

Universidad de las Ciencias Informáticas

Facultad 4



**Título: Concepción y desarrollo del
Módulo Gestión de Autenticación de los
Clientes del Proyecto Telebanca.**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor(es): Meybis Cruz Rodríguez
Anny García Sardiñas
Alejandro Ogando Septien

Tutor: Lic. David Batard Lorenzo

Junio 2007

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Meybis Cruz Rodríguez

Alejandro Ogando Septien

Firma del autor

Firma del autor

Anny García Sardiñas

David Batard Lorenzo

Firma del autor

Firma del tutor

DATOS DE CONTACTO DEL TUTOR

Lic. David Batard Lorenzo, Profesor Instructor de los Dptos. de IA e IGS.

Graduado de Ciencias de la Computación, en la UCLV y UH.

Actual Asesor de la Dirección de Investigaciones.

AGRADECIMIENTOS

Al Comandante en Jefe por haber tenido la genial idea de construir esta universidad del futuro, a la Facultad y sus profesores que tan pacientemente han contribuido a nuestra definitiva formación, al tutor de nuestro proyecto que tan instruidamente nos ha asesorado, a nuestros padres, hermanos y amigos que han apoyado de forma incondicional esta dedicación al estudio.

DEDICATORIA

Dedico este trabajo a mi madre, por haberme dado la vida, porque la mayor parte de lo que soy es por ella, por todo lo que ha pasado para darme lo mejor, por su ejemplo de guerrillera y por todo el amor que me ha dado a lo largo de toda mi vida.

A mi padre Ernesto por haber exigido de mí, por sus enseñanzas, por haber estado allí, a mi lado, a lo largo de estos 15 años.

A Claudy, la niña linda de mi vida que ya empieza, las cosas no son tan difíciles como parecen.

A mi abuela Paula, la viejita de mi alma, por ser su consentida, y por todo cuanto la quiero.

A mima mi superabuela, de su nieta ingeniera.

A mis tíos Luis y Rodolfito.

A Yassier, que es mi amor y mi vida.

A toda mi familia, de corazón.

Meybis

Dedico este trabajo principalmente a mi Madre y Padre, que me han dado todo el apoyo incondicional necesario para, estando tan lejos de casa, cumplir con mi vida universitaria y que hayan sido guía y luz inspiradora en mi vida. A mis tíos y tías que han confiado en mí y me han dado consejos en momentos duros. A mis hermanos TITO y LEO por ser de gran ayuda y estar siempre cuando los necesito. Quiero hacer una dedicatoria muy importante a mi hermano del alma ARIEL que lo quiero con mi vida, para que se le sirva de ejemplo en su venidera carrera.

A todos los quiero y siempre los tengo en el corazón y el pensamiento.

Alejandro

A mis padres: Mami y Papi, sin ustedes no hubiera podido llegar hasta aquí, su amor y ayuda en todo momento han sido fundamentales durante este largo trayecto, son mi guía y ejemplo, ustedes se merecen esto y mucho más.

A mi Tata del alma: Fanny has sido en todo momento hermana y amiga, acompañante y consejera. Quien me entiende más que nadie. Y porque este, también es tu sueño. Te quiero mucho.

A José Luis: Por ser más que mi propia vida, por ser la alegría y preocupación de todos en nuestra familia. Espero que algún día tu triunfo también sea mío.

A Nelson: Mi amor has estado presente desde el comienzo de mi carrera, me has dado tu apoyo y cariño en todo momento, también es para ti.

A toda mi familia por depositar en mí toda la confianza del mundo.

Anny

RESUMEN

Debido al amplio proceso de informatización de la sociedad cubana que se lleva a cabo en los últimos años y como respuesta a una necesidad de la población en general respecto al pago de los servicios se hace necesario la creación de un sistema para agilizar este proceso, con este fin surge Telebanca. Este sistema no cuenta con un módulo que automatice los procesos de autenticación de los clientes asociados y por ello no puede recibir los datos de los clientes, su entrada principal.

En el presente trabajo se presenta una propuesta del análisis, diseño, implementación y prueba del módulo Gestión de Autenticación de los Clientes del sistema TeleBanca. Con la implementación y desarrollo de este módulo las personas que se conviertan en clientes de TeleBanca podrán, entre otras prestaciones, contar con los documentos de autenticación requeridos, lo que les permitirá identificarse correctamente del resto de los clientes y tener la seguridad de que solo ellos podrán realizar operaciones en sus cuentas y por tanto se estará un paso más cerca de lograr elevar la eficiencia en el pago de los servicios, de interés tanto para la población como para las empresas y demás instituciones implicadas.

PALABRAS CLAVE: autenticación, clientes.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción.....	5
1.2 Estado del Arte.....	5
1.2.1 Uso de tarjetas inteligentes en los bancos a nivel mundial.....	6
1.2.2 Modernización y automatización del sistema bancario cubano	7
1.3 La Criptografía.....	8
1.3.1 Rijndael (el nuevo estándar AES)	9
1.3.2 Seguridad del algoritmo	10
1.3.2 Criptografía simplificada con Visual Estudio .NET.....	11
1.4 Tecnologías y Herramientas propuestas para el desarrollo del sistema	12
1.4.1 La tecnología .NET	12
1.4.2 Servicios Web (Web Services).....	15
1.5 Fundamentación de la metodología propuesta para el desarrollo del sistema	16
1.5.1 Proceso Unificado de Rational	16
1.5.2 Lenguaje de modelación	18
1.5.3 Herramientas de Modelado Visual y Desarrollo.....	18
1.6 Conclusiones.....	19
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	20
2.1 Introducción.....	20
2.2 Situación problemática	20
2.3 Objeto de automatización	20
2.4 La información que se maneja.....	21

2.5 Propuesta del Sistema.....	22
2.6 Modelo del negocio	25
2.7 Modelo del Dominio.....	25
2.7.1 Conceptos principales.....	26
2.7.2 Eventos principales.....	26
2.7.3 Diagrama de clases del dominio	27
2.8 Especificación de Requerimientos del Sistema	28
2.8.1 Requerimientos funcionales	28
2.8.2 Requerimientos no funcionales	29
2.9 Definición de los Casos de uso del Sistema	32
2.9.1 Descripción de los actores	32
2.9.2 Casos de uso del Sistema.....	33
2.9.3 Diagrama de casos de uso.....	36
2.9.4 Casos de uso expandidos.....	37
2.10 Conclusiones.....	61
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	63
3.1 Introducción.....	63
3.2 Análisis.....	63
3.2.1 Diagrama de clases del análisis	63
3.3 Diseño	65
3.3.1 Diagramas de secuencia.....	65
3.3.2 Diagrama de clases Web del Sistema.....	69
3.3.3 Descripción de las clases.....	71
3.3.4 Diagrama de clases persistentes	92

3.4 Conclusiones.....	94
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	95
4.1 Introducción.....	95
4.2 Implementación	95
4.2.1 Modelo de despliegue	95
4.2.2 Diagrama de componentes	96
4.2.3 Descripción de los componentes.....	99
4.3 Modelo de Prueba	101
4.3.1 Descripción de los casos de prueba de integración.....	101
4.4 Conclusiones.....	108
CONCLUSIONES	109
RECOMENDACIONES	110
REFERENCIAS BIBLIOGRÁFICAS.....	111
BIBLIOGRAFÍA.....	112
GLOSARIO DE TERMINOS.....	115

INTRODUCCIÓN

Durante las dos últimas décadas, el desarrollo tecnológico muestra una correlación cada vez mayor entre la Informática, las Telecomunicaciones, la Electrónica y la Automatización, proceso que ha devenido una nueva rama del saber denominada Tecnologías de la Información, de alta incidencia en la modernización y eficiencia de todos los sectores de la sociedad.

La Informatización de la Sociedad es el proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones en la vida cotidiana, para satisfacer las necesidades de todas las esferas de la sociedad, en su esfuerzo por lograr cada vez más eficacia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos. Una sociedad que aplique la informatización en todas sus esferas y procesos será más eficaz, eficiente y competitiva.

Tras el triunfo revolucionario, Cuba se propuso un camino de desarrollo que pudiera satisfacer por igual las necesidades espirituales y materiales básicas de su población, sobre la base de una distribución más justa y equitativa de la riqueza y de esa forma, salir de la crisis social y económica en que la isla se encontraba sumida. Hoy Cuba tiene un nuevo reto: la necesidad de dominar e introducir en la práctica social las Tecnologías de la Información y las Comunicaciones; y así convertirse en la Sociedad de la Información y lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitaría a nuestra sociedad acercarse más hacia el objetivo de un desarrollo sostenible.

La Sociedad de la Información debe ser para todos, es imprescindible su orientación hacia el desarrollo justo, equitativo, sostenible y alcanzable. Cuba ha trazado el Programa Nacional de Informatización de la Sociedad Cubana, que persigue catalizar el uso masivo de las Tecnologías de la Información a escala nacional, en las diferentes esferas y sectores buscando la optimización de recursos y el logro de mayor productividad en dichas esferas. Como expresión del proceso revolucionario cubano, se tiene al ciudadano en el centro de sus objetivos, buscando siempre cultivar su calidad de vida en su desempeño familiar, laboral, educacional, cultural, social y político.

La Universidad de Ciencias Informáticas, legítima expresión de los programas de la Batalla de Ideas que hoy se desarrollan en nuestro país y eficaz medio para lograr la informatización de la sociedad, hace su aporte a la industria cubana de software con la creación del proyecto TeleBanca, ejemplo de la puesta en práctica y desarrollo de estas políticas antes mencionadas.

En nuestro país actualmente la población paga ciertos servicios como por ejemplo: el servicio eléctrico, el servicio telefónico, las multas del tránsito, el consumo del agua, etc. Todos estos servicios hoy se cobran por separado en las oficinas de cobro y sucursales bancarias correspondientes y cuando el personal encargado asiste a las viviendas para proceder con el cobro de los mismos.

Esta dispersión y diversidad en el pago de los servicios causa una serie de inconvenientes para la población y para empresas e instituciones que brindan estos servicios pues para la población el pago de servicios es engorroso y dilatado, ya que debe trasladarse a múltiples lugares para hacer los diversos pagos y una vez allí muchas veces se encuentra con un gran número de personas esperando ser atendidas, tomándole esto de gran cantidad de tiempo en horario laboral, mientras que para las empresas el cobro de los servicios es una gestión que demanda de gran cantidad de personal en oficina para el trabajo en las áreas comerciales y además gran cantidad de personal circulando por la ciudad con sumas considerables de dinero. Además tienen que contratar muchas personas para llegar a cada uno de los hogares y para responder a las quejas y reclamaciones que se presenten. Con el desarrollo y puesta en práctica del proyecto Telebanca el cobro de estos servicios se realizaría de forma única y centralizada, desapareciendo los inconvenientes antes mencionados.

Para que TeleBanca ofrezca sus servicios a la población es necesaria la materialización del módulo Gestión de Autenticación de los Clientes pues sin este no es posible obtener los datos de las personas que han asistido a los bancos para suscribirse a los servicios de TeleBanca, por lo que el sistema no podría funcionar correctamente al no tener forma de obtener su información más valiosa, los datos de sus clientes. Además como consecuencia de lo anteriormente planteado se hace imposible crear la documentación necesaria para identificar a los clientes asociados. Sin este módulo Telebanca no podría operar ya que jamás se podría obtener los datos de los clientes asociados, información que consiste la entrada esencial de este sistema.

Este trabajo surge como necesidad de dar solución a las situaciones antes expuestas; por lo que el **problema** a solucionar en él es el siguiente:

¿Cómo desarrollar un módulo para el software Telebanca que automatice los procesos de captación, impresión y actualización de los datos de autenticación de los clientes asociados?

Por tanto el **objeto de estudio** de este trabajo son los procesos de gestión asociados a la creación de la autenticación de los clientes de TeleBanca.

De ello se deriva que el **campo de acción** que abarca este trabajo, son los procesos bancarios de impresión de tarjetas, impresión de reportes y conciliaciones bancarias.

El **objetivo general** de este trabajo será: desarrollar un módulo que automatice los procesos de gestión de autenticación de los clientes asociados a TeleBanca en un espacio de tiempo fiable, con alta calidad y correctamente documentado.

De acuerdo con esta propuesta se derivan los siguientes **objetivos específicos**:

- Realizar el Análisis y Diseño de los requerimientos, enmarcados dentro de los casos de uso.
- Realizar la implementación de todos los procesos que se describen utilizando la tecnología que se decida sea la adecuada.
- Asegurar la información confidencial almacenada en la base de datos mediante el uso de un mecanismo de encriptación.
- Desarrollar un sistema que satisfaga las necesidades e intereses de los clientes y usuarios finales.

Para cumplir con estos objetivos y resolver la situación problemática planteada, se proponen las siguientes **tareas**:

- Realizar entrevistas con los clientes e involucrados en el negocio para obtener información útil que sirva como punto de partida para la modelación del sistema.
- Estudio de las formas de autenticación de los clientes en el Banco Central de Cuba y en el mundo.
- Selección de la Metodología de Análisis y Diseño de sistemas informáticos que facilite la creación del sistema y de igual modo garantice la calidad del mismo. muy igual
- Selección de la metodología de Análisis y Diseño de sistemas informáticos que facilite la creación y garantice la calidad del sistema.
- Análisis y selección de tecnologías y herramientas que internacionalmente se utilizan para llevar a cabo sistemas como el que se pretende desarrollar.
- Realizar un estudio de los mecanismos de encriptación de la información que existen actualmente y proponer uno para usarlo en el desarrollo de la aplicación.
- Diseñar los casos de uso de prueba para comprobar que el software cumpla con todos los requisitos necesarios.

Tras la realización de este trabajo se esperan los siguientes **aportes prácticos**.

- Las personas que se conviertan en clientes de TeleBanca podrán obtener su tarjeta y PIN, lo que les permitirá identificarse correctamente del resto de los clientes y tener la seguridad de que solo ellos podrán realizar operaciones en sus cuentas.
- En caso de que un cliente pierda alguno de sus documentos de autenticación o necesite cambiarlos por cualquier otro motivo se le podrá crear una nueva documentación.
- Se le podrá dar de baja a un cliente luego de que este haya efectuado esta solicitud realizando una llamada telefónica o haya asistido a su sucursal bancaria correspondiente.
- Los bancos que se integren al servicio podrán tener información actualizada referente a sus clientes.
- Se podrá conocer cuantas tarjetas y pines se han impreso, así como cuanto material se necesitará para las próximas impresiones.
- Los estudiantes que se involucren en la elaboración del software contarán con la experiencia de haber participado en un proyecto de ciclo completo.

Este documento consta de cuatro capítulos en los cuales se describen paso a paso todo el proceso o etapas por la que transitó este trabajo. En el Capítulo 1 se investigan las tendencias actuales que existen en el mundo y en Cuba para autenticar a los clientes en los bancos. Se realiza un estudio de la Criptografía y se fundamenta la elección del algoritmo de cifrado que se usará para desarrollar la aplicación, así como una propuesta de las tecnologías, metodologías y herramientas de desarrollo de software. En el Capítulo 2 se modela el dominio del negocio y se realiza una propuesta de sistema, el Análisis y Diseño de los casos de uso se lleva a cabo en el Capítulo 3 y finalmente en el Capítulo 4 se desarrollan los principales diagramas de flujo de trabajo Implementación así como las pruebas de caja negra realizadas a la interfaz de la aplicación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se ofrece una breve panorámica de las tendencias actuales de introducción de nuevos mecanismos de autenticación en los procesos bancarios a nivel mundial y de la situación actual del sistema bancario cubano. Se fundamenta la elección del algoritmo de cifrado seleccionado para lograr la seguridad de los datos y se ofrece además una breve descripción de la metodología de desarrollo de software y de las herramientas y lenguajes de desarrollo de aplicaciones web escogidos para la confección del sistema.

1.2 Estado del Arte

Desde hace varios años, los bancos empezaron a ofrecer a sus clientes la posibilidad de realizar transacciones a través de los medios electrónicos y vienen fomentando la migración de sus clientes hacia estos medios debido a que son más baratos que la sucursal, mejoran la eficiencia de los procesos bancarios y con ellos los clientes logran una mejor disponibilidad de su horario.

Para realizar estas transacciones de una forma segura y confiable es necesario autenticar correctamente a los usuarios, o sea tener la seguridad de que un cliente es quien dice ser. A nivel mundial la autenticación se realiza mediante tarjetas magnéticas, tarjetas inteligentes, tokens, entre otras. Algunos bancos adicionan a las alternativas antes mencionadas técnicas de biométrica de reconocimiento de voz, huellas dactilares, etc.

- **Tarjetas magnéticas:** Son piezas de plástico con una banda magnética. La banda magnética contienen un criptograma con la información para autenticar al cliente, de esta forma cuando el cliente introduce la tarjeta y su número de identificación personal (PIN) en un cajero automático(ATM), o en una terminal de punto de venta(POS), la información que contiene la banda magnética se envía a un sistema que la descifra y verifica si a esta le corresponde el PIN especificado por el cliente, en caso de corresponder el cliente tiene acceso a su cuenta y de lo contrario se le muestra un mensaje en el cajero de clave incorrecta.

- Tarjetas inteligentes: Son tarjetas que poseen un microprocesador incrustado en la parte frontal. Este circuito integrado almacena una variada cantidad de información con sofisticados métodos de seguridad. Este integrado tiene la capacidad de comunicarse con el mundo exterior de diferentes formas, esta comunicación le permite intercambiar información con dispositivos como puntos de venta, cajeros automáticos o lectores de tarjetas inteligentes conectados a teléfonos, computadoras, máquinas dispensadoras y otros. La finalidad de la comunicación varía dependiendo de la aplicación para la que se este usando la tarjeta inteligente. Existen tres tipos de tarjetas inteligentes dependiendo de su forma de comunicación:
 - Tarjetas de Contacto: Estas tarjetas requieren ser insertadas en un lector de tarjetas inteligentes. Funcionan mediante el acoplamiento inductivo o capacitivo entre la tarjeta y el lector a menos de un milímetro de distancia, son las más comunes en aplicaciones financieras.
 - Tarjetas sin contacto o de acoplamiento remoto: Estas tarjetas requieren solamente estar próximas a la antena del lector. Tienen incorporada una placa capacitiva o una antena enrollada internamente que permite el acoplamiento con el lector mediante señales de radiofrecuencia, las que también alimentan la tarjeta en un arreglo de dispositivo de bajo consumo.
 - Tarjeta CombiCard: Esta tarjeta engloba las dos formas de comunicación anteriores

1.2.1 Uso de tarjetas inteligentes en los bancos a nivel mundial

En 1980 las empresas Bull CP8 y Motorola integran el microcontrolador y la memoria de las tarjetas inteligentes que existían en ese entonces en un solo dispositivo. El sistema bancario francés uso por muchos años una tarjeta que usaba este dispositivo, esta tarjeta bancaria contenía información de la cuenta, el número de identificación personal (PIN) y un registro de transacciones financieras que podría ser usado para reconciliar actividades cuestionables.

En diciembre del 2006 las entidades bancarias de Guatemala Banco Cuscatlán, Banco Industrial y Banco Uno se integraron al proyecto Visa payWave que lanzó Visa International Región América Latina y el Caribe (Visa ALC), en conjunto con VisaNet Guatemala, como parte de sus esfuerzos para expandir el uso de tarjetas inteligentes sin contacto en la región. Hasta esta fecha se habían emitido en América Latina y

el Caribe más de 21,4 millones de tarjetas inteligentes Visa basadas en tecnología EMV, lo que representa un 20 por ciento de todas las tarjetas de crédito Visa en la región. Estos avances en tecnología e innovación permitirán a los bancos expandir su cartera de productos y servicios.

Prodem, la empresa que ofrece microcréditos y otros servicios financieros por más de 14 años a la población pobre rural y urbana de Bolivia, desde el 2001 comenzó un proyecto con cajeros automáticos multilingüe de fácil uso. Funcionando solo con una tarjeta inteligente y un scanner de huellas digitales ofrecen un nivel de seguridad sin precedente a los clientes de Prodem. Las tarjetas inteligentes guardan todos los datos personales de los clientes, números de cuentas, registro de las transacciones y la huella digital, permitiendo a los cajeros automáticos funcionar sin estar permanentemente conectados con la red, una gran ayuda para las zonas rurales. Las máquinas también ofrecen una simplicidad funcional capaz de superar las barreras del idioma y el analfabetismo ofreciendo comandos de voz en español y varios dialectos locales. Para hacer las cosas aún más fáciles, los clientes van a poder acceder a los servicios solamente al tocar la pantalla.

Los bancos Bancomer, Serfin, Banamex, Bancrecer y Banorte en México actualmente utilizan el sistema de tarjetas inteligentes para la autenticación de sus usuarios, este sistema es introducido por algunas empresas como Mastercard y Visa.

1.2.2 Modernización y automatización del sistema bancario cubano

Como parte del proceso de modernización del sistema bancario, para lograr el pago y las transacciones bancarias totalmente automatizadas, desde octubre de 1997 se comenzó con la introducción de tarjetas con bandas magnéticas y la creación de una red de cajeros automáticos, bajo la marca "RED". Esta primera red de cajeros automáticos comenzó a funcionar con cinco unidades instaladas pertenecientes al Banco Popular de Ahorro(BPA), Banco de Crédito y Comercio(BANDEC), Banco Internacional de Comercio, S.A (BICSA) y al Banco Metropolitano, S.A(BM). Al cierre del 2001 se amplió a más de 100 la red de Cajeros Automáticos del país bajo la marca RED y la emisión de tarjetas de bandas magnéticas por bancos cubanos alcanzó las 550,000 unidades; cifras que hoy han aumentado considerablemente gracias al esfuerzo del estado cubano.

Actualmente cualquier persona que lo desee puede tener una cuenta bancaria de tarjeta magnética. El proceso de creación de las tarjetas de banda magnética se inicia cuando una persona la cual desea una cuenta bancaria de tarjeta magnética se dirige a una sucursal del banco y pide crearse la cuenta de este tipo. Posterior a esto el cliente firma el contrato y se le crea la cuenta en esta sucursal, la que enviará un mensaje a la sucursal electrónica con los datos del cliente para se le asocie a la cuenta una tarjeta y un PIN. Esta tarjeta además de tener en la banda magnética la información del cliente tendrá el número de la cuenta y asociado a este el número que identifica la tarjeta, el cual está compuesto de 16 dígitos. Cuando se concluye con el proceso de creación de la tarjeta la sucursal electrónica la envía junto con el PIN asociado a la Central de Cajeros Automáticos para que se impriman. Luego de la impresión, las tarjetas y los pines asociados se envían a la sucursal donde el cliente solicitó la cuenta para ser entregados.

A mediados de 1999 se empezaban a realizar estudios para la introducción de tarjetas inteligentes en la localidad minera de Moa con el objetivo de poner a prueba su aplicación como tarjeta de débito y monedero electrónico, en pesos cubanos y en pesos cubanos convertibles. Con la ayuda de esta tarjeta se pretende lograr que la mayor parte de los trabajadores posean al menos una cuenta bancaria en la que se deposite su salario en ambas monedas y utilizando la tarjeta puedan hacer todas sus transacciones. De manera general los bancos de Cuba aún no realizan sus operaciones mediante el uso de tarjetas inteligentes.

1.3 La Criptografía

Criptografía es la ciencia de la seguridad de la información, aunque muchas veces ha sido descrita como el arte o la ciencia de la escritura secreta. La palabra viene de las palabras griegas *kryptos*, que significa "oculto" y *graphos*, que significa "escritura", literalmente "escritura oculta". La Criptografía es una rama de las Matemáticas que se complementa con el Criptoanálisis y ambas técnicas forman la ciencia llamada Criptología.

Por medio de la Criptografía se puede transformar un mensaje inteligible, denominado texto en claro, en un criptograma o texto cifrado para después devolverlo a su forma original, sin que nadie que vea el mensaje cifrado sea capaz de entenderlo, de esta manera es posible almacenar o transmitir información de forma tal que sea revelada únicamente a aquellos que deban verla. El método o sistema empleado para cifrar el texto en claro se denomina algoritmo de cifrado.

En tiempos modernos, la Criptografía se ha convertido en una compleja batalla entre los mejores matemáticos del mundo y de los ingenieros en sistemas computacionales. Una de las principales tareas que tiene hoy la Criptografía es dar solución a los problemas de identificación, autenticación y privacidad de la información en los sistemas informáticos ya que debido a la naturaleza de un medio no físico no resultan útiles los métodos tradicionales de sellar o firmar documentos, con propósitos comerciales o legales.

Las técnicas de encriptación suelen dividir a los algoritmos de cifrado en dos grupos. Los algoritmos que utilizan una clave secreta única para cifrar y descifrar los datos se conocen como algoritmos de cifrado simétricos o de clave secreta y son la base de los algoritmos de cifrado clásico y los que utilizan una clave para cifrar los datos y una clave distinta para descifrarlos se denominan algoritmos de cifrado asimétricos, o de clave pública y forman el núcleo de las técnicas de cifrado modernas.

1.3.1 Rijndael (el nuevo estándar AES)

Rijndael es el algoritmo de cifrado creado por Vincent Rijmen y Joan Daemen, fue seleccionado en el 2000 como el nuevo estándar de cifrado AES por permitir la mejor combinación de seguridad-velocidad-eficiencia, así como por su sencillez y flexibilidad. Este sorprende tanto por su innovador diseño como por su simplicidad práctica; aunque tras él se esconda un complejo trasfondo matemático.

El algoritmo Rijndael es un sistema simétrico de cifrado por bloques. Su diseño permite la utilización de claves de sistema con longitud variable siempre que sea múltiplo de 4 bytes. La longitud de las claves utilizadas por defecto son 128 (AES-128), 192 (AES-192) y 256 (AES-256) bits. De la misma manera el algoritmo permite la utilización de bloques de información con un tamaño variable siempre que sea múltiplo de 4 bytes, siendo el tamaño mínimo recomendado de 128 bits, y el tamaño mínimo de 16 bytes.

El proceso de cifrado del algoritmo Rijndael está formado por un conjunto de “rondas”, entendiéndose por “rondas” la aplicación reiterada de 4 funciones matemáticas diferentes e invertibles sobre la información que se desea cifrar. En este proceso para cada ronda se utilizan subclaves que se generan de la clave de sistema elegida. El número de reiteraciones o vueltas de las 4 transformaciones sobre la información depende de la versión del algoritmo que se utilice. El proceso de descifrado del algoritmo Rijndael es muy sencillo, consiste en sustituir las transformaciones utilizadas en el cifrado por sus inversas e invertir el orden de aplicación de dichas transformaciones o funciones matemáticas.

1.3.2 Seguridad del algoritmo

El algoritmo Rijndael es el sistema de criptografía simétrica más robusto que se conoce en la actualidad. Según los estudios publicados, no existen ningún mecanismo comprobado para invertir el algoritmo estándar AES más eficiente que una búsqueda exhaustiva de claves (ataque por fuerza bruta). Por tanto, la fortaleza del sistema dependerá sólo de la longitud de la clave. Un atacante que desee realizar un ataque por fuerza bruta deberá tener en cuenta las siguientes consideraciones:

Cuando el texto en claro y la clave son desconocidos.

Si el atacante sólo dispone de un bloque cifrado, debería cifrar con todas las claves posibles todos los bloques en claro posibles, para ir comparando el resultando con el bloque cifrado. Para valores estándar del algoritmo, como un clave de 256 bits y un tamaño de bloque de 256 bits, se demuestra fácilmente que el algoritmo no se puede invertir.

Si se considera que el tamaño del bloque es de " v bits" y el tamaño de la clave es " n bits". El atacante debería probar para cada clave 2^v bloques posibles, y repetir este proceso para todas las claves (2^n). Actualmente resulta computacionalmente imposible realizar estos cálculos para valores estándar del algoritmo. De todas formas aunque fuera factible el cálculo encontraríamos un número muy elevado de parejas clave-texto en claro diferentes que producen el mismo bloque cifrado, luego el atacante no tendría capacidad para identificar cual de todas esas parejas es la válida. Esta es la situación normal, que es cuando el atacante sólo dispone de texto cifrado. Demostrándose que el algoritmo resulta imposible de invertir.

Texto en claro conocido y texto cifrado conocido

El atacante posee el texto en claro y el texto cifrado. Ante esta situación puede hacer lo que se conoce como un ataque de fuerza bruta puro, es decir cifrar el texto en claro con todas las claves posibles hasta producir un resultado que coincida con el texto cifrado. En esta situación un atacante necesitaría aplicar el algoritmo Rijndael al texto en claro para estar seguro de que ha obtenido la clave el siguiente número de veces:

- Para una clave de 128 bits, se necesitaría aplicar 2127 veces el algoritmo Rijndael sobre el texto en claro y compararlo con el texto cifrado.

