaciones Web basadas en Java, que es JavaServer Faces; debido a que es una aplicación sobre Internet se contempla el cumplimiento de los requisitos de seguridad adecuada en los WS; así mismo, previendo las necesidades futuras se cubre el requisito de que es una aplicación responsiva accesible a través de cualquier dispositivo electrónico tal como *tablets*, teléfonos inteligentes, entre otros. En cuanto a la metodología de desarrollo indicada, define claramente las responsabilidades de cada uno de los diferentes roles de trabajo para llevar a cabo el desarrollo, aunque en este caso es una adaptación para una persona, se tienen definidas las actividades a ejecutar. Otra ventaja de SCRUM es que permite que se obtenga un entregable en corto tiempo debido a su enfoque iterativo, el cual el cliente probará y, si lo considera necesario, solicitará algún

cambio en los requisitos sin que éstos representen un problema ya que la metodología es adaptable a los cambios.

Capítulo 3 Aplicación de la metodología

En este capítulo se presentan los *sprints* llevados a cabo para la construcción de la aplicación Web de *eBanking*, con base en la pila del producto realizada con la empresa SINC, a partir de la cual se realizó un análisis de requerimientos que dio paso a la realización de cada sprint de manera más detallada.

Para dar inicio a la aplicación de la metodología se realizó un análisis a grandes rasgos de la solución actual, en este caso el sistema principal, del cual forma parte el módulo de *eBanking*, es el sistema OpenFIN, una aplicación de escritorio cuyo núcleo principal está desarrollado en los lenguajes de programación C y PL y se instala en servidor con sistema operativo Ubuntu versión 14.04. La Base de datos del sistema se aloja en un servidor PostgreSQL y la versión que se utiliza actualmente es la 9.3. El sistema no está disponible mediante Internet, se accede mediante una sesión de VNC (*Virtual Network Computing*, Computación Virtual en Red), o bien, desde un acceso directo en el escritorio de la computadora; aunque la solución no está disponible por Web, se le da soporte continuo a cualquiera de las empresas que cuentan con este sistema accediendo mediante un túnel VPN (*Virtual Private Network*, Red Privada Virtual). Cómo parte del sistema OpenFIN, la aplicación Web de *eBanking*, ayuda a los usuarios finales a realizar operaciones desde su propia computadora mediante Internet. En primera instancia se requirió una aplicación que fuera utilizada mediante Internet, para estar al alcance de cada usuario de una EFNB desde cualquiera que fuera su ubicación. Dicho usuario es proveniente de cualquier EFNB que sea parte de los clientes de la empresa SINC y podrá realizar consultas de sus saldos, generar los reportes correspondientes de tales consultas y realizará transacciones de transferencias y pagos a sus créditos, así mismo es deseable cierta gestión de contraseñas así como su renovación o cambio y configuración de cuentas.

Pila del producto

Derivado de la primera revisión de lo que sería deseable en la aplicación de *eBanking*, surgió la pila del producto que contiene lo que el sistema requiere, de acuerdo a lo que la empresa SINC solicita como requisitos para considerar que la aplicación funciona correctamente.

En la tabla 3.1 se presenta la pila del producto generada y en ella también se plasma la importancia que se le asigna a cada aspecto que la aplicación Web debe de cubrir.

Tabla 3.1 Pila del producto

ID	Nombre	Importancia	Descripción	Cómo probarlo
1	Consulta de saldos	Alta	Consultar el resumen de saldos y el detalle movimientos de las cuentas	Realizar una consulta y descargar los reportes
2	Obtener un *token*		Enlazar la cuenta del cliente con su celular para generar *tokens*	Realizar la sincronización y generar *tokens* validos
2	Transferencias	Alta	Transferir fondos entre sus cuentas y entre cuentas de terceros	Realizar transferencias entre ambos tipos de cuentas
3	Pagos	Alta	Realizar pagos de crédito	Realizar un pago, verificar el saldo afectado
4	Configuración de montos	Moderada	Configurar los montos máximos a retirar por día y por periodo	Que el usuario pueda establecer el monto máximo a retirar

ID	Nombre	Importancia	Descripción	Cómo probarlo
5	Cambio de contraseña	Moderada	Cambiar la contraseña de acceso del usuario	Que el usuario pueda cambiar su contraseña actual

Para la pila del producto presentada se definieron los siguientes *sprints*: 1) El análisis de los requerimientos de la aplicación, 2) Especificación de la arquitectura del software, 3) Los mecanismos de seguridad a implementar que nos garanticen la autenticación, confidencialidad, integridad y el no repudio, 4) la Definición de los WS, 5) Desarrollo de los WS, 6) Creación de los clientes de los WS, 7) integración de la aplicación cliente y la aplicación proveedora de WS. Al cabo de dichos *sprints* se tendrá la solución completa.

3.1 Análisis de requerimientos

En el análisis de requisitos se definieron las operaciones que los usuarios de la aplicación Web van a realizar, para tal efecto, se crearon los casos de uso que representan las acciones que los diferentes usuarios de la aplicación van a ejecutar en la misma. En la figura 3.1 se muestran los tipos de usuarios y las acciones que podrán ejecutar.

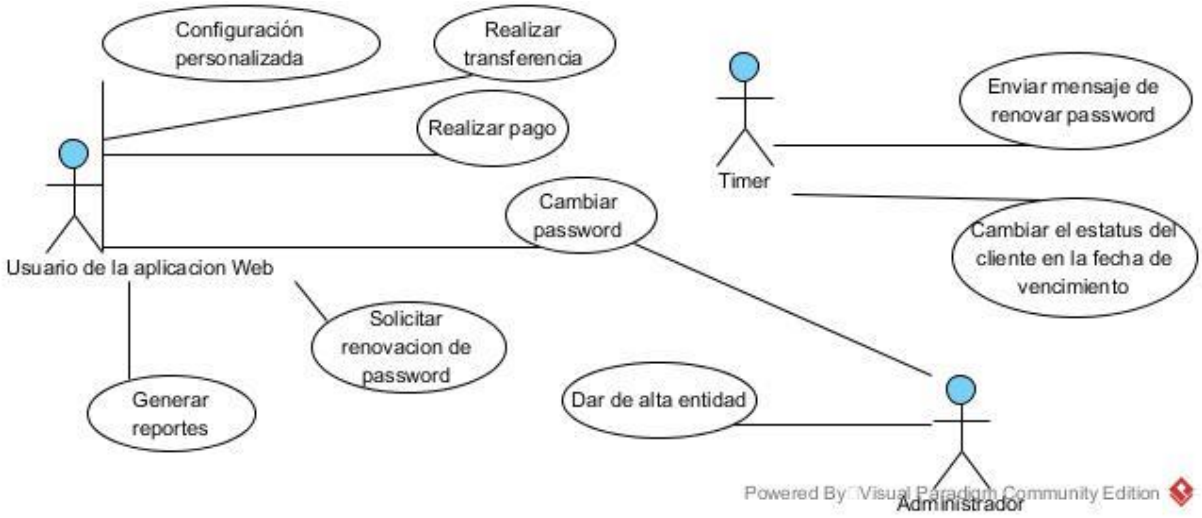


Figura 3. 1 Diagrama de casos de uso

Como se observa, existen dos tipos de usuarios y cada uno lleva a cabo ciertas operaciones en la aplicación, pero para el caso de acciones automáticas se hace referencia al calendarizador de tareas del sistema (*Timer*), el cual bajo determinadas condiciones de tiempo será un auxiliar para que el usuario final realice ciertas acciones o para evitar que un usuario con una contraseña vencida acceda a la aplicación cambiando su estatus.

3.2 Arquitectura de software

Representa la estructura de la aplicación Web de *eBanking* que dirige el desarrollo de la aplicación, así como la interacción de los componentes que la conforman y como se relacionan para presentar la funcionalidad general. En la figura 3.3 se muestra la arquitectura de software.

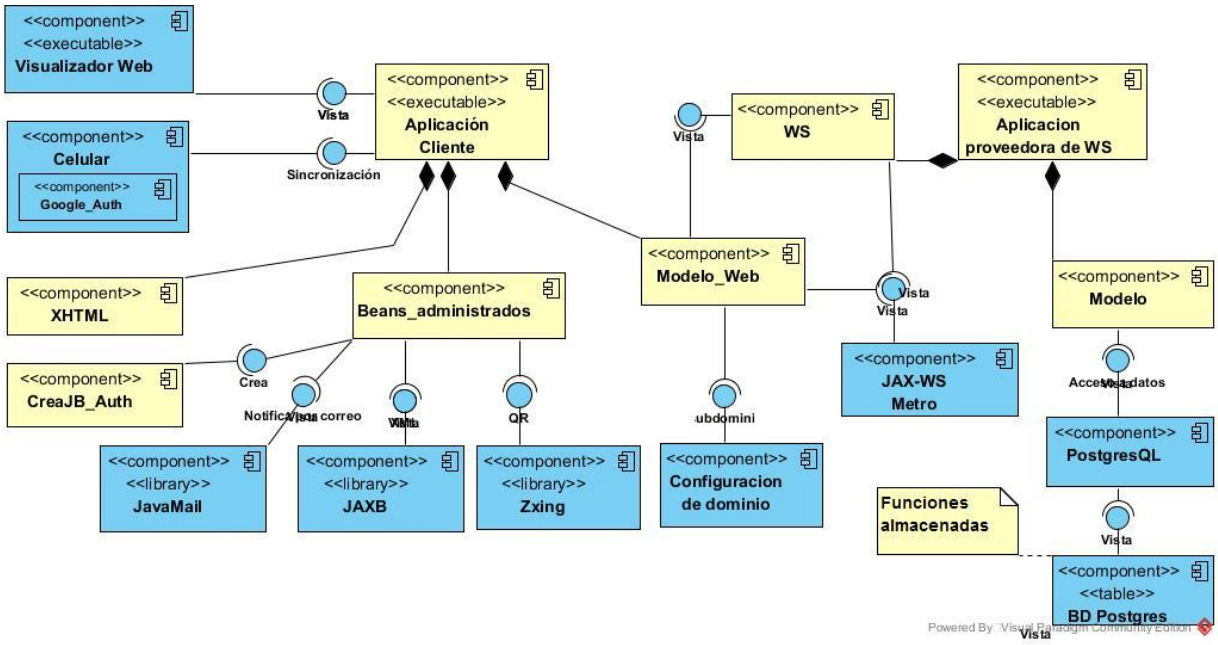


Figura 3. 2 Arquitectura de software

Como se aprecia en la figura anterior, se tienen dos componentes principales que son a) la aplicación Web cliente y los componentes que utiliza y b) la aplicación proveedora de WS, que es quien despliega los WS que utiliza la aplicación cliente para mostrar los datos necesarios al usuario final. Así mismo se muestran dos grupos de componentes, diferenciados por los colores azul y amarillo, donde los de color amarillo son los creados al escribir código y los azules son bibliotecas o artefactos usados para completar varias tareas importantes como conexión a bases de datos, generación de código QR, envío de correo electrónico y generación firma digital.

