Pontificia Universidad Católica de Valparaíso

Facultad de Ingeniería

Escuela de Ingeniería Informática

CLIENTE SEGURO.

TRANSACCIONES SEGURAS Y CONFIDENCIALES EN LA RED.

Autor:

Rodrigo Edmundo Sanza Pezoa

Informe final del Proyecto para optar al Título profesional de

Ingeniero de Ejecución en Informática

Profesor guía:

Iván Mercado Bermudes

Agosto 2006

Este trabajo va dedicado a mis padres, que siempre me brindaron el apoyo necesario para poder hacer de mí, una persona de bien.

Resumen

“Cliente Seguro” es una aplicación creada para mejorar la seguridad del comercio electrónico. Esta está orientada a ser una herramienta ofrecida por una entidad bancaria a sus clientes, a los cuales les permita realizar compras en Internet, sin temores.

“Cliente Seguro” basa su singularidad en que no necesita entregar datos personales a terceros y realiza su objetivo de forma confidencial y segura.

El desarrollo de esta aplicación, fue orientado bajo una metodología de trabajo basada en el paradigma de espiral y sustentado por pruebas y análisis constantes en su lógica de funcionamiento.

Abstract

“Safe Client” is a make application to improve the security of the electronic commerce. This is oriented to being a tool offered by a banking organization its clients, to who it allows them to make purchases in Internet, without fears.

“Safe Client” bases his singularity on which she does not need to give personal data to third and makes his objective of confidential and safe form.

The development of this application, was oriented under a methodology of work based on the paradigm of spiral and sustained by tests and constant analyses on its logic of operation.

CAPITULO I

Introducción

1. Introducción

En el presente informe se documenta los procedimientos y resultados de un trabajo de investigación en donde el objetivo buscado, fue crear una herramienta segura, anónima y confiable para el usuario de Internet al momento de realizar transacciones financieras en línea.

Debido a la desconfianza que existe por parte del público al momento de entregar datos que podrían ser usados de forma maliciosa por terceros, el comercio electrónico se ha visto afectado en forma drástica en su crecimiento esperado.

“Cliente Seguro”, es la alternativa desarrollada para el cliente actual, ya que su forma de trabajar es fácil de entender y fácil de usar.

A lo largo de este informe, se detallan los objetivos buscados con la investigación, se justifica el por qué de la inquietud planteada, se hace hincapié en lo que debe poseer el software a desarrollar, se dan a conocer algunas alternativas existentes en la actualidad y se presenta los resultados obtenidos. También se describen las herramientas ocupadas en su desarrollo y el porque de su elección.

La principal característica de esta aplicación, es permitir al usuario realizar compras en Internet sin necesidad de entregar datos personales o financieros a la Casa Comercial con la cual se realiza la transacción. “Cliente Seguro” posibilita ejecutar una transacción anónima ante la casa comercial, pero con el respaldo de una entidad bancaria. Así mismo, la mencionada entidad bancaria, no tiene necesidad de conocer el producto o servicio pagado por su afiliado.

Finalmente dentro de este informe, se expone un proceso de pruebas a la aplicación, se documenta el funcionamiento y características de sus interfaces y se adjunta el código fuente completo, tanto de las interfaces, como del web service diseñado para la comunicación con la base de datos.

1.1 Información descriptiva y de contexto general

Dada la naturaleza genérica del *software* desarrollado, no se tiene una organización objetivo específica para poder analizar. Sin embargo, se puede considerar a una entidad financiera bancaria (en adelante BANCO) como potencial cliente de la solución, y por ende, la organización a describir y la coordinación con una casa comercial que ofrezca productos y/o servicios vía Web (en adelante TIENDA).

Un banco tiene muchos productos y servicios, dentro de los cuales podemos destacar cuentas corrientes y tarjetas de crédito, los que en los últimos tiempos han visto aumentada su demanda por utilización para compras en Internet.

Para que un cliente de un banco pueda ocupar sus servicios en la actualidad, se le asigna una *password* y en algunos casos, se le entrega un dispositivo de seguridad adicional, el cual consiste en tener un segundo password aleatorio y momentáneo, los cuales permiten al cliente identificarse al sistema del banco como usuario válido e iniciar las operaciones y transacciones que desee (todo lo anterior asumiendo que el banco tiene un portal de servicios habilitado vía Web), por ejemplo: consultar saldos, realizar transferencias, revisar historial de movimientos, etc.

Para hacer una introducción al sistema más usado de pago en estos momentos, se describirá el uso de las tarjetas de crédito al comprar por Internet. Más adelante, se hace referencia a otros métodos existentes.

Cuando una persona desea algún producto o servicio de una tienda, debe entregar su número de tarjeta de crédito, junto a su validador (a través de la página Web de la tienda) para que dicha tienda realice el cargo a la entidad correspondiente, lo que naturalmente representa un riesgo al interactuar con tiendas de dudosa reputación, debido a que se basa en la confianza de

que la tienda cobre solamente lo estipulado en la transacción en línea y que los datos entregados, serán bien resguardados.

Eventualmente, la interacción de una persona con un banco o una tienda, es a través de una *interfaz* Web, la cual, solicita los datos de validación de la tarjeta de crédito. En ese momento, el usuario pasa a estar expuesto una serie de posibles agujeros de seguridad entre su PC y el Servidor que recepciona los datos por parte de la tienda.

1.2 Objetivos y Justificación del proyecto.

1.2.1 Objetivo General

- Desarrollar un *software genérico*, el cual permita realizar transacciones seguras y confiables por parte del usuario común y desde cualquier computador.

1.2.2 Objetivos Específicos

Acumular información sobre dinero electrónico y cómo se está usando.

Recopilar información de problemas de seguridad en los métodos utilizados en la actualidad en transacciones comerciales por Internet.

Entrelazar las alternativas existentes en el mercado en cuanto a seguridad en transacciones financieras, en una sola solución.

Que la solución anteriormente descrita sea económicamente factible de implementar para un Banco y así en sus clientes.

1.2.3 Justificación del proyecto

El hecho de que los usuarios aun presenten reticencia a la hora de entregar sus datos, de tarjeta de crédito o personales por Internet, ha provocado que el comercio electrónico no haya crecido de la forma exponencial en la que debería.

Al contar con una alternativa fácil de entender por el cliente en su forma de uso y en la cual pueda depositar su plena confianza, se espera un crecimiento exponencial en este tipo de

transacciones y a su vez un incremento fuerte en las ganancias para el Banco que la implemente.

1.3 Especificación de Requerimientos.

Para hacer viable este proyecto, se debe estar en condiciones de garantizar la satisfacción de una serie de requerimientos, los que se clasifican en los siguientes grupos:

1.3.1 Requerimientos Funcionales:

El sistema debe autenticar al usuario antes de realizar cualquier función.

Debe otorgar al usuario la posibilidad de solicitar comprobantes de montos, como también de anularlos en caso de no ser cobrados.

El usuario debe ser capaz de visualizar en pantalla los datos vinculados con la empresa que hizo el cobro de un comprobante.

Los montos solicitados para comprobantes, deben ser descontados de la cuenta que corresponda y confirmado el descuento una vez cobrado o restituidos en caso de ser anulado.

1.3.2 Requerimientos No Funcionales:

Se exigirá que la comunicación entre las partes sea lo suficientemente rápida como para no presentar molestias de espera en los clientes (no más allá de 5 segundos).

Se exigirá que los datos manejados y transferidos sean inviolables, y en el caso de ser interceptados sean inútiles para terceros.

El sistema en sí, debe ser fácil de entender para un usuario común.

1.3.3 Requerimientos de implementación:

El software debe ser posibles de instalar y ejecutar en los sistemas que usan actualmente los interesados.

1.4 Estado del Arte.

Al analizar las tecnologías de seguridad actuales del mercado, se puede encontrar, que todas presentan grandes avances y certeras soluciones a algunos agujeros de seguridad presentes en la Internet. Pero, sin embargo, a la hora de implementar soluciones para situaciones concretas, la complementación de estas, deja mucho que desear.

El número de usuarios que acude a la banca on-line para realizar sus consultas y transacciones financieras por Internet se incrementó en un 20% en el los últimos años, al pasar de 1,9 millones de personas en febrero de 2004 a un total de 2,3 millones en idéntico periodo del año 2005, según revela el panel de audiencia de Nielsen//NetRatings, compañía especializada en la medición y análisis de audiencias en Internet.

- ***Cybercash***

Cybercash se fundó en agosto de 1994 por Bill Melton quién previamente había fundado Verifone y Transactions Network Systems y Dan Lynch fundador de Interop. La compañía se creó con el objetivo de ofrecer transacciones económicas seguras a través de Internet incluyendo transacciones seguras en tarjetas de crédito y cheques electrónicos. Actualmente Cybercash es la única compañía internacional con licencia de exportación que ofrece un *algoritmo de encriptación* de 1024 bits RSA ofreciendo servicios de autentificación de tarjetas de crédito basadas en firma electrónica en tiempo real. Algunas aplicaciones como los comercios electrónicos de Interplanet ofrecen el método Cybercash en transacciones económicas a través de Internet sin necesidad de utilizar líneas de teléfono adicionales ni otros servicios tan sólo utilizando el mismo software.

Funcionamiento: cuando un cliente envía su número de tarjeta de crédito a una empresa "X", además de la verificación de fondos del cliente se realiza una verificación del domicilio y código postal que fueron enviados por el cliente, Cybercash en este caso, envía otro código (además de la autorización) el denominado AVS "Address Verification" el cual puede ser:

- **Y :** "Domicilio y Código Postal coinciden con Tarjeta"

- **A :** "Domicilio válido , Código Postal no coincide"

- **Z :** "Código postal válido , Domicilio no coincide"

- **N :** "Domicilio y Código Postal no coinciden "

- **R :** "Sistema no disponible"

- **S :** "Tarjeta no válida en el sistema "

A partir de este punto es a criterio de la Empresa ("X") permitir el cargo a la tarjeta, claro está que si permiten cargos a códigos **N** tanto ellos como su Banco ("BX") pueden enfrentar reclamos, aunque también cabe señalar que el hecho de Autorizar sólo códigos **Y** no garantiza ninguna reclamación.

- ***SET***

El sistema, denominado "SET fácil", permite al usuario titular de una tarjeta de monedero virtual (Virtual Cash) descargar en poco tiempo el programa de cartera digital SafeWallet (creado por la empresa SafeLayer y licenciado por Banesto) y solicitar un certificado personal a la entidad bancaria, que emite en línea un certificado electrónico SET. Acto seguido, el consumidor puede utilizar su certificado para realizar compras seguras en las tiendas que utilicen la terminal punto de venta (TPV) de Banesto [un programa que gestiona al comerciante la venta de productos en Internet].

Tecnicismo del funcionamiento:

- **Certificar las tarjetas de crédito en manos de los consumidores**, mediante técnicas criptográficas electrónicas. Se proveen pequeños programas a modo de "firma digital"

(que se incluyen ya en los propios navegadores de Internet), permitiendo pagar con alguna tarjeta siempre y cuando haya sido previamente certificada.

- **Certificar a los comercios** que lo deseen, habilitándolos para establecer operaciones de pago SET a través de Internet. Al comercio se le suministra un pequeño programa informático que le permite almacenar de modo seguro su certificación al tiempo que puede establecer una conexión y validar la operación entre él, el consumidor y la entidad financiera.

- Establecer *protocolos* estándares de intercambio entre las entidadades financieras.

- La **autentificación** de que todos los participantes en la operación sean quienes dicen ser, tanto desde el punto de vista del consumidor como del comercio.

- La **privacidad** de cada participante, es decir, sólo verán la información que les corresponda.

- La **integridad** de la transacción. Esta no puede ser alterada ni duplicada. El consumidor autoriza una transacción en concreto, para un comercio en concreto y sin alterar los importes de la transacción.

- ***Tarjeta Monedero***

El uso de este protocolo de seguridad está estrechamente relacionado con la llamada Virtual Cash Plus, una nueva versión de la tarjeta de monedero electrónico de Banesto, que permite ser recargada en un cajero de la red de 4B utilizando una tarjeta de débito expendida por

cualquier entidad bancaria, y utilizarla luego para realizar compras en Internet. Es usada principalmente en todas aquellas situaciones en las que hasta ahora se hace necesaria la disponibilidad de moneda fraccionaria, simplificando las operaciones de pago, en este tipo de comercios y evitando la problemática de la devolución de vuelto.

- ***Tarjetas de Crédito.***

Si bien el uso de las tarjetas de crédito está regido por protocolos de seguridad y también la mayoría de las casas comerciales usan el sistema de cifrado del canal por el cual se transmiten los datos (SSL). En el fondo, el usuario debe confiar en que la casa comercial a la cual está entregando sus datos financieros, tendrá una *base de datos* segura para guardarlos y a su vez, por ningún motivo hará mal uso de ellos, ya que el sistema de tarjetas de crédito, permite una flexibilidad en los montos a cobrar, que quedan al total criterio y honradez de la casa comercial con la cual se realiza la transacción.