- Para una clave de 192 bits, el ataque necesita aplicar 2191 veces el algoritmo.
- Para una clave de 256 bits, el ataque necesita aplicar 2255 veces el algoritmo.

Este tipo de cálculo resulta hoy por hoy impracticable, siempre a la espera de futuros ordenadores cuánticos. Estos cálculos no son factibles debido al enorme número de combinaciones a calcular. Para hacerse una idea de las dimensiones de este número de cálculos, romper un algoritmo Rijndael de 128 bits es 4.722.366.482.869.645.213.696 veces menos probable que una persona gane la lotería y muera fulminado por un rayo el mismo día. (Muñoz Muñoz, 2004)

1.3.2 Criptografía simplificada con Visual Estudio .NET

Para implementar criptografía en .NET podemos utilizar varias clases incorporadas en el Framework sin ser necesario conocer los detalles del funcionamiento de los métodos de cifrado para poder utilizarlos, por ejemplo la clase Rijndael, que representa la clase base de la que deben heredarse todas las implementaciones del algoritmo de cifrado simétrico Rijndael y la clase SymmetricAlgorithm provista en el namespace System.Security.Cryptography, que representa la clase base abstracta de la que deben heredarse todas las implementaciones de algoritmos simétricos como DES, Rijndael, entre otros. Adicionalmente esta última genera la clave y el vector de inicialización adecuadas para cada algoritmo automáticamente.

Todas las clases de .NET incluidas en esta tabla se encuentran en el espacio de nombre System.Security.Cryptography

Método de cifrado	Tipo de algoritmo	Clase de .NET que implementa el método
Data Encryption Standard (DES)	Simétricos	DESCryptoServiceProvider
RC2 (RSA Data Security, Inc.)	Simétricos	RC2CryptoServiceProvider
Rijndael	Simétricos	RijndaelManaged
TripleDES (utiliza tres cifrados DES seguidos)	Simétricos	TripleDESCryptoServiceProvider
Firmas DSS (Digital Signature Algorithm)	Asimétricos	DSACryptoServiceProvider
RSA (siglas del nombre de sus inventores Rivest, Shamir y Adelman)	Asimétricos	RSACryptoServiceProvider

1.4 Tecnologías y Herramientas propuestas para el desarrollo del sistema

Para el desarrollo del sistema se realizó un estudio sobre las posibles herramientas a utilizar en su construcción. Teniendo en cuenta la tendencia actual y las novedades en este campo se hace la siguiente propuesta.

1.4.1 La tecnología .NET

Microsoft .NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años. La idea fundamental de Microsoft .NET es un cambio de enfoque en lo que es la informática, pasando de un mundo de aplicaciones, sitios Web y dispositivos aislados a una infinidad de computadoras, dispositivos, transacciones y servicios que se conectan directamente y trabajan en conjunto para ofrecer soluciones más amplias y ricas en contenido.

Las personas tendrán el control sobre cómo, cuándo y qué información desean. Las computadoras, sistemas y servicios estarán en capacidad de colaborar e interoperar mutuamente para beneficiar al usuario, mientras que las empresas podrán ofrecer sus productos y servicios a los clientes apropiados, en el momento correcto y de la forma precisa, combinando procesos de manera mucho más granular de lo que es posible hoy.

Visual Studio.NET

Visual Studio.NET es un IDE desarrollado por Microsoft para el sistema operativo Microsoft Windows, este es un entorno contenedor que integra diseñadores visuales para la construcción de aplicaciones de escritorio, aplicaciones Web ASP.NET, Servicios Web, etc. Tiende a ofrecer asistentes para facilitar la labor y hacer que la escritura de código sea la menor posible.

Visual Studio .NET permite diseñar, desarrollar, depurar e instalar soluciones basadas en .NET. Todos los lenguajes que son incorporados a Visual Studio pueden acceder a un conjunto común de herramientas, editores y diseñadores. Permite crear aplicaciones con formularios Windows o formularios Web que integren datos y lógica propia de la aplicación.

Visual Studio .NET soporta los lenguajes .NET: C#, Visual Basic .NET y Managed C++, además de C++. Los programas desarrollados en esos lenguajes no se compilan a código máquina ejecutable (como por ejemplo hace C++) sino que son compilados a algo llamado CIL. Cuando los programas ejecutan la

aplicación CIL, ésta es compilada en ese momento al código de máquina apropiado para la plataforma en la que se está ejecutando. Los programas compilados a CIL pueden ejecutarse sólo en plataformas que tengan una implementación de .NET framework.

Crystal Reports

Crystal Reports es una herramienta para diseño de reportes creada mucho antes del nacimiento de la tecnología .NET; gracias a ésta una de las partes más tediosas e importantes del diseño de aplicaciones, el diseño de reportes, se nos hace más sencilla. Crystal Reports puede ser utilizado con diferentes bases de datos, entre ellas MS Access, SQL Server, Oracle, Informix, etc.

Crystal Reports 10 es la herramienta de elaboración de informes estándar para Visual Studio .NET. Esta herramienta permite crear contenido interactivo con calidad de presentación y almacenar informes en plataformas Windows y Web. Utiliza una arquitectura abierta y flexible, con estándares como XML, que permite compartir informes e información en la Web. Proporciona a los desarrolladores de .NET la mejor API posible, así como ver los datos y realizar análisis mediante una integración perfecta en todos los lenguajes de Visual Studio que se implementen en .NET. Además hace posible que se creen rápidamente informes complejos de aspecto profesional. En lugar de utilizar codificación, la interfaz de Crystal Report Designer permite crear y dar formato al informe que se desee. El eficaz motor de informes procesa los criterios de formato, agrupamiento y creación de gráficos que se especifique.

Lenguajes de programación

C# es un lenguaje de programación orientado a objetos, es el lenguaje de .NET construido especialmente para adaptarse de manera natural al framework y aprovechar al máximo todas sus características. Es un lenguaje sencillo, potente y que permite a los programadores crear una gran variedad de aplicaciones.

Su sintaxis básica se deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET el cual es similar al de Java. La sintaxis de C# es muy expresiva, sencilla y fácil de aprender, simplifica muchas de las complejidades de C++ y, a la vez, ofrece funciones eficaces tales como tipos de valores que aceptan valores NULL, enumeraciones, delegados, métodos anónimos y acceso directo a memoria, que no se encuentran en Java.

C# también admite métodos y tipos genéricos, que proporcionan mayor rendimiento y seguridad de tipos. Fue diseñado para combinar el control a bajo nivel de lenguajes como C y la velocidad de programación

de lenguajes como Visual Basic. Permite acceder a las características avanzadas de la plataforma sobre la que se trabaje, crear código muy eficiente en aquellos puntos de la aplicación que son críticos y acceder a las Interfaces de Programación de Aplicaciones (APIs) existentes.

ASP.NET

ASP.NET es la tecnología que constituye hoy la nueva apuesta de Microsoft para la creación de páginas web dinámicas. Sustituye a su predecesor ASP en el intento de desbancar a PHP como lenguaje estrella para la creación de sitios dinámicos. Permite a los desarrolladores escribir código más limpio y más fácil de reutilizar y compartir. Es un marco de trabajo de programación generado en Common Language Runtime que puede utilizarse en un servidor para generar eficaces aplicaciones Web. Ofrece varias ventajas importantes acerca de los modelos de programación Web anteriores:

- **Mejor rendimiento:** Puede aprovechar las ventajas del enlace anticipado, la compilación just-in-time, la optimización nativa y los servicios de caché desde el primer momento. Esto supone un incremento espectacular del rendimiento antes de siquiera escribir una línea de código.
- **Compatibilidad con herramientas de primer nivel:** El marco de trabajo de ASP.NET se complementa con un diseñador y una caja de herramientas muy completos en el entorno integrado de programación (Integrated Development Environment, IDE) de Visual Studio.
- **Eficacia y flexibilidad:** La biblioteca de clases de .NET Framework, la Mensajería y las soluciones de Acceso a datos se encuentran accesibles desde el Web de manera uniforme. ASP.Net es también independiente del lenguaje, por lo que puede elegir el que mejor se adapte a la aplicación o dividirla en varios lenguajes.
- **Simplicidad:** Facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la implementación y la configuración de sitios.
- **Facilidad de uso:** Emplea un sistema de configuración jerárquico, basado en texto, que simplifica la aplicación de la configuración al entorno de servidor y las aplicaciones Web. No se requiere el reinicio del servidor, ni siquiera para implementar o reemplazar el código compilado en ejecución.
- **Escalabilidad y disponibilidad:** El motor de tiempo de ejecución de ASP.NET controla y administra los procesos de cerca, por lo que si uno no se comporta adecuadamente (filtraciones, bloqueos), se puede crear un proceso nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes.

- Posibilidad de personalización y extensibilidad: Permite insertar código en el nivel adecuado, siendo posible extender o reemplazar cualquier subcomponente del motor de tiempo de ejecución de ASP.NET con su propio componente escrito personalizado.
- Seguridad: Con la autenticación de Windows integrada y la configuración por aplicación, se puede tener la completa seguridad de que las aplicaciones están a salvo.

1.4.2 Servicios Web (Web Services)

Los Web Services son una nueva tecnología que permite que dos o más aplicaciones heterogéneas compartan datos, vía Internet, sin importar su plataforma o ubicación. De esta manera, ninguna de las dos partes modifica su infraestructura ni compromete su seguridad pero las dos obtienen exactamente los mismos datos en el momento de ser generados por alguna de las partes. Los Web Services trabajan bajo la arquitectura cliente/servidor y pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos Web estándar, como XML, SOAP, UDDI o WSDL, básicamente la interacción entre ambos está definida en un archivo con formato XML que describe formalmente y sin ambigüedades el tipo de datos a transmitir y los procesos disponibles. Esto ayuda a tener muy altos niveles de confiabilidad, interoperabilidad e integración.

Ofrecen las siguientes ventajas tecnológicas:

- Interoperabilidad: Cualquier servicio Web puede interactuar con cualquier otro servicio Web. El protocolo SOAP permite que cualquier servicio pueda ser ofrecido o utilizado independientemente del lenguaje o ambiente en que se haya desarrollado.
- Omnipresencia: Los servicios Web se comunican utilizando HTTP y XML. Cualquier dispositivo que trabaje con éstas tecnologías puede tanto ser un cliente del servicio como servidor en algunas circunstancias.
- Esfuerzo mínimo: Los conceptos detrás de los servicios de Web son fáciles de comprender y se ofrecen herramientas de desarrollo específicas por WebLogic, Sun, Apache que permiten a los programadores implementar rápidamente servicios web con SOAP.

1.5 Fundamentación de la metodología propuesta para el desarrollo del sistema

La rama de la metodología, dentro de la ingeniería de software, se encarga de elaborar estrategias de desarrollo de software que promuevan prácticas adaptativas en vez de predictivas; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente. Entre las metodologías que hoy se consideran más importantes se encuentran: RUP, XP y MSF. (Sanchez, 2004)

Lo más importante para elegir la metodología a usar para el desarrollo del software, es determinar el alcance que este tendrá y luego ver cual es la que más se acomoda a la aplicación. La Metodología RUP es la propuesta que se hace para desarrollar este proyecto pues la Metodología XP se recomienda para proyectos de corto plazo y aunque la Metodología MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología se opta por RUP ya que es más adaptable para proyectos de largo plazo y el equipo de desarrollo tiene más experiencia en la misma.

1.5.1 Proceso Unificado de Rational

El Proceso Unificado de Desarrollo de Software o Proceso Unificado (PU) es un popular framework de proceso de desarrollo de software iterativo e incremental, el cual puede y debe ser personalizado a las características propias de las organizaciones o proyectos. RUP o el Proceso Unificado de Rational es el refinamiento realizado por Rational Software del Proceso Unificado, más conocido y mejor documentado que existe actualmente, este en unión del Lenguaje Unificado de Modelado (UML), constituye una metodología unificada, cohesiva y comprensiva para el desarrollo de sistemas del software, la cual hoy tiene la primacía ante importantes estándares en la comunidad de desarrollo.

RUP es una metodología Orientada a Objeto que proporciona una línea directiva para todos los aspectos y estados del desarrollo de un software, o sea proporciona una guía práctica y detallada a través de todas las fases del ciclo de desarrollo que se puede ajustar para adaptarse fácilmente a las necesidades de una amplia variedad de proyectos y organizaciones. RUP unifica el equipo de desarrollo de software y optimiza la productividad de cada miembro ofreciéndoles la experiencia de los líderes en la industria y las lecciones aprendidas en miles de proyectos, permitiendo así producir software de alta calidad, en el tiempo previsto y dentro del presupuesto.

RUP tiene tres características fundamentales:

- Guiado por los casos de uso: emplea los casos de uso para manejar el desarrollo del proceso desde el inicio del proyecto hasta el despliegue.
- Centrado en la arquitectura: El proceso busca entender la estática más significativa y los aspectos dinámicos en términos de la arquitectura del software. La arquitectura es una función de las necesidades de los usuarios y se captura en la esencia de los casos de uso.
- Iterativo e incremental: Se dividen los proyectos grandes en proyectos más pequeños o mini-proyectos, cada mini-proyecto comprende una iteración que produce un incremento. Una iteración puede abarcar todos los flujos de trabajo en el proceso.

RUP consiste en ciclos que se pueden repetir a lo largo del tiempo de vida de un sistema. Un ciclo consiste en cuatro fases de desarrollo: Inicio, Elaboración, Construcción y Transición, cada una de las cuales está organizada en un número de iteraciones separadas. Cada fase enfoca el equipo del proyecto en un aspecto particular del proyecto y tiene asociados varios hitos. Estos hitos ayudan a evaluar el progreso del proyecto y asegura que el proyecto se entregará con los rasgos y la calidad requerida. Para pasar de una fase a otra es necesario haber cumplido con el hito de dicha fase. Cada ciclo se concluye con un release del producto. Las fases son las siguientes:

Fase de Inicio: En esta fase la idea central es desarrollar una visión del producto. Se repasa y confirma la comprensión que se tiene de los procesos del negocio y se definen los casos de uso del negocio. La fase de Inicio establece la viabilidad del producto y delimita el alcance del proyecto.

Fase de Elaboración: Durante esta la fase la mayoría de los Casos del Uso se especifican en detalle y se diseña la arquitectura base del sistema.

Fase de Construcción: Durante la fase de Construcción el software es diseñado, escrito y probado, o sea el producto se mueve de la línea base de la arquitectura a un sistema listo para ser entregado al usuario final. La línea base de la arquitectura crece para convertirse en el sistema completo tal como el diseño es refinado en el código.

Fase de Transición: En esta fase se entrega el sistema a los usuarios finales, la meta es asegurar la satisfacción de los interesados. Esta fase a menudo se comienza con un release beta de la aplicación. Otras actividades incluyen la realización del manual de usuario, así como la identificación y corrección de

los errores y la realización de ajustes y actualizaciones basadas en la retroalimentación con los usuarios finales. Esta fase culmina con un postmortem consagrado a aprender lecciones para ciclos futuros.

1.5.2 Lenguaje de modelación

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es un lenguaje de modelado de sistemas de software ampliamente reconocido y utilizado. Es un lenguaje gráfico que se utiliza para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Puede soportar diferentes metodologías de desarrollo de software, como es el caso de RUP, pero no especifica cual de ellas usar. Con UML se pueden modelar desde complejos sistemas para empresas hasta sistemas basados en web o sistemas de tiempo real. Es un lenguaje muy expresivo, que cubre todas las vistas necesarias para desarrollar un sistema.

1.5.3 Herramientas de Modelado Visual y Desarrollo

Rational es actualmente conocida como una familia de software de IBM para el levantamiento de requerimientos, diseño, construcción, pruebas y administración de proyectos en el proceso desarrollo de software. Sus productos están centrados en la metodología RUP.

Entre sus productos tenemos a Rational Rose Enterprise Edition, ahora con el nombre Rational Software Architect, su última versión mejorada.

Rational Rose Enterprise Edition es una herramienta de representación gráfica para modelar las etapas del ciclo de vida del desarrollo del software, utilizando UML, y que al permitir la aplicación práctica de la metodología RUP consigue agilizar el trabajo. Esta herramienta permite mejorar la productividad en el desarrollo de software disminuyendo el coste de los mismos en términos de tiempo y de dinero, logrando así software de mayor calidad y la satisfacción del cliente.

1.6 Conclusiones

Las formas de autenticación antes mencionada no corresponden con el modelo de sistema que pretendemos desarrollar por lo que nuestro módulo utilizará un método de autenticación diferente: utilizaremos una matriz de diez filas con diez columnas y el Número de identificación personal (PIN) para identificar a los clientes.

Para el desarrollo del sistema se utilizarán las siguientes tecnologías por las características fundamentadas con anterioridad: la plataforma .NET de Microsoft (C# y ASP .NET) y como metodología y herramienta para el diseño RUP y Rational Rose Enterprise Edition respectivamente.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En este capítulo se describe la situación problemática y los procesos que serán objeto de automatización, se realiza la propuesta del sistema a desarrollar y se analizan los requerimientos funcionales y no funcionales. Se realiza además el modelado del dominio, así como la definición de los casos de uso, los actores que intervienen en los mismos y se muestra el diagrama de casos de uso.

2.2 Situación problemática

En la actualidad en el país el cobro de los servicios a la población se efectúa de forma dispersa. Esta situación causa una serie de inconvenientes para la población y para empresas e instituciones que brindan estos servicios pues para la población el pago de servicios se dificulta al tener que asistir a las oficinas de cobro, que generalmente están geográficamente aisladas unas de otras, lo cual no es beneficioso para el país pues esta tarea se realiza en horario laboral, además para las empresas el cobro de los servicios es una gestión que demanda de gran cantidad de personal en oficina para el trabajo en las áreas comerciales y además gran cantidad de personal circulando por la ciudad con dinero encima. En Dada la situación anterior surge en el país la necesidad de desarrollar un sistema para automatizar el pago de los servicios, con este objetivo surge TeleBanca. Para que Telebanca opere correctamente se hace imprescindible la realización del módulo Gestión de Autenticación de los Clientes pues sin este no es posible obtener la entrada principal de información del sistema: los datos de las personas que desean pertenecer a TeleBanca, los datos de los clientes. Dada la situación anterior no es posible crear la documentación necesaria para identificar a los clientes asociados, motivo por el cual no será posible que las personas puedan efectuar el pago de los servicios utilizando la vía telefónica pues no habrá manera posible de identificar un cliente entre los demás. Lo anteriormente expuesto evidencia que sin este módulo Telebanca no podría funcionar ya que jamás se podría obtener la información de los clientes, entrada principal de este sistema.

2.3 Objeto de automatización

Se desea automatizar todos los procesos de gestión de autenticación de los clientes de Telebanca, lo que incluye la posibilidad de que se generen las autenticaciones de los clientes. Se implementaría la

funcionalidad de dar de baja a un cliente que no desee pertenecer más a TeleBanca, así como que será posible crearle una nueva documentación a un cliente que lo solicite, ya sea por pérdida de la documentación anterior o cualquier otro motivo personal. Se automatizará el proceso de impresión de tarjetas y pines. Será posible establecer una comunicación directa entre los bancos asociados y TeleBanca, lo cual permitirá que se notifiquen las autenticaciones creadas e impresas y las solicitudes de baja procesadas al final del día, ya sea automáticamente o de forma auxiliar. También será objeto de automatización generar reportes de las impresiones de lotes de pines y tarjetas respectivamente, así como de los lotes que se imprimirán, lo cual permitirá conocer los materiales que será necesario solicitarle al almacén. Además se podrán captar ficheros de matrices para que estas sean posteriormente impresas en las tarjetas de autenticación. Para desarrollar todo lo que se propone en el sistema se cuenta con las herramientas necesarias, así como el apoyo de la institución y el soporte técnico necesario.

2.4 La información que se maneja

La información que se manejará serán los datos de las solicitudes de autenticación de los clientes:

- Número de la tarjeta.
- Nombre del cliente.
- Apellidos del cliente.
- Identificación del cliente.
- Tipo de identificación. (CI o pasaporte)
- País de nacionalidad del cliente.
- Número de sucursal.
- Matriz.
- Número de PIN.
- Fecha de orden de impresión.
- Lote de impresión: Las tarjetas y pines se agrupan por lotes para contar con una mejor organización en el momento de la impresión.
- Registro de las acciones del usuario: Se utiliza cuando se reimprimen tarjetas o pines, se registra el usuario de la operadora, la acción que realiza, fecha, hora y el identificador de las tarjetas que se reimprimieron.

- **Histórico:** Se guardan las tarjetas que se eliminan, en caso de que el cliente ya no desee pertenecer a TeleBanca o en caso de que este haya solicitado la creación de nuevos documentos de autenticación.
- **Notificaciones:** Se guarda diariamente la cantidad de tarjetas creadas, impresas y eliminadas para cada banco.

2.5 Propuesta del Sistema

El sistema que se propone presenta funcionalidades que ayudarán al trabajo de las operadoras que laborarán en el Call Center de TeleBanca, pues el mismo mostrará una interfaz fácil y amigable, en relación a la poca experiencia que las mismas tienen en la interacción con sistemas informáticos. Estas contarán con las herramientas de trabajo necesarias para realizar los distintos procesos de gestión de autenticación de los clientes. A continuación se explican las características generales que tendrá el sistema para realizar cada uno de los procesos requeridos.

2.5.1 Crear Autenticación

Este proceso no tiene interfaz visual para el usuario ya que se ejecuta internamente en el Sistema. A una hora predefinida el Sistema comienza a pedirle a los Web Services de los bancos asociados las Solicitudes de Creación de Autenticaciones y crea las tarjetas de autenticación de los clientes. Al finalizar este proceso se guarda en las notificaciones el resultado de la operación para que al día siguiente la Operadora de Autenticación tenga constancia del éxito o fracaso de la misma.

2.5.2 Crear Lote

Para crear un lote el Administrador de BT elige la opción “Crear Lote” que se encuentra en la página principal del módulo. Elige la sucursal a la cual desea crearle el lote y posteriormente selecciona las tarjetas, organizadas por fechas de creación, que integrarán el lote. Finalmente selecciona la opción “Crear” para crear un nuevo lote.

2.5.3 Imprimir Tarjetas.

Para imprimir las tarjetas la Operadora de Impresión de Tarjetas selecciona la opción “Imprimir Tarjeta” en la página principal del módulo, luego selecciona el lote o los lotes que desea imprimir y la opción

“Imprimir”. El sistema actualiza los datos del lote o los lotes seleccionados, cambiándoles el estado tarjeta a “Impreso” y finalmente imprime automáticamente las tarjetas.

2.5.4 Imprimir Pines.

Para imprimir los pines la Operadora de Impresión de Pines selecciona la opción “Imprimir Pines” en la página principal del módulo, luego selecciona el lote o los lotes que desea imprimir y la opción “Imprimir”. El sistema actualiza los datos del lote o los lotes seleccionados, cambiándoles el estado pin a “Impreso” y finalmente imprime automáticamente los pines.

2.5.5 Reimprimir y Finalizar Tarjetas

Para reimprimir o finalizar un lote la Operadora de Impresión de Tarjetas selecciona en la página principal del módulo la opción “Reimprimir Tarjetas”. La operadora puede seleccionar el lote que desea reimprimir y la opción “Reimprimir” para la reimpresión de un lote que haya quedado defectuoso o puede seleccionar el o lo(s) lotes que desea finalizar y la opción “Finalizar” para finalizar los lotes que no se requieran volver a imprimirse, en este último caso el sistema cambia el estado tarjeta “Impreso” por “Finalizado” y busca los lotes que tengan el estado pin y el estado tarjeta “Finalizado” y a las Tarjetas que pertenecen a estos lotes le cambia el estado “Creada” por “Activa” y le pone en la fecha de orden de impresión la fecha del sistema.

2.5.6 Reimprimir y Finalizar Pines

Para reimprimir o finalizar un lote la Operadora de Impresión de Pines selecciona en la página principal del módulo la opción “Reimprimir Pines”. La operadora puede seleccionar el lote que desea reimprimir y la opción “Reimprimir” para la reimpresión de un lote que haya quedado defectuoso o puede seleccionar el o lo(s) lotes que desea finalizar y la opción “Finalizar” para finalizar los lotes que no se requieran volver a imprimirse, en este último caso el sistema cambiará de los lotes seleccionados el estado pin “Impreso” a “Finalizado” y busca los lotes que tengan el estado pin y el estado tarjeta “Finalizado” y a las Tarjetas que pertenecen a estos lotes le cambia el estado “Creada” por “Activa” y le pone en la fecha de orden de impresión la fecha del sistema.

2.5.7 Captar Matrices

Para captar un fichero de matrices el Administrador de BT debe seleccionar la opción “Captar Matriz” que se encuentra en la página principal del módulo, luego debe seleccionar el fichero donde se encuentran las matrices y seleccionar la opción “Encriptar y Guardar”. Si no hay ningún problema con este fichero o con el formato de las matrices que se encuentran en el mismo las matrices se guardaran cifradas en la base de datos.

2.5.8 Procesar Solicitudes de Baja de Tarjetas

Este proceso no tiene interfaz visual para el usuario ya que se ejecuta internamente en el Sistema. A una hora predefinida el Sistema comienza a pedirle a los Web Services de los bancos asociados las Solicitudes de Baja de Tarjetas, luego elimina las tarjetas de autenticación de los clientes que solicitaron darse de baja de TeleBanca y les crea una nueva tarjeta de identificación a los clientes que solicitaron el cambio. Al finalizar este proceso se guarda en las notificaciones el resultado de la operación para que al día siguiente el Administrador de BT tenga constancia del éxito o fracaso de la misma.

2.5.9 Enviar Conciliaciones Automáticas

Este proceso no tiene interfaz visual para el usuario ya que se ejecuta internamente en el Sistema. A una hora predefinida el Sistema comienza a enviarle a los Web Services de los bancos asociados la cantidad de tarjetas que se crearon, se imprimieron y se eliminaron. Al finalizar este proceso se guarda en las notificaciones el resultado de la operación para que al día siguiente Administrador de BT tenga constancia del éxito o fracaso de la misma.

2.5.10 Enviar Conciliaciones Auxiliares

Para enviar las conciliaciones auxiliares el Administrador de BT selecciona el tipo de conciliación que desea enviar: Conciliación de Tarjetas Creadas, de Tarjetas Activas o de Tarjetas dadas de Baja, luego selecciona el banco al cual desea enviar la conciliación y por último escoge los días para los cuales desea enviar la conciliación.

2.5.11 Realizar Reporte

Para realizar un reporte la operadora selecciona en la página principal del Módulo Gestión de Autenticación de los clientes la opción realizar reporte, luego puede seleccionar el tipo de reporte que

desea realizar: Reporte de pines impresos o Reporte de tarjetas impresas, según la operadora que sea, o Reporte de lotes por imprimir.

2.6 Modelo del negocio

Los objetivos del modelo del negocio son comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema, comprender los problemas actuales de la organización e identificar las mejoras potenciales, asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización y por último derivar los requerimientos del sistema que va a soportar la organización, con lo cual se refuerza la idea de que sea el propio negocio lo que determine los requisitos. Hay varias alternativas de desarrollar este proceso: se puede realizar un Modelo del Negocio o un Modelo del Dominio.

La empresa Banca Telefónica surge ante la necesidad de que la población pague de manera centralizada todos los servicios públicos que hoy se cobran en formas tan variadas y en lugares tan distintos y dispersos entre sí, entre ellos el pago de la electricidad, de servicios telefónicos, de servicios de agua, etc. Se puede entender por lo tanto no existe un negocio bien definido, ya que es una empresa que surge gracias a la creación del sistema que se pretende realizar y por lo tanto se realizará un Modelo del Dominio que capturará los tipos más importantes de objetos y eventos necesarios para realizar el Módulo Gestión de Autenticación de los Clientes.

2.7 Modelo del Dominio

El Modelo de Dominio muestra las clases conceptuales significativas en el dominio del problema, es decir, en el mundo real. UML utiliza diagramas de clases para representar el modelo del dominio, que muestran objetos del dominio o clases conceptuales y asociaciones entre ellas.