3.2.1 Vista de implementación:

Muestra los componentes que interactúan en la aplicación de *eBanking* de forma muy general. En la figura 3.4 se aprecia la vista de implementación:

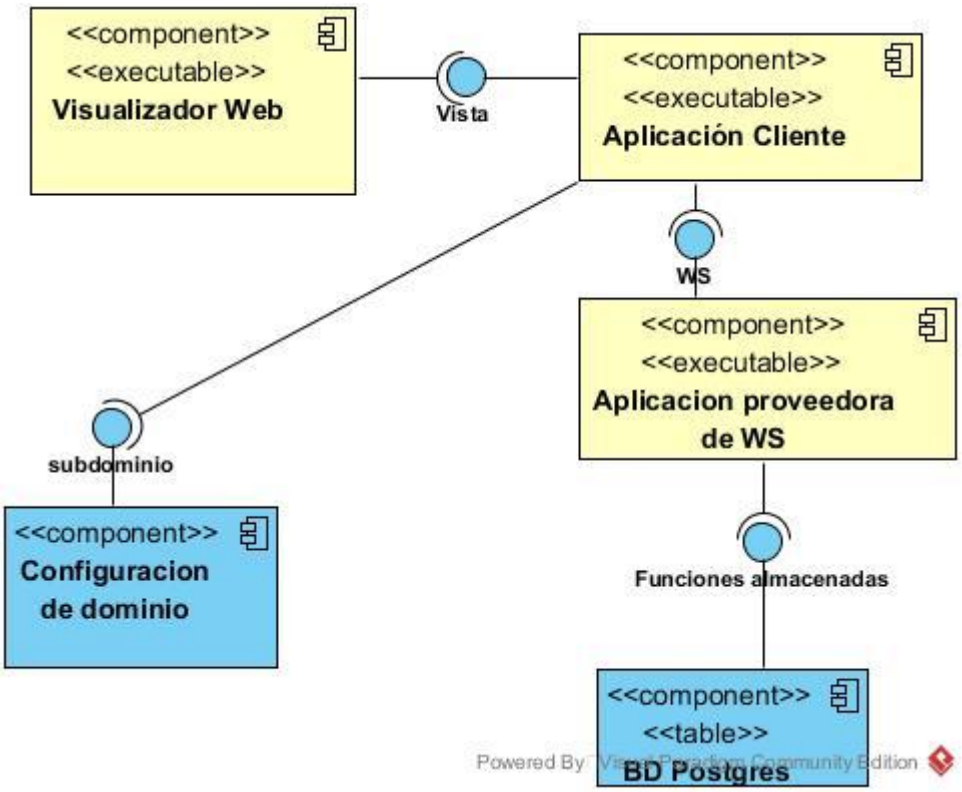


Figura 3. 3 Vista de implementación

En orden de aparición en el la figura anterior 1) el visualizador Web representa al *browser* que el usuario final utiliza para ver la aplicación Web de *eBanking* e interactuar con ella ejecutando cualquiera de las operaciones que le correspondan dependiendo del usuario que se trate; 2) Aplicación cliente representa la aplicación Web que visualiza el usuario de la EFNB donde llevará a cabo las operaciones deseadas con las cuentas de las que es dueño; 3) Aplicación proveedora de los servicios Web, como su nombre lo indica es la encargada de desplegar los WS que devuelven datos del usuario; 4) La configuración de dominio es un componente existente en la infraestructura de la empresa SINC, que provee los nombres de dominio de las diferentes entidades financieras que consumirán como un software como servicio la aplicación de *eBanking*; 5) La BD PostgreSQL es la Base de datos de la entidad financiera en particular, que contiene las funciones almacenadas que reciben o entregan datos mediante la aplicación Web de

eBanking. Nuevamente se presentan en color azul los componentes externos utilizados y en color amarillo los componentes desarrollados propiamente.

Diagrama de clases

Para obtener la vista estática de la aplicación, se realizó el diagrama de clases correspondiente, separando en dos paquetes las clases que se utilizan en una u otra aplicación (figura 3.2).

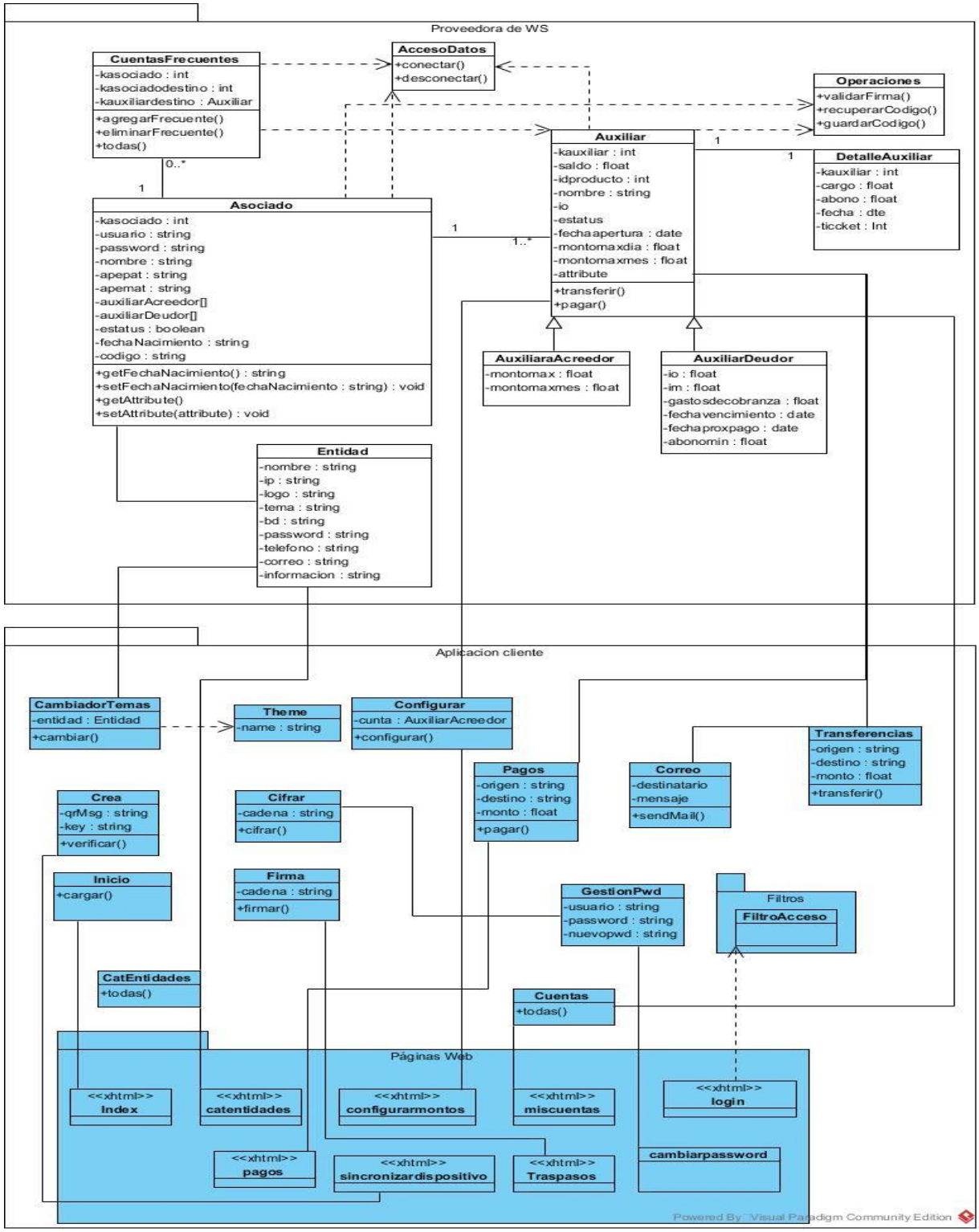


Figura 3. 4 Diagrama de clases

Se muestra en color blanco las clases que contempla la aplicación proveedora de WS y en color azul las clases que conforman a la aplicación de *eBanking*. En la sección del proveedor de WS se aprecia la clase de Asociado, como la clase que representa al usuario dueño de las cuentas de ahorro y crédito y la clase Auxiliar, que representa a un producto financiero, en este caso se divide en Auxiliares acreedores que son los productos de ahorro y auxiliares deudores que representan a los productos de crédito que puede poseer un usuario de la EFNB, las cuales también son utilizadas por la aplicación cliente. En la sección de la aplicación cliente se las clases del control y las páginas web con las que interactúa el usuario.

3.3 Definición de los servicios Web

El servicio Web que se contemplan para realizar las operaciones de consulta, transferencias, pagos y administración de cuentas propias, es un servicio basado en SOAP, como ya se expuso en el capítulo 2, a continuación, se enlista una breve descripción de cada una de las operaciones que se van a desplegar:

1. wslogin.- Devuelve el número de socio a partir de un inicio de sesión en la aplicación Web, mediante un usuario y una contraseña.
2. wsestadoCtaAcreedoras.- Devuelve Todas las cuentas acreedoras o deudoras del cliente autenticado en la aplicación.
3. wsestadoCtaDeudoras.- Devuelve Todas las cuentas acreedoras o deudoras del cliente autenticado en la aplicación.
4. wstransferencias.- Permite realizar transferencias de saldos de una cuenta a otra, ya sea cuentas propias o hacia cuentas de terceros, así mismo permite el registro de una cuenta de terceros que vaya a ser utilizada como cuenta frecuente para realizar transferencias.
5. wspagos.- Registra el pago de un abono a una cuenta de crédito, perteneciente al titular de la cuenta que se encuentra en sesión.

6. wsguardarCodigo.- Permite guardar el código de sincronización entre la aplicación de celular del titular de la cuenta y la aplicación Web.
7. wsrecuperarCodigo.- Permite recuperar el código de sincronización entre la aplicación de celular del titular de la cuenta y la aplicación Web, para validar los *tokens* ingresados al momento de solicitar una transacción.
8. wscambiarpassword.- Permite realizar el cambio de la contraseña del usuario actual.
9. wscuentasfrecuentes.- Devuelve la lista de cuentas frecuentes del cliente actual.
10. wsguardafrecuentes.- Permite almacenar una nueva cuenta frecuente.
11. wseliminafrecuentes.- Permite eliminar una cuenta frecuente.
12. wsconfiguramontos.- Permite establecer montos máximos de retiro, diarios y por periodo, en las cuentas de ahorro del usuario.
13. wsdetalleauxiliar.-Muestra el detalle de movimientos de los dos periodos más recientes del usuario propietario de una cuenta en particular
14. wsdetallemontosmax, lista de cuentas y sus montos máximos

El diagrama de descripción de los servicios muestra las operaciones que mencionadas de cada servicio y cómo es desplegado por la aplicación proveedora y consumido por la aplicación cliente

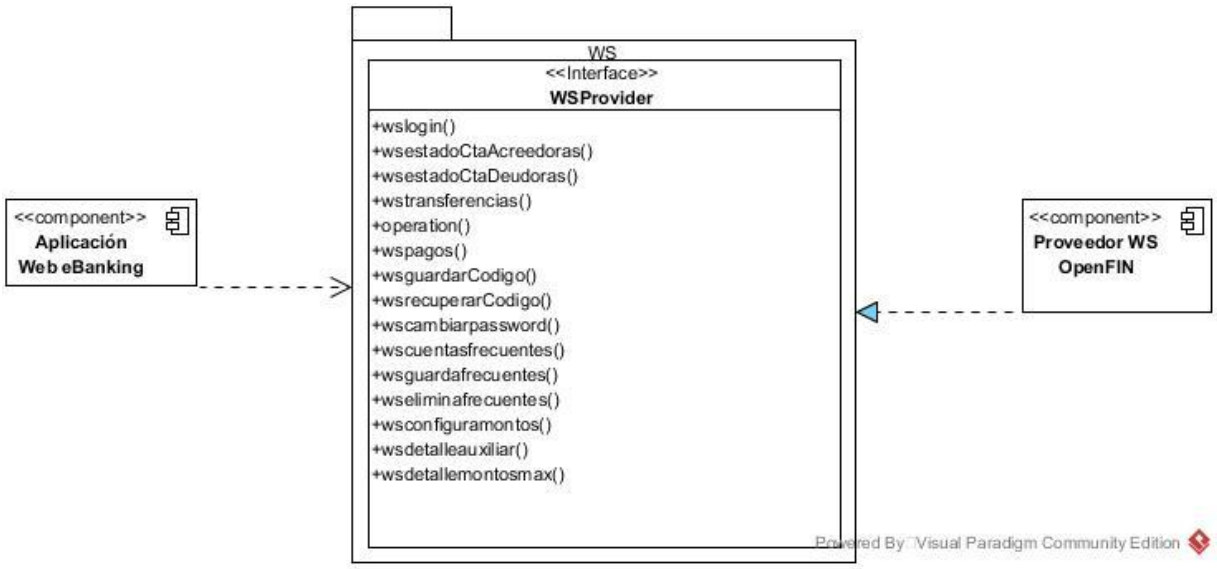


Figura 3. 5 Descripción del Servicios Web

3.4 Seguridad implementada en la aplicación Web de *eBanking*

En las aplicaciones basadas en Web, lo óptimo es cumplir con los siguientes aspectos al hacer uso de servicios Web. Del lado del cliente también es necesario garantizar que las peticiones hechas con las cuentas del usuario sean autorizadas por el mismo o asegurar que quien las está solicitando sea el dueño de la cuenta de la que se trate. En la aplicación Web de *eBanking* los mecanismos que se implementan son los siguientes:

1. Uso de conexión segura SSL para garantizar el no repudio

La aplicación se accede a través de un enlace SSL (*Secure Socket Layer*) que provee un canal de comunicación seguro, para cuando la información está en tránsito. El servidor se autentica con el cliente, envía un certificado para verificar que es auténtico, éste es el enfoque utilizado para las transacciones de banca en línea. SSL usa una combinación de cifrado de clave pública y privada para

comunicaciones seguras, la seguridad en el nivel de aplicación complementa la seguridad del nivel de transporte.