Además muchas veces, los usuarios, no toman las debidas precauciones de comprobación de esta casa comercial antes de entregar sus datos. En muchos sitios de Internet, se recomiendan algunos puntos básicos a tener en cuenta, antes de confiar en una casa comercial y aquí se detallan algunos de ellos:

Comprobar el prestigio y trayectoria de la empresa.
Verificar que posea un sistema de canal *encriptado* para el traspaso de datos (SSL).
Asegurar que tengan una dirección física y no solo una Virtual.

- ***Tarjeta de Crédito Virtual.***

Esta opción, es ofrecida por algunos Bancos para sus clientes que poseen alguna Tarjeta de Crédito de su compañía.

El sistema funciona de la siguiente forma:

Se selecciona el tipo de compra que se realizará (nacional o internacional).
Se elije la Tarjeta de Crédito desde la que se creará una “Tarjeta Virtual”.

Se le asigna un cupo a la nueva tarjeta. El sistema creará una nueva tarjeta, la que poseerá una nueva fecha de vencimiento, nuevo código de verificación y nueva numeración.

Una vez realizada la compra, el monto se descontará desde la tarjeta original.

CAPITULO II

Entorno y herramientas.

2. Entorno y herramientas.

2.1 Herramientas de desarrollo

2.1.1 Visual Basic .NET

Visual Basic es un *lenguaje de programación* desarrollado por Alan Cooper para Microsoft. El lenguaje de programación es un dialecto de BASIC, con importantes añadidos. Su primera versión fue presentada en 1991 con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y en cierta medida también la programación misma. Su versión **.NET** está orientada a la construcción de aplicaciones Web, como así también, crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de *plataforma* y que permita un rápido desarrollo de *aplicaciones*.

Por estas razones y dado que es una herramienta compatible con la mayoría de las plataformas usadas en el mundo, se escogió para ser la base de desarrollo de este software. Con esta herramienta, se desarrollaron las Interfaces de uso del sistema, como así también, todo el código de funcionamiento de sus distintas funciones.

2.1.2 SQL Server 2000

SQL Server es una plataforma global de base de datos que ofrece administración de datos empresariales con herramientas integradas de inteligencia empresarial. El motor de la base de datos SQL Server 2000 ofrece almacenamiento seguro y confiable tanto para datos relacionales como estructurados, lo que permite crear y administrar aplicaciones de datos altamente disponibles y con muy buen rendimiento.

Por la gran compatibilidad e integración que existe entre Visual Basic y SQL Server, además de la abundante información con respecto a su uso y ejemplos disponibles en Internet, se ha seleccionado esta base de datos, para servir de centro de recopilación de datos para este trabajo.

2.2 Herramientas de trabajo.

2.2.1 Internet Information Server

Es el *servidor* Web de Microsoft que corre sobre plataformas Windows. Dado que el IIS está tan íntimamente integrado con el sistema operativo, es relativamente fácil de administrar. Este servicio convierte a un computador en un servidor de Internet o *Intranet* es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente (servidor Web). El servidor Web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

En lo concerniente a este trabajo, se escogió Internet Information Server como servidor para las páginas Web y Web Service que se usan en la aplicación, por ser parte innata de la plataforma Microsoft, la cual se ocupa junto con las demás herramientas del trabajo. Además se usará la funcionalidad de *SSL* (Secure Socket Layer) que ofrece esta herramienta, para establecer los canales de comunicación entre las partes involucradas.

2.2.2 Triple Des

En criptografía el Triple DES se llama al algoritmo que hace triple cifrado del *DES*. También es conocido como TDES, fue desarrollado por IBM en 1978. Cuando se descubrió que una clave de 56 bits no era suficiente para un ataque de fuerza bruta, TDES fue elegido como forma de agrandar el largo de la llave sin necesidad de cambiar de algoritmo de encriptación. Este método de cifrado es inmune al ataque por encuentro a medio camino, doblando la longitud efectiva de la clave, pero en cambio es preciso triplicar el número de operaciones de cifrado, haciendo este método de cifrado muchísimo más seguro que el DES.

No llega a ser un cifrado múltiple, porque no son independientes todas las subclases. Este hecho se basa en que DES tiene la característica matemática de no ser un grupo, lo que implica que si se encripta el mismo bloque dos veces con dos llaves diferentes se aumenta el tamaño efectivo de la llave.

La variante más simple del Triple DES funciona de la siguiente manera:

$$C = E_{DES}^{k3}\left(D_{DES}^{k2}\left(E_{DES}^{k1}(M)\right)\right)$$

Donde M es el mensaje a encriptar y k1, k2 y k3 las respectivas llaves DES.

Por ser el estándar usado por la mayoría de los medios de pago electrónico y ser una de las mejores alternativas de seguridad a la feche, en lo concerniente a la encriptación de información, Triple des fue usado en este trabajo.

2.2.3 MD5

En criptografía, MD5 (acrónimo de *Message-Digest Algorithm 5*, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado. El código MD5 fue diseñado por Ronald Rivest en 1991. Los resúmenes MD5 se utilizan extensamente en el mundo del software para proporcionar la seguridad de que un archivo descargado de Internet no se ha alterado. Comparando una suma MD5 publicada con la suma de comprobación del archivo descargado, un usuario puede tener la confianza suficiente de que el archivo es igual que el publicado por los desarrolladores. Esto protege al usuario contra los 'Caballos de Troya' o 'Troyanos' y virus que algún otro usuario malicioso pudiera incluir en el software. La comprobación de un archivo descargado contra su suma MD5 no detecta solamente los archivos alterados de una manera maliciosa, también reconoce una descarga corrupta o incompleta.

La codificación del MD5 de 128 bits es representada típicamente como un número de 32 dígitos hexadecimal. En el siguiente ejemplo el código de 28 bytes ASCII será tratado con MD5 y se verá su correspondiente hash de salida:

MD5("Esto si es una prueba de MD5") =e07186fbff6107d0274af02b8b930b65

Un simple cambio en el mensaje da un cambio total en la codificación hash, en este caso se cambiaron dos letras, el "si" por un "no".

MD5("Esto no es una prueba deMD5")=dd21d99a468f3bb52a136ef5beef5034

Otro ejemplo sería la codificación de un campo vacío:

MD5("") = d41d8cd98f00b204e9800998ecf8427e

Al momento de desarrollar esta aplicación, MD5 es la forma más segura conocida, de conseguir un resumen Hash, por lo tanto será usada en las partes necesarias.

2.2.4 Web Services Enhancements 2.0 (WSE)

WSE es una implementación de diferentes especificaciones WS-*. Los servicios Web ofrecen comunicación para sistemas con diferentes sistemas operativos y plataformas de desarrollo. Para conseguir este objetivo, se basan en una familia de especificaciones de protocolos industriales para los servicios Web, que generalmente se denomina WS-*. WSE 2.0 proporciona implementaciones de muchas de estas especificaciones.

Web Services Security y WS-Addressing proporcionan seguridad de extremo a extremo en el nivel de los mensajes. La especificación WS-Security, que ha sido recientemente ratificada como estándar, describe la forma de asegurar los servicios Web en el nivel de los mensajes, en lugar de en el del protocolo de transferencia o en el de la conexión. Las soluciones en el nivel de transporte actuales, como SSL/TLS, proporcionan un sólido cifrado y autenticación de datos punto a punto, aunque presentan limitaciones cuando un servicio intermedio debe procesar o examinar un mensaje. Por ejemplo, un gran número de organizaciones implementan un firewall que realiza un filtrado en el nivel de la aplicación para examinar el tráfico antes de pasarlo a una red interna.

Si un mensaje debe pasar a través de varios puntos para llegar a su destino, cada punto intermedio debe reenviarlo a través de una nueva conexión SSL (figura 1). En este modelo, el mensaje original del cliente no está protegido mediante cifrado puesto que atraviesa

servidores intermedios y para cada nueva conexión SSL que se establece se realizan operaciones de cifrado adicionales que requieren una gran cantidad de programación.

Figura 1. Seguridad en el nivel del protocolo frente a seguridad en el nivel de los mensajes

WS-Addressing, desempeña un papel fundamental en la seguridad en el nivel de los mensajes, puesto que proporciona los mecanismos para enviar los mensajes de un modo independiente del transporte. Esto permite enviar un mensaje seguro a través de cualquier transporte y enrutarlo con facilidad.

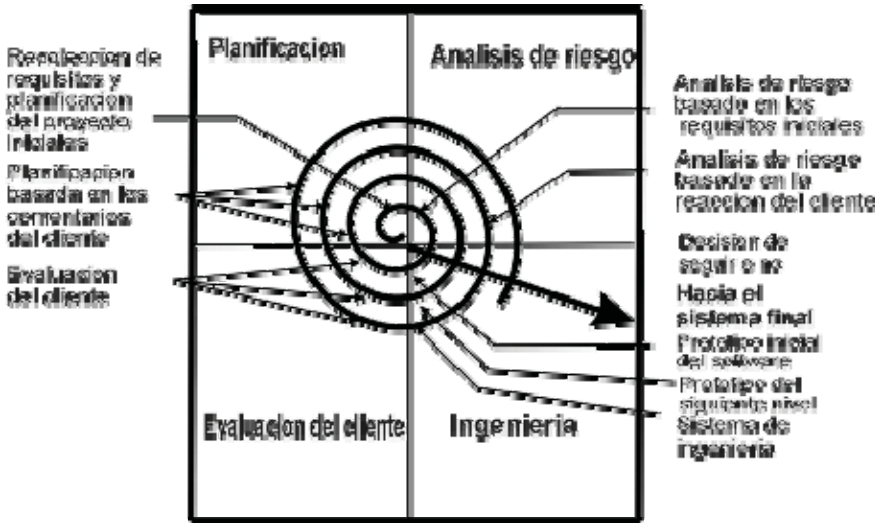
La protección del mensaje en lugar del uso del protocolo de transporte ofrece varias ventajas. En primer lugar, resulta más flexible puesto que se pueden firmar o cifrar partes del mensaje en lugar del mensaje completo. De este modo, los intermediarios pueden ver las partes del mensaje destinadas a ellos. Un ejemplo de esto sería un servicio Web que enruta mensajes SOAP y puede inspeccionar las partes no cifradas de los mismos para determinar a dónde enviarlos, mientras que otras partes permanecen cifradas. En segundo lugar, los intermediarios pueden agregar sus propios encabezados al mensaje y firmarlos para llevar a cabo el registro de auditorias. Por último, esto implica que el mensaje protegido se puede enviar a través de diferentes protocolos, como SMTP, FTP y TCP, sin necesidad de basarse en el protocolo para la seguridad.

2.3 Metodología

2.3.1 Paradigma de trabajo.

Debido a la naturaleza del proyecto (investigación), se ha decidido utilizar una metodología acorde con el mismo, por lo tanto, se regirá por el **Paradigma Espiral**.

Esto se puede entender ya que el paradigma Espiral, permite evaluaciones iterativas de los requisitos, factor muy importante a la hora de ir descubriendo material nuevo y relevante para la investigación. Como así también, debido al progreso consecuente con el tiempo y la puesta en práctica de los conocimientos adquiridos con el estudio que se vaya realizando.



"Figura1, Paradigma Espiral"

1. **Planificación:** determinación de objetivos, alternativas y restricciones.
2. **Análisis de riesgo**: análisis de alternativas e identificación/resolución de riesgos.
3. **Ingeniería**: desarrollo del producto del "siguiente nivel",
4. **Evaluación del cliente**: Valorización de los resultados de la ingeniería.

Durante la primera vuelta alrededor de la espiral se definen los objetivos, las alternativas y las restricciones, y se analizan e identifican los riesgos. Si el análisis de riesgo indica que hay una incertidumbre en los requisitos, se puede usar la creación de prototipos en el cuadrante de ingeniería para dar asistencia en el desarrollo.

El cliente evalúa el trabajo de ingeniería (cuadrante de evaluación de cliente) y sugiere modificaciones. Sobre la base de los comentarios del cliente se produce la siguiente fase de planificación y de análisis de riesgo. En cada bucle alrededor de la espiral, la culminación del análisis de riesgo resulta en una decisión de "seguir o no seguir".

Con cada iteración alrededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se construyen sucesivas versiones del software, cada vez más completa y, al final, al propio sistema operacional.

El paradigma del modelo en espiral para la ingeniería de software es actualmente el enfoque más realista para el desarrollo de software y de sistemas a gran escala. Utiliza un enfoque evolutivo para la ingeniería de software, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel evolutivo. Utiliza la creación de prototipos como un mecanismo de reducción de riesgo, pero, lo que es más importante permite a quien lo desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución de prototipos.