Las clases conceptuales pueden representar objetos que existen o eventos que suceden en el entorno donde estará el sistema. Para identificar las clases conceptuales se usan estrategias tales como utilizar una lista de categorías habituales que normalmente merece la pena tener en cuenta tales como objetos tangibles o físicos, lugares, transacciones, etc. y realizar un análisis del lenguaje natural en las descripciones textuales de un dominio, o sea, identificar las clases conceptuales mediante frases nominales. Es importante utilizar siempre el vocabulario del dominio al nombrar las clases conceptuales, excluir las características irrelevantes y no añadir conceptos que se encuentren fuera del problema que se

está estudiando. De manera general el Modelo del Dominio ayuda a los usuarios, clientes, desarrolladores e interesados a utilizar un vocabulario común para poder entender el contexto en que se sitúa el sistema.

2.7.1 Conceptos principales

Banco: Institución financiera cuyos servicios tienen que ver con los procesos de depósito, préstamos, créditos, etc.

Sucursal: Oficina perteneciente a un banco para distribuir el trabajo en las diferentes regiones.

Cliente: Persona que asiste a la sucursal para crearse una cuenta bancaria de tarjeta magnética.

Sucursal electrónica: Oficina que se encarga de la personalización de las tarjetas de las sucursales, o sea se encarga de la asignación de la tarjeta a un cliente específico.

Cuenta bancaria de tarjeta magnética: Depósito que puede ser utilizado en cualquier momento a voluntad y requerimiento del cliente a través de una tarjeta magnética y un PIN. Es una forma de captación de depósitos que constituyen un servicio que ofrece el banco y al cual acuden y se acogen los clientes debido a la confianza, comodidad, seguridad y el control que se le da a los fondos que ponen bajo la custodia del banco.

Contrato: Documento que firma el cliente como constancia de la realización de la cuenta bancaria.

Central de Cajeros: Oficina que se dedica a la introducción en la banda magnética e impresión en el frente de la tarjeta magnética de determinada información única relacionada con los datos del cliente.

Tarjeta magnética: Documento de autenticación del banco que contribuye a la identificación de los clientes y por tanto a la seguridad.

PIN: Número de Identificación Personal (clave) asociado a una tarjeta.

2.7.2 Eventos principales

Pertenece_a: Proceso donde una sucursal pertenece a un banco.

Asiste_a: Proceso donde un cliente asiste a una sucursal para solicitar la creación de una cuenta bancaria y posteriormente para recibir la tarjeta magnética y el PIN correspondiente.

Firma_un: Proceso donde un cliente firma un contrato.

Es_archivado: Proceso donde un contrato es archivado en la sucursal como constancia de la cuenta bancaria abierta al cliente.

Envía_información: Proceso donde la sucursal envía los datos de los nuevos clientes a la sucursal electrónica.

Personaliza: Proceso donde la sucursal electrónica realiza la asignación de la tarjeta a un cliente específico.

Envía_información_personalizada: Proceso donde la sucursal electrónica envía a la Central de Cajeros la información personalizada de los clientes para que esta última imprima la documentación.

Imprime: Proceso donde la Central de Cajeros imprime la documentación de los clientes.

Tiene_asociado: Proceso mediante el cual a una tarjeta se le asocia un PIN

Es_enviada: Proceso donde se envían las tarjetas a las sucursales para sean entregadas a los clientes.

Es_enviado: Proceso donde se envían los pines a las sucursales para sean entregados a los clientes.

Tiene: Proceso donde un cliente tiene una cuenta bancaria de tarjeta magnética, así una tarjeta y un PIN para poder usar su cuenta.

Tiene_asignado: Proceso donde una cuenta bancaria de tarjeta magnética tiene asignado un PIN para poder acceder a esta.

Tiene_asignada: Proceso donde una cuenta bancaria de tarjeta magnética tiene asignada una tarjeta para poder acceder a esta.

2.7.3 Diagrama de clases del dominio

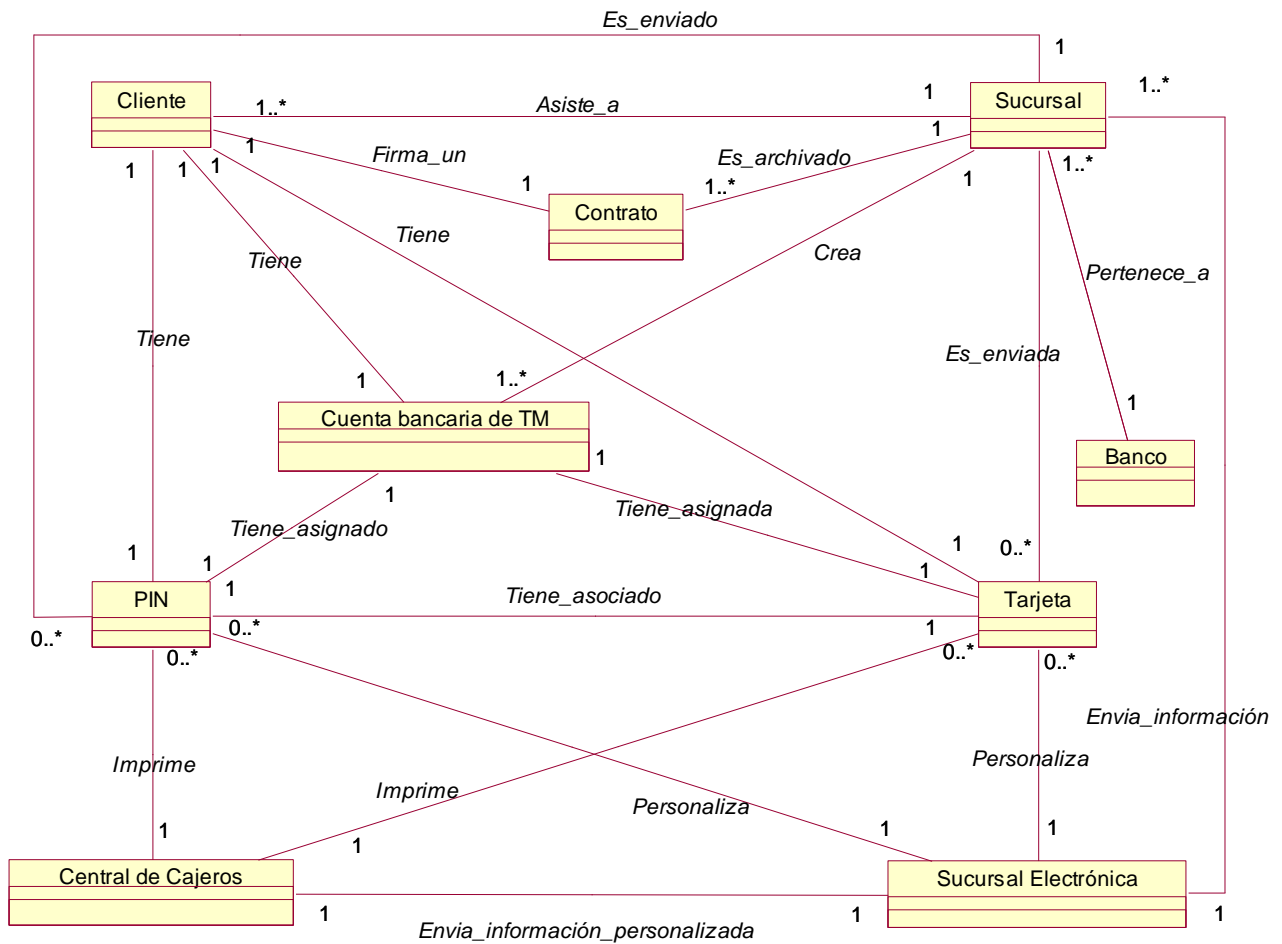


Figura 2.1 Diagrama de Clases del Dominio.

2.8 Especificación de Requerimientos del Sistema

2.8.1 Requerimientos funcionales

Los requerimientos funcionales no son más que la determinación clara y concisa de qué debe ser capaz de hacer el sistema. Definen las funciones que el sistema será capaz de realizar y describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

RF1: Captar solicitudes de generación de autenticaciones.

*Captar las solicitudes de altas llegadas de las sucursales a través de los Web Services.

RF2: Crear la tarjeta con matriz y pin asociado.

*Generar las Autenticaciones de los Clientes (crear una tarjeta con pin y matriz asociada para cada cliente).

RF3: Captar Solicitudes de Baja de Tarjetas.

*Captar las solicitudes de baja llegadas de las sucursales a través de los Web Services. Las solicitudes de baja pueden ser “deshabilitadas” cuando el cliente no puede o no desea utilizar más la tarjeta y “pedida” si este no puede usar la tarjeta, ya sea por pérdida u otro motivo, y desea que se le cree una nueva.

RF4: Permitir crear lotes de impresión.

*Los lotes de impresión están compuestos por las tarjetas de una sucursal.

RF5: Imprimir las tarjetas de autenticación de los clientes.

*Se imprime la matriz y el número de la tarjeta.

RF6: Reimprimir las tarjetas de autenticación de los clientes.

*Se reimprimen tarjetas en caso de que las impresiones anteriores no hayan tenido la calidad requerida.

RF7: Imprimir los pines de autenticación de los clientes.

RF8: Reimprimir los pines de autenticación de los clientes.

*Se reimprimen pines en caso de que las impresiones anteriores no hayan tenido la calidad requerida.

RF9: Finalizar la impresión de los lotes.

*Se finaliza un lote para que no se posible volverlo a imprimir.

RF10: Captar las matrices de identificación de los clientes.

*Captar las matrices de un fichero y guardarlas cifradas en la base de datos.

RF11: Permitir realizar conciliaciones de las tarjetas creadas, activas y dadas de baja con los bancos.

*Se concilian las tarjetas que se imprimieron y las solicitudes de baja que se procesaron.

RF12: Permitir realizar reportes de los lotes impresos en el día y reportes de la cantidad de pines y tarjetas que faltan por imprimirse de cada sucursal.

2.8.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Requerimientos de apariencia o interfaz externa

La aplicación propuesta será usada por personas que no necesariamente tienen conocimientos profundos de informática, por lo que la interfaz debe ser amigable y fácil de usar, de manera que no sea una dificultad para los usuarios el uso de la misma.

Requerimientos de Usabilidad

La aplicación será utilizada por operadores que no necesariamente tendrán experiencia en el uso de la computadora, a estos se les dará un adiestramiento básico en el uso de la computadora y fundamentalmente en el uso de la aplicación. Estas personas tendrán un nivel de acceso en la aplicación, en la cual realizarán solamente las acciones que tienen asignadas.

Requerimientos de Rendimiento

Para un funcionamiento óptimo de la aplicación se seguirán las diferentes técnicas de elaboración en la Web, que faciliten el rápido acceso a sus páginas. La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor, y la velocidad de las consultas en la Base de Datos.

Requerimientos de Soporte

Para garantizar el soporte de esta herramienta, se documentará la aplicación con un manual de ayuda para los usuarios, así como la posibilidad de emitir sus quejas y sugerencias a los desarrolladores de la herramienta, por correo o por teléfono, realizar mantenimiento al sistema y darle solución a cualquier problema que surja con la aplicación.

Requerimientos de Portabilidad

El sistema deberá funcionar sobre cualquier plataforma. Los servidores Web y de Base de datos deben estar en la misma computadora bajo el sistema operativo Windows Server 2003.

Requerimientos de Seguridad

Confiabledad: la información manejada por el sistema debe estar protegida de acceso no autorizado y divulgación. Las matrices que se asociarán a las distintas tarjetas de autenticación de los clientes deben estar cifradas en la Base de Datos.

Integridad: la información manejada por el sistema debe ser objeto de cuidadosa protección contra la corrupción y estados inconsistentes.

Disponibilidad: La aplicación deberá estar disponible en todo momento para aquellas personas con acceso a la información y los mecanismos utilizados para lograr la seguridad no deben ser un obstáculo a los usuarios para obtener los datos deseados en un momento dado.

Requerimientos de Software

Para la implementación del sistema se usará como herramienta de desarrollo Visual Studio .Net y Gestor de Base de Datos SQL Server 2003. El Servidor Web y de Base de Datos deben estar bajo el sistema operativo Windows Server 2003 y en las computadoras de los clientes solo se requiere de navegador para Internet o Intranet, bajo cualquier sistema operativo.

Requerimientos de Hardware

En el cliente se requiere de una máquina de 128 MB de RAM y en la computadora donde se encuentran el servidor Web y el de Base de Datos debe haber 2 Giga de RAM y 80 GB de capacidad del disco duro. Todas las computadoras que requieran conectarse con los servidores, dígase la de administración como las de los usuarios, deben estar conectadas en red, la cual debe tener al menos 100 MB de velocidad. También se necesitan dos impresoras, una para imprimir pines que debe ser de matriz de punto y otra para las tarjetas que debe ser laser.

Restricciones en el diseño y la implementación

Se debe realizar una herramienta que permita dar respuesta en el menor tiempo posible, garantizando la calidad del sistema y la conexión con el Web Services, la base de datos debe ser desarrollada en SQL Server. Para garantizar el desarrollo de la aplicación se utilizará como guía el Proceso Unificado de Desarrollo, obteniendo como documentación los distintos artefactos propuestos. Se utilizará para realizar

los modelos del sistema UML (Unified Modelling Language) y como herramienta de apoyo a este lenguaje de modelación se utiliza Rational Rose Enterprise Edition. Para la gestión de configuración se utilizará Subversión y Tortoise.

2.9 Definición de los Casos de uso del Sistema

Utilizando las facilidades que brinda el UML, se representarán los requisitos funcionales del sistema mediante un diagrama de casos de uso. Para ello hay que definir de acuerdo a lo planteado en los epígrafes anteriores, cuáles serían los actores que van a interactuar con el sistema y los casos de uso que van a representar las funcionalidades.

2.9.1 Descripción de los actores

Un actor representa cualquier cosa que interactúe con el sistema, puede ser un humano o un software o hardware. (Álvarez, y otros, 2000) Por lo que un actor no es parte del sistema, sino un rol que se juega dentro del sistema, este puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una máquina que interactúa con el sistema.

Actor	Descripción
Operadora de Impresión de Pines	Persona encargada de imprimir los Pines que han sido asociados. Además realiza las acciones de la Operadora de Autenticación.
Operadora de Impresión de Tarjetas	Persona encargada de imprimir las Tarjetas que han sido creadas. Además realiza las acciones de la Operadora de Autenticación.
Operadora de Autenticación	Persona encargada de realizar reportes, es una generalización de la Operadora de Impresión de Pines y de Tarjetas.
Administrador de BT	Persona encargada de administrar los servicios que brinda la banca telefónica y los bancos asociados, de captar las matrices para la autenticación a los

	clientes, de crear los lotes de impresión y de enviar las conciliaciones a los Bancos.
Web Service Banco	Aplicación a través de la cual se interactúa con la Base de Datos del Banco.

2.9.2 Casos de uso del Sistema

Un caso de uso “es la descripción de una secuencia de interacciones entre el sistema y uno o más actores en la que se considera al sistema como una caja negra y en la que la que los actores obtienen resultados observables”. (Pressman, 2001)

Los casos de uso no son parte del diseño (cómo), sino parte del análisis (qué). De forma que al ser parte del análisis nos ayudan a describir qué es lo que es sistema debe hacer. Los casos de uso son lo que hace el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario.

CU-1	CU_Crear Autenticación
Actor	Web Service Banco
Descripción	El caso de uso se inicia cuando el sistema automáticamente, a una hora escogida previamente se ejecuta y comienza la generación de autenticaciones. Obtiene todas las solicitudes de generación de autenticaciones de los bancos con los datos de los clientes en caso de que existan solicitudes. El sistema genera dichas autenticaciones y guarda un mensaje de notificación.
Referencia	RF1, RF2

CU-2	CU_Crear Lote
Actor	Administrador de BT
Descripción	El caso de uso se inicia cuando el Administrador de BT elige la opción Crear Lote, creándose así un lote de impresión para los pines y tarjetas de una sucursal.
Referencia	RF4

CU-3	CU_Imprimir Tarjetas
Actor	Operadora de Impresión de Tarjetas
Descripción	El caso de uso se inicia cuando la Operadora de Impresión de Tarjetas elige la opción Imprimir Tarjetas. El sistema imprime el lote de tarjetas seleccionado por la Operadora de Impresión de Tarjetas.
Referencia	RF5

CU-4	CU_Imprimir Pines
Actor	Operadora de Impresión de Pines
Descripción	El caso de uso se inicia cuando la Operadora de Impresión de Pines elige la opción Imprimir Pines. El sistema imprime el lote seleccionado por la Operadora de Impresión de Pines.
Referencia	RF7

CU-5	CU_Reimprimir y Finalizar Tarjetas
Actor	Operadora de Impresión de Tarjetas
Descripción	El caso de uso se inicia cuando la Operadora de Impresión de Tarjetas elige la opción Reimprimir Tarjetas. La Operadora de Impresión de Tarjetas puede reimprimir un lote de tarjetas impreso anteriormente o finalizar uno o varios lotes para que no sea posible volverlos a imprimir.
Referencia	RF6, RF9

CU-6	CU_Reimprimir y Finalizar Pines
Actor	Operadora de Impresión de Pines
Descripción	El caso de uso se inicia cuando la Operadora de Impresión de Pines elige la opción Reimprimir Pines. La Operadora de Impresión de Pines puede reimprimir un lote de pines impreso anteriormente o finalizar uno o varios lotes para que no sea posible volverlos a imprimir.

Referencia	RF6, RF9
------------	----------

CU-7	CU_Captar Matrices
Actor	Administrador de BT
Descripción	El caso de uso se inicia cuando el Administrador de BT selecciona la opción Captar Matrices. El Sistema muestra un fichero donde se encuentran las nuevas matrices y luego permite cifrarlas y guardarlas.
Referencia	RF10

CU-8	CU_Procesar Solicitudes de Baja de Tarjetas
Actor	Web Service Banco
Descripción	El caso de uso se inicia cuando el Sistema le pide al Banco las Solicitudes de Baja de Tarjetas. Si existen dichas solicitudes, el Sistema permite deshabilitar la tarjeta y hacer una nueva autenticación al cliente si este lo solicitó.
Referencia	RF3

CU-9	CU_Enviar Conciliaciones Automáticas
Actor	Web Service Banco
Descripción	El caso de uso se inicia cuando el Sistema a una hora determinada comienza a obtener las Conciliaciones de Tarjeta Creada, Tarjeta Impresa y Solicitudes de Baja procesadas en el día y se las envía a los Web Services de los Bancos asociados.
Referencia	RF11

CU-10	CU_Enviar Conciliaciones Auxiliares
Actor	Web Service Banco, Administrador de BT
Descripción	El caso de uso se inicia cuando el Administrador de BT selecciona la opción Conciliaciones Auxiliares. El Administrador de BT selecciona el tipo de conciliación que desea enviar, la fecha de la conciliación y el banco asociado y el Sistema permite obtener la conciliación y enviársela al Web Service del Banco seleccionado.

Referencia	RF11
------------	------

CU-11	CU_Realizar Reporte
Actor	Operadora de Autenticación
Descripción	El caso de uso se inicia cuando la Operadora de Autenticación selecciona la opción Realizar Reporte. El sistema muestra automáticamente el reporte según los criterios seleccionados por la Operadora de Autenticación. El reporte puede imprimirse en caso que así lo desee la Operadora de Autenticación.
Referencia	RF12

2.9.3 Diagrama de casos de uso

Un diagrama de casos de uso del sistema es la representación de un conjunto de casos de uso para un sistema, los actores y la relación entre ambos.

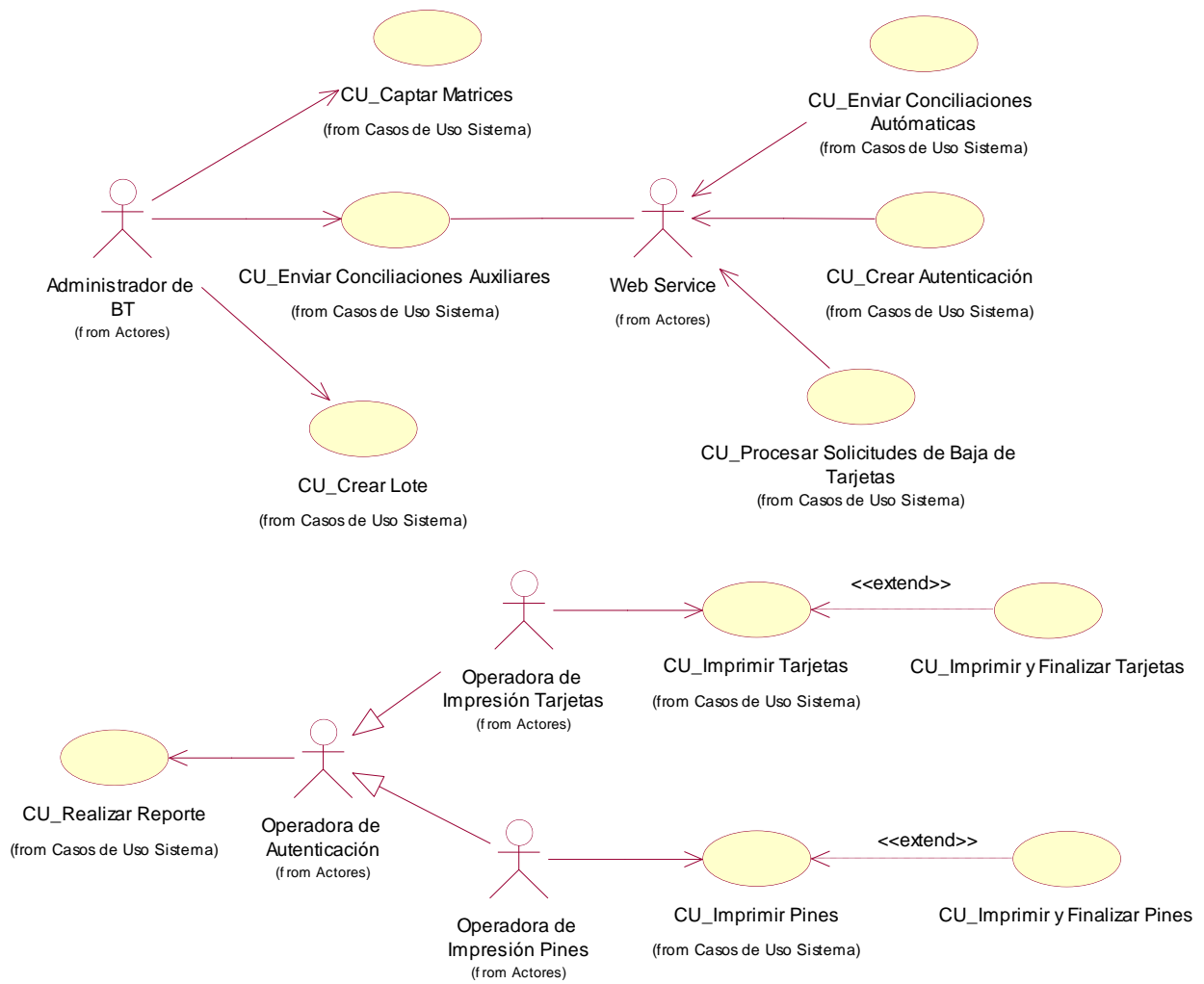


Figura 2.2 Diagrama de Casos de uso del Sistema.

2.9.4 Casos de uso expandidos

2.9.4.1 Caso de uso Crear Autenticación

CU-1	CU_ Crear Autenticación.
------	--------------------------

Propósito	El caso de uso CU_ Crear Autenticación tiene como objetivo crear las autenticaciones de los clientes de TeleBanca, o sea crear una tarjeta con matriz y pin para cada cliente.	
Actores: Web Service Banco		
Resumen: El caso de uso se inicia cuando el sistema automáticamente se ejecuta y comienza la generación de autenticaciones. Obtiene todas las solicitudes de generación de autenticaciones de los bancos con los datos del cliente en caso de que existan solicitudes y genera dichas autenticaciones. El sistema le pide a los bancos la conciliación de de las sucursales que estos tienen y guarda las tarjetas creadas junto con una notificación.		
Referencias	RF1, RF2	
Acción del actor	Respuesta del sistema	
	<p>1. El sistema automáticamente obtiene la lista de todos los Bancos asociados, el nombre, número, la dirección del Web Service y la contraseña de cada uno.</p> <p>El Sistema trata a cada uno de los Bancos asociados. Ver Sección Gestión de Bancos.</p>	
	<p>2. El Sistema guarda en las notificaciones el mensaje: "Se procesaron X solicitudes de Autenticación."</p>	
Sección Gestión de Bancos		
Acción del actor	Respuesta del sistema	
	<p>3. El Sistema obtiene las matrices disponibles en la Base de Datos y las guarda en un arreglo de matrices.</p>	
	<p>4. El sistema obtiene los identificadores de las tarjetas existentes en la base de datos.</p>	
	<p>5. El sistema le pide al Web Service las solicitudes de autenticación.</p>	

6. El Web Service envía las solicitudes de autenticación.	
	7. El sistema verifica que no se repitan los identificadores de las nuevas tarjetas a crear con las que ya existen en la base de datos.
	8. El sistema para cada nueva solicitud verifica que queden matrices disponibles.
	9. El sistema convierte a cada solicitud en una tarjeta: a cada una le pone como la fecha de orden de impresión de la documentación la fecha del sistema. Le asigna a la tarjeta una matriz cifrada y mediante un algoritmo de asociación aleatoria le crea un PIN. Pone la tarjeta en estado “creada” y el PIN en estado “creado” y cambia en el arreglo de matrices el estado de cada matriz a “ocupada”.
	10. El sistema verifica que se hayan procesado solicitudes de autenticación para el banco.
	11. El sistema le pide al Web Service del banco la relación de sus sucursales.
12. El Web Service envía la conciliación de las sucursales que tiene el banco.	
	13. El sistema actualiza los datos de las sucursales en la base de datos.
	14. El Sistema guarda las nuevas tarjetas creadas en la base de datos.
	15. El sistema actualiza el nuevo estado de las matrices utilizadas en la base de datos.
	16. El sistema incrementa la cantidad de solicitudes procesadas.

Flujos alternativos	
Acción del actor	Respuesta del sistema
Acción 1 F1	1 F1.1 El Sistema guarda en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada, no se pueden obtener los bancos asociados por estar la conexión con la Base de Datos deshabilitada. Se procesaron 0 solicitudes de autenticación".
	1 F1.2 Se termina el caso de uso.
Acción 1 F1.1	1 F1.1.1 El Sistema guarda en el fichero Notificaciones.xml el mensaje: "Solicitud de Autenticaciones cancelada, no se pueden obtener los bancos asociados por estar la conexión con la Base de Datos deshabilitada. Se procesaron 0 solicitudes de autenticación".
	1 F1.1.2 Se termina el caso de uso.
Acción 1 F2	1 F2.1 El Sistema guarda en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada, no existen bancos asociados en la BD. Se procesaron 0 solicitudes de autenticación".
	1 F2.2 Se termina el caso de uso.
Acción 1 F2.1	1 F2.1.1 El Sistema guarda en el fichero Notificaciones.xml el mensaje: "Solicitud de Autenticaciones cancelada, no existen bancos asociados en la BD. Se procesaron 0 solicitudes de autenticación".
	1 F2.1.2 Se termina el caso de uso.

Acción 2	2.1 El Sistema guarda en el fichero Notificaciones.xml el mensaje: "Se procesaron X solicitudes de Autenticación."
	2.2 Se termina el caso de uso.
Acción 3 F1	3 F1.1 El Sistema guarda en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada, no se pueden obtener matrices, no hay conexión con la BD. Se procesaron X solicitudes de autenticación".
	3 F1.2 Se termina el caso de uso.
Acción 3 F1.1	3 F1.1.1 El Sistema guarda en el fichero Notificaciones.xml el mensaje: "Solicitud de Autenticaciones cancelada, no se pueden obtener matrices, no hay conexión con la BD. Se procesaron X solicitudes de autenticación".
	3 F1.1.2 Se termina el caso de uso.
Acción 3 F2	3 F2.1 El Sistema guarda en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada, no hay matrices disponibles en la BD. Se procesaron X solicitudes de autenticación".
	3 F2.2 Se termina el caso de uso.
Acción 3 F2.1	3 F2.1.1 El Sistema guarda en el fichero Notificaciones.xml el mensaje: "Solicitud de Autenticaciones cancelada, no hay matrices disponibles en la BD. Se procesaron X solicitudes de autenticación".
	3 F2.1.2 Se termina el caso de uso.