2. Para garantizar la confidencialidad e integridad de los mensajes SOAP

Se utiliza firma digital. Las firmas digitales emplean las técnicas de cifrado, pero el fin de la firma es distinto. En lugar de cifrar el mensaje, una firma le da confianza al destinatario en dos trozos de información: el remitente del mensaje, y el mensaje mismo. El remitente cifra el mensaje con su clave privada. Esta "firma" luego se envía al destinatario junto al mensaje original. El destinatario intenta verificar la firma, lo que significa descifrar la firma con la clave pública del remitente, y comparar los resultados con el mensaje original. Si este proceso tiene éxito, el destinatario está seguro de que solo el remitente es el originador del mensaje, porque nadie más, excepto el remitente, tiene su clave privada. Si el mensaje original hubiera cambiado en su traslado por la red, la firma descifrada no coincidiría, así que una verificación de firma exitosa significa también que el mensaje no ha sido interferido. Generalmente, en vez de firmar todo el mensaje, el remitente calcula un "resumen" del mensaje, firma ese resumen, y lo pasa junto al mensaje descifrado. Esto brinda el mismo efecto sin tener que duplicar el tamaño de cada mensaje y también reduce el tiempo de procesamiento para crear y verificar las firmas.

3. Para asegurar la autenticación y el control de acceso:

Se emplean mecanismos de autenticación mediante usuario y contraseña para el inicio de sesión en la cuenta del usuario final. Por otra parte, se solicita un token de seguridad para autorizar transacciones invocadas por el usuario, el mecanismo de la autorización de transacciones funciona de la siguiente manera:

i. La aplicación Web cliente genera una clave única y la presenta en pantalla mediante un código QR.
ii. Se requiere el uso de un teléfono inteligente en el cual tendrá que instalarse la aplicación de *Google Authenticator*, ésta leerá el código QR generado por la aplicación Web para sincronizar la cuenta del usuario de la aplicación Web con la aplicación de *Google Authenticator*.
iii. Una vez sincronizadas, la aplicación de *Google Authenticator*, genera *tokens* generados a partir de la clave que se leyó de la aplicación Web.
iv. Cuando el usuario titular de la cuenta solicite una transacción se verificará la autorización de dicha transacción, con base a verificar el token generado en la aplicación que se crean con base a una llave generada en la misma aplicación comparándola con la generada por el dispositivo movil.

4. Notificación de movimientos

Se utiliza la biblioteca Java Mail para enviar correos de notificación al usuario cada vez que se efectúe una operación con su cuenta de ahorro. La información que se presentará es un aviso de realización de una transacción con el detalle del número de la cuenta origen, la cuenta destino y el monto de la transacción.

5. Manejo de sesiones entre capas

El manejo de objetos de sesión, permite conocer durante las peticiones, el id de sesión del cliente que está haciendo dichas solicitudes, desde que inicia sesión, durante el transcurso de las peticiones y hasta que termine la sesión. Después de que el cliente introduce sus credenciales, (usuario y contraseña) se genera un ID de sesión en el cliente y se envía durante la autenticación hacia el proveedor de WS. Si el proveedor de WS verifica que el usuario existe y el proceso de autenticación fue exitoso, entonces el ID de sesión se almacena en el proveedor de los WS. Posteriormente, cuando el usuario solicite una petición al proveedor de

WS, la aplicación cliente siempre enviará el ID de sesión y el proveedor de WS siempre verificará dicho ID comparándolo con el que tiene almacenado. Si no coinciden, el proveedor de WS rechazará la petición.

6. Cifrado de contraseñas

En este apartado se utilizó la autenticación con usuario y contraseña, pero dado que estos datos permiten el acceso hacia información personal delicada, la contraseña utilizada debe ser secreta y por ende debe evitarse que otra persona la conozca. Por lo anterior, se utilizó el cifrado de contraseña con el algoritmo MD5, de tal manera que la aplicación de *eBanking*, al recibir los datos por parte del usuario, de inmediato cifra la contraseña, luego la envía ya cifrada a la aplicación proveedora de WS y a su vez, la envía cifrada hacia la base de datos.

Esquema de la base de datos

Al esquema de BD que se utiliza actualmente con el sistema OpenFIN fue necesario anexar algunas tablas que permitirán registrar las acciones ejecutadas por el usuario dentro de la aplicación de *eBanking*, así como también delimitar las operaciones efectuadas por los usuarios que tienen una cuenta dentro de la aplicación de eBanking de tal manera que no se mezcle con las operaciones efectuadas por el sistema mediante la aplicación de escritorio y de esta manera también evitar campos nulos en las tablas originales, dichas tablas de nueva creación registran operaciones como son inicio de la sesión y transacciones realizadas con sus cuentas, gestión de contraseña y sincronización del dispositivo móvil. En la figura 3.2 se muestra el esquema de la base de con las tablas de nueva creación.

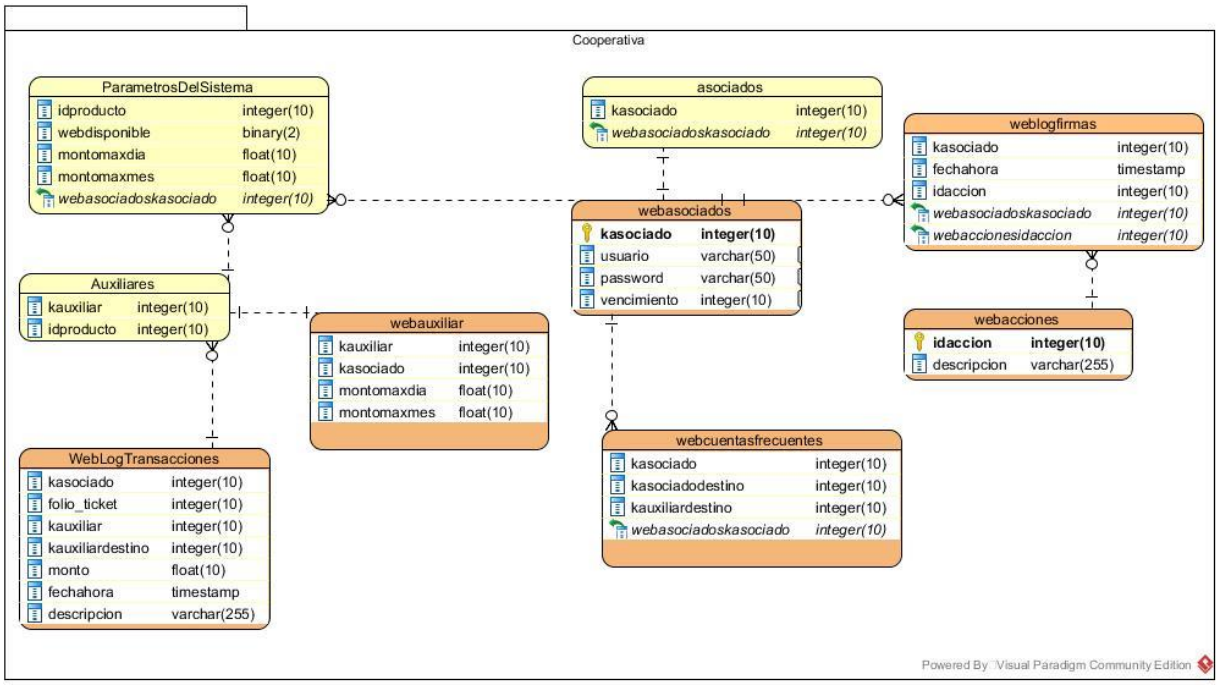


Figura 3. 6 Tablas de nueva creación necesarias para el registro de operaciones mediante la aplicación Web

En la figura anterior, se muestran las tablas adicionadas al esquema de base de datos original, donde las tablas de color amarillo son parte del esquema original y las tablas de color naranja fueron creadas para poder interactuar con la aplicación de *eBanking* y se relacionan con las tablas originales para obtener datos históricos o datos que se actualizan en la operación diaria de una EFNB. Cabe mencionar que ni la aplicación de *eBanking* ni la aplicación proveedora de los WS llegan directamente a la base de datos de la entidad financiera, para acceder a sus datos o enviar datos se hacen invocaciones a varias funciones almacenadas creadas por la emprea SINC, para asegurar que no se tenga acceso directo a la base de datos.

Funciones almacenadas

Cómo parte de los mecanismos de seguridad se utilizan las funciones almacenadas para obtener los datos del usuario tales como cuentas de ahorro y crédito, saldos de las cuentas y para el envío de transacciones hacia la base de datos. La aplicación proveedora de WS hace llamadas a las funciones

mencionadas, creadas en la base de datos propia de la entidad financiera y de esta forma se evita escribir comandos o consultas directas hacia la base de datos para poder obtener o enviar operaciones; además, dichas funciones las desarrolló el personal de la empresa SINC como una medida de seguridad respecto a los datos, por lo tanto, los detalles de la construcción de dichas funciones son desconocidos y por tal motivo no se detallan en el presente documento, únicamente se invocan para obtener los datos requeridos en cada llamada solicitada en cada una de las operaciones contempladas en la aplicación *eBanking*. Cabe hacer notar que la interfaz de tales funciones se definió en conjunto entre el personal de SINC y la autora.

Las funciones almacenadas requeridas fueron las siguientes:

Función para	**Descripción**
Autenticación en sistema (*login*)	Verifica el usuario y contraseña ingresados y si son correctos, devuelve los datos del usuario
Estado de cuenta acreedores	Devuelve el listado de las cuentas acreedoras del usuario
Estado de cuenta deudores	Devuelve el listado de las cuentas deudoras del socio
Detalle de movimientos acreedor	Devuelve el detalle de movimientos efectuados en una cuenta acreedoras del usuario, durante los dos últimos meses
Detalle de movimientos deudor	Devuelve el listado de las cuentas deudoras del socio, durante los dos últimos meses
Transferencias y pagos	Registra los datos de la transferencia hacia cuentas y el monto de dicha transferencia, se utiliza esta misma función para registrar pagos

Función para	Descripción
Alta de cuentas de terceros	Registra una cuenta frecuente del usuario actual
Baja de cuenta de terceros	Elimina una cuenta frecuente perteneciente al usuario actual
Cambio de contraseña	Envía la nueva contraseña del usuario
Configuración de montos máximos de retiro diario	Registra los montos máximos de retiro por día
Configuración de montos máximos de retiro por periodo	Registra los montos máximos de retiro por mes

Con lo anterior se permite realizar las operaciones básicas para un usuario final de la aplicación de *eBanking* evitando las consultas directas hacia la base de datos.

3.5 Desarrollo de los servicios Web

Con base en el listado visto anteriormente en el apartado “Descripción del Servicio Web”, se desarrolló un servicio Web con las operaciones descritas, utilizando para ello la propia API (Interfaz de programación de aplicaciones) de Java, para obtener los servicios Web en XML cumpliendo con la especificación de las operaciones del mismo apartado de tal manera que dichas operaciones quedaran disponibles para ser consumidas por la aplicación Web cliente y ésta a su vez poder presentarlas al usuario final una vez que se tuviera el cliente de WS.