2.3.2 Lenguaje Unificado de Modelado (UML).

El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

Parte III

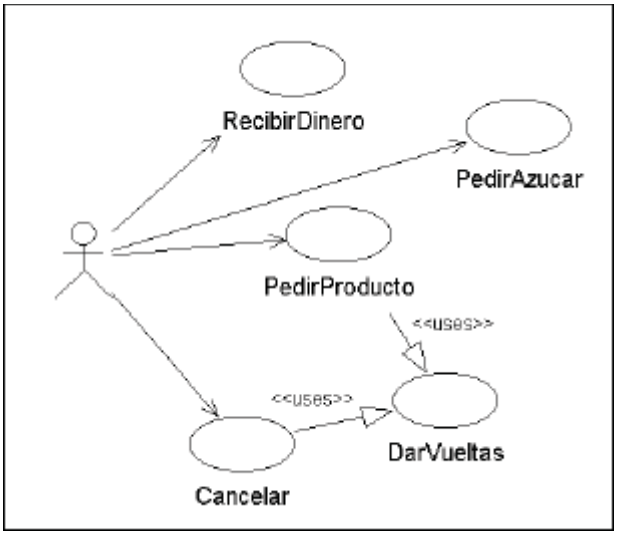
Solución planteada.

3. Entorno y Herramientas

3.1 Conceptos de un diagrama de Casos de Uso

Un diagrama de Casos de Uso muestra las distintas operaciones que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones).

Se muestra como ilustración los casos de uso de la máquina de café.



3.1.1 Caso de uso

Se representa en el diagrama por una elipse, denota un requerimiento solucionado por el sistema. Cada caso de uso es una operación completa desarrollada por los actores y por el sistema en un diálogo. El conjunto de casos de uso representa la totalidad de operaciones desarrolladas por el sistema. Va acompañado de un nombre significativo. En el caso del

ejemplo se tienen como casos de uso de la cafetera RecibirDinero, PedirAzucar, PedirProducto, DarVueltas y Cancelar.

3.1.2 Actor

Es un usuario del sistema, que necesita o usa algunos de los casos de uso. Se representa mediante un monigote, acompañado de un nombre significativo, si es necesario.

3.1.3 Relaciones en un diagrama de casos de uso

Entre los elementos de un diagrama de Casos de uso se pueden presentar tres tipos de relaciones, representadas por líneas dirigidas entre ellos (del elemento dependiente al independiente)

Comunica (communicates). Relación entre un actor y un caso de uso, denota la participación del actor en el caso de uso determinado. En el diagrama de ejemplo todas las líneas que salen del actor denotan este tipo de relación.

Incluye (incluye, uses). Relación entre dos casos de uso, denota la inclusión del comportamiento de un escenario en otro. El caso es usado, siempre que el caso que lo usa es ejecutado. En el caso del ejemplo el caso de uso Cancelar incluye en su comportamiento DarVueltas; y PedirProducto incluye también DarVueltas

Extiende (extends). Relación entre dos casos de uso, denota cuando un caso de uso es una especialización de otro. Por ejemplo, podría tenerse un caso de uso que extienda la forma de pedir azúcar, otra que permita escoger el tipo de azúcar (normal, dietético moreno) y además la cantidad en las unidades adecuadas para cada caso (cucharaditas o cubitos, etc). La principal diferencia con el “Include”, es que representa una parte de la funcionalidad del caso, que no siempre ocurre.

3.2 Diagramas de Casos de Uso Explicativos.

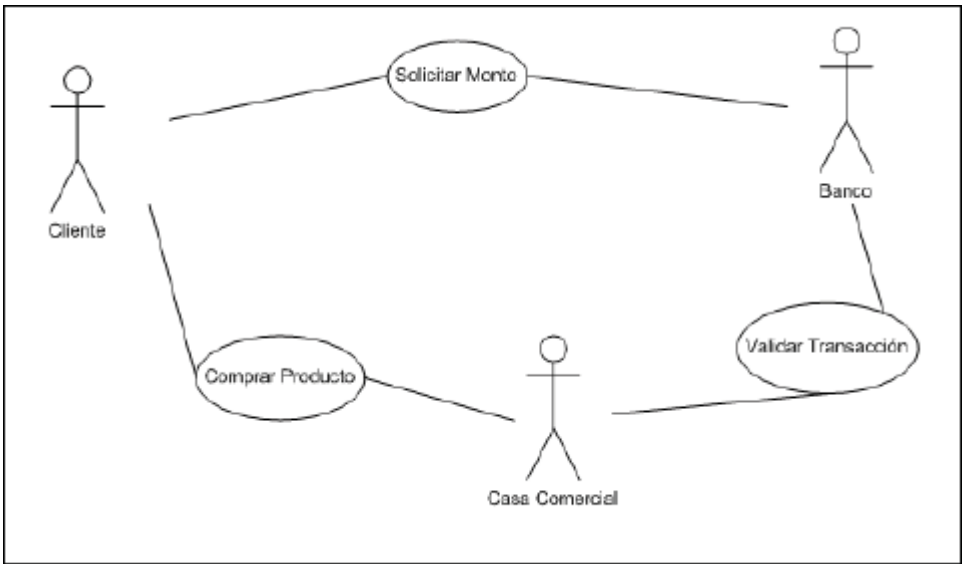


Diagrama 1: General

Caso de Uso	**Solicitar Monto**
Actores	Cliente(iniciador), Banco
Tipo	Principal
Descripción	Un Cliente visita la página web del Banco y desde esta, solicita un comprobante por el monto requerido.

Caso de Uso	**Comprar Producto**
Actores	Cliente (iniciador), Casa Comercial

Tipo	Principal
Descripción	El Cliente selecciona un producto, selecciona la forma de pago y posteriormente, entrega el comprobante y su password a la casa comercial, la página web de esta.

Caso de Uso	**Validar Transacción**
Actores	Casa Comercial (iniciador), Banco
Tipo	Principal
Descripción	La Casa Comercial, entrega al Banco, a través de su web service, el comprobante y password entregados por el cliente, además del valor del producto seleccionado y su identificación. El Banco se encarga de validar los datos y dar una respuesta a la Casa Comercial, sobre si la transacción es válida o no.

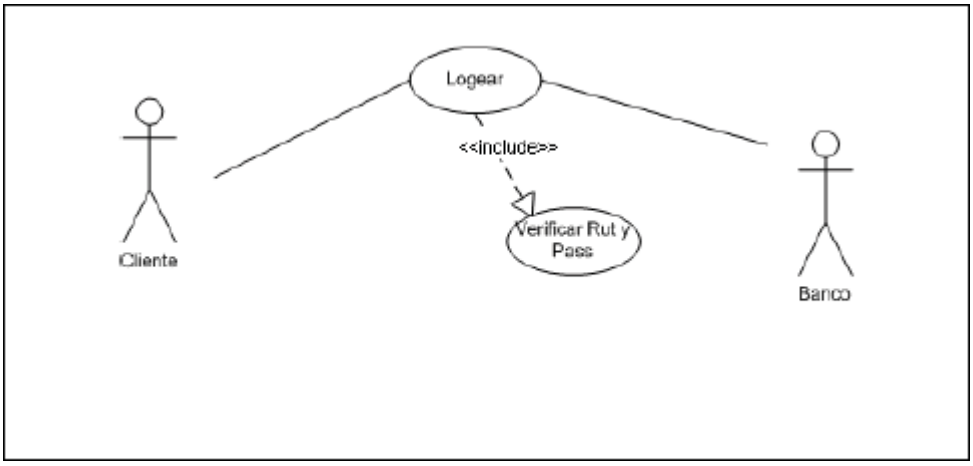


Diagrama 2: Conectarse.

Caso de Uso	**Logear**
Actores	Cliente (iniciador), Banco
Tipo	Principal
Descripción	El Cliente Ingresa su Rut (sin dígito verificador) y su password en la página web del Banco. Luego este, devuelve una respuesta de rechazo o concede el acceso a la siguiente página con funciones opcionales.

Caso de Uso	**Verificar Rut y Pass**
Actores	Banco
Tipo	Secundario
Descripción	La función Login, corrobora los datos ingresados por el usuario, frente a los que posee guardados en su base de datos.

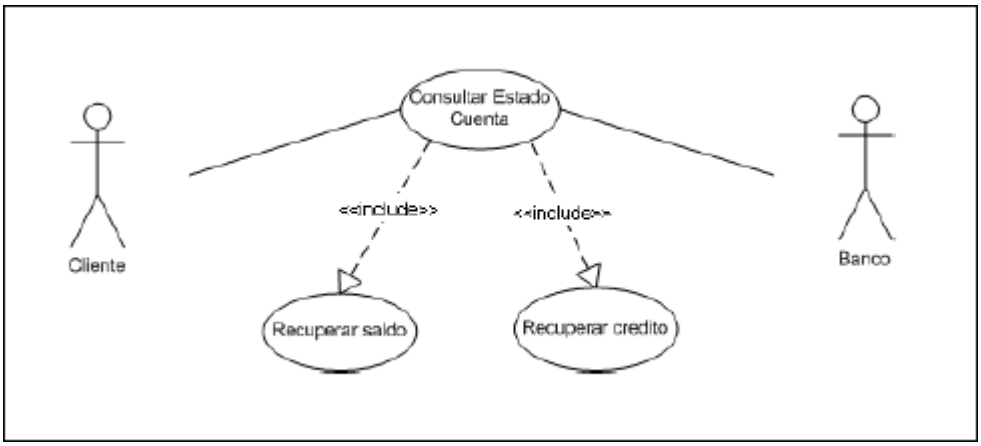


Diagrama 3: Estados de cuenta.

Caso de Uso	**Consultar estado cuenta.**
Actores	Cliente (iniciador), Banco.
Tipo	Principal
Descripción	El Cliente, a través de la página del Banco, puede conocer su saldo efectivo disponible y el crédito que el mismo Banco le tiene disponible para compras.

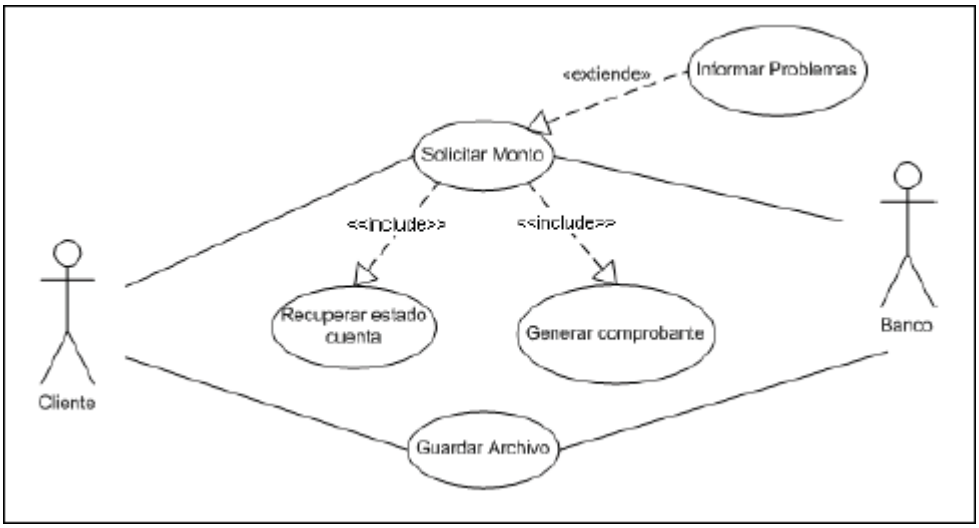


Diagrama 4: Solicitud de comprobante.

Caso de Uso	**Solicitar Monto**
Actores	Cliente (iniciador), Banco.
Tipo	Principal
Descripción	El Cliente, a través de la página del Banco, ingresa el monto de dinero que requiere para una compra. Posteriormente, el Banco genera un archivo de texto, que sirve como comprobante para el monto solicitado, el cual es entregado al cliente.

Caso de Uso	**Guardar Archivo**
Actores	Banco (iniciador), Cliente

Tipo	Secundario
Descripción	Una vez generado el archivo por el Banco, el Cliente debe guardarlo en su ordenador o medio de almacenamiento digital.

Caso de Uso	**Informar Problemas**
Actores	Banco
Tipo	Secundario
Descripción	El Banco informará al Cliente en caso de que se presente algún caso de problema a la hora de generar el comprobante y este proceso no se pueda concretar.

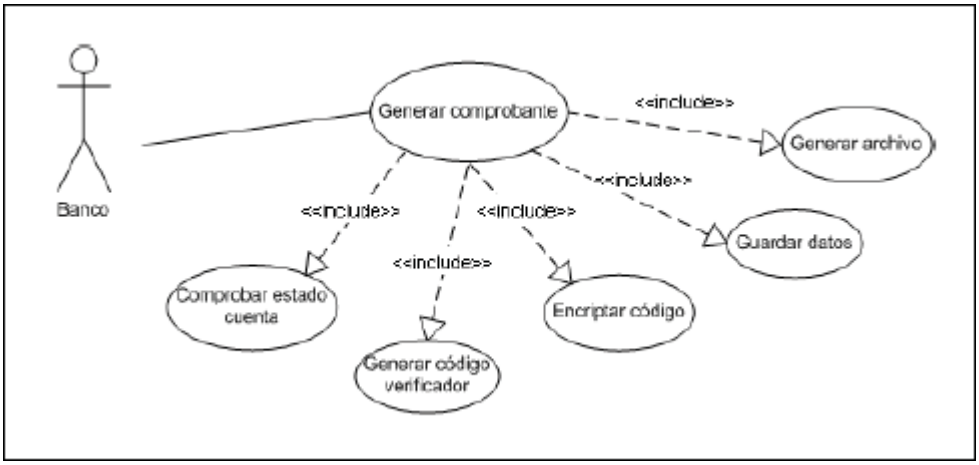


Diagrama 5: Generar Comprobante.