Acción 4	4.1 El Sistema guarda en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada, no se puede obtener los id de las tarjetas existentes, no hay conexión con la base de datos. Se procesaron X solicitudes de autenticación".
	4.2 Se termina el caso de uso.
Acción 4.1	4.1.1 El Sistema guarda en el fichero Notificaciones.xml el mensaje: "Solicitud de Autenticaciones cancelada, no se puede obtener los id de las tarjetas existentes, no hay conexión con la base de datos. Se procesaron X solicitudes de autenticación".
	4.1.2 Se termina el caso de uso.
Acción 5	5.1 El Sistema guarda en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada, no existe conexión con el Web Service Y del banco XX. Se procesaron X solicitudes de autenticación".
	5.2 Se termina el caso de uso.
Acción 5.1	5.1.1 El Sistema guarda en el fichero Notificaciones.xml el mensaje: "Solicitud de Autenticaciones cancelada, no existe conexión con el Web Service X del banco XX. Se procesaron X solicitudes de autenticación".
	5.1.2 Se termina el caso de uso.
Acción 6	6.1 El Sistema guarda en la tabla Notificación el mensaje: "No existen nuevas solicitudes del banco XX".
	6.2 Se pasa a la acción 3 del caso de uso para procesar el próximo banco.

Acción 6.1	6.1.1 El Sistema guarda en el fichero Notificaciones.xml el mensaje: "Solicitud de Autenticaciones cancelada, no hay Conexión con la ase de datos. Se procesaron X solicitudes de autenticación".
	6.1.2 Se termina el caso de uso.
Acción 6.2	6.2.1 Se pasa a la acción 2 del caso de uso.
Acción 7	7.1 Se elimina esa solicitud de la lista de solicitudes.
	7.2 Verifica que queden solicitudes en la lista de solicitudes a procesar.
	7.3 Se pasa a la acción 7 para verificar el próximo identificador de tarjeta.
Acción 7.2	7.2.1 Se pasa a la acción 10 del caso de uso.
Acción 7.3	7.3.1 Se pasa a la acción 8 del caso de uso.
Acción 8	8.1 Se realizan las acciones 10, 11, 11, 12, 13, 14, 15 y 16 del caso de uso.
	8.2 El Sistema guarda en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada, matrices agotadas en BD. Se procesaron X solicitudes de autenticación".
	8.3 Se termina el caso de uso.
Acción 8.2	8.2.1 El Sistema guarda en el fichero Notificaciones.xml el mensaje: "Solicitud de Autenticaciones cancelada, matrices agotadas en BD. Se procesaron X solicitudes de autenticación".
	8.2.2 Se termina el caso de uso.
Acción 10	10.1 Se pasa a la acción 3 del caso de uso para procesar el próximo banco.
Acción 10.1	10.1.1 Se pasa a la acción 2 del caso de uso.

Acción 11	11.1 El Sistema guarda en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada. Error en la conexión con el Web Services. Se procesaron X solicitudes de autenticación".
	11.2 Se termina el caso de uso.
Acción 11.1	11.1.1 El Sistema guarda en el fichero Notificaciones.xml el mensaje: "Solicitud de Autenticaciones cancelada. Error en la conexión con el Web Service. Se procesaron X solicitudes de autenticación".
	11.1.2 Se termina el caso de uso.
Acción 13	13.1 El Sistema guarda en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada. Error en la conexión con la Base de Datos. Se procesaron X solicitudes de autenticación".
	13.2 Se termina el caso de uso.
Acción 13.1	13.1.1 El Sistema guarda en el fichero Notificaciones.xml el mensaje: "Solicitud de Autenticaciones cancelada. Error en la conexión con la Base de Datos. Se procesaron X solicitudes de autenticación".
	13.1.2 Se termina el caso de uso.
Acción 14	14.1 El Sistema guarda en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada. Error al guardar tarjetas o actualizar matrices del banco X, tarjetas no guardadas. Se procesaron X solicitudes de autenticación".
	14.2 Se termina el caso de uso.

Acción 14.1	14.1.1 El Sistema guarda en el fichero Notificaciones.xml el mensaje: “Solicitud de Autenticaciones cancelada. Error al guardar tarjetas o actualizar matrices del banco X, tarjetas no guardadas. Se procesaron X solicitudes de autenticación”.
	14.1.2 Se termina el caso de uso.
Acción 15	15.1 El Sistema guarda en la tabla Notificación el mensaje: “Solicitud de Autenticaciones cancelada. Error al guardar tarjetas o actualizar matrices del banco X, matrices no actualizadas. Se procesaron X solicitudes de autenticación”.
	15.2 Se termina el caso de uso.
Acción 15.1	15.1.1 El Sistema guarda en el fichero Notificaciones.xml el mensaje: “Solicitud de Autenticaciones cancelada. Error al guardar tarjetas o actualizar matrices del banco X, matrices no actualizadas. Se procesaron X solicitudes de autenticación”.
	15.1.2 Se termina el caso de uso.

Flujos alternativos

Acción 1 F1: Si el Sistema no puede obtener los datos de los bancos a los que debe conectarse por estar la conexión con la Base de datos deshabilitada.

Acción 1 F1.1: Si el Sistema no puede guardar en la tabla Notificación el mensaje: “Solicitud de Autenticaciones cancelada, no se pueden obtener los bancos asociados por estar la conexión con la Base de Datos deshabilitada. Se procesaron 0 solicitudes de autenticación”.

Acción 1 F2: Si no hay Bancos asociados.

Acción 1 F2.1: Si el Sistema no puede guardar en la tabla Notificación el mensaje: “Solicitud de Autenticaciones cancelada, no existen bancos asociados en la BD. Se procesaron 0 solicitudes de autenticación”.

Acción 2: Si el Sistema no puede guardar en las notificaciones el mensaje: "Se procesaron X solicitudes de Autenticación.", por estar la conexión con la Base de Datos deshabilitada.

Acción 3 F1: Si el Sistema no puede obtener las matrices disponibles en la Base de Datos por estar la conexión deshabilitada.

Acción 3 F1.1: Si el Sistema no puede guardar en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada, no se pueden obtener matrices, no hay conexión con la BD. Se procesaron X solicitudes de autenticación".

Acción 3 F2: Si no hay matrices disponibles.

Acción 3 F2.1: Si el Sistema no puede guardar en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada, no hay matrices disponibles en la BD. Se procesaron X solicitudes de autenticación".

Acción 4: Si el Sistema no puede obtener los identificadores de las tarjetas de la base de datos por estar la conexión deshabilitada.

Acción 4.1: Si el Sistema no puede guardar en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada, no se puede obtener los id de las tarjetas existentes, no hay conexión con la base de datos. Se procesaron X solicitudes de autenticación".

Acción 5: Si no hay conexión con el Web Services para pedirle las solicitudes de creación de autenticación.

Acción 5.1: Si el Sistema no puede guardar en la tabla Notificación el mensaje: "Solicitud de Autenticaciones cancelada, no existe conexión con el Web Service X del banco XX. Se procesaron X solicitudes de autenticación".

Acción 6: Si el Web Service no envía solicitudes.

Acción 6.1: Si el Sistema no tiene conexión con la base de datos para guardar en la tabla Notificación el mensaje: "No existen nuevas solicitudes del banco XX".

Acción 6.2: Si no hay más bancos que procesar.

Acción 7: Si el Sistema verifica que la tarjeta ya existe.

Acción 7.2: Si el sistema verifica que quedan no existen solicitudes en la lista de solicitudes.

Acción 7.3: Si no existen más identificadores de tarjetas que procesar.

Acción 8: Si el sistema verifica que se agotaron las matrices disponibles.

Acción 8.2: Si el Sistema no puede guardar en la tabla Notificación el mensaje: “Solicitud de Autenticaciones cancelada, matrices agotadas en BD. Se procesaron X solicitudes de autenticación”.

Acción 10: Si el sistema verifica que no se procesaron solicitudes de autenticación.

Acción 10.1: Si el sistema verifica que no se procesaron solicitudes de autenticación y no hay más bancos que procesar.

Acción 11: Si no hay conexión con el Web Service.

Acción 11.1: Si el Sistema no puede guardar en la tabla Notificación el mensaje: “Solicitud de Autenticaciones cancelada, Error al guardar tarjetas o actualizar matrices del banco X, no hay conexión con la base de datos. Se procesaron X solicitudes de autenticación”.

Acción 13: Si el Sistema no puede guardar las sucursales en la Base de Datos por estar la conexión deshabilitada.

Acción 13.1: Si el sistema no puede guardar en la tabla Notificación el mensaje: “Solicitud de Autenticaciones cancelada. Error en la conexión con la Base de Datos. Se procesaron X solicitudes de autenticación”.

Acción 14: Si el Sistema no puede guardar las nuevas tarjetas creadas en la Base de Datos por estar la conexión deshabilitada.

Acción 14.1: Si el Sistema no puede guardar en la tabla Notificación el mensaje: “Solicitud de Autenticaciones cancelada. Error al guardar tarjetas o actualizar matrices del banco X, tarjetas no guardadas. Se procesaron X solicitudes de autenticación”.

Acción 15: Si el Sistema no puede actualizar el nuevo estado de las matrices en la Base de Datos por estar la conexión deshabilitada.

Acción 15.1: Si el Sistema no puede guardar en la tabla Notificación el mensaje: “Solicitud de Autenticaciones cancelada. Error al guardar tarjetas o actualizar matrices del banco X, matrices no actualizadas. Se procesaron X solicitudes de autenticación”.

2.9.4.2 Caso de uso Crear Lote. Ver Anexo I

2.9.4.3 Caso de uso Imprimir Tarjetas

CU-3	CU_Imprimir Tarjetas
------	----------------------

Propósito	El caso de uso Imprimir Tarjetas es el que permite a la Operadora de Impresión de Tarjetas imprimir las tarjetas que han sido creadas al generarse las autenticaciones.	
Actores	Operadora de Impresión de Tarjetas	
Resumen:	El caso de uso se inicia cuando la Operadora de Impresión de Tarjetas elige la opción Imprimir Tarjetas. El sistema imprime el lote de tarjetas seleccionado por la Operadora de Impresión de Tarjetas.	
Referencias	RF5	
	Acción del actor	Respuesta del sistema
	1. El caso de uso se inicia cuando la Operadora de Impresión de Tarjetas selecciona la opción “Imprimir Tarjetas”.	2. El Sistema obtiene los identificadores de los lotes con estado Tarjeta “creado” y la cantidad de tarjetas a imprimir en cada uno de ellos.
		3. El Sistema muestra los identificadores de los lotes con estado Tarjeta “creado” y la cantidad de Tarjetas a imprimir en cada uno de ellos y activa la opción “Cancelar”.
	4. La Operadora de Impresión de Tarjetas selecciona el o los lotes que desea imprimir.	5. El Sistema muestra la cantidad de tarjetas seleccionadas a imprimir y activa la opción de “Imprimir”.
	6. La Operadora de Impresión de Tarjetas selecciona la opción “Imprimir”.	7. El Sistema muestra el mensaje: “¿Está seguro que desea imprimir?”.
	8. La Operadora de Impresión de Tarjetas acepta la opción “Aceptar”.	9. El Sistema obtiene los datos correspondientes a la fecha y hora de impresión y el identificador del usuario de la Operadora de Impresión de Tarjetas.
		10. El Sistema obtiene en las tarjetas que corresponden al o los lote(s) seleccionado(s) el número de tarjeta a imprimir y la matriz cifrada mediante el identificador localizador de su posición y la descifra.

	11. El Sistema le manda los datos de las Tarjetas a imprimir a la impresora.
	12. El Sistema imprime automáticamente las tarjetas.
	13. El Sistema actualiza estos datos en el o los lote(s) seleccionado(s), cambiándoles el estado tarjeta del o los lote(s) a "Impreso".
	14. Ver caso de uso Reimprimir y Finalizar Tarjeta Acción 2.
Flujos alternativos	
Acción del actor	Respuesta del sistema
Acción 2 F1	2 F1.1 El Sistema muestra el mensaje: "No se han encontrado Lotes pendientes de impresión de Tarjetas."
2 F1.2 La Operadora de Impresión de Tarjetas acepta el mensaje.	2 F1.3 Se termina el caso de uso.
Acción 2 F2	2F2.1 El sistema muestra el mensaje "Error en la conexión con la base de datos."
2 F2.2 La Operadora de Impresión de Tarjetas acepta el mensaje.	2 F2.3 Se termina el caso de uso.
Acción 4	4.1 Se termina el caso de uso.
Acción 6	6.1 Se termina el caso de uso.
Acción 8	8.1 Pasar a la Acción 3.
Acción 10	10.1 El sistema muestra el mensaje: "Error en la conexión con la base de datos."
10.2 La Operadora de Impresión de Tarjetas acepta el mensaje.	10.3 Se termina el caso de uso.
Acción 11	11.1 Pasar a la Acción 5.
Acción 13	13.1 El sistema muestra el mensaje: "Error en la conexión con la base de datos."

13.2 La Operadora de Impresión de Tarjetas acepta el mensaje.	13.3 Se termina el caso de uso.
Puntos de extensión	
CU_Reimprimir y Finalizar Tarjetas.	

Flujos alternativos

Acción 2 F1: Si el Sistema no encuentra lotes con estado tarjeta “creado”.

Acción 2 F2: Si el Sistema no puede obtener los identificadores de los lotes con estado tarjeta “creado” por estar la conexión con la Base de datos deshabilitada.

Acción 4: Si la Operadora de Impresión de Tarjetas selecciona la opción “Cancelar”.

Acción 6: Si la Operadora de Impresión de Tarjetas selecciona la opción “Cancelar”.

Acción 8: Si la Operadora de Impresión de Tarjetas selecciona la opción “Cancelar”.

Acción 10: Si el Sistema no puede obtener los datos para imprimir las tarjetas por estar la conexión con la Base de datos deshabilitada.

Acción 11: Si el Sistema no encuentra la impresora para mandarle los datos a imprimir.

Acción 13: Si el Sistema no puede cambiar el estado tarjeta del lote o los lotes a “Impreso” por estar la conexión con la Base de datos deshabilitada.

2.9.4.4 Caso de uso Imprimir Pines

CU-4	CU_Imprimir Pines	
Propósito	El caso de uso Imprimir Pines es el que permite a la Operadora de Impresión de Pines imprimir los pines que han sido creados al generarse las autenticaciones.	
Actores	Operadora de Impresión de Pines	
Resumen:	El caso de uso se inicia cuando la Operadora de Impresión de Pines elige la opción Imprimir Pines. El sistema imprime el lote seleccionado por la Operadora de Impresión de Pines.	
Referencias	RF7	
	Acción del actor	Respuesta del sistema
	1. El caso de uso se inicia cuando la Operadora de Impresión de Pines selecciona la opción “Imprimir Pines”.	2. El Sistema obtiene los identificadores de los lotes con estado pin “creado” y la cantidad de pines a imprimir en cada uno de ellos.

	3. El Sistema muestra los identificadores de los lotes con estado pin "creado" y la cantidad de pines a imprimir en cada uno de ellos y la opción "Cancelar".
4. La Operadora de Impresión de Pines selecciona el o los lotes que desea imprimir.	5. El Sistema muestra la cantidad de pines seleccionados a imprimir y activa la opción de "Imprimir".
6. La Operadora de Impresión de Pines selecciona la opción "Imprimir".	7. El Sistema muestra el mensaje "¿Está seguro que desea Imprimir?".
8. La Operadora de Impresión de Pines selecciona la opción "Aceptar".	9. El Sistema obtiene los datos correspondientes a la fecha, hora de impresión y el identificador del usuario de la Operadora de Impresión de Pines.
	10. El Sistema obtiene en las tarjetas que corresponden al o los lote(s) seleccionado(s) el número de pin correspondiente, el número de la tarjeta y los datos del cliente referentes a nombre, primer apellido, segundo apellido, identificador del cliente (carnet de identidad o pasaporte), el país, sucursal y banco.
	11. El Sistema le manda los datos de los pines a imprimir a la impresora.
	12. El Sistema imprime automáticamente los pines.
	13. El Sistema actualiza estos datos en el o los lote(s) seleccionado(s), cambiándoles el estado pin del o los lote(s) a "Impreso".
	14. Ver caso de uso Reimprimir y Finalizar Pines Acción 2.
Flujos alternativos	

Acción del actor	Respuesta del sistema
Acción 2 F1	2 F1.1 El Sistema muestra el mensaje "No se han encontrado Lotes pendientes de impresión de PIN".
2 F1.2 La Operadora de Impresión de Pines acepta el mensaje.	2 F1.3 Se termina el caso de uso.
Acción 2 F2	2 F2.1 El Sistema muestra el mensaje "Error en la conexión con la base de datos.".
2 F2.2 La Operadora de Impresión de Pines acepta el mensaje.	2 F2.3 Se termina el caso de uso.
Acción 4	4.1 Se termina el caso de uso.
Acción 6	6.1 Se termina el caso de uso.
Acción 8	8.1 Pasar a la Acción 3.
Acción 10	10.1 El Sistema muestra el mensaje: "Error en la conexión con la base de datos.".
10.2 La Operadora de Impresión de Pines acepta el mensaje.	10.3 Se termina el caso de uso.
Acción 11	11.1 Pasar a la Acción 5.
Acción 13	13.1 Se termina el caso de uso.
Puntos de extensión	
CU_Reimprimir y Finalizar Pines	

Flujos alternativos

Acción 2 F1: Si el Sistema no encuentra lotes con estado pin "creado".

Acción 2 F2: Si el Sistema no puede obtener los identificadores de los lotes con estado pin "creado" por estar la conexión con la Base de datos deshabilitada.

Acción 4: Si la Operadora de Impresión de Pines selecciona la opción "Cancelar".

Acción 6: Si la Operadora de Impresión de Pines selecciona la opción "Cancelar".

Acción 8: Si la Operadora de Impresión de Pines selecciona la opción "Cancelar".

Acción 10: Si el Sistema no puede obtener los datos para imprimir los pines por estar la conexión con la Base de datos deshabilitada.

Acción 11: Si el Sistema no encuentra la impresora para mandarle los datos a imprimir.

Acción 13: Si el Sistema no puede cambiar el estado pin del lote o los lotes a “Impreso” por estar la conexión con la Base de datos deshabilitada.

2.9.4.5 Caso de uso Reimprimir y Finalizar Tarjetas

CU-5	CU_Reimprimir y Finalizar Tarjetas	
Propósito	El caso de uso Reimprimir y Finalizar Tarjetas es el que permite a la Operadora de Impresión de Tarjetas reimprimir o Finalizar lo lotes que se hayan impreso.	
Actores	Operadora de Impresión de Tarjetas	
Resumen:	El caso de uso se inicia cuando la Operadora de Impresión de Tarjetas elige la opción Reimprimir Tarjeta. El sistema permitirá la reimpresión o la finalización del lote seleccionado por la Operadora de Impresión de Tarjetas. Para el caso de la finalización se cambiará el estado Tarjeta para que no se puedan volver a imprimir las tarjetas que pertenecen a ese Lote.	
Referencias	RF6, RF9	
	Acción del actor	Respuesta del sistema
	1. El caso de uso se inicia cuando la Operadora de Impresión de Tarjetas selecciona la opción “Reimprimir Tarjetas”.	2. El Sistema muestra un formulario con los lotes que tengan el estado tarjeta “Impreso” y la opción “Marcar Todos”.
	3. La Operadora de Impresión de Tarjetas selecciona: a) El lote que desea reimprimir. Ver sección Reimprimir Tarjetas. b) El o los lote(s) que desea finalizar. Ver sección Finalizar Tarjeta.	
	Sección Reimprimir Tarjeta	
		4. El sistema activa la opción “Reimprimir” y

	“Finalizar”.
5. La Operadora de Impresión de Tarjeta selecciona la opción “Reimprimir”.	6. El Sistema obtiene en las tarjetas que pertenecen al lote a reimprimir el número de la tarjeta, nombre, primer apellido e identificador del cliente.
	7. El Sistema muestra por separado los datos, dando la posibilidad a la Operadora de Impresión de Tarjetas de seleccionar el rango de tarjetas a reimprimir o seleccionar la opción “Cancelar”.
8. La Operadora de Impresión de Tarjetas especifica la(s) tarjeta(s) que desea reimprimir y selecciona la opción “Aceptar”.	9. El Sistema muestra el mensaje: “¿Está segura que desea reimprimir?”.
10. La Operadora de Impresión de Tarjetas selecciona la opción “Aceptar”.	11. El Sistema obtiene de las tarjetas especificadas por la Operadora de Impresión de Tarjetas la matriz asociada mediante el identificador de matriz la descifra.
	12. El Sistema le manda los datos de las tarjetas a imprimir a la impresora.
	13. El Sistema imprime automáticamente las tarjetas.
	14. El Sistema guarda un registro de las acciones del usuario, donde copia que la operadora va a reimprimir, el identificador del usuario de la Operadora de Impresión de Tarjetas, la fecha y hora de reimpresión y el identificador de las tarjetas.
	15. Pasar a la Acción 2.
Sección Finalizar Tarjeta	
Acción del actor	Respuesta del sistema

	16. El sistema activa la opción "Finalizar".
17. La Operadora de Impresión de Tarjetas selecciona la opción "Finalizar".	18. El Sistema muestra el mensaje "Si finaliza no podrá volver a imprimir ninguna de las tarjetas que forman el lote o los lotes escogidos por usted. ¿Está seguro que desea finalizar? ".
19. La Operadora de Impresión de Tarjetas selecciona la opción "Aceptar".	20. El Sistema busca todos los lotes que tengan el estado tarjeta "Impreso" y lo cambia por "Finalizado" y busca los lotes que tengan el estado pin y el estado tarjeta "Finalizado" y en las tarjetas que pertenecen a estos lotes le cambia el estado "Creada" por "Activa" y le pone en la fecha de orden de impresión la fecha del sistema.
	21. El Sistema muestra el mensaje: "El proceso de impresión ha concluido satisfactoriamente".
22. La Operadora de Impresión de Tarjetas acepta el mensaje.	23. Se termina el caso de uso.
Flujos alternativos	
Acción del actor	Respuesta del sistema
Acción 2 F1	2 F1.1 El Sistema muestra el mensaje: "No quedan lotes de tarjetas por finalizar..."
2 F1.2 La Operadora de Impresión de Tarjetas acepta el mensaje.	2 F1.3 Se termina el caso de uso.
Acción 2 F2	2 F2.1 El Sistema muestra el mensaje: "Error en la conexión con la base de datos."
2 F2.2 La Operadora de Impresión de Tarjetas acepta el mensaje.	2 F2.3 Se termina el caso de uso.
Acción 6	6.1 El Sistema muestra el mensaje: "Error en la conexión con la base de datos."
6.2 La Operadora de Impresión de Tarjetas	6.3 Pasar a la Acción 2.

acepta el mensaje.	
Acción 8	8.1 Pasar a la Acción 2.
Acción 10	10.1 Pasar a la Acción 7.
Acción 11	11.1 El Sistema muestra el mensaje: "Error en la conexión con la base de datos.".
11.2 La Operadora de Impresión de Tarjeta acepta el mensaje.	11.3 Pasar a la Acción 2.
Acción 12	12.1 Pasar a la Acción 7.
Acción 14	14.1 El Sistema muestra el mensaje: "Error en la conexión con la base de datos".
14.2 La Operadora de Impresión de Tarjeta acepta el mensaje.	14.3 Pasar a la Acción 2.
Acción 19	19.1 Pasar a la Acción 2.
Acción 20	20.1 El Sistema muestra el mensaje: "Error en la conexión con la base de datos.".
20.2. La Operadora de Impresión de Tarjeta acepta el mensaje.	20.3 Pasar a la Acción 2.
Puntos de extensión	
No tiene.	

Flujos alternativos

Acción 2 F1: Si el sistema no encuentra los lotes que tengan el estado tarjeta "Impreso"

Acción 2 F2: Si el sistema no puede obtener los lotes que tengan el estado tarjeta "Impreso" por estar la conexión con la Base de datos deshabilitada.

Acción 6: Si el sistema no puede obtener de las tarjetas que pertenecen al lote a reimprimir el número de la tarjeta, nombre, primer apellido e identificador del cliente por estar la conexión con la Base de datos deshabilitada.

Acción 8: Si la Operadora de Impresión de Tarjetas selecciona la opción "Cancelar".

Acción 10: Si la Operadora de Impresión de Tarjeta selecciona la opción “Cancelar”.

Acción 11: Si el Sistema no puede obtener de todas las tarjetas seleccionadas los datos necesarios para imprimir los pines por estar la conexión con la Base de datos deshabilitada.

Acción 12: Si el Sistema no encuentra impresora para mandarle los datos a imprimir.

Acción 14: Si el Sistema no puede guardar un registro de las acciones del usuario por estar la conexión con la Base de datos deshabilitada.

Acción 19: Si la Operadora de Impresión de Tarjetas selecciona la opción “Cancelar”.

Acción 20: Si el Sistema no puede buscar todos los lotes que tengan el estado tarjeta “Impreso” y cambiarlo por “Finalizado” y buscar todos los lotes que tengan el estado pin y el estado tarjeta “Finalizado” y en las Tarjetas que pertenecen a estos lotes cambiar el estado “Creada” por “Activa” y poner en la fecha orden de impresión la fecha del sistema por estar la conexión con la Base de datos deshabilitada.

2.9.4.6 Caso de uso Reimprimir y Finalizar Pines

CU-6	CU_Reimprimir y Finalizar Pines	
Propósito	El caso de uso Reimprimir y Finalizar Pines es el que permite a la Operadora de Impresión de Pines reimprimir o finalizar los pines que se hayan impreso.	
Actores	Operadora de Impresión de Pines	
Resumen:	El caso de uso se inicia cuando la Operadora de Impresión de Pines elige la opción Reimprimir Pines. El sistema permitirá la reimpresión o la finalización del lote seleccionado por la Operadora de Impresión de Pines. Para el caso de la finalización se cambiará el estado Pin para que no se puedan volver a imprimir los Pin que pertenecen a ese Lote.	
Referencias	RF8, RF9	
	Acción del actor	Respuesta del sistema
	1. La Operadora de Impresión de Pines selecciona la opción “Reimprimir Pines”.	2. El Sistema muestra un formulario con los lotes que tengan el estado pin “Impreso” y la opción “Marcar Todos”.
	3. La Operadora de Impresión de Pines selecciona:	

<p>a) El lote que desea reimprimir. Ver sección Reimprimir Pines.</p> <p>b) El o los lote(s) que desea finalizar. Ver sección Finalizar Pines.</p>	
Sección Reimprimir Pines	
Acción del actor	Respuesta del sistema
	4. El sistema activa la opción “Reimprimir” y “Finalizar”.
5. La Operadora de Impresión de Pines selecciona la opción “Reimprimir”.	6. El Sistema obtiene en las tarjetas que pertenecen al lote a reimprimir el número de la tarjeta, nombre, primer apellido e identificador del cliente.
	7. El Sistema muestra por separado los datos, dándole la posibilidad a la Operadora de Impresión de Pines de seleccionar el rango de pines a reimprimir o seleccionar la opción “Cancelar”.
8. La Operadora de Impresión de Pines especifica el o los pines que desea reimprimir y selecciona la opción “Aceptar”.	9. El Sistema muestra el mensaje “¿Está seguro que desea reimprimir?”.
10. La Operadora de Impresión de Pines selecciona la opción “Aceptar”.	11. El Sistema obtiene de las tarjetas especificadas por la Operadora de Impresión de Pines el número de pin correspondiente, número de la tarjeta, nombre, apellidos e identificador del cliente (carné de identidad o pasaporte), el país, sucursal y banco.
	12. El Sistema le manda los datos de los pines a imprimir a la impresora.
	13. El Sistema imprime automáticamente los pines.

	14. El Sistema guarda un registro de las acciones del usuario, donde copia que la operadora va a reimprimir, el identificador del usuario de la Operadora de Impresión de Pines, la fecha y hora de reimpresión y el identificador de las tarjetas.
	15. Pasar a la Acción 2.
Sección Finalizar Pines	
Acción del actor	Respuesta del sistema
	16. El sistema activa la opción "Finalizar".
17. La Operadora de Impresión de Pines selecciona la opción "Finalizar".	18. El Sistema muestra el mensaje "Si finaliza no podrá volver a imprimir ninguno de los pines que forman el lote o los lotes escogidos por usted. ¿Está seguro que desea finalizar? "
19. La Operadora de Impresión de Pines selecciona la opción "Aceptar".	20. El Sistema busca todos los lotes que tengan el estado pin "Impreso" y lo cambia por "Finalizado" y busca los lotes que tengan el estado pin y el estado tarjeta "Finalizado" y en las Tarjetas que pertenecen a estos lotes le cambia el estado "Creada" por "Activa" y le pone en la fecha de orden de impresión la fecha del sistema.
	21. El Sistema muestra el mensaje: "El proceso de impresión ha concluido satisfactoriamente".
22. La Operadora de Impresión de Pines acepta el mensaje.	23. Se termina el caso de uso.
Flujos alternativos	
Acción del actor	Respuesta del sistema
Acción 2 F1	2 F1.1 El Sistema muestra el mensaje: "No quedan lotes de pines por finalizar...".