3.6 Creación de los clientes de los Servicios Web

Se desarrolló la programación para ejecutar la invocación de los WS en la aplicación cliente, para desplegar todas las operaciones propias de cada usuario,

posteriormente todas las operaciones se probaron para corroborar que reciben y que entregan la información correcta para cada petición del usuario final. En este caso se refiere a las pruebas de las operaciones básicas de un usuario de una EFNB (consultas, transferencias, pagos, cambio de contraseña y configuración de cuentas).

3.7 Integración de la aplicación cliente y la aplicación proveedora de los WS

Una vez que se finalizó con el desarrollo de los WS y el llamado de los mismos y la programación de las páginas de la aplicación Web de *eBanking*, se integró en su conjunto la solución. En la aplicación de *eBanking* se trabajó principalmente con la vista de las páginas Web, ya que era lo restante por crear una vez probada la la funcionalidad misma que será presentada en el capítulo de resultados.

Capítulo 4 Resultados

Como resultado de este trabajo de tesis se creó una aplicación Web que lleva por nombre “Aplicación Web de *eBanking* para entidades financieras no bancarias”.
Es una aplicación que se basa en una arquitectura orientada a servicios para realizar las operaciones básicas (consultas, transferencias y pagos) con las cuentas de ahorro y crédito de un usuario final de una EFNB. Debido a la naturaleza de la aplicación, cuenta con diversos mecanismos de seguridad; los ya vistos en los capítulos anteriores; que evitan los ataques más importantes en las aplicaciones Web.

En el presente capítulo se presenta el caso de estudio, con el objetivo de mostrar los resultados obtenidos con la aplicación.

Caso de estudio

El caso de estudio fue desarrollado con una EFNB real cliente de la empresa SINC y consistió en lo siguiente:

- Tener usuario registrado en el sistema OpenFIN,
- tener cuentas de ahorro y crédito activas pertenecientes al usuario,
- utilizar la cuenta de ese usuario para realizar las transacciones de consultas de saldos por detalle y resumen,
- generación de reportes,
- transferencias entre cuentas propias,
- alta y baja de cuentas de terceros,
- transferencias a cuentas de terceros,
- pagos y cambios de contraseña del usuario actual

Fue necesario verificar que funcionara correctamente la aplicación de *eBanking* al realizar las operaciones mencionadas y produjera los resultados esperados

verificando también que no fuera posible efectuar operaciones de ningún tipo, sobre todo las que afectan directamente el manejo de saldos de las cuentas si el usuario no estaba identificado en la aplicación, si el usuario no era dueño de la cuenta o si el usuario no fuera un usuario auténtico del sistema.

Para llevar a cabo el caso de estudio fue necesario que en la base de datos de la entidad financiera no bancaria se crearan las tablas adicionales a su esquema original, las cuales fungen como auxiliares para almacenar las operaciones realizadas por la aplicación Web de *eBanking*, también se crearon las funciones almacenadas y se cargaron en la misma base de datos, se asignó un usuario y una contraseña para entrar a la aplicación de *eBanking* y se asignó una fecha de vencimiento para el usuario, el proceso fue el siguiente: A un usuario existente en el sistema principal OpenFIN se le registró dentro del mismo OpenFIN, un correo electrónico que sirve como su nombre de usuario para la aplicación *eBanking* así como una contraseña pensada por él mismo, ambos datos son requeridos para iniciar sesión en la aplicación Web de *eBanking*. El proceso de registro de usuario y contraseña se realiza una única vez y solo lo puede hacer el personal del departamento de sistemas de la EFNB o algún usuario que tenga perfil de administrador dentro del sistema OpenFIN en la EFNB. Una vez que se registran los datos mencionados, el usuario inicia sesión en la aplicación de *eBanking* para realizar las operaciones que desee. En la figura 4.1 se muestra la pantalla del inicio de sesión para usuarios registrados.

Figura 4.1. Figura 4. 1 Ventana de inicio de sesión de usuarios registrados

Cuando el usuario inicia sesión en la aplicación le es posible realizar las operaciones que desea hacer en su sesión.

Sincronización del dispositivo móvil con la aplicación *eBanking*

Lo primero que un usuario debe hacer al utilizar la aplicación Web de *eBanking* es sincronizar su dispositivo móvil con la aplicación Web, de esta manera se generará una clave común entre ambas aplicaciones. En la figura 4.2 se muestra la página en la cual aparecen las instrucciones para el usuario acerca de cómo sincronizar su dispositivo móvil.

Figura 4. 2 Ventana de sincronización de dispositivos

Al realizar el proceso de sincronización se verificó que la aplicación Web crea una clave secreta, dicha clave sirve para generar un código QR el cual es escaneado por el celular del usuario y de esta manera quede sincronizada la cuenta con la aplicación del celular. Así mismo la clave secreta se almacena en la base de datos de la EFNB. En la figura 4.3 se muestra la generación del código QR y el proceso de guardar en la base de datos.

Figura 4. 3 Ventana de generación del código QR

La clave secreta que ya se almacenó en la base de datos será recuperada posteriormente para que la aplicación Web genere los *tokens* de autorización de transacciones y verifique que corresponden con los *tokens* mostrados en el dispositivo móvil para poder llevar a cabo una transacción.

Consulta de saldos de ahorro y crédito

En este apartado, la aplicación Web de e*Banking* llama al WS que regresa todas las cuentas de ahorro y al WS que regresa todas las cuentas de crédito. El WS a su vez hace una llamada a la función almacenada correspondiente y una vez que obtiene el resultado, devuelve un arreglo por cada tipo de cuentas. La aplicación de *eBanking* procesa estos datos para mostrar en pantalla el resumen de saldos, el cual es visto por el usuario como un listado de todas sus cuentas de ahorro que presenta el número de la cuenta, el nombre del producto, y el saldo total que tiene la cuenta a la fecha de la consulta, sin presentar detalles a los retiros o depósitos realizados con dicha cuenta. En el caso del crédito el proceso interno el mismo que en las cuentas de ahorro, solo que en éste caso se muestran productos de crédito y lo que el

usuario ve es el número de cuenta, nombre del producto, saldo total a la fecha actual, interés ordinario, interés moratorio, abono mínimo y fecha del próximo abono. En la figura 4.4 se muestra el módulo del resumen de saldos en la opción del menú “Mis cuentas”.

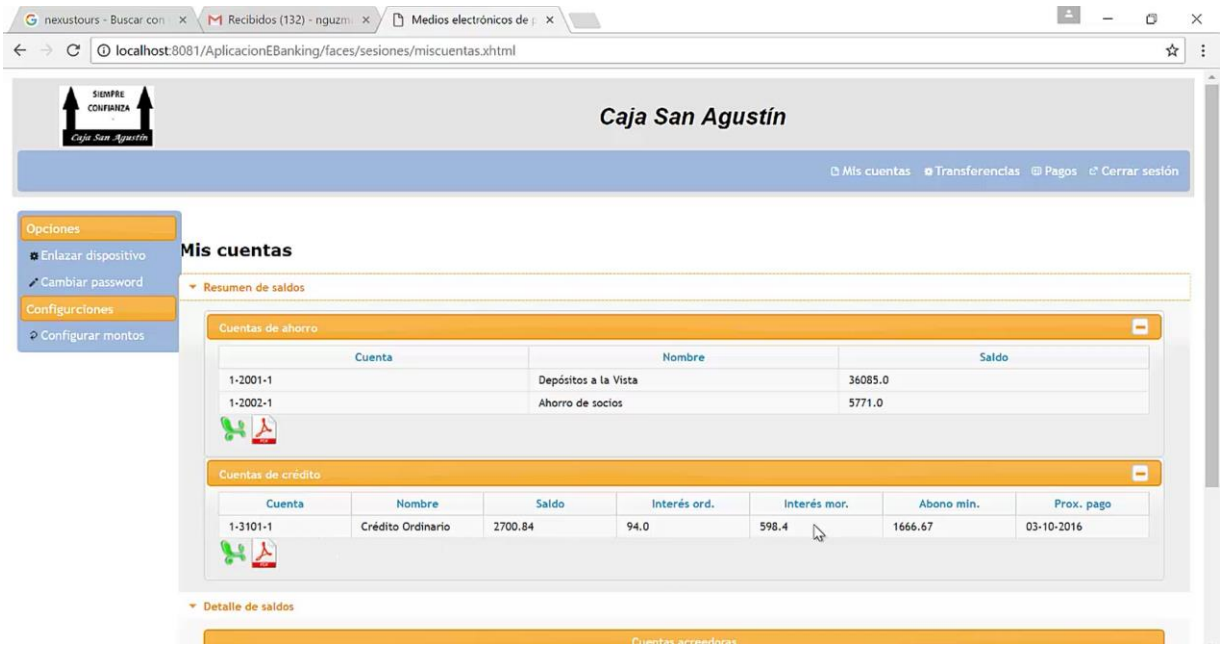


Figura 4. 4 Módulo de resumen de saldos de ahorro y crédito

Cada sección del resumen de saldos, tiene un botón para descargar el reporte en Excel o en PFD, para que el usuario cuente con un archivo de sus saldos. En la figura 4.5 se muestra el archivo con los saldos de ahorro de un usuario que tiene dos cuentas.

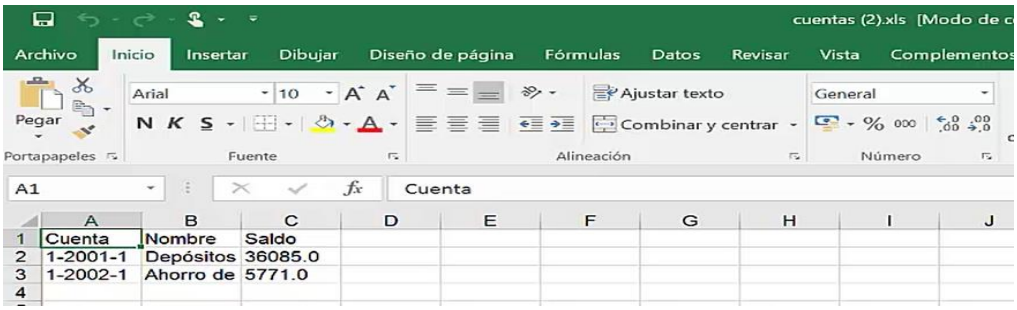


Figura 4. 5 Vista de un resumen de saldos en Excel

El detalle de movimientos aparece en una sección debajo del apartado de resumen de saldos, y esta sección presenta un botón por cada cuenta de ahorro o crédito, el cual genera una pantalla con el detalle de los movimientos efectuados sobre una cuenta, este detalle se calcula a partir de la fecha actual y dos meses hacia atrás, en la figura 4.6 se muestra la sección de detalle de movimientos.