Caso de Uso	**Generar comprobante**
Actores	Banco
Tipo	Principal
Descripción	El Banco realiza el proceso de validación y posterior creación del comprobante solicitado por el Cliente, recorriendo varios pasos necesarios para que esto ocurra.

Caso de Uso	**Comprobar estado cuenta**
Actores	Banco
Tipo	Secundario
Descripción	El proceso de creación del comprobante, verifica que exista saldo disponible para respaldar el monto solicitado por el Cliente o en su defecto, que este tenga crédito suficiente.

Caso de Uso	**Generar código verificador**
Actores	Banco
Tipo	Secundario
Descripción	Mediante una función interna, el sistema crea un número serial aleatorio, el cual servirá de identificador único para cada

	comprobante solicitado.

Caso de Uso	**Encriptar código**
Actores	Banco
Tipo	Secundario
Descripción	El código generado con anterioridad, es tomado por una función y encriptado, con la clave del Banco, para posteriormente, este string, ser integrado en el interior del archivo entregado al cliente.

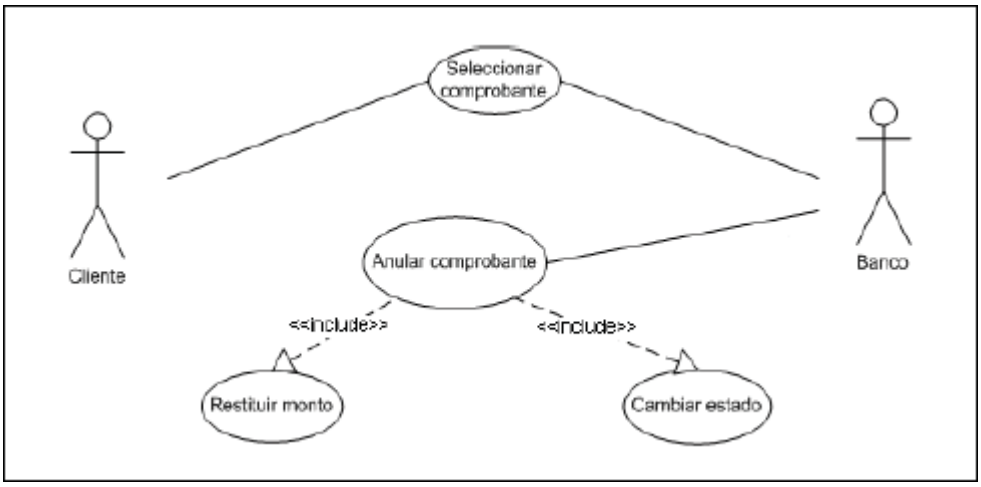


Diagrama 6: Anular un comprobante

Caso de Uso	**Seleccionar comprobante**

Actores	Cliente (iniciador), Banco
Tipo	Principal
Descripción	El Cliente visita la sección de “Comprobantes no cobrados” y esta le dará la opción, si así lo estima conveniente, de seleccionar uno de estos y solicitar su anulación.

Caso de Uso	**Anular comprobante**
Actores	Banco
Tipo	Secundario
Descripción	Luego de que el Cliente solicitó la anulación de un comprobante, se rescatan los datos relacionados con este y se procede a cambiar su estado interno, junto con restituir al saldo contable del cliente, el monto por el cual se había generado.

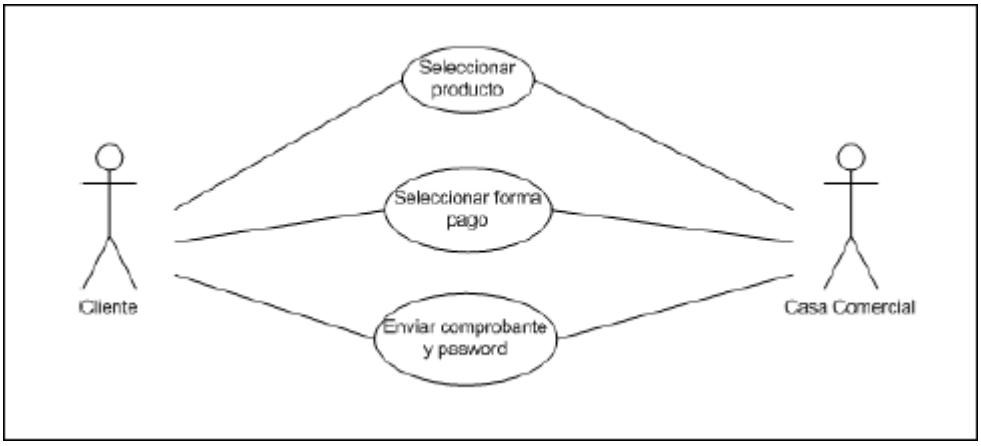


Diagrama 7: Comprando.

Caso de Uso	**Seleccionar Producto**
Actores	Cliente (iniciador), Casa Comercial
Tipo	Secundario
Descripción	El Cliente dentro de la página de la Casa Comercial, selecciona el producto que desea adquirir.

Caso de Uso	**Seleccionar forma de pago**
Actores	Cliente (iniciador), Casa Comercial
Tipo	Secundario
Descripción	El Cliente debe escoger, dentro de las opciones otorgadas por la Casa Comercial, su forma de pago, la que importa en este caso es “Cliente Seguro”.

Caso de Uso	**Enviar comprobante y password**
Actores	Cliente (iniciador), Casa Comercial
Tipo	Principal

Descripción	El Cliente, mediante la página de la Casa Comercial y una vez escogida su forma de pago, debe entregar el archivo relativo a su comprobante con el cual cancelará la compra y a su vez, una password que acredite que el comprobante que entrega, corresponde a la persona que lo envía.

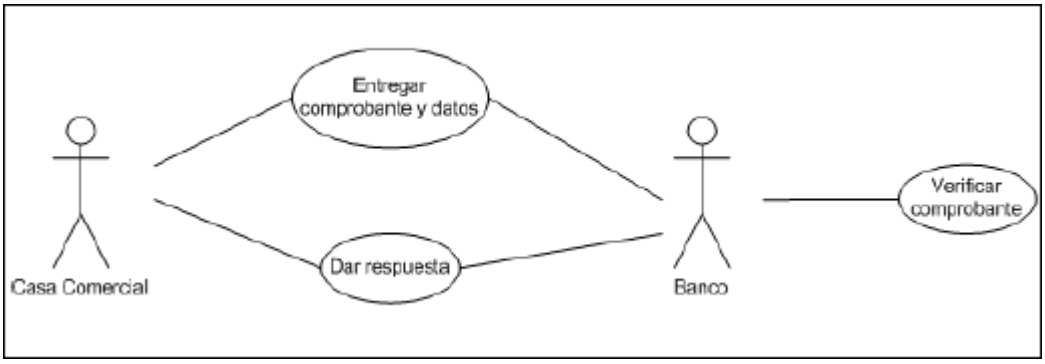


Diagrama 8: Validar Transacción

Caso de Uso	**Entregar comprobante y datos**
Actores	Casa Comercial (iniciador), Banco
Tipo	Secundario
Descripción	La Casa Comercial, mediante su servidor Web, se conecta con el Web service del Banco y hace entrega del archivo y password entregados por el cliente, además de su propia identificación y el costo del producto seleccionado.

Caso de Uso	**Verificar comprobante**
Actores	Banco
Tipo	Principal
Descripción	El Banco ejecuta el proceso de corroboración interna, para comprobar que el archivo y password entregados por el Cliente a la Casa Comercial, sean validos.

Caso de Uso	**Dar Respuesta**
Actores	Banco (iniciador), Casa Comercial
Tipo	Secundario
Descripción	El Banco entrega una respuesta a la Casa Comercial, informando si la transacción pudo ser realizada con éxito o si surgió algún problema durante el proceso.

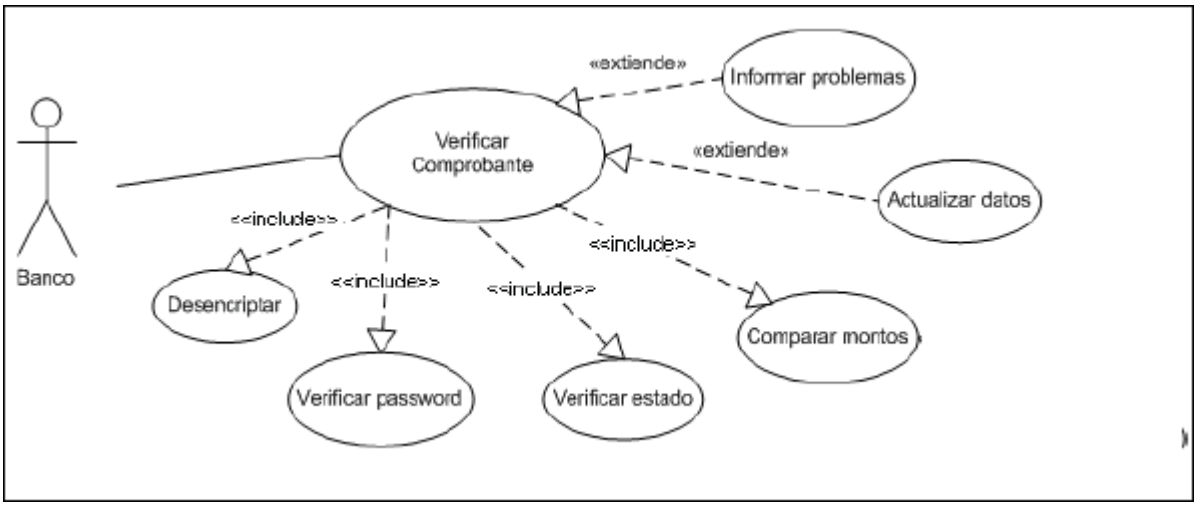


Diagrama 9: Validar archivo

Caso de Uso	**Verificar comprobante**
Actores	Banco
Tipo	Principal
Descripción	Cuando el Banco recibe por parte de la Casa Comercial, el contenido del archivo entregado por el Cliente y su password, este procede a corroborar que el comprobante no venga alterado, que aun sea valido y que corresponda con la password entregada, al mismo Cliente.

Caso de Uso	**Desencriptar**
Actores	Banco

Tipo	Principal
Descripción	Se desencripta el contenido del archivo, con la clave del Banco, para así corroborar que el texto no sufrió alteraciones.

Caso de Uso	**Verificar password**
Actores	Banco
Tipo	Principal
Descripción	Se comprueba que la password entregada por el Cliente, sea la que tiene registrada en la base de datos del Banco, en asociación con la del dueño del comprobante entregado.

Caso de Uso	**Verificar estado**
Actores	Banco
Tipo	Principal
Descripción	Se comprueba que el archivo aun sea valido, o sea, que no haya sido anulado, ni cobrado con anterioridad.

Caso de Uso	**Comparar montos**
Actores	Banco
Tipo	Principal
Descripción	Se verifica que el monto del comprobante entregado, sea adecuado para el costo del producto escogido. En caso de ser menor, el sistema informará que la transacción no puede ser realizada. Por otra parte, en caso de ser mayor, el sistema sumará el sobrante, al saldo contable, del Cliente dueño de la cuenta que generó el comprobante.