2 F1.2 La Operadora de Impresión de Pines acepta el mensaje.	2 F1.3 Se termina el caso de uso
Acción 2 F2	2 F2.1 El Sistema muestra el mensaje: "Error en la conexión con la base de datos.".
2 F2.2 La Operadora de Impresión de Pines acepta el mensaje.	2 F2.3 Se termina el caso de uso
Acción 6	6.1 El Sistema muestra el mensaje: "Error en la conexión con la base de datos.".
6.2 La Operadora de Impresión de Pines acepta el mensaje.	6.3 Pasar a la Acción 2.
Acción 8	8.1 Pasar a la Acción 2.
Acción 10	10.1 Pasar a la Acción 7.
Acción 11	11.1 El Sistema muestra el mensaje: "Error en la conexión con la base de datos.".
11.2 La Operadora de Impresión de Pines acepta el mensaje.	11.3 Pasar a la Acción 2.
Acción 12	12.1 Pasar a la Acción 7.
Acción 14	14.1 El Sistema muestra el mensaje: "Error en la conexión con la base de datos.".
14.2 La Operadora de Impresión de Pines acepta el mensaje.	14.3 Pasar a la Acción 2.
Acción 19	19.1 Pasar a la Acción 2.
Acción 20	20.1 El Sistema muestra el mensaje: "Error en la conexión con la base de datos.".
20.2. La Operadora de Impresión de Pines acepta el mensaje.	20.3 Pasar a la Acción 2.

Flujos alternativos

Acción 2 F1: Si el sistema no encuentra los lotes que tengan el estado pin "Impreso".

Acción 2 F2: Si el sistema no encuentra los lotes que tengan el estado pin “Impreso” por estar la conexión con la Base de datos deshabilitada.

Acción 6: Si el sistema no puede obtener de las tarjetas que pertenecen al lote a reimprimir el número de la tarjeta, nombre, primer apellido e identificador del cliente por estar la conexión con la Base de datos deshabilitada.

Acción 8: Si la Operadora de Impresión de Pines selecciona la opción “Cancelar”.

Acción 10: Si la Operadora de Impresión de Pines selecciona la opción “Cancelar”.

Acción 11: Si el Sistema no puede obtener de todas las tarjetas seleccionadas los datos necesarios para imprimir los pines por estar la conexión con la Base de datos deshabilitada.

Acción 12: Si el Sistema no encuentra la impresora para mandar los datos a imprimir.

Acción 14: Si el Sistema no puede guardar un registro de las acciones del usuario por estar la conexión con la Base de datos deshabilitada.

Acción 19: Si la Operadora de Impresión de Pines selecciona la opción “Cancelar”.

Acción 20: Si el Sistema no puede buscar todos los lotes que tengan el estado pin “Impreso” y cambiarlo por “Finalizado” y buscar todos los lotes que tengan el estado pin y el estado tarjeta “Finalizado” y en las tarjetas que pertenecen a estos lotes cambiar el estado “Creada” por “Activa” y poner en la fecha orden de impresión la fecha del sistema por estar la conexión con la Base de datos deshabilitada.

2.9.4.7 Caso de uso Captar Matrices. Ver Anexo I

2.9.4.8 Caso de uso Procesar Solicitudes de Baja de Tarjetas. Ver Anexo I

2.9.4.9 Caso de uso Enviar Conciliaciones Automáticas. Ver Anexo I

2.9.4.10 Caso de uso Enviar Conciliaciones Auxiliares. Ver Anexo I

2.9.4.11 Caso de uso Realizar Reporte. Ver Anexo I

2.10 Conclusiones

En este capítulo se comenzó modelar el desarrollo de la propuesta de solución, lo que permitió obtener a partir del análisis de los procesos del dominio, un listado con los requerimientos que debe tener el sistema. Estos se representaron mediante un Diagrama de Casos de Uso y se describieron paso a paso todas las acciones de los actores del sistema con los casos de uso con los que interactúan. Con lo ya

establecido se puede empezar con la construcción del sistema, tratando de que se cumplan todos los requerimientos y las funciones que han sido consideradas necesarias en este capítulo.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

En el presente capítulo se efectuará el modelado del análisis y del diseño del sistema los cuales son muy importantes en el desarrollo de software ya que constituye la vista lógica de la arquitectura. Por lo que se ajusta el resultado de estos modelos a las tecnologías y lenguajes que serán utilizados. A través de esta fase, se toman decisiones estratégicas y tácticas para cumplir los requerimientos funcionales y de calidad de un sistema.

3.2 Análisis

El Análisis abarca las abstracciones primarias de clases y objetos, así como los mecanismos presentes en el dominio del problema, o sea del mundo real, y no se consideran clases que definen detalles y soluciones en el sistema de software. Las clases que se modelan son identificadas, con sus relaciones y descritas en un diagrama de clases.

3.2.1 Diagrama de clases del análisis

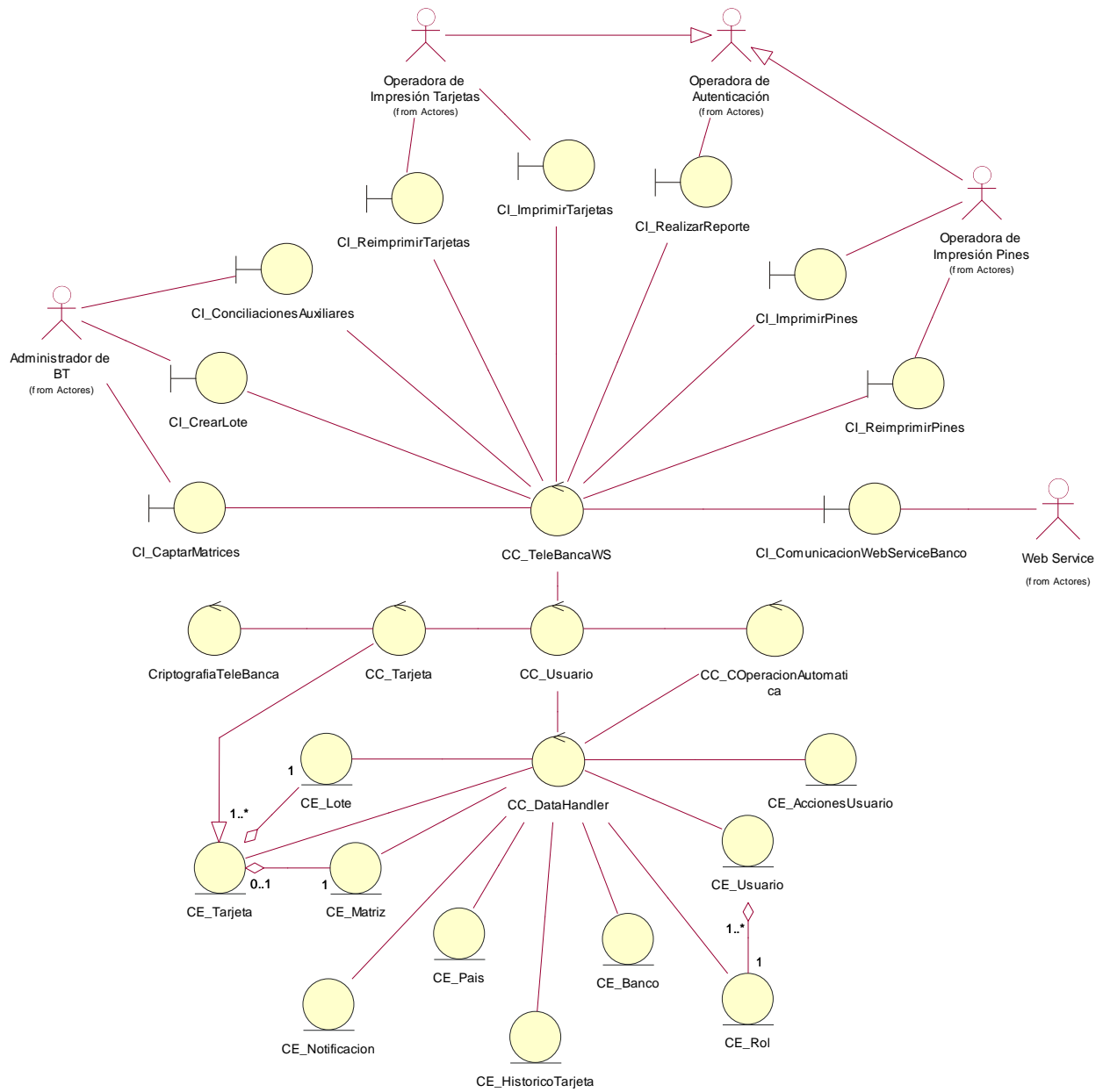


Figura 3.1 Diagrama de clases del análisis.

3.3 Diseño

En el diseño el resultado del análisis es expandido a una solución técnica ya que se agregan nuevas clases que proveen de la infraestructura técnica: interfaces de usuario, manejo de bases de datos para almacenar objetos en una base de datos, comunicaciones con otros sistemas, etc. El diseño resulta en especificaciones detalladas para la fase de programación.

3.3.1 Diagramas de secuencia

Los diagramas de secuencia forman parte del modelado dinámico del sistema y proporcionan una vista detallada de los casos de uso. Son diagramas que muestran la interacción organizada de objetos, mediante mensajes que se envían entre sí, en una secuencia de tiempo. Son útiles para observar la vida de los objetos en un sistema, identificar llamadas a realizar o posibles errores del modelado estático que imposibiliten el flujo de información.

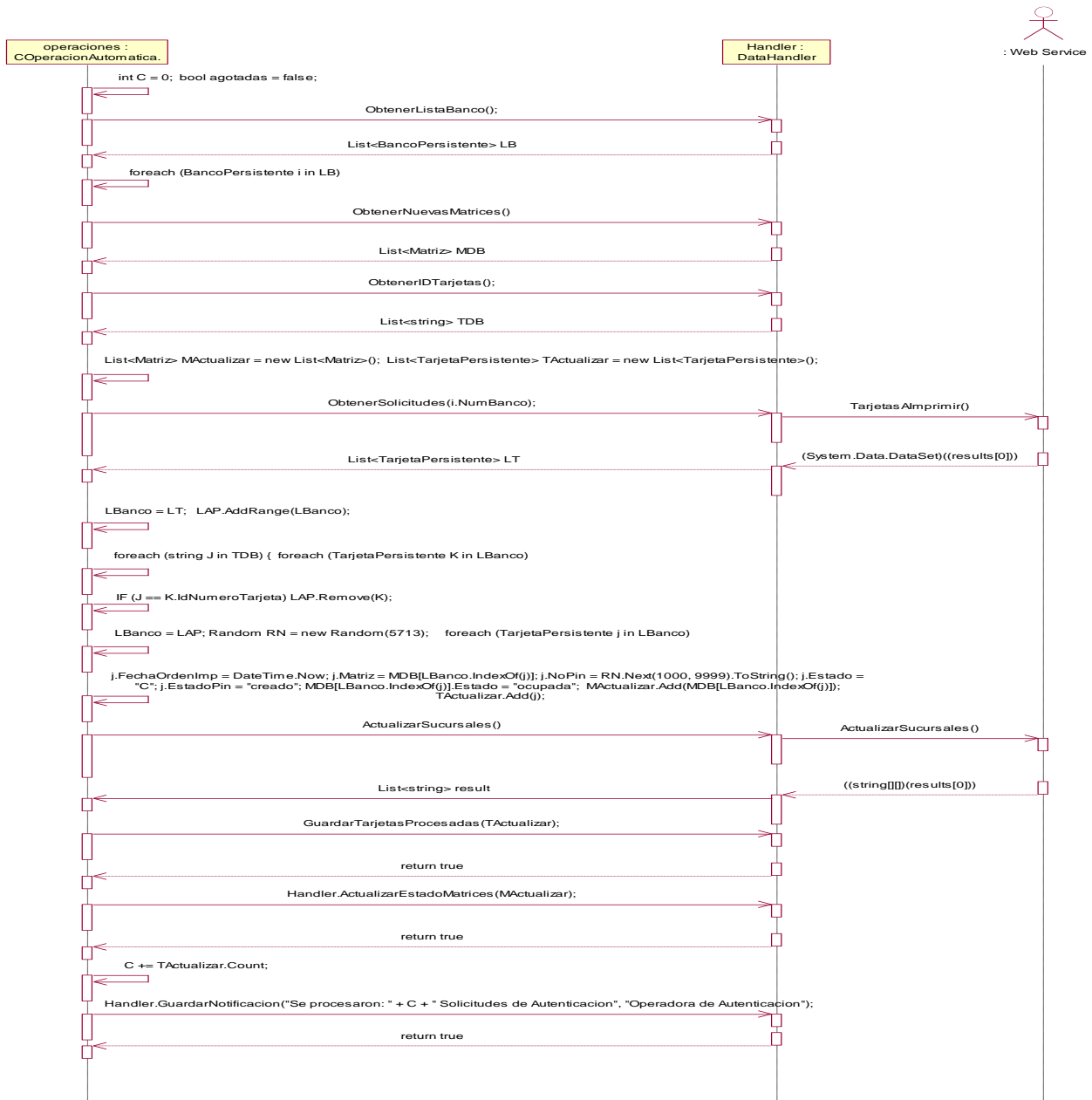


Figura 3.2 Diagrama de secuencia caso de uso CU_Crear Autenticación.

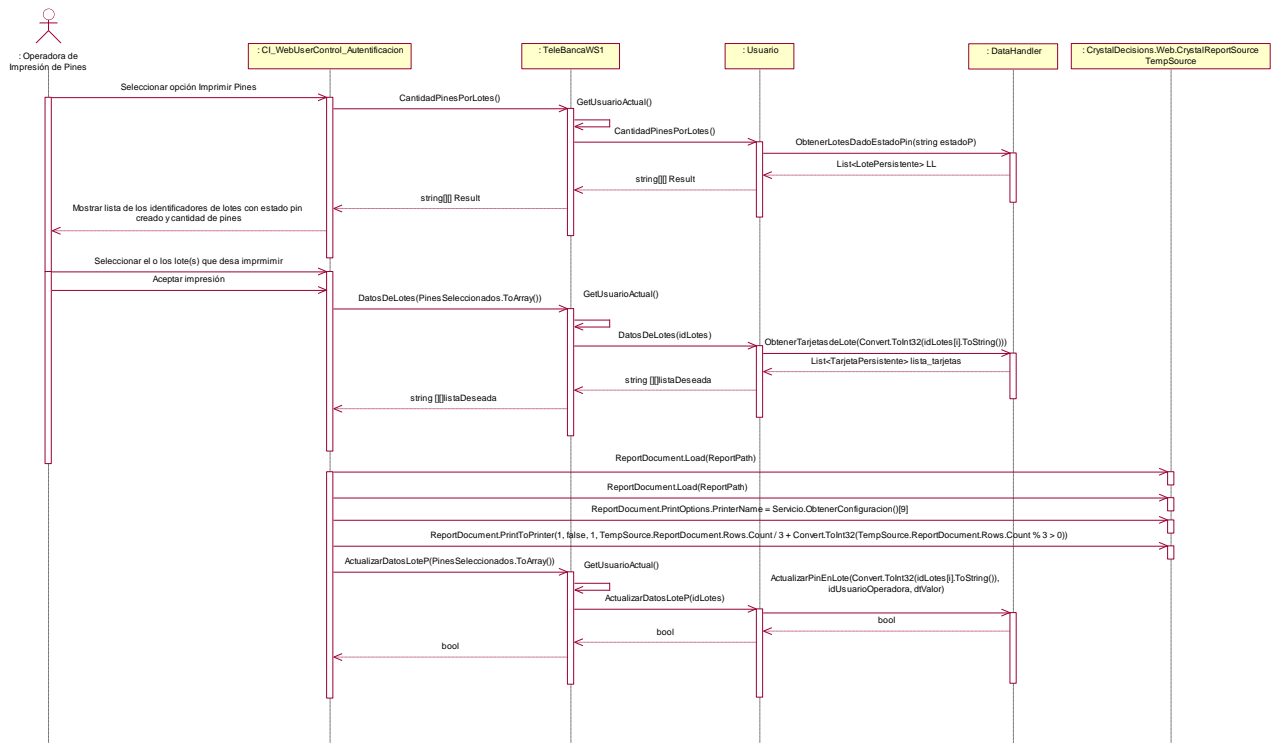


Figura 3.4 Diagrama de Secuencia caso de uso CU_Imprimir Pines.

- 3.3.1.1 Diagrama de secuencia caso de uso CU_Crear Lote. Ver Anexo II
- 3.3.1.2 Diagrama de secuencia caso de uso CU_Reimprimir y Finalizar Tarjetas. Ver Anexo II
- 3.3.1.3 Diagrama de secuencia caso de uso CU_ Reimprimir y Finalizar Pines. Ver Anexo II
- 3.3.1.4 Diagrama de secuencia caso de uso CU_Captar Matrices. Ver Anexo II
- 3.3.1.5 Diagrama de secuencia caso de uso CU_Procesar Solicitudes de Baja de Tarjetas. Ver Anexo II
- 3.3.1.6 Diagrama de secuencia caso de uso CU_Enviar Conciliaciones Automáticas. Ver Anexo II
- 3.3.1.7 Diagrama de secuencia caso de uso CU_Enviar Conciliaciones Auxiliares. Ver Anexo II
- 3.3.1.8 Diagrama de secuencia caso de uso CU_Realizar Reporte. Ver Anexo II

Para obtener un nivel correcto de abstracción y detalle que permita obtener un resultado final, es mejor modelar los artefactos del sistema, es decir, modelar las páginas, los enlaces entre ellas, todo el código que irá creando las páginas, así como el contenido dinámico de estas una vez que estén en el navegador

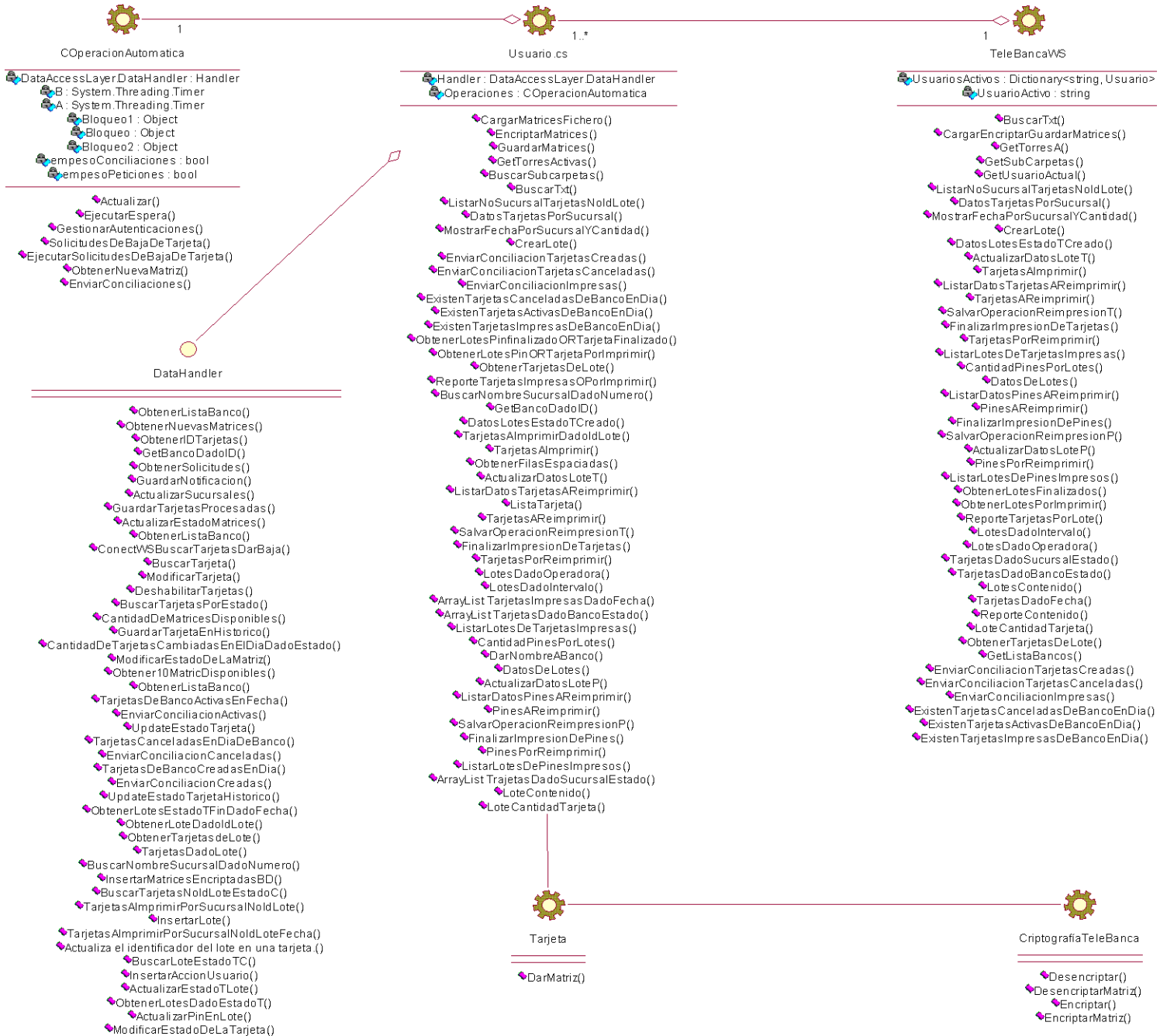


Figura 3.6 Diagrama de clases Web. Capa de Negocio.

3.3.3 Descripción de las clases

Nombre: TeleBancaWS	
Tipo de clase: Controladora	
Atributo	Tipo
UsuariosActivos	Dictionary<string, Usuario>
UsuarioActivo	String
Responsabilidades	
Nombre:	string[][] GetDataMenu(int pModulo)
Descripción:	Carga la interfaz del módulo.
Nombre:	Usuario GetUsuarioActual
Descripción:	Devuelve el usuario que está activo en el Sistema.
Caso de Uso: Realizar Reporte	
Nombre:	ArrayList ObtenerLotesFinalizados()
Descripción:	Verifica el tipo de operadora de impresión actual y llama al método de la clase Usuario para obtener los lotes finalizados.
Nombre:	Datoslote[] ObtenerLotesPorImprimir()
Descripción:	Verifica el tipo de operadora de impresión actual, llama al método de la clase Usuario para obtener los lotes pendientes de impresión y luego obtiene los datos referentes a nombre del banco, nombre de sucursal, nombre de la Operadora de Impresión de los Lotes Obtenidos.
Nombre:	DatosTarjetas[] ReporteTarjetasPorLote(string[] idlotes)
Descripción:	Dado un arreglo de identificadores de lote, llama al método de la clase Usuario que devuelve las tarjetas y los datos de las tarjetas que tengan un identificador de lote especificado, para obtener las tarjetas de cada identificador de lote y devolver un arreglo de datos de tarjetas.
Nombre:	LotePersistente[] LotesDadoIntervalo(DateTime fechaI, DateTime fechaF)
Descripción:	Llama al método de la clase Usuario que obtiene los lotes dado un intervalo de días.
Nombre:	ArrayList LotesDadoOperadora(string operadora)
Descripción:	Llama al método de la clase Usuario que obtiene los lotes dado la operadora que lo imprimió.
Nombre:	ArrayList TarjetasDadoSucursalEstado(string sucursal, string estado)
Descripción:	Llama al método de la clase Usuario que muestra las tarjetas que pertenecen a una sucursal y que tienen un estado especificado.
Nombre:	ArrayList TarjetasDadoBancoEstado(string banco, string estado)
Descripción:	Llama al método de la clase Usuario que muestra las tarjetas que pertenecen a un banco y que tienen un estado especificado.
Nombre:	ArrayList LotesContenido(int idLote)
Descripción:	Llama al método de la clase Usuario que dado un identificador de lote obtiene el contenido del lote.
Nombre:	ArrayList TarjetasDadoFecha(DateTime fecha)

Descripción:	Llama al método de la clase Usuario que dada una fecha, devuelve todas las tarjetas impresas en esa fecha.
Nombre:	ArrayList ReporteContenido(int idlote)
Descripción:	Llama al método de la clase Usuario que dado un identificador de lote muestra un reporte del lote.
Nombre:	int LoteCantidadTarjeta(int idlote)
Descripción:	Llama al método de la clase Usuario que dado el un identificador de lote obtiene la cantidad de tarjetas que pertenecen a ese lote.
Nombre:	string[] ObtenerTarjetasDeLote(int id_lote)
Descripción:	Llama al método de la clase Usuario que dado el un identificador de lote obtiene los números de las tarjetas que pertenecen a ese Lote.
Caso de Uso: Crear Lote	
Nombre:	ListarNoSucursalTarjetasNoldLote()
Descripción:	Llama al método de la clase Usuario que lista los datos de las sucursales que tienen tarjetas sin identificador de lote y en estado creada.
Nombre:	DatosTarjetasPorSucursal(string noSucursal)
Descripción:	Llama al método de la clase Usuario que muestra la cantidad de tarjetas con su sucursal correspondiente que están sin imprimir y sin identificador de lote.
Nombre:	MostrarFechaPorSucursalYCantidad(string noSucursal)
Descripción:	Llama al método de la clase Usuario que muestra las fechas que pertenecen a la sucursal seleccionada con su cantidad al lado.
Nombre:	CrearLote(string noSucursal, ArrayList fechasSeleccionadas)
Descripción:	Llama al método de la clase Usuario que crea el lote con las tarjetas a imprimir y actualizar dichos registros en la tabla Tarjeta.
Caso de Uso: Imprimir Tarjetas	
Nombre:	DatosLotesEstadoTcreado()
Descripción:	Llama al método de la clase Usuario que obtiene identificadores de lotes existentes con la cantidad de tarjetas de cada uno
Nombre:	ActualizarDatosLoteT(ArrayList idLotes)
Descripción:	Llama al método de la clase Usuario que actualiza datos en el registro del o los lote(s) que se seleccionaron para imprimir.
Nombre:	TarjetasAlImprimir(ArrayList idLotes)
Descripción:	Llama al método de la clase Usuario que obtiene los datos de las tarjetas que se van a imprimir dado el o los lote(s) seleccionados.
Nombre:	ListarDatosTarjetasAReimprimir(int idLote)
Descripción:	Llama al método de la clase Usuario que lista los números de tarjetas a reimprimir dado un identificador de lote con un índice al lado para usar a la hora de escoger que rango se quiere volver a imprimir.
Nombre:	TarjetasAReimprimir(int indiceInicio, int indiceFin, int IdLote)
Descripción:	Llama al método de la clase Usuario que reimprime el rango de tarjetas seleccionadas.
Nombre:	SalvarOperacionReimpresionT(int indiceInicio, int indiceFin, int IdLote)
Descripción:	Llama al método de la clase Usuario que salva la operación de reimprimir a la Tabla AccionUsuario.