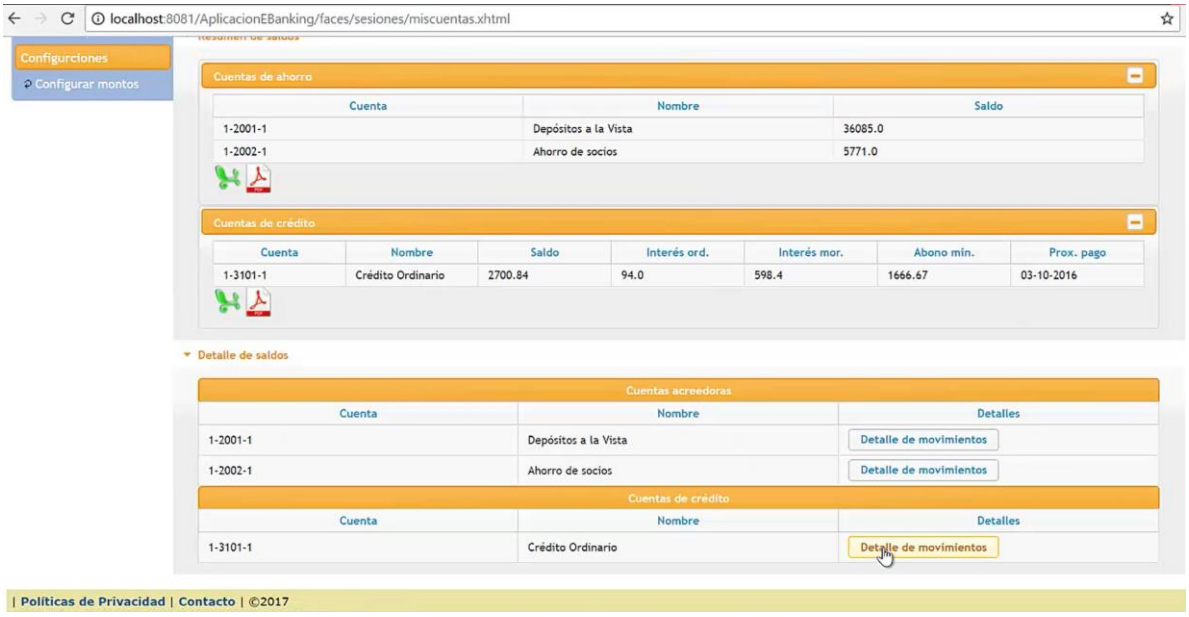


Figura 4. 6 Ventana de generación del código QR

El botón “Detalle de movimientos” invoca al WS que regresa un reporte con el detalle de cargos y abonos que se han hecho con una cuenta en particular; ya sea de ahorro o crédito; el concepto de la transacción y una referencia que indica hacia qué cuenta

se transfirió la cantidad retirada o desde qué cuenta se originó la cantidad depositada; el reporte muestra los retiros y abonos efectuados durante los dos últimos meses desde la fecha actual hacia atrás. En la figura 4.7 se muestra la figura del detalle de movimientos de la cuenta de un usuario.

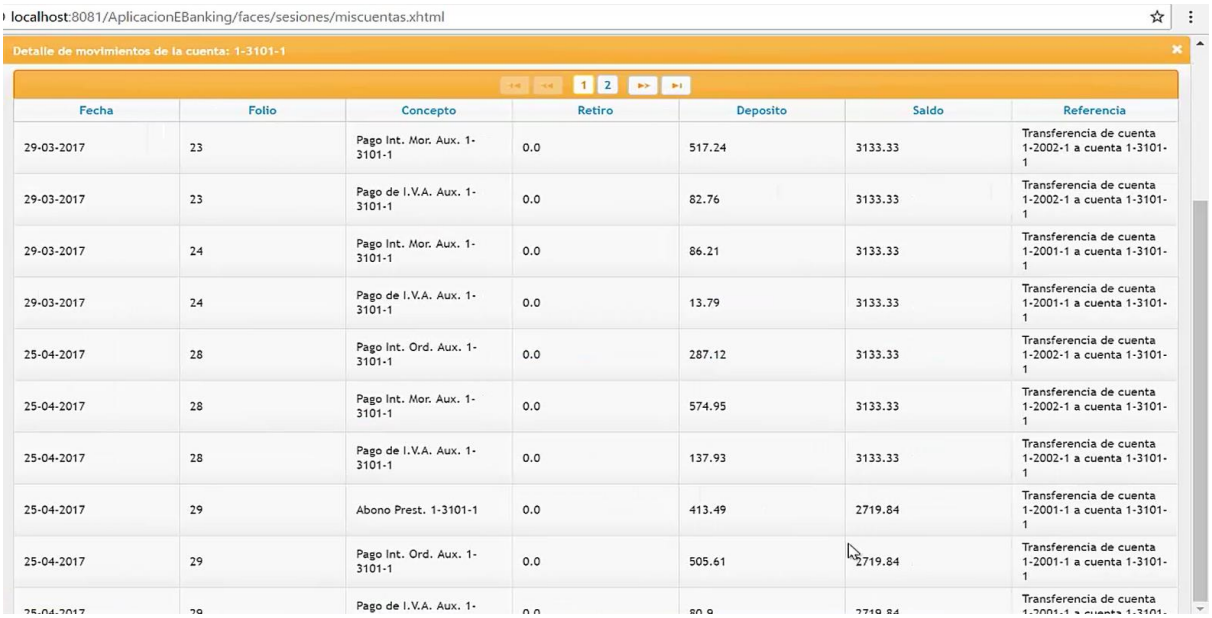


Figura 4. 7 Detalle de movimientos en la cuenta de un usuario

Transferencias

En la sección de transferencias, tal como su nombre lo indica, se lleva a cabo la operación de transferencias de saldos hacia cuentas propias del usuario, pero en el segundo apartado se encuentra la opción de transferencias hacia cuentas de terceras personas, siempre y cuando esas cuentas de terceros pertenezcan a la misma EFNB.

Ahora bien, cuando el usuario selecciona realizar una transferencia entre cuentas propias, nuevamente, para que las listas de las cuentas de ahorro sean visibles, se invoca al WS encargado de devolver las cuentas de ahorro del cliente. En la pantalla se visualizan dos listas desplegables, uno para seleccionar la cuenta origen y otro

para seleccionar la cuenta destino y en ambas listas aparecen las cuentas de ahorro del usuario de la sesión actual. Pero, aunque aparecen las mismas cuentas en ambas listas, la aplicación de *eBanking* verifica que, al seleccionar una cuenta de origen, no aparezca la misma en la lista de cuentas destino, debido a que no tiene sentido realizar una transferencia hacia la misma cuenta. Otros campos importantes en la sección de transferencias son, el campo de monto y el campo del código de autorización. Ambos están validados, el primero para aceptar números solamente y el segundo esta validado para que sean solo seis dígitos, debido a que el código de autorización siempre será de seis dígitos. También se manda un mensaje de error si los campos requeridos (marcados con un asterisco) no se llenan apropiadamente.

La figura 4.8 muestra la operación de transferencia, que cuando tiene los datos ingresados correctamente es exitosa. Mientras que en la figura 4.9 se muestra la generación de un *token* mediante el dispositivo móvil, el cual como se ha visto, funciona como un código de autorización para que una transacción proceda o no.

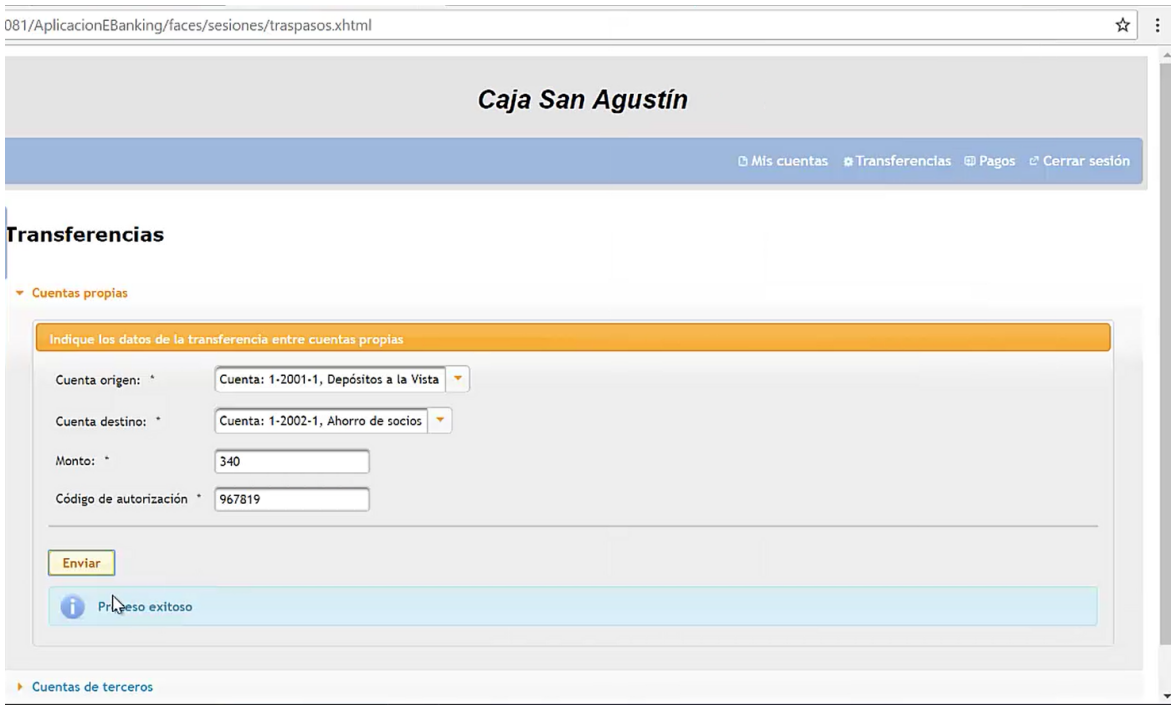


Figura 4. 8 Operación de transferencia entre cuentas propias

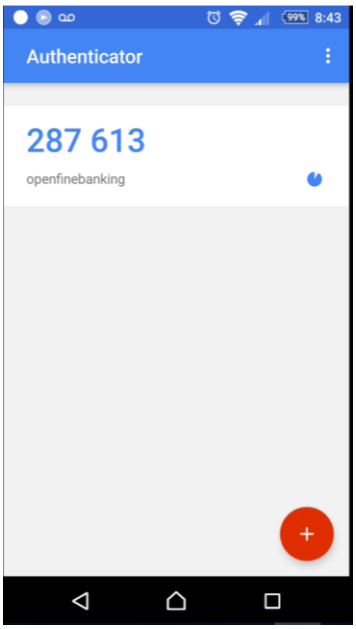


Figura 4. 9 Generación de un token mediante el dispositivo móvil

La aplicación de *eBanking* antes de permitir que una transacción se lleve a cabo, verifica que el código de seis dígitos ingresado por el usuario es correcto. Para comprobar que el código es el correcto, la aplicación Web de *eBanking* invoca a un WS que le regresa la clave secreta almacenada anteriormente, cuando se sincronizó con el celular o dispositivo móvil y con esta clave genera un código de seis dígitos dinámico, luego el usuario ingresa el código generado por la aplicación del celular y la aplicación de *eBanking* verifica si son coincidentes tanto la generada por ella y la introducida por el usuario, si es así entonces significa que corresponde al mismo usuario y que es auténtica. La clave cambia cada treinta segundos, por lo cual una vez generado el código en su celular, el usuario debe ingresar el código y enviarlo de inmediato para que se verifique. Se muestra un ejemplo en la figura 4.10 de la impresión en consola a modo de prueba, para ver el resultado de la verificación del código escrito por el usuario y el código generado por la aplicación Web.

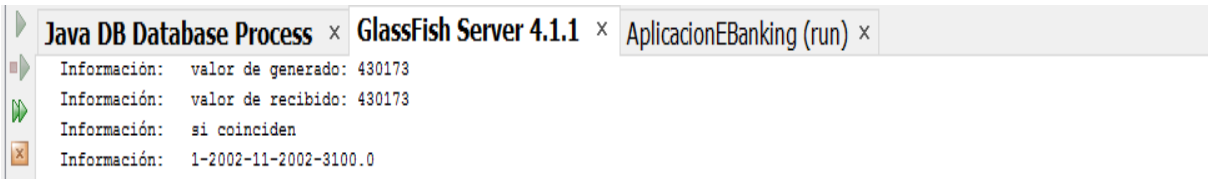


Figura 4. 10 Módulo de resumen de saldos de ahorro y crédito

Para el caso de las transferencias entre cuentas de terceros es necesario registrar la cuenta de terceros hacia la cual se realizará la transferencia. Cómo en todas las acciones realizadas por el usuario, el alta de una cuenta de ahorro de una tercera persona se realiza invocando un WS que recibe un número de cuenta, solo eso se requiere para guardarla. En este sentido la aplicación Web de *eBanking* no realiza ninguna verificación de que si la cuenta existe o no, ya que la parte encargada de dicha verificación es la función almacenada en la base de datos de la EFNB, si es una cuenta válida, entonces la guarda y de lo contrario envía un error. Ver la figura 4.11.

Figura 4. 11 Pantalla de alta de una cuenta de terceros

Cuando ya se guardó una cuenta de terceros, es posible realizar la trasferencia de la misma manera que se realiza una transferencia entre cuentas propias, sólo que, en

esta ocasión, en el menú de cuentas de destino, solo aparecerán las cuentas de terceros agregadas previamente. En la figura 4.12 se aprecia una pantalla de transferencias hacia cuentas de terceros.