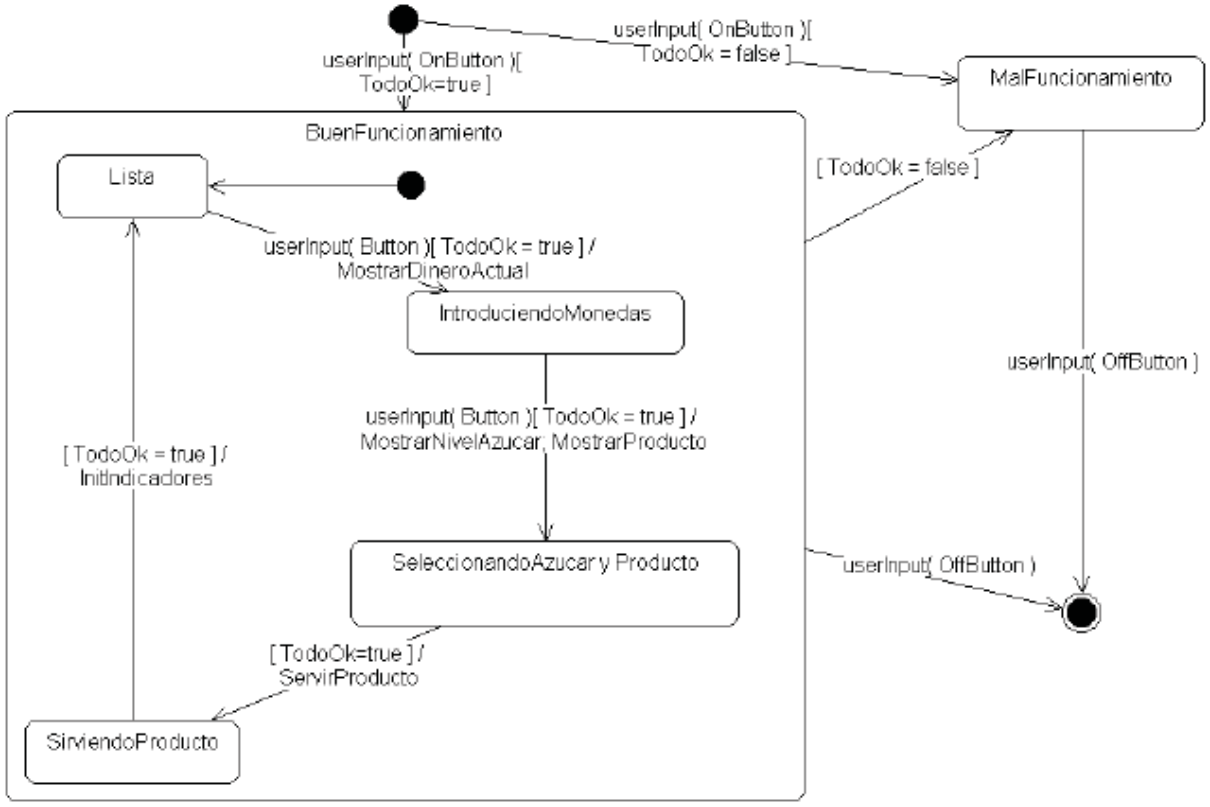
Caso de Uso	**Actualizar datos**
Actores	Banco
Tipo	Principal
Descripción	El Banco se encarga de guardar los datos referentes a la Casa Comercial que cobró el comprobante. Actualiza el estado del comprobante y el estado de cuenta del Cliente, en caso de ser necesario.

Caso de Uso	**Informar problemas**
Actores	Banco

Tipo	Secundario
Descripción	En caso de presentarse problemas para terminar la transacción satisfactoriamente, el Banco se lo hará saber a la Casa Comercial y por medio de esta, al Cliente.

3.3 Conceptos básicos en un Diagrama de Estados

Muestra el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro. Ejemplo de la cafetera



3.3.1 Estado

Identifica un periodo de tiempo del objeto (no instantáneo) en el cual el objeto esta esperando alguna operación, tiene cierto estado característico o puede recibir cierto tipo de estímulos. Se

representa mediante un rectángulo con los bordes redondeados, que puede tener tres compartimientos: uno para el nombre, otro para el valor característico de los atributos del objeto en ese estado y otro para las acciones que se realizan al entrar, salir o estar en un estado (entry, exit o do, respectivamente). En el caso del ejemplo anterior, se tienen cuatro estados (EnFuncionamiento, SinCambio, SinIngredientes, MalFuncionamiento), en los cuales se desarrollan ciertas acciones al entrar; por ejemplo, al entrar al estado SinIngredientes se debe realizar la acción "Indicador SinIngredientes en On".

Se marcan también los estados iniciales y finales mediante los símbolos  y  , respectivamente.

3.3.2 Eventos

Es una ocurrencia que puede causar la transición de un estado a otro de un objeto. Esta ocurrencia puede ser una de varias cosas:

Condición que toma el valor de verdadero o falso
Recepción de una señal de otro objeto en el modelo
Recepción de un mensaje
Paso de cierto período de tiempo, después de entrar al estado o de cierta hora y fecha particular

El nombre de un evento tiene alcance dentro del paquete en el cual está definido, no es local a la clase que lo nombre.

En el caso del ejemplo anterior se encuentra nombrado en varias transiciones el evento userInput, que recibe como parámetro un Button, para indicar el botón que ha sido presionado por el usuario de la máquina de café.

3.4 Diagramas de Estado.

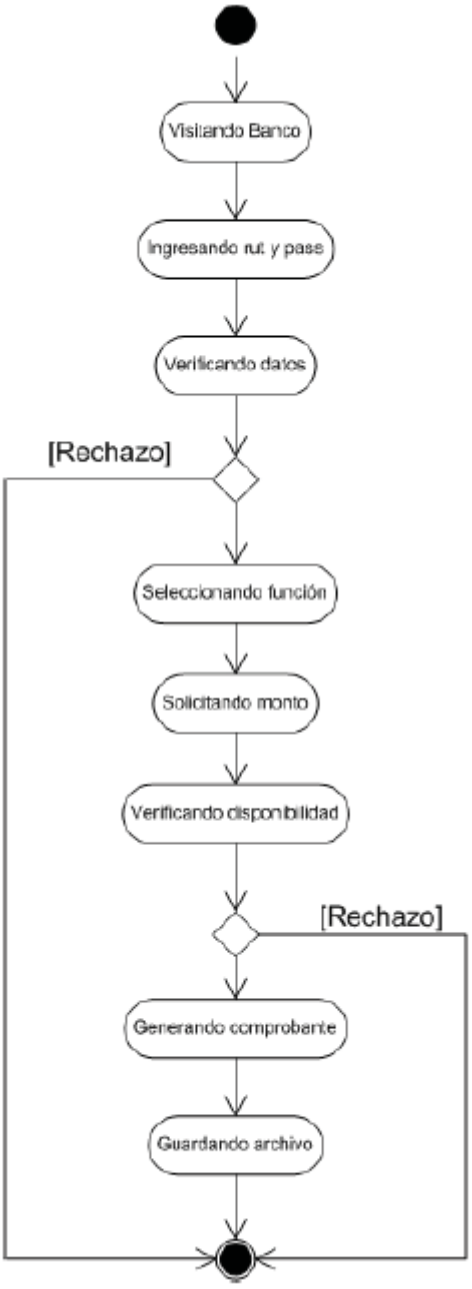


Diagrama 10: Solicitando monto.

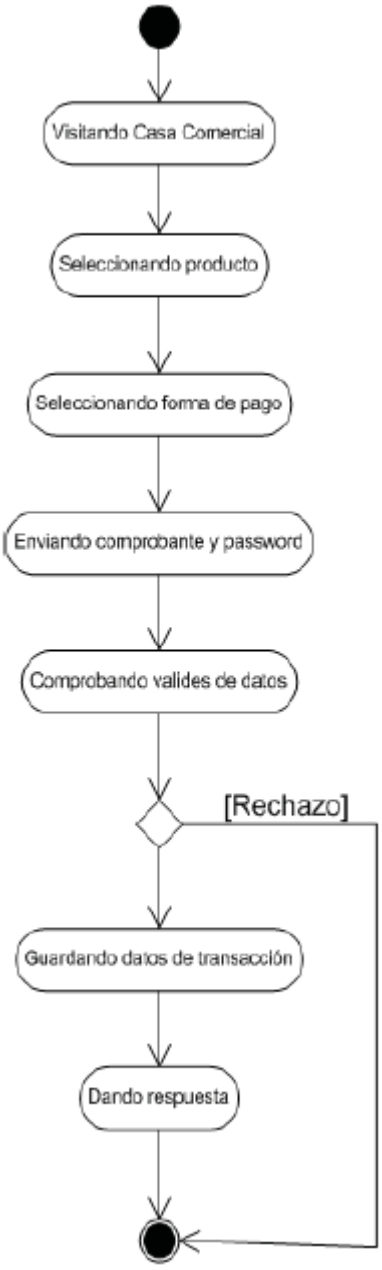
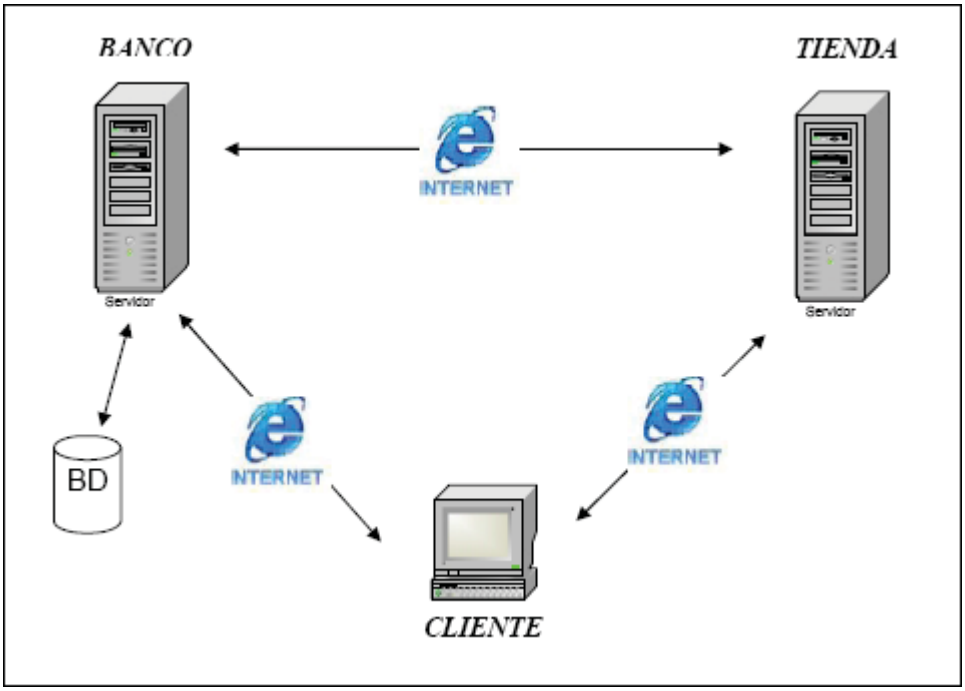


Diagrama 11: Compra de producto.

3.5 Descripción de funcionamiento y lógica del sistema.

3.5.1 Esquema gráfico de la arquitectura del Sistema.



Arquitectura Lógica:

Arquitectura de 3 capas (Interfaz – Web Service – Base de Datos). Esta es usada tanto en el portal del Banco, como en el de la Tienda.

Entidades involucradas en el Sistema.

Terminal Cliente
Portal del Banco
Portal Tienda Comercial

3.6 Diagrama de clases.

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento.

Un diagrama de clases esta compuesto por los siguientes elementos:

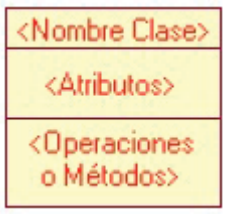
Clase: atributos, métodos y visibilidad. Relaciones: Herencia, Composición, Agregación, Asociación y Uso.

Elementos

- **Clase**

Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella se puede modelar el entorno en estudio (una Casa, un Auto, etc.).

En UML, una clase es representada por un rectángulo que posee tres divisiones:



En donde:

- **Superior**: Contiene el nombre de la Clase

- **Intermedio**: Contiene los atributos (o variables de instancia) que caracterizan a la Clase.

- **Inferior**: Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno.

- **Relaciones entre Clases:**

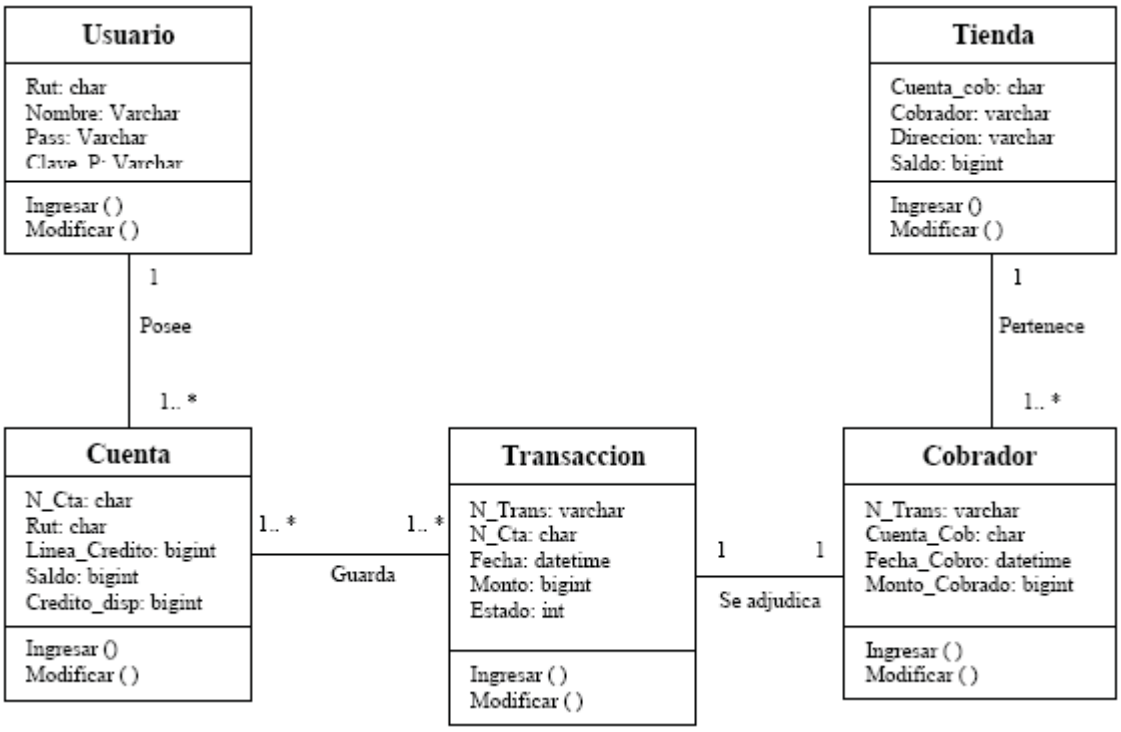
Ahora ya definido el concepto de Clase, es necesario explicar como se pueden interrelacionar dos o más clases (cada uno con características y objetivos diferentes).