Nombre:	FinalizarImpresionDeTarjetas(int ID_Lote)
Descripción:	Llama al método de la clase Usuario que actualiza estadoT dado el id del Lote, poniendo en estado finalizado "F".
Nombre:	TarjetasPorReimprimir()
Descripción:	Llama al método de la clase Usuario que devuelve verdadero si hay lotes con estado de tarjetas en impreso "I".
Nombre:	ListarLotesDeTarjetasImpresas()
Descripción:	Llama al método de la clase Usuario que lista los lotes con estado tarjeta en impreso "I".
Caso de Uso: Imprimir Pines	
Nombre:	CantidadPinesPorLotes()
Descripción:	Llama al método de la clase Usuario que obtiene los lotes y la cantidad de pines en cada uno de ellos.
Nombre:	DatosDeLotes(ArrayList idLotes)
Descripción:	Llama al método de la clase Usuario que obtiene los datos del o los lote(s) a imprimir.
Nombre:	ListarDatosPinesAREimprimir(int idLote)
Descripción:	Llama al método de la clase Usuario que muestra los datos de los pines que se quieren reimprimir.
Nombre:	PinesAREimprimir(int indiceInicio, int indiceFin, int IdLote)
Descripción:	Llama al método de la clase Usuario que permite escoger el intervalo de los pines que se quieren reimprimir y también manda los datos a imprimir.
Nombre:	FinalizarImpresionDePines(int ID_Lote)
Descripción:	Llama al método de la clase Usuario que permite finalizar la impresión de los pines imposibilitando que se vuelva a imprimir cualquiera de estos.
Nombre:	SalvarOperacionReimpresionP(int indiceInicio, int indiceFin, int IdLote)
Descripción:	Llama al método de la clase Usuario que permite guardar en la tabla AccionUsuario que la operadora reimprimió pines.
Nombre:	ActualizarDatosLoteP(ArrayList idLotes)
Descripción:	Llama al método de la clase Usuario que permite actualizar datos en la tabla Lote.
Nombre:	PinesPorReimprimir()
Descripción:	Llama al método de la clase Usuario que devuelve verdadero si hay lotes con estado de pines impreso "I".
Nombre:	ListarLotesDePinesImpresos()
Descripción:	Llama al método de la clase Usuario que devuelve los lotes que están en estado impreso "I".
Caso de Uso: Conciliaciones Auxiliares	
Nombre:	string[] GetListaBancos()
Descripción:	Llama al método de la clase DataHandler que obtiene una lista con los identificadores de todos los bancos.
Nombre:	string EnviarConciliacionTarjetasCreadas(DateTime fecha, string NombreBanco)
Descripción:	Llama al método de la clase Usuario que envía las conciliaciones de las Tarjetas Creadas.
Nombre:	string EnviarConciliacionTarjetasCanceladas(DateTime fecha, string

	NombreBanco)
Descripción:	Llama al método de la clase Usuario que envía las conciliaciones de las tarjetas Canceladas.
Nombre:	string EnviarConciliacionImpresas(DateTime fecha, string NombreBanco)
Descripción:	Llama al método de la clase Usuario que envía las conciliaciones de las Tarjetas Impresas.
Nombre:	bool ExistenTarjetasCanceladasDeBancoEnDia(DateTime f, string IDB)
Descripción:	Llama al método de la clase Usuario que verifica si existen tarjetas canceladas para un banco en una fecha dada.
Nombre:	bool ExistenTarjetasActivasDeBancoEnDia(DateTime f, string IDB)
Descripción:	Llama al método de la clase Usuario que verifica si existen tarjetas activas para un banco en una fecha dada.
Nombre:	bool ExistenTarjetasImpresasDeBancoEnDia(DateTime f, string IDB)
Descripción:	Llama al método de la clase Usuario que verifica si existen tarjetas impresas para un banco en una fecha dada.
Caso de Uso: Captar Matrices	
Nombre:	CargarEncriptarGuardarMatrices(string dir)
Descripción:	Carga un fichero de matrices, cifra las matrices y las guarda cifradas en la base de datos.
Nombre:	BuscarTxt(string path)
Descripción:	Devuelve los archivos de extensión .txt que existan en un path especificado por el usuario activo.
Nombre:	GetTorresA()
Descripción:	Obtiene el usuario activo que invoca al método que devuelve una tabla con los nombres de las unidades lógicas de la computadora, el tipo de unidad y especifica si esta está lista.
Nombre:	GetSubCarpetas(string p)
Descripción:	Devuelve las subcarpetas que hay en la dirección que selecciona el usuario activo.

Nombre: Usuario	
Tipo de clase: Controladora	
Atributo	Tipo
Handler	DataAccessLayer.DataHandler
Operaciones	COperacionAutomatica
Responsabilidades	
Caso de Uso: Conciliaciones Auxiliares	
Nombre:	string EnviarConciliacionTarjetasCreadas(DateTime f, string IDBanco)
Descripción:	Envía las conciliaciones de las tarjetas creadas para un banco en una fecha dada.
Nombre:	string EnviarConciliacionTarjetasCanceladas(DateTime f, string IDBanco)
Descripción:	Envía las conciliaciones de las tarjetas canceladas para un banco en una fecha dada.
Nombre:	string EnviarConciliacionImpresas(DateTime f, string IDBanco)

Descripción:	Envía las conciliaciones de las tarjetas impresas para un banco en una fecha dada.
Nombre:	bool ExistenTarjetasCanceladasDeBancoEnDia(DateTime f, string IDB)
Descripción:	Verifica si existen tarjetas canceladas para un banco en una fecha dada.
Nombre:	bool ExistenTarjetasActivasDeBancoEnDia(DateTime f, string IDB)
Descripción:	Verifica si existen tarjetas activas para un banco en una fecha dada.
Nombre:	bool ExistenTarjetasImpresasDeBancoEnDia(DateTime f, string IDB)
Descripción:	Verifica si existen tarjetas impresas para un banco en una fecha dada.
Caso de Uso: Realizar Reporte	
Nombre:	ArrayList TarjetasDadoBancoEstado(string nombrebanco, string estado)
Descripción:	Obtiene las tarjetas que pertenecen a un banco dado y con un estado específico.
Nombre:	ArrayList TrajetasDadoSucursalEstado(string nombresucursal, string estado)
Descripción:	Obtiene las tarjetas que pertenecen a una sucursal dada y con un estado específico.
Nombre:	ArrayList TarjetasImpresasDadoFecha(DateTime fecha)
Descripción:	Obtiene las tarjetas impresas en una fecha.
Nombre:	ArrayList ListaTarjeta()
Descripción:	Obtiene el nombre de las tarjetas según el identificador del lote.
Nombre:	ReporteContenido[] LoteContenido(int idLote)
Descripción:	Dado un identificador del lote muestra un reporte del lote.
Nombre:	LotePersistente[] LotesDadoOperadora(string operadora)
Descripción:	Obtiene los lotes dado un intervalo de días.
Nombre:	LotePersistente[] LotesDadoIntervalo(DateTime fechaI, DateTime fechaF)
Descripción:	Obtiene los lotes dado la operadora que lo imprimió.
Nombre:	int LoteCantidadTarjeta(int idlote)
Descripción:	Obtiene la cantidad de tarjetas que pertenecen a un lote.
Nombre:	List<LotePersistente> ObtenerLotesPinfinalizadoORTarjetaFinalizado(string funcionalidadOperadora)
Descripción:	Obtiene los lotes con el PIN y las tarjetas en estado finalizado.
Nombre:	List<LotePersistente> ObtenerLotesPinORTarjetaPorImprimir(string funcionalidadOperadora)
Descripción:	Obtiene los lotes con el PIN y las tarjetas en estado creado.
Nombre:	List<TarjetaPersistente> ObtenerTarjetasDeLote(int id_lote)
Descripción:	Obtiene las tarjetas que pertenecen a un lote dado el identificador del mismo.
Nombre:	List<DatosTarjetas> ReporteTarjetasImpresasOPorImprimir(int idlote)
Descripción:	Obtiene los datos de las tarjetas que pertenecen a un lote dado.
Nombre:	string BuscarNombreSucursalDadoNumero(string NoSucursal)
Descripción:	Obtiene el nombre de una sucursal dado su número.
Nombre:	string GetBancoDadoID(string idBanco)
Descripción:	Obtiene el nombre de un banco dado su identificador.
Caso de Uso: Crear Lote	
Nombre:	ListarNoSucursalTarjetasNoldLote()
Descripción:	Obtiene la lista de los datos de las sucursales que tienen tarjetas sin identificador

	de lote y en estado creada.
Nombre:	DatosTarjetasPorSucursal(string noSucursal)
Descripción:	Obtiene la cantidad de tarjetas con su sucursal correspondiente que están sin imprimir y sin identificador de lote.
Nombre:	MostrarFechaPorSucursalYCantidad(string noSucursal)
Descripción:	Obtiene para una sucursal especificada las fechas en que existen tarjetas por imprimir y la cantidad correspondiente a cada una de ellas.
Nombre:	CrearLote(string noSucursal, ArrayList fechasSeleccionadas)
Descripción:	Permite crear el lote con las tarjetas a imprimir pertenecientes a una sucursal determinada y que se crearon en las fechas especificadas, así como actualizar el identificador del lote en las tarjetas correspondientes.
Caso de Uso: Imprimir Tarjetas	
Nombre:	DatosLotesEstadoT Creado()
Descripción:	Permite obtener identificadores de lotes existentes con la cantidad de tarjetas de cada uno.
Nombre:	TarjetasAlImprimirDadoldLote(int idLote)
Descripción:	Permite obtener los datos que se imprimirán de cada tarjeta de todas las tarjetas de un lote dado un identificador del mismo.
Nombre:	TarjetasAlImprimir(ArrayList idLote)
Descripción:	Permite obtener los datos de las tarjetas que se van a imprimir dado el identificador de un lote.
Nombre:	ObtenerFilasEspaciadas(string filaOrig)
Descripción:	Permite obtener las filas con el formato que se deben mostrar por la tarjeta.
Nombre:	ActualizarDatosLoteT(ArrayList idLotes)
Descripción:	Permite actualizar datos en el registro del o los lote(s) que se seleccionaron para imprimir y poner en estadoT Impreso "I".
Nombre:	ListarDatosTarjetasAREimprimir(int idLote)
Descripción:	Permite listar los números de tarjetas a reimprimir dado un identificador de lote con un índice al lado para usar a la hora de escoger que rango se quiere reimprimir.
Nombre:	TarjetasAREimprimir(int indiceInicio, int indiceFin, int IdLote)
Descripción:	Permite reimprimir el rango de tarjetas seleccionadas.
Nombre:	SalvarOperacionReimpresionT(int indiceInicio, int indiceFin, int IdLote)
Descripción:	Permite guardar la operación de reimpresión en la tabla AccionUsuario.
Nombre:	FinalizarImpresionDeTarjetas(int ID_Lote)
Descripción:	Actualizar el estado de las tarjetas que estén en un mismo lote, pone en el estado finalizado "F".
Nombre:	TarjetasPorReimprimir()
Descripción:	Devuelve verdadero si hay lotes con estado de tarjetas en impreso "I".
Nombre:	ListarLotesDeTarjetasImpresas()
Descripción:	Obtiene lo identificadores de los lotes que tengan el estado de las tarjetas impreso "I".
Caso de Uso: Imprimir Pines	
Nombre:	CantidadPinesPorLotes()

Descripción:	Permite obtener los lotes con la cantidad de pines correspondiente a cada uno de ellos.
Nombre:	DarNombreABanco(string numTarjeta)
Descripción:	Permite obtener el nombre del banco según el número de una tarjeta.
Nombre:	DatosDeLotes(ArrayList idLotes)
Descripción:	Permite obtener los datos del lote o lotes a imprimir.
Nombre:	ActualizarDatosLoteP(ArrayList idLotes)
Descripción:	Permite actualizar los datos en la tabla Lote.
Nombre:	ListarDatosPinesAReimprimir(int idLote)
Descripción:	Permite mostrar los datos de los pines que se quieren reimprimir.
Nombre:	PinesAReimprimir(int indiceInicio, int indiceFin, int IdLote)
Descripción:	Permite escoger el intervalo de los pines que se quieren reimprimir y mandar los datos a imprimir.
Nombre:	SalvarOperacionReimpresionP(int indiceInicio, int indiceFin, int IdLote)
Descripción:	Permite guardar la operación de reimpresión de pines en la tabla AccionUsuario.
Nombre:	FinalizarImpresionDePines(int ID_Lote)
Descripción:	Permite finalizar la impresión, imposibilitando que se vuelva a imprimir cualquiera de estos pines.
Nombre:	PinesPorReimprimir()
Descripción:	Devuelve verdadero si hay lotes con estado de pines impreso "I".
Nombre:	ListarLotesDePinesImpresos()
Descripción:	Permite mostrar los lotes que están en estado impreso "I".
Caso de Uso: Captar Matrices	
Nombre:	CargarMatricesFichero(string dir)
Descripción:	Permite abrir un fichero que contiene matrices y guarda las mismas en una lista de matrices.
Nombre:	EncriptarMatrices(List<Matriz> matrices)
Descripción:	Cifra una lista de matrices.
Nombre:	GuardarMatrices(List<Matriz> matrices)
Descripción:	Guarda una lista de matrices cifradas en la Base de Datos.
Nombre:	GetTorresActivas()
Descripción:	Devuelve información referente a los nombres de las unidades lógicas de la computadora, el tipo de unidad y si esta está lista.
Nombre:	BuscarSubcarpetas(string path)
Descripción:	Devuelve las subcarpetas que existan en una dirección específica.
Nombre:	BuscarTxt(string path)
Descripción:	Devuelve los archivos de extensión .txt que existan en una dirección específica.

Nombre: COperacionAutomatica	
Tipo de clase: Controladora	
Atributo	Tipo
Configuracion	Configuracion

Handler	DataAccessLayer.DataHandler
B	System.Threading.Timer
A	System.Threading.Timer
Bloqueo1	Object
Bloqueo2	Object
empesoConciliaciones	Bool
empesoPeticiones	Bool
Responsabilidades	
Caso de Uso: Crear Autenticación, Procesar Solicitudes de Baja de la Tarjeta y Conciliaciones Automáticas	
Nombre:	void Actualizar()
Descripción:	Invoca al método EjecutarEspera(Object Info)cada 20 minutos.
Nombre:	void EjecutarEspera(Object Info)
Descripción:	Si se cumple una condición de tiempo ejecuta los métodos EnviarConciliaciones() y SolicitudesDeBajaDeTarjeta() y si se cumple otra condición ejecuta el método GestionarAutenticaciones.
Caso de Uso: Crear Autenticación	
Nombre:	void GestionarAutenticaciones()
Descripción:	Obtiene las solicitudes de creación de autenticaciones y crea las tarjetas de autenticación de los clientes.
Caso de Uso: Procesar Solicitudes de Baja de la Tarjeta.	
Nombre:	void SolicitudesDeBajaDeTarjeta()
Descripción:	Invoca al método que obtiene las solicitudes de baja de los bancos asociados y en caso de existir solicitudes especifica la cantidad que fueron procesadas.
Nombre:	string EjecutarSolicitudesDeBajaDeTarjeta()
Descripción:	Obtiene las solicitudes de baja de los bancos asociados.
Nombre:	ObtenerNuevaMatriz()
Descripción:	Matriz Asigna aleatoriamente una nueva matriz a una tarjeta.
Caso de Uso: Conciliaciones Automáticas	
Nombre:	void EnviarConciliaciones()
Descripción:	Envia automáticamente a los webServices de los bancos asociados las conciliaciones de las tarjetas creadas, activas y canceladas.

Nombre: Tarjeta	
Tipo de clase: Controladora	
Responsabilidades	
Nombre:	Tarjeta()
Descripción:	Constructor por defecto
Nombre:	DarMatriz()
Descripción:	Verifica que una matriz está cifrada, llama al método de la clase CriptografiaTeleBanca que descripta las filas de una matriz y luego retorna la matriz ya descriptada.

Nombre: CriptografiaTeleBanca	
Tipo de clase: Controladora	
Responsabilidades	
Nombre:	CriptografiaTeleBanca()
Descripción:	Constructor de la clase.
Nombre:	string Encriptar(string texto)
Descripción:	
Nombre:	string Desencriptar(string textoEncriptado)
Descripción:	
Nombre:	EncriptarMatriz(string[] filas)
Descripción:	Cifra un arreglo de filas de una matriz.
Nombre:	DesencriptarMatriz(string[] filasEncriptadas)
Descripción:	Descifra un arreglo de filas cifradas de una matriz

Nombre: DataHandler	
Tipo de clase: Interfaz	
Atributo	Tipo
cString	String
MyCurrent	HttpContext
Responsabilidades	
Caso de Uso: Crear Autenticación	
Nombre:	ObtenerListaBanco()
Descripción:	Obtiene la lista de los bancos asociados a TeleBanca.
Nombre:2	ObtenerNuevasMatrices()
Descripción:	Obtiene las matrices que no están asignadas a ninguna tarjeta.
Nombre	ObtenerIDTarjetas()
Descripción:	Obtiene la lista de los identificadores de las tarjetas existentes.
Nombre:	ObtenerSolicitudes(string IDBanco)
Descripción:	Obtiene una lista de solicitudes dado un identificador de un banco
Nombre:	GetBancoDadoID(string ID_Banco)
Descripción:	Obtiene un banco dado el identificador del mismo
Nombre:	GuardarNotificacion(string message, string idRol)
Descripción:	Guarda un mensaje y el rol que tendrá acceso a leer el mismo, agregando además la fecha y hora actual del sistema.
Descripción:	Devuelve verdadero o falso según
Nombre:	ActualizarSucursales()
Descripción:	Obtiene las nuevas sucursales de los bancos y las actualiza en la base de datos
Nombre:	GuardarTarjetasProcesadas(List<TarjetaPersistente> l_procesadas)
Descripción:	Guarda las tarjetas creadas en la base de datos.
Nombre:	ActualizarEstadoMatrices(List<Matriz> lm)
Descripción:	Actualiza el estado de las matrices que se asignan a las tarjetas creadas.
Caso de Uso: Procesar Solicitudes de Baja de la Tarjeta.	
Nombre:	ObtenerListaBanco()

Descripción:	Obtiene la lista de los bancos asociados.
Nombre:	GuardarNotificacion(string message, string idRol)
Descripción:	Guarda un mensaje en la base de datos para que lo lea un usuario determinado.
Nombre:	ConectWSBuscarTarjetasDarBaja(BancoPersistente b)
Descripción:	Se encarga de establecer conexión con el Web Service del Banco para obtener las solicitudes de baja de las tarjetas.
Nombre:	BuscarTarjeta(string tarjeta)
Descripción:	Se encarga de obtener una tarjeta dado el identificador de la misma.
Nombre:	ModificarTarjeta(string numero, TarjetaPersistente t)
Descripción:	Modifica en la base de datos algunos campos de una tarjeta.
Nombre:	DeshabilitarTarjetas()
Descripción:	Se encarga de copiar las tarjetas que estén en estado “deshabitada” que se encuentren en la tabla tarjeta en la tabla TLB_HistoricoTarjetas y llamar al método que elimina una tarjeta de la tabla TLB_Tarjeta para que las elimine.
Nombre:	BuscarTarjetasPorEstado(string estado)
Descripción:	Obtiene la lista de tarjetas que tengan un estado especificado.
Nombre:	CantidadDeMatricesDisponibles()
Descripción:	Obtiene la cantidad de matrices disponibles en la base de datos.
Nombre:	GuardarTarjetaEnHistorico(TarjetaPersistente t)
Descripción:	Guarda en la tabla TLB_HistoricoTarjetas los datos de una tarjeta.
Nombre:	CantidadDeTarjetasCambiadasEnElDiaDadoEstado(string estado)
Descripción:	Devuelve la cantidad de tarjetas, de un estado, que fueron procesadas en un día.
Nombre:	ModificarEstadoDeLaMatriz(int id, string nuevoEstado)
Descripción:	Cambia el estado de una matriz.
Nombre:	Obtener10MatricDisponibles()
Descripción:	Obtener diez matrices del total disponible.
Caso de Uso: Conciliaciones Automáticas	
Nombre:	ObtenerListaBanco()
Descripción:	Obtiene la lista de los bancos asociados.
Nombre:	GuardarNotificacion(string message, string idRol)
Descripción:	Guarda un mensaje en la base de datos para que lo lea un usuario determinado.
Caso de Uso: Conciliaciones Auxiliares	
Nombre:	GetBancoDadoID(string ID_Banco)
Descripción:	Obtiene el banco según el ID.
Casos de Uso: Conciliaciones Automáticas y Conciliaciones Auxiliares	
Nombre:	TarjetasDeBancoActivasEnFecha(DateTime fecha, string idBanco)
Descripción:	Obtiene las tarjetas activas en una fecha para un determinado banco.
Nombre:	TarjetasCanceladasEnDiaDeBanco(DateTime d, string idBanco)
Descripción:	Obtiene las tarjetas canceladas en una fecha para un determinado banco.
Nombre:	TarjetasDeBancoCreadasEnDia(DateTime f, string IDBanco)
Descripción:	Obtiene las tarjetas creadas en una fecha para un determinado banco.
Nombre:	EnviarConciliacionActivas(string[] ids, string Url)
Descripción:	Envía conciliación de tarjetas activas al Web Service.

Nombre:	EnviarConciliacionCanceladas(string[] t, string Url)
Descripción:	Envía al Web Service la conciliación de tarjetas canceladas.
Nombre:	EnviarConciliacionCreadas(string[] lti, string Url)
Descripción:	Envía al Web Service las conciliaciones creadas en el día.
Nombre:	UpdateEstadoTarjetaHistorico(string IdTarjeta, string estado)
Descripción:	Actualiza el estado de la tarjeta en el histórico.
Nombre:	UpdateEstadoTarjeta(string IdTarjeta, string estado)
Descripción:	Actualiza el estado de una tarjeta en la BD.
Caso de Uso: Realizar Reportes	
Nombre:	ObtenerLotesEstadoPinFinDadoFecha(string estado, DateTime f)
Descripción:	Obtener los lotes de pines según un estado y una fecha.
Nombre:	ObtenerLotesEstadoTFinDadoFecha(string estado, DateTime f)
Descripción:	Obtener los lotes de tarjetas según un estado y una fecha.
Nombre:	ObtenerLotesDadoEstadoPin(string estadoP)
Descripción:	Obtiene los lotes según el estado en que se encuentre el pin.
Nombre:	ObtenerLotesDadoEstadoT(string estadoT)
Descripción:	Obtiene los lotes según el estado en que se encuentre la tarjeta.
Nombre:	ObtenerTarjetasdeLote(int Id_lote)
Descripción:	Obtiene una lista de tarjetas de un lote determinado.
Nombre:	TarjetasDadoLote(int lote)
Descripción:	Obtiene una lista de tarjetas que tengan un mismo identificador de lote.
Nombre:	GetBancoDadoID(string ID_Banco)
Descripción:	Obtiene el banco según el ID.
Nombre:	BuscarNombreSucursalDadoNumero(string NoSucursal)
Descripción:	Permite obtiene el nombre de una sucursal dado el número de la misma.
Caso de Uso: Captar Matrices	
Nombre:	InsertarMatricesEncriptadasBD(List<Matriz> listaMatricesEncriptadas)
Descripción:	Inserta las matrices cifradas en la base de datos.
Caso de Uso: Crear Lote	
Nombre:	BuscarTarjetasNoldLoteEstadoC()
Descripción:	Retorna la lista de las sucursales que tengan tarjetas sin identificador de lote asignado y tengan el estado creado "C".
Nombre:	TarjetasAlprimirPorSucursalNoldLote(string noSucursal)
Descripción:	Permite obtener las tarjetas que no tengan asignado identificador de lote y que correspondan a una sucursal determinada.
Nombre:	BuscarNombreSucursalDadoNumero(string NoSucursal)
Descripción:	Permite obtiene el nombre de una sucursal dado el número de la misma.
Nombre:	ObtenerCantPorFechaYNoSucursal(DateTime fecha, string noSucursal)
Descripción:	Obtener la cantidad de tarjetas que se crearon de una sucursal especifica en una fecha determinada.
Nombre:	InsertarLote(LotePersistente lote)
Descripción:	Permite insertar un lote y retorna el identificador del mismo.
Nombre:	TarjetasAlprimirPorSucursalNoldLoteFecha(string noSucursal, DateTime f)

Descripción:	Obtiene las tarjetas de una sucursal, que se crearon en una fecha determinada.
Nombre:	ActualizarTarjetasAlImprimir(string IdNumeroTarjeta, int idLote)
Descripción:	Actualiza el identificador del lote en una tarjeta.
Casos de Uso: Imprimir Tarjetas e Imprimir Pines	
Nombre:	ObtenerTarjetasdeLote(int Id_lote)
Descripción:	Obtiene las tarjetas de un lote dado el identificador del lote.
Nombre:	ObtenerLoteDadoldLote(int idLote)
Descripción:	Permite obtener un lote dado su identificador.
Nombre:	BuscarTarjeta(string tarjeta)
Descripción:	Permite obtener una tarjeta dado su identificador.
Nombre:	InsertarAccionUsuario(AccionUsuarioPersistente accionUsuario)
Descripción:	Permite insertar en la tabla AccionUsuario la operación de reimprimir.
Nombre:	ModificarEstadoDeLaTarjeta(string tarjeta, string nuevoEstado)
Descripción:	Modifica el estado de una tarjeta.
Caso de Uso: Imprimir Tarjetas	
Nombre:	BuscarLoteEstadoTC()
Descripción:	Busca los lotes de tarjetas que tengan el estado creado.
Nombre:	ActualizarCamposLoteOrdenImp(int idLote, string idUsuarioOperadora, DateTime dtValor)
Descripción:	Actualiza en un lote los datos referentes a la Operadora de Impresión, el estado de las tarjetas y la hora en que se efectúa la impresión.
Nombre:	ActualizarEstadoTLote(int idLote, DateTime f)
Descripción:	Actualiza el estado en que se encuentran las tarjetas de un lote, poniendo el mismo en finalizado "F".
Nombre:	ObtenerLotesDadoEstadoT(string estadoT)
Descripción:	Obtiene los lotes y sus tarjetas correspondientes dado un estado tarjeta especificado.
Caso de Uso: Imprimir Pines	
Nombre:	ObtenerLotesDadoEstadoPin(string estadoP)
Descripción:	Obtiene los lotes y sus tarjetas correspondientes dado un estado pin especificado.
Nombre:	ObtenerListaBanco()
Descripción:	Obtiene la lista de todos los bancos asociados.
Nombre:	ActualizarPinEnLote(int idLote, string idUsuarioPin, DateTime dtValor)
Descripción:	Actualiza en un lote los datos referentes a la Operadora de Impresión, el estado de los pines y la hora en que se efectúa la impresión.
Nombre:	BuscarNombreSucursalDadoNumero(string NoSucursal)
Descripción:	Permite obtiene el nombre de una sucursal dado el número de la misma
Nombre:	ActualizarEstadoPinLote(int idLote, DateTime f)
Descripción:	Actualiza en el lote los datos referentes a la Operadora de Impresión de Pines y el campo estado de pines lo cambia a finalizado "F".

Nombre: Configuración	
Tipo de clase: Entidad	
Atributo	Tipo

direccServBD	string
horaConciliaciones	DateTime
direccServFTP	string
usuarioFTP	string
contraseñaFTP	string
tiempoInactividad	DateTime
horaInicioPetic	DateTime
direccSalvaBD	string
horaSalva	DateTime
impresoraPin	string
impresoraTarjeta	string
Responsabilidades	
Nombre:	Configuracion()
Descripción:	Constructor por defecto que inicializa las variables del tipo DateTime.
Nombre:	Configuracion(string direccServBD, DateTime horaConciliaciones, string direccServFTP, string usuarioFTP, string contraseñaFTP, DateTime tiempoInactividad, string direccSalvaBD, DateTime horaInicioPetic, DateTime horaSalva, string imprPin, string imprTarj)
Descripción:	Constructor de la clase.
Nota:	Tiene además todos los métodos set y get de los atributos.

Nombre: Notificacion	
Tipo de clase: Entidad	
Atributo	Tipo
id_Notificacion	int
Descripcion	string
id_rol	string
Fecha	DateTime
Responsabilidades	
Nombre:	Notificacion()
Descripción:	Constructor por defecto.
Nombre:	Notificacion(string descripcion, string id_rol)
Descripción:	Constructor de la clase.
Nota:	Tiene además todos los métodos set y get de los atributos.

Nombre: AccionUsuarioPersistente	
Tipo de clase: Entidad	
Atributo	Tipo
Usuario	string
Funcionalidad	string
Fecha	DateTime
Descripcion	List<string>

Responsabilidades	
Nombre:	AccionUsuarioPersistente()
Descripción:	Constructor por defecto.
Nombre:	Notificacion(string aUsuario, string aFuncionalidad,DateTime aFecha, List<string> aDescripcion)
Descripción:	Constructor de la clase.
Nota:	Se implementan todos los métodos set y get de los atributos.

Nombre: BancoPersistente	
Tipo de clase: Entidad	
Atributo	Tipo
Nombre	string
webServices	string
Password	string
numBanco	string
Abreviatura	string
Responsabilidades	
Nombre:	BancoPersistente()
Descripción:	Constructor por defecto.
Nombre:	BancoPersistente(string Nombre, string WebServices, string PassWord, string NumBanco, string Abreviatura)
Descripción:	Constructor de la clase.
Nota:	Se implementan todos los métodos set y get de los atributos.

Nombre: LotePersistente	
Tipo de clase: Entidad	
Atributo	Tipo
idLote	int
idUsuarioTarjeta	string
fechaHoralmpTarjeta	DateTime
estadoT	string
idUsuarioPin	string
fechaHoralmpPin	DateTime
estadoP	string
Tarjetas	List<TarjetaPersistente>
Responsabilidades	
Nombre:	LotePersistente()
Descripción:	Constructor por defecto.
Nombre:	LotePersistente(int idLote, string UsuarioTarjeta, DateTime FechaHoraTarjeta,DateTime FechaHoraPin, string UsuarioPin, string EstadoT, string EstadoP)
Descripción:	Constructor de la clase.