Figura 4. 12 Pantalla de transferencias a terceros

La eliminación de cuentas de terceros, es un proceso sencillo para el usuario, ya que solo tiene que seleccionar la cuenta a eliminar y aceptar el proceso. Nuevamente en este caso la aplicación Web, invoca un WS que solicita la eliminación de la cuenta, esta petición llega hasta la función almacenada que elimina una cuenta de terceros y esa misma función almacenada es la que verifica que la cuenta existe y es parte de las cuentas de terceros que el usuario tiene como sus cuentas frecuentes. Como se ve en la figura 4.12 para eliminar una cuenta el usuario solamente selecciona cual desea eliminar.

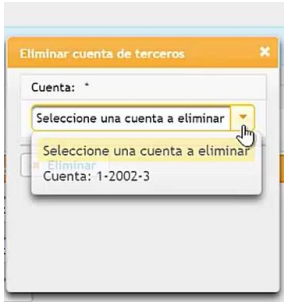


Figura 4. 13 Pantalla de eliminación de una cuenta de terceros

Pagos

La operación de pagos, funciona similar a la operación de transferencias, salvo que ahora en la sección de cuenta de destino, se invoca al WS que nos regresa todas las cuentas de crédito que el usuario tiene activas en el sistema OpenFIN, entonces, el usuario indica cuál de las cuentas de crédito será su cuenta de destino, es decir a cuál quiere depositar un abono, tomando el saldo de sus ahorro seleccionado. Además se solicita un código de autorización de la transacción, mismo que se valida antes de realizar la operación (figura 4.14).

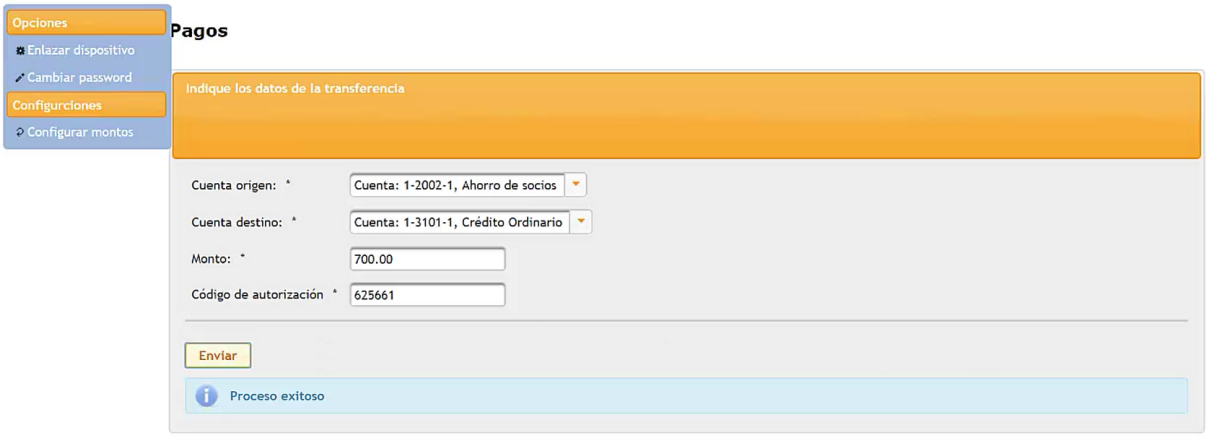


Figura 4. 14 Pantalla de pago de una cuenta crédito

Cabe señalar que, en este apartado solamente aparecerán las cuentas de crédito pertenecientes al usuario y no existe la opción de agregar cuentas de crédito de terceros. En todo caso si un tercero quiere pagar su crédito, el usuario titular de la cuenta solo podría realizarle una transferencia hacia su cuenta de ahorro y entonces el tercero ya podría tomar de su ahorro para pagar su propio crédito.

Cambio de contraseña

La sección de “Cambiar *password*” sirve para que el usuario cambie su contraseña si considera que la actual pudo haber sido robada o descubierta y es conocida por otra

persona. Esta operación invoca a un WS que recibe como datos de entrada el nombre del usuario, la contraseña actual y la nueva contraseña, así mismo el WS invoca a la función almacenada que recibe los mismos datos y verifica si el usuario y la contraseña actual coinciden, si coinciden entonces pone en su lugar la nueva. Algo de suma importancia en esta operación es que cuando el usuario ingresa sus contraseñas, inmediatamente la aplicación de *eBanking* aplica un algoritmo de cifrado, para evitar que las contraseñas sean enviadas en texto legible. En la siguiente tabla se muestran las instrucciones aplicadas, en donde la primera línea hace un llamado a la función cifrar, enviando la contraseña actual y lo recibe en una cadena ya cifrada y lo mismo aplica a la nueva contraseña, posteriormente se mandan ambos valores ya cifrados hacia el WS (listado 4.1) y del WS a la base de datos, donde la función almacenada correspondiente verifica que la original al usuario antes de aplicar la renovación de la contraseña.

Listado 4. 1 Código donde se aplica el cifrado a la contraseña

1	String pwdcifrado=c.cifrar(password);
2	String newpwdcifrado=c.cifrar(nuevopassword);

Mientras se está realizando el proceso de cambio de contraseña, la aplicación Web le avisa al usuario si su nueva contraseña es fuerte o es débil en cuanto a la utilización de diferentes caracteres, esto le ayuda a decidir una mejor combinación y que sea más difícil de adivinar (figura 4.15).

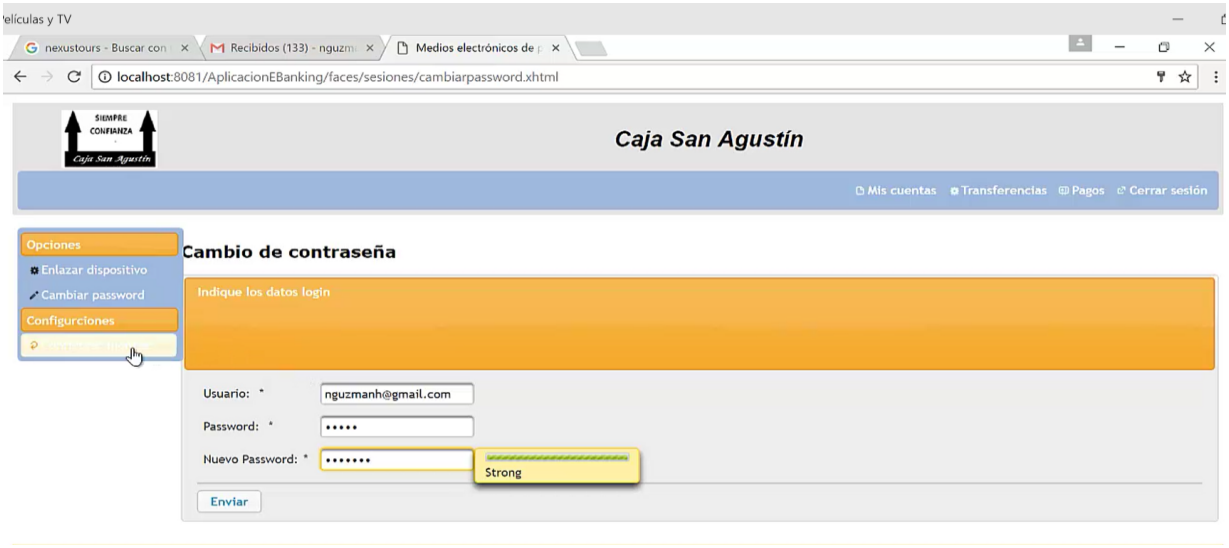


Figura 4. 15 Proceso de cambio de contraseña

Configuración de montos de retiro

En este módulo, la aplicación Web de *eBanking* permite al usuario decir la cantidad máxima de dinero que permitirá retirar diariamente y por un periodo. Para llevar a cabo este proceso, se presenta un listado de las cuentas de ahorro del usuario, esto se hace invocando al WS que arroja el listado de cuentas del usuario actual. A su vez el WS consulta a la función almacenada que recibe el id del usuario y busca entonces sus cuentas activas. Ver la figura 4.16, donde el usuario selecciona la cuenta que desea configurar.

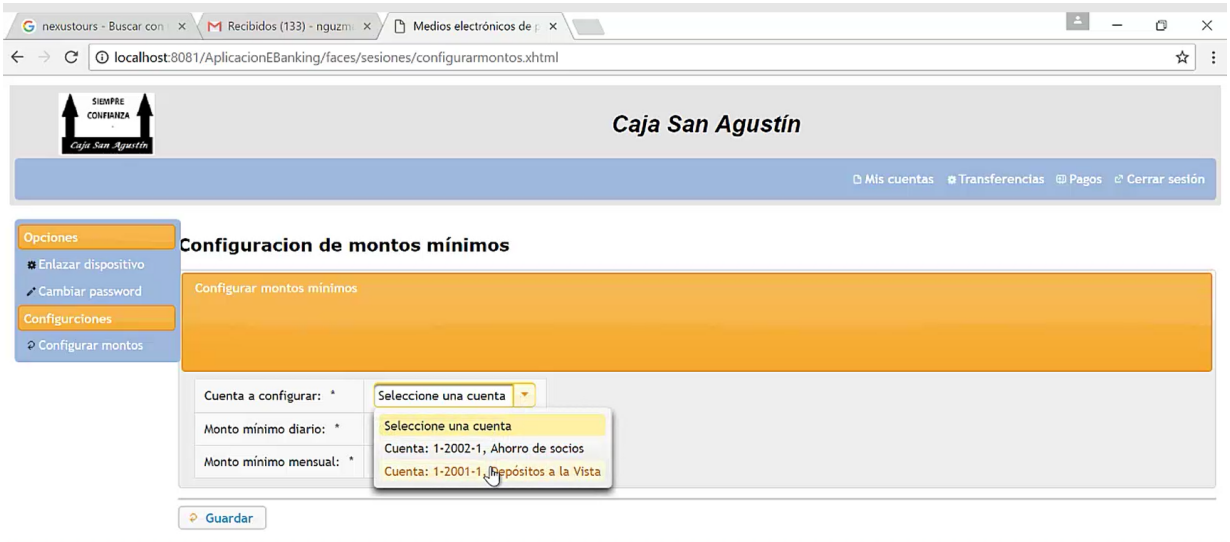


Figura 4.16 Selección de la cuenta a configurar

Cuando los campos se llenan correctamente, entonces el usuario solicita guardar los montos ingresados. Esto lo que hace es invocar al WS encargado de guardar los montos establecidos, enviándolos a la función almacenada. Se incluyen validaciones del lado del cliente para asegurar de que el usuario ingrese montos y no otros caracteres que no representen una cantidad. En la figura 4.17 se muestra la inserción de montos de retiro para una cuenta en específico.