Antes es necesario explicar el concepto de cardinalidad de relaciones: En UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y éstas pueden ser:

- **uno o muchos**: 1..* (1..n)

- **0 o muchos**: 0..* (0..n)

- **número fijo**: m (m denota el número).



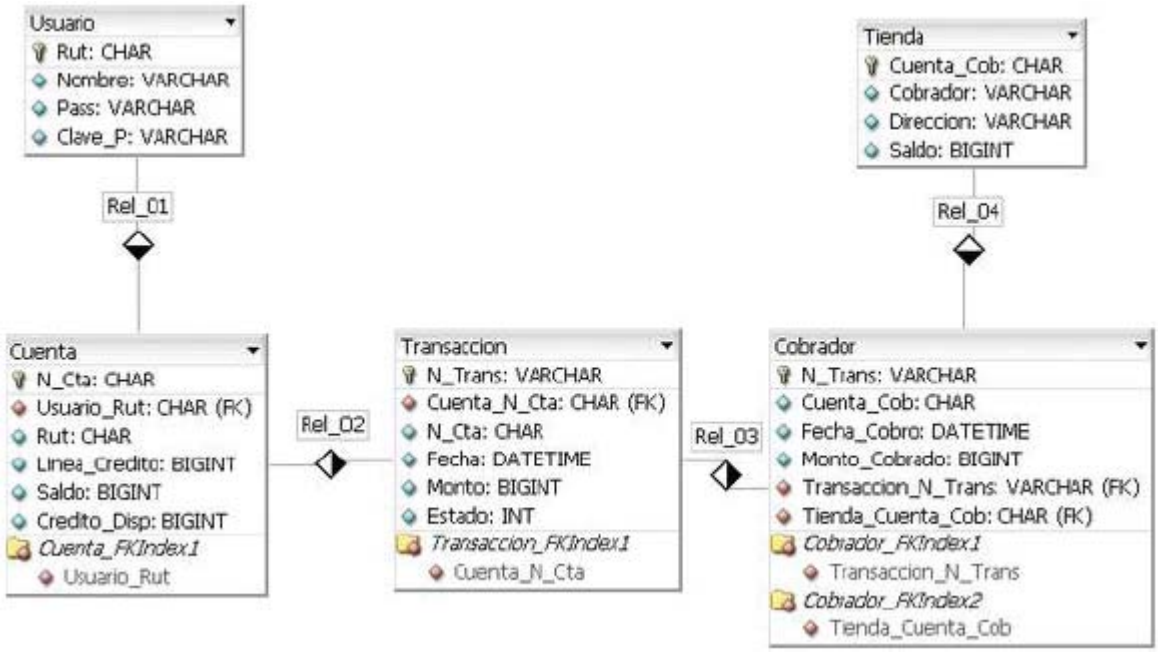
3.7 Modelo de Base de Datos.

El manejo de las Bases de Datos hoy, a escala empresarial, se ha convertido en una fortaleza con respecto a otras organizaciones que no realizan sus procedimientos de esta forma, ya que si bien es conocido, las Bases de Datos, proporcionan un conjunto de datos relacionados lógicamente, a través de los cuales pueden organizarse y consultarse de una forma ordenada y

práctica, dando como resultado requerimientos de información a un proceso manejado en una dependencia, y por un usuario específico, garantizando así la ejecución óptima de los mismos dentro de la empresa.

Para el caso del Banco, que es nuestro Cliente en este caso, y el cual proporciona los datos que utiliza nuestro Sistema, se debió modelar sus datos en base a las siguientes tablas, para evitar la redundancia de estos y proporcionar integridad a los datos.

Se hace hincapié, que este modelo de base de datos, está pensado para ser acoplado a la Base de Datos de la empresa en cuestión y por tal motivo, se han excluido intencionalmente, tablas que para el funcionamiento independiente de la aplicación, no son necesarias, pero que si lo serán al momento de encontrarse en pleno funcionamiento dentro de la entidad financiera.



3.8 Interfaces del Sistema

Figura 1. Portada del Banco.

En esta página, el cliente debe ingresar sus datos de autentificación, que en este caso son: "Rut" y "Password". Luego de ingresados y confirmados, los datos se envían al servidor web del Banco, para ser comparados con lo que existe en su base de datos.

Figura 2. Menú de opciones del Banco.

Este menú recibirá al cliente una vez que sus datos fueron validados en la página principal. Las opciones mostradas por el menú, permiten al usuario realizar distintas tareas como son:

- **Estado de cuenta:** al seleccionar, el cliente se encontrará con los datos de su "Saldo Contable", que se refiere al dinero efectivo que posee en su cuenta en ese momento y con "Crédito", que se refiere al saldo en su línea de crédito que tiene disponible.

- **Solicitud de monto:** esta opción le lleva a la página desde donde podrá solicitar un comprobante por el monto que requiera.

- **Montos sin cobrar:** dentro de esta sección, el cliente podrá ver los comprobantes que tiene vigentes, que aun no han sido utilizados y también podrá anular alguno si fuese necesario.

- **Montos cobrados:** este link dirigirá al cliente a un historial de sus comprobantes ya utilizados, con la información completa de montos, fechas y Casa Comercial por cual fue hecho efectivo.

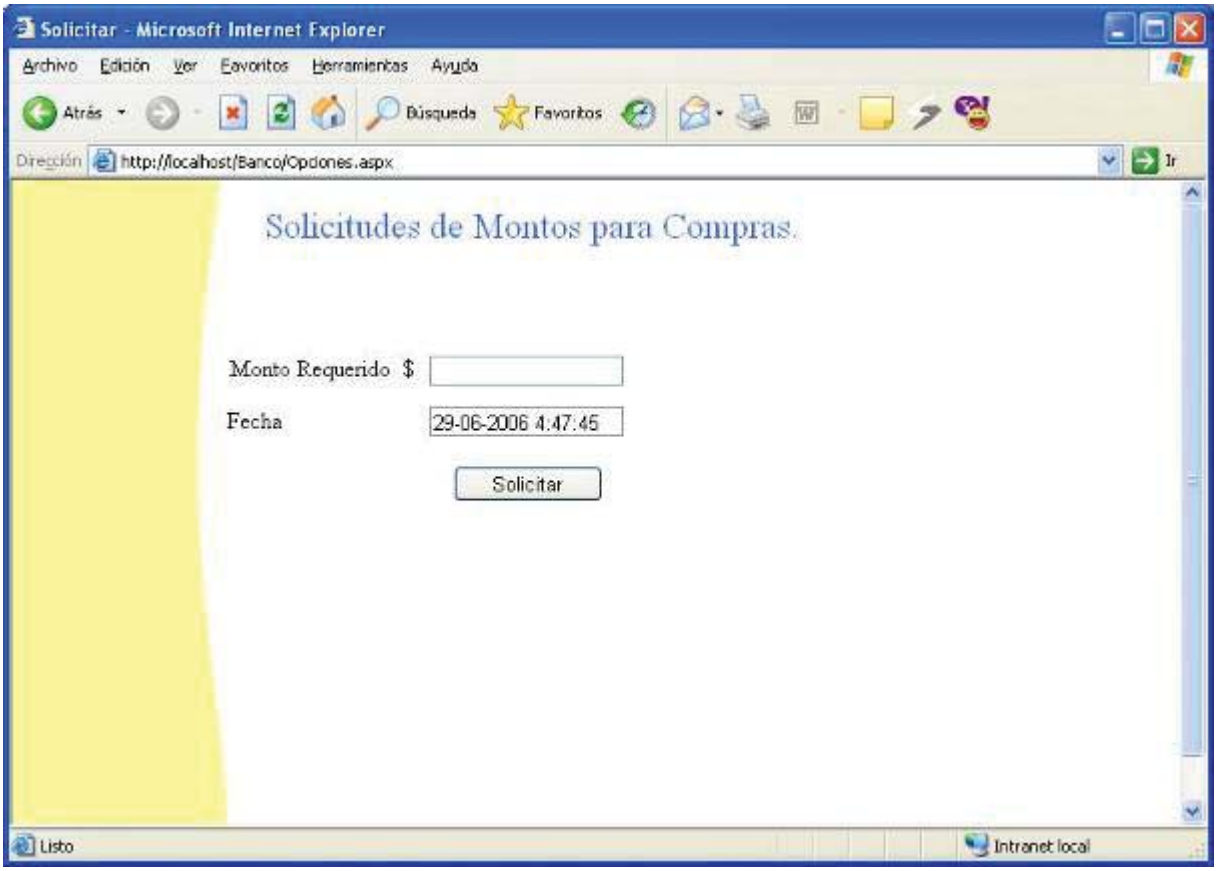


Figura 3. Interfaz de Solicitud de Dinero.

Esta es la interfaz fundamental del sistema, ya que es desde la cual se hará la solicitud de los comprobantes que el cliente necesite para realizar sus transacciones online. El cliente solo debe ingresar el monto en dinero requerido y esperar la confirmación junto con el archivo necesario.

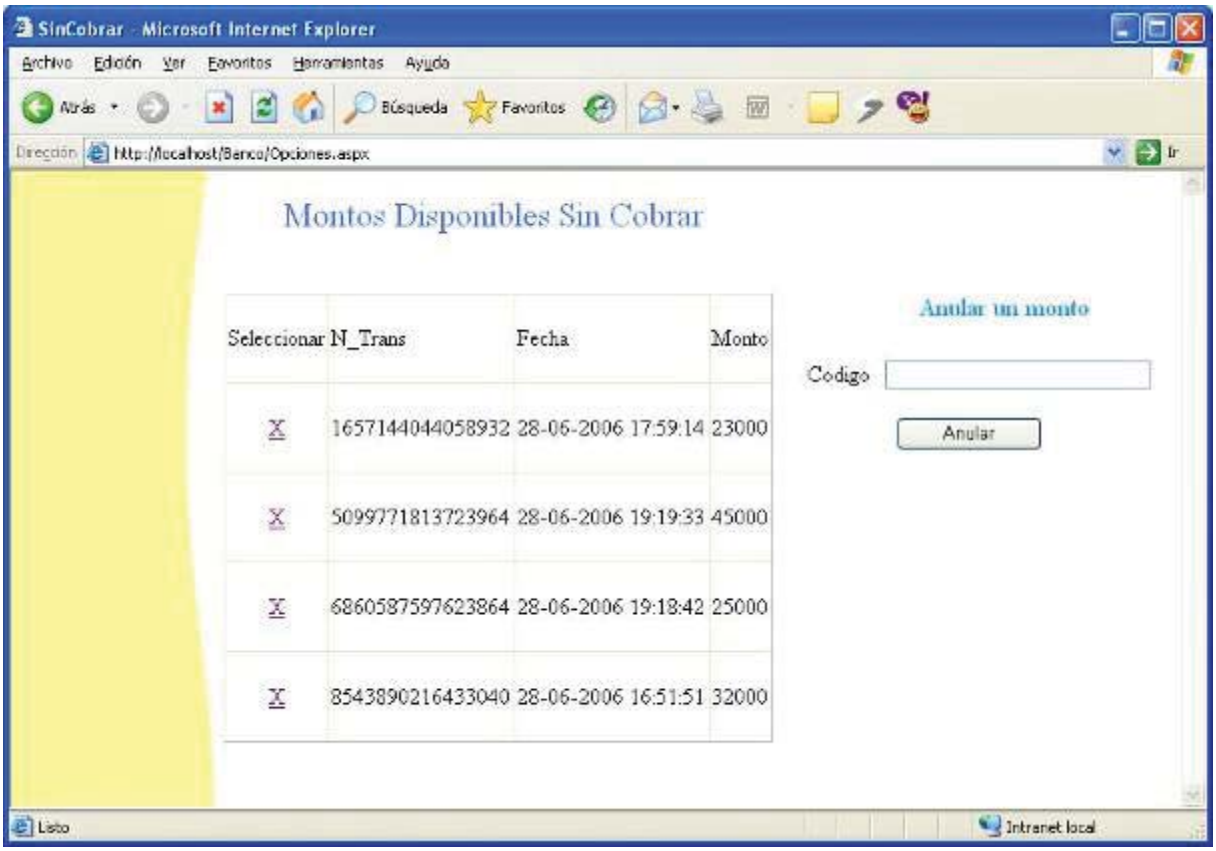


Figura 4. Interfaz para anulación de transacciones.