Nombre:	LotePersistente(string estadoT, string estadoP)
Descripción:	Constructor que se usa para el caso de uso Realizar Reporte
Nota:	Se implementan todos los métodos set y get de los atributos.

Nombre: Matriz	
Tipo de clase: Entidad	
Atributo	Tipo
id_matriz	int
Estado	string
Encriptada	Boolean
Filas	string[]
Responsabilidades	
Nombre:	Matriz(int id,string fil1, string fil2, string fil3, string fil4, string fil5,string fil6, string fil7, string fil8, string fil9, string fil10,Boolean encript)
Descripción:	Constructor de la clase.
Nota:	Se implementan todos los métodos set y get de los atributos.

Nombre: TarjetaPersistente	
Tipo de clase: Entidad	
Atributo	Tipo
idNumeroTarjeta	string
noPin	string
nombrePropietario	string
Apellidos	string
noSucursal	string
fechaOrdenImp	DateTime
estadoPin	string
Estado	string
Matriz	Matriz
idLote	int
idCliente	string
tipoidentificacion	string
idPais	string
Responsabilidades	
Nombre:	TarjetaPersistente()
Descripción:	Constructor de la clase.
Nota:	Se implementan todos los métodos set y get de los atributos.

Nombre: UsuarioPersistente	
Tipo de clase: Entidad	
Atributo	Tipo

Usuario	string
Nombre	string
Contrasena	string
Activo	bool
carnetIdentidad	string
Rol	RolPersistente
Responsabilidades	
Nombre:	UsuarioPersistente(string Usuario, string Contrasena, string Nombre, RolPersistente Rol,bool Activo, string CarnetIdentidad)
Descripción:	Constructor de la clase.
Nota:	Se implementan todos los métodos set y get de los atributos.

Nombre: RolPersistente	
Tipo de clase: Entidad	
Atributo	Tipo
Funcionalidades	List<int>
nombreMenu	List<int>
valorMenu	List<int>
Modulo	List<int>
Nombre	string
Descripcion	string
Responsabilidades	
Nombre:	RolPersistente()
Descripción:	Constructor de la clase.
Nota:	Se implementan todos los métodos set y get de los atributos.

Nombre: ReporteContenido	
Tipo de clase: Entidad	
Atributo	Tipo
nroTarjeta	string
Identificacion	string
nombrePropietario	string
primerApellido	string
Banco	string
nroSucursal	string
idLote	Int
Responsabilidades	
Nombre:	ReporteContenido()
Descripción:	Constructor por defecto.
Nombre:	ReporteContenido(string nrotarjeta, string identificacion, string nombrePropietario,string primerApellido, string banco, string nroSucursal,int idlote)
Descripción:	Constructor de la clase.

Nota:	Se implementan todos los métodos set y get de los atributos.
-------	--

Nombre: DatosLote	
Tipo de clase: Entidad	
Atributo	Tipo
numeroLote	string
nombreSucursal	string
nombreBanco	string
cantidadTotalTarjetas	int
Operadora	string
Responsabilidades	
Nombre:	Datoslote()
Descripción:	Constructor por defecto.
Nombre:	Datoslote(LotePersistente lote,string nombreSucursal,string nombreBanco,string nombreOperadora)
Descripción:	Constructor de la clase.
Nota:	Se implementan todos los métodos set y get de los atributos.

Nombre: DatosTarjetas	
Tipo de clase: Entidad	
Atributo	Tipo
Handler	DataHandler
numeroLote	string
nombreSucursal	string
Banco	string
numeroTarjeta	string
Nombre	string
Apellidos	string
Identificacion	string
Operadora	string
Responsabilidades	
Nombre:	DatosTarjetas()
Descripción:	Constructor por defecto.
Nombre:	DatosTarjetas(TarjetaPersistente tarjeta, string nombreSucursal, string nombreBanco, string nombreOperadora)
Descripción:	Constructor de la clase.
Nota:	Se implementan todos los métodos set y get de los atributos.

Nombre: WebUserControl_Autenticacion	
Tipo de clase: Interfaz	
Responsabilidades	

Nombre:	void Page_Load(object sender, EventArgs e)
Descripción:	Llama al método GetDataMenu de la clase TeleBancaWS para mostrar la interfaz pasándole el ID del Modulo.
Nombre:	void Menu1_MenuItemClick(object sender, MenuEventArgs e)
Descripción:	Activa la MultiView1, que es donde se muestra el mensaje de presentación del Modulo.
Casos de Uso: Captar Matrices	
Nombre:	void Matriz_Activate(object sender, EventArgs e)
Descripción:	Activa la TreeView1 y llama al método GetTorresA() para obtener las torres Activas.
Nombre:	void ListBox1_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Activa el Button1, lo pone en true.
Nombre:	void Button2_Click(object sender, EventArgs e)
Descripción:	Activa la MultiView1, que es donde se muestra el mensaje de presentación del Modulo.
Nombre:	void Button1_Click(object sender, EventArgs e)
Descripción:	Llama al método CargarEncriptarGuardarMatrices() de la clase TeleBancaWS pasandole la direccion donde se encuentra la matriz.
Casos de Uso: Enviar Conciliaciones Auxiliares	
Nombre:	void View_Concilaciones_Activate(object sender, EventArgs e)
Descripción:	Llama al método GetIDBancos() de la clase TeleBancaWS.
Nombre:	void RadioButtonList_C_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Habilita el DropDownList_C_Bancos.
Nombre:	void DropDownList_C_Bancos_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Activa las opciones Calendar_C y Button_C_Aceptar
Nombre:	void Button_C_Aceptar_Click(object sender, EventArgs e)
Descripción:	Llama a los métodos ExistenTarjetasActivasDeBancoEnDia(), ExistenTarjetasCanceladasDeBancoEnDia() y ExistenTarjetasImpresasDeBancoEnDia() de la clase TeleBancaWS en dependencia de la selección del usuario.
Nombre:	void EnviarConciliaciones()
Descripción:	Llama a los métodos EnviarConciliacionImpresas(), EnviarConciliacionTarjetasCanceladas() y EnviarConciliacionTarjetasCreadas() de la clase TeleBancaWS en dependencia de la selección del usuario.
Nombre:	void Button_C_Cancelar_Click(object sender, EventArgs e)
Descripción:	Activa la MultiView1, que es donde se muestra el mensaje de presentación del Modulo.
Casos de Uso: Crear Lotes	
Nombre:	void View_CrearLotes_Activate(object sender, EventArgs e)
Descripción:	Llama al método ListarNoSucursalTarjetasNoldLote() de la clase TeleBancaWS.
Nombre:	void DropDownList_CL_Sucursales_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Llama al método DatosTarjetasPorSucursal() de la clase TeleBancaWS.
Nombre:	void Button_CL_Aceptar_Click(object sender, EventArgs e)

Descripción:	Activa la View_CrearLotes_Fechas.
Nombre:	void Button_CL_Cancelar_Click(object sender, EventArgs e)
Descripción:	Activa la MultiView1, que es donde se muestra el mensaje de presentación del Modulo.
Nombre:	void View_CrearLotes_Fechas_Activate(object sender, EventArgs e)
Descripción:	Llama al método MostrarFechaPorSucursalYCantidad() de la clase TeleBancaWS.
Nombre:	void CheckBoxList_CL_Fechas_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Carga la cantidad de tarjetas que no tienen lotes por fecha.
Nombre:	void Button_CL_FechasCrear_Click(object sender, EventArgs e)
Descripción:	Llama al método Confirmar pasándole un string con el mensaje de confirmación y un string con el nombre del método al cual es se va a redireccionar.
Nombre:	void CrearLote()
Descripción:	Llama al método CrearLote() de la clase TeleBancaWS, que es el encargado de mandar crear.
Nombre:	void Button_CL_FechasCancelar_Click(object sender, EventArgs e)
Descripción:	Te activa la MultiView en la que puedas escoger la Sucursal
Caso de Uso: Imprimir Pines	
Nombre:	void Imprimir_Pines_Activate(object sender, EventArgs e)
Descripción:	Llama al método ObtenerConfiguracion() de la clase TeleBancaWS para comprobar si existe la impresora de pines y despues llama al metodo CantidadPinesPorLotes() de la clase TeleBancaWS.
Nombre:	void CheckBoxList_IP_Lotes_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Muestra la cantidad de lotes con el estado en los pines sin imprimir.
Nombre:	void Button_IP_Aceptar_Click(object sender, EventArgs e)
Descripción:	Llama al método Confirmar pasándole un string con el mensaje de confirmación y un string con el nombre del método al cual es se va a redireccionar.
Nombre:	void ImprimirLotes()
Descripción:	Llama al método DatosDeLotes() de la clase TeleBancaWS el cual obtiene las tarjetas sin pines según el lote y se lo manda al Crystal Report para que se imprima.
Nombre:	void Button_IP_Cancelar_Click(object sender, EventArgs e)
Descripción:	Activa la MultiView1, que es donde se muestra el mensaje de presentación del Modulo.
Nombre:	void Button_IP_Reimprimir_Click(object sender, EventArgs e)
Descripción:	Llama al método ListarDatosPinesAReimprimir() de la clase TeleBancaWS.
Nombre:	void Button_IP_ReimpFinalizar_Click(object sender, EventArgs e)
Descripción:	Llama al método Confirmar pasándole un string con el mensaje de confirmación y un string con el nombre del método al cual es se va a redireccionar.
Nombre:	void FinalizarImpP()
Descripción:	Llama al método FinalizarImpresionDePines() de la clase TeleBancaWS.
Nombre:	void GridView_IP_Reimprimir_PageIndexChanging(object sender, GridViewPageEventArgs e)
Descripción:	Activa el GridView_IP_Reimprimir.

Nombre:	void Button_IP_ReimpAceptar_Click(object sender, EventArgs e)
Descripción:	Llama al método Confirmar pasándole un string con el mensaje de confirmación y un string con el nombre del método al cual es se va a redireccionar.
Nombre:	void ReimprimirPines()
Descripción:	Llama al método PinesAreimprimir() de la clase TeleBancaWS el cual obtiene las tarjetas sin pines según el lote y se lo manda al Crystal Report para que se imprima.
Nombre:	void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Cuenta las filas del GridView y las muestra.
Nombre:	void Button_IP_ReimpCancelar_Click(object sender, EventArgs e)
Descripción:	Limpia el GridView y te acita el MultiView de Imprimir y Finalizar Pines.
Nombre:	void View2_Activate(object sender, EventArgs e)
Descripción:	Llama al método ListarLotesDePinesImpresos() de la clase TeleBancaWS.
Nombre:	void CheckBoxList1_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Según la cantidad de Items que tenga el CheckBox, activa los botones Reimprimir e Imprimir Pines.
Nombre:	void Button6_Click(object sender, EventArgs e)
Descripción:	Activa todos los CheckBox en true.
Nombre:	void Button5_Click(object sender, EventArgs e)
Descripción:	Activa todos los CheckBox en false.
Caso de uso: Imprimir Tarjeta	
Nombre:	void View_Imprimir_Tarjetas4_Activate(object sender, EventArgs e)
Descripción:	Llama al método ObtenerConfiguracion() de la clase TeleBancaWS para comprobar si existe la impresora de tarjetas y despues llama al metodo DatosLotesEstadoTCreado() de la clase TeleBancaWS.
Nombre:	void CheckBoxList_IT_Lotes_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Muestra la cantidad de lotes con estado tarjeta sin imprimir.
Nombre:	void Button_IT_Aceptar_Click(object sender, EventArgs e)
Descripción:	Llama al método Confirmar pasándole un string con el mensaje de confirmación y un string con el nombre del método al cual es se va a redireccionar.
Nombre:	void ImprimirLotesT()
Descripción:	Llama al método TarjetasAlImprimir() de la clase TeleBancaWS el cual obtiene las tarjetas según el lote y se lo manda al Crystal Report para que se imprima.
Nombre:	void Button_IT_Cancelar_Click(object sender, EventArgs e)
Descripción:	Activa la MultiView1, que es donde se muestra el mensaje de presentación del Modulo.
Caso de uso: Reimprimir y Finalizar Tarjetas	
Nombre:	void Button_IT_Reimprimir_Click(object sender, EventArgs e)
Descripción:	Llama al método ListarDatosTarjetasAreimprimir() de la clase TeleBancaWS.
Nombre:	void Button_IT_Finalizar_Click(object sender, EventArgs e)
Descripción:	Llama al método Confirmar pasándole un string con el mensaje de confirmación y un string con el nombre del método al cual es se va a redireccionar.
Nombre:	void FinalizarImpT()

Descripción:	Llama al método FinalizarImpresionDeTarjetas() de la clase TeleBancaWS.
Nombre:	void GridView_IT_Reimp_PageIndexChanging(object sender, GridViewPageEventArgs e)
Descripción:	Activa el GridView_IT_Reimp.
Nombre:	void Button_IT_ReimpAceptar_Click(object sender, EventArgs e)
Descripción:	Llama al método Confirmar pasándole un string con el mensaje de confirmación y un string con el nombre del método al cual es se va a redireccionar.
Nombre:	void ReimprimirTarjetas()
Descripción:	Llama al método TarjetasAREimprimir() de la clase TeleBancaWS el cual obtiene las tarjetas sin pines según el lote y se lo manda al Crystal Report para que se imprima.
Nombre:	void DropDownList3_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Cuenta las filas del GridView y las muestra.
Nombre:	void Button_IT_ReimpCancelar_Click(object sender, EventArgs e)
Descripción:	Limpia el GridView y te activa el MultiView de Imprimir y Finalizar Tarjetas.
Nombre:	void View3_Activate(object sender, EventArgs e)
Descripción:	Llama al método ListarLotesDeTarjetasImpresas() de la clase TeleBancaWS.
Nombre:	void CheckBoxList2_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Según la cantidad de Items que tenga el CheckBox, activa los botones Reimprimir e Imprimir Tarjetas.
Nombre:	void Button3_Click(object sender, EventArgs e)
Descripción:	Activa todos los CheckBox en false.
Nombre:	void Button4_Click(object sender, EventArgs e)
Descripción:	Activa todos los CheckBox en true.
Caso de Uso	Enviar Realizar Reportes.
Nombre:	void View_RealizarReporte_Activate(object sender, EventArgs e)
Descripción:	Es la View de Realizar Reportes con los RadioButtons Lotes Impresos y Lotes por Imprimir deshabilitados y el botón Aceptar en false.
Nombre:	void RadioButtonList_RR_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Cuando seleccionas un RadioButton se activa el botón Aceptar.
Nombre:	void Button_RR_Aceptar_Click(object sender, EventArgs e)
Descripción:	Llama al método ObtenerLotesPorImprimir() de la clase TeleBancaWS, si escoges el RadioButton Lotes Por Imprimir de lo contrario te activa la View ReportesCreados.
Nombre:	void View_ReporteCreados_Activate(object sender, EventArgs e)
Descripción:	Llama al método ObtenerLotesFinalizados() de la clase TeleBancaWS.
Nombre:	void Button_RR_Camcelar_Click(object sender, EventArgs e)
Descripción:	Activa la MultiView1, que es donde se muestra el mensaje de presentación del Modulo.
Nombre:	void Button_RR_CreadosCancelar_Click(object sender, EventArgs e)
Descripción:	Activa la MultiView de Realizar Reportes.
Nombre:	void Button_RR_CreadosAceptar_Click(object sender, EventArgs e)

Descripción:	Llama al método ReporteTarjetasPorLote() de la clase TeleBancaWS.
Nombre:	void CheckBoxList3_SelectedIndexChanged(object sender, EventArgs e)
Descripción:	Activa los botones Aceptar y Cancelar según los CheckBox.
Nombre:	void RellenarHijos(String dir)
Descripción:	Llama al método GetSubCarpetas() de la clase TeleBancaWS para obtener las subcarpetas de la seleccionada.
Nombre:	void TreeView1_TreeNodeExpanded(object sender, TreeNodeEventArgs e)
Descripción:	Llama al método RellenarHijos()
Nombre:	void TreeView1_SelectedNodeChanged(object sender, EventArgs e)
Descripción:	Busca en la carpeta seleccionada todos los Archivos de extensión (.txt).

3.3.4 Diagrama de clases persistentes

Este modelo debe ser usado siempre que se modelen datos persistentes, en sistemas que manipulan gran cantidad de información, contenida en medios de almacenamiento persistente.

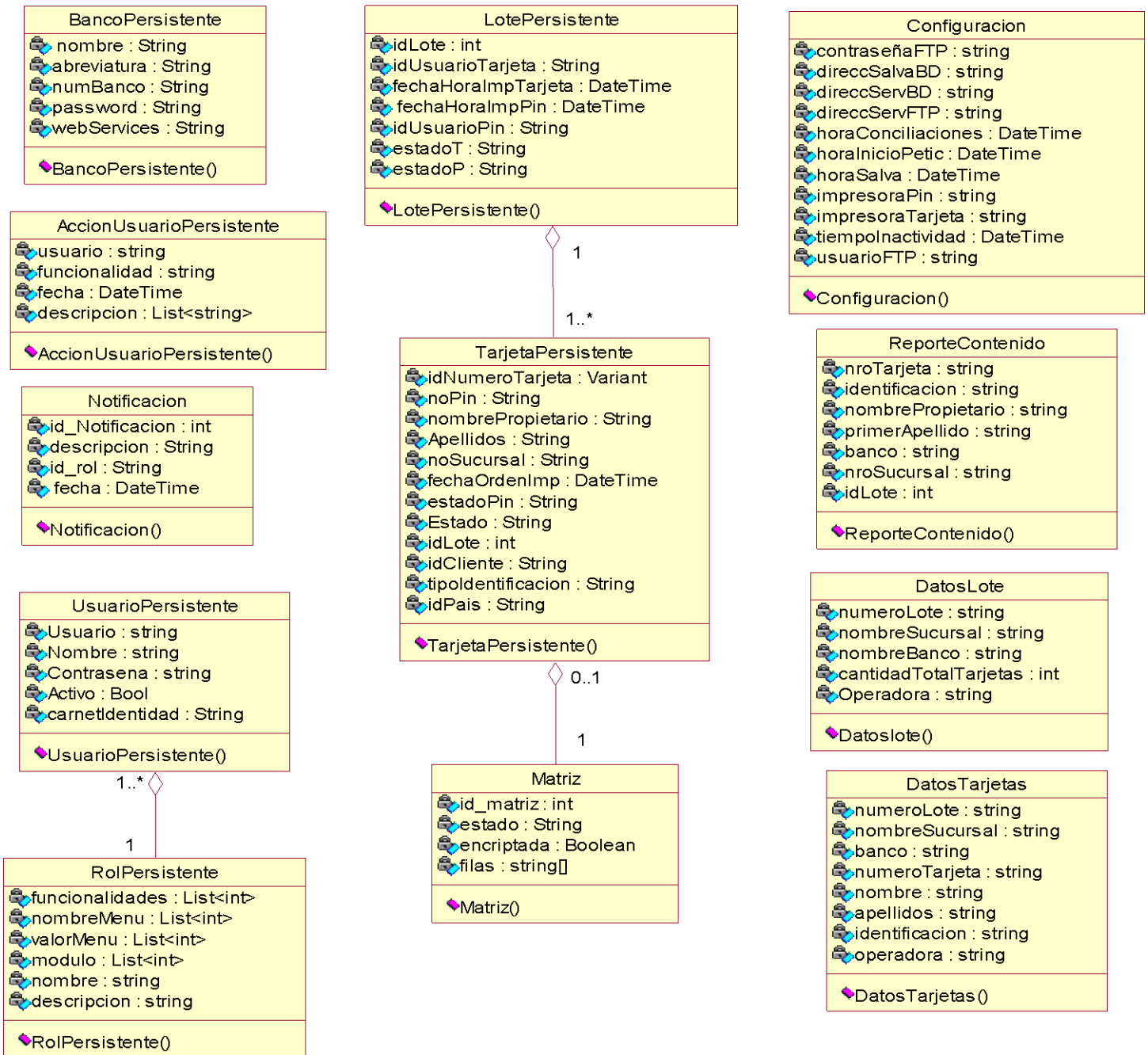


Figura 3.7 Diagrama de clases persistentes.

3.4 Conclusiones

En este capítulo se finaliza la etapa de análisis del sistema, habiéndose hecho el modelo de clases de análisis. También se concluye así con la etapa de diseño del software. Se realizaron los diagramas de interacción mediante diagramas de secuencia, de forma tal que se desarrollara un diagrama de interacción para cada una de las funcionalidades descritas, para lograr como resultado el diagrama de diseño Web del sistema. Todos estos elementos nos dan brindan una idea más específica del sistema que se propone.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción

En el presente capítulo se realizará el modelo de implementación, en el cual se desarrollaran el diagrama de componentes y el diagrama de despliegue como conclusión al desarrollo de la aplicación. También se efectuarán las pruebas de calidad al sistema desarrollado como colofón imprescindible para lograr un software que cumpla con los requerimientos anteriormente planteados.

4.2 Implementación

4.2.1 Modelo de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema ya que muestra los nodos y componentes de software que se encuentran en el sistema y su distribución en el mismo. Los nodos pertenecen al mundo material, un nodo es un elemento físico, que existe en tiempo de ejecución y representa un recurso computacional que generalmente tiene alguna memoria y, a menudo, capacidad de procesamiento, sirven para modelar la topología del hardware sobre el que se ejecuta el sistema. Un nodo representa normalmente un procesador o un dispositivo sobre el que se pueden desplegar los componentes, por lo que el diagrama de despliegue sitúa el software en el hardware que lo contiene.

El diagrama que se muestra a continuación representa un modelo ideal que está en correspondencia con la arquitectura cliente-servidor en tres capas, compuesto por ocho nodos y la conexión entre ellos.

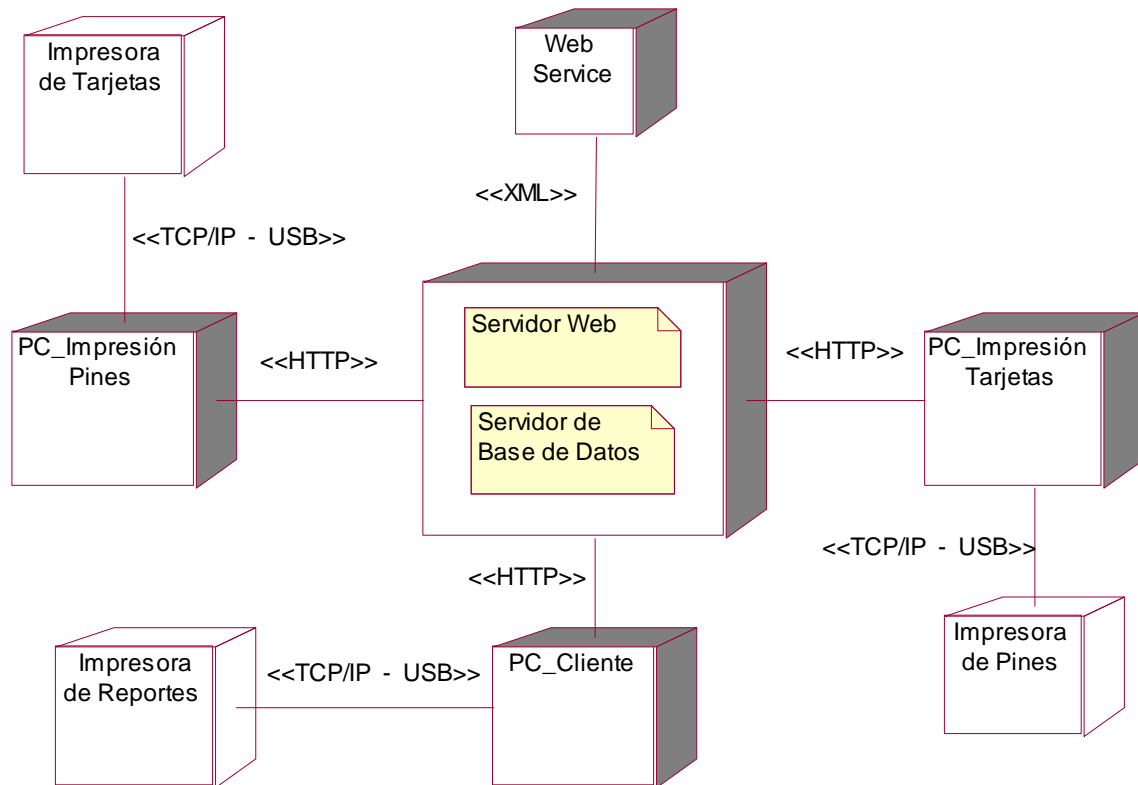


Figura 4.1 Diagrama de despliegue.

4.2.2 Diagrama de componentes

Un componente se corresponde con una o varias clases, interfaces o colaboraciones. Los componentes pertenecen al mundo físico, es decir, representan un bloque de construcción al modelar aspectos físicos de un sistema. Una característica básica de un componente es que: “debe definir una abstracción precisa con una interfaz bien definida, y permitiendo reemplazar fácilmente los componentes más viejos con otros más nuevos y compatibles.” En UML todos los elementos físicos se modelan como componentes.

Los diagramas de componentes se utilizan para modelar la vista estática de un sistema, muestran las organizaciones y dependencias lógicas entre componentes software que formarán un sistema, sean éstos componentes de código fuente, binarios o ejecutables. Cuando se realiza un diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas

utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.



Figura 4.2 Diagrama de componentes por paquetes.

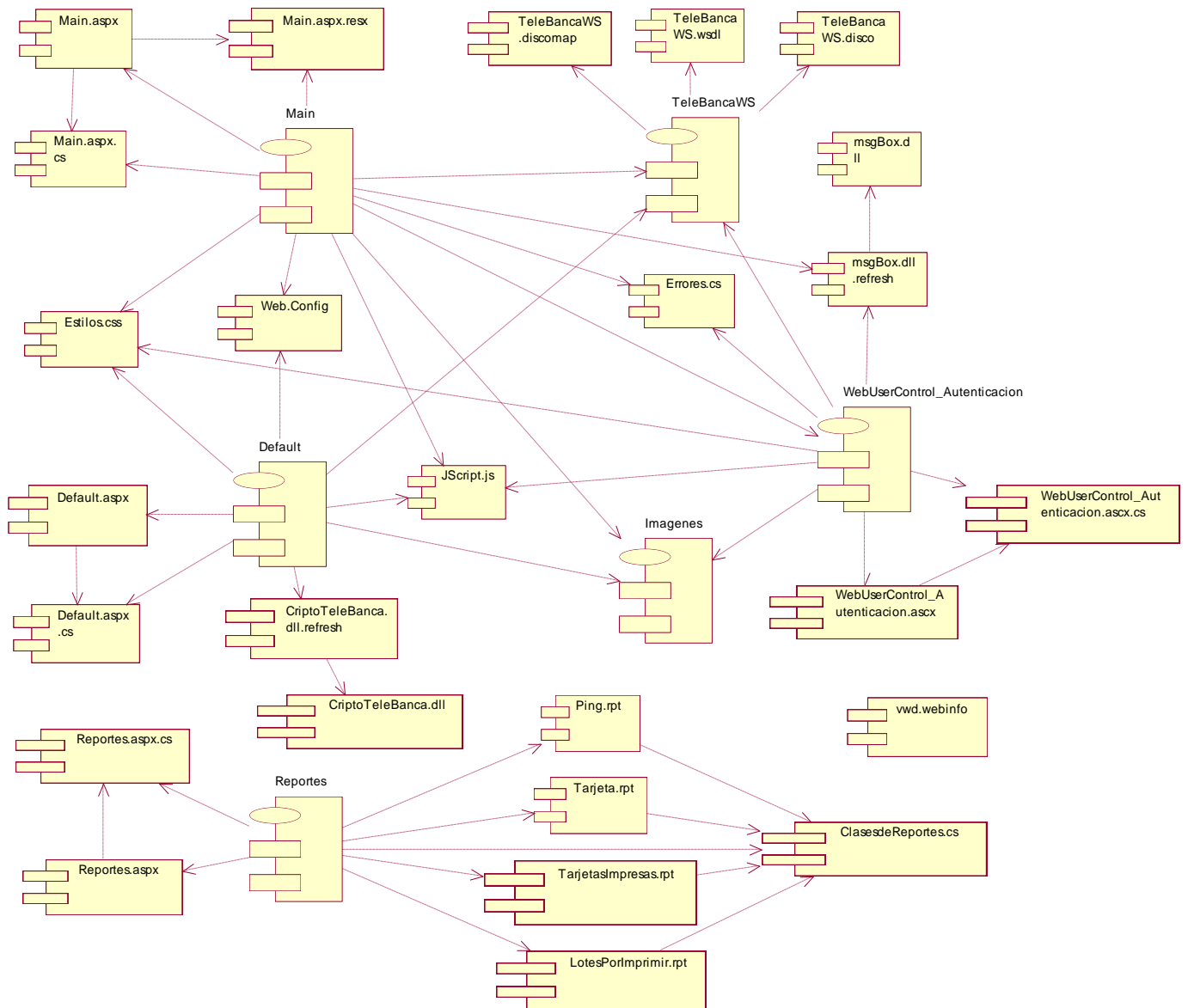


Figura 4.3 Diagrama de componentes del paquete Presentación.