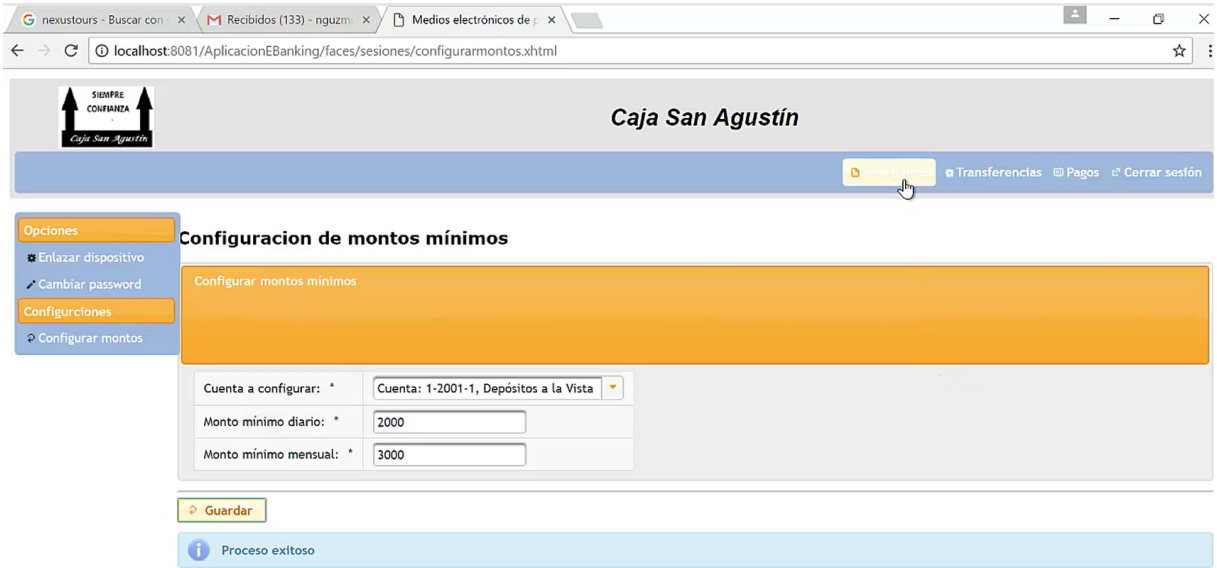


Figura 4. 17 Pantalla de especificación de montos

Aspectos de seguridad

Como resultado de la implementación de los mecanismos de seguridad ya mencionados en el apartado 3.4 “Seguridad implementada en la aplicación Web de *eBanking*” en el capítulo tres, la aplicación funciona adecuadamente en las siguientes circunstancias:

Para desplegar la aplicación en un canal HTTP seguro se habilitaron SSL (*Secure Socket Layer*, Capa de puertos seguros) y TLS (*Transport Layer Security*, Seguridad de la Capa de transporte) para que la aplicación de *eBanking* se despliegue en un protocolo de aplicación HTTPS que sirve para crear un canal cifrado. Debido a que el certificado utilizado es un certificado auto firmado, aparece una advertencia en la barra de direcciones pero la aplicación Web de *eBanking* funciona adecuadamente. En este punto cabe mencionar que una vez liberada la aplicación en un ambiente de producción real, la empresa adquirirá un certificado digital emitido por una entidad certificadora, el cual se reemplazará por el certificado auto firmado, de esta manera el visualizador dejará de mostrar la advertencia debido a que un certificado real está respaldado por su entidad certificadora como una autoridad que representa en la autenticidad de dichos certificados. En la figura 4.18 se muestra la pantalla de la aplicación funcionando con el certificado auto firmado.

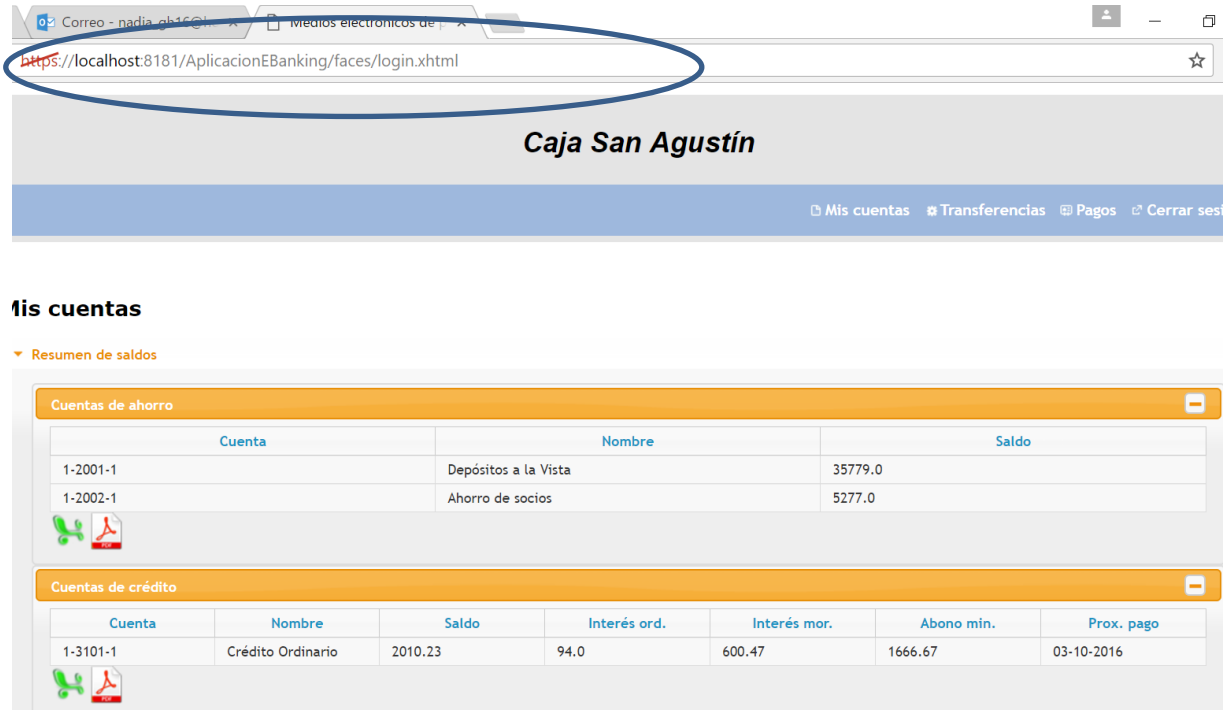
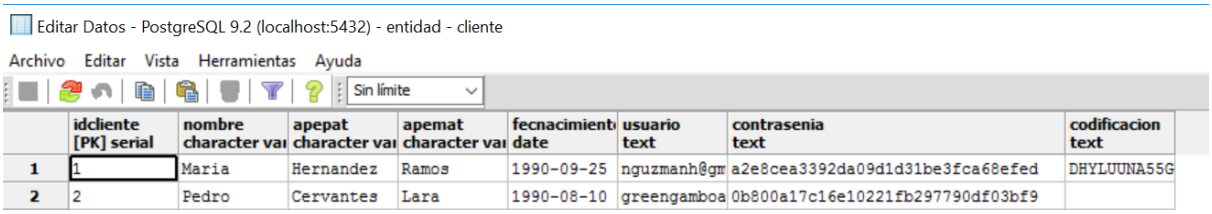


Figura 4. 18 Https habilitado para la aplicación de eBanking

Cifrado de contraseña

Como medida de prevención ante una intercepción de la información mientras está viaja por la red hacia su destino, se implementó el cifrado de datos sensibles. Si el usuario ingresa una contraseña, se le aplica un algoritmo de cifrado para que no sea legible en caso de ser interceptado. La aplicación de *eBanking* manda la contraseña cifrada al WS y el WS también lo manda cifrada hasta la base de datos y de esa manera se almacena. Aunque no se tiene acceso directo hacia la base de datos original, se creó una base de datos de prueba para verificar el almacenamiento y la recuperación de la contraseña, de una manera transparente para el usuario, ya que él introduce la contraseña normal siempre. En la figura 4.19 se muestra el ejemplo de cómo se ve una contraseña cifrada almacenada en la base de datos de prueba



	idcliente [PK] serial	nombre character var	apepat character var	apemat character var	fecnacimient date	usuario text	contrasenia text	codificacion text
1	1	Maria	Hernandez	Ramos	1990-09-25	nguzmanh@gm	a2e8cea3392da09d1d31be3fca68efed	DHYLUUNA55G
2	2	Pedro	Cervantes	Lara	1990-08-10	greengamboa	0b800a17c16e10221fb297790df03bf9	

Figura 4. 19 Contraseña almacenada en la base de datos

Para asegurar la integridad de los datos enviados, sobre todo en el caso de las operaciones de transferencias y pagos, se implementó la firma digital, para firmar los mensajes SOAP y de esta manera asegurar que los datos que el usuario envía no sean cambiados por otros datos, ya que, si fueran cambiados dichos datos, la firma recibida en el WS y la firma generada en el WS no podrían coincidir. Para efectos de prueba, se realizó una impresión de pantalla, de la firma generada por el WS y la firma recibida por parte de la aplicación y se mandó a imprimir el mensaje “Firma coincide” si es que coinciden. En la figura 4.20 se aprecia la impresión de una comparación de firmas digitales

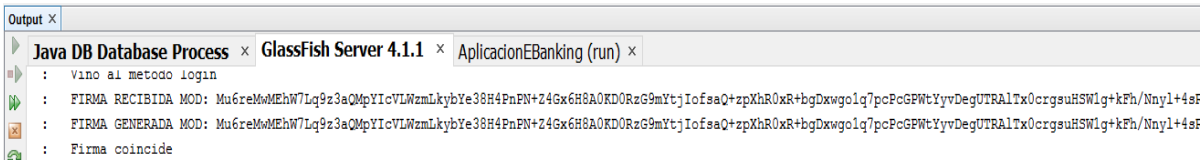


Figura 4. 20 Impresión de la firma digital y el resultado de la comparación de la firma recibida y la generada

Al hacer la prueba de invocación al WS, si se verifica que las firmas coinciden entonces entrega el resultado correcto. En la figura 4.21 se aprecia la firma que recibe el WS y los resultados que entrega después de verificar que es correcta.