Esta interfaz a parte de mostrar al cliente los comprobantes que tiene aun vigentes sin cobrar, le da la opción de anular alguno en caso de ser necesario, para tal efecto, solo debe seleccionarlo, marcando la X que se encuentra al lado izquierdo del número de transacción vinculado con el comprobante en cuestión.

Figura 5. Interfaz donde se entrega el comprobante a la Tienda.

Esta página corresponde a la Casa Comercial y se llegará una vez elegido el artículo que se desea adquirir y la forma de pago que se utilizará. En este sitio, el cliente debe adjuntar el archivo (comprobante) con el cual realizará el pago del producto y luego ingresar la Password de validación de usuario. Una vez comprobado el archivo y datos de entrada, recibirá una confirmación en caso de resultar satisfactoria la transacción o un aviso en caso de presentarse algún problema en el proceso.

Capitulo IV

Pruebas, seguridad y comparaciones del sistema.

4. Pruebas, seguridad y comparaciones del sistema.

4.1 Plan de Pruebas.

Luego de concluir la etapa de construcción del sistema, se debió realizar las respectivas pruebas que aseguran, que el software es de calidad y que cumple con los requerimientos planteados en la etapa de Análisis y Diseño.

4.1.1 Objetivos de las Pruebas

El principal objetivo, es demostrar que el sistema implementado, cumple con los requerimientos planteados y su seguridad no presenta riesgo a los diferentes casos que se puedan presentar.

El sistema se encuentra dividido en 2 partes:

1. Todo lo que tiene que ver con el Banco y su prestación de servicios.
2. Funciones y prestaciones de la casa comercial.

En la primera parte, se encuentra la página Web del Banco, las funciones de transacción con la Base de datos (Web Service), un servidor que permita conexiones SSL, y un sistema que permita crear y validar los documentos emitidos por el Banco como dinero electrónico.

La segunda parte, es la que pertenece a la casa comercial. En esta se encuentra una aplicación, la cual conecte con la página Web de la empresa y permita realizar las transacciones pertinentes para la recepción del pago de los productos y posterior conexión con el Banco, para validar las transacciones.

4.1.2 Técnicas de Prueba

Para realizar el plan de prueba, existen dos técnicas de prueba muy conocidas:

Pruebas de Caja Negra.
Pruebas de Caja Blanca.

El primer enfoque mencionado, se realiza sobre las interfaces del software, es decir, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información interna, se mantiene.

El segundo tipo de prueba (caja blanca), se basa en el examen de los detalles procedimentales, es decir, se comprueban los caminos lógicos del software, proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles en el código del programa.

Existen distintos tipos de pruebas asociadas a estas técnicas, pero para construir el plan de prueba de este sistema, se utilizaron pruebas de caja negra, en lo que concierne a garantizar los datos ingresados y modificados en relación con la base de datos accesada, entregando avisos de confirmación o de error según sea el caso. Y de caja blanca en lo que a seguridad se refiere.

A continuación se describen las técnicas de caja negra usadas para probar este prototipo.

Para realizar las pruebas de caja negra, se utilizaron dos técnicas pertenecientes a la misma:

Análisis de valores límites.
Técnicas de grafo causa-efecto.

La primera técnica se utiliza cuando los casos a probar corresponden al correcto ingreso de los datos en el sistema, se puede mencionar como por ejemplo, el comprobar el campo “RUT”, en donde es necesario validar que tenga un número exacto de caracteres y que concuerde con alguno que ya se encuentre dentro de la base de datos.

La segunda técnica se utiliza en aquellos casos en donde se debe validar que cualquier dato ingresado, corresponda con un determinado estándar y de acuerdo a esto, analizar como responde la aplicación.

4.1.3 Estrategias de Prueba en programación.

Una estrategia de prueba, pretende encontrar posibles errores que el software implementado pudiese tener. Para cumplir con este objetivo, se deben realizar diferentes tipos de pruebas en cada nivel del sistema, para esto existen las siguientes pruebas:

Pruebas de Unidad: se ejecuta el plan de pruebas a nivel de módulo.
Pruebas de Integración: se ejecutan los casos de prueba una vez integrados los módulos que componen el sistema.
Pruebas de Validación: se ejecutan las pruebas de caja negra que demuestran la conformidad con los requisitos y reglas de la empresa.
Prueba del Sistema: se ejecuta para ejercitar el software y comprobar que este funciona en forma correcta.

Por el enfoque genérico de este software y tomando en cuenta que su base es la seguridad informática, las pruebas del prototipo fueron basadas en confirmar la robustez y confiabilidad que presta el sistema a la hora de trabajar en línea y bajo sistemas remotos.

Por otra parte, las pruebas de caja negra, fueron realizadas por una tercera persona, ajena al sistema, para confirmar un plan de prueba arbitrario y no dirigido, en cuanto a validación en el ingreso de datos.

4.1.4 Casos de Prueba en caja negra.

- **Módulo Portada**.

Caso	1
Función	Login
Campo	Rut
Caso de prueba	Ingreso Vacío
Resultado esperado	Mensaje: **Datos No Validos**
Resultado obtenido	Mensaje: **Datos No Validos**

Caso	2
Función	Login
Campo	Rut
Caso de prueba	Caracteres alfanuméricos
Resultado esperado	Mensaje: **Datos No Validos**
Resultado obtenido	Mensaje: **Datos No Validos**

Caso	3
Función	Login
Campo	Rut

Caso de prueba	Rut no existente
Resultado esperado	Mensaje: **Datos No Validos**
Resultado obtenido	Mensaje: **Datos No Validos**

Caso	4
Función	Login
Campo	Pass
Caso de prueba	Ingreso Vacío
Resultado esperado	Mensaje: **Datos No Validos**
Resultado obtenido	Mensaje: **Datos No Validos**

Caso	5
Función	Login
Campo	Pass
Caso de prueba	Clave no correspondiente
Resultado esperado	Mensaje: **Datos No Validos**
Resultado obtenido	Mensaje: **Datos No Validos**

- **Módulo Solicitar Monto**.

Caso	1
Función	InsMonto
Campo	Monto
Caso de prueba	Ingreso Vacío
Resultado esperado	Mensaje: **El monto solicitado, no es un valor valido.**
Resultado obtenido	**Error en la carga de página**
Solución desarrollada	**Respuesta condicionada en caso de excepciones**

Caso	2
Función	InsMonto
Campo	Monto
Caso de prueba	-10000
Resultado esperado	Mensaje: **No es posible generar un comprobante por el monto requerido.**

Resultado obtenido	**Comprobante creado.**
Solución desarrollada	**Condición validadora para montos positivos.**

Caso	3
Función	InsMonto
Campo	Monto
Caso de prueba	Monto < 500
Resultado esperado	Mensaje: **No es posible generar un comprobante por el monto requerido.**
Resultado obtenido	Mensaje: **No es posible generar un comprobante por el monto requerido.**

Caso	4
Función	InsMonto
Campo	Monto
Caso de prueba	Dkjghgos

Resultado esperado	Mensaje: **El Monto solicitado no es un valor valido.**
Resultado obtenido	Mensaje: **El Monto solicitado no es un valor valido.**

- **Módulo Compras**.

Caso	1
Función	Validar
Campo	Archivo
Caso de prueba	Ingreso Vacío.
Resultado esperado	Mensaje: **Se produjo un error en el proceso, es posible que su archivo esté corrupto.**
Resultado obtenido	Mensaje: **Se produjo un error en el proceso, es posible que su archivo esté corrupto.**

Caso	2

Función	Validar
Campo	Archivo
Caso de prueba	Comprobante Modificado.
Resultado esperado	Mensaje: **Se produjo un error en el proceso, es posible que su archivo esté corrupto.**
Resultado obtenido	Mensaje: **Se produjo un error en el proceso, es posible que su archivo esté corrupto.**

Caso	3
Función	Validar
Campo	Archivo
Caso de prueba	Comprobante Usado.
Resultado esperado	Mensaje: **Este comprobante, fue utilizado con anterioridad.**
Resultado obtenido	**Cobro del comprobante nuevamente.**
Solución desarrollada	**Campo de estado del comprobante en la base de datos.**

Caso	4
Función	Validar
Campo	Archivo
Caso de prueba	Archivo no correspondiente.
Resultado esperado	Mensaje: **Se produjo un error en el proceso, es posible que su archivo esté corrupto.**
Resultado obtenido	Mensaje: **Se produjo un error en el proceso, es posible que su archivo esté corrupto.**

Caso	5
Función	Validar
Campo	Password
Caso de prueba	No correspondiente con el comprobante.
Resultado esperado	Mensaje: **Password no Valida.**
Resultado obtenido	Mensaje: **Password no Valida.**

4.2 Seguridad del sistema.

Factor muy importante dentro de este software, es la seguridad, ya que todo está basado en ganar la confianza total del cliente en el sistema y además, brindar una seguridad absoluta al momento de trabajar con sus valores.

4.2.1 Amenazas en seguridad.

Se dirá que se entiende por amenaza una condición del entorno del sistema (persona, máquina, suceso o idea) que, dada una oportunidad, podría dar lugar a que se produjese una violación de la seguridad (confidencialidad, integridad, disponibilidad o uso legítimo).

La política de seguridad y el análisis de riesgos deben identificar las amenazas que han de ser contrarrestadas, especificando los servicios y mecanismos de seguridad necesarios.

Las amenazas a la seguridad en una red pueden caracterizarse modelando el sistema como un flujo de información desde una fuente, como por ejemplo un fichero o una región de la memoria principal, a un destino, como por ejemplo otro fichero o un usuario.

Un ataque no es más que la realización de una amenaza. Las cuatro categorías generales de amenazas o ataques son las siguientes:

Interrupción: un recurso del sistema es destruido o se vuelve no disponible. Este es un ataque contra la disponibilidad. Ejemplos de este ataque son la destrucción de un elemento hardware, como un disco duro, cortar una línea de comunicación o deshabilitar el sistema de gestión de ficheros.

Intercepción: una entidad no autorizada consigue acceso a un recurso. Este es un ataque contra la confidencialidad. La entidad no autorizada podría ser una persona, un programa o un computador. Ejemplos de este ataque son pinchar una línea para hacerse con datos que circulen por la red y la copia ilícita de ficheros o programas (intercepción de datos), o bien la lectura de las cabeceras de paquetes para develar la identidad de uno o más de los usuarios implicados en la comunicación observada ilegalmente (intercepción de identidad).

Modificación: una entidad no autorizada no sólo consigue acceder a un recurso, sino que es capaz de manipularlo. Este es un ataque contra la integridad. Ejemplos de este ataque son, el cambio de valores en un archivo de datos, alterar un programa para que funcione de forma diferente y modificar el contenido de mensajes que están siendo transferidos por la red.

Fabricación: una entidad no autorizada inserta objetos falsificados en el sistema. Este es un ataque contra la autenticidad. Ejemplos de este ataque son la inserción de mensajes engañosos en una red o añadir registros a un archivo. Estos ataques se pueden asimismo clasificar de forma útil en términos de ataques pasivos y ataques activos.

Ataques pasivos

En los ataques pasivos el atacante no altera la comunicación, sino que únicamente la escucha o monitoriza, para obtener información que está siendo transmitida.

Sus objetivos son la intercepción de datos y el análisis de tráfico, una técnica más sutil para obtener información de la comunicación, que puede consistir en:

- Obtención del origen y destinatario de la comunicación, leyendo las cabeceras de los paquetes monitorizados.

- Control del volumen de tráfico intercambiado entre las entidades monitorizadas, obteniendo así información acerca de actividad o inactividad inusuales.

- Control de las horas habituales de intercambio de datos entre las entidades de la comunicación, para extraer información acerca de los períodos de actividad.

Los ataques pasivos son muy difíciles de detectar, ya que no provocan ninguna alteración de los datos. Sin embargo, es posible evitar su éxito mediante el cifrado de la información y algunos otros mecanismos.