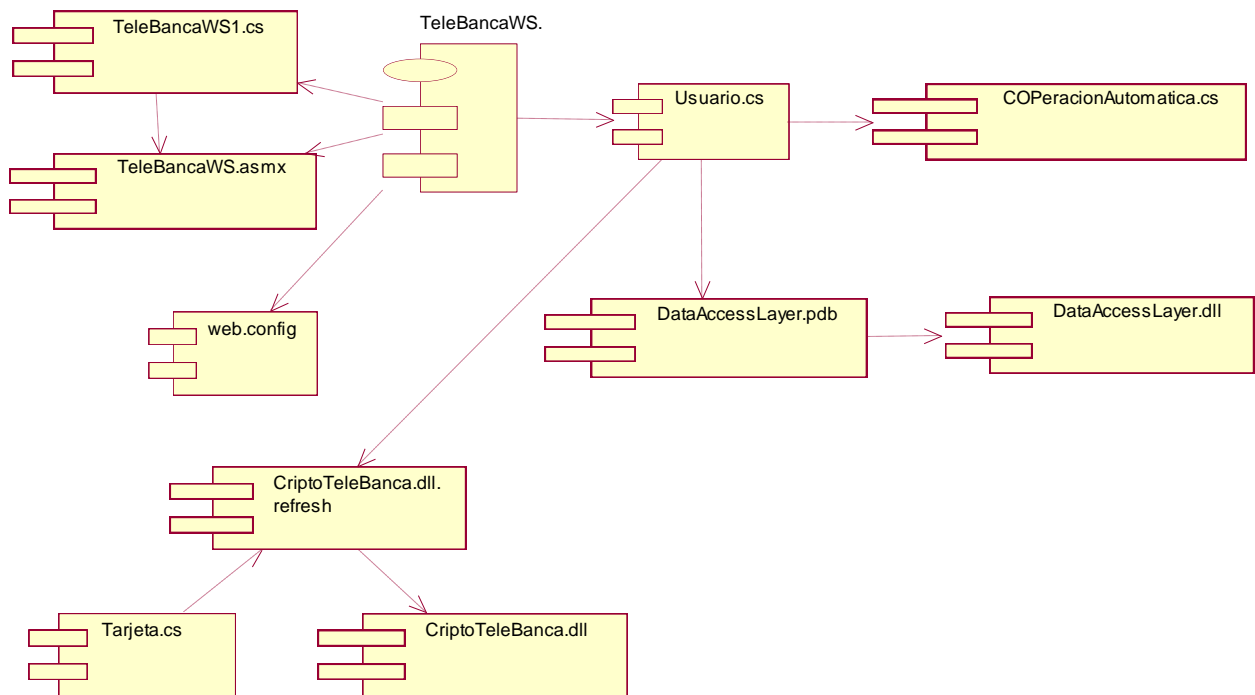


Figura 4.4 Diagrama de componentes del paquete Lógica de Negocio.

4.2.3 Descripción de los componentes

Componentes del paquete Lógica de Negocio	Propósito
TeleBancaWS1.cs	Fichero contenedor de código del Web Service.
TeleBancaWS.asmx	Fichero que soporta el Web Service.
web.config	Configuración de parámetros del sitio web.
Usuario.cs	Fichero contenedor de código de la clase Usuario.
DataAccesLayer.pdb	Interfaz de comunicación entre el sistema y la capa de acceso a datos.
DataAccesLayer.dll	Librería de comunicación con la capa de acceso a datos.
COPeracionAutomatica.cs	Fichero que contiene la clase COPeracionAutomatica.
CriptoTeleBanca.dll	Librería de la clase CriptografiaTeleBanca

CriptoTeleBanca.dll.refresh	Interfaz de comunicación del sistema con la librería CriptoTeleBanca.dll
-----------------------------	--

Componentes del paquete Presentación	Propósito
Default.aspx	Página Web de autenticación de la aplicación.
Default.aspx.cs	Componente de código de la Página Web de autenticación de la aplicación.
Main.aspx	Página Web de inicio de la aplicación.
Main.aspx.cs	Fichero de código contenedor de la Página Web de inicio de la aplicación.
Main.aspx.resx	Componente auxiliar para la Página Web de inicio de la aplicación.
Reportes.aspx	Página Web para realizar los reportes.
Reportes.aspx.cs	Fichero de código contenedor de la Página Web de reportes.
Web.Config	Contiene los parámetros del Sitio Web.
WebUserControl_Autenticacion.ascx	Componente de la página WebUserControl_Autenticacion.
WebUserControl_Autenticacion.ascx.cs	Fichero de código la página WebUserControl_Autenticacion.
CriptoTelebanca.dll	Librería para la criptografía de autenticación de los usuarios.
CriptoTelebanca.dll.refresh	Interfaz entre la librería CriptoTelebanca.dll y la aplicación.
msgBox.dll	Librería para los mensajes de error.
msgBox.dll.refresh	Interfaz entre la librería de los mensajes de error y la aplicación.
Ping.rpt	Fichero que contiene los datos que se imprimirán en el PIN del cliente.

Tarjeta.rpt	Fichero que contiene los datos que se imprimirán en la tarjeta del cliente.
TarjetasImpresas.rpt	Fichero de Reporte de impresión de lotes impresos.
LotesPorImprimir.rpt	Fichero de Reporte de impresión de lotes por imprimir.
ClasesdeReportes.cs	Clases que contienen los datos que saldrán en los reportes
Errores.cs	Fichero de código para el tratamiento de errores.
Paquete TeleBancaWS	Describe las funcionalidades del Web Service de la lógica del negocio.
Tarjeta.cs	Fichero que contiene la clase Tarjeta.
Estilos.css	Describe el estilo de las páginas web.
vwd.webinfo	Fichero que contiene información del sitio web.
JScript.js	Fichero que contiene código java Script de algunas funciones para el navegador.
TeleBancaWS.discomap	Fichero que proporciona una interfaz de comunicación con el Web Service de lógica del negocio.
TeleBancaWS.wsdl	Fichero que proporciona una interfaz de comunicación con el Web Service de lógica del negocio.
TeleBancaWS.disco	Fichero que proporciona una interfaz de comunicación con el Web Service de lógica del negocio.

4.3 Modelo de Prueba

4.3.1 Descripción de los casos de prueba de integración

Caso de uso: CU_Crear Lote

Entrada	Resultados	Condiciones
El Administrador de BT selecciona la sucursal a la cual desea crearle el lote.	El sistema muestra el nombre de la sucursal con la cantidad de tarjetas a imprimir y activa el botón "Aceptar".	El Administrador de BT debe estar autenticado al sistema y haber seleccionado la opción Crear Lotes.

El Administrador de BT selecciona la opción "Aceptar".	El sistema muestra el o los día(s) en que existen tarjetas pendientes de BT para la sucursal seleccionada con la cantidad de tarjetas a imprimir para cada día.	El Administrador de BT debe haber seleccionado la sucursal a la cual desea crearle el lote.
El Administrador de BT selecciona el o los días que pertenecerán al lote.	El sistema obtiene la cantidad total de tarjetas a imprimir que seleccionó el Administrador y la muestra, activando el botón "Crear".	El Administrador de BT debe haber seleccionado la opción "Aceptar".
El Administrador de BT selecciona la opción "Crear".	El sistema muestra el mensaje: "¿Está segura que desea crear el lote?".	El Administrador de BT debe haber seleccionado el o los día(s) que pertenecerán al lote.
El Administrador de BT selecciona la opción "Aceptar".	El Sistema muestra el mensaje: "El lote ha sido creado satisfactoriamente."	El Administrador de BT debe haber selecciona la opción "Crear".

Caso de uso: CU_Imprimir Tarjetas

Entrada	Resultados	Condiciones
La Operadora de Impresión de Tarjetas selecciona el o los lote(s) que desea imprimir.	El Sistema muestra la cantidad de tarjetas seleccionadas a imprimir y activa la opción de "Imprimir".	La Operadora de Impresión de Tarjetas debe estar autenticada al sistema y debe haber seleccionado la opción Imprimir Tarjetas.
La Operadora de Impresión de Tarjetas selecciona la opción "Imprimir".	El Sistema muestra el mensaje: "¿Está seguro que desea imprimir?".	La Operadora de Impresión de Tarjetas debe haber seleccionado el o los lote(s) que desea imprimir.
La Operadora de Impresión de	El Sistema muestra un formulario	La Operadora de Impresión de

Tarjetas acepta la opción "Aceptar".	con los lotes que tengan el estado tarjeta "Impreso" y la opción "Marcar Todos".	Tarjetas debe haber seleccionado la opción "Imprimir".
--------------------------------------	--	--

Caso de uso: CU_ Reimprimir y Finalizar Tarjetas, Sección Finalizar Tarjeta

Entrada	Resultados	Condiciones
La Operadora de Impresión de Tarjeta selecciona el o los lote(s) que desea finalizar.	El sistema activa la opción "Finalizar".	La Operadora de Impresión de Tarjetas debe haber aceptado la opción "Aceptar"
La Operadora de Impresión de Tarjetas selecciona la opción "Finalizar".	El Sistema muestra el mensaje "Si finaliza no podrá volver a imprimir ninguna de las tarjetas que forman el lote o los lotes escogidos por usted. ¿Está seguro que desea finalizar? ".	La Operadora de Impresión de Tarjeta debe haber seleccionado el o los lote(s) que desea finalizar.
La Operadora de Impresión de Tarjetas selecciona la opción "Aceptar".	El Sistema muestra el mensaje: "El proceso de impresión ha concluido satisfactoriamente".	La Operadora de Impresión de Tarjetas debe haber seleccionado la opción "Finalizar".

Caso de uso: CU_ Imprimir Pines

Entrada	Resultados	Condiciones
La Operadora de Impresión de Pines selecciona el o los lote(s) que desea imprimir.	El Sistema muestra la cantidad de pines seleccionados a imprimir y activa la opción de "Imprimir".	La Operadora de Impresión de Pines debe estar autenticada al sistema y debe haber seleccionado la opción Imprimir Pines.
La Operadora de Impresión de Pines selecciona la opción "Imprimir".	El Sistema muestra el mensaje "¿Está seguro que desea Imprimir?".	La Operadora de Impresión de Pines debe haber seleccionado el o los lote(s) que desea imprimir.

La Operadora de Impresión de Pines selecciona la opción "Aceptar".	El Sistema muestra un formulario con los lotes que tengan el estado pin "Impreso" y la opción "Marcar Todos".	La Operadora de Impresión de Pines debe haber seleccionado la opción "Imprimir".
--	---	--

Caso de uso: CU_ Reimprimir y Finalizar Pines, Sección Finalizar Pines

Entrada	Resultados	Condiciones
La Operadora de Impresión de Pines selecciona el o los lote(s) que desea finalizar.	El sistema activa la opción "Finalizar".	La Operadora de Impresión de Pines debe haber seleccionado la opción "Aceptar".
La Operadora de Impresión de Pines selecciona la opción "Finalizar".	El Sistema muestra el mensaje "Si finaliza no podrá volver a imprimir ninguno de los pines que forman el lote o los lotes escogidos por usted. ¿Está seguro que desea finalizar? ".	La Operadora de Impresión de Pines debe haber seleccionado el o los lote(s) que desea finalizar.
La Operadora de Impresión de Pines selecciona la opción "Aceptar".	El Sistema muestra el mensaje "El proceso de impresión ha concluido satisfactoriamente".	La Operadora de Impresión de Pines debe haber seleccionado la opción "Finalizar".

Caso de uso: CU_ Reimprimir y Finalizar Pines, Sección Reimprimir Pines

Entrada	Resultados	Condiciones
La Operadora de Impresión de Pines selecciona el lote que desea reimprimir.	El sistema activa la opción "Reimprimir" y "Finalizar".	La Operadora de Impresión de Pines debe estar autenticada al sistema y debe haber seleccionado la opción Reimprimir Pines.
La Operadora de Impresión de Pines selecciona la opción	El Sistema muestra por separado los datos, dándole la posibilidad a	La Operadora de Impresión de Pines debe haber seleccionado

“Reimprimir”.	la Operadora de Impresión de Pines de seleccionar el rango de pines a reimprimir o seleccionar la opción “Cancelar”.	el lote que desea reimprimir.
La Operadora de Impresión de Pines especifica el o los pines que desea reimprimir y selecciona la opción “Aceptar”.	El Sistema muestra el mensaje “¿Está seguro que desea reimprimir?”.	La Operadora de Impresión de Pines debe haber seleccionado la opción “Reimprimir”.

Caso de uso: CU_ Captar Matrices

Entrada	Resultados	Condiciones
El Administrador de BT expande el directorio raíz para buscar la carpeta que contiene el archivo de matrices.	El sistema permite desplegar el árbol de carpetas.	El Administrador de BT debe haber seleccionado la opción Captar Matrices.

Entrada	Resultados	Condiciones
El Administrador de BT selecciona el directorio raíz antes de expandirlo.	El sistema muestra el mensaje “La dirección PC: no existe”.	El Administrador de BT debe haber seleccionado la opción Captar Matrices.

Entrada	Resultados	Condiciones
El Administrador de BT selecciona la carpeta que tiene el fichero que contiene las matrices.	El sistema busca los archivos de extensión .txt existentes en la carpeta seleccionada.	El Administrador de la BT debe haber expandido el directorio raíz para buscar la carpeta que contiene el archivo de matrices.

Entrada	Resultados	Condiciones
El Administrador de BT	El sistema muestra el mensaje	El Administrador de la BT debe

selecciona la torre de discos 3½ (Floppy) y no hay disco.	“La dirección: A no existe”	haber expandido el directorio raíz para buscar la carpeta que contiene el archivo de matrices.
---	-----------------------------	--

Entrada	Resultados	Condiciones
El Administrador de BT selecciona la torre de DVD y no hay disco.	El sistema muestra el mensaje “La dirección: F no existe”	El Administrador de la BT debe haber expandido el directorio raíz para buscar la carpeta que contiene el archivo de matrices.

Entrada	Resultados	Condiciones
El Administrador de BT selecciona el fichero que contiene las matrices.	El Sistema activa la opción “Encriptar y Guardar”.	El Administrador de BT debe haber seleccionado la carpeta que tiene el fichero que contiene las matrices.

Entrada	Resultados	Condiciones
El Administrador de BT selecciona la opción “Encriptar y Guardar”.	El Sistema muestra el mensaje: “Se captaron: X nuevas matrices”.	El Administrador de BT debe haber seleccionado el fichero que contiene las matrices.

Entrada	Resultados	Condiciones
El Administrador de BT selecciona la opción “Encriptar y Guardar”.	El Sistema muestra el mensaje: “El fichero especificado está vacío”.	El Administrador de BT debe haber seleccionado el fichero que contiene las matrices.

Entrada	Resultados	Condiciones
El Administrador de BT selecciona la opción “Encriptar y Guardar”.	El Sistema muestra el mensaje: “No se pueden encriptar”.	El Administrador de BT debe haber seleccionado el fichero que

Guardar”.	las matrices del fichero especificado debido a que el formato no es correcto”.	contiene las matrices.
-----------	--	------------------------

Entrada	Resultados	Condiciones
El Administrador de BT selecciona la opción “Encriptar y Guardar”.	El Sistema muestra el mensaje:”El fichero contiene matrices que ya existen”.	El Administrador de BT debe haber seleccionado el fichero que contiene las matrices.

Caso de uso: CU_Enviar Conciliaciones Auxiliares

Entrada	Resultados	Condiciones
El Administrador de BT selecciona uno de los Bancos mostrados.	El Sistema muestra un calendario y Activa la opción “Aceptar”	El Administrador de BT debe estar autenticado y debe haber seleccionado la opción Conciliaciones Auxiliares.
El Administrador de BT selecciona la opción de “Aceptar” realizar la conciliación de tarjeta activa.	El Sistema obtiene los números de las tarjetas activas en esa fecha y para ese banco y activa la opción de “Enviar”.	El Administrador de BT debe haber seleccionado uno de los bancos mostrados.

Caso de uso: CU_ Realizar Reporte

Entrada	Resultados	Condiciones
La Operadora de Autenticación selecciona la opción “Lotes Impresos”.	El sistema Activa la opción “Aceptar”.	La Operadora de Autenticación debe estar autenticada al sistema y debe haber seleccionado la opción Realizar Reporte.

La Operadora de Autenticación selecciona la opción "Aceptar".	El Sistema obtiene los Lotes que tengan el estado de tarjetas o de pines "Impreso" en dependencia del rol que solicite y muestra sus números en un formulario.	La Operadora de Autenticación debe haber seleccionado la opción "Lotes Impresos".
La Operadora de Autenticación selecciona un Lote.	El sistema Activa la opción "Aceptar".	La Operadora de Autenticación debe haber seleccionado la opción "Aceptar".
La Operadora de Autenticación selecciona la opción "Aceptar".	El Sistema muestra el reporte y brinda la opción de imprimirlo.	La Operadora de Autenticación debe haber seleccionado un Lote.

Entrada	Resultados	Condiciones
La Operadora de Autenticación selecciona la opción "Lotes por Imprimir".	El sistema Activa la opción "Aceptar".	La Operadora de Autenticación debe estar autenticada al sistema y debe haber seleccionado la opción Realizar Reporte.
La Operadora de Autenticación selecciona la opción "Aceptar".	El Sistema muestra el reporte y brinda la opción de imprimirlo.	La Operadora de Autenticación debe haber seleccionado la opción "Lotes por Imprimir".

4.4 Conclusiones

En este capítulo se realizó el modelo de implementación, donde se confeccionaron el diagrama de componente y el diagrama de despliegue los que constituyeron la parte final de la modelación del módulo de Gestión de Autenticación de los Clientes del Proyecto Telebanca. También se desarrollaron las pruebas de calidad a nuestro sistema, específicamente la prueba de caja negra, la cual es realizada a la interfaz del software.

CONCLUSIONES

Al finalizar la etapa correspondiente a la implementación y prueba del sistema, se arriba a las siguientes conclusiones:

Los procesos bancarios evolucionan a la par que lo hacen las tecnologías de la información y las comunicaciones, por ello que cada día soportan nuevas formas de pago y por tanto nuevos y más sofisticados métodos de autenticar a sus clientes.

Las técnicas de criptoanálisis mejoran cada día y ello hace necesario la creación de nuevos y mejores métodos de cifrado para lograr seguridad en los sistemas.

Los servicios web ofrecen ilimitadas potencialidades para la integración de aplicaciones, reduciendo el costo en desarrollo y mantenimiento de software para las empresas que utilicen este tipo de componente software.

Con el estudio realizado a lo largo de todas las etapas de desarrollo y el sistema que se ha logrado implementar, se cumple con el objetivo general de este trabajo: desarrollar un módulo que automatice los procesos de gestión de autenticación de los clientes asociados a TeleBanca en un espacio de tiempo fiable, con alta calidad y correctamente documentado.

RECOMENDACIONES

Hechas las conclusiones del trabajo, se recomienda:

- Integrar todos los módulos del proyecto TeleBanca con vistas a obtener un sistema que funcione como un todo.
- Desarrollar nuevos mecanismos de autenticación que resulten más comprensibles y fáciles de usar a toda la población en general.
- Estudiar los nuevos estándares de cifrado que surjan en los próximos años con vistas a actualizar el sistema y garantizar la seguridad de los datos.
- Desarrollar una segunda versión que incluya nuevas funcionalidades como por ejemplo un sistema más completo de búsquedas y generación de reportes.
- Continuar con la investigación para garantizar nuevas mejoras en futuras versiones del sistema.

REFERENCIAS BIBLIOGRÁFICAS

Álvarez, Sofía and Hernández, Anaisa. 2000. *Metodología para el desarrollo de aplicaciones con tecnología Orientada a Objetos utilizando notación UML.* La Habana : s.n., 2000.

Muñoz Muñoz, Alfonso. 2004. *Criptosistema Rjindael a fondo. Algoritmo Criptográfico Rjindael.* Madrid : s.n., 2004.

Pressman, Roger S. 2001. *"Ingeniería de Software, un enfoque práctico".* Quinta Edición. s.l. : McGraw-Hill, 2001.

Sanchez, María A. Mendoza. 2004. <http://www.informatizate.net>. *Metodologías De Desarrollo De Software.* [Online] Junio 7, 2004. [Cited: Febrero 4, 2007.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

BIBLIOGRAFÍA

Criptografía. [Disponible en:

<http://www.textoscientificos.com/criptografia/cripto>

Criptografía moderna. [Disponible en:

http://www.terra.es/personal6/morenocerro2/seguridad/cripto/cripto_2.html

Criptografía: métodos clásicos 2007. [Disponible en:

<http://criptociencia.blogspot.com/2007/03/criptografa-mtodos-clscicos.html>

Criptología. 2006. [Disponible en:

<http://www.segu-info.com.ar/criptologia/criptologia.htm>

Criptología - Algoritmos Simétricos Modernos (Llave Privada) 2006. [Disponible en:

<http://www.segu-info.com.ar/criptologia/simetricos.htm>

Definición de banco. Disponible en:

<http://www.definicion.org/banco>

Rational Unified Process. 2007. [Disponible en:

http://whatis.techtarget.com/definition/0,,sid9_gci810206,00.html

Seguridad y algoritmos de encriptación. 2006. [Disponible en:

<http://www.cryptoforge.com.ar/seguridad.htm>

The Unified Process can anchor your development efforts. 2002. [Disponible en:

<http://builder.com.com/5100-6315-1044872.html>

What is the Unified Modeling Language?, 2000. [Disponible en:

http://microgold.com/Stage/UML_FAQ.html#UML1

AVILEZ, J. *Bancos*. Disponible en:

<http://www.monografias.com/trabajos11/bancs/bancs.shtml>

BRAUN, D.; J. SIVILS, *et al.* *What is UML?*, 2001. [Disponible en:

http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/index.htm

CORREA, E. E. E. *Criptografía Simétrica - Parte 1*, 2006. [Disponible en:

http://www.elguille.info/colabora/NET2006/ericeec_CriptografiaSimetricaParte1.htm

EDUARD LLULL. *BULMA: Criptografía – Data Encryption Standar (DES)*, 2003.

KRAFFT, A. R. *Criptografía: ciencia de la seguridad* 2003. [Disponible en:

<http://www.alambre.info/2003/12/22/criptografia-ciencia-de-la-seguridad/>

KRUCHTEN, P. *Rational Unified Process*, Addison Wesley ProfessionalPub

MARAÑÓN, G. Á. *El nuevo estándar de cifrado*, 2000 [Disponible en:

<http://www.iec.csic.es/criptonomicon/susurros/susurros25.html>

MENDOZA, M. A. S. *Metodologías De Desarrollo De Software*, 2004. [Disponible en:

http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html

MUÑOZ, A. M. *CRIPTOSISTEMA RIJNDAEL. A FONDO*, 2004.

Crystal Reports 10 .NET 2003 [Disponible en:

<http://www.willydev.net/CrystalDesde0/>

¿Qué es ASP.NET? , 2001. [Disponible en:

<http://es.gotdotnet.com/quickstart/aspplus/doc/whatisaspx.aspx>

ALVAREZ, R. *Introducción a la programación en ASP*, 2004. [Disponible en:

<http://www.desarrolloweb.com/articulos/244.php>

BROBERG, J. C. *Glossary for the OASIS WebService Interactive Applications (WSIA/WSRP)*. [Disponible en:

www.oasis-open.org/committees/wsia/glossary/wsia-draft-glossary-03.htm

C, B. G. *Una visión general I*, 2001. [Disponible en:

<http://www.desarrolloweb.com/articulos/1537.php>

CERAMI, E. *web Services essentials* O'Reilly, 2002. 304 p. 0-596-00224-6

ESCOVAR, M. *XML Web Services*, 2003. [Disponible en:

<http://www.desarrolloweb.com/articulos/1545.php>

PARIHAR, M. *ASP.NET*. 2003. p. 8441513856

PÉREZ, J. S. *PROGRAMACIÓN CON ASP.NET*. 2002. 960 p.

RAMÍREZ, E. M. *Introducción a los Web Services* 2001. [Disponible en:

<http://www.desarrolloweb.com/articulos/1852.php>

RAMÍREZ, I. E. M. *Lo que debo saber sobre Crystal Reports y VB .NET*, 2007. [Disponible en:

<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art192.asp#Conc>

RODRÍGUEZ, N. B. *Análisis Web Services (WS) Segunda Parte (SOAP, WSDL y UDDI)*, 2003.

[Disponible en:

<http://www.eusa.net/pages/research/web-services-2.do;jsessionid=771D8082C566966F2CEBF2237EE0213B>

SANT'ANNA, M. Reportes en .NET con Crystal Reports, 2003. [Disponible en:
<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art11.asp>

Autenticación. 2005 [Disponible en:

<http://www.aichile.org/glosario.htm>

LA AUTOMATIZACIÓN DEL SISTEMA BANCARIO CUBANO Disponible en:

http://www.cubagob.cu/des_eco/banco/espanol/automatizacion/automatizacion.htm

MEDIOS ELECTRÓNICOS DE PAGOS. 2007. 16 N°3-2005: 71-80.

ALONSO, I.; T. PRADAS, *et al.* *EL COMERCIO ELECTRÓNICO EN CUBA*, 2002. [Disponible en:

http://www.eleconomista.cubaweb.cu/2002/nro160/160_308.html

BAZOBERRY, E. *Tarjetas Inteligentes Multilingües (ATM) en Bolivia*, 2001.

[Disponible en:

<http://www.protic.org/proyectos.shtml?apc=y2R-1--&x=20172379>

CORZO, I. E. *MICROPROCESADORES PARA DINERO ELECTRÓNICO*.

Disponible en:

<http://neutron.ing.ucv.ve/revista-e/No3/SmartCard.html>

GARCÍA, A. V. *Seguridad en los nuevos medios de pago*, 2000. [Disponible en:

<http://www.iec.csic.es/criptonomicon/articulos/expertos30.html>

GARCÍA, J. O. *Modalidades en los Sistemas de Autenticación Biométrica*.

Disponible en:

<http://gavab.escet.urjc.es/recursos/JOrtega04.pdf>

MOREJON, A. R. *SISTEMA FINANCIERO DE CUBA*, 2002. [Disponible en:

http://www.eleconomista.cubaweb.cu/2002/nro151/151_259.html

OLIVERA, P. L. *Banco Popular de Ahorro: lo primero es el cliente*, 2005. [Disponible en:

<http://www.cadenagramonte.cubaweb.cu/economia/141005.asp>

SIBANC. *Automatización*, 2003. [Disponible en:

<http://www.bc.gov.cu/Espanol/automatizacion.asp>

GLOSARIO DE TERMINOS

Cliente: Persona que utiliza el servicio brindado por las Empresas (En este caso es el que accederá al pago de los mismos a través de la Banca Telefónica).

Centro de Llamadas: Lugar único donde se recibirán todas las llamadas telefónicas, las operadoras ejecutarán las transacciones a partir de las indicaciones recibidas por el cliente.

IDE: Un entorno de desarrollo integrado o en inglés Integrated Development Environment (IDE) es un programa compuesto por un conjunto de herramientas para un programador. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

Pago de Servicios: Pago de un cliente a una Empresa por el servicio recibido o deuda contraída con la misma.

EMV: Consorcio formado por Visa, Europago y Master Card creado con la finalidad de crear las especificaciones industriales que permitirán estandarizar las tarjetas inteligentes y los lectores, garantizando así la universalidad de su uso, independientemente del fabricante o del emisor de la tarjeta.

Servicios de Banca Telefónica: Servicios que pueden ser pagados a través de la modalidad de Banca Telefónica (Agua, Luz, Teléfono, etc.).

Tarjetas de autenticación: Tarjeta necesaria para poder acceder el cliente a los servicios de Banca Telefónica. Contribuye a la seguridad del sistema.

Matrices: Cuadro compuesto con números colocados en filas y columnas. La matriz se imprime en las tarjetas de autenticación del servicio de Banca Telefónica.

PIN: Numero de Identificación Personal (clave) asociado a una tarjeta.

Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador): Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

XML: eXtensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho

mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.