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:wmlogin xmlns:ns2="http://wsprovider/">
            <usuario>usuario1</usuario>
            <password>usuario1</password>
            <firma>L9dlOh8gdYSeWm3P5+XJDzObymx+rct7f7QqnjXDyEAhn72SCOB/9119PZ1LhdmnP6q1DvUTJklj5AyueDRQK+pdYbH1/DOABfmt+ZMYuG6kPEwxDeaL9cqAHs6rZ8tw7GuyzkmRAP2Sesh
            <session>123456</session>
        </ns2:wmlogin>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:wmloginResponse xmlns:ns2="http://wsprovider/">
            <return>
                <encontrado>true</encontrado>
                <kasociado>1</kasociado>
                <nombre>Maria</nombre>
            </return>
        </ns2:wmloginResponse>
    </S:Body>
</S:Envelope>
```

Figura 4. 21 Invocación al WS para probar los resultados cuando las firmas son coincidentes

Ahora bien, pensando en que los datos sensibles del usuario pudieran ser conocidos por otra persona, la aplicación de *eBanking* siempre solicitará un código de autorización después de cada petición de transacción, como ya se ha descrito en los párrafos anteriores, al realizar transferencias o pagos. El código solicitado siempre tendrá que ingresarse segundos antes de completar la transacción, siempre será generado solamente por el dispositivo móvil del usuario titular de la cuenta y será dinámico, es decir, siempre será diferente, cambiará cada treinta segundos para asegurar que solamente el dueño de la cuenta pueda generarlo e introducirlo en el momento indicado. De esta manera si la contraseña de la cuenta fue descubierta, aun así, no es posible efectuar una transacción si no se tiene el celular que genera los códigos o *tokens* válidos y este dispositivo o celular solo pertenece al dueño de la cuenta.

Existen también usuarios externos o bien usuarios auténticos que pretendan saltarse la autenticación debido a que no cuenten con un usuario y contraseña válidos o bien usuarios que sí cuenten con ello, pero quieran evitarse la autenticación, y posiblemente simplemente vayan directamente a la barra del visualizador y tecleen la ruta completa de la página de transferencias, tratando de hacer una transferencia a su favor. Para esto la aplicación que cuenta con el filtro de *FacesServlet*, redirecciona al usuario automáticamente hacia la página de inicio de sesión,

impidiendo que entre a las páginas de transferencias y/o pagos. El usuario que no se ha identificado en la aplicación, es redireccionado automáticamente a la página de ingreso (*login*) como se observa como en la figura 4.20

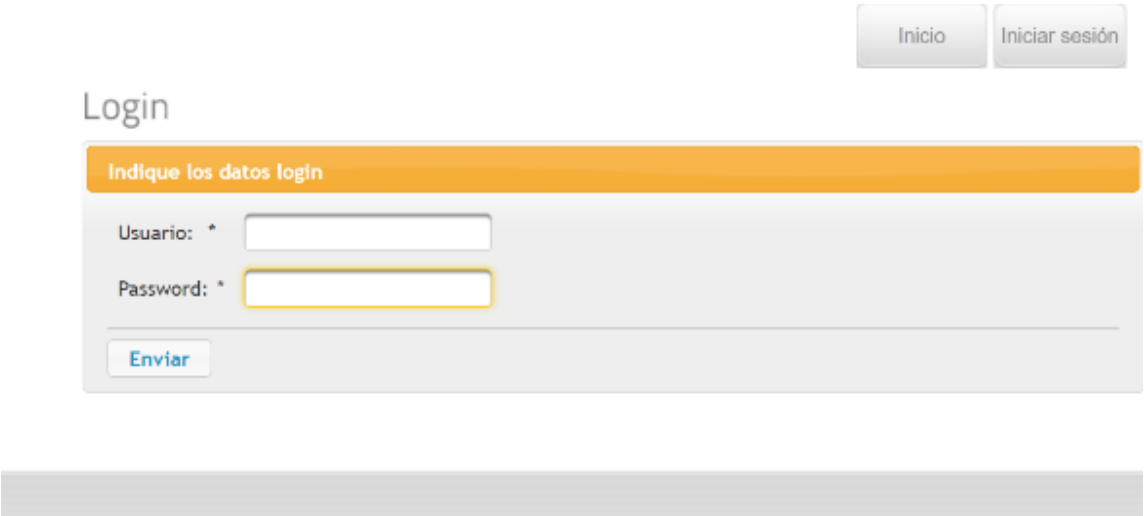


Figura 4. 22 Página a la que se redirecciona un usuario no firmado

Finalmente, cómo resultado también la aplicación Web de *eBaking*, mantiene al usuario propietario de la cuenta informado en todo momento de cada transacción que se realiza sobre cada cuenta que le pertenezca, esto con el fin de que tenga el control de lo que pasa con su cuenta y siempre tenga la información a la mano, pero también pueda actuar a tiempo en caso de que él no haya solicitado cualquier transacción y solicitar el bloqueo de su cuenta o el cambio de su contraseña o tomar alguna medida en caso de que se haya presentado algún movimiento no reconocido por el dueño de la cuenta. En la figura siguiente se presenta un ejemplo de la notificación recibida por correo acerca de una transacción hecha con su cuenta, detallando la cuenta origen, la cuenta destino y el monto de la transacción.

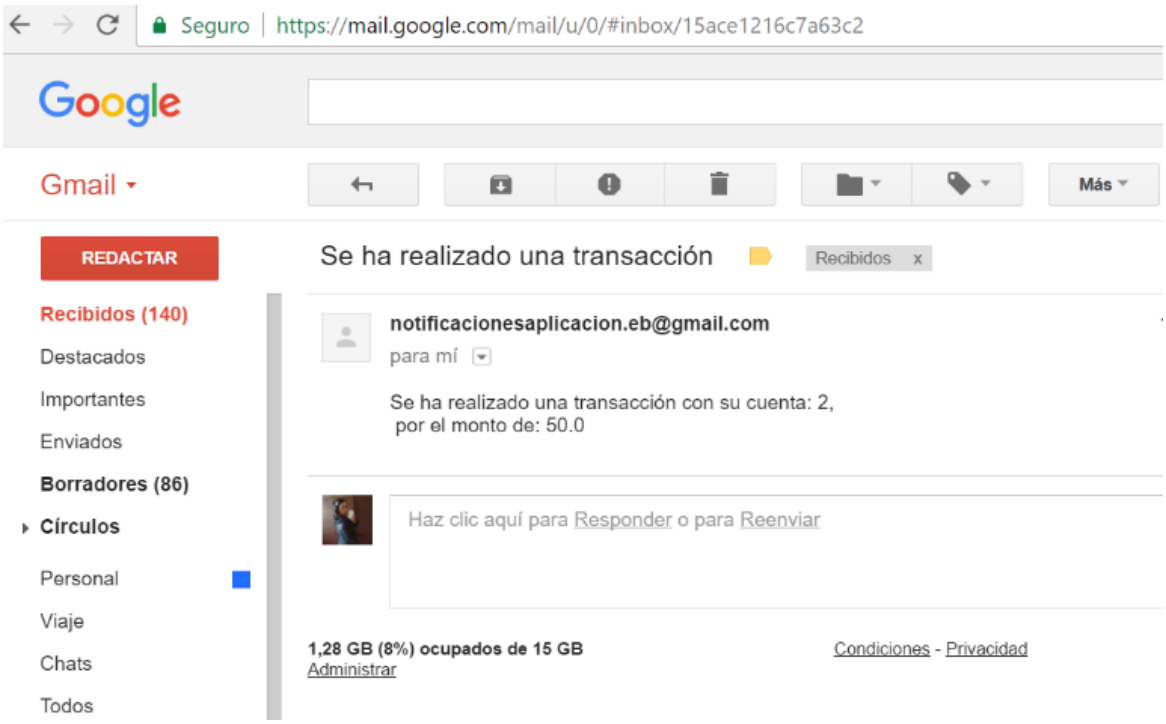


Figura 4.21 Notificación de transacción con una cuenta del usuario

Capítulo 5 Conclusiones y recomendaciones

Es importante destacar que en el sector de las entidades financieras no bancarias al igual que en otros sectores, la aplicación de las tecnologías de información juega un papel muy importante ya que ayuda a satisfacer los requerimientos de los usuarios de las mismas, así como también satisface los requerimientos de las mismas empresas y de las personas que laboran en dichas empresas. En este caso el desarrollo de la aplicación Web de *eBanking* significó un reto y a su vez un logro para el sector.

Considerando su desarrollo y construcción, la utilización JSF, como un lenguaje que es el estándar en desarrollo Web, fue posible construir una solución basada en Web compuesta de dos aplicaciones: la aplicación proveedora de los WS y la aplicación de *eBanking* que es la que consume dichos servicios. Gracias a que forman parte de un lenguaje estándar, permitió la interoperabilidad con el sistema de gestión principal de la empresa SINC y así poder comunicar dos plataformas diferentes hasta lograr la operación con el usuario final. Esto a su vez hizo posible acercar servicios financieros básicos a los usuarios finales de las EFNB de manera transparente, tal y como si el usuario estuviera utilizando el sistema original.

Ahora bien, como se mencionó a lo largo del presente documento, la naturaleza delicada de los datos que se manipulan dentro de la aplicación Web requiere la protección adecuada en contra de diversas amenazas, por lo tanto, se trabajó en mecanismos de seguridad que después de efectuar las diferentes pruebas, resultaron ser efectivos y arrojar los resultados esperados para contrarrestar las amenazas más importantes que sufren las aplicaciones Web.

Debido a que se construyó una aplicación orientada a servicios, aparte de la interoperabilidad con el sistema original, se creó un software que fuera distribuido como servicio, lo cual quiere decir que se habilita o deshabilita para el uso de diversas EFNB, siempre y cuando se cuente con la infraestructura para habilitar y

administrar diferentes subdominios que permitan a usuarios de distintas entidades, acceder solamente a la entidad a la cual pertenecen.

Uno de los principales objetivos de la banca social es la inclusión de la población a servicios financieros sin importar la cantidad de ahorros que puedan tener. Cuando se trata de soluciones basadas en Web, los servicios financieros se acercan aún más a la población, debido que es más fácil acceder a un servicio desde su propia casa, trabajo, escuela o cualquier parte donde se encuentre el usuario.

En conclusión, se satisficieron todos los requerimientos planteados desde los diferentes puntos de vista al utilizar un sistema de *eBanking* como un medio para facilitar las operaciones y fungir como una herramienta poderosa dentro de las empresas de banca social, a través de un equipo de cómputo o de un dispositivo móvil y una conexión a Internet.

Recomendaciones

Es necesario probar la aplicación en un ambiente real, darle un uso y detectar posibles errores que se presentan ya en un ambiente donde existen diversos usuarios y múltiples transacciones, escenarios que solamente ocurren en producción, tanto en procesos operativos como en condiciones de seguridad informática y en la red. Sin embargo, estas pruebas no se realizaron debido a que dependen de decisiones propias de la empresa.

Por otro lado, como un trabajo a futuro, sería conveniente robustecer el módulo de administración para que sea posible manejar niveles de usuarios administradores y permisos de los mismos de tal manera que exista un súper usuario y diversos usuarios que puedan dar de alta y baja o modificación de EFNB.

Bibliografía

[1] H. E. Martínez, “S Ituaci{ó}n Actual Del S Istema De a Horro,” 2008.

[2] G. G. García, “El ahorro popular en México. Perspectiva,” *CONDUCEF*, 2015.

[3] Corporativo, “Misión y Visión,” 2014. [Online]. Available: http://www.cnbv.gob.mx/CNBV/Paginas/Misi%C3%B3n-y-Visi%C3%B3n.aspx.

[4] Deloitte, “Banking disrupted: How technology is threatening the traditional European retail banking model,” 2015.

[5] Z. B. Omariba and N. B. Masese, “Security and Privacy of Electronic Banking,” *Int. J. Comput. Sci. Issues*, vol. 3, no. 3, p. 262, 2013.

[6] M. Shema, *Haking Web Apps*. Waltham MA 02451: Syngress, 2012.

[7] Z. B. Omariba and N. B. Masese, “Securit Y and Privacy of Electronic Banking,” *Int. J. Comput. Sci. Issues*, vol. 3, no. 3, p. 262, 2013.

[8] Microsoft, “Interoperabilidad. Apertura tecnológica para el progreso,” 2016. [Online]. Available: https://www.microsoft.com/spain/responsabilidad_corporativa/prioridades/interoperabilidad/. [Accessed: 14-May-2016].

[9] Javier Alex Torres Rojas, “La Tecnología Java Server Faces,” Perú, 2005.

[10] “Why PrimeFaces.” [Online]. Available: http://www.primefaces.org/whyprimefaces.

[11] D. k. Barry, *Web Services, Service-Oriented Architectures and Cloud Computing*. Waltham Estados Unidos, 2013.

[12] “About WS-I,” 2009. [Online]. Available: http://www.ws-i.org/about/Default.aspx.

[13] “¿Qué es SOAP?” [Online]. Available: https://www.ibm.com/support/knowledgecenter/es/SSKM8N_8.0.0/com.ibm.etools.mft.doc/ac55770_.htm.

[14] “WS-Security.” .

[15] “¿Qué es eBanking?,” 2088. [Online]. Available:

http://www.ebankingnews.com/columnas/%C2%BFque-es-eBanking-006.

[16] T. Erl, *Cloud Computing. Concepts, technology and architecture*. Prentice Hall.

[17] E. Juarez, “La banca digital inicia su despegue en México,” CD. Mexico, p. 4, 01-Jun-2015.

[18] M. Fedrizzi, A. Molinari, and V. Ventre, “a Model for Evaluating the Transaction Risk in EBanking,” *ResearchGate*, no. January 2016, pp. 172–178, 2015.

[19] S. M. Darwish and A. M. Hassan, “A model to authenticate requests for online banking transactions,” *Alexandria Eng. J.*, vol. 51, no. 3, pp. 185–191, 2012.

[20] M. Hosburgh, “InfoSec Reading Room,” *SANS Inst.*, p. 16, 2014.

[21] V. Khu-smith and C. J. Mitchell, “Using GSM to Enhance E-Commerce Security,” pp. 75–81, 2002.

[22] Kamesh and N. Sakthi Priya, “Security enhancement of authenticated RFID generation,” *Int. J. Appl. Eng. Res.*, vol. 9, no. 22, pp. 5968–5974, 2014.

[23] M. I. P. Salas and E. Martins, “Security testing methodology for vulnerabilities detection of XSS in web services and WS-security,” *Electron. Notes Theor. Comput. Sci.*, vol. 302, pp. 133–154, 2014.

[24] D. Zissis and D. Lekkas, “Addressing cloud computing security issues,” *Futur. Gener. Comput. Syst.*, vol. 28, no. 3, pp. 583–592, 2012.

[25] S. G and M. S, “Securing Software as a Service Model of Cloud Computing: Issues and Solutions,” *Int. J. Cloud Comput. Serv. Archit.*, vol. 3, no. 4, pp. 1–11, 2013.