Ataques activos

Estos ataques implican algún tipo de modificación del flujo de datos transmitido o la creación de un falso flujo de datos, pudiendo subdividirse en cuatro categorías:

1. Suplantación de identidad: el intruso se hace pasar por una entidad diferente. Normalmente incluye alguna de las otras formas de ataque activo. Por ejemplo, secuencias de autenticación pueden ser capturadas y repetidas, permitiendo a una entidad no autorizada acceder a una serie de recursos privilegiados suplantando a la entidad que posee esos privilegios, como al robar la contraseña de acceso a una cuenta.
2. Reactuación: uno o varios mensajes legítimos son capturados y repetidos para producir un efecto no deseado, como por ejemplo ingresar dinero repetidas veces en una cuenta dada.
3. Modificación de mensajes: una porción del mensaje legítimo es alterada, o los mensajes son retardados o reordenados, para producir un efecto no autorizado. Por ejemplo, el mensaje "Ingresa un millón de pesos en la cuenta A" podría ser modificado para decir "Ingresa un millón de pesos en la cuenta B".
4. Degradación fraudulenta del servicio: impide o inhibe el uso normal o la gestión de recursos informáticos y de comunicaciones. Por ejemplo, el intruso podría suprimir todos los mensajes dirigidos a una determinada entidad o se podría interrumpir el servicio de una red inundándola con mensajes inútiles. Entre estos ataques se encuentran los de denegación de servicio, consistentes en paralizar temporalmente el servicio de un servidor de correo, Web, FTP, etc.

4.2.2 Implementaciones de seguridad.

Se declara para cada tipo de amenaza, las medidas tomadas para prevenirlas, ya sea dentro del mismo sistema o lo que se necesita implementar en el lugar donde este funcionará.

Interrupción: para prevenir los ataques de interrupción, se debe tener en cuenta la seguridad física y de conexión entre las partes que ofrecerán el servicio.

Las empresas deben asegurarse de que sus servidores no son vulnerables a las calamidades físicas. Se debe colocar estos equipos en una sala segura y con buena ventilación, no en un pasillo ni debajo de una mesa donde alguien pueda golpearlos o derramar café sobre ellos. O alterarlos con fines maliciosos. En la sala de servidores no debe haber ventanas y debe tener una sola puerta que pueda cerrar. Las carcasas de los servidores también deben estar cerradas para impedir que se modifiquen los componentes internos. Se debe hacer una relación de los empleados que tienen las llaves de la sala de servidores. También se debe conservar un registro de los números de serie de los servidores y marcarlos con la información de la compañía, de modo que se puedan identificar y recuperar si se roban.

Por otro lado, las empresas deberán asegurarse de usar filtros de seguridad como son Firewalls, IDS e IPS.

Intercepción: el sistema Cliente Seguro, se conecta bajo protocolos de seguridad a la hora de comunicarse entre las partes. Tanto entre Cliente – Banco y Cliente – Tienda, existe una conexión SSL, la que resguarda la seguridad de los datos que se transportan, ya que se basa en un canal encriptado, lo que hace que si una persona se metiese en el canal de comunicación, no pudiese entender lo que por allí viaja.

Por otra parte, la comunicación entre los web service de la Tienda y del Banco, se ve respaldada por la encriptación de datos por parte de Web Services Enhancements 2.0 (WSE).

Y por último, las claves de acceso de cada usuario, no son guardadas en forma legible en la base de datos, sino que su correspondiente hash emitido con el algoritmo MD5.

Modificación: los archivos o "Comprobantes" emitidos por el Banco, contienen el hash del código de transacción correspondiente y este a su vez, encriptado con el algoritmo Triple Des. Lo que hace casi imposible a una persona mal intencionada adueñarse de este comprobante, poder descifrar lo que lleva dentro y en caso de que así fuese, no lo podría modificar, ya que el hash acusaría la modificación realizada y por lo tanto, la no concordancia con la información que posee el Banco, lo que haría que este comprobante no sea valido.

Fabricación: al igual que en el punto anterior, se puede decir que el sistema cuenta con seguridad de identidad, ya que con el uso de Triple Des en la autentificación del autor de los comprobantes, se puede asegurar la originalidad de los mismos. Por otra parte, el uso de contraseñas para hacer validas las transacciones por parte del Cliente, asegura su autentificación y no repudio.

4.3 Comparaciones con sistemas ya existentes.

		Cliente Seguro
Cybercash	El cliente igual debe enviar datos	No necesita entregar datos del

	de su tarjeta de crédito, para ser verificados.	cliente para realizar la transacción.
Set	Se requiere instalar un software por parte del usuario en su computador, lo que imposibilita al cliente de poder acceder desde cualquier PC.	Solo requiere una conexión a Internet y un medio de almacenamiento digital.
Tarjeta Monedero	La tarjeta se carga por medio de otra de debito emitida por el Banco. En caso de perdida de la tarjeta, también se pierde el monto guardado en ella.	El uso de los comprobantes, es con carga a la cuenta corriente directamente y en caso de perdida, los montos no pueden ser cobrados sin la clave de validación y a su vez, pueden ser anulados.
Tarjeta de Crédito	Cualquier persona que encuentre una tarjeta de crédito que no ha sido bloqueada, puede hacer uso de ella por Internet, sin ningún repudio. Además, al momento de usarse en compras en la red, se debe entregar los datos de la misma y confiar en que no serán usados de forma inapropiada.	Los comprobantes emitidos por el Banco, solo pueden ser usados por el usuario que conozca el password de validación. Además, al momento de usar un comprobante, solo se debe entregar como dato adicional, la password de validación. La cual no tiene ninguna utilidad, sin un comprobante no cobrado, del mismo usuario.
Tarjeta Virtual	Si bien este sistema resuelve el problema de entregar datos por Internet, a la vez dificulta su funcionalidad, ya que solo puede	El comprobante de "Cliente Seguro", no encasilla en tiempos limitados de uso. Además permite su uso por montos inferiores a su

	ser usada una vez, en un plazo no mayor a 48 horas y con cargo al mismo monto por el cual fue emitida.	valor, entregando el vuelto correspondiente.

Capitulo V

Conclusiones y Bibliografía.

5. Conclusiones y Bibliografía.

5.1 Conclusiones.

Muchas alternativas fueron consideradas desde el inicio de este trabajo.

Lo que comenzó como una idea ambiciosa, innovadora, algo fantástica pero a la vez muy practica dentro del ramo de Seguridad Informática, se convirtió finalmente en una realidad concreta y funcional, aunque un poco alejada de la idea original.

En un comienzo “Billetera Electrónica” como fue bautizado, sería un software que constaría de 3 partes. La primera, una aplicación cliente que residiría en el PC del cliente y la cual mediante contraseña y validación biométrica, permitiría la conexión con la segunda parte, la aplicación servidor de la entidad financiera. Una vez realizada la solicitud, el Banco produciría un comprobante, el cual debía ser guardado por el cliente en su pendrive especialmente diseñado para esta función y el cual poseería un sistema de seguridad de validación de huella digital, para poder ser usado.

La tercera parte, sería la correspondiente a la Casa Comercial, la cual aceptaría el medio de pago y el archivo recibido podría ser usado inmediatamente, como si de un billete cualquiera se tratara, sin necesidad de ser validado con el Banco o estar conectado en línea con él.

Solo al final de su vida útil (cuando un usuario decidiera hacerlo efectivo), este archivo dejaría de ser digital, volvería al Banco y este haría la transformación en pesos.

Tras evaluaciones tecnológicas y de seguridad, las conclusiones fueron categóricas. Un sistema seguro, no debe dar oportunidad a un usuario mal intencionado de poder manipularlo

a su antojo y sin una supervisión. Por lo tanto la comprobación biométrica que se realizaría en la aplicación cliente y en el pendrive, serían muy factibles de ser violadas.

Por otra parte, el no contar con una validación en línea del archivo entregado como forma de pago, podría ser fruto de fraudes futuros y de los cuales no abría evidencia hasta muy tarde.

Luego de cambiar bastante su formato y tras decidir como funcionaría la versión final de la aplicación, se optó por rebautizarlo de una forma más acorde a su realidad, ahí apareció “Cliente Seguro”.

El cual está constituido de una interfaz de conexión en el Banco y otra en la Tienda, las cuales a su vez se conectan entre si y con el cliente, permitiendo un canal de comunicación seguro y fácil de utilizar.

Finalizado el trabajo, se puede decir que el objetivo buscado de lograr una alternativa segura, fácil de usar y entendible fácilmente por los clientes al momento de comprar por Internet, logró ser alcanzado.

Se desarrolló una herramienta que es capas de reemplazar a las tarjetas de crédito en una forma mucho más segura y anónima a la hora de hacer compras en el Internet.

Además se logró que la identidad del comprador, tanto como sus datos, sean totalmente transparentes para la tienda, la cual, si el cliente no lo desea, nunca sabrá con quien hizo negocios. A su vez, el Banco, no tiene conocimiento alguno de cual fue el producto adquirido por el usuario. La entidad financiera solo se encarga de respaldar la transacción y de hacer efectiva la misma.

La herramienta “Cliente Seguro”, es factible de ser implementada por alguna entidad financiera, la que a su vez, se debe encargar de habilitar el servicio para sus clientes y hacer una relación directa con las casas comerciales.

La aplicación que en este informe se detalla, es una combinación de practicas de seguridad usadas en la actualidad por sistemas existentes, pero que en su conjunto, logran una

herramienta confiable, segura, fácil de usar y que se espera, tenga una muy buena llegada en el comercio electrónico.

Dentro de este mismo informe, se pueden apreciar comparaciones con alternativas que en estos momentos se encuentran disponibles para el usuario a nivel mundial. Estas comparaciones dejan en claro que aunque algunas se aproximan a la herramienta "Cliente Seguro" en cuanto a la seguridad que presentan, esta última siempre ofrece ventajas adicionales que la hacen ser la mejor alternativa de mercado.

Por último, no está demás agregar que al finalizar el camino de tantos años de estudio y trabajo dentro de esta Universidad, puedo mencionar lo fructífero y beneficioso que fue el obtener una formación profesional otorgada de parte de otros profesionales de excelencia y a su vez el gran desarrollo humano que logré sacar como consecuencia de una estrecha relación con mis pares estudiantes, con los cuales logré aprender a ejecutar un buen trabajo en equipo, lo cual es tan necesario en estos días al momento de programar y desarrollar tareas o proyectos, grandes y ambiciosos.

5.2 Bibliografía y referencias.

Roger Pressman "Ingeniería de Software: un enfoque práctico" Editorial McGraw-Hill.

Sebastián Firtman, "Seguridad Informática" Editado por MP Ediciones S.A. 2005.

Iván Jacobson, Grady Booch, James Rumbaugh "El Proceso Unificado de Desarrollo de Software" Pearson Educación s.a Madrid, 2000.

Martin Fowler, Kendall Scout "UML Gota a Gota" Mexico: Addison Wesley Longman, 1999.

http://www.iee.es/bases/articulos/come003.htm

http://www.infopeople.com/aaii/comercio/

http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/dntaloc/html/winsecurity.asp

http://www.fd.com.ar/products.htm

http://nti.uji.es/docs/nti/net/dinero_electronico.html

http://www.delitosinformaticos.com/

http://www.tuguialegal.com/firmadigital4.htm

http://www.microsoft.com/spanish/msdn/articulos/archivo/140305/voices/wserolebasedsec.asp

Capitulo VI

Apéndices

Apéndice A

- **Glosario**

Algoritmo de encriptación: Conjunto de instrucciones orientadas a volver ilegible información considerada importante. La información una vez encriptada, solo puede leerse aplicándole una clave.

Aplicación: Programa escrito en cualquier lenguaje, que permite trabajar a un usuario en una computadora.

Autenticar: Corroborar una identidad.

Base de datos: Almacén de datos relacionados con diferentes modos de organización.

DES: (Data Encryption Estándar) Algoritmo de encriptación estándar.

Encriptación: Cifrado. Proceso para volver ilegible información considerada importante.

Firewall: Herramienta de seguridad que controla el tráfico de entrada/salida de una red.

Interfaz: Es la parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario.

Intranet: Es una red montada exclusivamente dentro de una empresa e incluso en un hogar con varias computadoras.

Lenguaje de programación: En informática, es cualquier forma de escritura (lenguaje) que posee determinadas instrucciones que combinadas correctamente (dependiendo del resultado que se desee), podrán ser interpretadas y así resultar en un programa, página web, etc.

Paradigma: Un paradigma de programación provee (y determina) la visión y métodos de un programador en la construcción de un programa o subprograma. Diferentes paradigmas resultan en diferentes estilos de programación y en diferentes formas de pensar la solución de problemas (con la solución de múltiples “problemas” se construye una aplicación).

Password: Palabra clave o contraseña de acceso.

Plataforma: En informática, determinado software y/o hardware con el cual una aplicación es compatible y permite ejecutarla. Una plataforma es, por ejemplo, un sistema operativo.

Portal: Sitio web cuyo objetivo es ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse buscadores, foros, compra electrónica, etc.

Protocolo: En red de computadoras, un protocolo es el lenguaje (conjunto de reglas formales) que permite comunicar a dos nodos (computadoras).

Servidor: Es la computadora central en un sistema de red que provee servicios a otras computadoras.

Software: Son los programas almacenados en un computador.

Software genérico: Se refiere al software que no está encasillado o ajustado para funcionar bajo un solo ambiente de trabajo o plataforma y puede ser usado en la mayoría de los computadores.

SSL: (Secure Socket Layer). Protocolo diseñado para proveer comunicaciones encriptadas en Internet.

Apéndice B

Texto Apéndice B en PDF, en: