CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Departamento de Ingeniería Eléctrica
Sección de Computación

Implementación de un Monedero Digital Móvil

Tesis que presenta el:

Ing. Efrén Clemente Cuervo

para obtener el Grado de Maestro en Ciencias en la especialidad de Ingeniería Eléctrica opción Computación

Director de la Tesis

Dr. Francisco José Rambó Rodríguez Henríquez

México, D.F. Noviembre 2005

A mi esposa Martha quien con intuición y amor me animó e impulsó para cumplir esta meta en mi vida. Y aunque habría merecido algo mejor, aun así, con todas sus imperfecciones a ella le pertenece (20-24-48).

Agradecimientos

Agradezco a ese ser superior, al que yo llamo Dios, por bendecirme y estar a mi lado para fortalecer mi espíritu en cada momento de mi vida .

Con mucho cariño quiero agradecer a mis padres Efrén y Cecilia quienes con su amor, orientación y apoyo incondicional han hecho de mi lo que soy. De igual forma a mis hermanos Flor y Erick por estar ahí cuando los pude haber necesitado.

Con un infinito amor agradezco a mi esposa Martha y a mi hija Itzel quienes aunque un poco lejos de mi, siempre estuvieron en mi corazón dándome la motivación necesaria para el inicio, el desarrollo y la culminación de esta tesis.

A mis amigos y familiares les agradezco que siempre hayan confiado en mi y me hayan alentado para llevar a cabo esta pequeña meta en mi proyecto de vida.

Un sincero agradecimiento a mi director de tesis, el Dr. Francisco José R. Rodríguez Henríquez por su aliento y exigencias para la realización de esta tesis.

Agradezco al CINVESTAV, ya que durante estos dos años de maestría fue la institución donde curse mis materias y pude desarrollar satisfactoriamente mi tesis.

Doy las gracias a CONACYT, ya que gracias a su apoyo financiero proporcionado a partir de su programa de becas y del proyecto número 45306, pude sustentar mis gastos durante la realización de mi tesis.

Quisiera darle un especial agradecimiento a Sofía Reza quien ante cualquier duda y petición me atendía con una sonrisa y con la mejor disponibilidad.

También quisiera agradecer al departamento de Servicios Escolares y a la biblioteca de Ingeniería Eléctrica por los servicios y ayuda que me brindaron cuando lo necesité.

Resumen

Con la llegada del comercio electrónico, hoy en día se han introducido una serie de variantes sobre las formas tradicionales de comercio. Los sistemas de pago asociados a este tipo de comercio son aquellos que pueden realizarse utilizando dinero asociado a una tarjeta emitida por una entidad financiera o bien mediante dinero electrónico, utilizando un protocolo de pago específico.

El uso del dinero electrónico como modelo de pago surgió apenas en la década pasada, en la actualidad son varias las empresas y gobiernos que comienzan a apostar por este modelo de pago ya que poco a poco se empieza a vislumbrar que será el sistema de pago del futuro, esto debido a que el dinero electrónico pretende aprovechar las ventajas de los medios electrónicos y de igual forma las ventajas que presenta el dinero metálico para así conformar un sistema de pago más completo, seguro y fácil de usar. Sin embargo, en la actualidad aún no se ha adoptado de manera formal y estándar, una terminología determinada respecto del dinero electrónico.

Con el surgimiento de las redes inalámbricas y los dispositivos móviles ligeros tales como las PDA (del inglés Personal Digital Assistant), el objetivo principal de esta tesis de crear un monedero digital móvil utilizando los ya mencionados recursos tecnológicos y alguno de los protocolos de dinero electrónico propuestos en la literatura, los cuales ya tienen una eficiencia y seguridad comprobadas, esto resulta ser en una opción muy viable para lograr la pronta aceptación de este modelo de pago.

Así pues en esta tesis se realiza una implementación de un sistema de dinero electrónico. Dicho sistema puede trabajar para computadoras de escritorio, computadoras portátiles y para dispositivos móviles ligeros, en concreto PDA´s. Asimismo, en esta tesis se plantean algunas posibles aplicaciones que pueden dársele a los protocolos de dinero electrónico planteados en la literatura, para con ello, dar solución a diferentes problemas en los que se involucra un proceso de intercambio de algún producto o servicio por algún "token digital", el cual debe permitir que sea verificado únicamente por la composición del mismo, sin recurrir a una consulta directa con la entidad que emite dicho "token digital".

Abstract

With the advent of electronic commerce, quite a few new mechanisms have surged which have significantly modified the way that traditional commercial transactions are made.

Among different payment schemes presently used in modern electronic commerce systems, we can mention magnetic cards emitted by a financial organization and electronic money (e-cash) implemented by means of a specific security protocol payment.

E-cash paradigm was first proposed by Chaum in the mid-nineties of last century. Less than ten years after, there are a number of companies and governments which have already adopted this model of payment. We might guess that e-cash has the potential to become the dominant payment system in the near future. This prediction can be made based on the fact that electronic money strives to combine the main features of credit/debit cards on one side and traditional note banks, on the other.

E-cash design goal is thus, to come out with a safer, easier and more complete payment system. Nevertheless, despite all its potential, at present time there is no widely accepted standard, not even a specific terminology in the context of electronic money.

In this Thesis we address the problem of developing a reliable and trustable mobile electronic wallet, which includes the definition of four main entities: The Bank, The store, the Authority and the Clients. Clients in our system can be executed from light mobile platforms such as Personal Digital Assistants (PDA). Those clients can wirelessly interact with other system's entities which are typically hosted/executed in desktop/laptop personal computers. Moreover, our system implements three of the most prominent e-cash security protocols proposed in the open literature which have proven to be both, efficient and secure.

Finally, we have considered possible extensions of the e-cash concept in the context of contracting services by using some sort of "digital token". This token can be verified/authenticated solely by their composition, in an off-line fashion.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción al Dinero Electrónico

Desde tiempos remotos el hombre ideó sistemas para dar valor a las cosas y poder intercambiarlas, primero se utilizó el trueque, pero éste no fue una solución muy efectiva pues el comercio siguió en crecimiento y no fue suficiente, por lo que se tomó la determinación de adoptar ciertos productos que fueran aceptados de un modo general como unidad de cambio y medida de valor. De esta manera surge el concepto de dinero, el cual ha sido una de las invenciones más importantes en la historia de la humanidad [28].

En la actualidad el comercio es una actividad de la economía de los pueblos, destinada a relacionar a los sectores producción y consumo, que se realiza tanto en el área nacional como internacional, la moneda de cada uno de los paises se utiliza para medir las transacciones y en el campo internacional hay que correlacionar el valor de las diferentes monedas para facilitar la medida de compra y venta de bienes y servicios.

1.1. Definición de dinero

El concepto y uso del dinero común ha ido evolucionando con el paso del tiempo siendo cada vez mayor el grado de abstracción que éste proporciona.

Los economistas tradicionalmente han definido el dinero como un medio de intercambio socialmente aceptado, una representación abstracta de un valor (independiente del valor inherente al papel o metal) respaldada por una autoridad y generalmente admitida para la realización de intercambios comerciales [29]. La función obvia del dinero es ser un medio de intercambio, para esto debe cumplir con algunas características como: tener un alto valor por unidad, ser fácilmente divisible y difícil de falsificar. La segunda función del dinero es consistir en una forma simple y conveniente de almacenaje de poder de compra [29].

Hoy en día la moneda actual no tiene un valor intrínseco asociado, equivalente a la cantidad de dinero que representa; por lo tanto, su aceptación es resultado de un consenso. Se trata de unidades monetarias preestablecidas que, normalmente, una persona de una determinada región geográfica aceptaría. Su validez se basa en conceptos legales acompañados de la aceptación social [25]. Hoy en día podemos decir que una moneda es una pieza de cualquier

metal, regularmente en forma de disco y acuñada con los distintivos elegidos por la autoridad emisora para acreditar su legitimidad y valor [28] (análogo a este concepto es el concepto de billete).

El dinero metálico presenta las siguientes cualidades:

- Permite el anonimato del comprador.
- Es valido en cualquier lugar (dentro del contexto de su validez).
- Su manejo es fácil.

Sin embargo, también presenta las siguientes desventajas:

- Su propietario es quien lo porta.
- La portabilidad de grandes cantidades es peligrosa y difícil.
- Requiere que la transacción sea llevada a cabo en persona.

1.2. Definición de dinero electrónico

Con la llegada del comercio electrónico se introdujeron una serie de variantes sobre las formas tradicionales de comercio; en donde destacan aquellas transacciones que son realizadas a través de la red, con ubicaciones físicas del negocio del comerciante no identificables fácilmente por el comprador y donde además cada una de las transacciones realizadas pueden ser observadas por otras entidades ajenas a la operación. Los sistemas de pago asociados a este tipo de comercio son aquellos que pueden realizarse utilizando dinero asociado a una tarjeta emitida por una entidad financiera (dinero plástico) o bien mediante dinero electrónico, utilizando un protocolo de pago específico [25].

Se pretende que el dinero electrónico se comporte de manera semejante al dinero tradicional pero sustituyendo el soporte tradicional por el soporte electrónico, esto es, sustituyendo el papel por bits. En principio, de forma teórica, no tiene condicionamientos cuantitativos por lo que podría utilizarse tanto para pagos de pequeña cuantía como de media o gran cantidad [28].

Se pretende, pues, que el dinero electrónico ofrezca las mismas propiedades que el dinero metálico:

- Aceptación universal.
- Pago garantizado.
- Inexistencia de costes para el usuario.
- Anonimato.

Además de algunas propiedades extra que le pueden ser agregadas, tales como:

- Portabilidad de grandes cantidades sin problema.
- Rastreabilidad Condicional.

El uso del dinero electrónico como modelo de pago surgió apenas en la década pasada, en la actualidad son varias las empresas y gobiernos que comienzan a apostar por este modelo de pago ya que poco a poco se empieza a vislumbrar que será el sistema de pago del futuro, esto debido a que el dinero electrónico pretende aprovechar las ventajas de los medios electrónicos y de igual forma las ventajas que presenta el dinero metálico para así conformar un sistema de pago más completo, seguro y fácil de usar.

1.3. Dinero plástico vs. dinero electrónico.

Hoy por hoy los sistemas de pago electrónico constituyen una de las formas de comerciar más utilizadas en el mundo. Es por eso, que para situarnos en el marco referente al estado del dinero electrónico es necesario mostrar sus ventajas y desventajas tratando de compararlas con las ventajas y desventajas del dinero plástico (tarjeta de crédito/débito).

Como bien sabemos, el modelo de pago electrónico más común en la actualidad es el pago con tarjeta de crédito/débito. Si se realiza una compra en Internet utilizando una tarjeta de crédito/débito como medio de pago, la transacción comercial se ordena a través de una pagina WEB pero la validación y la realización efectiva del pago se efectúan a través de los circuitos tradicionales de procesamiento de las operaciones con tarjeta de crédito/débito, esto es, los ya muy conocidos "puntos de venta". Sin embargo, uno de los más importantes temores de los usuarios es la seguridad relacionada con el envío por la red de datos confidenciales de las tarjetas de crédito/débito. Una solución a este problema es el estándar de cifrado SET (Secure Electronic Transaction) que proponen las más importantes compañías de tarjetas de crédito [27]. Sin embargo, la desventaja más importante que presenta este modelo de pago electrónico es la falta de anonimato. Al dar el número de tarjeta en cada compra, se va dejando un rastro fácil de seguir, que permite recabar información acerca del consumidor, como poder adquisitivo, hábitos de compra, gustos personales, etc.

El uso del dinero plástico como modelo de pago electrónico presenta las siguientes ventajas:

- Permite transacciones de sumas de dinero muy grandes.
- Su portabilidad y forma de uso es fácil y segura.
- Para realizar la transacción no es necesario la presencia del comprador.
- Se puede obtener dinero en efectivo a partir de ésta.

y las siguientes desventajas:

- No permite el anonimato del comprador.
- Su validez depende de que el vendedor tenga los medios para verificar la transacción.
- Requiere que se verifique el saldo del cliente en línea para la autorización de la transacción.

El dinero electrónico constituye la solución más ambiciosa para sistemas de pago electrónicos y, aunque las características propias de este dinero podrían hacer pensar que puede ser trivialmente duplicado, los sistemas de dinero electrónico propuestos a lo largo de la historia han demostrado que es posible proveer mecanismos criptográficos con los cuales la entidad financiera pueda asegurarse que los usuarios no podrán hacer mal uso del sistema y de que, en caso que así ocurriera, el banco tendría los mecanismos necesarios para encontrar al culpable/tramposo.

El uso del dinero electrónico como modelo de pago electrónico presenta las siguientes ventajas:

- Mantiene el anonimato del comprador.
- No requiere que el vendedor esté conectado en el momento de la transacción.
- Su portabilidad y forma de uso es fácil y segura.
- Para realizar la transacción no es necesario la presencia del comprador.

Sin embargo tambien presenta problemas que aún no se resuelven del todo:

- Aún no es posible manejar cantidades de dinero muy grandes.
- Requiere de los medios electrónicos necesarios para su uso.
- La utilización de más de una vez de una moneda electrónica.

Con esta pequeña comparación podemos ver que los dos sistemas presentan ventajas y desventajas de igual manera es posible presumir que el dinero electrónico podría ser el siguiente paso en el proceso de evolución del dinero, pero para esto aún falta tiempo, pero no dudemos que es muy factible que se dé más pronto de lo que nos podamos imaginar.

1.4. Características del dinero electrónico

Para que un sistema de dinero electrónico pueda considerarse como tal, debe cumplir con ciertas propiedades. Todas ellas hacen que el sistema sea de menor o mayor calidad dependiendo de cuántas de estas propiedades cumpla el sistema. A continuación se describen lo que serían las propiedades deseables del dinero electrónico tal y como fueron propuestas en [4]:

- *Independencia:* La seguridad del dinero electrónico no puede depender de ninguna condición física. El dinero debe ser enviado a través de la red, por lo que su seguridad no puede depender de que dicha red sea segura.

- *Seguridad:* El dinero no puede ser copiado ni reutilizado. Debido a que estamos hablando de una moneda electrónica, ésta estaría constituida de bytes los cuales digitalmente podrían ser copiados y reutilizados sin ningún problema. Por lo que se deben de establecer mecanismos con los que se pueda establecer la autenticidad y la reutilización de dichas monedas.

- *Privacidad:* Se debe garantizar el anonimato del comprador, siempre y cuando las transacciones sean válidas. Cuando un comprador use monedas electrónicas no debe ser posible conocer su identidad a través de sus compras. Pero si éste intentara realizar algún tipo de fraude dándole un mal uso a sus monedas electrónicas, el banco sería capaz de obtener su identidad para después realizar las actividades legales en contra de éste.

- *Pago fuera de línea:* Las transacciones deben ser realizadas fuera de línea. Cuando una transacción se realice entre comprador y vendedor, el vendedor no debería de estar conectado con el banco para verificar el pago del comprador.

- *Transferibilidad:* El dinero puede ser transferido a otros. Esta propiedad permitiría a un usuario transferir sus monedas a otros quienes más adelante podrán usar dichas monedas sin ningún problema.

- *Divisibilidad:* Una "pieza" de dinero puede ser dividida en otras de menor denominación. Esto permite que los pagos no requieran un número exacto de monedas electrónicas y así disminuir tanto el tráfico como la cantidad de operaciones que se deben de hacer para validar cada una de las monedas.

En el siguiente capitulo utilizaremos estas propiedades como métrica para evaluar y hacer una comparación entre algunos de los protocolos más importantes.

1.5. Estado del arte

El dinero electrónico es relativamente nuevo ya que surgió en la década de los ochentas y desde entonces poco a poco se ha ido perfeccionando; de igual manera ha crecido su aceptación por las entidades financieras de todo el mundo. A continuación se presenta una breve reseña de la evolución que el dinero electrónico ha tenido desde su nacimiento hasta nuestros días.

En 1988 David Chaum [2] se da cuenta del problema que las transacciones electrónicas usando tarjetas de débito/crédito implican un problema con respecto al anonimato de los compradores. Chaum propone una forma de realizar pagos electrónicos de manera anónima e introduce el concepto de "dinero electrónico"por su similitud con el dinero metálico, el cual, garantiza el anonimato del comprador. Sin embargo, el concepto original de dinero electrónico presentaba dos limitaciones inherentes debido a su característica electrónica; éstas eran que el dinero podía ser copiado y reutilizado. Para evitar la reutilización de las monedas electrónicas, Chaum [2] propone que el banco mantenga una lista de todas las monedas gastadas y verifique cada una de las monedas depositadas con la lista, para que así se evite la reutilización de la misma. Para lograr esto, el banco, debería de estar en línea en el momento de la transacción o de lo contrario la tienda no podría tener la garantía de que el pago fuera válido.

El forzar a que el banco estuviera conectado al momento de realizarse el pago resultó ser una restricción muy fuerte que sería corregida por Chaum, Fiat y Naor en [3]. Este modelo es llamado *"dinero electrónico fuera de linea"* por su principal característica, la cual es que el protocolo trabaja fuera de línea. El hablar de un sistema de pago fuera de línea se refiere a que el proceso de depósito/cobro de las monedas electrónicas se realiza en una instancia de tiempo diferente a la del protocolo de pago/compra, es decir, que el vendedor o la tienda no necesita estar conectada al banco para verificar la validez de la moneda. Otra característica importante que aportan Chaum, Fiat y Naor en su protocolo es la garantía de anonimato, la cual se logra utilizando firmas digitales a ciegas basadas en RSA y utilizando un esquema de "corte y selección". Con estas dos herramientas en este protocolo se muestra que aunque las transacciones sean realizadas fuera de línea, el banco puede ser capaz de detectar cualquier tipo de fraude además de tener la posibilidad de obtener la identidad del tramposo de una manera casi inmediata. Debido al esquema de corte y selección (cut-and-choose), propuesto por Michael O. Rabin [1], utilizado en este protocolo, la oportunidad que se tiene para poder cometer algún fraude está determinado por el valor de k/2 donde k es el parámetro de seguridad utilizado por el banco y el cual debe ser mayor a dos. Dependiendo del tamaño de dicho parámetro la probabilidad de cometer fraude se reduce al aumentar el tamaño de k, debido a que la oportunidad de que algún comprador intente utilizar la misma moneda sin ser detectado estaría dada por la probabilidad de que fuera cuestionado por los mismos valores dos veces, es decir, 1 en $2^{k/2}$[17].

Más tarde Okamoto y Otha [4] proponen un protocolo en el cual ademas se establecen las características que deberían de ser deseables en el dinero electrónico (independencia, seguridad, privacidad, pago fuera de línea, transferibilidad, divisibilidad) las cuales ya fueron mencionadas en la sección anterior. Este protocolo utiliza como herramientas criptográficas: un esquema de mutuo acuerdo, la raíz cuadrada de un número módulo N y la representación de un árbol binario. Por otro lado la seguridad del sistema recae en el problema de factorización y en las funciones hash que éste utiliza. La evolución que presentan Okamoto y Otha a partir de lo propuesto por Chaum, Fiat y Naor se resume en dos características: la transferibilidad y la divisibilidad de la moneda electrónica, siendo la divisibilidad la característica que más relevancia presentó, para lograrla Okamoto y Otha utilizaron una estructura de un árbol binario que permite la divisibilidad de una pieza de dinero, de forma que, dicha pieza no pueda ser reutilizada ni falsificada. Sin embargo, más adelante, Okamoto y Otha reconocieron que

este protocolo puede ser rastreable electrónicamente. Debido a que este protocolo proponía un subprotocolo llamado "Protocolo de obtención de licencia", donde al cliente se le da un número licencia, con el cual más adelante obtendría las monedas a través del protocolo de retiro de fondos y a su vez el usuario puede gastar dichas monedas con el protocolo de pago/compra. Debido a que el número de licencia es constante y se utiliza en los protocolos de retiro de fondos y en el protocolo de pago/compra el número de licencia puede ser ligado con las monedas gastadas y de esta forma rastrear electrónicamente los pagos realizados por el comprador, comprometiendo así la propiedad del anonimato del sistema.

En 1993 surgen los protocolos llamados "de un sólo término"(single-term) los cuales son mucho más eficientes que los protocolos de corte y selección, ya que en lugar de usar k/2 términos para crear y satisfacer el reto, en este esquema sólo es necesario un término para comprobar la autenticidad de la moneda o para crear la firma que le dará validez. El primero en crear este tipo de protocolos fue N. Ferguson en [10] quien propone un protocolo donde combina firmas de RSA y firmas a ciegas aleatorias. En este sistema se propone el uso del esquema de compartición de secretos con el cual se atrapa a quien intente cometer un fraude. Desafortunadamente, la seguridad del sistema resulta ser no comprobable, y su propuesta se ve opacada por el surgimiento del protocolo propuesto por Brands [9].

S. Brands en [9] propone un protocolo, el cual es un refinamiento de los protocolos propuestos por: Brands, Chaum, Cramer, Ferguson y Pedersen [6], Chaum y Pedersen [7] y Cramer y Pedersen [8]. En este protocolo Brands hace uso de las propiedades homomórfas de los logaritmos discretos, basándose para ello en dos conceptos previamente estudiados: las firmas digitales de Schnorr y el problema de representación en grupos de orden primo. Hoy en día este protocolo es considerado como uno de los más eficientes en la historia del dinero electrónico, pues muchas de las nuevas ideas que han surgido en la actualidad usan como base este protocolo.

Cuando en 1996 ya se hace una realidad más palpable la posibilidad de usar el dinero electrónico como otra forma de pago, algunos investigadores como M. Jakobsson y M. Yung se dan cuenta de que con los sistemas convencionales de transacciones electrónicas algunos delitos como: el chantaje, el lavado de dinero y las compras ilegales, podían ser rastreados y resueltos sin mayor problema; pero que, el usar algún sistema de dinero electrónico hacía de estos delitos el "crimen perfecto", esto debido a la garantía de anonimato que ofrecían dichos sistemas. Es por ello que en [11] dichos autores propusieron un sistema de dinero electrónico dónde es posible realizar de manera incondicional la cancelación de los fondos en la cuenta del usuario y/o dónde el anonimato del usuario podría revocarse; esto siempre bajo la autorización de una Corte de Justicia, la cual determinaría si se volvía necesaria la suspención de las garantías a las que tiene derecho el usuario del sistema. Para lograrlo, ellos proponen que exista una "organización de los derechos del consumidor"la cual llevaría un registro de todas las transacciones de retiro de monedas electrónicas y de esta manera si se demuestra, ante las leyes civiles, que resulta necesario conocer la identidad o el rastro de alguna moneda, entonces esta organización debería de proporcionar la información necesaria para la obtención de tales datos. Por supuesto que se asume que dicha organización resultaría de confianza tanto para el banco como para los usuarios del sistema. Sin embargo, para lograr

esto, en el sistema propuesto en [11] se vuelve a caer en el paradigma de dinero electrónico en línea ya que no sólo el banco debería de estar en comunicación con la "organización de consumidores"al momento del proceso de retiro de monedas digitales, sino que también, al momento de realizar el proceso de pago/compra el vendedor tendría que verificar la validez del retiro de la moneda que está recibiendo con la "organización de consumidores". Esta propuesta dió pie a que se presentaran nuevos esquemas de dinero electrónico en los que se intentaría lograr un equilibrio entre el anonimato del usuario y la rastreabilidad del mismo.

Es así como Chan, Frenkel y Tsiounnis [12] introducen un nuevo esquema al cual ellos bautizan con el nombre de: "Sistema Imparcial de Dinero Electrónico Fuera de Línea", al cual por sus siglas en inglés hoy en día se le conoce como FOLC (Fair Off-Line E-Cash). En su propuesta ellos agregaron una cuarta entidad al modelo básico propuesto en [3], a esta entidad la llamaron la "autoridad"con la cual ellos pretendian garantizar el anonimato del comprador siempre y cuando éste se mantenga al margen de la ley, de lo contrario el banco o cualquier otra institución encargada de hacer cumplir las leyes, podría solicitar ya sea el rastreo de una moneda o bien el rastreo del propietario de una moneda. La principal ventaja de este nuevo esquema reside en el hecho de que la entidad "autoridad"se mantiene fuera de línea en todos los procesos (retiro, pago/compra y depósito). Esto se logra a través de la utilización del concepto de "pruebas indirectas de discurso"(Indirect Discourse Proofs). Así mismo al esquema básico le son agregados dos subprotocolos: el protocolo de rastreo de monedas y el protocolo de rastreo de propietarios de monedas, en los cuales sólo interactuarán el banco y la autoridad. Con esto, Chan, Frenkel y Tsiounnis trataron de impedir delitos tales como: el lavado de dinero, los chantajes o pagos de rescate, etc. Todo esto de una manera eficiente sin afectar los desempeños de los demás protocolos. Este trabajo es considerado como una extensión del trabajo realizado por M. Jakobsson y M. Yung, además ellos se basan en el protocolo de S. Brands, al cual le hacen las modificaciones pertinentes para lograr convertirlo en un sistema FOLC.

Cuando el problema del equilibrio que debía existir entre el anonimato y la rastreabilidad quedó resuelto por diversas propuestas se retomó el tema de la divisibilidad, que Okamoto y Otha en [4] propusieran y resolvieran en su momento de manera muy eficiente, pero que ya para ese entonces, con las nuevas técnicas desarrolladas resultaba ineficiente y que además no permitía una rastreabilidad condicional. Es así como de 1997 a 1999 surgen diversos trabajos, en cuanto a la divisibilidad de la moneda electrónica respecta. De estos trabajos, uno de los más interesantes es el trabajo de A. Chan, Y. Frenkel y Y. Tsiounnis [18] en 1998 quienes proponen un esquema muy eficiente para lograr la divisibilidad y más aún, su propuesta permite el uso de los métodos de rastreo o de revocación de anonimato existentes hasta entonces.

Más tarde, con el surgimiento de diversos esquemas de firmas de grupo, comienzan a surgir aplicaciones como el dinero electrónico utilizando este tipo de firmas, tal es el caso de Maitland Boyd en [20] quienes presentan un protocolo de dinero electrónico basándose en el trabajo realizado por Ateniese, Camenisc, Joye y Tsudik en [19] con el cual pretendían usar este tipo de firmas para lograr un sistema de dinero electrónico más eficiente, sin embargo el protocolo propuesto no permitía ni la revocación del anonimato ni la rastreabilidad del usuario

y/o las monedas. No fue hasta el 2004 cuando utilizando este mismo esquema Popescu en [23] le agrega la revocación del anonimato para hacerlo más completo aunque no tan eficiente como los propuestos anteriormente.

Con el surgimiento de las redes inalámbricas, los dispositivos móviles de tercera generación y las PDAs, la idea de establecer un monedero digital dentro de estos dispositivos y establecer un comercio electrónico móvil resultó ser buena y de gran aceptación. Existen algunas propuestas como en el caso de Woosek Ham en su tesis de maestría [21]. Su propuesta está enfocada a dispositivos móviles, sin embargo, un defecto muy notorio en este protocolo es la publicación de las monedas electrónicas que debe realizar el banco ante los vendedores. El publicar estos datos pone seriamente en duda su característica de fuera de línea.

En la actualidad la implementación de protocolos antiguos, tales como los propuestos en [3, 4, 9, 10], los cuales ya tienen una eficiencia y seguridad comprobadas, resulta ser la opción más viable para la adaptación/implementación de estos sistemas en dispositivos móviles.

1.6. Esquemas de dinero electrónico

En la actualidad los sistemas de dinero electrónico son implementados utilizando dos tipos de esquemas: el básico y el FOLC.

1.6.1. Esquema básico de dinero electrónico

El esquema básico fue el primero en surgir, fue propuesto por Chaum, Fiat y Naor en [3] y desde que surgió ha sido el modelo base para la creación tanto de nuevos protocolos como de nuevos esquemas, estos últimos agregando entidades o comportamientos diferentes entre las entidades. Este esquema está constituido por tres entidades:

- El *banco*: Es la entidad financiera, la cual será la encargada de proporcionar y respaldar el dinero electrónico que se le dé al comprador. Así mismo en ella recaerá la responsabilidad de atrapar a quien intente realizar algún tipo de fraude.
- El *comprador*: Será la entidad que utilice el dinero electrónico para gastarlo.
- El *vendedor*: Esta entidad tiene como fin el intercambiar sus servicios o productos por monedas electrónicas, las cuales, deberá verificar con sus propios medios asegurando la autenticidad de éstas. Y más adelante podrá abonar dichas monedas a su cuenta en el banco.

La interacción que dichas entidades desempeñan entre sí y el proceso general que lleva a cabo el protocolo, se podría ver a través del siguiente esquema simplificado:

Normalmente todo el esquema anterior es dividido en tres fases, teniendo cada una de ellas su protocolo específico y cada una de ellas realizadas en instancias de tiempo diferentes:

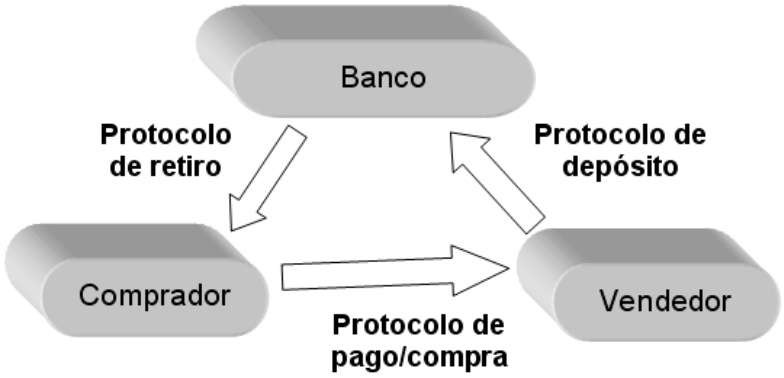

Figura 1.1: Modelo básico

- El protocolo de retiro de fondos: Es la interacción entre el banco y el comprador para la obtención de monedas electrónicas.
- El protocolo de pago/compra: Es donde el comprador gasta sus monedas intercambiándolas por algún producto o servicio que ofrece el vendedor.
- El protocolo de depósito/cobro: Es la última fase del esquema general en donde se establecen los requisitos que deberá cumplir el vendedor para que pueda abonar una moneda a su cuenta en el banco.

Cabe aclarar que aparte de estos tres subprotocolos, la entidad financiera debe de contar con alguna forma de identificar a la persona que intente realizar un doble gasto de alguna moneda.

De esta forma se componen la mayoría de los protocolos de dinero electrónico; aún así, existen diversas propuestas las cuales muestran ventajas y desventajas y cada una de manera diferente.

1.6.2. Esquema FOLC de dinero electrónico

Este esquema es una extensión del modelo de dinero electrónico básico fuera de línea y fue propuesto por Yung-Frankel-Tsiounnis en [16] donde se presenta un nuevo concepto llamado "Fair Off-line e-Cash" (FOLC), sistema en el cual se pretende lograr un balance entre la necesidad del anonimato y la prevención de actividades criminales. Esta arquitectura está compuesta por las mismas tres entidades que componen el modelo básico (*banco, comprador* y *vendedor*) mas una cuarta entidad la cual esta definida como, la *autoridad*:

- La *autoridad*: será la encargada de proporcionar la información del rastreo de una moneda o del propietario de una moneda, siempre y cuando esté justificado legalmente. Esta acción podrá realizarse en cualquier momento simplemente con la sospecha de la infracción de algún delito por parte del *comprador*.

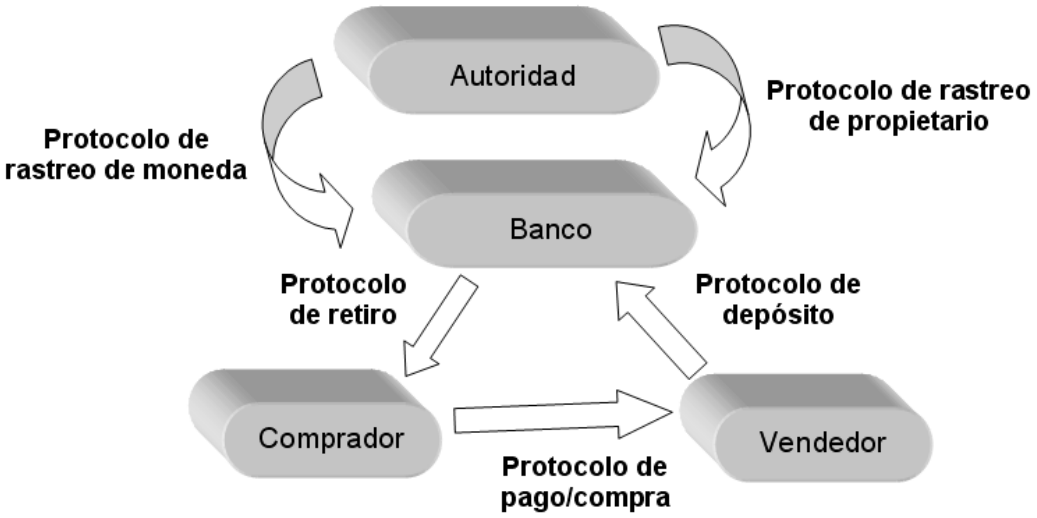

Figura 1.2: Modelo FOLC

Así mismo se puede observar en la figura siguiente que el modelo FOLC está compuesto por 5 protocolos, 3 de los cuales son los mismos del esquema básico del dinero electrónico fuera de línea (protocolo de retiro, protocolo de pago/compra y protocolo de depósito).

Los otros dos protocolos incorporados al esquema básico son los siguientes:

- Protocolo de rastreo de propietario: este protocolo es llevado a cabo por la entidad *banco* y la entidad *autoridad* y se utiliza para rastrear la identidad del propietario de una moneda especifica. Para lograrlo el *banco* le envía a la *autoridad* una "vista" de lo que recibió en el protocolo de depósito y la *autoridad* le devuelve una cadena que contiene la información con la cual el *banco* podrá obtener la identidad del cliente vía la base de datos de la cuentas bancarias.
- Protocolo de rastreo de moneda: este protocolo también es llevado a cabo por la entidad *banco* y la entidad *autoridad* y se utiliza para rastrear una moneda que fue creada a partir del protocolo de retiro. Para lograrlo el *banco* le proporciona a la autoridad una "vista"del protocolo de retiro con la cual la autoridad sería capaz de obtener cierta información con la cual el *banco* sería capaz de saber de acuerdo a la información de los protocolos de depósito que él posee, donde fue gastada la moneda.

Capítulo 2

Fundamentos Criptográficos

La palabra *criptografía* proviene del griego *kryptos* que significa oculto, y *graphia*, que significa escritura. Su definición formal nos dice que la criptografía es el estudio de las técnicas matemáticas relacionadas con aspectos de seguridad de la información [14], entre las disciplinas que engloba cabe destacar la Teoría de la Información, la Complejidad Algorítmica y la Teoría de Números.

A través de la criptografía, la información puede ser protegida contra el acceso no autorizado, su interceptación, su modificación y la inserción de información extra. También puede ser usada para prevenir el acceso y uso no autorizado de los recursos de una red o sistema informático y para prevenir a los usuarios la denegación de los servicios a los que sí están permitidos. Actualmente, la criptografía es la metodología para proporcionar la seguridad de las redes informaticas, para lograrlo la criptografía provee de los siguientes servicios o funciones de seguridad[15]:

- Confidencialidad: los usuarios autorizados tienen acceso a la información.
- Integridad de la información: garantía ofrecida a los usuarios de que la información original no será alterada, ni intencional ni accidentalmente.
- Autenticación
  - De usuario: es un proceso que permite al sistema verificar si el usuario que pretende acceder o hacer uso del sistema es quien dice ser.
  - De remitente: es el proceso que permite a un usuario certificar que el mensaje recibido fue de hecho enviado por el remitente y no por un suplantador.
  - Del destinatario: es el proceso que permite garantizar la identidad del usuario destinatario.
- No repudio
  - En origen: que cuando se reciba un mensaje, el remitente no pueda negar haber enviado dicho mensaje.

  - En destino: que cuando se envía un mensaje, el destinatario no pueda negar haberlo recibido cuando le llegue.

- De actualidad (no replay) : consiste en probar que el mensaje es actual, y que no se trata de un mensaje antiguo reenviado.

En este capítulo se presentan los fundamentos más importantes para la comprensión de las técnicas criptográficas empleadas para los sistemas de dinero electrónico. En la sección 2.1 revisaremos de manera general la teoría de números, enfocandonos a los conceptos básicos. La sección 2.2 revisará el concepto de Función Hash. En la sección 2.3 veremos lo concerniente a los sistemas criptográficos de llave pública. Algunos de los sistemas de firma digital serán revisados en la sección 2.4 y por último en la sección 2.5 veremos las firmas a ciegas.

2.1. Teoría elemental de números

La teoría elemental de números es uno de los pilares más importantes en la criptografía, es por ello que en esta sección se revisan los fundamentos generales necesarios para establecer las bases matemáticas para la comprensión cabal de las herramientas criptográficas que se usan en los sistemas de dinero electrónico.

2.1.1. Nociones básicas

Comenzaremos definiendo las nociones básicas de la teoría elemental de números [22]:

Definición 1 (Números Enteros)
Los números enteros se definen como el conjunto de los números Z={...,-2,-1,0,1,2,3,...}. Dentro de este conjunto está el subconjunto de los números naturales, N={1,2,3,4,...}. Es decir, el subconjunto de los números enteros positivos (mayores que 0).

Definición 2 (Divisibilidad)
Sean a y b dos enteros con $a \neq 0$ podemos decir que a divide a b si existe un entero k tal que $b = ak$. Esto es denotado por $a|b$. También se dice que a es un factor o divisor de b, y que b es un múltiplo de a. Algunas propiedades de la divisivilidad son:

- *Para todo $a \neq 0$, $a|a$. De igual forma $1|b$ para cualquier b.*
- *Si $a|b$ y $b|c$ entonces $a|c$.*
- *Si $a|b$ y $a|c$ entonces $a|(sb + tc)$ para cualquier entero s y t.*

Así pues, de la formula de la división entera, la cual nos dice que el dividendo a es igual al divisor b multiplicado por el cociente q, más un resto r, se obtiene el teorema de la división el cual nos dice que:

Teorema 1 (Teorema de la división)
Sean $a \in Z$ y $b \in N$. Entonces existen q, $r \in Z$ con $0 \leq r < b$ tales que $a = mq + r$. Además q y r son únicos.

Definición 3 (Máximo Común Divisor)
Dados dos enteros a y b distintos de 0*, decimos que el entero* $d > 1$ *es un máximo común divisor, o MCD, de a y b si* $d|a$, $d|b$ *y para cualquier otro entero c tal que* $c|a$ *y* $c|b$ *entonces* $c|d$*. En otras palabras d es el entero positivo mayor que divide tanto a a como a b. Algunas propiedades del máximo común divisor son:*

- *mcd(a,b) = mcd(|a|,|b|)*
- *mcd (ka,kb) =k mcd(a,b)*
- *mcd(a,b) = d* $\iff$ $d|a, d|b$ *y mcd(a/d,b/d)=1*

Es posible calcular el máximo común divisor mediante un procedimiento que se conoce como el algoritmo de Euclides [22], el cual, se describe en el algoritmo 1.

Algoritmo 1 Algoritmo de Euclides (cálculo del máximo común divisor)

Entrada: dos enteros positivos a y b donde $a \geq b$.
Salida: el máximo común divisor de a y b.

```
1: while b ≠ 0 do
2:     r ← a mod b, a ← b, b ← r
3: end while
4: return a
```

Definición 4 (Números Primos)
Decimos que un número entero $p > 1$ *es primo si sus únicos divisores positivos son* 1 *y p.*

Definición 5 (Primos Relativos)
Se dice que dos enteros a y b son primos relativos si mcd(a,b)=1*.*

Definición 6 (Números Compuestos)
Si un número entero $q > 1$ *no es primo, se le llama número compuesto. Por tanto, un entero q será compuesto si y sólo si existen a,b enteros positivos (menores que q) tales que* $q = ab$*.*

Teorema 2 (Teorema Fundamental de la Aritmética)
Cualquier número natural $n > 1$ *o bien es primo, o bien se puede descomponer como un producto de potencias de números primos,* $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$*, en donde* p_i *es un numero primo y* e_i *es un numero entero positivo. Además esta descomposición es única (salvo el orden de los factores).*

2.1.2. Aritmética modular

Definición 7 (Congruencia)
Dado $m \in Z$ *,* $m > 1$*, se dice que* $a, b \in Z$ *son congruentes módulo m si y sólo si* $m|(a - b)$*. Se denota esta relación como* $a \equiv b(mod m)$*. Donde m es el módulo de la congruencia. Es importante darse cuenta de que si m divide a* $a - b$*, esto supone que ambos a y b tienen el mismo resto al ser divididos por el módulo m.*

La relación de congruencia módulo m es una relación de equivalencia para todo $m \in Z$. Es decir, cumple con las siguientes propiedades:

1. Reflexiva: Si $a \in Z$ entonces $a \equiv a(\mathrm{mod}\, m)$.
2. Simétrica: Con $a, b \in Z$ tenemos que: si $a \equiv b(\mathrm{mod}\, m)$ entonces $b \equiv a(\mathrm{mod}\, m)$.
3. Transitiva: Con $a, b, c \in Z$ tenemos que: si $a \equiv b(\mathrm{mod}\, m)$ y $b \equiv c(\mathrm{mod}\, m)$ entonces $a \equiv c(\mathrm{mod}\, m)$.

Así mismo es posible definir al conjunto Z_m como el conjunto de los residuos módulo m, el cual esta conformado por $Z_m = \{0, 1, 2, ..., m-1\}$. Del teorema de la división es fácil ver que para todo entero a existe un residuo r, el cual se encontraría dentro del conjunto Z_m.

Si $m \in N$ y $a, b, c, d \in Z$ tales que $a \equiv b(\mathrm{mod}\, m)$ y $c \equiv d(\mathrm{mod}\, m)$ entonces las siguientes propiedes se cumplen:

1. $a + c \equiv b + d(\mathrm{mod}\, m)$
2. $a - c \equiv b - d(\mathrm{mod}\, m)$
3. $a.c \equiv b.d(\mathrm{mod}\, m)$

De las propiedades anteriores podemos concluir que en la aritmética modular las operaciones de suma, resta y multiplicación son realizadas de igual forma que en la aritmética normal, teniendo sólo la diferencia de que cada resultado intermedio es reducido módulo m. En seguida se explica con más detalle cómo se realizan cada una de las operaciones en aritméticas básicas modulares:

Suma y Resta Modular. Si $a, b \in Z_m$ entonces podemos definir la operación suma $a + b(\mathrm{mod}\, m)$ como un elemento dentro de Z_m. Ya que para realizar esta operación primero habría que sumar a y b como si fuera aritmética normal para despues dividir el resultado por m y reportar como resultado de la suma el residuo. Este residuo será el resultado de la suma de a y b módulo m.

Por ejemplo: $17 + 20(\mathrm{mod}\, 22) = 15$

Las propiedades más importantes de una suma modular son:

1. Es conmutativa $a + b(\mathrm{mod}\, m) = b + a(\mathrm{mod}\, m)$.
2. Es asociativa $(a + b) + c(\mathrm{mod}\, m) = a + (b + c)(\mathrm{mod}\, m)$.
3. Tienen elemento neutro (0), con el cual $a + 0 = a(\mathrm{mod}\, m)$.
4. Propiedad Cancelativa. Para toda a y b en Z_m existe un único valor x en Z_m con el cual $a + x = b(\mathrm{mod}\, m)$.

La última propiedad nos permite definir la resta modular: poniendo el valor de $b = 0$, podemos ver que para cada a en Z_m existe un único valor x en Z_m tal que $a + x \equiv 0 (\mathrm{mod}\, m)$.

Multiplicación Modular. Si $a, b \in Z_m$ entonces podemos definir la operación multiplicación $a \cdot b (\mathrm{mod}\, m)$ como un elemento dentro de Z_m. Ya que para realizar esta operación primero habría que multiplicar a y b como si fuera aritmética normal para después dividir el resultado por m y reportar como resultado de la multiplicación el residuo. Este residuo será el resultado de la multiplicación de a y b módulo m.

Por ejemplo: $17 \cdot 20 (\mathrm{mod}\, 22) = 10$.

Las propiedades más importantes de una multiplicación modular son:

1. Es conmutativa $a \cdot b (\mathrm{mod}\, m) = b \cdot a (\mathrm{mod}\, m)$.
2. Es asociativa $(a \cdot b) \cdot c (\mathrm{mod}\, m) = a \cdot (b \cdot c) (\mathrm{mod}\, m)$.
3. Tienen elemento neutro (1), con el cual $a \cdot 1 = a (\mathrm{mod}\, m)$
4. Propiedad cancelativa: si mcd (m,c)=1 y $a \cdot c \equiv b \cdot c (\mathrm{mod}\, m)$, entonces $a \equiv b (\mathrm{mod}\, m)$. Si m es un número primo se garantiza que esta propiedad siempre se cumpla.

A partir de esta última propiedad podemos hacer la siguiente definición:

Definición 8 (Inverso Modular)
Se dice que un entero a tiene inverso módulo m si existe un entero b tal que $1 \equiv ab (\mathrm{mod}\, m)$. *Así pues el entero b será el inverso de a y se denota como* a^{-1}, *el inverso de un número* $a (\mathrm{mod}\, m)$ *existirá si y sólo si existen dos números enteros x, y tales que* $ax + my = 1$ *y estos números existirán si y sólo si mcd(a,m)=1. Para obtener el inverso modular de un número a se recurre al algortimo de Euclides extendido [22], con el cual es posible encontrar la solución a la ecuación* $ax + my = 1$.

División Modular. Con la definición anterior podemos decir que si $a, b \in Z_m$ y m es un número primo podemos realizar la división de a entre b definiéndola como una multiplicación de $a \cdot b^{-1} (\mathrm{mod}\, m)$, donde b^{-1} es el inverso modular de b.

Por ejemplo: $17/20 (\mathrm{mod}\, 23)$, esto sería igual a $17 \cdot (20)^{-1} (\mathrm{mod}\, 23)$, en donde $(20)^{-1} (\mathrm{mod}\, 23) = 15$. Por lo tanto $17/20 (\mathrm{mod}\, 23) = 17 \cdot 15 (\mathrm{mod}\, 23) = 2$.

Recalcando una vez más que la operación de la división sólo estará definida para cuando mcd(b,m)=1.

Exponenciación Modular En aritmética módulo m la exponenciación modular es menos costosa de realizar que en la aritmética entera. Nótese que $a \cdot b \bmod m = [(a \bmod m) \cdot (b \bmod m)] \bmod m$, y que por muy grande que sea el exponente, nunca es necesario multiplicar por enteros mayores que m.

Algoritmo 2 Algoritmo extendido de Euclides.

Entrada: dos enteros positivos a y b donde $a \geq b$.
Salida: d =MCD(a, b) y dos enteros x,y los cuales satisfacen la ecuación $ax + by = d$.

```
1: if b = 0 then
2:    d ← a, x ← 1, y ← 0
3:    return (d, x, y)
4: end if
5: x1 ← 0, x2 ← 1, y1 ← 1, y2 ← 0
6: while b > 0 do
7:    q ← ⌊a/b⌋, r ← a − qb, x ← x2 − qx1, y ← y2 − qy1
8:    a ← b, b ← r, x2 ← x1, x1 ← x, y2 ← y1, y1 ← y
9: end while
10: d ← a, x ← x2, y ← y2
11: return (d, x, y)
```

Además hay un método que nos permite ahorrar pasos de cálculo. La forma más obvia para calcular por ejemplo, $12^{26} (\mathrm{mod}\, 23)$ es multiplicar por 12 un total de 25 veces, reduciendo módulo 23 tras cada multiplicación. Pero un método más eficiente que sólo necesita seis multiplicaciones está basado en la siguiente observación:

Por ejemplo:
$12^2 = 144 = 6 (\mathrm{mod}\, 23)$
$12^4 = 62 = 36 = 13 (\mathrm{mod}\, 23)$
$12^8 = 132 = 169 = 8 (\mathrm{mod}\, 23)$
$12^{16} = 82 = 64 = 18 (\mathrm{mod}\, 23)$

Con esto se han calculado cuatro cuadrados, y como podemos descomponer el exponente en potencias de 2, así $26 = 16 + 8 + 2$, se puede reescribir el cálculo anterior como:

$12^{26} = 12^{(16+8+2)}$
$= 12^{16} * 12^8 * 12^2$
$= 18 * 8 * 6 = 864 = 13 (\mathrm{mod}\, 23)$

Este algoritmo se conoce como el algoritmo binario para la exponenciación [14], a continuación se detalla este algoritmo:

Raices Primitivas Para comprender el concepto de raíz primitiva es necesario primero establecer las siguientes definiciones:

Definición 9 (Función ϕ de Euler)
Se define la función ϕ de Euler como la función $\phi : N \Rightarrow N$ que a cada n le hace corresponder el número de enteros $x(1 < x < n)$, que son primos relativos con n, esto es, cuyo mcd(x,n)=1. Si n es un número primo p entonces cualquier x será primo relativo con él, por lo tanto siempre se cumplirá que $\phi(p) = p - 1$.

Algoritmo 3 Algoritmo binario para la exponeneciación

Entrada: $a \in Z_n$ y un entero $0 \leq k < n$, donde la representación binaria de k es $k = \sum_{i=0}^{t} k_i 2^i$.
Salida: $a^k \bmod n$.

```
1: b ← 1, if k = 0 return b
2: A ← a
3: if k_0 = 1 then
4:    b ← a
5: end if
6: for i ← 1 to t do
7:    A ← A^2 mod n
8:    if k_i = 1 then
9:       b ← A · b mod n
10:   end if
11: end for
12: return (b)
```

Teorema 3 (Teorema de Euler)
Si un número $a \in Z$ y verifica que mcd(m,a)=1 entonces $a^{\phi(m)} \equiv 1(\bmod m)$.

Definición 10 (Orden de un número x)
Si x y m son primos relativos, llamamos orden de x módulo m al menor entero r tal que $a^r = 1(\bmod m)$.

Definición 11 (Raíz Primitiva)
Sea m un número primo y $g \in Z_m$, entonces decimos que g es una raíz primitiva módulo m, si y sólo si el orden de g módulo m es igual al valor de la función $\phi(m)$. Por el teorema de Euler podemos observar que siempre existirá una solución pues al menos $g^{\phi(m)} = 1(\bmod m)$.

Sea g una raíz primitiva de un número primo p, las siguientes propiedades se deben de cumplir:

1. Si n es un entero, entonces $g^n \equiv 1(\bmod p)$ si y sólo si $n \equiv 0(\bmod p-1)$.
2. Si j y k son dos enteros, entonces $g^j \equiv g^k(\bmod p)$ si y sólo si $j \equiv k(\bmod p-1)$.

2.1.3. El problema de la factorización

Factorizar un número significa encontrar los números primos tales que al multiplicarlos nos den como resultado el número que estamos factorizando[22]. Más formalmente y acotando el problema, si tenemos un número compuesto n el cual esta dado por:

$n = pq$

El problema reside en encontrar los número primos p y q a partir del conocimiento de n.

Por ejemplo:

$10 = 2 \cdot 5 \qquad 903 = 21 \cdot 43$

Este es uno de los problemas más viejos de la teoría elemental de números y aunque parezca fácil al principio conforme se hace más grande el número que se desea factorizar, el tiempo para encontrar los factores primos crece exponencialmente [15]. El problema de la factorización tiene su fundamento el teorema fundamental de la aritmética descrito en la sección 2.1.1.

Actualmente bajo este problema, el cual es computacionalmente muy difícil de resolver, está basada la seguridad de algunos algoritmos, tal es el caso de RSA, el cual revisaremos más adelante.

2.1.4. El problema del logaritmo discreto

Otro de los problemas de la teoría elemental de números es el llamado Problema del Logaritmo Discreto [22], el cual nos dice que:

Dado un primo p y sean a y x dos números enteros diferentes de 0 módulo p es fácil calcular β:

$\beta \equiv a^x (\mathrm{mod}\, p)$

Sin embargo realizar el problema inverso, es decir encontrar el valor de x donde $a^x \equiv \beta (\mathrm{mod}\, p)$, en aritmética modular es un problema difícil.

Por ejemplo: Si $2^x \equiv 9 (\mathrm{mod}\, 11)$ tendríamos que $2^6 \equiv 2^{16} \equiv 2^{26} \equiv 9 (\mathrm{mod}\, 11)$. Cualquiera de los tres valores $(6, 16, 26)$ sería correcto. Sin embargo el problema sería casi imposible de resolver con los medios computacionales actuales si usaramos un número de 1024 bits.

De los algoritmos criptográficos más importantes que tienen sus bases matemá-ticas en la dificultad de resolver este problema tenemos al algoritmo de ElGamal y al algoritmo DSA.

2.2. Funciones hash

Un algoritmo de hash es una función que toma un mensaje M de longitud arbitraria y devuelve como resultado una secuencia de caracteres de longitud fija [15] (comunmente 128 o 160 bits). El resultado de aplicar la función Hash $H(M)$ a una secuencia de caracteres se denomina digesto, valor de hash o simplemente hash (h).

Una función hash debe cumplir con las siguientes propiedades:

1. Dado un M cualquiera, la función hash $H(M)$ debe ser facil de calcular.
2. Dado un M cualquiera, obtener M a partir de $H(M)$ debe ser computacionalmente difícil.

3. La función hash debe ser resistente a colisiones, es decir, computacionalmente no debe ser posible encontrar M y M' tales que $H(M) = H(M')$.

Las funciones hash son aplicadas principalmente para resolver problemas relacionados con la integridad de mensajes, lo que se conoce como MDC (Modification Detection Codes) , en este caso digamos, de manera muy general, que a un mensaje M se le aplica un algoritmo hash y se manda junto con el propio mensaje, al recibirlo el receptor aplica la misma función hash al mensaje M y comprueba que sea igual al hash recibido, esto únicamente tiene como objetivo el proteger de errores en la transmisión o bien en el almacenamiento de los datos contenidos en M.

Otra de las aplicaciones que las funciones hash tienen son aquellas relacionadas con los procesos de verificación de la autenticidad de mensajes y de su origen, lo que se conoce como MAC (Massage Authentication Codes), para lograr esto, se combina el mensaje M con una llave secreta K, se les aplica un hash $H(M, K)$, y se envía $[M, H(M, K)]$. El receptor puede comprobar la autenticidad y la integridad del mensaje porque conoce la llave secreta K y puede recalcular $H\prime(M, K)$. Si $H(M, K) = H\prime(M, K)$ la verificación es exitosa.

Otra aplicación que se le ha dado a las funciones hash es para el almacenamiento de contraseñas de acceso. Esto con el fin de que sólo el conocedor de la contraseña C podría ingresar de manera correcta al sistema ya que el algoritmo de autenticación verificaría el hash $H(C)$ almacendo con el hash $H(C')$ calculado a partir de la contraseña tecleada por el usuario en el momento de la autenticación.

Una tercera aplicación de las funciones hash es la generación de números pseudoaleatorios. En donde dado un número aleatorio s (la semilla), se le aplica una función f para crear una secuencia de numeros a partir de la semilla s, s_0, s_1, s_2,...,s_t, después a los valores obtenidos de esta secuencia se le aplica una función hash H para con ello obtener los valores aleatorios H(s), H(s_0), H(s_1), H(s_2),...,H(s_t).

Los algoritmos de hash más conocidos y usados en la actualidad son: SHA-1 y MD5. El primero fue desarrollado por la NSA, para ser incluido en el estandar DSS (Digital Signature Standard) procesa bloques de 512 bits y produce una salida de 160 bits. El segundo es el resultado de una serie de mejoras que su diseñador Ron Rivest ha llevado a cabo a partir de su algoritmo MD2 pasando por MD4 hasta llegar a lo que hoy conocemos como MD5. De igual manera que el SHA-1, MD5 procesa tramás de 512 pero produce una salida de 128 bits.

2.3. Sistemas de llave pública

Un algoritmo de llave pública realiza el cifrado y un descifrado de un mensaje M utilizando dos llaves diferentes [15], una llave a la cual llamaremos llave privada K_{priv}, la cual debe mantenerse en secreto y otra llave a la cual llamaremos llave pública K_{pub}, la cual, deberá darse a conocer a todas aquellas personas o entidades con las cuales se desea mantener una comunicación segura. De tal forma que para que un remitente B pueda enviar un mensaje cifrado a un receptor A, deberá contar con una función de cifrado E que recibirá como

parámetros de entrada el mensaje y la llave pública del receptor, dicha función de cifrado debe garantizar que sólo A podrá descifrar el mensaje utilizando su llave privada, de tal forma que si tenemos que la cifra esta dada por:

$$C = E(M, K_{pubA})$$

Entonces la única manera de obtener el mensaje M es aplicando la función de descifrado D la cual tendrá como parámetros la cifra C obtenida del proceso anterior y la llave privada de la entidad receptora (en este caso A). Así pues, el receptor obtendría el mensaje M a partir de:

$$M = D(C, K_{privA})$$

Una característica muy importante en estos sistemas es que el algoritmo debe asegurar la dificultad computacional de descubrir la llave privada K_{priv} a partir de la pública K_{pub}.

Estos sistemas pueden emplearse para establecer comunicaciones seguras cuando se usan canales de comunicación inseguros puesto que únicamente viaja por el canal inseguro la llave pública, que sólo sirve para cifrar, o para llevar a cabo autenticaciones. Sin la llave privada (que no es deducible a partir de la llave pública) un observador no autorizado del canal de comunicación será incapaz de descifrar el mensaje cifrado.

2.3.1. RSA

De entre todos los algoritmos de llave pública, RSA es el más usado y también quizás el más sencillo de entender e implementar. Una peculiaridad de este algoritmo es que sus dos llaves sirven indistintamente tanto para cifrar como para autenticar. Debe su nombre a sus tres inventores: Ronald Rivest, Adi Shamir y Leonard Adleman, que publicaron por primera vez el método RSA en 1977. Ha estado bajo patente de los Laboratorios RSA hasta el 20 de septiembre de 2000, por lo que su uso comercial estuvo restringido hasta esa fecha.

RSA, se basa en el problema de la factorización visto en la sección 2.1.3. Las llaves pública y privada se calculan a partir de un número que se obtiene como producto de dos primos grandes. Un atacante que quiera recuperar el texto en claro a partir del texto cifrado y de la llave pública, tiene que enfrentarse al ya mencionado problema de factorización.

Como todo esquema de llave pública, el algoritmo RSA se compone de tres subprocesos los cuales se describe a continuación [15]:

1. Generación del par de llaves

   - Para generar un par de llaves (K_{priv}, K_{pub}), en primer lugar se eligen aleatoriamente dos números primos grandes, p y q, para después calcular el producto $n = pq$. Para que la factorización de n sea computacionalmente muy difícil de obtener p y q deben cumplir con las siguientes restricciones [14]:

a) El tamaño de p y q debe ser aproximado a 512 bits (como mínimo).

b) La diferencia de p y q no debe ser muy pequeña, ya que si $p \approx q$ entonces $p \approx \sqrt{n}$ y por lo tanto sería computacionalmente fácil de calcular la factorización de n pues sólo se probaría con los números impares cercanos a $\sqrt{n}$.

c) Los números p y q deben ser primos fuertes. Se dice que un número primo p es llamado primo fuerte si cumple con las siguientes restricciones :

1) $p-1$ tiene un factor primo grande, al cual llamaremos r;
2) $p+1$ tiene un factor primo grande; y
3) $r-1$ tiene un factor primo grande;

- Posteriormente se escoge un número e tal que $1 < e < \phi(n)$ y que además sea primo relativo con $\phi(n)$. Este par de números (e, n) pueden ser conocidos por cualquiera, y constituyen la llamada llave pública K_{pub}
- Se calcula el inverso de e módulo $\phi(n)$, por lo que se debe cumplir que $ed \equiv 1(\mathrm{mod}\,\phi(n))$,así la llave privada será el par (d, n). Este número d debe mantenerse secreto y sólo será conocido por el propietario del par de llaves.

2. Cifrado del mensaje con la llave pública

   - Para obtener el mensaje cifrado C a partir del mensaje en claro M, se realiza $C = M^e(\mathrm{mod}\,n)$

3. Descifrado del mensaje con la llave privada

   - Para recuperar el mensaje original M a partir del cifrado se realiza $M = C^d(\mathrm{mod}\,n)$

2.3.2. ElGamal

Otro de los algoritmos de llave pública más usados es el llamado algoritmo de ElGamal, bautizado (de igual manera que RSA) en honor de su inventor Taher Elgamal, fue propuesto en 1985 y basa su seguridad en el problema del logaritmo discreto visto en la sección 2.1.4.

Los subprocesos que componen al algoritmo de ElGamal se describen a conti-nuación[22]:

1. Generación del par de llaves:

   - Se escoge un número primo p lo suficientemente grande (512 - 1024 bits), tal que $p = kq + 1$ en donde k es un número entero pequeño (comúnmente $(1 < k < 6)$) y q es un número primo grande.
   - Se escoge aleatoriamente una raíz primitiva g modulo p, como se vio en la sección 2.1.2
   - Se escoge un número aleatorio x tal que $0 < x < p$, donde x será la llave privada.
   - Se calcula $h = g^x(\mathrm{mod}\,p)$, donde (h, g, p) será la llave pública.

2. Cifrado del mensaje con la llave pública

   - Se obtiene la llave pública (h, g, p).
   - Se escoge un entero aleatorio k tal que $0 < k < p - 1$ y se calcula $c_1 = g^k (\mathrm{mod}\, p)$ y $c_2 = h^k m (\mathrm{mod}\, p)$.
   - Así el mensaje cifrado sería (c_1, c_2) .

3. Descifrado del mensaje con la llave privada

   - Se obtiene el mensaje a partir de $M = c_2 (c_1^x)^{-1}$ donde x es la llave privada.

2.4. Firmas digitales

La firma digital es una herramienta criptográfica que permite garantizar la autoría y la integridad de los documentos digitales [15], posibilitando que éstos gocen de una característica que únicamente era propia de los documentos en papel. Además de estas dos características la firma digital cuenta con una tercera, el no repudio en origen, es decir el firmante no puede negar que su firma es la que esta en dicho documento [15].

Los protocolos de firma digital combinan los algoritmos de hash con los algoritmos de llave pública para poder lograr su cometido.

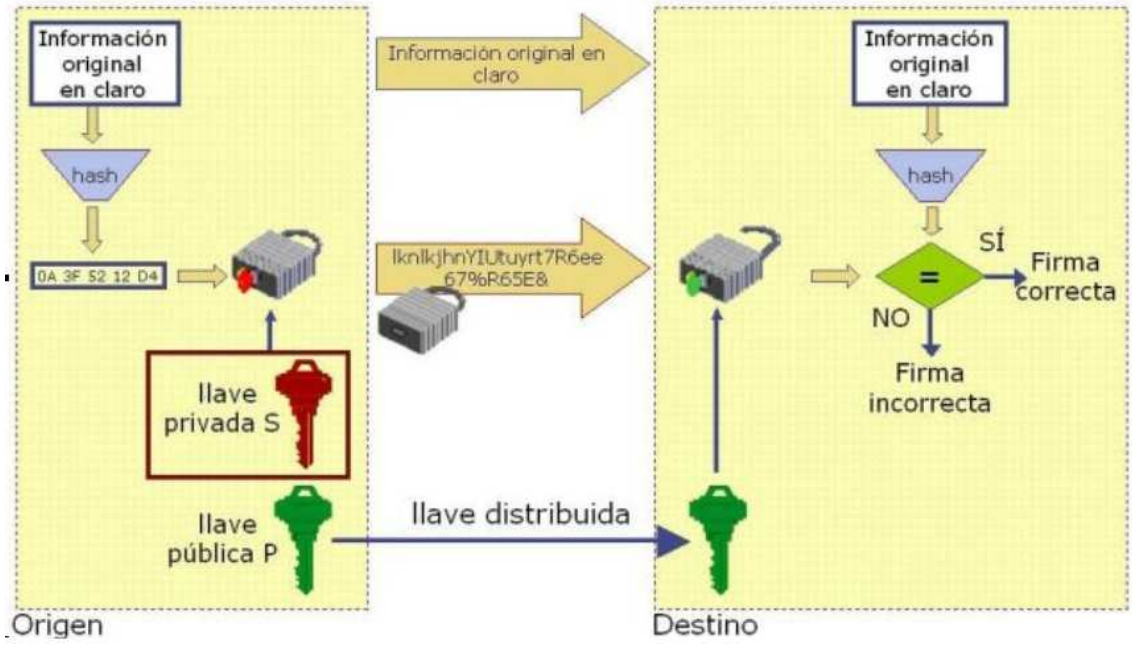

Figura 2.1: Proceso general de firma y verificación

En la figura 2.1 podemos ver el proceso de firma y verificación y a continuación se explican como funcionan de manera general los algoritmos de firma y verificación:

- Proceso de Firma:

    1. Se toma el mensaje M y se crea un hash del mensaje que se va a firmar, utilizando una cierta función hash $H(M)$.
    2. Usando la llave privada K_{priv} se cifra el hash generado en el paso anterior para así obtener la firma digital $F = C(H(M), K_{priv})$.
    3. Se envía la firma F conjuntamente al receptor conjuntamente con el mensaje original M (el cual puede ir cifrado o no, según se requiera).

- Proceso de verificación

    1. Se reciben el mensaje M y la firma F
    2. Utilizando la llave pública K_{pub} del remitente se descifra la firma recibida obteniendo el hash que envió el remitente $H(M) = D(F, K_{pub})$.
    3. Paralelamente, se calcula el hash del mensaje recibido $H'(M)$ utilizando la misma función hash que utilizó el remitente.
    4. Si ambos $H(M)$ (el hash descifrado) y $H'(M)$ (el hash calculado) coinciden, entonces queda verificada la firma y de esta forma puede asegurarse que el mensaje sólo ha podido ser originado por el remitente y que además el mensaje ha llegado íntegramente.

Evidentemente, la firma F no podrá ser manipulada por nadie una vez generada, porque si se cambia un sólo bit de la firma fallaría la verificación de ésta en el destino.

A continuación veremos los esquemas de firma digital que utilizados en los protocolo de dinero electrónico estudiados en esta tesis.

2.4.1. Firmas digitales usando RSA

RSA puede ser utilizado para generar firmas digitales de la siguiente manera[15]:
En primera instancia supondremos que las llave pública y privada del firmante fueron creadas de acuerdo a los lineamiento descritos en el subproceso de generación de llaves en la sección 2.3.1. Así pues tendremos la llave pública $K_{pubA} = (e_A, n_A)$ y la llave privada $K_{privA} = (d_A, n_A)$ del firmante A. Ahora bien los procesos de firma y verificación para RSA se describen a continuación[15]:

- Proceso de firma digital

    1. A un mensaje M se le aplica una función hash definida para asi obtener su digesto $H(M)$.

2. Con la llave privada el firmante calcula la firma $F = (H(M))^{d_A} (\mathrm{mod}\, n_A)$
3. Se envian los datos (F, M) al receptor, quien verificará la firma.

- Proceso de verificación de la firma digital

1. Se obtiene la llave pública del firmante (e_A, n_A)
2. Al mensaje recibido M se le aplica una función hash para así obtener su digesto $H'(M)$.
3. El receptor calcula $H(M) = (F)^{e_A} (\mathrm{mod}\, n_A)$
4. Si el hash descifrado a partir de la firma $H(M)$ es igual al hash calculado del mensaje recibido $H'(M)$ se verifica que el mensaje fue firmado correctamente y se acepta el mensaje, de lo contrario se rechaza el mensaje pues la firma no verifica.

2.4.2. Firmas digitales usando ElGamal

ElGamal cuenta también con un esquema de firma digital[22]. Para realizar una firma digital con elGamal se realiza primero la selección de las llaves pública y privada del firmante, este proceso se lleva a cabo de la misma manera que el subproceso de selección de llaves descrito en la sección 2.3.2. Así pues tendremos la llave pública $K_{pubA} = (h_A, g_A, p_A)$ y la llave privada $K_{privA} = (x_A)$ del firmante A. Ahora bien los procesos de firma y verificación para elGamal se describen a continuación[22]:

- Proceso de Firma

1. A un mensaje M se le aplica una función hash definida para así obtener su digesto $H(M)$.
2. Se escoge aleatoriamente un número k tal que $0 < k < \phi(p_A)$ y $mcd(k, \phi(p_A)) = 1$.
3. Se calcula $r \equiv g^k (\mathrm{mod}\, p_A)$
4. Se calcula $s \equiv (H(M) - x_A r)k^{-1} (\mathrm{mod}\, p_A - 1)$. Si $s = 0$ entonces se repite el proceso desde 2.
5. Así la firma estará conformada por (r, s) y le será enviada al receptor junto con el mensaje M.

- Proceso de verificación de firma.

1. Se obtiene la llave pública del firmante (h_A, g_A, p_A).
2. Al mensaje recibido M se le aplica una función hash para asi obtener su digesto $H'(M)$.

3. Se verifica que $0 < r < p$ y $0 < s < p - 1$, hecho lo anterior se verifica que $g_A^{H'(M)} \equiv h^r r^s (\mathrm{mod}\, p_A)$
4. Si se cumple la congruencia anterior se acepta la firma y el mensaje, de lo contrario se rechaza el mensaje pues la firma no verifica.

2.4.3. Firmas digitales usando el esquema de Schnorr

Este esquema está basado en el problema de logaritmo discreto descrito en la sección 2.1.4, fue propuesto en 1991 por Schnorr en [5] y está compuesto por tres subprocesos, los cuales son:

- Proceso de inicialización: Se toma como entrada un parámetro de seguridad n, con el cual se escogerá un número primo q de manera aleatoria, de tal forma que la cantidad de bits de $|q| = n$, y que exista un un número primo p tal que $p = kq + 1$ para algún $k > 1$. También se escoge una raíz primitiva g módulo q. Por último una función hash H, la cual se asume que se comporta como un oráculo aleatorio.

  La entidad que firma, debe obtener un número aleatorio $u \in Zq$, el valor de u es considerado la llave secreta $K_{Priv} = (u)$, después se calcula $I \equiv g^u (\mathrm{mod}\, p)$, y se conforma la llave pública $K_{Pub} = (I, p, q, g)$.

- Proceso de firma:

  1. Se obtiene la llave privada (p, q, g, u) y el mensaje M .
  2. Se calcula $y \equiv g^x (\mathrm{mod}\ p)$, en donde x es un número aleatorio tal que $1 < x < q-1$
  3. Se obtiene $e \equiv H(m|I|y)$ con $|\ \mathrm{e}\ | > 1$, y
  4. Se calcula $r \equiv eu + m (\mathrm{mod}\, p)$
  5. La firma del mensaje M queda conformada por (y, e, r) y ambos son enviados al receptor

- Proceso de verificación

  1. Se obtiene la llave pública del firmante (p, q, g, I)
  2. Se recibe el mensaje M junto con la firma (y, e, r) y se procede a verificar que la congruencia $g^r \equiv I^e y (\mathrm{mod}\, p)$ y que $e = H(m|I|y)$
  3. Si estas dos comprobaciones son verdaderas entonces se acepta la firma y el mensaje, de lo contrario se rechaza el mensaje pues la firma no verifica.

En el esquema de firma Schnorr se requiere sólo una exponenciación módulo p y una multiplicación módulo q. Dependiendo del algoritmo de hash usado, el tiempo del cálculo de $H(m|I|y)$ es relativamente pequeño. Para el proceso de verificación se requieren dos exponeciaciones módulo p. Por lo tanto, como se puede apreciar, el esquema de firmas Schnorr no poveé de una gran ganancia computacional con respecto al esquema de firmas ElGamal, sin embargo, si proveé de un tamaño de firma mucho más pequeño con respecto a firmas generadas por el equema de elGamal [14].

2.5. Firmas a ciegas

El concepto de firma a ciegas fue introducido en 1982 por David Chaum en[2]. Las firmas a ciegas son firmas digitales que permiten firmar un documento sin revelar su contenido[15]. Para que la idea quede más clara, podemos decir que es el efecto similar a poner una hoja de papel carbón en el documento y meterlo en un sobre. Firmando el sobre, el documento que se encuentra dentro queda firmado. Así, aunque eliminemos el sobre, la firma del documento permanecerá intacta.

Un esquema general de firma a ciegas funcionaría de la siguiente manera[15]:

1. La entidad A toma un documento y lo multiplica por un valor aleatorio. Dicho valor es conocido como el factor de ocultamiento.
2. La entidad A envía el documento opacado por el factor de ocultamiento a la entidad B
3. La entidad B recibe y firma el documento opacado, para después enviarlo de vuelta a la entidad A
4. La entidad A recibe el documento opacado y firmado, entonces lo divide por el factor de ocultamiento y de esta manera se obtendrá únicamente el documento firmado por la entidad B.

Obviamente el protocolo sólo funcionará si la función de firma y la multiplicación son conmutativas.

Existe otra forma de firmar a ciegas y se le conoce como firma casi a ciegas, la cual funciona de manera similar a la firma ciegas, sólo que en este caso se cifran n documentos y se entregan a la entidad que va a firmarlos. Luego la entidad solicita que le sean entregadas las llaves para que se descifren $n-1$ documentos al azar, de forma tal que el contenido de estos documentos queden revelados y la entidad que va a emitir la firma pueda corroborar que los $n-1$ documentos contienen los datos necesarios para que sean firmados y con una probabilidad muy alta, podrá inducir que los documentos restantes también contendrán datos válidos, así que toma 1 de esos documentos restantes y lo firma a ciegas. Evidentemente, cuanto más grande sea n, la probabilidad de engañar a la entidad que firma a ciegas es menor.

Las firmas a ciegas tiene diversas aplicaciones tal es el caso de los sistemas de control de acceso anónimo, el voto electrónico y el dinero electrónico.

2.5.1. Firmas a ciegas con RSA

Con RSA es posible realizar firmas a ciegas de la siguiente manera:

Supongamos que la entidad B cuenta ya con su conjunto de llaves la pública (e_B, n_B) y la privada (d_B, n_B) y que la entidad A desea que su documento M sea firmado a ciegas por la entidad B. Para ello se realiza el siguiente procedimiento:

1. La entidad A escoge el factor de ocultamiento el cual es un número k tal que mcd(k,n)= 1. Y aplica dicho factor al mensaje de la siguiente manera:

   $t = Mk^{e_A} (\mathrm{mod}\, n_A)$

2. La entidad A envía el documento opacado a la entidad B quien simplemente al recibirlo lo firma con

   $t^{d_A} = (Mk^{e_A})^{d_A} (\mathrm{mod}\, n_A)$

3. La entidad A recibe el mensaje firmado y simplemente le quita el factor de ocultamiento de la siguiente manera:

   $s = (t^{d_A})k^{-1} (\mathrm{mod}\, n_A)$

4. Finalmente el resultado obtenido es el mensaje firmado por B

   $s = m^{d_A} (\mathrm{mod}\, n_A)$

Capítulo 3

Protocolos de DE

Como se mostró en el capítulo 2 existen hoy en día diversos protocolos de dinero electrónico. Dado que el objetivo de esta tesis es lograr la implementación de un sistema de dinero electrónico en dispositivos móviles, fueron seleccionados aquellos protocolos que de acuerdo a la literatura mostraron el mejor desempeño hace 10 años o más ya que el poder de cómputo de los dispositivos móviles de la actualidad puede compararse con las computadoras de aquel entonces, con base a esto, después se llevo a cabo una comparativa de las prestaciones y eficiencia de cada uno de los protocolos seleccionados, esta comparativa entre los protocolos seleccionados puede verse en la sección 3.1.

En base a dicha comparativa se eligieron 2 protocolos de dinero electrónico, el propuesto por S.Brands [9] el cual utiliza el esquema Básico de DE y el propuesto por Tsiounis et al [12] el cual es una modificación al protocolo de S. Brands para que utiliza el modelo FOLC, estos protocolos son descritos en las secciones 3.2 y 3.3 respectivamente. Después en la sección 3.3 se realizó una variante al esquema propuesto por Tsiounis utilizando otro sistema de llave pública para realizar una comparación entre ambos. Finalmente en la sección 3.5 se discute el problema de divisibilidad, ya que esta es una propiedad de la que carecen los protocolos seleccionados. Ahí se mencionan las posibles soluciones para resolver este problema.

3.1. Revisión de los protocolos estudiados

En esta sección se llevo a cabo una comparativa entre algunos de los protocolos de DE, en la tabla 3.1 se muestran los 5 protocolos seleccionados así como los fundamentos criptográficos en los que recae la seguridad de cada uno de ellos:

La primera comparativa realizada fue determinada por las prestaciones que ofrecen, es decir cuáles de las propiedades deseables del DE vistas en la sección 2.4 cumple cada uno de los protocolos seleccionados. El resultado de esta comparativa se puede apreciar en la tabla 3.2. En donde como puede apreciarse la tabla esta ordenada de acuerdo a aquellos protocolos que cumplen con el mayor número de propiedades deseables del DE.

Una segunda comparativa fue realizada en base al tipo de rastreabilidad que puede ser llevada a cabo usando cada uno de los protocolos seleccionados, además se muestra bajo cual de los dos esquemas (descritos en la sección 1.6) trabaja cada uno de los protocolos revisados. En la tabla 3.3 podemos ver el resultado de la comparativa que se realizó.

Protocolo	Fundamentos Criptográficos	Publicado
Chaum-Fiat-Naor [3]	Firmas a ciegas con RSA, Funciones Hash	1988
Okamoto-Otha [4]	Múltiples firmas a ciegas con RSA, Funciones Hash	1991
Ferguson [10]	Firmas aleatorias a ciegas con RSA, Funciones Hash	1993
Brands [9]	Problema del Logaritmo Discreto, Prop. Homomorfas de los Logaritmos	1993
Tsiounis et al. [12]	Problema del Logaritmo Discreto, ElGamal	1997
Ham [21]	Problema del Logaritmo Discreto, Funciones Hash	2002

Cuadro 3.1: Protocolos seleccionados

Protocolo	P_1	P_2	P_3	P_4	P_5	P_6
Okamoto-Otha [4]	Si	Si	Si	Si	Si	Si
Chaum-Fiat-Naor [3]	Si	Si	Si	Si	No	No
Ferguson [10]	Si	Si	Si	Si	No	No
Brands [9]	Si	Si	Si	Si	No	No
Tsiounis et al. [12]	Si	Si	Si	Si	No	No
Ham [21]	Si	Si	Si	No	No	No
P_1(Independencia), P_2(Seguridad), P_3(Privacidad), P_4(Pago fuera de línea), P_5(Transferibilidad), P_6(Divisibilidad)						

Cuadro 3.2: Propiedades que cumplen los protocolos de DE seleccionados

Protocolo	Rastreabilidad	Esquema
Chaum-Fiat-Naor [3]	Ninguna	Básico
Okamoto-Otha [4]	Directa	Básico
Ferguson [10]	Directa	Básico
Brands [9]	Ninguna	Básico
Tsiounis et al. [12]	Condicional	FOLC
Ham [21]	Directa	Básico

Cuadro 3.3: Tipo de rastreabilidad y esquema de los protocolos seleccionados

Finalmente fue revisado el desempeño de cada uno de los protocolos en base al número de exponenciaciones e inversos que son realizdos por parte de la entidad *comprador* en cada uno de los procesos de cada uno de los protocolos, así mismo se especifica el tamaño de la

Protocolo	Retiro	Pago	Depósito	Tamaño de llave
Chaum-Fiat-Naor [3]	k exp.+ k/2 inv.	0	0	1024 bits
Okamoto-Otha [4]	03 exp + 1 inv	16 exp.	0	1024 bits
Ferguson [10]	06 exp.+ 3 inv.	05 exp. + 1 inv.	0	1024 bits
Brands [9]	10 exp.+ 1 inv.	0	0	512 bits
Tsiounis et al. [12]	12 exp.+ 1 inv.	2 exp.	0	512 bits
Ham [21]	2 exp.	2 inv.	1 exp.	512 bits

Cuadro 3.4: Comparativa en cuanto al computo necesario para la creación de una moneda electrónica

llave que debe tener el sistema para que se mantenga seguro. En la tabla 3.4 se muestra los resultados realizados en dicho análisis.

Con este pequeño análisis practicado a este conjunto de protocolos de DE se decidió que los protocolos a implementar fueran los propuestos por Brands [9] y Tsiounis et al [12]. Justificando esta elección con las siguientes observaciones:

1. Son los más eficientes del lado del *comprador*.
2. Debido a que su seguridad esta basada en números de 512 bits su almacenamiento por lo tanto también es el más optimo para nuestros fines.
3. Al implementar estos protocolos se podrá realizar una comparación de los dos esquemas de DE vistos en la sección 1.6

3.2. El protocolo de S. Brands

En esta sección será descrito el protocolo propuesto por S. Brands [9]. Para ello serán descritos en las siguientes subsecciones los cinco procesos de los que se compone este protocolo (inicialización, retiro, pago/compra, depósito y control de fraudes)

3.2.1. Proceso de inicialización

Primero que nada el *banco* realiza un procedimiento de inicialización, el cual se ejecuta una sola vez y consiste en que el *banco* escoge dos primos p y q tal que $q = (p-1)/2$. Sea g el cuadrado de una raíz primitiva módulo p. Entonces se escogen dos números aleatorios a los cuales se elevará g mod p, obteniendo:

$g_1 \equiv g^{x_1}$, $g_2 \equiv g^{x_2} \pmod{p}$, donde x_1, $x_2 \in Z_q$

Una vez obtenidos g, g_1 y g_2 serán publicados de la misma manera que dos funciones hash $H(v_1, v_2, v_3, v_4, v_5)$ y $H_0(v_1, v_2, v_3, v_4)$ como se puede apreciar la primera es una función hash de 5 valores y la segunda es una función hash de 4 valores la definición exacta de cada una de estas se puede ver en [9], sin embarigo en esta tesis se utilizó el algoritmo de hash MD5 para la realización de cada una de las funciones hash involucradas. Por último se escoge un número aleatorio X_B el cual será la clave secreta de la entidad *banco* y con el cual se crearán h, h_1 y h_2 los cuales son publicados y serán los que identificarán al *banco*:

$h \equiv g^{X_B}$, $h_1 \equiv g_1^{X_B}$ y $h_2 \equiv g_2^{X_B} \pmod{p}$

Por otro lado, cada entidad *comprador* escoge aleatoriamente un número secreto $u_1 \in Z_q$ y calcula lo que será el identificador de su número de cuenta I:

$I \equiv g_1^{u_1} \pmod{p}$

Una vez que el *comprador* es autenticado ante el *banco*, el número I es enviado al *banco* junto con la información correspondiente del usuario (nombre, dirección, etc.). Finalmente el *comprador* calcula y almacena z':

$z' \equiv h_1^{u_1} h_2 \equiv (I g_2)^{X_B} \pmod{p}$

Por último el banco registra a cada una de las entidades *vendedor* con un número de identificación $M \in Z_q$, y una vez concluido este proceso de inicialización las entidades *banco*, *comprador* y *vendedor* están listas para comenzar a interactuar entre sí de acuerdo a los siguientes procesos:

3.2.2. Proceso de retiro de fondos

El objetivo de este protocolo es que el comprador obtenga una moneda válida creada por el *banco*, siempre y cuando el *banco* esté seguro de conocer la identidad de quien solicita la moneda. Al final la moneda estará representada por una sextupla que será formada por los valores de A, B, z, a, b, r. Para lograr esto se sigue el siguiente orden.

1. *Solicitud de monedas.* El comprador solicita una moneda al *banco* identificándose con su número I, el *banco* verifica que exista en la base de datos una cuenta con dicho número, de igual forma que tenga el crédito suficiente para efectuar el retiro de la moneda solicitada. Si todo va bien se continúa de lo contrario el proceso se detiene. En este paso se envía el mensaje etiquetado como M1, el cual contiene el valor de I y el valor del monto del retiro, tal como se muestra en la figura 3.1, este mensaje será implementado en el capítulo 4 con el nombre de *RequestRetiro*.

2. *Definición del identificador de la moneda.* El *banco* escoge un número aleatorio $w \in Z_q$ con el cual se identificará a la moneda, podemos decir que será el número de serie asignado a la moneda, así el *banco* calcula a' y b' y se los envía al *comprador* en el

comprador		*banco*
1. Solicitud de la moneda	I, Retiro	
1.1	(M1) $\Longrightarrow$	
2. Identificador de la moneda	a', b'	$w \in_R Z_q$
2.1	(M2) $\Longleftarrow$	$a' = g^w$ y $b' = (A_1')^w$
3. Creación de la moneda		
3.1 $A = (Ig_2)^s$		
3.2 $z' = h_1^{u_1} h_2 h_3^{s^{-1}}, z = z'^s$		
3.3 $x_1, x_2, u, v \in_R Zq$		
3.4 $B_1 = g_1^{x_1}, B_2 = g_2^{x_2}$		
3.5 $B = [B_1, B_2]$		
3.6 $a = (a')^u g^v$		
3.7 $b = (b')^{su} A^v$		
3.8 $c = H(A\|B\|z\|a\|b)$	c'	
3.9 $c' = c/u$	(M3) $\Longrightarrow$	
4. Firma de la moneda	r'	
4.1	(M4) $\Longleftarrow$	$r'^{=} c' X_B + w$
5. Composición de la moneda		
5.1 $r = r'u + v \bmod q$		
5.2 Verificar:		
$g^r \equiv h^{c'} a'$		
$(Ig_2)^{r'} \equiv z'^{c'} b'$		

Figura 3.1: Proceso de retiro para el protocolo de Brands

mensaje M2. Tal como se muestra en la figura 3.1. En la implementación realizada en el capítulo 4, los valores del mensaje M2 son encapsulados en un mensaje llamado *Response01*.

3. *Inicio de creación de la moneda*. El comprador escoge cinco números aleatorios y calcula A, B, a, b, utilizando los números aleatorios escogidos y los números recibidos a' y b', tal y como se muestra en los pasos del 3.1 al 3.7 descritos en la figura 3.1. Una vez hecho esto en los pasos 3.8 y 3.9 se calcula y envía c' al *banco* en el mensaje número M3, tal como se muestra en la figura 3.1. En la implementación realizada en el capítulo 4, el contenido del mensjae 3 corresponde al mensaje llamado *Request02*.

4. *Firma de la moneda*. El *banco* recibe c' y utilizando el valor de w (identificador de la moneda) y el valor de X_B (clave secreta del *banco*) el *banco* calcula r' (firma de la moneda) para enviarle ésta al *comprador* en el mensaje M4, tal como lo muestra la

figura 3.1. En la implementación realizada en el capítulo 4, este valor corresponde al mensaje llamado *Response02*.

5. *Composición de la moneda.* El *comprador* recibe finalmente el valor de r' y crea con éste el valor de r como se muestra en la figura 3.1 en el paso 5.1, se realiza 2 comprobaciónes para verificar que la firma sea válida y finalmente, si las congruencias son correctas el comprador establece la moneda como la sextupla conformada por los valores (A, B, z, a, b, r), de lo contrario se vuelve a realizar el proceso.

3.2.3. Proceso de pago/compra

En este proceso la entidad *comprador* intenta utilizar una moneda para el pago de un producto o servicio a una entidad *vendedor*, para esto, el protocolo consta de dos fases: la primera en la que se verifica que la moneda sea válida y la segunda fase que consiste en recabar la información necesaria para que la entidad *banco* más adelante tenga los datos suficientes para descubrir la identidad de quien intente cometer un fraude tratando de gastar una misma moneda dos o más veces. Para lograr esto se realizan los siguientes pasos:

1. *Establecimiento de la compra.* Para comenzar este protocolo el *vendedor* y el *comprador* se ponen de acuerdo con respecto al producto y a la cantidad de monedas que se necesitaran para efectuar la compra. En este paso se envía un mensaje etiquetado como M1, en el que se estableces los valores del identificador del pedido que se va a pagar y el monto a pagar, tal y como se muestra en la figura 3.2. En la implementación realizada en el capítulo 4 el mensaje número 1 de este proceso es llamado *RequestPago*.

2. *Envio de la moneda.* En este punto el *comprador* le envía (en el mensaje M2) la moneda conformada por la sextupla A, B, z, a, b, r junto con los valores A_1 y A_2 al *vendedor*, el cálculo de los valores A_1 y A_2 solo podran ser creados por el verdadero propietario de la moneda ya que para crear dichos valores se requiere el conocimiento del valor de s el cual fue escogido aleatoriamente por el *comprador* en el proceso de retrio, por lo tanto es el único que conoce su valor. En la figura 3.2 se detalla la construcción de A_1 y A_2 en el paso 2.1. En la implementación realizada en el capítulo 4 el mensaje M2 de este proceso es llamado *Request11*.

3. *Validación de la Moneda.* El vendedor establece la validez de la moneda verificando que la firma de la moneda sea correcta y que el propietario de la moneda sea el que está enviandola. Para ello calcula dos valores y verifica que las congruencias entre los valores calculados y los valores de la moneda concuerden, con dichas congruencias se demuestra en [9] que el vendedor puede aceptar o rechazar la moneda con toda la certeza de que la moneda que está aceptando es válida y la que está rechazando es inválida.

4. *Hash para el control de fraudes.* Si todo va bien el vendedor calcula d, el cual, no es más que la evaluación de la función hash $H_0(A, B, M, t)$ donde M es el identificador del vendedor establecido en el proceso de inicialización de la entidad *vendedor* y t es una marca de tiempo. Una vez calculado d se le envía al *comprador* en el mensaje M3

de este proceso, tal y como se muestra en la figura 3.2. En la implementación realizada en el capítulo 4 el mensaje M3 de este proceso es llamado *Response11*.

5. *Datos para el control de fraudes*. Finalmente el comprador calcula r_1 y r_2 tal y como se muestra en los pasos 5.1 y 5.2 de la figura 3.2, estos datos le servirán al *banco* para obtener (en el proceso de control de fraudes) la identidad de quien intente realizar un doble gasto. Una vez caluculados r_1 y r_2 son enviados a la entidad *vendedor* en el mensaje M4. En la implementación realizada en el capítulo 4 el mensaje M4 de este proceso es llamado *Request21*.

6. *Aceptación o rechazo de la moneda*. Finalmente con los datos r_1 y r_2 recibidos del *comprador* el *vendedor* realiza dos ultimas comprobaciones, las cuales se describen en los pasos 6.1 y 6.2 de la figura 3.2, con éstas el *vendedor* decide si acepta o rechaza la moneda.

comprador		*vendedor*
1. Establecimiento de compra	IdPedido, Monto	
1.1	(M1)$\Longleftrightarrow$	
2. Envio de Moneda		
2.1 $A_1 = I^s, A_2 = g_2^s$		
2.2	$A_1, A_2, A, B, (z, a, b, r)$	
	(M2) $\Longrightarrow$	
3. Verificación de Moneda		
3.1		Verifica que:
		$A = A_1 A_2,\ A \neq 1$
		$\text{Firma}(A, B) = (z, a, b, r)$
4. Envio de Hash	d	
4.1	(M3) $\Longleftarrow$	$d = H(A_1 \vert B_1 \vert A_2 \vert B_2 \vert M \vert t)$
5. Para el control de fraudes		
5.1 $r_1 = d(u_1 s) + x_1$	r_1, r_2	
5.2 $r_2 = ds + x_2$	(M4) $\Longrightarrow$	
6. Aceptación de la moneda		Verifica que:
6.1		$g_1^{r_1} = A_1^d B_1$
6.2		$g_2^{r_2} = A_2^d B_2$

Figura 3.2: Proceso de pago/compra para el protocolo de Brands

3.2.4. Proceso de déposito/cobro

En este protocolo el vendedor intenta cobrar las monedas obtenidas de los compradores, haciendo que el *banco* le abone a su cuenta dichas monedas. Para ello el se realizan los siguientes pasos:

1. *Reenvio de datos.* Primero, los datos que el vendedor obtuvo en el proceso de pago/compra son reenviados al *banco* en el mensaje M1 de este proceso, el cual se describe en la figura 3.3. Es decir los datos de la moneda $(A_1, A_2, A, B, z, a, b, r)$ y los datos para el control de fraudes (r_1, r_2 y d). En la implementación realizada en el capítulo 4 todos estos datos son encapsulados en el mensaje llamado *Request21.*

2. *Verificación de datos.* Una vez recibidos los datos el *banco* verifica, como se puede ver en la figura 3.3, en los pasos 2.1 y 2.2 la construcción correcta de la moneda y en los pasos 2.3 y 2.4 la contrucción correcta de los valores para el control de fraudes, si las comprobaciones realizadas son correctas el *banco* abona el valor de la moneda a la cuenta del *vendedor*. Por útlimo, el *banco*, intenta almacenar la moneda en su base de datos siempre y cuando dicha moneda no se encuentre ya almacenada; si el *banco* detecta que la moneda ya había sido gastada anteriormente destina la moneda recibida al proceso de control de fraudes.

vendedor		*banco*
1. Reenvio de datos		
1.1	$A_1, A_2, A, B, (z, a, b, r), d, r_1, r_2$	
	(M1) $\Longrightarrow$	
2. Verificación de datos		Verifica que:
2.1		$A = A_1 A_2,\ A \neq 1$
2.2		$\text{Firma}(A, B) = (z, a, b, r)$
2.3		$g_1^{r_1} = A_1^d B_1$
2.4		$g_2^{r_2} = A_2^d B_2$

Figura 3.3: Proceso de depósito para el protocolo de Brands

3.2.5. Proceso de control de fraudes

Existen varias formas de cometer fraude en los protocolos de dinero electrónico la más importante y la que haremos mención en este caso es la reutilización de una moneda electrónica. Para evitar este tipo de fraude el protocolo de S. Brands determina un mecanismo para conocer la identidad de quien intente gastar una moneda dos o más veces. Para esto supongamos que el *vendedor*1 le envía al *banco* la tripleta (r_1, r_2, d) en el mensaje número 1 en el el proceso de depósito detallado en la figura 3.3 y un *vendedor*2 le envía de igual forma, al *banco* la tripleta (r_1', r_2', d'), ambos envíos son producidos por el gasto de una misma moneda (A, B, z, a, b, r) con un diferente vendedor.

Así pues, el banco con un simple cálculo, obtendría el valor de u_1 a continuación se muestra dicho proceso [22]:

Dado que:

$r_1 - r_1' \equiv u_1 s(d - d')$ y $r_2 - r_2' \equiv s(d - d') \bmod q$

entonces:

$u_1 \equiv (r_1 - r_1')/(r_2 - r_2') \bmod q$

por lo tanto el banco podría calcular:

$I \equiv g_1^{u_1} \bmod p$

y de esta manera identificar al tramposo.

3.3. Protocolo de Yung-Frankel-Tsiounis

Como se ha dicho anteriormente este protocolo es una modificación al protocolo de S. Brands[9], como tal, solo explicaremos las modificaciones o agregados que se hicieron al protocolo base.

Para empezar recordemos que este protocolo trabaja bajo el modelo FOLC (ver sección 1.6.2) por lo que en primer lugar se debe tener presente que se cuenta con una nueva entidad a la que llamaremos la *autoridad*.

3.3.1. Proceso de inicialización

La *autoridad*: Escoge aleatoriamente una llave privada $X_T \in Zq$ y se crea una llave pública la cual esta dada por: $f_2 \equiv g_2^{X_T} \bmod p$.

El *banco*: realiza el mismo proceso de inicialización que se estableció en el protocolo de S.Brands (sección 3.2) agregando a éste cálculo un nuevo valor de g3, el cual es calculado de la misma forma la que se calcularon g_1 y g_2 en el proceso de inicialización del protocolo de S. Brands, es decir:

$g_1 \equiv g^{x_1}$, $g_2 \equiv g^{x_2}$ y $g_3 \equiv g^{x_3} \pmod p$, donde x_1, x_2, $x_3 \in Z_q$

De esta manera ahora contamos con g, g_1, g_2 y g_3 los cuales son publicados de la misma manera que tres funciones hash $H(v_1, v_2, v_3, v_4, v_5, v_6)$, $H_0(v_1, v_2, v_3, v_4, v_5)$ y $H_1(v_1)$. Cada una de estas funciones esta definida utilizando el algoritmo de hash MD5. Por último se escoge un número aleatorio $X_B \in Z_q$ con el cual se crearán h, h_1, h_2 y h_3 de la misma manera que en protocolo de S. Brands y los cuales son publicados y serán los que identificarán al *banco*.

El *comprador* y el *vendedor*: realizan exactamente el mismo proceso que en el protocolo de S. Brands visto en la sección 3.2.

Con esto termina el proceso de inicialización para este protocolo. Ahora veremos como fueron alterados los demás procesos para lograr que la entidad *autoridad* pueda rastrear ya sea al propietario de una moneda o a la moneda misma.

3.3.2. Proceso de Retiro modificado

Para lograr ahora el rastreo de monedas se realiza una modificación al proceso de retiro de la sección 3.3.2 y ésta reside en la construcción del valor de A en la moneda (paso 5.1), el cual, en lugar de realizarse con 2 generadores se realiza con 3 (g_1, g_2 y g_3) en donde los primeros 2 son usados para la verificación de la moneda y para el rastreo del propietario respectivamente y el tercero es usado para el rastreo de la moneda. Las modificaciónes al proceso de retiro propuesto por S. Brands pueden verse resaltdas en la figura 3.4.

Como se puede apreciar en la figura 3.4 la primera modificación en este protocolo está en el paso número 2, en donde se crean dos valores A'_1 y A'_2, el primero servirá para la generación posterior de A y el segundo será el que le permita a la entidad *autoridad* realizar el proceso de rastreo de moneda. Estos valores son enviados en el mensaje M1', el cual, en el capítulo 4 corresponde al mensaje llamado Request01.

Ahora bien, para que el rastreo de moneda funcione el *banco* debe asegurarse que realmente los datos A'_1 y A'_2 vayan a ser utilizados para la creación de la moneda, para ello el *banco* realiza un subproceso extra el cual es llamado "prueba de igualdad de logaritmos"[16], este subproceso puede verse en la figura 3.5 y es llevado a cabo de manera paralela junto con el paso 2.

Con esto también se modificó la creación de z y z' (paso 5.2) como se puede ver en la figura 3.4. Finalmente la verificación para la comprobación de la moneda ahora debe incluir el valor de g_3, tal y como se ve en el paso 7.2 de la figura 3.4.

3.3.3. Proceso de pago/compra modificado

La idea para lograr el rastreo de propietario es simple y se logra modificando el proceso de pago/compra. Debido a que cada una de las monedas lleva consigo de manera empotrada la identidad del propietario (I), lo único que se agrega al proceso de pago/compra básico propuesto por S. Brands es un cifrado de la identidad del usuario utilizando el esquema de llave pública de ElGamal, tal y como se ve en los pasos 2.2 al 2.4 de la figura 3.6, de tal manera que dicha cifra también sea ligada a la moneda. Para ello, se realiza una prueba indirecta de discurso (indirect discourse proof) [16] durante el protocolo de pago/compra, dicha prueba se realiza en los pasos 5.2 al 5.4, 5.3 y 6.3, de la figura 3.6. De tal manera que el *vendedor* pueda estar seguro de que la identidad cifrada sea la misma que la empotrada en la moneda, así el protocolo de pago modificado se ve en la figura 3.6.

comprador		*banco*
1. Solicitud de la moneda	I, Retiro	
1.1	(M1) $\Longrightarrow$	
2. Datos para el rastreo de moneda		
2.1 $s \in_R Z_q$	A'_1, A'_2	
2.2 $A'_1 = Ig_2 g_3^{s^{-1}}, A'_2 = f_2^s$	**(M1')** $\Longrightarrow$	
3. Prueba de igualdad de logaritmos		
3.1 Probar: $log_{A'_1/(Ig_2)} g_3 = log_{f_2} A'_2$		
4. Identificador de la moneda	a', b'	$w \in_R Z_q$
4.1	(M2) $\Longleftarrow$	$a' = g^w$ y $b' = (A'_1)^w$
5. Creación de la moneda		
5.1 $A = (A'_1)^s$		
5.2 $z' = h_1^{u_1} h_2 h_3^{s^{-1}}, z = z'^s$		
5.3 $x_1, x_2, u, v \in_R Zq$		
5.4 $B_1 = g_1^{x_1}, B_2 = g_2^{x_2}$		
5.5 $B = [B_1, B_2]$		
5.6 $a = (a')^u g^v$		
5.7 $b = (b')^{su} A^v$		
5.8 $c = H(A\|B\|z\|a\|b)$	c'	
5.9 $c' = c/u$	(M3) $\Longrightarrow$	
6. Firma de la moneda	r'	
6.1	(M4) $\Longleftarrow$	$r'^{=} c' X_B + w$
7. Composición de la moneda		
7.1 $r = r'u + v \bmod q$		
7.2 Verificar: $g^r \equiv h^{c'} a', (Ig_2 g_3)^{r'} \equiv z'^{c'} b'$		

Figura 3.4: Proceso de retiro modificado para permitir el rastreo de monedas

Estos son los unicos procesos que se ven modificados significativamente para permitir el rastreo de monedas y el rastreo de propietario, ya que, el proceso de deposito se realizaría de manera analoga al propuesto por S. Brands, en donde el *vendedor* reenvía los datos recibidos en el proceso de pago/compra hacia la entidad *banco* para que ésta realice la mismas comprobaciones que debió haber realizado el *vendedor* para aceptar la moneda. Con estas modificaciones a los procesos básicos, en una conjunción entre la entidad *banco* y la entidad *autoridad* se lleva a cabo los rastreos solicitados de la siguiente manera:

comprador		*banco*
Probar que: $log_{A'_1/(Ig_2)} g_3 = log_{f_2} A'_2$		
$y \in_R Z_q$ $\alpha = (A'_1/(Ig_2))^y, \beta = (f_2)^y$ $\gamma = H(g_3 \vert \alpha \vert (A'_1/(Ig_2)) \vert A'_2 \vert \beta \vert f_2)$ $\rho = \gamma s + y$	$\alpha, \beta, \gamma, \rho$ $\Longrightarrow$	
		Verifica que: $(A'_1/(Ig_2))^\rho = g_3^\gamma \alpha$ $(f'_2)^\rho = (A'_2)^\gamma \beta$

Figura 3.5: Supproceso para le prueba de la igualdad de logaritmos

- *Rastreo de prpietario.* Para lograr el rastreo del propietario de la moneda, el *banco* tendría que enviar los datos D_1 y D_2 a la *autoridad* (este mensaje es llamado *Request31* en el capítulo 4) para que con la llave privada de ésta calcule $I = D_1/(D_2)^{X_T}$, como se sabía de antemano I es el número de identificación asociado a la cuenta del *comprador* en la base de datos de la entidad *banco*, por lo que, la *autoridad* solo tendría que devolverle dicho valor al *banco* para que éste obtenga los datos correspondientes del *comprador*. Para esto se confía en la comprobación que debió de haber realizado el *vendedor*.

- *Rastreo de moneda.* Para llevar a cabo el rastreo de la moneda el *banco* únicamente tendría que enviar el valor de A'_2 (este mensaje es etiquetado como *Request41* en el capítulo 4) a la entidad *autoridad* para que ésta calcule $A_2'^{X_T} \equiv g_2^s \equiv A_2$ y devuelva este valor al *banco*, con lo cual permitiría que el *banco* supiera que *vendedor* le proporciono dicho valor y así de esta manera se podría saber en donde fue gastada la moneda en cuestión.

3.4. Variante del protocolo de Tsiounis et al.

En esta sección se propuso una variante al protcolo descrito en la sección 3.3 en donde se invulucre menor computo del lado del comprador.

3.4.1. Proceso de inicialización

La *autoridad*: realiza el mismo proceso descrito en la sección 3.3.1.

El *banco*: realiza el mismo proceso de inicialización que se estableció en el protocolo de S.Brands (sección 3.2), con la diferencia de que ahora los valores h, h_1, h_2 y h_3 se crearan de

comprador		*vendedor*
1. Establecimiento de compra	IdPedido, Monto	
1.1	(M1)$\Longleftrightarrow$	
2. Envio de Moneda		
2.1 $A_1 = I^s, A_2 = g_2^s$		
2.2 $m \in_R Zq$		
2.3 $D_1 = If_2^m$		
2.4 $D_2 = g_2^m$	$A_1, A_2, D_1, D_2, A, B, (z, a, b, r)$	
	(M2) $\Longrightarrow$	
3. Verificación de Moneda		Verifica que:
3.1		$A = A_1A_2g_3$, $A \neq 1$,
		$D_2 \neq 1$
		Firma$(A, B) = (z, a, b, r)$
4. Envio de Hash		
4.1		$d = H(A_1\|B_1\|A_2\|B_2\|M\|t)$
4.2		$s_0, s_1, s_2 \in_R Zq$
4.3		$D' = D_1^{s_0}g_2^{s_1}, D_2^{s_2}$
4.4	d, D', f_2'	$f_2' = f_2^{s_0}g_2^{s_2}$
	(M3) $\Longleftarrow$	
5. Para el control de fraudes		
5.1 $r_1 = d(u_1s) + x_1$		
5.2 $r_2 = ds + x_2$		
5.3 $V = H_1((D')^s)/(f_2')^{ms})$	r_1, r_2, V	
	(M4) $\Longrightarrow$	
6. Aceptación de la moneda		Verifica que:
6.1		$g_1^{r_1} = A_1^dB_1$
6.2		$g_2^{r_2} = A_2^dB_2$
6.3		$V = H_1(A_1^{s_0}A_2^{s_1})$

Figura 3.6: Proceso de pago/compra con rastreo de propietario

la siguiente manera:
$h \equiv g^{X_B}$, $h_1 \equiv g_1^{X_B}$ y $h_2 \equiv g_2^{X_B} \pmod p$ y $h_3 \equiv f_2^{X_B}$

El *comprador* y el *vendedor*: realizan exactamente el mismo proceso que en el protocolo de S. Brands visto en la sección 3.2.

Con esto termina el proceso de inicialización para este protocolo. Ahora veremos como fueron alterados lo procesos de retiro y de pago/compra. Para lograr con ello que se realice el menor computo posible del lado del *comprador*, sin perder la rastreabilidad condicional.

3.4.2. Modificación al proceso de retiro

La modificación realizada a este proceso consiste en disminuir el costo computacional para la realización de la prueba de logaritmos que se realizaba en el proceso mostrado en la figura 3.5, para ello nuestra modificación tiene su fundamento en el esquema Schnorr de autenticación [5], esta modificación puede verse en el paso número 2 del proceso que se muestra en la figura 3.7.

Como se puede apreciar en la figura 3.7 el costo para garantizar el rastreo de moneda se ve diminuido en una exponenciación y en un cálculo de una funcion hash, con respecto al subproceso mostado en la figura 3.5.

Lo interesante de esta modificación es que el rastreo de moneda se lleva a cabo de la misma manera en la que se hacia con el protocolo de Tsiounis et al. Es decir, para llevar a cabo el rastreo de la moneda el *banco* únicamente tendría que enviar el valor de A_2' a la entidad *autoridad* para que ésta calcule $A_2'^{X_T} \equiv g_2^s \equiv A_2$ y devolver este valor al *banco*, con lo cual permitiría que el *banco* supiera que *vendedor* le proporciono dicho valor y así de esta manera se podría saber en donde fue gastada la moneda en cuestión.

3.4.3. Modificación al proceso de pago/compra

La modificación realizada a este proceso va ligada a la modificación que se realizó anteriormente al proceso de retiro para lograr el rastreo de monedas. Como primer objetivo se quería reducir el computo del lado del *comprador* para cifrar la identidad del comprador para que posteriormente la entidad *autoridad* pudiera realizar un rastreo de la identidad del comprador, sin embargo, la reduccíon de computo no fue significativa en este proceso ya que solo se logro evitar una multiplicación, en el proceso de pago/compra modificado puede verse en la figura 3.8.

Ahora bien, para lograr el rastreo del propietario de la moneda, el *banco* tendría que enviar los datos (D1,D2) a la *autoridad* para que con su llave privada la entidad *autoridad* calcule $I = D_2 - (D_1)^{X_T}$, para depués devolverle este valor al *banco* y que éste pueda determinar la identidad del *comprador* buscando en su base de datos el identificador recibido.

3.5. El problema de la divisibilidad

Como ya se habrá notado ninguno de los anteriores protocolos de DE cumple con la propiedad de la divisivilidad (ver sección 2.4) esta carencia podría ser resuelta de varias maneras, a continuación se mencionan 3 de las formas más prácticas que pueden dar solución a la carencia que presentan estos protocolo.

1. Manejando monedas divisibles
2. Manejando monedas de diferentes denominaciones
3. Manejando saldo a favor

comprador		*banco*
1. Solicitud de la moneda	I, Retiro	
1.1	(M1) $\Longrightarrow$	
2. Datos para el rastreo de moneda		
2.1 $s, k \in_R Z_q$		
2.2 $A'_1 = Ig_2 f_2^{s^{-1}}$,	A'_1, A'_2	
2.3 $A'_2 = f_2^{s^{-1}}, y = g_2^k$	**(M1′)** $\Longrightarrow$	
3. Esquema Schnorr de autenticación	e	
3.1	**(M1″)** $\Longleftarrow$	$e = H(I\|y\|X_B)$
3.2 $r = es^{-1} + k$	r	
3.3	**(M1‴)** $\Longrightarrow$	Verifica que: $f_2^r = (A'_1/(Ig_2))^e y$ $A'_1/A'_2 = Ig_2 A'_2$
4. Identificador de la moneda	a', b'	$w \in_R Z_q$
4.1	(M2) $\Longleftarrow$	$a' = g^w$ y $b' = (A'_1)^w$
5. Creación de la moneda		
5.1 $A = (A'_1)^s$		
5.2 $z' = h_1^{u_1} h_2 h_3^{s^{-1}}, z = z'^s$		
5.3 $x_1, x_2, u, v \in_R Zq$		
5.4 $B_1 = g_1^{x_1}, B_2 = g_2^{x_2}$		
5.5 $B = [B_1, B_2]$		
5.6 $a = (a')^u g^v$		
5.7 $b = (b')^{su} A^v$		
5.8 $c = H(A\|B\|z\|a\|b)$	c'	
5.9 $c' = c/u$	(3) $\Longrightarrow$	
6. Firma de la moneda	r'	
6.1	(M4) $\Longleftarrow$	$r'^{=} c' X_B + w$
7. Composición de la moneda		
7.1 $r = r'u + v \bmod q$		
7.2 Verificar: $g^r \equiv h^{c'} a', (A'_1)^{r'} \equiv z'^{c'} b'$		

Figura 3.7: Proceso de retiro modificado para permitir el rastreo de monedas con un computo más reducido

De acuerdo a las opciones antes mencionadas la primera y la tercera opción fueron descartadas ya que la primera representaría un aumento en el computo que debe desarrollar el *comprador* [18] y la tercera representaría tener una absoluta confianza en la entidad *vendedor* de que nuestro saldo a favor sería almacenado y posteriormente entregado, lo que para un sistema finaciero traería consigo demasiadas alteraciones, tal vez para un sistema de consumo interno podría ser una buena opción pero en nuestro caso no lo es.

comprador		*vendedor*
1. Establecimiento de compra	IdPedido, Monto	
1.1	(M1)$\Longleftrightarrow$	
2. Envio de Moneda		
2.1 $A_1 = I^s, A_2 = g_2^s$		
2.2 $m \in_R Zq$		
2.3 $D_1 = g_2^m$		
2.4 $D_2 = I + f_2^m$	$A_1, A_2, D_1, D_2, A, B, (z, a, b, r)$	
	(M2) $\Longrightarrow$	
3. Verificación de Moneda		Verifica que:
3.1)		$A = A_1 A_2 f_2,\ A \neq 1,$
		$D_2 \neq 1$
		$\text{Firma}(A, B) = (z, a, b, r)$
4. Envio de Hash		
4.1)		$d = H(A_1 \vert B_1 \vert A_2 \vert B_2 \vert M \vert t)$
5.2)		$s_0 \in_R Zq$
5.3)		$D' = D_1^{s_0} D_2$
5.4)	d, D', f_2', f_3'	$f_2' = f_2 g_2^{s_0}, f_3' = D_1^{s_0}/g_2^{s_0}$
	(M3) $\Longleftarrow$	
5. Para el control de fraudes		
5.1) $r_1 = d(u_1 s) + x_1$		
5.2) $r_2 = ds + x_2$		
5.3) $V = H([(D' - f_2')/f_3']^s)$	r_1, r_2, V	
	(M4) $\Longrightarrow$	
6. Aceptación de la moneda		Verifica que:
6.1)		$g_1^{r_1} = A_1^d B_1$
6.2)		$g_2^{r_2} = A_2^d B_2$
6.3)		$V = H(A_1 A_2^{s_0})$

Figura 3.8: Proceso de pago/compra modificado para permitir rastreo de propietario

Por lo que se opto por escoger la opción del manejo de diferentes denominaciones, para ello el *banco* debe generar diferentes llaves privadas y públicas (un par por cada denominación). En nuestro caso se utilizaron únicamente tres denominaciones, por lo que, fueron generadas 3 llaves privadas en el proceso de inicialización X_{B_1}, X_{B_2} y X_{B_3} una para firmar las monedas de 100, otra para firmar las monedas de 200 y finalmente una última para firmar las monedas de 500. De igual forma se crearon tres diferentes tetrapletas de h's, (las llaves públicas):

$h100 \equiv g^{X_{B_1}}$, $h100_1 \equiv g_1^{X_{B_1}}$ y $h100_2 \equiv g_2^{X_{B_1}}$, $h100_3 \equiv g_3^{X_{B_1}} \pmod p$

$h200 \equiv g^{X_{B_2}}$, $h200_1 \equiv g_1^{X_{B_2}}$ y $h200_2 \equiv g_2^{X_{B_2}}$, $h100_3 \equiv g_3^{X_{B_2}} \pmod{p}$

$h500 \equiv g^{X_{B_3}}$, $h100_1 \equiv g_1^{X_{B_3}}$ y $h100_2 \equiv g_2^{X_{B_3}}$, $h100_3 \equiv g_3^{X_{B_3}} \pmod{p}$

De esta manera sería manejado el par de llaves correspondiente dependiendo de la denominación de la moneda a descargar en el proceso de retiro, de igual forma en el proceso de pago/compra se utilizarían los valores de las h's correspondiente a la denominación de la moneda con la que se esté realizando el pago. Este manejo de varias denominaciónes funciona para cualquiera de los protocolos seleccionados.

Dadas estas observaciónes en esta tesis se opto por el uso de los protocolos mencionados en las secciones 3.2, 3.3 y 3.4 usando 3 diferentes denominaciones para las monedas que se usen en nuestro sistema de dinero electrónico móvil.

Capítulo 4

Implementación del Sistema DEM

A lo largo de la tesis hemos visto y estudiado todo lo referente al dinero electrónico, desde sus bases criptográficas hasta la revisión de algunos de los protocolos propuestos a lo largo de la historia del dinero electrónico.

Este trabajo tiene como objetivo principal el desarrollo de una implementación eficiente, segura y portable para dispositivos móviles, para lograrlo se desarrolló un sistema de software, el cual fue diseñado e implementado como una aplicación WEB, esto para brindarle una mayor versatilidad. Para lograrlo el sistema está diseñado basado en el paradigma cliente - servidor [30].

En este capitulo describiremos cómo se diseñó e implementó nuestro sistema de dinero electrónico móvil (DEM). El sistema será diseñado para soportar los dos esquemas más comunes de dinero electrónico: el esquema Básico y el esquema FOLC.

4.1. Diseño del sistema DEM

En esta sección describiremos el diseño del Sistema de Dinero Electrónico Móvil (DEM), comenzando por la descripción general del sistema, en donde se explicará de manera concisa el sistema; seguido por la arquitectura propuesta, la cual, nos mostrará de manera global la estructura a manera de capas del sistema DEM. Continuaremos con la especificación de los submódulos que componen la capa de la aplicación WEB de cada una de las entidades; finalmente en la sección 4.4 se muestra el diagrama de secuencias correspondiente a cada uno de los proceso que realizan las entidades.

4.1.1. Descripción del sistema DEM

El sistema es una emulación tan real como fue posible, de lo que debería ser un sistema de dinero electrónico, por lo que, el sistema propuesto en esta tesis tiene los elementos suficientes para poder emular de forma creíble un sistema de este tipo.

Se decidió que el sistema fuera desarrollado a manera de un sistema de aplicación WEB, para así lograr ejemplificar de manera más convincente un sistema de dinero electrónico. Esencialmente, nuestro sistema define tres servidores independientes (la *autoridad*, el *banco* y el *vendedor*) los cuales funcionan de forma autónoma uno de otro. Por otra parte, tenemos dos tipos de clientes (los *compradores*) que estarán definidos por el medio que usen para conectarse a los servidores, ya sea mediante una computadora de escritorio o una computadora portátil (PC), o bien mediante un asistente digital personal (PDA). El acceso a estos servidores se realiza vía Internet (de forma alámbrica ó inalámbrica), aunque también podría llevarse a cabo mediante una red inalámbrica ad-hoc. A su vez, el sistema puede trabajar con tres diferentes protocolos, el propuesto por Brands[9], el propuesto por Yannis et al [12] y la variante de estos.

Fueron definidos dos escenarios con los cuales es posible que trabaje el sistema. El escenario A el cual establece que todas las entidades se comuniquen a través de Internet ya sea alámbrica o inalámbricamente y el escenario B donde es posible que mediante un dispositivo móvil de manera inalámbrica el *comprador* pueda acceder ya sea al servidor del *banco* para descargar monedas de manera local o bien que el mismo *comprador* pueda gastar sus monedas con el servidor del *vendedor* estando presente en las instalaciones de la tienda. Cada uno de los canales de comunicación es establecido como un canal seguro, utilizando el protocolo de comunicación TLS sobre HTTP, es decir, el protocolo de comunicación es el conocido HTTPS. En la figura 4.1 se puede observar con mayor detalle los escenarios que pretendemos abarcar con nuestro sistema.

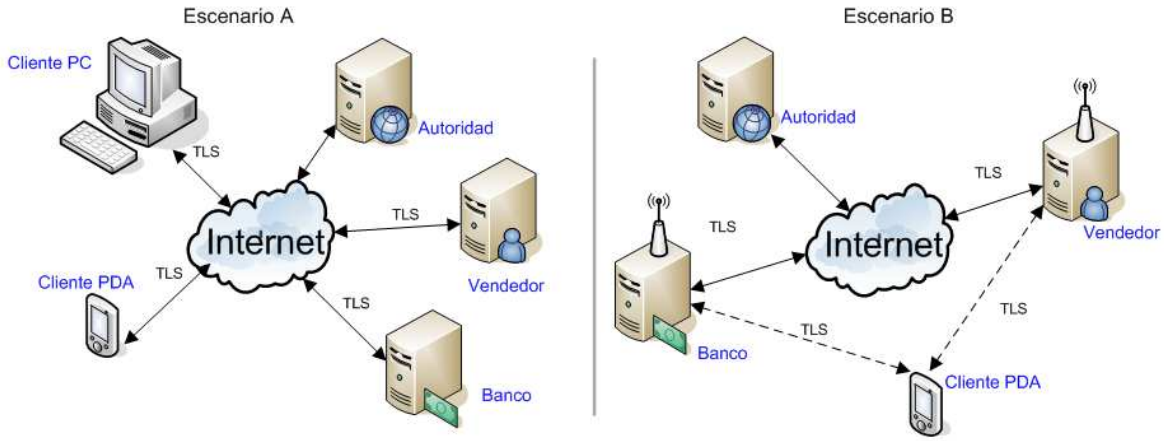

Figura 4.1: Posibles escenarios en los que pudiera trabajar el sistema

Es necesario recalcar que en la figura 4.1 sólo se hace mención a la forma en la que las entidades se comunicarán unas con otras (dependiendo del escenario), pero en ningún momento se establece que dicha comunicación será permanente o en la misma instancia de tiempo. Más adelante se especificará con más detalle la forma en la que los procesos (retiro, pago/compra, depósito, etc.) son desarrollados por cada una de las entidades.

Cada una de las tres entidades servidor cuenta con:

- Un sitio WEB dinámico, mediante el cual, el usuario puede ingresar, ver la información necesaria, y manipular dicha información de acuerdo a sus deseos y limitaciones del sitio WEB, de esta manera también puede iniciar la aplicación correspondiente, la cual realizará la interacción entre el cliente (el *comprador*) y el servidor al que se está ingresando (puede ser el *banco* o el *vendedor*).
- Un sitio WEB administrativo, el cual ayuda al usuario administrador a realizar los procesos administrativos del sistema, para realizar el ingreso a este sitio se requiere de un nombre de usuario y su correspondiente contraseña.
- Una aplicación WEB, con la cual, el usuario (PC) al ingresar al sitio WEB puede interactuar de acuerdo a las especificaciones del protocolo de dinero electrónico que se esté utilizando en ese momento.

Por otro lado los clientes (PDA), cuentan con:

- Una aplicación Java Frame, con la que el usuario (PDA) al ingresar y establecer los parámetros requeridos podrá interactuar con el servidor al que se esté conectando de acuerdo a las especificaciones del protocolo de dinero electrónico que se haya especificado.

Por último los clientes (PC) se componen únicamente de:

- Un navegador WEB, el cual, debe soportar el manejo de Java Applets, con dicho navegador el usuario podrá interactuar con los diferentes servidores via HTTPS, pudiendo realizar ya sea el retiro de monedas electrónicas o bién el pago/compra de algún producto o servicio utilizando las monedas electrónicas descargadas.

No olvidemos que tanto los clientes como los servidores deben de contar con lo necesario para que la comunicación entre ellos se lleve a cabo mediante el protocolo de comunicación TLS. Esto para efectos de mayor seguridad, tratando de evitar ataques como el del "intruso de enmedio", robo de información, etc.

El sistema lleva a cabo los siguientes procesos generales:

- Proceso de retiro, con el sistema es posible retirar monedas electrónicas, a partir del ingreso al servidor *banco*, de forma segura y eficiente. Almacenando dichas monedas en un archivo el cual a su vez puede ser guardado en cualquier dispositivo de almacenamiento secundario (disco duro, memoria flash, etc.). En este proceso sólo interviene la entidad *comprador* (PC o PDA) y el servidor *banco* (recuérdese que se han planteado dos posibles escenarios la figura 4.2 muestra las entidades y la forma en la que se comunicarían las dos entidades antes mencionadas).

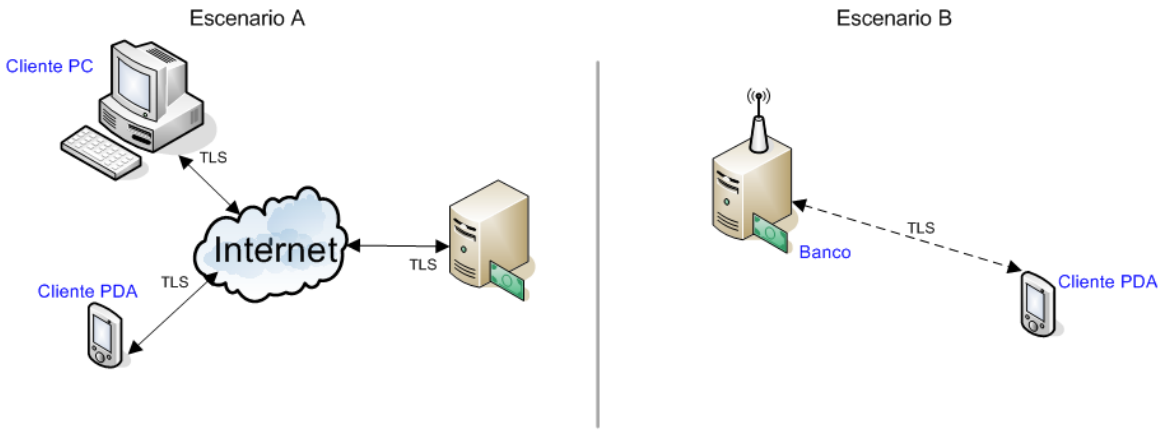

Figura 4.2: Proceso de retiro de monedas electrónicas

- Proceso de pago/compra, una vez que el cliente (*comprador*) ya sea la PC o la PDA tiene disponible el archivo que contiene las monedas electrónicas válidas, entonces es posible que éste acceda al servidor del *vendedor* para gastar dichas monedas a cambio de obtener algún producto, bien o servicio que el *vendedor* tenga a disposición de sus clientes. En la figura 4.3 se presentan los dos escenarios posibles para la realización de este proceso.

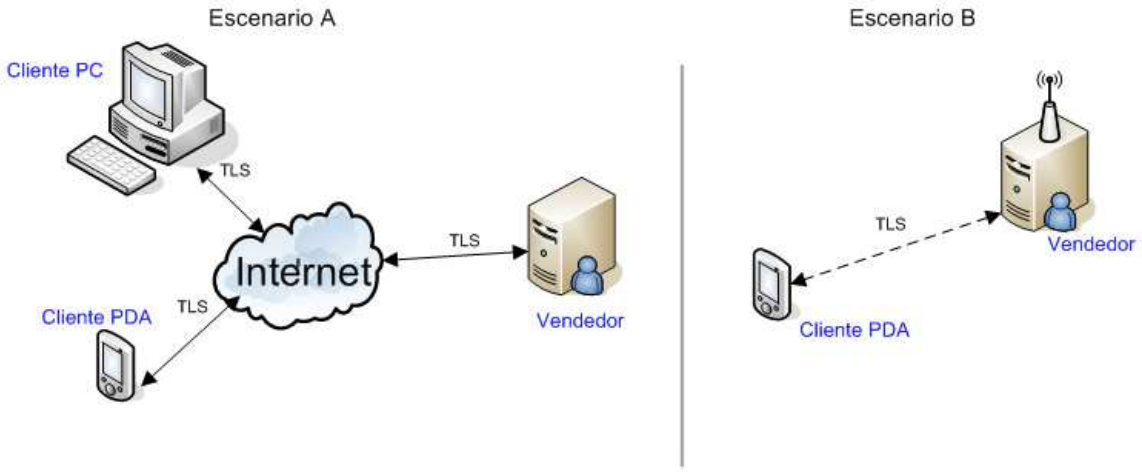

Figura 4.3: Proceso de pago/compra con monedas electrónicas

- Proceso de depósito, cuando el *vendedor* tiene en su poder las monedas electrónicas que los *compradores* le han entregado a cambio de alguno de sus productos o servicios, el *vendedor* tiene la posibilidad de depositar dichas monedas en su cuenta para así obtener

el dinero real que representan dichas monedas electrónicas; para llevar a cabo el proceso de depósito es necesario que el *vendedor* le envíe al *banco* las monedas recibidas de los compradores para que el *banco* abone el monto de dichas monedas a la cuenta del *vendedor*, esto siempre y cuando las monedas sean válidas. Las entidades y la manera en la que se comunicarán dichas entidades entre sí en cualquiera de los escenarios posibles puede verse en la figura 4.4.

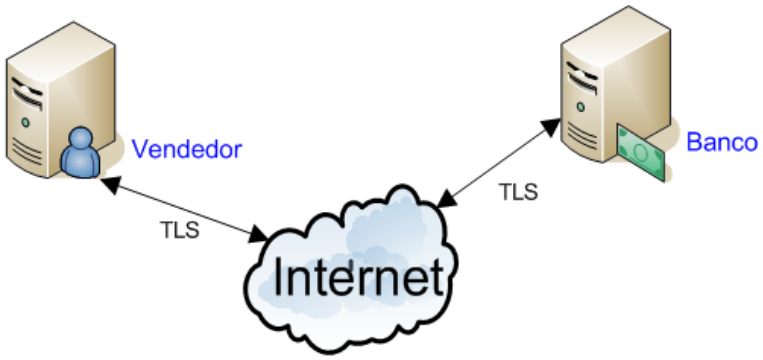

Figura 4.4: Proceso de depósito de monedas electrónicas

- Proceso rastreo de propietario y de moneda, cuando alguno de los compradores o la compra en sí, se vean envueltos bajo sospecha de algún crimen, el sistema es capaz de permitir el rastreo del propietario de la moneda o bien el rastreo de donde fue gastada alguna moneda. Para la realización de este proceso es necesario que el servidor *banco* se comunique con el servidor *autoridad* para que ésto último le proporcione la información necesaria y así el *banco* pueda obtener los datos requeridos. Claramente para que el *banco* realice este proceso debe de contar con una autorización legal y una autorización electrónica para acceder al servidor *autoridad* e iniciar el proceso de rastreo. En la figura 4.5 se muestra la forma en la que se comunican las entidades involucradas en este proceso.
- Proceso Control de fraudes, cuando algún *comprador* malicioso intente gastar las monedas dos o más veces, el *banco* cuenta con los mecanismos necesarios para obtener la identidad del usuario tramposo de manera inmediata, a partir de un doble depósito de la misma moneda, y de esta forma establecer las medidas pertinentes en contra de dicho usuario. Este es un proceso interno del *banco* y no requiere comunicación con alguna otra entidad para llevarse a cabo.

A continuación se describe la arquitectura del sistema donde se verá a detalle cómo están compuestas cada una de las entidades, así como el objetivo que tiene cada uno de los componentes que las conforman.

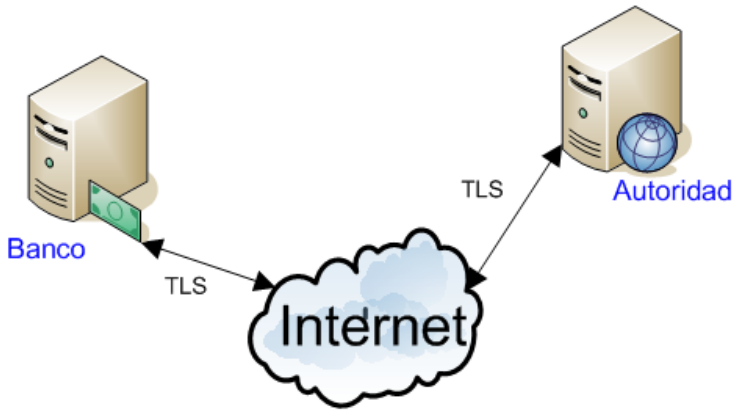

Figura 4.5: Proceso de rastreo de propietario y de moneda

4.2. Arquitectura del sistema DEM

La arquitectura se implementó bajo el modelo cliente-servidor de tres capas [30] ya que el procesamiento de la aplicación, la administración de la base de datos y la ejecución de la aplicación están perfectamente separados.

Como ya se mencionó en la subsección 4.1.1 el sistema cuenta con tres entidades servidor: la *autoridad*, el *banco* y el *vendedor*, sin embargo, si se ve con más detalle a estas entidades, en algunos procesos ellas pueden tambien fungir a manera de clientes solicitando o enviando alguna información a algún servidor. Además de estas tres entidades servidor tenemos dos entidades cliente (*comprador* PC, *comprador* PDA). A continuación se comienza definiendo la arquitectura de cada una de las entidades servidor, siguiendo con la definición de la arquitectura de las entidades cliente para por último revisar la arquitectura del sistema completo.

La arquitectura en base a la cual fueron construidos los tres servidores es muy similar. En primera instancia tenemos que cada uno de los servidores está creado basando su funcionamiento en el modelo entrada-proceso-salida [30]. Esto quiere decir que cualquiera de los servidores funciona a partir de alguna información solicitada a través de una petición WEB, el servidor deberá responder devolviendo ya sea una pagina WEB o bien la información solicitada, la cual ya ha sido procesada a partir de la información enviada en la petición del cliente.

En la figura 4.6 podemos ver la arquitectura de las tres entidades servidor (la *autoridad*, el *banco* y el *vendedor*), cada una de ellas mostrada a manera de capas en donde la comunicación fluye de manera vertical. Cada entidad está conformada de un servidor de base de datos, de un conjunto de servlets o de CGI´s los cuales realizan los procesos necesarios de acuerdo a la petición WEB solicitada, de un servidor WEB, el cual debe de tener soporte para el manejo

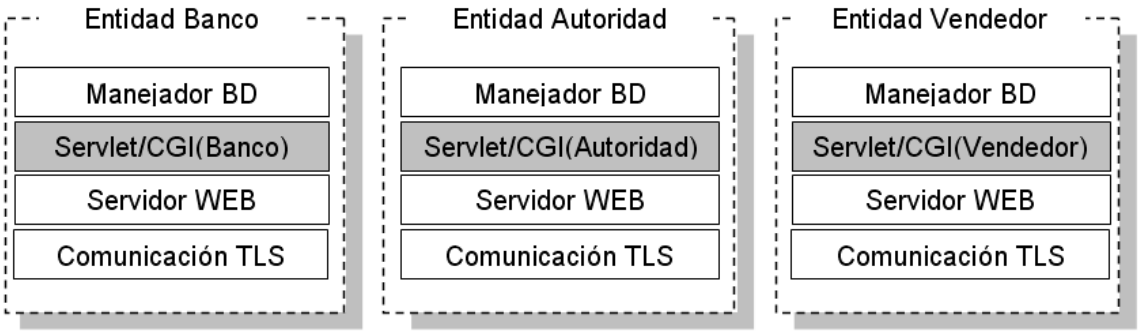

Figura 4.6: Arquitectura de las tres entidades servidor

de CGI´s/Servlets y finalmente de un manejador de comunicación segura mediante TLS. Los CGI´s/Servlets son realmente el corazón de sistema, ya que en ellos está programada la funcionalidad que deben tener cada una de las entidades.

La entidad *comprador* PC, está constituida únicamente por un navegador WEB el cual debe contar con la capacidad de procesar applets, ya que es de esta forma como el *comprador* realizará los procesos principales descritos en la sección anterior (aunque otra opción de diseño sería crear una aplicación cliente que interactúe directamente con los servidores). Cabe aclarar que dichos applets serán descargados del servidor al que se esté ingresando vía el navegador WEB. La figura 4.7 muestra la constitución de esta entidad.

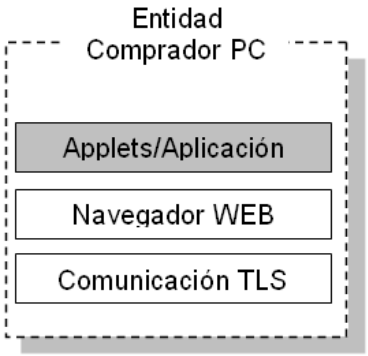

Figura 4.7: Arquitectura de la entidad *comprador* para PC

La entidad *comprador* PDA, fue constituida de manera diferente ya que los navegadores existentes para la PDA no tienen soporte para applets, por lo que esta entidad quedó constituida únicamente por la comunicación TLS y por una aplicación, con la cual se realizan los procesos descritos en la sección anterior por parte del *comprador*. En la figura 4.8 podemos ver cómo quedó constituida esta entidad.

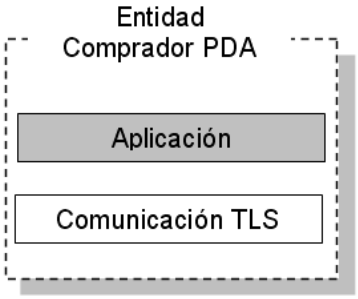

Figura 4.8: Arquitectura de la entidad *comprador* para PDA

A continuación en la figura 4.9 podemos ver la arquitectura general del sistema, utilizando las entidades anteriormente descritas. Como se puede observar, la arquitectura de capas presentada para cada una de las entidades del sistema funciona de manera similar, siendo la capa de más abajo la capa de la comunicación segura, la cual utiliza el protocolo TLS. Observe que todos los procesos llevan a cabo su comunicación a través de dicha capa, manteniendo así la seguridad de los datos que viajan a través de la red. Finalmente notemos que los servidores (*banco* y *vendedor*) están diseñados para responder a una solicitud de una PC o la de una PDA.

4.3. Diagramas de secuencias del sistema DEM

En la sección anterior se vio la composición a manera de capas de cada una de las entidades del sistema DEM, en la figura 4.9 se mostró la arquitecura general del sistema, en base a dicha arquitectura se realizaron los siguientes diagramas de secuencias, en los cuales, se muestra la interacción que existe entre cada una de las entidades en determinado proceso.

4.3.1. Proceso de retiro

Para la realizción de este proceso intervienen sólo la entidad *comprador* (PC o PDA) y la entidad *banco*. Una mayor descripción de este proceso puede verse en la sección 4.1.1. El diagrama de secuencias de la figura 4.10 nos muestra la interacción que existe entre las entidades *comprador* PC y el *banco*.

Como ya se dijo en la sección 4.2, la composición de la arquitectura del *comprador* para PC es diferente a la arquitectura de la entidad *comprador* para PDA, por lo que, el correspondiende diagrama de secuencias también difiere uno de otro. En la figura 4.11 podemos ver el diagrama de secuencias para el proceso de retiro entre la entidad *comprador* para PDA y el *banco*.

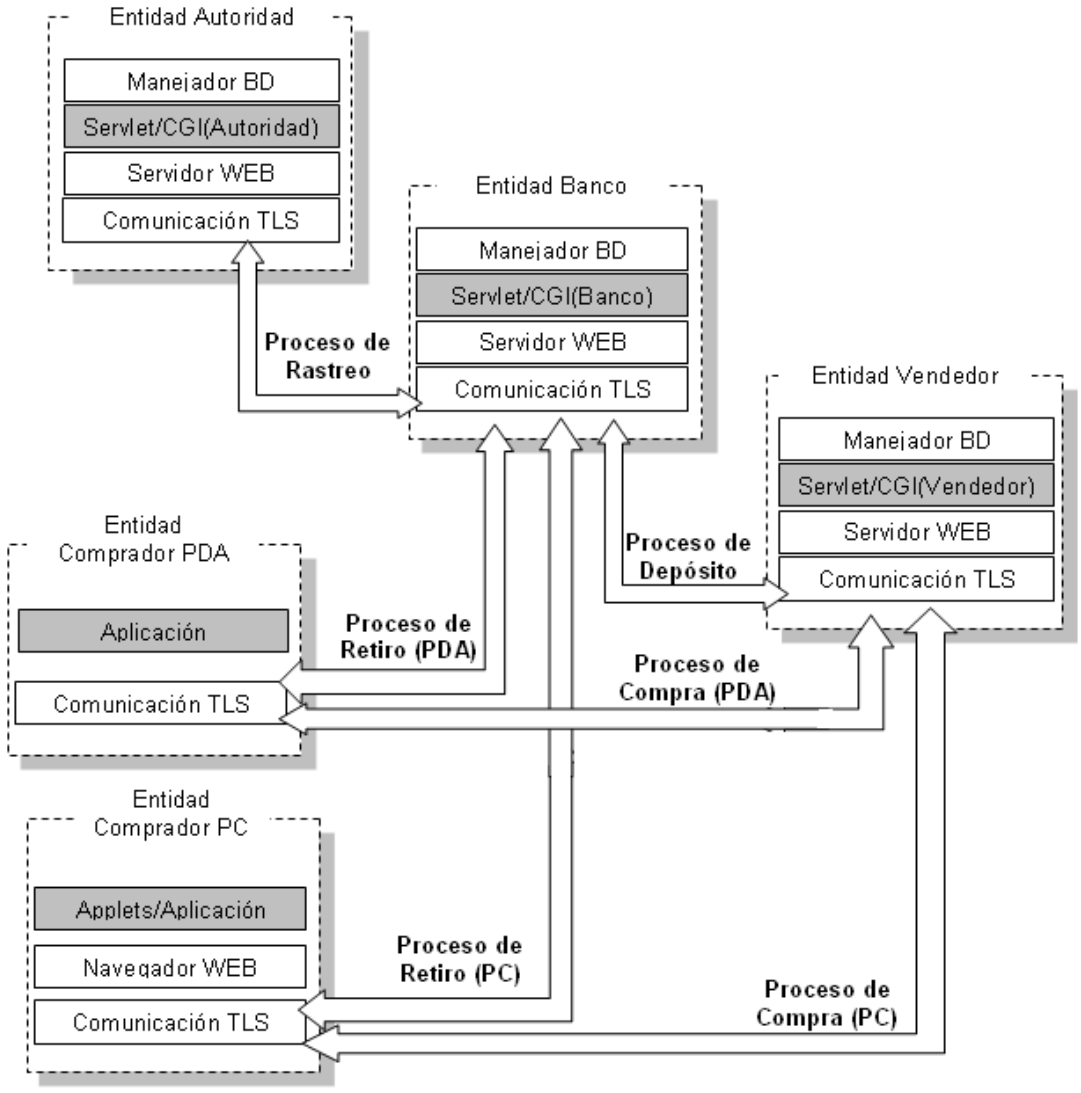

Figura 4.9: Arquitectura completa del sistema DEM

4.3.2. Proceso de pago/compra

En este proceso intervienen la entidad *comprador* (PC o PDA) y la entidad *vendedor*. En la sección 4.1.1 se expuso a detalle el objetivo que persigue la realización de este proceso. En el diagrama de secuencias de la figura 4.12 se puede ver la interacción que se implementó entre las entidades *comprador* PC y el *vendedor* para poder realizar este proceso.

De igual manera que el proceso de retiro, el proceso de pago/compra puede ser realizado también por una entidad *comprador* para PDA. En la figura 4.13 podemos ver el diagrama de secuencias para el proceso de pago/compra entre la entidad *comprador* para PDA y el *banco*.

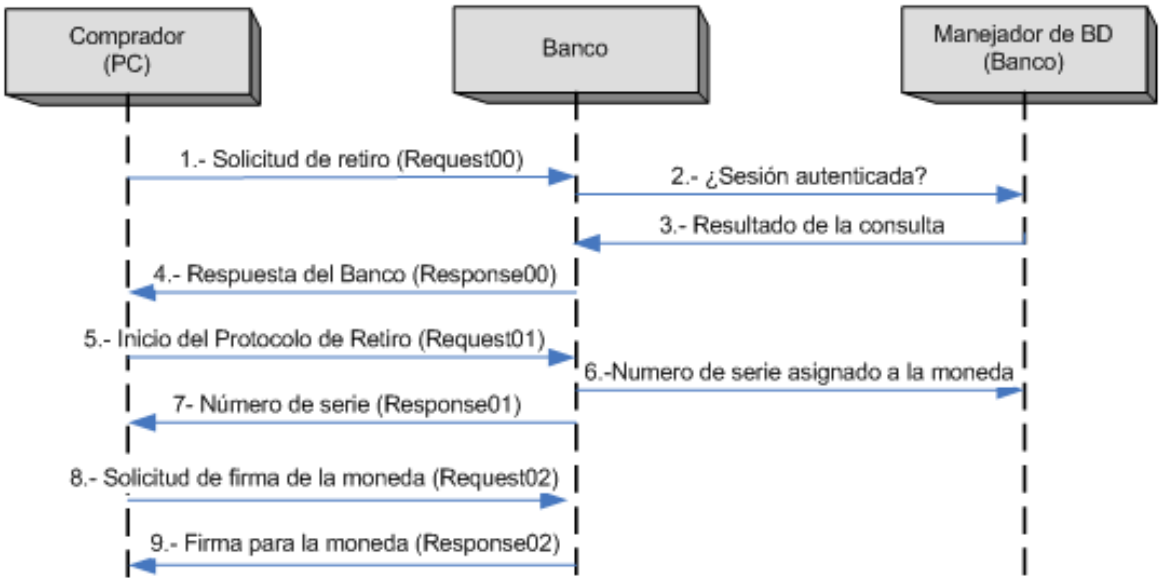

Figura 4.10: Diagrama de secuencias del proceso de retiro para PC

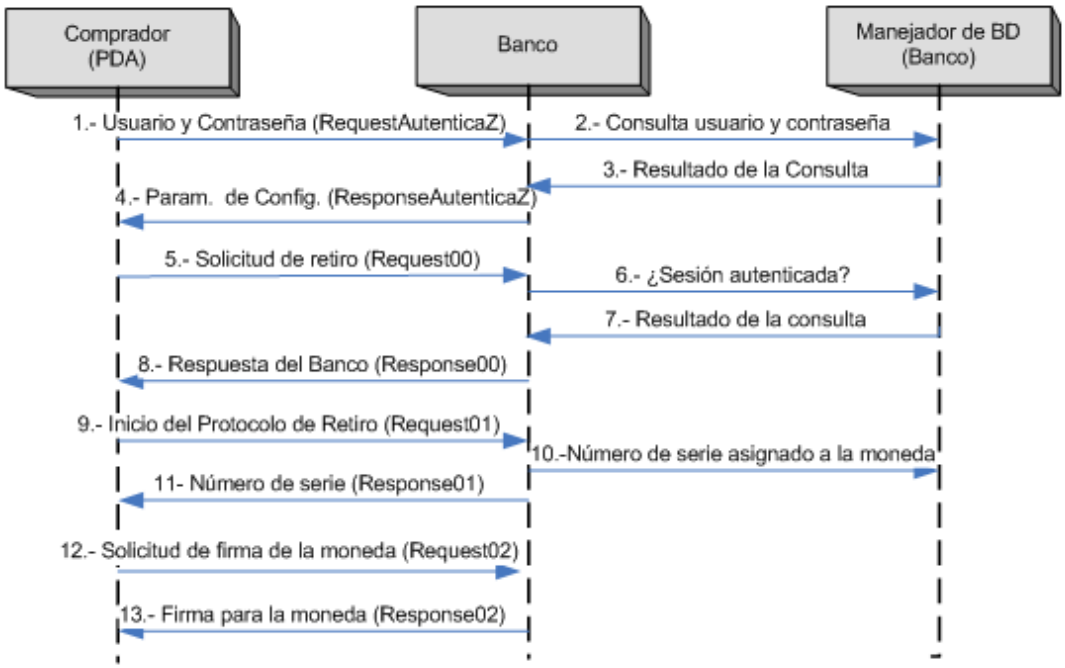

Figura 4.11: Diagrama de secuencias del proceso de retiro para PDA

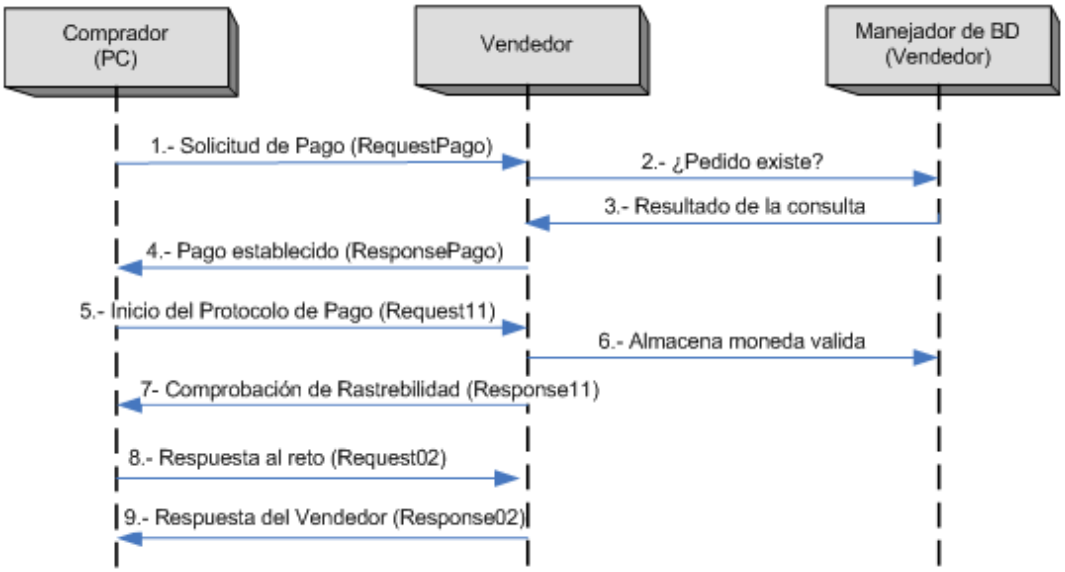

Figura 4.12: Diagrama de secuencias del proceso de pago/compra para PC

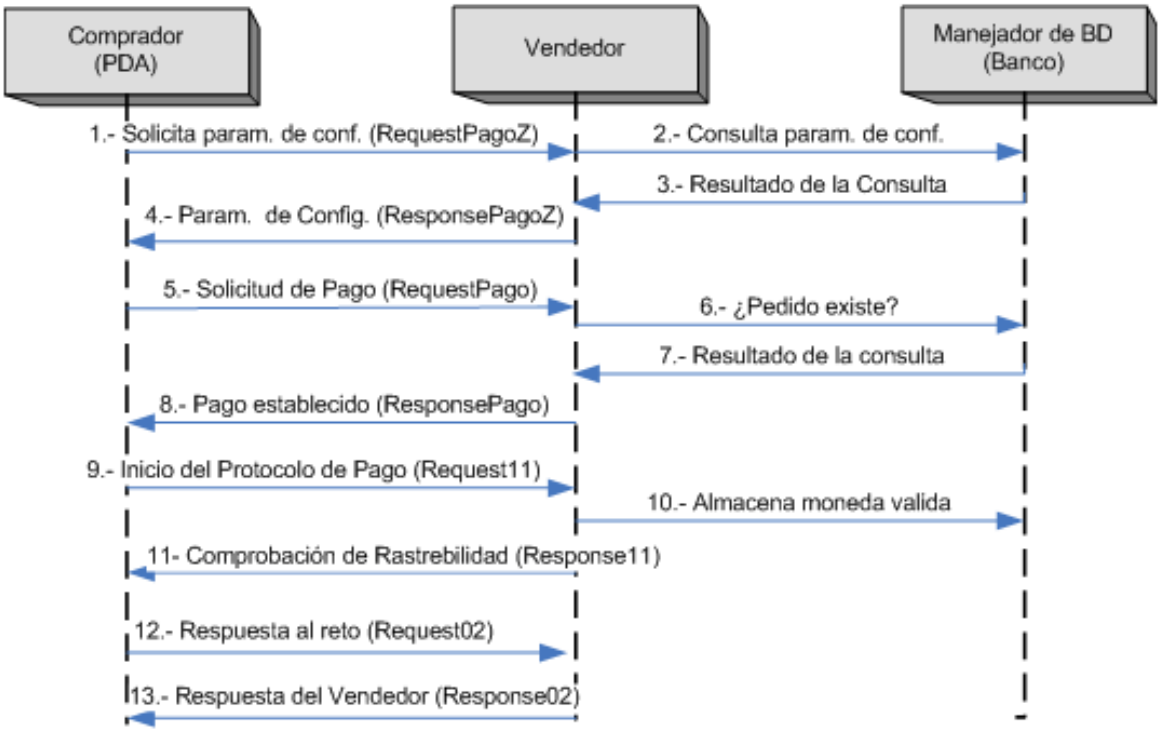

Figura 4.13: Diagrama de secuencias del proceso de pago/compra para PDA

4.3.3. Proceso de depósito

En este proceso intervienen la entidad *vendedor* y la entidad *banco*. Una mayor descripción del objetivo de este proceso puede verse en la sección 4.1.1. En la figura 4.14 se muestra

el diagrama de secuencias de las entidades *vendedor* y *banco*, exponiendo la interacción que fue implementada para lograr que se realice este proceso de pago/compra. En este proceso debemos notar que el *vendedor* esencialmente funge como servidor. En este proceso el *vendedor* se comporta como cliente, ya que es quien le solicita a la entidad *banco* que realice el proceso de depósito con las monedas que le está enviando.

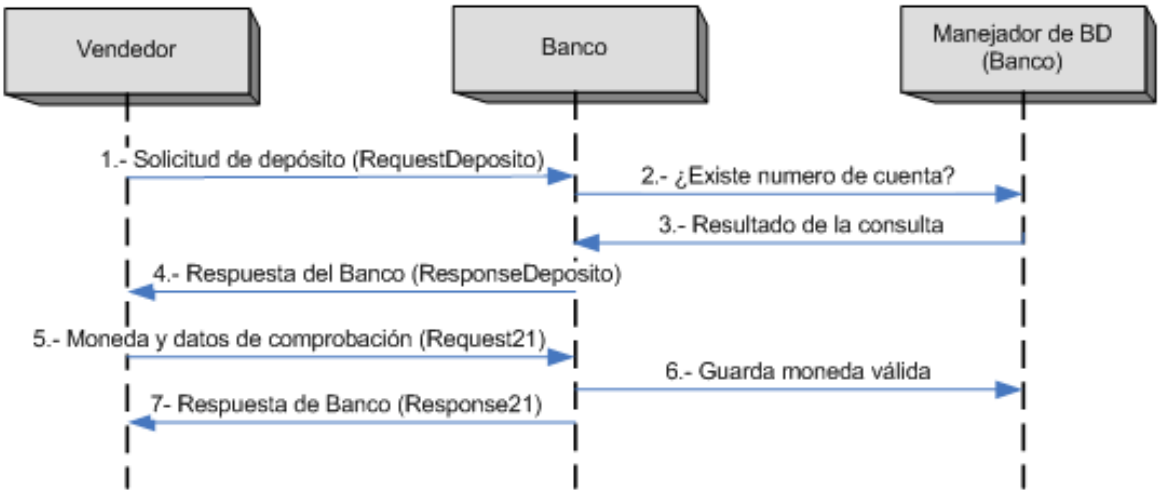

Figura 4.14: Diagrama de secuencias del proceso de depósito

4.3.4. Proceso de rastreo

En el Proceso de rastreo intervienen la entidad *banco* y la entidad *autoridad*. Para revisar con mayor detalle las tareas a cargo de este proceso puede verse la sección 4.1.1. El diagrama de secuencias de la figura 4.15 nos muestra la interacción que existe entre las entidades *banco* y la entidad *autoridad* cuando se realiza el proceso de rastreo. Debe notarse que en este proceso el *banco* actúa como cliente solicitandole a la entidad *autoridad* la petición para efectuar el rastreo ya sea de un propietario de un moneda o bien de una moneda.

4.4. Estructura interna del sistema DEM

En la sección anterior se mostró a partir de los diagramas de secuencias la interacción que existe en los diferentes procesos del sistema. En esta sección se revisa la composición interna del bloque de los CGI´s/Servlets para cada una de las entidades del sistema DEM, mostrando los submódulos que conforman cada una de éstas y además explicando de manera muy general el proceso que realiza cada uno de estos submódulos.

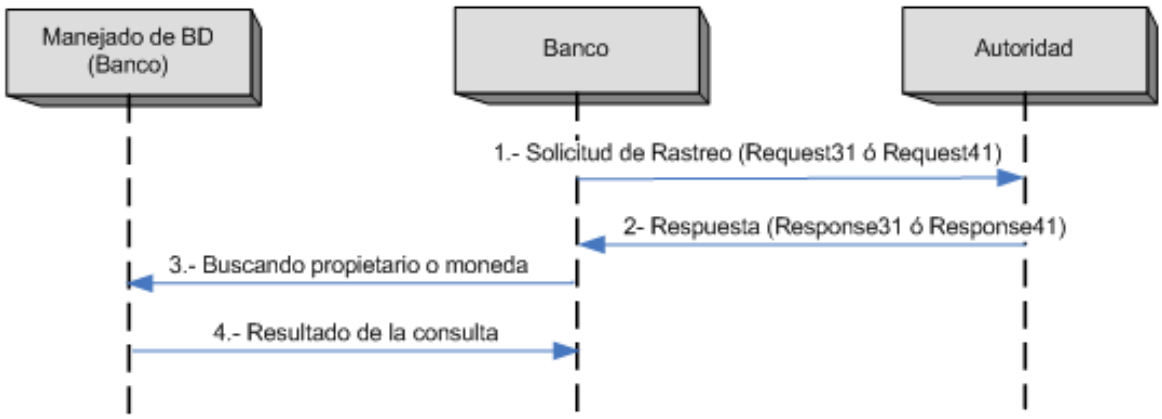

Figura 4.15: Diagrama de secuencias del proceso de rastreo

4.4.1. El *banco*

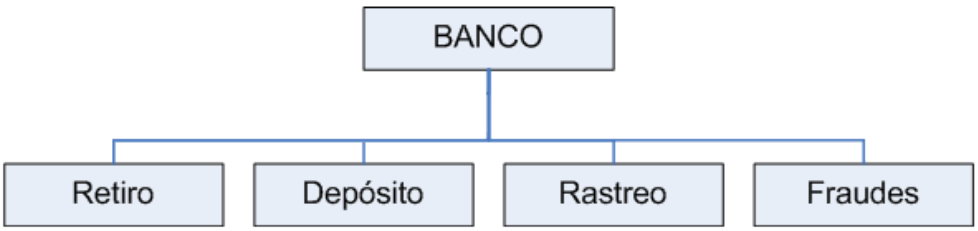

Figura 4.16: Diagrama a bloques de la entidad *banco*

De acuerdo a la descripción del sistema que se presentó en la subsección 4.1.1, en la aplicación WEB del *banco* se llevan a cabo 4 procesos (retiro, depósito, rastreo y control de fraudes), de acuerdo a estos proceso fueron creados los submódulos que componen esta entidad. En la figura 4.16 se muestra el diagrama a bloques de los principales submódulos que conforman la entidad *banco*.

En el apéndice A se muestra con mayor detalle los servlets y las clases que componen cada uno de los submódulos de la entidad *banco*.

A continuación se describe de manera muy general el proceso que es realizado por cada uno de los submódulos de la entidad *banco*.

Submódulo de retiro. Como se puede apreciar en la figura 4.17, cuando se solicita un retiro de monedas electrónicas el submódulo es ejecutado el proceso de retiro; en el cual, se

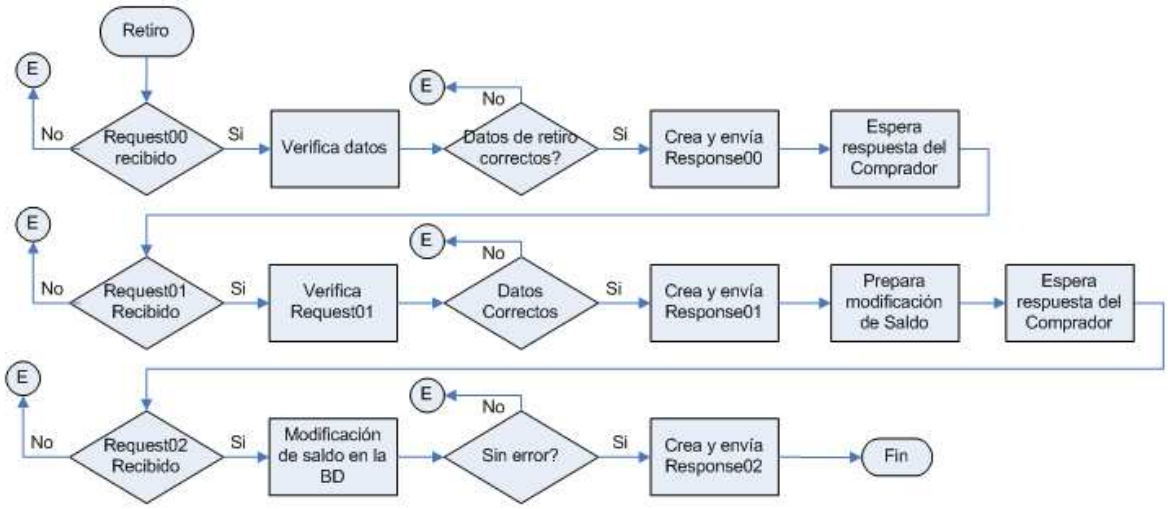

Figura 4.17: Proceso que realiza el submódulo de retiro

debe recibir en primera instancia un objeto Request00. Al recibirlo serán verificados los datos de dicho objeto, si son correctos entonces los datos contenidos en el Request00 son procesados y es creado y enviado al *comprador* un objeto Response00, para después esperar a que la respuesta del *comprador* sea un objeto Request01 el cual de igual manera es verificado y procesado para crear un objeto Response01, si todo va bien el *banco* se prepara para modificar el saldo de la cuenta del *comprador* pero antes de hacerlo el *banco* debe recibir un objeto Request02 verificarlo y procesarlo para que finalmente se modifique el saldo en la BD y le devuelva al *comprador* un objeto Response02. Cada uno de estos objetos Request y Response es creado y verificado de acuerdo al protocolo que se encuentra en uso.

Submódulo de depósito. Si algún *vendedor* solicita depositar una o varias monedas el *banco* a través de su submódulo de depósito realiza el proceso que se muestra en la figura 4.18. Éste espera recibir primero un objeto Request21 por cada una de las monedas a depositar. Entonces el *banco* verifica los datos contenidos en dicho objeto empezando por la moneda, la cual se encuentra contenida en el objeto Request21. Si la moneda y los datos de comprobación son correctos se crea y envía la respuesta al *vendedor*, se deposita el monto de la moneda en la cuenta del *vendedor* y se guarda la moneda en la base de datos. Si la moneda es una moneda duplicada se inicia el proceso de control de fraudes marcando la moneda como duplicada. De igual manera que en el proceso anterior la composición del objeto Request21 depende del protocolo que se esté utilizando.

Submódulo de rastreo El submódulo de rastreo lleva a cabo su proceso de acuerdo al diagrama del flujo presentado en la figura 4.19, en donde, primero el administrador del *banco*, una vez autenticado como tal, solicita el proceso de rastreo, el *banco* despliega entonces las monedas que ha recibido, el administrador selecciona las monedas a procesar y el *banco* crea y envía a la entidad *autoridad* un objeto Request31 si es un rastreo de propietario o un

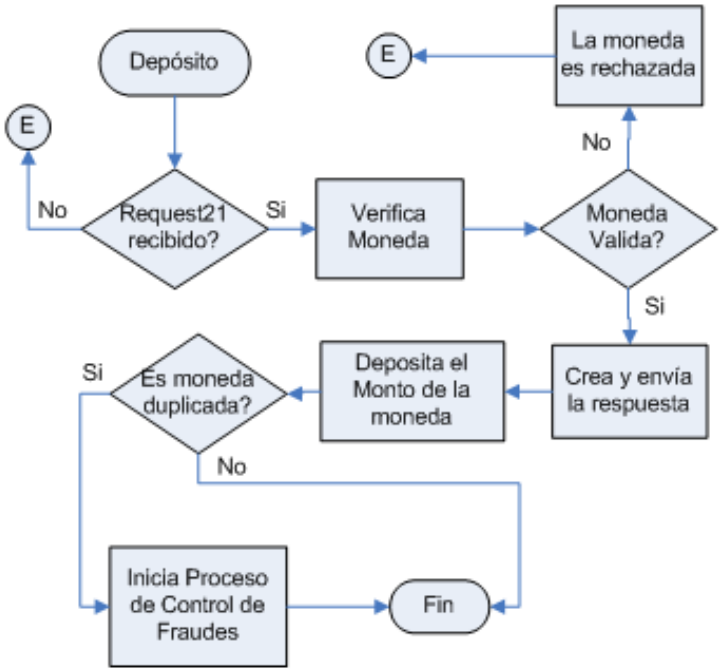

Figura 4.18: Proceso que realiza el submódulo de depósito

objeto Request41 si es un rastreo de moneda, dependiendo que opción se haya seleccionado se deberá recibir un objeto Response31 o un objeto Response41 con los cuales el *banco* será capaz de procesar la información que contiene el objeto recibido y desplegar ya sea la identidad del *comprador* o bien la tienda en la que fue gastada la moneda. Este proceso sólo puede ser llevado a cabo si para el retiro y el depósito de las monedas electrónicas se utilizo el protocolo propuesto en esta tesis o el protocolo propuesto por Yannis et al. [12].

Submódulo de control de fraudes En la figura 4.20 se muestra el diagrama de flujo del proceso que realiza el submódulo de control de fraudes, el cual, es llevado a cabo de la siguiente manera: primero el administrador del *banco*, una vez que se ha autenticado como tal, solicita el proceso de control de fraudes. El *banco* despliega entonces una lista con las monedas duplicadas dentro del sistema, el administrador escoge las monedas a procesar, el *banco* toma dichas monedas y las procesa de acuerdo al protocolo establecido y de esta manera encuentra el identificador del *comprador* tramposo, el *banco* busca en su Base de datos el nombre que corresponde al identificador encontrado y lo despliega en una lista de tramposos dando la opción para que el administrador cancele la cuenta de dicho *comprador*.

4.4.2. El *vendedor*

Acá se describe la aplicación WEB de la entidad *vendedor*, de acuerdo a la descripción del sistema que se presenta en la subsección 4.1.1. El *vendedor* lleva a cabo 2 procesos

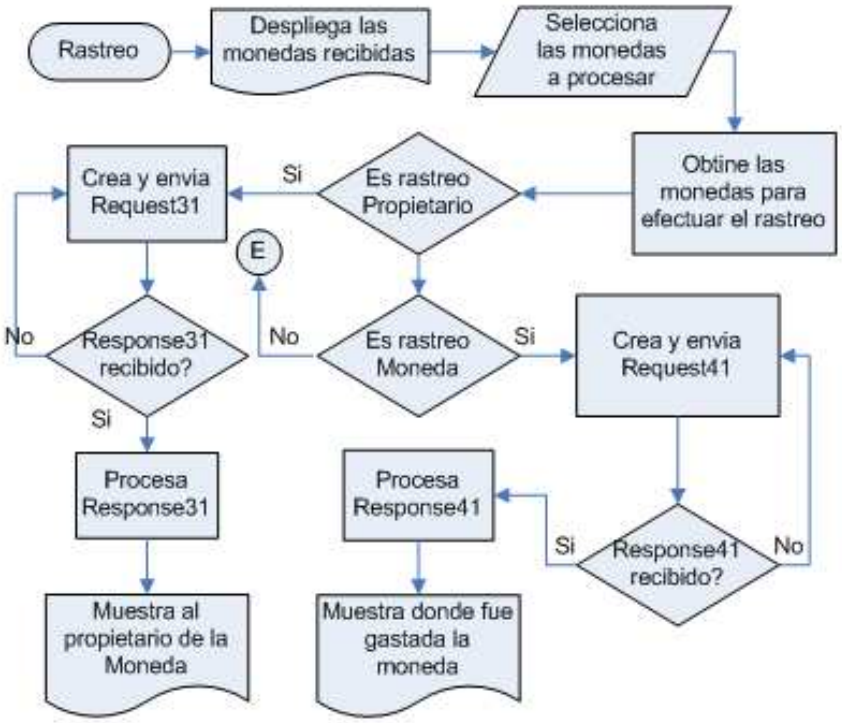

Figura 4.19: Proceso que realiza el submódulo de rastreo

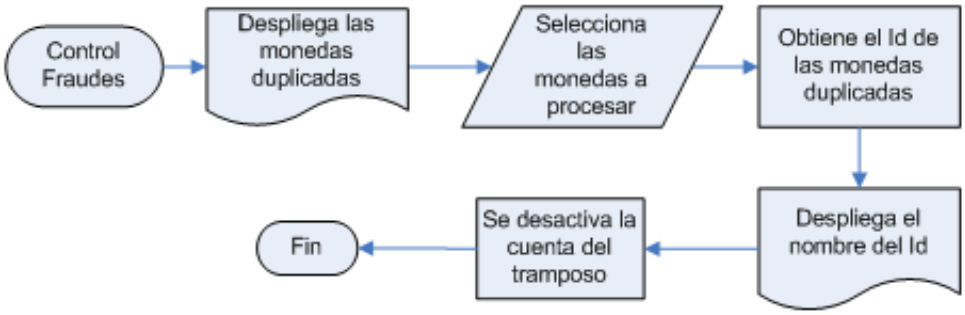

Figura 4.20: Proceso que realiza el submódulo de control de fraudes

(pago/compra y depósito), para ello cuenta con otro conjunto de servlets con los cuales el *vendedor* podrá realizar cada uno de estos procesos, en la figura 4.21 se muestra el diagrama a bloques de los principales submódulos que conforman la entidad *vendedor*.

En el apéndice A se muestra con mayor detalle los servlets y las clases que componen cada uno de los submódulos de la entidad *vendedor*.

El funcionamiento de cada uno de los submódulos que componen a la entidad *vendedor* se revisa a continuación:

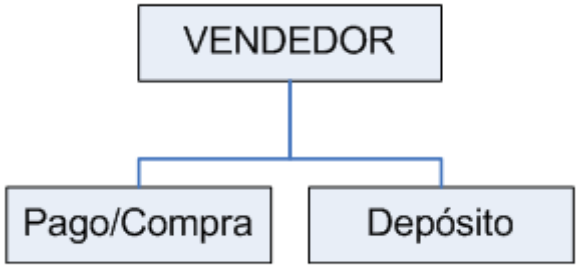

Figura 4.21: Diagrama a bloques de la entidad *vendedor*

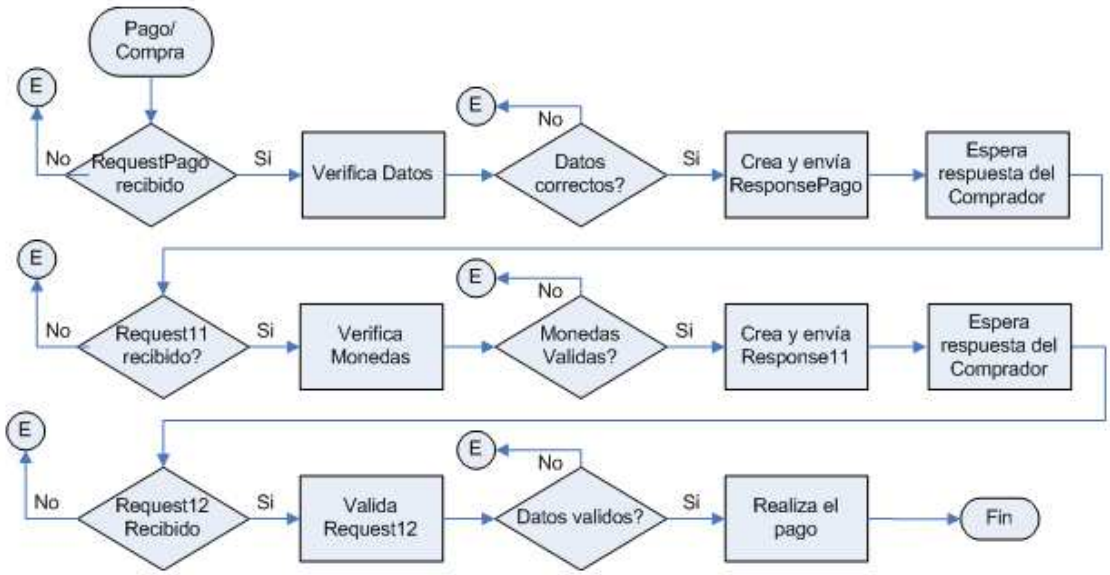

Figura 4.22: Proceso que realiza el submódulo de pago/compra

Submódulo pago/compra Cuando se solicita el pago de un pedido realizado en la tienda virtual de la entidad *vendedor*, utilizando monedas electrónicas, el modulo de pago/compra realiza el proceso que se presenta en el diagrama de flujo que se muestra en la figura 4.22. Primero se debe recibir un objeto RequestPago, al recibirlo serán verificados los datos de dicho objeto, si son correctos entonces se le pide al *comprador* que envíe las monedas electrónicas, estas vendrán encapsuladas en el objeto Request11 junto con los datos para efectuar la verificación de las mismas, al recibir el objeto Request11 el *vendedor* valida las monedas, si son válidas le pide un último reto enviándole un objeto Request12, si el *comprador* es el verdadero propietario de las monedas entonces responderá con un objeto Response12, el cual es validado por el *vendedor*. Si es correcta la validación el pago con las monedas electrónicas es realizado. Los datos que conforman los objetos Request y Response, para este proceso, dependen del protocolo que se esté utilizando. Para un mayor detalle del contenido de estos

objetos refiérase a la sección correspondiente del capitulo 3.

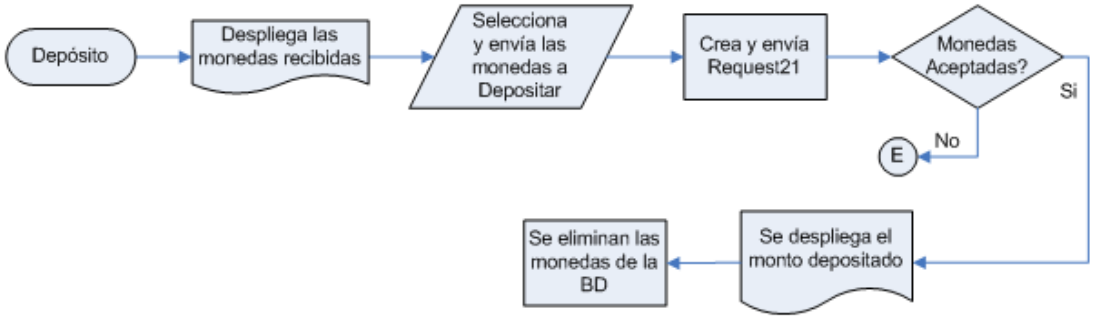

Figura 4.23: Proceso que realiza el submódulo de depósito

Submódulo de depósito En la figura 4.23 se puede observar el diagrama de flujo que describe el proceso que realiza el submódulo de depósito, el cual, es llevado a cabo de la siguiente manera por la entidad *vendedor*: primero el administrador de la tienda virtual, una vez autenticado como tal, solicita el proceso de depósito, el *vendedor* despliega entonces las monedas que ha recibido, el administrador selecciona las monedas a depositar y la entidad *vendedor* crea y envía a la entidad *banco* un objeto Request21 por cada una de las monedas a depositar. Si el *banco* acepta la moneda devuelve el identificador y la denominación de la moneda aceptada, para que al final el *vendedor* despliegue el monto aceptado por el *banco* y los identificadores de las monedas no aceptadas. El contenido del objeto Request21 puede variar dependiendo del protocolo con el que se este trabajando.

4.4.3. El *comprador*

Como ya se dijo esta entidad tiene dos facetas: la que se realiza en una PC y la que se realiza por un PDA. Sin embargo, las dos tienen el mismo objetivo y su operación y funcionamiento es muy similar, por lo que el diagrama a bloques presentado en la figura 4.24 corresponde a las dos facetas de esta entidad.

En el apéndice A se muestra con mayor detalle los servlets y las clases que componen cada uno de los submódulos de la entidad *comprador*. El funcionamiento de cada uno de los submódulos que componen a la entidad *comprador* se revisa a continuación:

Submódulo de retiro. El submódulo de retiro es utilizado cuando el *comprador* requiere descargar monedas electrónicas del *banco*, para ello se ejecuta el proceso que se muestra en el diagrama de flujo de la figura 4.25. Allí, lo primero que se realiza es la autenticación dentro del sistema utilizando un nombre de usuario y contraseña correctos. Una vez autenticado, el *comprador* procede a crear y enviar al *banco* un objeto Request00, el cual, será el encargado de iniciar el proceso de retiro. El *banco* responderá con un objeto Response00. Si toda va

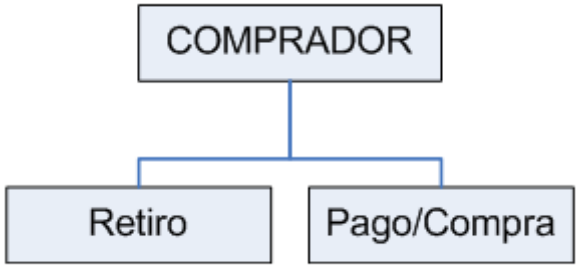

Figura 4.24: Diagrama a bloques de la entidad *comprador*

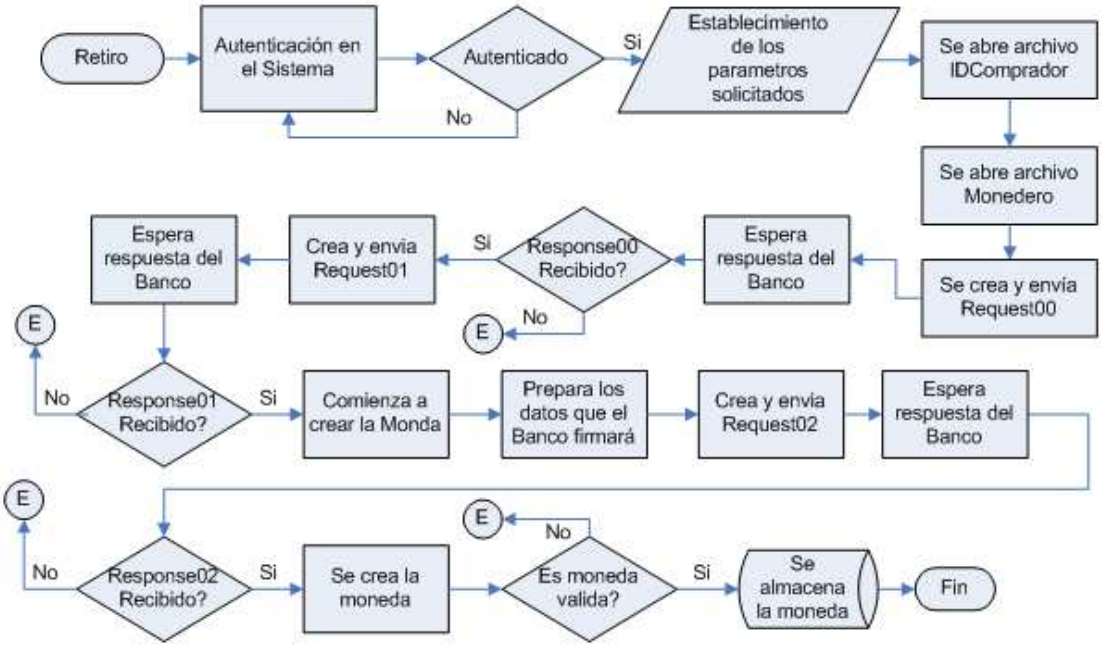

Figura 4.25: Proceso que realiza el submódulo de retiro

bien, el *comprador* crea y envía un objeto Resquest01 por cada moneda solicitada en donde se especifica la denominación de la moneda que se pretende descargar además de otros datos que posteriormente servirán para la creación de la moneda. Si no hay ningún problema, el *banco* responde con un objeto Response01. Al recibir este objeto el *comprador* puede empezar a crear la moneda. Una vez hechos los cálculos necesarios (de acuerdo al protocolo de DE que se esté utilizando) el *comprador* crea y envía un objeto Request02 el cual será procesado por el *banco* para generar la firma de la moneda. La cual, será devuelta al *comprador* mediante un objeto Response02. Al recibir este objeto el *comprador* verifica que la firma realizada por el *banco* sea la correcta y corresponda a la moneda que el *comprador* generó. Finalmente si todo va bien, el *comprador* almacena la moneda en el archivo monedero que en un principio se especificó. La composición de los objetos Request y Response están determinados por el

protocolo de dinero electrónico que se este usando en el momento.

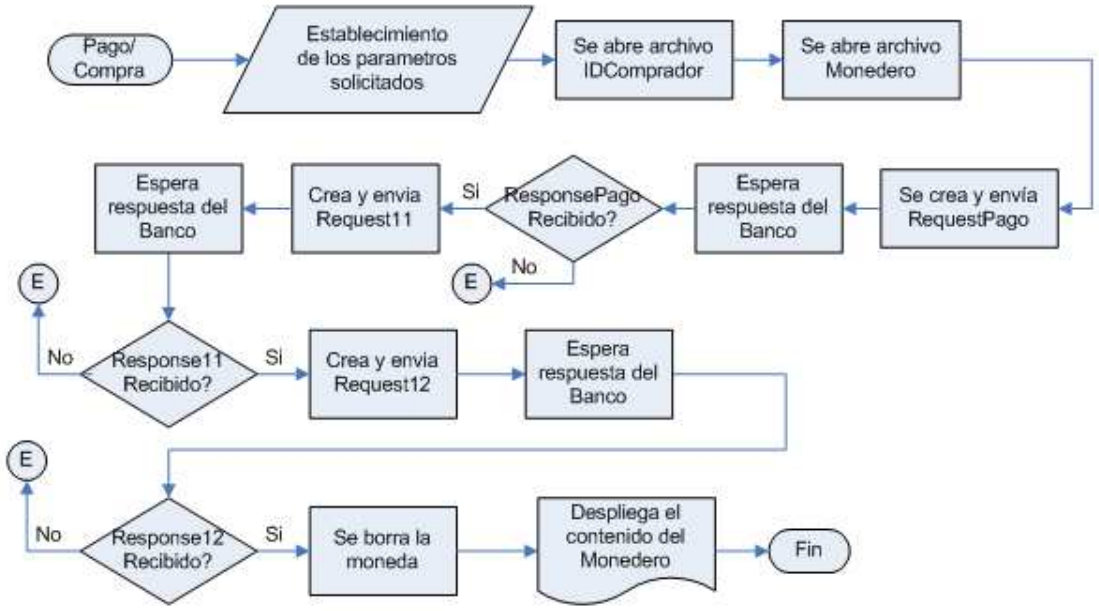

Figura 4.26: Proceso que realiza el submódulo de pago/compra

Submódulo de pago/compra. El submódulo de pago/compra es ejecutado cuando el *comprador* realiza el pago de un pedido que ha sido realizado en una tienda virtual, utilizando monedas electrónicas, dicho submódulo, implementa el proceso que se describe en el digrama de flujo que se muestra en la figura 4.26. En el diagrama de flujo se puede apreciar que primero se debe crear y enviar un objeto RequestPago, en donde estarán encapsulados los datos referentes al pago por realizar. Si todo va bien, el *comprador* crea un objeto Request11 en el cual el *comprador* encapsula las monedas electrónicas en dicho objeto junto con los datos para efectuar la verificación de las mismas. Al recibir el objeto Request11, el *vendedor* valida las monedas, si éstas resultan válidas, le pide un último reto al *comprador* enviándole un objeto Response11, si el *comprador* es el verdadero propietario de las monedas entonces creará y enviará un objeto Request12, el cual es validado por el *vendedor*. Si es correcta la validación, el pago con las monedas electrónicas es realizado y le es devuelto al *comprador* un objeto Response12 en donde se le hace saber si el pago fue aceptado o no. Los datos que conforman los objetos Request y Response para este proceso dependen del protocolo que se esté utilizando, para un mayor detalle del contenido de estos objetos refiérase a las secciones 3.2.3, 3.3.2 3.4.2 en donde se detalla cada uno de los protocolos implementados y se describe a mayor detalle el contenido de cada uno de los objetos utilizados.

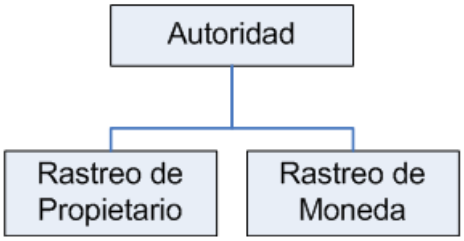

Figura 4.27: Diagrama a bloques de la entidad *autoridad*

4.4.4. La *autoridad*

La *autoridad*, cuenta con un único submódulo, el cual, realiza los procesos de rastreo de propietario y de rastreo de monedas. En la figura 4.27 se puede ver el diagrama a bloques de la aplicación WEB de esta entidad.

En el apéndice A se muestra con mayor detalle los servlets y las clases que componen el modulo de rastreo de la entidad *autoridad*.

El funcionamiento del submódulo que componen a la entidad *autoridad* se revisa a continuación:

Submódulo de rastreo. La entidad *autoridad* con su submódulo de rastreo realiza el proceso que se puede observar en el diagrama de flujo de la figura 4.28, en donde, primero se verifica si la dirección IP del *banco* está autorizada para realizar un proceso de rastreo. Después establece que tipo de rastreo será, si es un rastreo de propietario debe recibir un objeto Request31 y si es un rastreo de moneda debe recibir un objeto Request41, de acuerdo a la solicitud realizada el objeto de entrada es procesado para así obtener el dato buscado, encapsularlo en el objeto Response correspondiente y finalmente enviarlo como respuesta a la entidad *banco*, la cual con dicha información podrá obtener ya sea la identidad del propietario de la moneda o bien la tienda en la que fue gastada la moneda.

4.5. Detalles de la implementación

La implementación realizada tiene las siguientes peculiaridades:

- Todas las entidades servidor fueron realizadas usando la tecnología Servlets de Java [31].
- Todas las entidades servidor fueron implementadas en maquinas pentium III con 128MB de RAM y una comunicación a internet de alta velocidad.

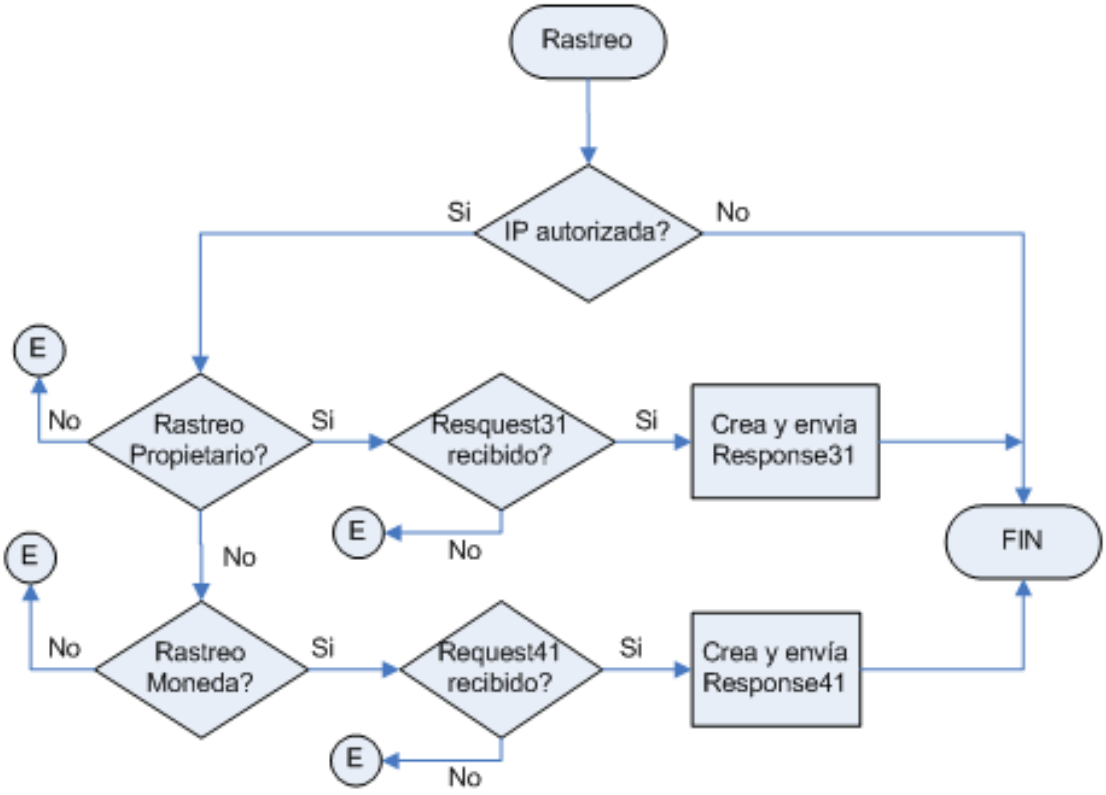

Figura 4.28: Proceso que realiza el submódulo de rastreo

- Como servidor WEB se utilizó Apache Tomcat en su versión 5.5.2
- Como capa de comunicación segura se utilizó la proporcionada por el mismo servidor WEB.
- Como manejador de base de datos se utilizó MySQL en su versión 1.4.
- Como entorno de desarrollo para las PC´s se utilizó el J2SDK 1.5.2_02.
- Como entorno de desarrollo para las PDA´s se utilizó el J2SDK 1.4.2_02 complementado con el entorno PersonalJava [32].
- La aplicación realizadas para PDA´s son ejecutadas con la maquina virtual de Java de Jeode[33].
- La PDA en la que se probó el sistema es una SHARP Zaurus SL-5600.
- Para la implementación de la entidad *comprador* en PDA´s se desarrolló una aplicación en Java, mientras que para las PC´s se crearon applets firmados.

4.6. Resultados de la implementación

Como resultado de la implementación realizada, a continuación se muestran las gráficas de tiempos tomados en los procesos de retiro de los protocolos implementados, ya que este proceso es el que requiere de mayor computo, utilizando como entidad *comprador* una PDA SHARP Zaurus SL-5600 y como entidad *banco* una maquina pentium III con 128 MB de RAM.

En las siguiente gráficas no se debe perder de vista que un agregado a nuestra implementación es el manejo de un canal seguro de comunicación entre las entidades utilizando el protocolo TLS. Esto implica logicamente un tiempo extra al inicio de la comunicación. De ahí que se pueda observar un desfase constante en las gráficas al inicio de cada uno de los procesos de retiro implementados.

En la figura 4.29 se muestra la gráfica comparativa en tiempo de los tres protocolos implementados en el proceso de retiro, utilizando una llave de 128 bits.

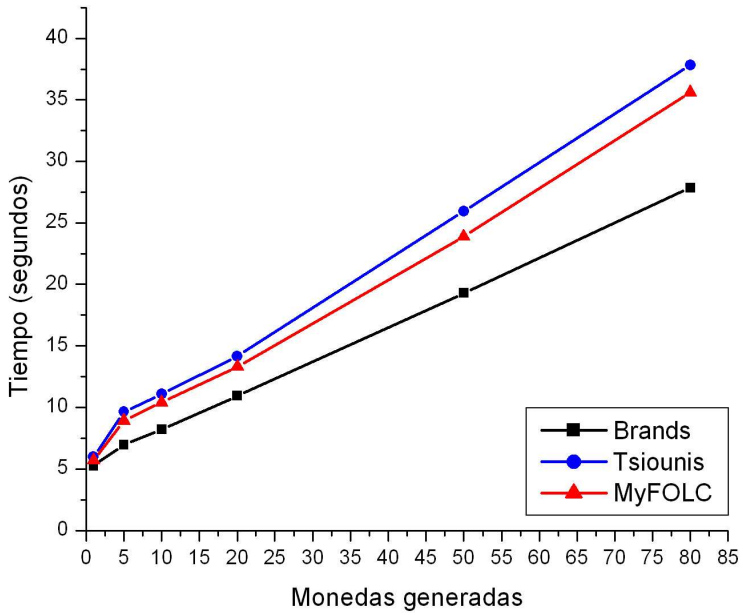

Figura 4.29: Gráfica de tiempos del proceso de retiro para una llave de 128 bits

En la figura 4.30 se muestra la gráfica comparativa en tiempo de los tres protocolos implementados en el proceso de retiro, utilizando una llave de 256 bits.

Finalmente en la figura 4.31 se muestra la gráfica comparativa en tiempo de los tres protocolos implementados en el proceso de retiro, utilizando una llave de 512 bits.

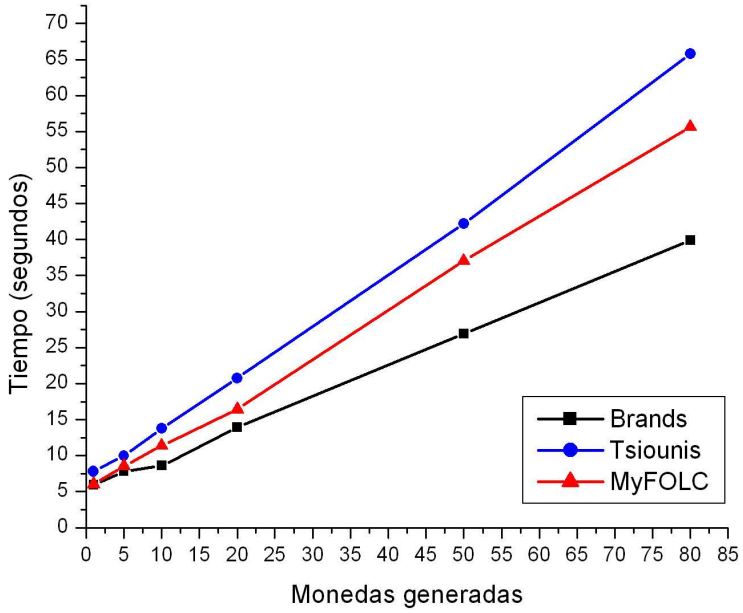

Figura 4.30: Gráfica de tiempos del proceso de retiro para una llave de 256 bits

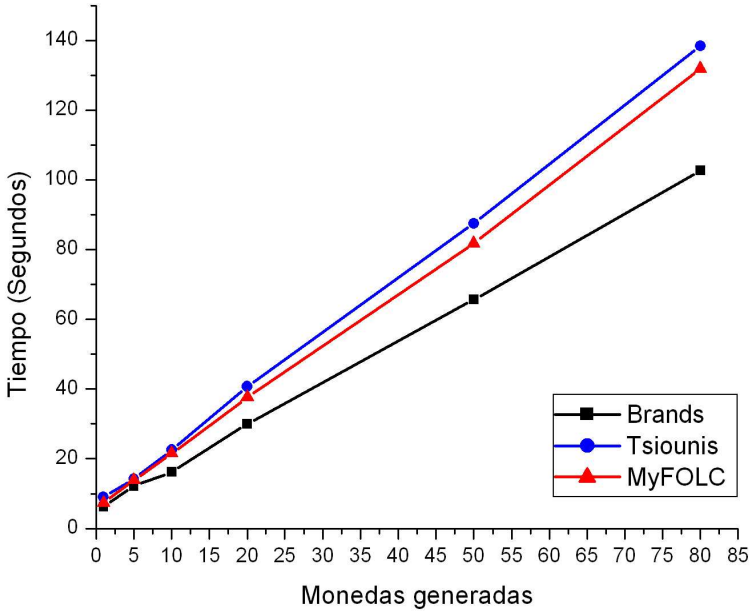

Figura 4.31: Gráfica de tiempos del proceso de retiro para una llave de 512 bits

Capítulo 5

Otras aplicaciones del DE

Como se vio en el capítulo 2, los sistemas de dinero electrónico comenzaron a desarrollarse desde hace más de dos décadas, a raíz de la necesidad de posibilitar el cumplimiento de las obligaciones monetarias a través de redes y sin proceder al traslado material de dinero tradicional (billetes y monedas de curso legal).

A medida que el dinero electrónico (DE) se ha popularizado, más y más bienes y servicios de comercios electrónicos pueden ser pagados con él, desgraciadamente aún no se da el salto para hacer del dinero electrónico algo portable y de aceptación universal. Con el tiempo, y en la medida en que el dinero electrónico se acerque a las prestaciones del dinero común en cuanto a facilidad de uso, rapidez, aceptación en comercios, anonimato y conveniencia, monedas y billetes tenderán a desaparecer en el futuro, como han desaparecido ya las conchas y dientes de animales como moneda de cambio de antaño.

Hoy en día existe una gran variedad de protocolos ya en funcionamiento, otros en pruebas y aún otros más en fase de diseño. En la actualidad nos encontramos en el proceso de investigar, probar y rediseñar todas aquellas propuestas que intentan dar solución al problema del DE. Cuando se disipen los restos de la explosión de actividad de investigación y experimentación con distintas formas de dinero electrónico, habrán sobrevivido los sistemas mejor adaptados a las necesidades que exige el mundo actual. Aunque todavía escucharemos por algunos años más el agradable tintineo de las monedas, ya se está sembrando la simiente de una Sociedad de la Información en la que el dinero electrónico (DE) sea la moneda de cambio, mientras tanto, ¿Por qué no tomar las investigaciones realizadas y aplicarlas en problemas prácticos y no tan ambiciosos, como lo es un sistema financiero completo?

Si vemos al dinero electrónico como algo más que un sistema financiero/bancario, como el que se presentó en el capítulo 4, es posible vislumbrar un abanico de aplicaciones en las cuales el dinero electrónico (DE) tiene cabida, aplicaciones en las que se hace necesario el intercambio de un token por un bien o servicio, pero sin que se involucre necesariamente a la entidad financiera, en donde tal vez una franquicia pueda generar sus propias monedas o vales electrónicos, dándoselas a sus empleados como una prestación laboral, para que ellos puedan disponer de ese dinero electrónico y gastarlo en los productos de la misma tienda; aplicaciones como esta son aplicaciones creativas que se le pueden dar al dinero electrónico.

A continuación presentaremos algunas de las aplicaciones en las cuales es posible utilizar el dinero electrónico (DE) como una solución creativa, práctica y eficiente, adaptando los protocolos de dinero electrónico existentes, para ello en la sección 5.1 hablaremos un poco de las plataformas en las que se ha implementado los sistemas de dinero electrónico en la actualidad, para después en las secciones posteriores ver los tipos de sistemas en los que el uso del dinero electrónico también tiene cabida.

5.1. Plataformas que permiten implementar sistemas DE

El establecimiento de la plataforma para la que es desarrollado un sistema es fundamental, ya que de ella depende mucho los alcances y las limitaciones del sistema. En la actualidad el dinero electrónico tiene básicamente tres plataformas de desarrollo: en tarjetas inteligentes, en dispositivos móviles y en computadoras de escritorio. En esta sección revisaremos cada una de ellas.

5.1.1. Tarjetas inteligentes (SmartCards)

Físicamente, una tarjeta inteligente tiene la forma de una tarjeta de crédito normal, con la diferencia de que posee un microprocesador incrustado en la parte frontal de la tarjeta. Este circuito integrado almacena una variada cantidad de información con sofisticados métodos de seguridad. El integrado de la tarjeta inteligente es un microcontrolador, este dispositivo es el que le da a la tarjeta la capacidad de realizar una serie de operaciones computacionales, de almacenar información y poder tomar decisiones [34].

El circuito integrado de la tarjeta tiene la capacidad de comunicarse con el mundo exterior de diferentes formas, esta comunicación le permite intercambiar información con dispositivos como puntos de venta (POS), cajeros automáticos (ATM), o lectores de tarjetas inteligentes conectados a teléfonos, computadoras etc. La finalidad de la comunicación varía dependiendo de la aplicación para la que se este usando la tarjeta inteligente [34].

En la actualidad la aplicación de las tarjetas inteligentes es muy amplia, desde tarjetas de teléfonos, tarjetas de seguros, tarjetas para entrar en compañías y tener acceso a áreas restringidas, tarjetas universitarias que contienen información del estatus de cada estudiante con respecto a los servicios universitarios, tarjetas de crédito, tarjetas de débito y las recientes tarjetas de dinero electrónico (TDE), estas últimas creadas en 1992 gracias al consorcio EMV [35] integrado por Europago, Master Card y VISA (las tres más grandes organizaciones de pago). Este consorcio se creó 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 [36].

Más de diez años del uso de las tarjetas inteligentes como monederos electrónicos hacen que esta plataforma se encuentre lo suficientemente madura para la implementación de siste-

mas de este tipo. Aunque las limitaciones que presentan son muy marcadas lo que implica que no todos los protocolos de dinero electrónico pueda ser implementados en esta plataforma.

5.1.2. Dispositivos móviles ligeros

Los constantes avances tecnológicos han permitido la creación de una gran variedad de dispositivos móviles ligeros equipados con un cada vez mayor poder de cómputo. Como ejemplos tenemos a los asistentes personales digitales (PDA's) y a los teléfonos celulares de tercera generación.

El auge de estos dispositivos en la actualidad ha dado lugar a nuevos escenarios, los cuales, se caracterizan por el hecho de que ahora el usuario se puede trasladar junto con los servicios computacionales que le son útiles para el desarrollo de alguna tarea que antes implicaba el tener una infraestructura computacional preestablecida.

El uso de teléfonos celulares de tercera generación como monedero electrónicos se remonta al año 2004 cuando por primera vez surgió una aplicación comercial en Japón, la empresa responsable se llama NTT DoCoMo y llamó a su sistema "Edy e-Money System", con dicho sistema se pretendía crear en un teléfono celular un monedero electrónico de prepago o de postpago, utilizando para ello una combinación de la tecnología de una tarjeta inteligente de múltiple aplicación con un teléfono celular [37].

Recientemente también se está tomando en cuenta a las PDA´s como plataformas para implementar monederos electrónicos, tal es el caso de esta tesis en donde se implementan algunos protocolos en una PDA utilizando la tecnología de Java para dispositivos móviles. Aunque muchos de los protocolos comerciales de dinero electrónico fueron desarrollados para PC´s, muchos de estos han comenzando a trasladar sus sistemas para que sean soportados por PDA´s o bien se ha comenzado a desarrollar protocolos específicos para PDA's como se muestra en [21].

El desarrollo de sistemas DEM para dispositivos móviles tiene ciertas limitantes, estas son variables dependiendo del dispositivo móvil que se esté utilizando. Normalmente la característica que limita al sistema es la capacidad de almacenamiento.

5.1.3. Computadoras de escritorio

Es obvio que los protocolos de dinero electrónico en una primera instancia fueron desarrollados para esta plataforma. Siendo el primer objetivo de todas las empresas comerciales que proveen de sistemas de dinero electrónico, el facilitar y darle mayor eficiencia a las transacciones, reforzar y sostener el poder adquisitivo en Internet garantizando el anonimato del comprador.

Tal vez la empresa más conocida, ya que fue la precursora del dinero electrónico es DigiCash. DigiCash es una empresa radicada en Amsterdam y creada David Chaum, reconocido

experto en criptografía. El aporte de Digicash al comercio electrónico es un producto denominado Ecash. Ecash está diseñado para realizar pagos seguros entre computadoras, ya sea por correo electrónico o Internet [38].

Actualmente las empresas de dinero electrónico desarrollan sus propios algoritmos de dinero electrónico, y las implementaciones normalmente trabajan bajo ambientes WEB o bien bajo el esquema cliente-servidor utilizando aplicaciones realizadas por las mismas empresas.

Hoy en día los sistemas desarrollados e implementados en esta plataforma no tienen limitante alguno pues las capacidades que presentan las PC´s son suficientes para hacer que los sistemas de DE no tengan limitante alguna.

5.2. Sistema de recompensas

Los programas de recompensas son una serie de beneficios adicionales, que los clientes leales reciben por utilizar los productos o servicios de una empresa en específico. Un programa de recompensas es una de las más poderosas herramientas de mercadotecnia que un negocio puede tener. A diferencia de las efímeras promociones de temporada, un programa de lealtad eslabona toda una serie de estrategias secuenciales que permiten tocar las fibras sensibles de los clientes, quienes establecen lazos duraderos con la empresa al afiliarse a estos beneficios.

Normalmente todo mundo ha oído hablar de los sistemas de recompensas de tarjetas de crédito tan famosos como los de American Express, en donde por el uso de su tarjeta de crédito en alguna compra, el usuario de dicha tarjeta obtiene una recompensa en un número determinado de puntos, los cuales, se abonan a una cuenta especial que irá almacenando la cantidad de puntos que se acumulen. Cuando el comprador asiduo obtiene cierta cantidad de puntos puede cambiar éstos desde boletos de cine, hasta paquetes de viaje, boletos de avión, etc. De esta manera el *vendedor* garantiza que su producto o servicio sea consumido por los clientes quienes efímeramente ven una ventaja más al usar dicho producto o servicio.

Desgraciadamente el manejo de la cuenta de puntos y de cobro de los mismos conlleva al manejo de una cuenta extra, lo cual, trae consigo un proceso extra que, al menos para el usuario, es una pérdida de tiempo pues sus "puntos"aunque están ahí y los puede gastar sin ningún problema, en la realidad el tiempo para intercambiar esos puntos por la recompensa deseada tarda en ser llevada a cabo. Lo ideal sería que el usuario al hacer la compra y obtener sus "puntos"pudiera en cualquier momento tomar esos puntos y llevarlos a una tienda para poder gastarlos en el producto que éste desee. Para lograrlo es posible adaptar un sistema de dinero electrónico para satisfacer las necesidades de un sistema de puntos de recompensa y lograr con ello una mayor versatilidad, sobre todo, en el cambio de esos puntos por un producto o servicio deseado por el cliente de manera casi inmediata.

A continuación se definen los procesos más importantes que se realizan en un sistema de recompensas común:

- Asignación de puntos: Este proceso se realiza cuando el *comprador* compra o usa alguno de los productos o servicios del *vendedor*, cuando ocurre esto el *vendedor* le asigna una

cantidad de puntos a una cuenta especial del comprador dependiendo del monto de la compra realizada.

- Canje de puntos: Este proceso se realiza cuando el *comprador* tiene una cierta cantidad de puntos y desea canjearlos por un producto o recompensa ofrecida por el *vendedor* a cambio de un número determinado de puntos.

Una propuesta para que un sistema de dinero electrónico pueda ser adaptado para que funja como un sistema de recompensas de puntos se presenta enseguida. Adaptando un sistema de DE a este tipo de sistemas para ello tendríamos que el modelo básico presentado en la sección 2.6.1 quedará modificado de la siguiente manera: la entidad *banco* ahora será denominada como la entidad *vendedor*, y será la encargada de generar en lugar de monedas electronicas, vales electrónicos; la entidad que anteriormente se denominaba *vendedor* ahora será la entidad Canjeador y será la encargada de recibir y validar los vales electrónicos para cambiarlos por algún producto o servicio, la figura 5.1 muestra como quedaría alterado el esquema básico de DE, el cual queda conformado de la siguiente manera:

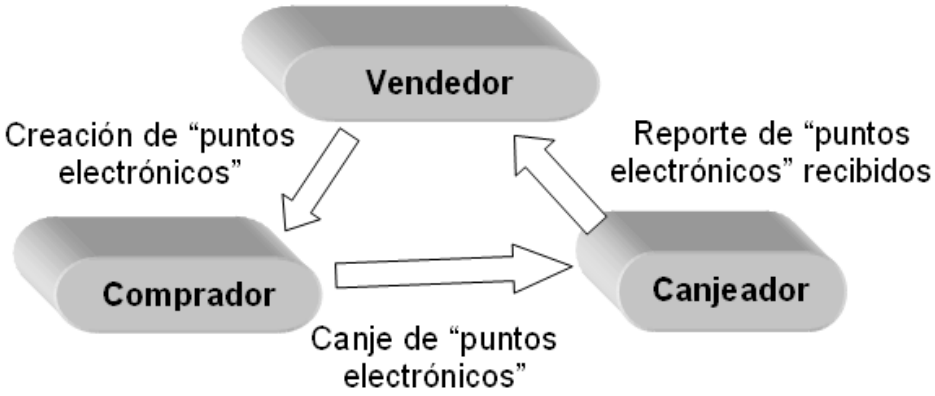

Figura 5.1: Esquema de DE modificado para sistemas de recompensas

- Asignación de Puntos. La modificación a este proceso sería en el momento de entregarle los puntos al *comprador*, ya que ahora en lugar de asignarle los puntos a una cuenta especial, el *vendedor* podría utilizar un protocolo de dinero electrónico para, en lugar de crear monedas electrónicas, crear "puntos electrónicos"y depositárselos tal vez en una tarjeta inteligente o bien en una PDA.

  En la figura 5.2 se describe el proceso de asignación de puntos utilizando un sistema de DE, en el cual, primero el *comprador* realiza una compra de algún producto o servicio; después el *vendedor* obtiene la cantidad de "puntos electrónicos"que debe retribuirle al comprador de acuerdo al monto de la compra; entonces el *vendedor* le solicitaría al *comprador* su monedero electrónico para llevar a cabo el proceso de retiro que se establece en los sistemas de DE, la única alteración en este protocolo es que el *vendedor*

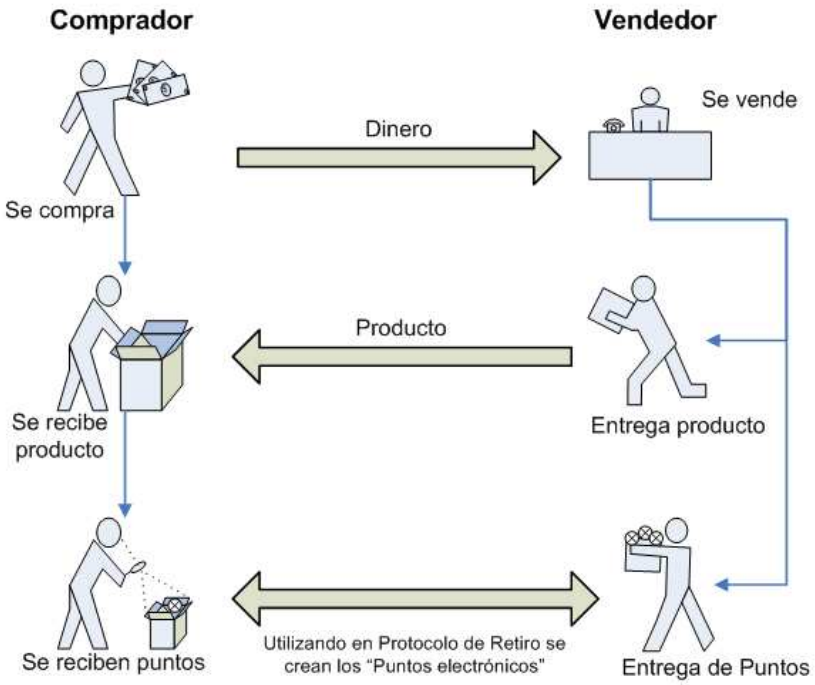

Figura 5.2: Proceso de asignación de puntos utilizando DE

sería quien inicia y realiza dicho proceso, así pues el comprador utilizando su monedero electrónico (en PDA o tarjeta inteligente) recibe y almacena los "puntos electrónicos".

- Canje de Puntos: No se necesita mucho análisis para determinar que el canje de puntos podría realizarse de la misma manera que se realiza el proceso de pago/compra en un sistema de DE. Un proceso de canje de puntos podría realizarse de la siguiente manera:

  En la figura 5.3 se describe el proceso de canje de puntos utilizando un sistema de DE, a sabiendas de que el comprador tiene en su monedero electrónico los "puntos electrónicos"tan sólo sería necesario que el comprador localizará aquel producto que desea canjear por los puntos que el posee y realizar el proceso de pago/compra establecido en el sistema de DE para canjear los puntos electrónicos por un producto o servicio. Lo interesante de aplicar un sistema de DE en sistemas de recompensas, es que los "puntos electrónicos"podrían ser cambiados por el usuario en cualquier momento y las tiendas o lo prestadores de servicio de canje no tendrían que tener acceso a la cuenta de puntos del *comprador* pues ellos sólo requieren validar los puntos electrónicos. Esto también permitiría que el anonimato del cliente quede asegurado pues el *vendedor* no sabrá jamás en donde fueron gastados los puntos otorgados al cliente. Otra cuestión interesante es que tanto la compra, la asignación de puntos y el canje de puntos pueden ser realizadas a través de un comercio electrónico o bien en un comercio establecido.

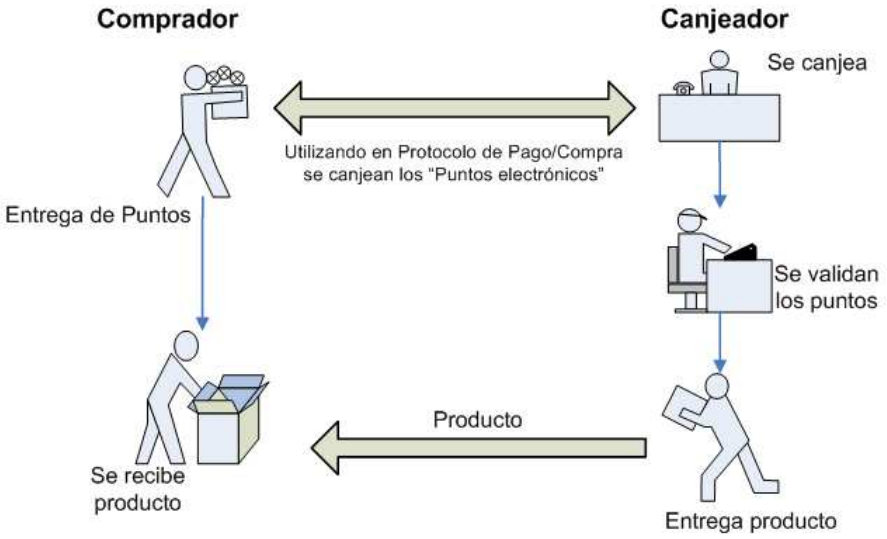

Figura 5.3: Proceso de canje de puntos utilizando DE

Estas y muchas ventajas pueden verse a manera de mercadotecnia o a manera de costo/beneficio tanto para el que ofrece el sistema de recompensas como para el que lo utiliza.

5.3. Sistemas de prepago

Actualmente la vanguardia en sistemas de pago es el prepago. Este modelo de pago es una excelente herramienta para los proveedores de servicios que apuestan a este nicho como una estrategia comercial y de mercadotecnia. A través de la diversificación, las empresas tienen la mira en el incremento de sus ingresos así como en lograr el aumento y fidelidad de su clientela.

Un modelo de prepago común y corriente consiste básicamente en hacer que el comprador pague anticipadamente por una cierta cantidad de un producto. Al realizar este pago anticipado el vendedor le debe entregar una especie de "vale"al comprador, con la cual, el comprador podrá hacer válido su pago adquiriendo el producto deseado pero en una instancia de tiempo diferente. Un sistema de este tipo tiene tres procesos principales:

- La compra de vales: En este proceso el comprador realiza un pago por adelantado de una cierta cantidad de productos o servicios y el vendedor le entrega un vale con el cual en una instancia de tiempo diferente podrá canjear dicho vale por el producto o servicio por el que se realizo previamente el pago.

- El cambio de vales: Cuando el comprador decide realizar el cambio de uno o varios vales, recurre al vendedor, le entrega el vale correspondiente, el vendedor verifica la

autenticidad y la fecha de caducidad del vale y de ser aceptado le entrega al comprador el producto o servicio correspondiente.

- El reporte de los vales recibidos: Cuando los vales son recibidos por un *vendedor* diferente al que los expidió, digamos una sucursal foránea, debe ser creado un reporte de los vales recibidos por dicho vendedor foráneo para que le sea entregado a la sucursal matriz, esto para que la sucursal matriz le proporcione un reporte completo al comprador, para que con ello el comprador verifique que los vales comprados fueron efectivamente gastados.

Los principales problemas que se presentan en este tipo de sistemas surgen cuando se requiere implementar a mediana y gran escala, esto debido a que el vendedor, debe de invertir una considerable cantidad de dinero para garantizar que sus vales tengan la suficiente seguridad impresa para evitar ser falsificados o bien que éstos sean manipulados de una forma diferente para la que fueron creados.

A continuación propondremos un sistema de prepago adaptando un sistema de DE para que funja como tal. Para ello, el esquema básico presentado en la sección 2.6.1, se ve alterado de la siguiente manera: la entidad *banco* es ahora denominada Matriz, y la entidad *vendedor* ahora sería la entidad Sucursal; en un sistema pensado a gran escala, en uno donde sólo se tiene un generador y un receptor de vales, no sería necesario separar estas entidades. En la figura 5.4 se puede observar cómo queda consituido el esquema básico modificado para un sistema de prepago.

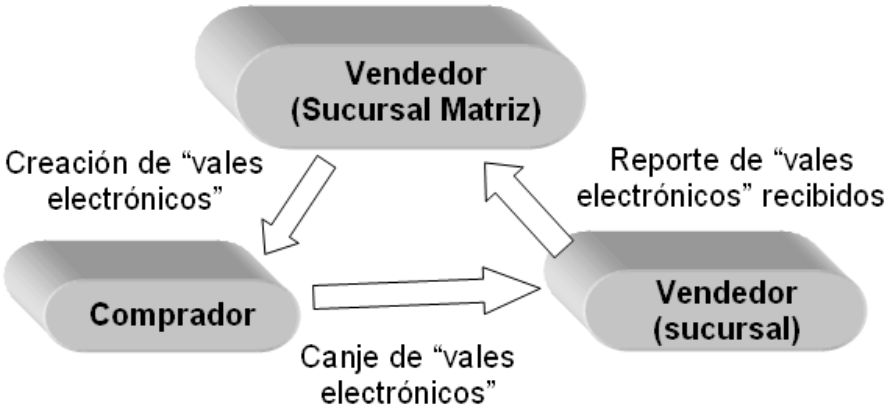

Figura 5.4: Esquema de DE modificado para sistemas de prepago

Como se puede ver en la figura 5.4 el *vendedor* (sucursal matriz) funge como el creador y el *vendedor* (sucursal) es el receptor de las monedas electrónicas validas, a las que cambiaremos su nombre por "vales electrónicos", así pues el *comprador* realiza un prepago y el *vendedor* (sucursal matriz) crea (utilizando alguno de los protocolos de dinero electrónico) un "vale

electrónico". El *comprador* recibe su "vale electrónico"y ahora podrá gastar dicho vale en otra instancia de tiempo, inclusive en otra tienda (sucursal del mismo vendedor).

Para que este tipo de sistema de prepago electrónico tenga un verdadero éxito, debe pensarse como plataforma de desarrollo en tarjetas inteligentes o bien en dispositivos móviles ligeros. Además podría pensarse en alterar los protocolos para que los "vales electrónicos"tenga fecha de caducidad, pues es como son manejados en los sistemas tradicionales de la actualidad.

A continuación se revisan tres casos específicos en los que la solución utilizando vales de prepago en papel es muy conocida, pero en donde se podría implementar un sistema de prepago adaptando un sistema de dinero electrónico, garantizando mayores condiciones de seguridad, versatilidad y un menor costo a mediano plazo.

5.3.1. Café-internet.

Es muy común que en estos negocios se venda determinado número de horas/mes por servicio de conexión pagando por adelantado. Esto brinda un valor agregado a los clientes, que pueden controlar de forma adecuada sus gastos en este tipo de lugares. Si utilizamos un sistema de DE para dar solución a este problema podríamos pensar en un sistema de la siguiente manera:

- Compra de vales. Un cliente acudiría con el proveedor del servicio y le compraría un número determinado de horas en Internet. El vendedor podría entonces proporcionar los "vales electrónicos"al cliente utilizando un protocolo de retiro de un sistema de DE, ya sea que se los deposite en una tarjeta inteligente o bien que estos vales vayan a dar a un servidor de cuentas ya sea local o externamente. Este servidor le asignaría una cuenta al usuario, en dicha cuenta se almacenaran sus "vales electrónicos"para que el usuario en cualquier momento y desde cualquier maquina del Café-Internet pueda acceder a dicha cuenta (obviamente restringida por una contraseña). En la figura 5.5 se ejemplifica de manera general lo antes mencionado.
- Intercambio de vales. Cuando se requiera hacer efectivos los "vales electrónicos", el usuario ingresaría al sistema del Café-internet para descargar los "vales electrónicos"utilizando alguno de los protocolos de pago/compra de los sistemas de DE depositaría en la maquina dichos vales, para que la maquina del Café-internet determine de acuerdo a los vales depositados el tiempo de conexión a Internet a la que tiene derecho el cliente. Este proceso puede usarse de manera análoga para proveer Internet a dispositivos móviles propios del cliente, en una área local, digamos una plaza o centro comercial. Todo lo anterior se ejemplifica en la figura 5.6.

5.3.2. Papelerías escolares

Es muy común que en Universidades públicas y privadas se les de una bonificación de material escolar a los alumnos (generalmente servicio de copias). Para ello el servicio de

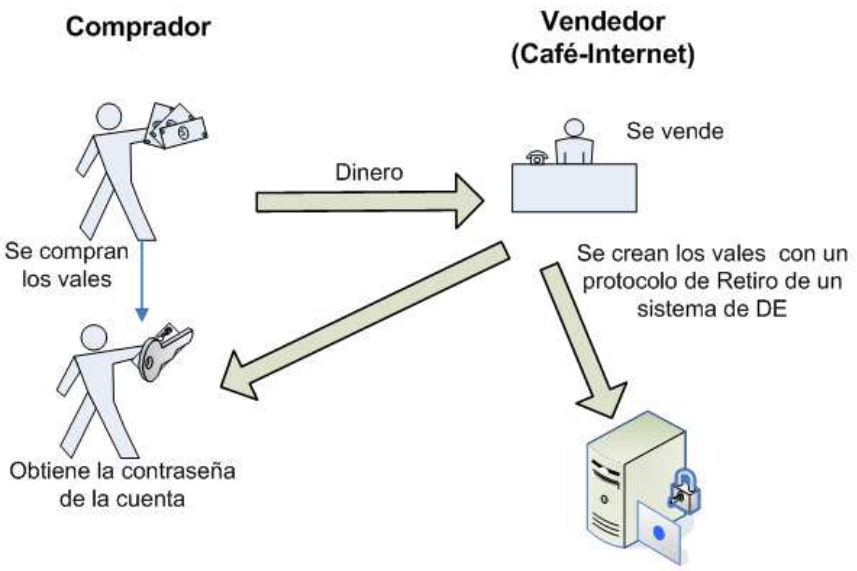

Figura 5.5: Compra de vales para un Café-internet utilizando DE

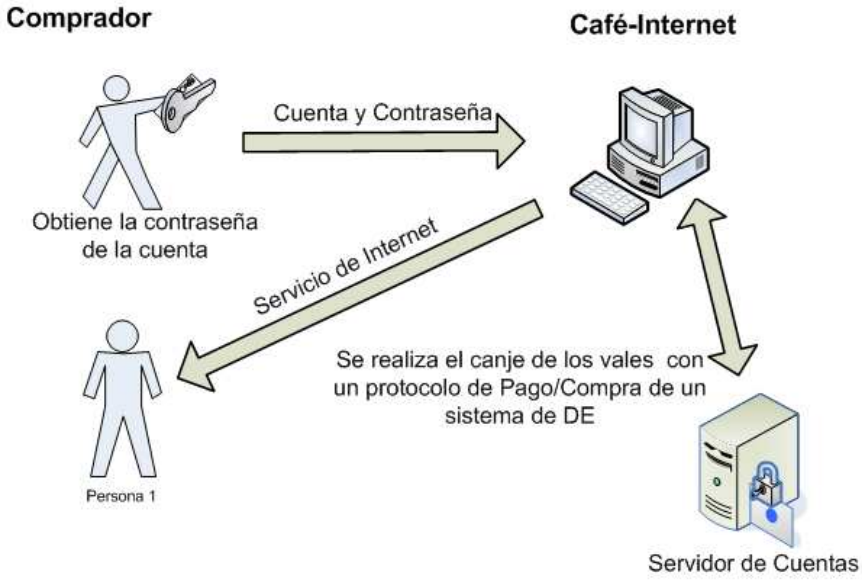

Figura 5.6: Intercambio de vales utilizndo DE

fotocopiado le proporciona al alumno un vale de copias o material escolar, al cual tiene derecho dentro de la institución educativa. Si utilizamos un sistema de DE para dar solución a este problema podríamos pensar en un sistema de la siguiente manera:

- Compra de vales. El alumno acudiría a recoger sus "vales electrónicos"los cuales serian creados y depositados en una tarjeta inteligente y dicha tarjeta le sería dada al alumno.

Tal como se ejemplifica en la figura 5.7.

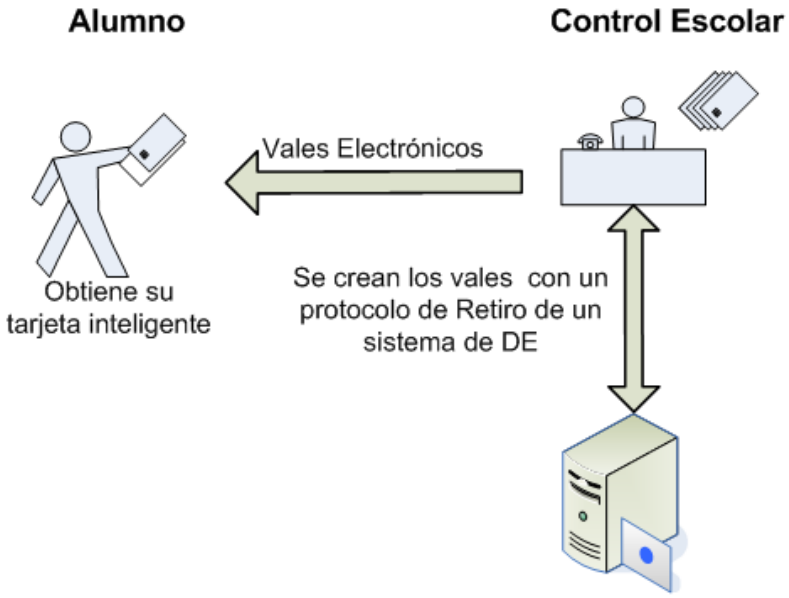

Figura 5.7: Compra de vales de copias utilizando DE

- Intercambio de vales. Cuando el alumno quiera realizar algún fotocopiado acudirá a cualquier maquina de copias del campus para ingresar su tarjeta, la maquina fotocopiadora realizaría un protocolo de pago/compra para obtener y validar los vales que se encuentran dentro de la tarjeta inteligente del alumno. Tal y como se muestra en la figura 5.8.

5.3.3. Franquicias de gasolineras.

Es posible ofrecer un servicio de prepago de vales de gasolina para que los administradores de grandes flotillas distribuyan dichos vales entre sus unidades y así durante el viaje recarguen gasolina en cualquiera de las sucursales de la misma franquicia. Este servicio permite darle un mayor control al dueño de las flotillas que desea mantener un control de los consumos de gasolina que tienen sus unidades. Si utilizamos un sistema de DE para dar solución a este problema podríamos pensar en un sistema de la siguiente manera:

- Compra de vales. Como se puede ver de manera general en la figura 5.9, en primer lugar el *comprador* (dueño de la flotilla) realiza un prepago para la compra de "vales electrónicos"de gasolina. Para ello el *vendedor* al recibir el dinero crea diferentes monederos (utilizando tarjetas inteligentes) llenándolas con vales electrónicos, de acuerdo a la distribución señalada por el *comprador*. Esto lo realiza el vendedor usando un protocolo de retiro de un sistema de DE. Al final el *vendedor* le entrega un conjunto de tarjetas inteligentes para que el *comprador* las distribuya a los operadores de su flotilla.

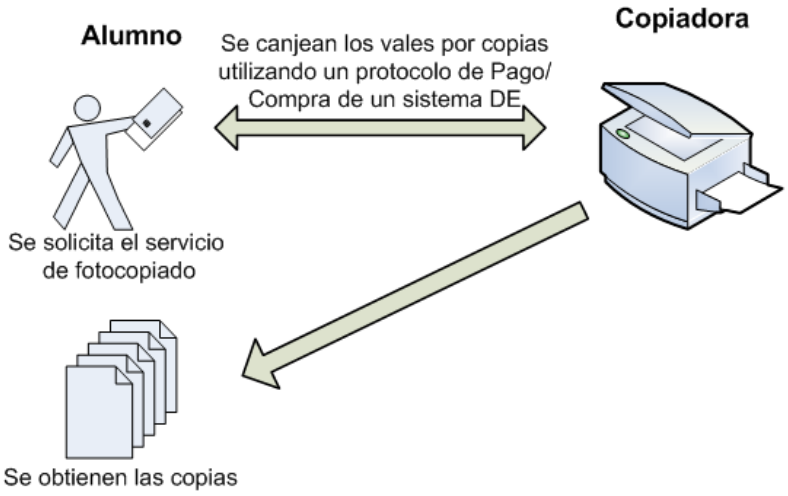

Figura 5.8: Intercambio de vales de copias utilizando DE

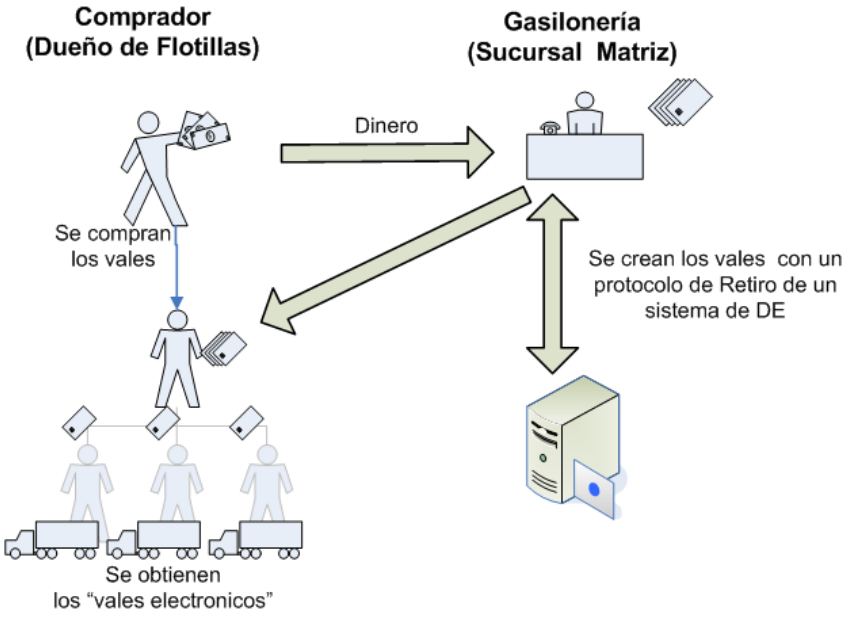

Figura 5.9: Compra de vales de gasolina utilizando DE

- Intercambio de vales. Cuando los operadores de las unidades de las flotillas, requieren cargar su unidad con más gasolina, deberán únicamente presentarse en una de las sucursales de la franquicia que acepte el "vale electrónico", para esto, la sucursal deberá contar con los medios necesarios para leer el monedero electrónico (tarjeta inteligente) y llevar a cabo un protocolo de pago/compra de un Sistema de DE, para con esto validar y recibir los "vales electrónicos"y proporcionar el combustible requerido.

Este proceso podría llevarse a cabo en cualquiera de las sucursales de la franquicia sin requerir que dichas sucursales se mantengan conectadas a la matriz para verificar la autenticidad de los "vales electrónicos".

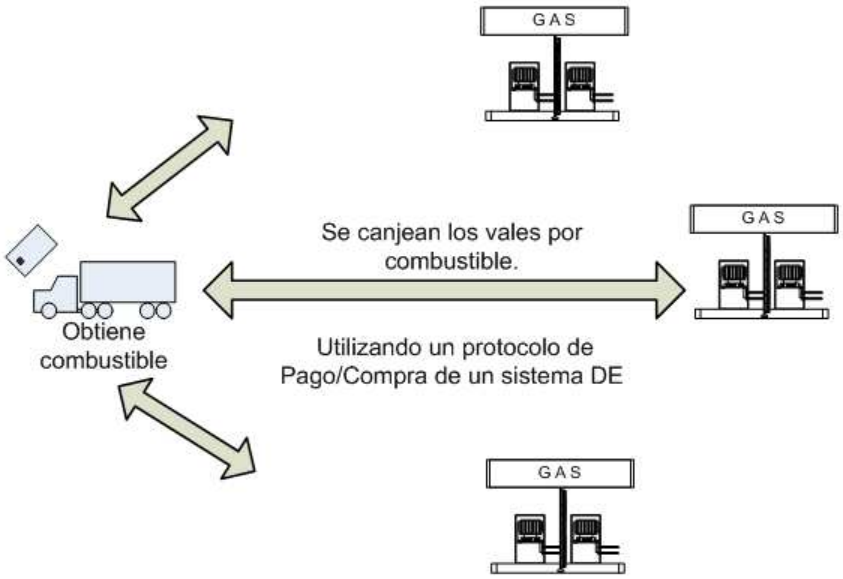

Figura 5.10: Intercambio de vales de gasolina utilizando DE

- Reporte de ventas. Como un agregado extra en este sistema es posible que mediante una combinación de los protocolos de Depósito y Rastreo de los sistemas de DE, se pueda proporcionar al *comprador* (Dueño de la flotilla) un reporte de dónde y cuándo se gastaron los vales que él compro.

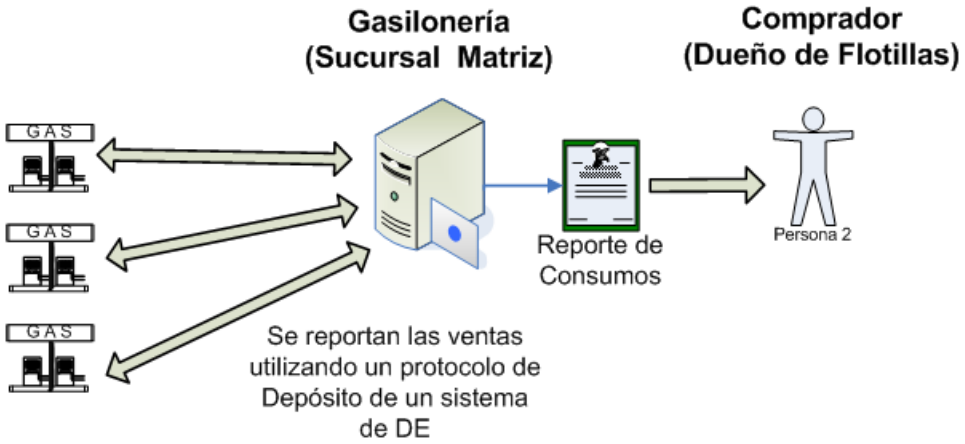

Figura 5.11: Reporte de vales de gasolina consumidos utilizando DE

Como se vió en los casos anteriores es muy factible adaptar los sistemas de dinero electrónico para que estos funcionen como sistemas de prepago, utilizando las propiedades de éstos mismos para que proporcionen ventajas que superen a los sistemas tradicionales de prepago.

5.4. Sistemas de apuestas en línea

Los sistemas de apuestas en línea, también conocidos como juegos de casino por Internet o I-apuestas, es un fenómeno reciente. Con estos sistemas, los jugadores de apuestas pueden usar sus tarjetas de crédito para apostar en sus juegos favoritos, pudiéndolo hacer ahora desde cualquier lugar sin tener que estar físicamente en algún casino.

De manera sencilla, los casinos en línea son simulaciones generadas por computadora que ofrecen apuestas en tiempo real en juegos clásicos de casino. Los sitios de casino son casi idénticos a los establecidos en un local físico, permitiendo a los clientes seleccionar un juego, comprar fichas y apostar dinero real contra la casa, o en algunos casos contra otros jugadores. Para lograrlo los casinos deben transferir la información y los fondos de forma segura y confidencial, para lograrlo utilizan sistemas similares a los que emplean los bancos a nivel mundial. Se pueden definir tres procesos generales con los cuales debe trabajar un casino en línea.

- Deposito de fondos. En este proceso el apostador deposita a su cuenta en el casino una cantidad de dinero vía tarjeta de crédito, deposito o cheque. Con dicha cuenta el apostador puede realizar sus apuestas en los juegos del casino.

- Manejo de apuestas. Cuando el apostador ingresa a algún juego virtual, utilizará el dinero depositado en su cuanta del casino para realizar la apuesta. Entonces el juego deberá consultar si su cuenta tiene los fondos suficientes para aceptar la apuesta del apostador. Si el apostador gana, el juego depositará las ganancias del apostador a la cuenta correspondiente. En caso de que pierda el monto de la apuesta será retirado de la cuenta del apostador.

- Retiro de fondos. El apostador tiene la opción de retirar, de su cuenta en el casino, el monto de su dinero depositado ahí, este proceso lo realiza el casino depositándole de nuevo al apostador en su cuenta bancaria o bien dándole un cheque.

Revisando los procesos anteriores, es fácil imaginar que un sistema de DE puede ser una solución para el manejo de cuentas de los casinos en línea. Un sistema de DE implementado para casinos en línea proveería de una mayor versatilidad y seguridad tanto para los clientes como para los casinos. En la figura 5.12 se propone un esquema para el manejo de un casino utilizando un sistema de DE, el esquema presentado es una modificación del esquema básico de DE presentado en la sección 2.6.1.

Como se puede apreciar en la figura 5.12 se cambió el nombre a la entidad *banco* por la entidad *casino* y la entidad *vendedor* fue eliminada, y quedaría contenida dentro de la misma entidad *casino*. Así mismo la entidad *comprador* ahora sería denominada como la entidad Apostador y en lugar de obtener "monedas electrónicas"se obtendrían las fichas del casino a las que denominaremos "fichas electrónicas".

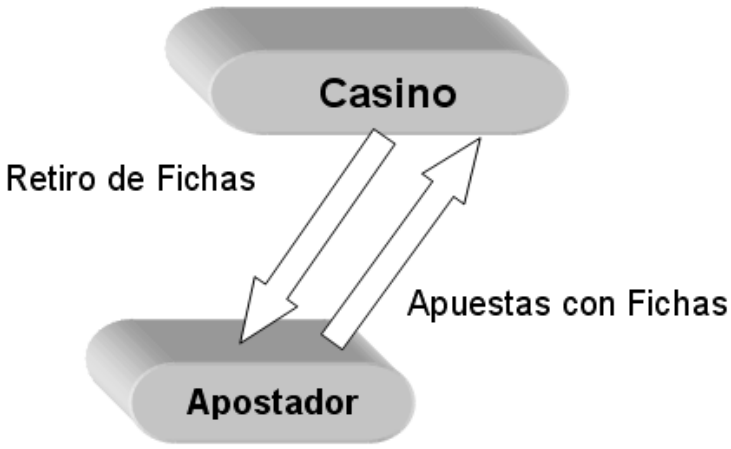

Figura 5.12: Esquema de DE modificado para sistemas de apuestas en línea

Con el proposito de mostrar de una manera más detallada cómo es posible adaptar los sistemas de DE para resolver tanto los problemas descritos en este capítulo como otros más, se desarrolló una implementación de un casino en línea sencillo el cual hace uso de un sistema de DE para el manejo de las apuestas, utilizando el esquema de la figura 5.12. En la siguiente sección describiremos de forma más detallada cómo fue la adapatción de un sistema de DE para el uso de este casino en línea y en el apéndice C puede verse la aplicación obtenida.

5.4.1. *casino* en línea

En el sistema de casino en línea implementado, por sencillez, sólo se incluyó un único juego de casino, *el blackjack*, en donde el jugador siempre esta compitiendo contra la banca. El objetivo de este juego es pedir cartas hasta que la suma de sus puntos llegue lo más cerca posible a 21, sin pasarse, el que quede más cerca de 21 gana. Si las primeras dos cartas que recibe el jugador, suman 21 tiene blackjack y automáticamente gana el juego.

Para la implementación de este pequeño casino en línea, se decidió usar como protocolo de DE el propuesto por S. Brands [6], aunque podríamos usar cualquiera de los protocolos de DE conocidos. Para mayor detalle de este protocolo vease la sección 3.2

En el casino en línea implementado adaptando un sistema de DE, pueden llevarse acabo los siguientes procesos entre las entidades:

- Activación de cuenta: Este proceso se realiza solamente una vez cuando el usuario es dado de alta por primera vez en el sistema del casino, abonando un monto inicial para la apertura de su cuenta en el casino. Consiste en el establecimiento de la contraseña de acceso al casino y de la obtención de un archivo (archivo de identificación), este archivo contiene la llave publica y privada del usuario para la generación posterior de las fichas

electrónicas, dicho archivo deberá ser generado del lado del usuario pues el casino no debe conocer la llave privada del usuario.

- Depósito y retiro de fichas electrónicas: Cuando el usuario ha realizado su activación de cuenta (estableciendo su contraseña y descargando el archivo de identificación) podrá tener acceso a un sistema WEB en el cual, se le mostrará un estado de su cuenta dentro del casino, así mismo, en dicho sistema WEB tendrá la posibilidad de hacer: retiros de fichas electrónicas para posteriormente apostarlas y jugar el juego en línea y, de igual forma, podrá hacer depósitos de las fichas electrónicas (tal vez ganadas) a su cuenta para posteriormente poder realizar, si es que así lo desea el usuario, un retiro de dinero real vía transferencia bancaria.

- Establecimiento de apuestas: Una vez que el usuario ya ha descargado las fichas electrónicas de su cuenta del casino, puede ahora con esas fichas establecer las apuestas que desee dentro de los sistemas de juego del casino en línea. Para ello el juego del casino debe contar con los mecanismos necesarios para realizar un proceso de pago/compra utilizando un protocolo de DE entre él y el apostador.

- Desarrollo del juego: Es aquí donde se lleva a cabo el juego entre la entidad Apostador en contra del *casino*. Una vez que se ha realizado una apuesta el usuario apostador comienza a jugar en contra del casino de acuerdo a las reglas de juego establecidas; si el usuario pierde pasará lo mismo con su apuesta. Si el usuario gana entonces el casino deberá contar con los mecanismos necesarios para devolverle al usuario la cantidad de fichas electrónicas que paguen la apuesta realizada con éste.

- Obtención de ganancias: Cuando el usuario apostador gana en un juego, se realiza un proceso de retiro utilizando un protocolo de DE para así obtener en ese momento las fichas electrónicas correspondientes al pago de la apuesta realizada en contra del casino por el usuario apostador.

Dado que el juego en sí (el BlackJack) queda fuera del objetivo de esta tesis, no hablaremos del desarrollo de éste y sólo nos enfocaremos a los procesos en los que se involucra el manejo de fichas electrónicas.

Arquitectura del sistema

Básicamente la entidad casino esta conformada por un manejador de base de datos, un conjunto de servlets o CGI´s (los cuales realizaran los procesos correspondientes a esta entidad), un servidor WEB y una capa de comunicación segura mediante TLS. Por otro lado, la entidad Apostador estaría constituida por un conjunto de applets (los cuales realizaran los procesos de esta entidad), un navegador WEB y una capa de comunicación segura usando TLS (esta última puede ser provista por el mismo navegador). En la figura 5.13 se puede observar, a manera de capas, la arquitectura de las entidades que conforman el sistema; de tal manera, que la comunicación interna entre cada una de las capas es llevada a cabo de manera vertical, mientras que la comunicación externa entre las entidades es llevada a cabo únicamente mediante la capa de comunicación TLS.

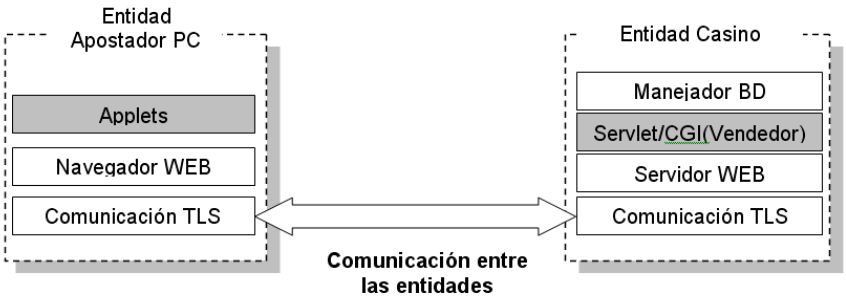

Figura 5.13: Arquitectura del casino en línea implementado usando un sistema de DE

Estructura interna del sistema

A continuación se muestra la composición interna de la capa de aplicación constituida por los applets en el caso de la entidad Apostador y por los Servlets en el caso de la entidad *casino*, mostrando los módulos y submódulos principales que conforman a cada una de las entidades del sistema.

En la figura 5.14 se muestra cómo está constituida internamente la entidad *casino*, la cual, tienes dos módulos principales: el módulo del sistema WEB y el módulo del juego en línea (en este caso el BlackJack).

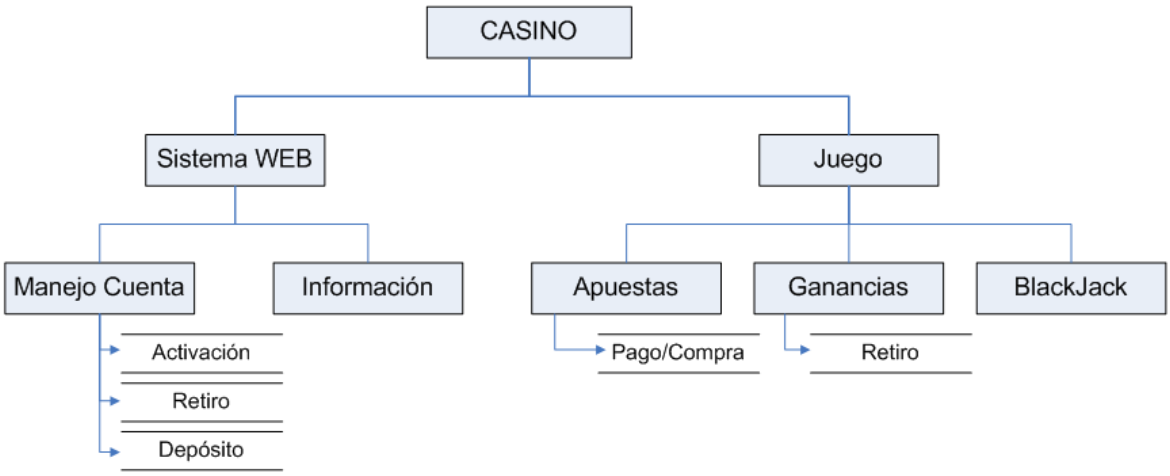

Figura 5.14: Estructura interna de los módulos que componen a la entidad *casino*

El módulo del sistema WEB a su vez está compuesto por un submódulo de manejo de cuenta y un submódulo de presentación de información. El primero básicamente será el encargado de realizar los procesos de activación de cuenta, retiro y depósito de fichas electrónicas; mientras que el segundo será el encargado de mostrar la información de las páginas referente al casino en línea.

El módulo del juego consiste básicamente de tres submódulos. El submódulo del apuestas con el que se establecerán las apuestas del usuario apostador en contra del casino, utilizando las fichas electrónicas que deberá tener en su posesión el usuario apostador. Este proceso es llevado a cabo mediante un protocolo de pago/compra de DE. El submódulo de ganancias es con el que el casino pagaría (si es que el casino perdió en el juego) a los usuarios apostadores utilizando un protocolo de retiro de un sistema de DE para que de esta forma el usuario apostador obtenga las fichas electrónicas correspondientes al pago de su apuesta. Finalmente este módulo debe contar con el submódulo de juego con el cual se desarrolla el juego en línea entre la entidad *apostador* y la entidad *casino*, en este submódulo se encuentran las reglas para determinar quién gana y quién pierde.

La constitución de la entidad interna de la entidad *apostador* se muestra en la figura 5.15, la cual, tienes tres módulos principales: el módulo del sistema juego en línea (en este caso el BlackJack), el módulo de ganancias y el módulo de apuestas.

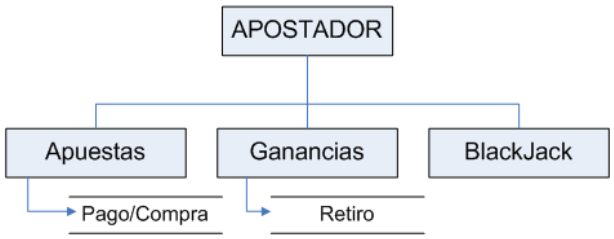

Figura 5.15: Estructura interna de los módulos que componen a la entidad *apostador*

Detalle de procesos

Los procesos realizados por el submódulo de manejo de cuenta (activación, retiro y depósito de fichas electrónicas) quedaron implementados exactamente de la misma manera que se implementarían en un sistema de DE utilizando el protocolo de DE propuesto por S. Brands, el cual es descrito en la sección 3.2. Por lo tanto, en esta sección sólo detallaremos los procesos que realizan los módulos de ganancias y de apuestas.

El diagrama de secuencias de la figura 5.16 muestra la interacción entre las entidades *casino* y *apostador* para la realización del proceso del establecimiento de la apuesta, este proceso ya fue descrito en la sección 5.4.1. Cabe mencionar que este proceso será realizado antes de que comience el juego en línea.

El diagrama de secuencias de la figura 5.17 muestra la interacción entre las entidades *casino* y *apostador* para la realización del proceso de obtención de ganancias, este proceso ya fue descrito en la sección 5.4.1. Este proceso será realizado únicamente cuando el apostador haya ganado y por lo tanto requiera cobrar las ganancias de la apuesta realizada.

Finalmente si comparamos la arquitectura y los diagramas de secuencias descritos en esta sección con los establecidos en el capitulo 4, en donde se implementa un sistema de DE, podremos notar que las alteraciones son mínimas en cuanto al comportamiento interno de los

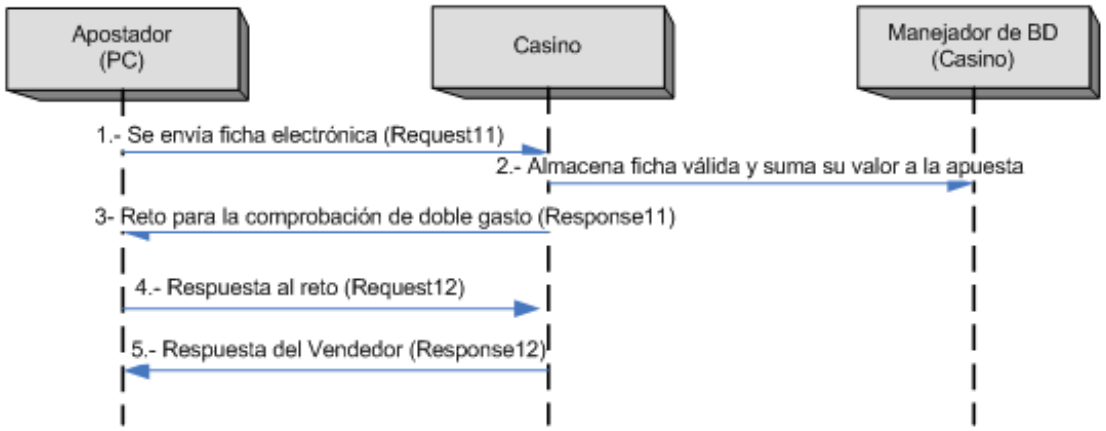

Figura 5.16: Diagrama de secuencias del establecimiento de la apuesta

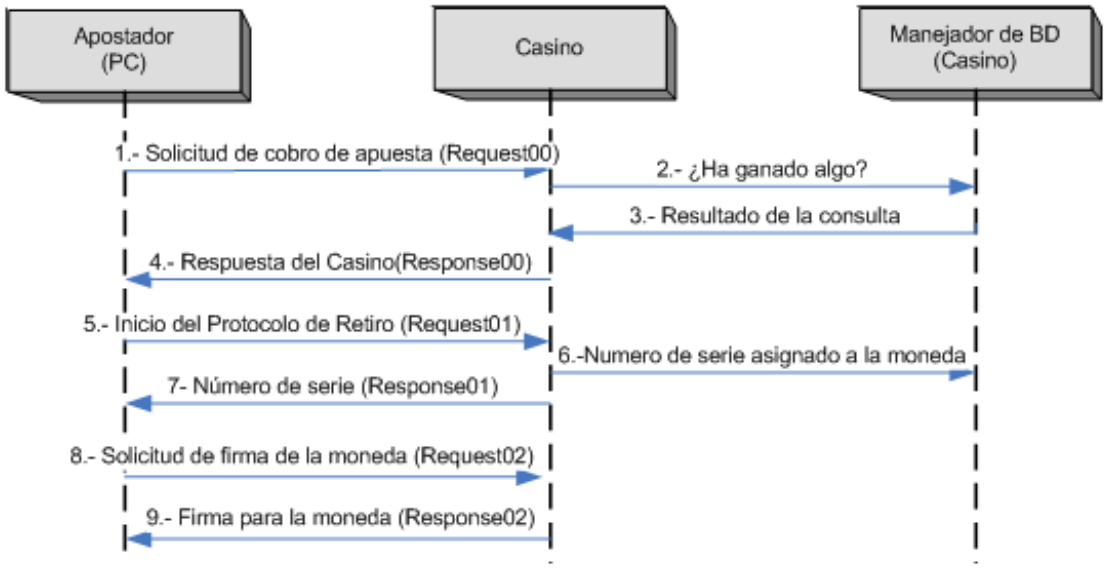

Figura 5.17: Diagrama de secuencias del proceso de la obtención de ganancias

procesos generales y que los procesos de DE utilizados para la resolución de este problema no sufren cambios muy leves en su composición interna.

Capítulo 6

Conclusiones

Dado que en la actualidad se ha comenzado a manifestar poco a poco el modelo de pago conocido como dinero electrónico y en vista de que hoy en día son varias las empresas y gobiernos que comienzan a apostar por este modelo de pago. La necesidad de proponer aquellos medios tecnológicos que podrían ser las plataformas para el desarrollo masivo de este modelo pago, resultó ser una excelente motivación para la realización de esta tesis.

Tomando lo anterior como base, en esta tesis se implementó un sistema de dinero electrónico el cual puede utilizar 3 diferentes protocolos de dinero electrónico y que además puede ser utilizado tanto por PCs como por PDAs. Utilizando el lenguaje de programación multiplataforma Java, se demostró que la implementación de un sistema creado para PCs puede ser implememententado sin demasiados contratiempos en un dispositivo móvil ligero, como lo es una PDA. El sistema de dinero electrónico móvil implementado en esta tesis cuenta con las siguientes caracteristicas:

1. Cumple con 4 de las 6 características deseables de los sistemas de dinero electrónico (independencia, seguridad, privacidad y pago fuera de línea). No cumple con la propiedad de transferibilidad y divisibilidad.

2. Puede trabajar bajo los dos esquemas principales de dinero electrónico planteados en la literatura (el básico y el FOLC).

3. La comunicación entre las entidades es llevada a cabo mediante un canal seguro utilizando el protocolo de comunicación seguro TLS.

4. Trabaja utilizando 3 diferentes denominaciones de moneda electrónica.

5. Tiene como base uno de los protocolos más eficientes de dinero electrónico, el propuesto por S. Brands [9].

Más aún, en esta tesis se plantearon algunas posibles aplicaciones que pueden dársele a los protocolos de DE planteados en la literatura, para con estos, dar una solución a diferentes problemas en los que se involucra un proceso de intercambio de algún producto o servicio por

algún *token digital*, el cual debe permitir que sea verificado únicamente por la composición del mismo, sin recurrir a una consulta directa con la entidad que emite dicho *token digital*.

Finalmente podemos puntualizar que hoy en día, los sistemas de dinero electrónico estan en la posibilidad de ser una realidad tecnológica en proceso de incorporación a los hábitos sociales y reglas económicas de las comunidades donde se implementen; sin embargo los proveedores de estos productos y/o servicios aún deben hacer concesiones mutuas, dadas las exigencias de estandarización que plantea el intercambio propio de la actividad comercial y la uniformidad requerida para expandir el mercado. Si esto no sucede, se tendrán comunidades cerradas tecnológicamente y con una limitación artificial para ejercer libremente el objetivo final: la libre actividad comercial. Posiblemente la implementación de estos sistemas de dinero electrónico en dispositivos móviles ligeros traiga consigo que su aceptación sea más rápida y al alcance de una mayor comunidad de usuarios.

Como trabajo a futuro se plantea lo siguiente:

- Realizar un estudio entre los leguajes disponibles para PDA (C/C++, Pearl, Python, etc.) para con ello implementar el sistema de dinero electrónico móvil trando de lograr un mejor desempeño que el presentado en esta tesis pero sin perder en gran medida la portabilidad para el uso del sistema en diferentes dispositivos móviles.
- Proveer de las caracteristicas faltantes (divisivilidad y transferibilidad) al protocolo de S.Brands sin afectar demasiado el buen desempeño que tiene en estos momentos.
- Ahondar dentro de lo posible en las problemas que se comentan en esta tesis, para demostrar la factibilidad de la implementación de un sistema de dinero electrónico como una solución alternativa a dichos problemas.
- Experimentar en aquellas plataformas de más bajos recursos como lo podrían ser los telefonos celulares de tercera generación.
- Buscar otras aplicaciones en las que el dinero electrónico pueda dar una solución alternativa.

Apéndice A

Estructura interna del sistema DEM

En este apéndice se revisa la composición interna del bloque de los CGI´s o Servlets, es decir las clases que conforman la parte que fue programada para el desarrollo de esta tesis; se muestra el diseño de los diagramas de clases correspondientes a cada entidad, explicando los detalles más importantes de cada una de las clases que lo conforman.

A.1. El *banco*

Como se mencionó en la subsección 4.1.1 en la pagina 49, esta entidad deberá de contar con un sitio WEB dinámico, un sitio WEB administrativo y una aplicación WEB. Todas ellas fueron programadas utilizando la tecnología Servlets de Java.

En la figura A.1 se muestran los servlets que componen a los sitios WEB que manejan el sistema. Estos sólo sirven para el manejo y la presentación de la información en paginas WEB. El sistema define tres conjuntos de servlets: los encargados de la activación de la cuenta, los encargados de la parte administrativa del *banco* en donde se realizaran la detección de fraudes y el rastreo de monedas y los encargados de la banca en línea en donde se podrá manejar una cuenta bancaria teniendo las opciones de ver el estado de cuenta, descargar monedas electrónicas y realizar transferencias a otras cuentas.

De acuerdo a la descripción del sistema que se presentó en la subsección 4.1.1, el *banco* lleva a cabo 4 procesos (retiro, depósito, rastreo y control de fraudes), para ello cuenta con otro conjunto de servlets con los cuales el *banco* podrá realizar cada uno de estos procesos, en la figura A.2 se muestra el diagrama de clases que componen a la entidad *banco* para la realización de los procesos propios de ésta entidad.

Las clases que conforman la entidad *banco* se describen a continuación:

Para el proceso de retiro:

- Request00, Request01, Request02: estas clases son utilizadas para recibir las peticiones que los clientes realizan a los diferentes servlets, en las diferentes instancias del proceso de retiro.

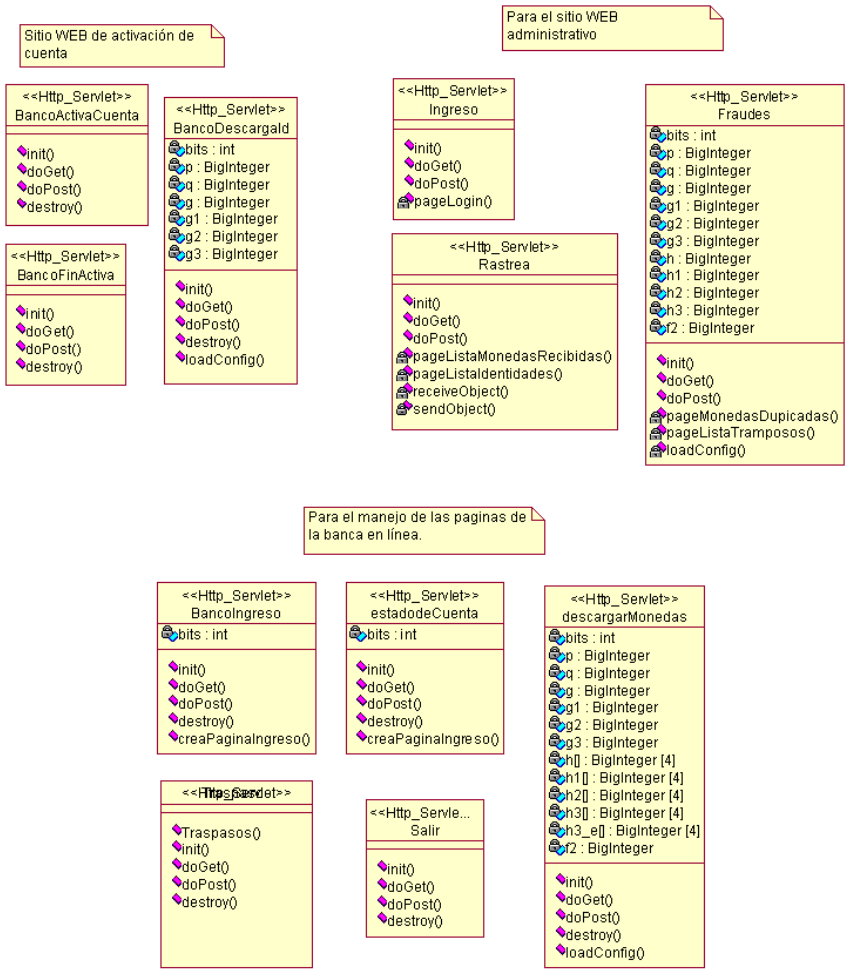

Figura A.1: Diagrama de clases del sitio WEB del *banco*

- Response01, Response02, Response03: Son los diferentes objetos con los que los servlets pueden responder después de haber procesado la petición realizada por el objeto Request correspondiente.

- BancoRetiro: Esta clase es un servlet el cual iniciará el proceso de retiro de monedas, para lo cual recibe, mediante una petición Post, un objeto Request00, el cual contendrá los datos de la cuenta del retiro, el monto del retiro, la denominación. El objetivo primordial de este servlet es validar dicha información y preparar al sistema para el proceso real de retiro, si todo está en orden devolverá un objeto Response00.

- BancoRetiroBrands: Este servlet se encarga de llevar a cabo el proceso de retiro de monedas electrónicas, mediante el protocolo propuesto por S. Brands, para mayor detalle vea la sección 3.3.

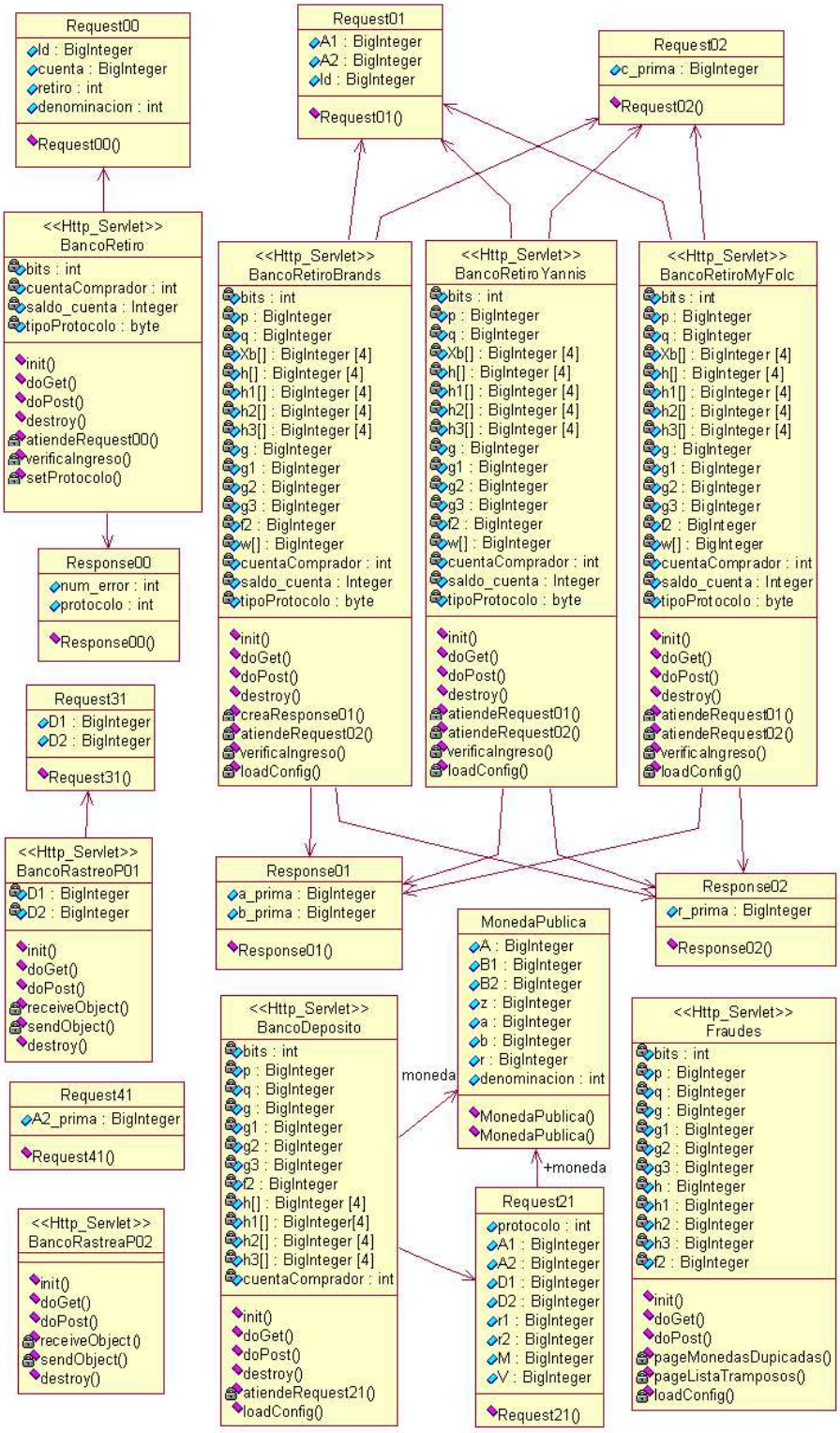

Figura A.2: Diagrama de clases que realizan los procesos de la entidad *banco*

- BancoRetiroYannis: Este servlet se encarga de llevar a cabo el proceso de retiro de monedas electrónicas, mediante el protocolo propuesto por Yannis et al., para mayor detalle vea la sección 3.4.
- BancoRetiroMyFolc: Este servlet se encarga de llevar a cabo el proceso de retiro de monedas electrónicas, mediante el protocolo propuesto en esta tesis, para mayor detalle véase la sección 3.5.

Para el proceso de depósito:

- *banco*Deposito: Esta clase es el servlet que recibirá las monedas para verificar la autenticidad de las mismas. Si la moneda es auténtica se abonará el valor de la moneda a la cuenta del depositante; en caso contrario la moneda será rechazada. Para ello está clase recibe mediante una petición WEB un objeto Request21.
- Request21: este objeto contiene dentro de sí un objeto MonedaPublica y los datos que permiten verificar la autenticidad de la misma. Los cuales no son otra cosa que los datos recibidos en el proceso de pago/compra por el *vendedor*, quien es el que ahora, intenta abonar la moneda recibida a su cuenta.

Para el proceso de rastreo:

- *banco*RastreoP01: Es el servlet encargado de iniciar el proceso de rastreo de propietario con la entidad *autoridad*. Para lograrlo este servlet cuando se invoca mediante la petición WEB despliega todas las monedas que han sido recibidas para que el usuario seleccione aquellas que quiera rastrear. Cuando se invoca nuevamente mediante otra petición WEB debe recibir como datos los valores D1 y D2 correspondientes a las monedas enviadas para así crear el objeto Request31 y enviarlo a la entidad *autoridad*.
- *banco*RastreoP02: Es el servlet encargado de iniciar el proceso de rastreo de moneda con la entidad *autoridad*. Para lograrlo este servlet cuando se invoca mediante la petición WEB despliega todas las monedas que han sido recibidas para que el usuario seleccione aquellas que quiera rastrear. Cuando se invoca nuevamente mediante otra petición WEB debe recibir como datos los valores A2', correspondientes a las monedas enviadas, para así crear el objeto Request41 y enviarlo a la entidad *autoridad*.

Para el proceso de control de fraudes:

- Fraudes: es el único servlet que actúa en este proceso, cuanto se invoca mediante una petición WEB, despliega todas aquellas monedas que fueron recibidas 2 o más veces. Posteriormente el usuario seleccionaría a partir de éstas cuales desea desenmascarar y de esta manera son enviadas de nuevo al mismo servlet. Éste realiza los cálculos necesarios para así encontrar la identidad del tramposo y dar la opción para que su cuenta sea cancelada en el sistema.

A.2. El *vendedor*

De la misma manera que el *banco*, la entidad *vendedor* debe contar con un sitio WEB dinámico, un sitio WEB administrativo y una aplicación WEB. Todas ellas programadas utilizando la tecnología servlets de Java, en la figura A.3 se muestran los servlets que componen a los sitios WEB que manejan el sistema.

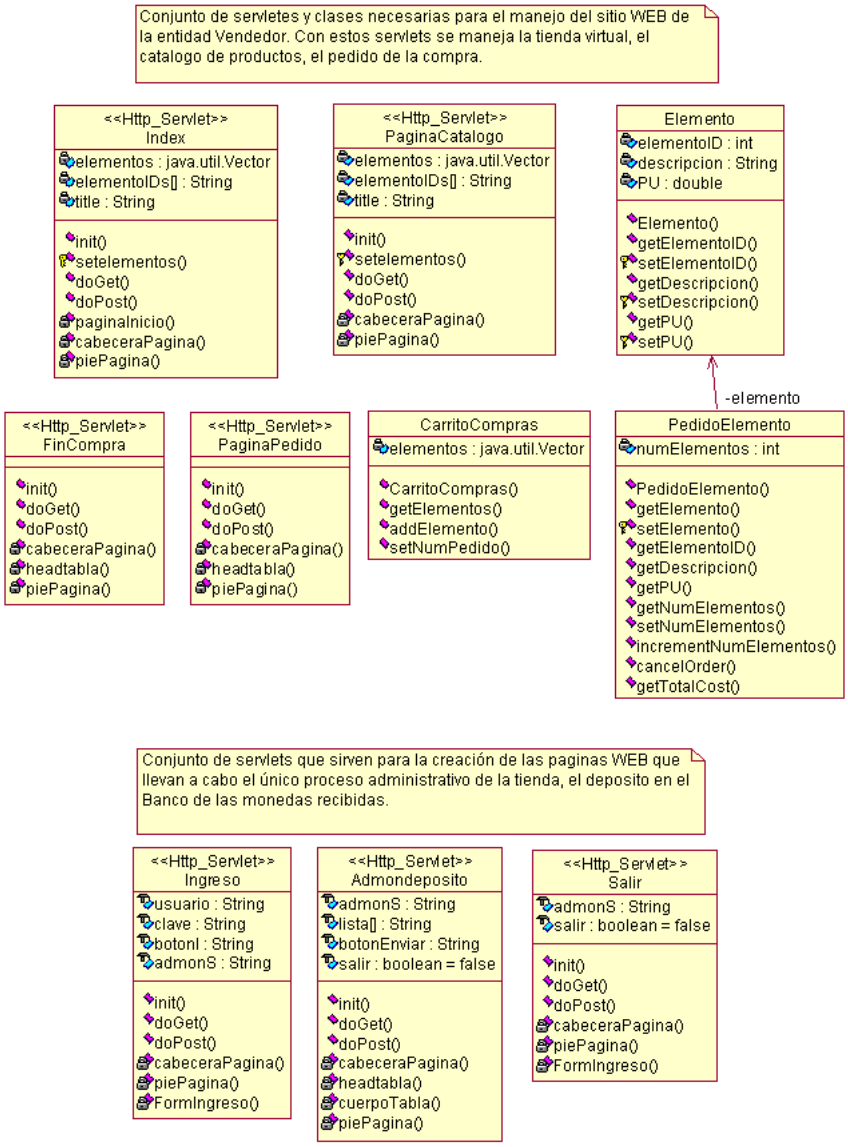

Figura A.3: Diagrama de clases del sitio WEB de la entidad *vendedor*

De acuerdo a la descripción del sistema que se presenta en la subsección 4.1.1, el *vendedor* lleva a cabo 2 procesos (Pago/Compra y Depósito), para ello cuenta con otro conjunto de servlets con los cuales el *vendedor* podrá realizar cada uno de estos procesos, en la figura A.4

se muestra el diagrama de clases que componen a la entidad *vendedor* para la realización de los procesos propios de ésta entidad.

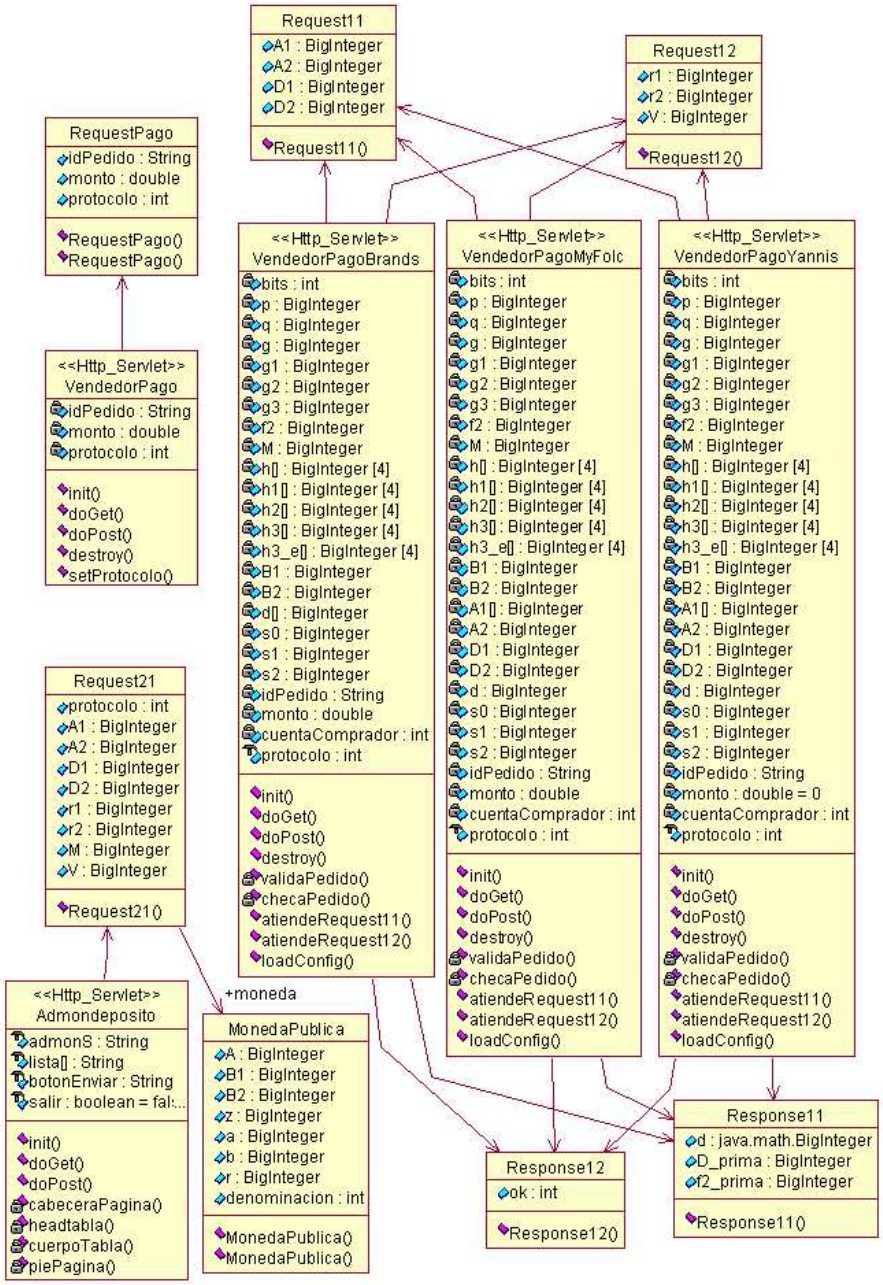

Figura A.4: Diagrama de clases que realizan los procesos de la entidad *vendedor*

Las clases que conforman la entidad *vendedor* se describen a continuación:

Para el proceso de pago/compra:

- RequestPago: Esta clase es la utilizada por la entidad *comprador* para solicitarle a la entidad *vendedor* el inicio del proceso de pago/compra. Encapsulando dentro de este objeto los datos pertinentes al pago por realizar.
- Request11, Request12: Son las clases necesarias para la requisición de datos que la entidad *comprador* solicita a la entidad *vendedor* y que deberán ser contestadas por el *vendedor* por su correspondiente clase Response. Dependiendo del protocolo de Pago/Compra que se esté utilizando.
- Response11, Response12: Son las clases necesarias para interpretar las respuestas que devuelve el *vendedor* a las peticiones hechas por el *comprador*, en el curso del proceso de Pago/Compra.
- VendedorPago: Esta clase, es el servlet con el que se inicia el proceso de pago/compra, al ser invocado mediante una petición Post y recibiendo un objeto RequestPago, inicia una sesión y establece el protocolo de pago/compra con el cual interactuaran las dos entidades. Si todo va bien devuelve un valor entero que identifica ya sea el número de error ocurrido o la aceptación correcta del objeto RequestPago.
- VendedorPagoBrands: Este servlet será el encargado de llevar a cabo el proceso de pago/compra por parte de la entidad *vendedor*, mediante el protocolo propuesto por S. Brands, para mayor detalle véase la sección 3.2.
- VendedorPagoYannis: Este servlet será el encargado de llevar a cabo el proceso de pago/compra por parte de la entidad *vendedor*, mediante el protocolo propuesto por Yannis et al., para mayor detalle véase la sección 3.3.
- VendedorPagoMyFOLC: Este servlet será el encargado de llevar a cabo el proceso de pago/compra por parte de la entidad *vendedor*, mediante el protocolo propuesto en esta tesis, para mayor detalle véase la sección 3.4.

Para el proceso de Depósito:

- Request21: este objeto contiene dentro de sí un objeto MonedaPublica y los datos que permiten verificar la autenticidad de la misma. Los cuales no son otra cosa que los datos recibidos en el proceso de pago/compra por el *vendedor*, quien es el que ahora, intenta abonar la moneda recibida a su cuenta.
- AdmonDeposito: Este es el servlet principal que realiza el proceso de depósito de monedas electrónicas, cuando es invocado mediante una petición WEB el servlet desplegará una lista de las monedas recaudadas por las ventas electrónicas, para que el usuario seleccione las monedas que desee depositar a su cuenta, una vez seleccionadas, se mandará a invocar el mismo mediante otra petición WEB enviándose de nuevo las monedas seleccionadas, para que se creen los objetos Request21 correspondientes a cada moneda y sean enviados a la entidad *banco* para su procesamiento. Al final este servlet desplegará el monto depositado y el número de monedas aceptadas por el *banco*.

A.3. El *comprador*

Como ya se dijo esta entidad tiene dos facetas: la que se realiza en una PC y la que se realiza por un PDA. Aunque las dos tienen el mismo objetivo su operación y funcionamiento es un poco diferente. La que es realizada por una PC es llevada a cabo mediante 2 applets, y la realizada por la PDA es efectuada por 2 aplicaciones específicas para la PDA que se esté utilizando.

El applet que realizan el proceso de retiro interactuando con la entidad *banco*, está constituido por el conjunto de clases que se muestra en la figura A.5.

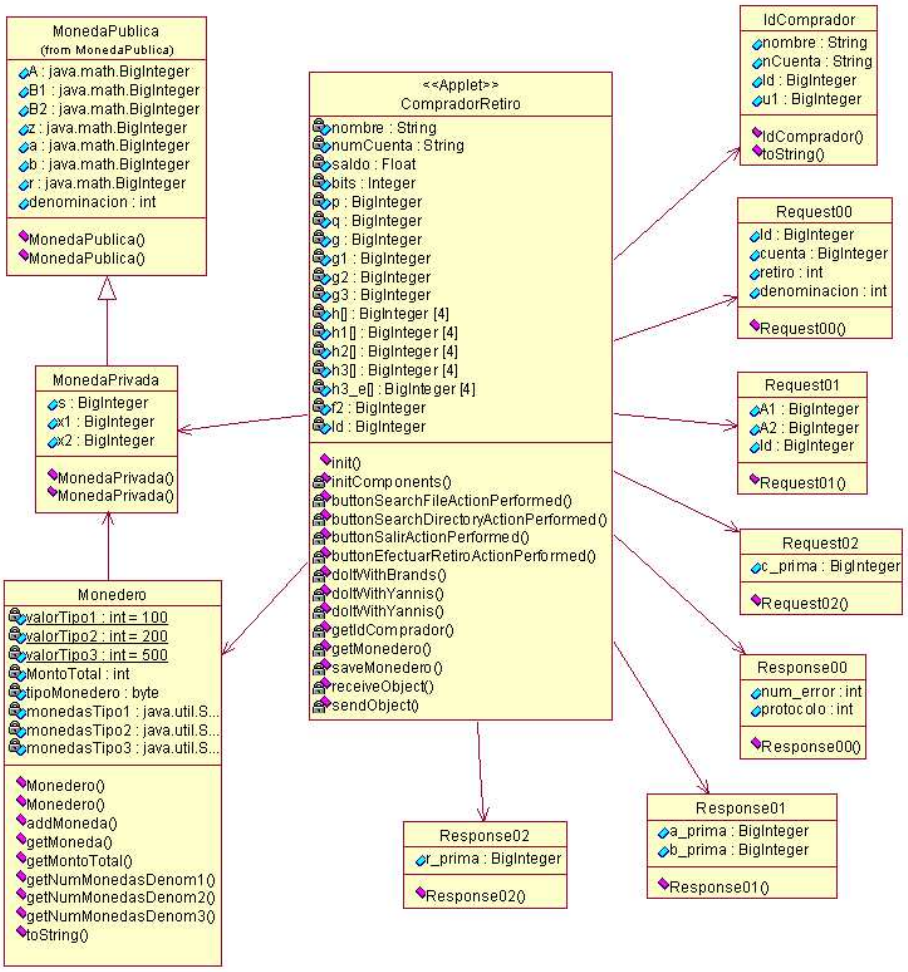

Figura A.5: Diagrama de clases para el proceso de retiro de la entidad *comprador* PC

Una breve descripción de las clases que conforman el diagrama de la figura A.5 se da a continuación:

- CompradorRetiro: Es la clase principal del applet. Esta clase despliega una interfaz de usuario con la cual el usuario *comprador* alimentando dicha interfaz con los parámetros requeridos, podrá efectuar la descarga de las monedas electrónicas a su archivo monedero. Para efectuar este proceso es posible hacerlo utilizando cualquiera de los tres protocolos implementados, siendo la entidad *banco* la que establece con que protocolo serán descargadas las monedas electrónicas.

- Monedero: Es la clase con las que se crea el archivo monedero el cual contendrá dentro de sí mismo una colección de objetos MonedaPrivada. Con esta clase se podrá gastar o descargar las monedas electrónicas. Este objeto sólo puede ser usado si se tiene el objeto IdComprador.

- IdComprador: Esta clase contiene los datos únicos de cada comprador, en especial lo que podríamos llamar como la llave privada del cliente y es con ésta con la que las monedas obtenidas a través del proceso de retiro son creadas.

- Request00, Request01, Request02 : Son las clases necesarias para la requisición de datos que la entidad *comprador* solicita a la entidad *banco* y que deberán ser contestadas por el *banco* por su correspondiente clase Response

- Response00, Response01, Response02 : Son las clases necesarias para interpretar las respuestas que devuelve el *banco* a las peticiones hechas por el *comprador*.

La aplicación para la PDA que realizan el proceso de retiro interactuando con la entidad *banco*, está constituida por el conjunto de clases que se muestra en la figura A.6.

Las clases que conforman esta aplicación son casi las mismas que se utilizaron en el applet, las clases que difieren se explican a continuación:

- MainFrame: Esta es la clase principal de tipo Frame con la cual la aplicación iniciará, su objetivo es crear la interfaz de usuario.

- AutenticaciónFrame: Es el Frame donde se llevará a cabo el proceso de autenticación y se cargaran los parámetros necesarios para realizar el proceso de retiro.

- CompradorRetiro: Es la clase principal con la que se llevará a cabo el proceso de retiro, dependiendo del protocolo establecido.

- ResultadosFrame: En este Frame se mostrarán los resultados del proceso de retiro de monedas electrónicas.

De igual manera el proceso de pago/compra se lleva a cabo mediante un applet o mediante una aplicación para PDA, las cuales, interactuarán con el servidor de la entidad *vendedor*, el applet está constituido por el conjunto de clases que se muestra en la figura A.7.

La descripción de las clases es la siguiente:

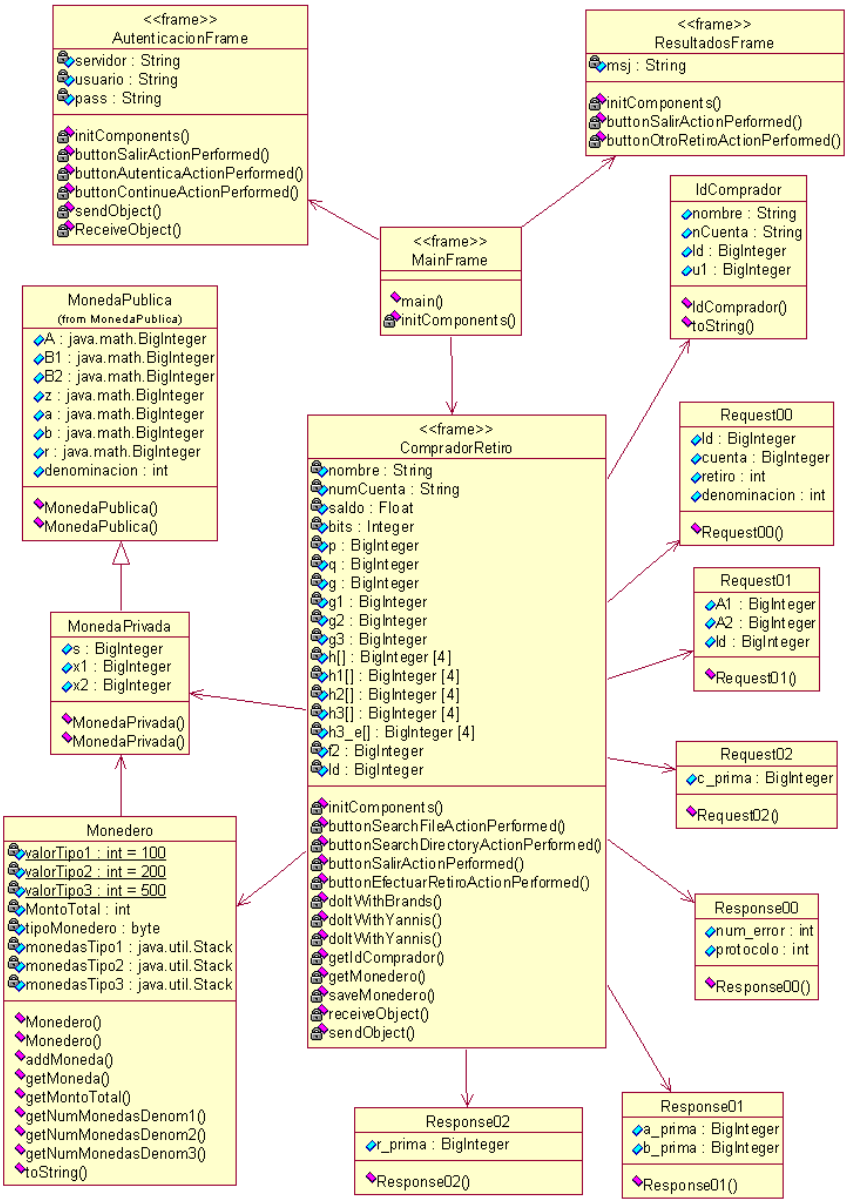

Figura A.6: Diagrama de clases para el proceso de retiro de la entidad *comprador* PDA

- *comprador*Pago: Es la clase principal del applet. Esta clase despliega una interfaz de usuario con la cual el usuario *comprador* alimentando dicha interfaz con los parámetros requeridos, podrá efectuar el pago/compra del pedido realizado electrónicamente en la tienda virtual utilizando las monedas electrónicas de su archivo monedero. Para efectuar este proceso es posible hacerlo utilizando cualquiera de los tres protocolos implementados, siendo la entidad *vendedor* la que establece con que protocolo serán recibidas las monedas electrónicas. Es necesario que, las monedas que vayan a ser utilizadas por el *comprador* hayan sido creadas con el mismo protocolo con el que se

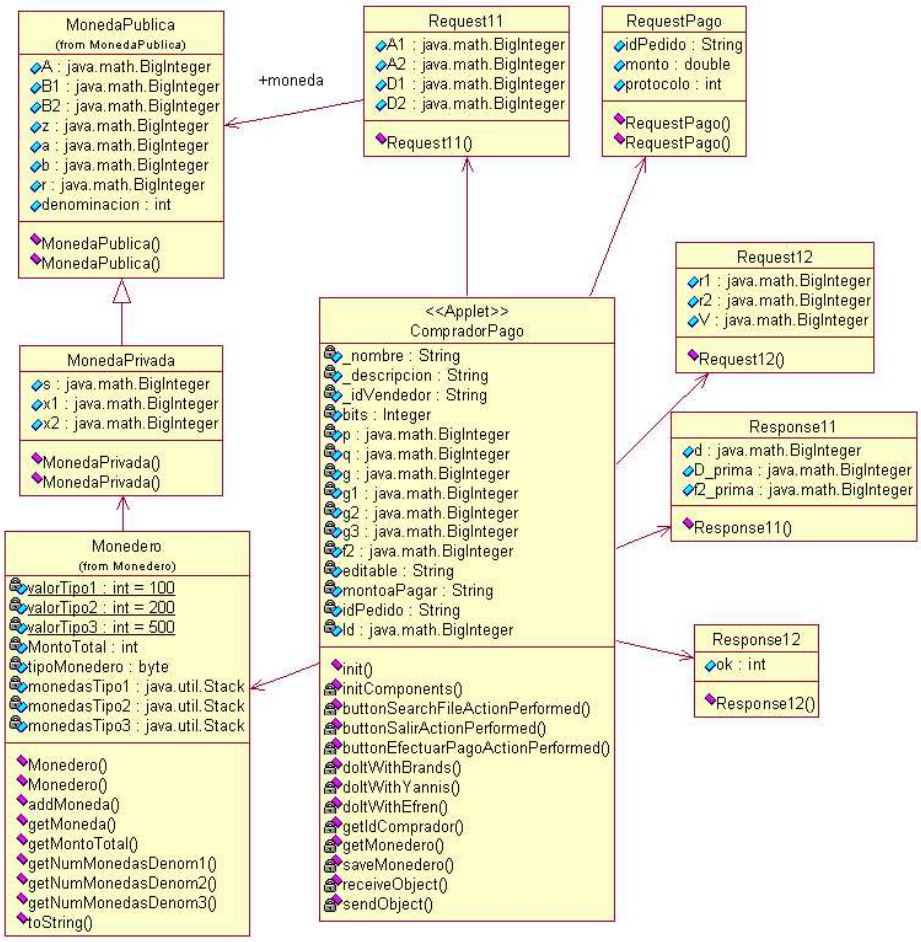

Figura A.7: Diagrama de clases para el proceso de pago/compra de la entidad *comprador* PC

realizará el proceso de Pago/Compra.

- RequestPago, Request11, Request12: Son las clases necesarias para la requisición de datos que la entidad *comprador* solicita a la entidad *vendedor* y que deberán ser contestada por el *vendedor* por su correspondiente clase Response. Dependiendo del protocolo de pago/compra que se esté utilizando.

- Response11, Response12: Son las clases necesarias para interpretar las respuestas que devuelve el *vendedor* a las peticiones hechas por el *comprador*, en el curso del proceso de Pago/Compra.

La aplicación para la PDA quedó conformada como se muestra en el siguiente diagrama de clases de la figura A.8, en donde como se podrá observar son reutilizadas la mayoría de las clases que se usaron en el applet.

Las clases que difieren a las del applet se describen a continuación:

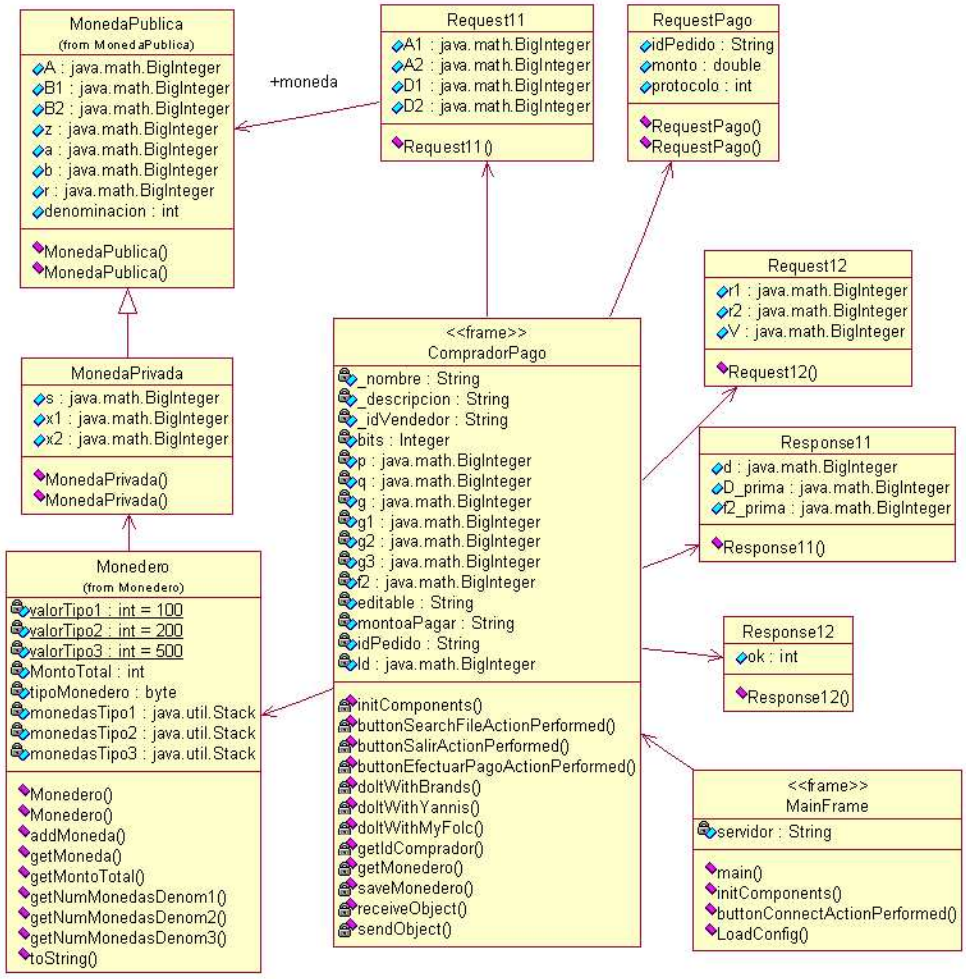

Figura A.8: Diagrama de clases para el proceso de pago/compra de la entidad *comprador* PDA

- MainFrame: Es la clase principal que será cargada inicialmente al ejecutar la aplicación, esta clase tiene como objetivo el inicializar la interfaz del usuario y conectarse al servidor para obtener los parámetros necesarios para realizar el proceso de pago/compra.
- CompradorRetiro: Realiza la misma función que en el applet, la única diferencia es que esta clase crea una interfaz de usuario de acuerdo a las limitaciones de la PDA.

A.4. La *autoridad*

La *autoridad*, realiza un único proceso, el de rastreo, en donde dependiendo de los datos que se le proporcionen esta entidad le devolverá al *banco* la información necesaria para que éste pueda obtener los datos de rastreo que se están buscando. En la figura A.9 se puede ver el diagrama de clases, y a continuación se detalla cada una de ellas:

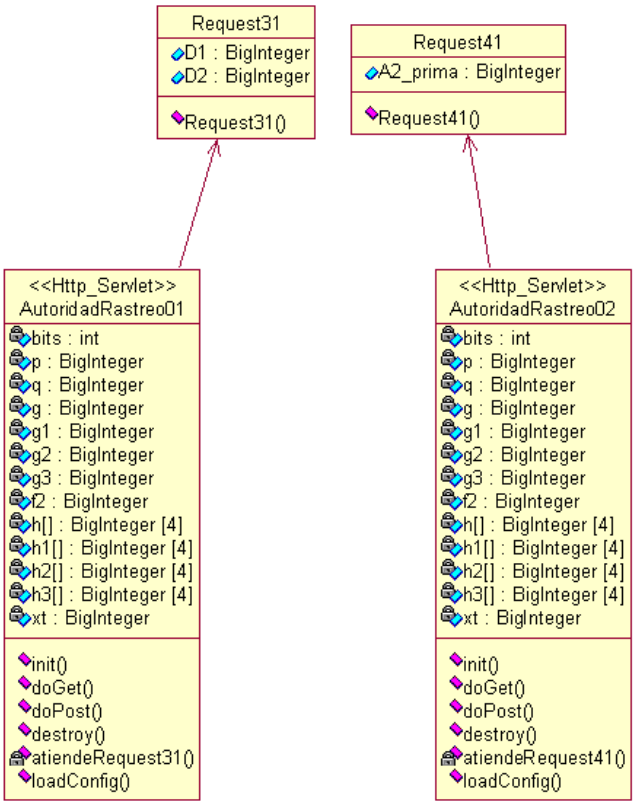

Figura A.9: Diagrama de clases de la entidad *autoridad*

El diagrama de clases presentado en la figura A.10 esta conformado básicamente por:

- AutoridadRastreo01: Este servlet realiza el proceso de rastreo de propietario solicitado por el *banco*, cuando la IP del *banco* fue autorizada para tener acceso al servidor, éste le envía el objeto Request31 mediante una petición Post y el servlet procesa los datos contenidos en dicho objeto para después devolver un valor numérico de tipo BigInteger con el cual el *banco* podrá obtener la identidad del propietario de la moneda.
- AutoridadRastreo02: Este servlet realiza el proceso de rastreo de moneda solicitado por el *banco*, cuando la IP del *banco* fue autorizada para tener acceso al servidor, éste le envía el objeto Request41 mediante una petición Post y el servlet procesa los datos contenidos en dicho objeto para después devolver un valor numérico de tipo BigInteger con el cual el *banco* podrá obtener los datos de donde fue gastada la moneda en cuestión.

Apéndice B

Funcionamiento del Sistema DEM

En este apéndice se muestra de manera muy general como son realzados los proceso de retiro y de pago/compra tanto para computadoras como para dispositivos móviles ligeros, utilizando las aplicaciones desarrolladas en esta tesis.

B.1. Proceso de retiro utilizando una PC

Cuando se utiliza una PC es necesario entrar al sitio WEB de la entidad *banco* para realizar un retiro de monedas electrónicas. El proceso de retiro podría generalizarse en tres pasos, los cuales se describen a continuación

1. Al ingresar al sitio WEB será desplegada la página WEB que se muestra en la figura B.1, en donde en la parte derecha se encuentran los campos de número de cuenta y contraseña los cuales deberán ser ingresados correctamente para que el usuario sea autenticado en el sistema.

2. Al ingresar los datos correctamente y haber presionado el botón de "Entrar"el sistema WEB de la entidad *banco* desplegará la pagina WEB que se muestra en la figura B.2 en donde tenemos tres opciones "Mostrar estado de cuenta", "Descargar monedas electrónicas"y "Salir". Para efectuar el proceso de retiro se da un click en "Descargar monedas electrónicas".

3. Cuando se ingresa en esta pagina la entidad *banco* genera una pagina en donde se descarga un applet firmado, solicitándole al usuario la carga del mismo, si el usuario acepta navegador mostrará la pagina WEB mostrada en la figura B.3.

En el applet mostrado en la figura B.3 se deberá ingresar por parte del usuario: la ruta del archivo de identificación, la ruta del archivo monedero, el monto del retiro y finalmente la denominación de las monedas con las que se efectuará dicho retiro. Una vez establecidos estos parámetros únicamente habrá que presionar el botón de "Efectuar Retiro"para que el proceso de retiro se lleve a cabo.

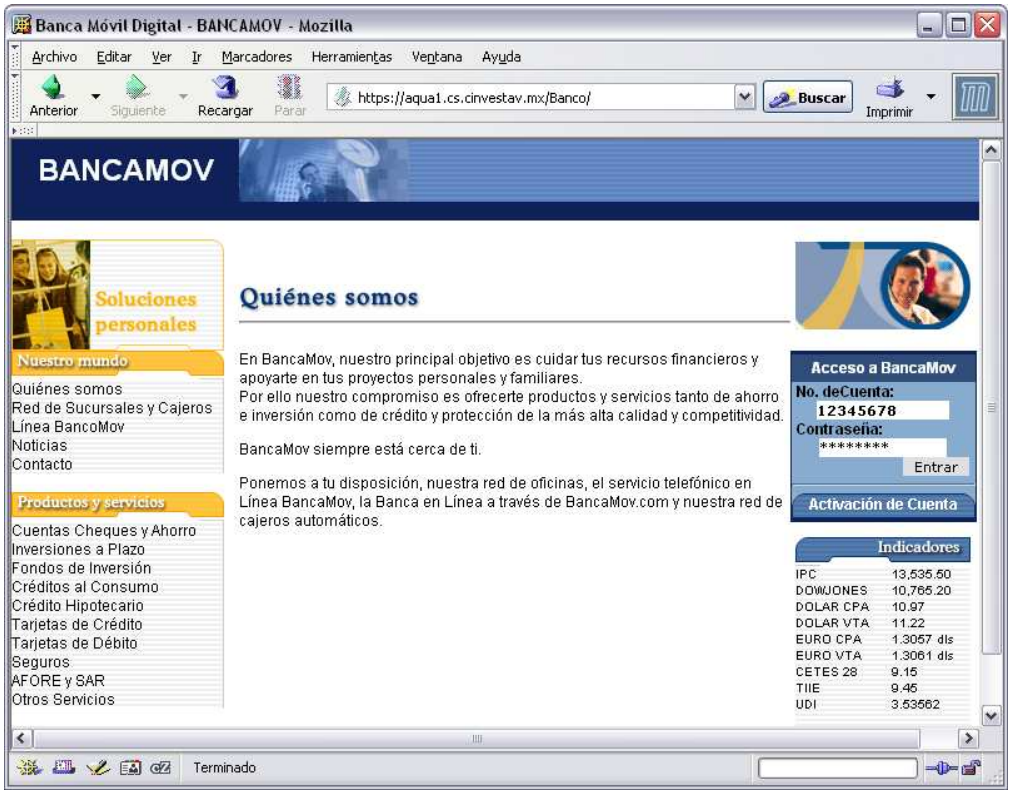

Figura B.1: Pagina WEB inicial de la entidad *banco*

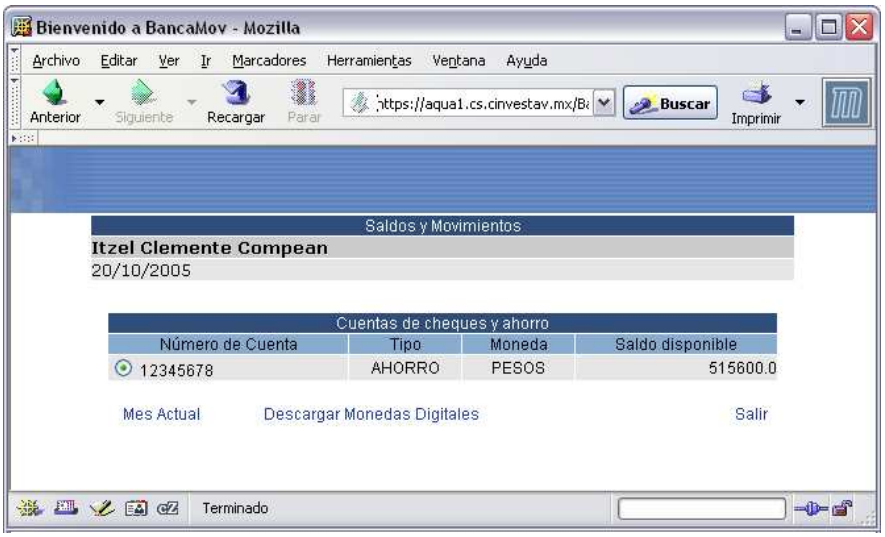

Figura B.2: Pagina WEB del manejo de cuenta de la entidad *banco*

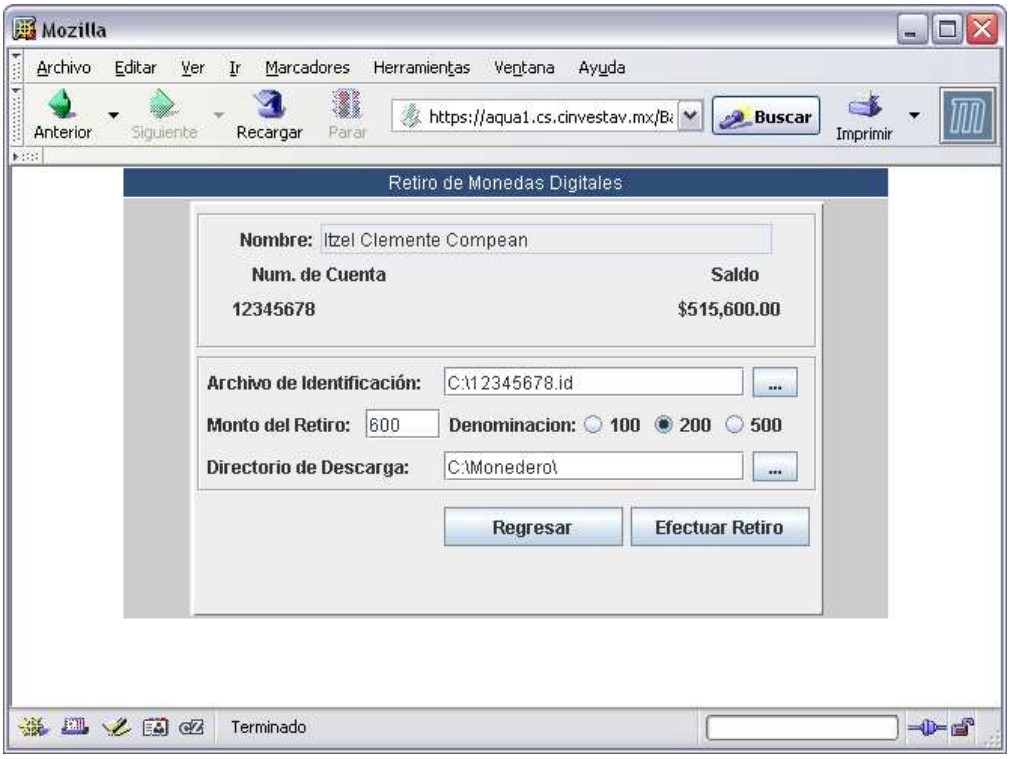

Figura B.3: Pagina WEB para la descarga de monedas de la entidad *banco*

B.2. Proceso de retiro utilizando una PDA

Cuando se utiliza una PDA para realizar un retiro de monedas electrónicas, es necesario iniciar la aplicación "Retiro"dentro del menú "Monedero", al hacerlo, se abrirá entonces la aplicación para efectuar la descarga de monedas electrónicas.

El proceso de retiro en la aplicación de la PDA consta de dos sencillos pasos:

1. Se establece la dirección del servidor de la entidad *banco* a la cual se conectará para descargar las monedas electrónicas, de igual manera son digitados el número de cuenta y la contraseña que fueron asignados previamente en el proceso de activación. En la figura B.4 podemos apreciar la pantalla inicial de la aplicación, en donde se establecen los parámetros previamente mencionados.

   Una vez establecidos estos parámetros, se presiona el botón de "Siguiente". Al presionarlo se realizará el proceso de autenticación dentro del sistema y si la autenticación es correcta se continuará con el proceso.

2. Al presionar el botón de "Siguiente"y efectuarse una autenticación correcta se desplegará la pantalla que se muestra en la figura B.5 en donde cómo se puede apreciar en la parte superior serán mostrados los datos de la cuenta del cliente, para enseguida solicitar la dirección del archivo de identificación y la dirección del archivo monedero; por último se le solicita al usuario el monto del retiro y la denominación de las monedas con las que se efectuará dicho retiro

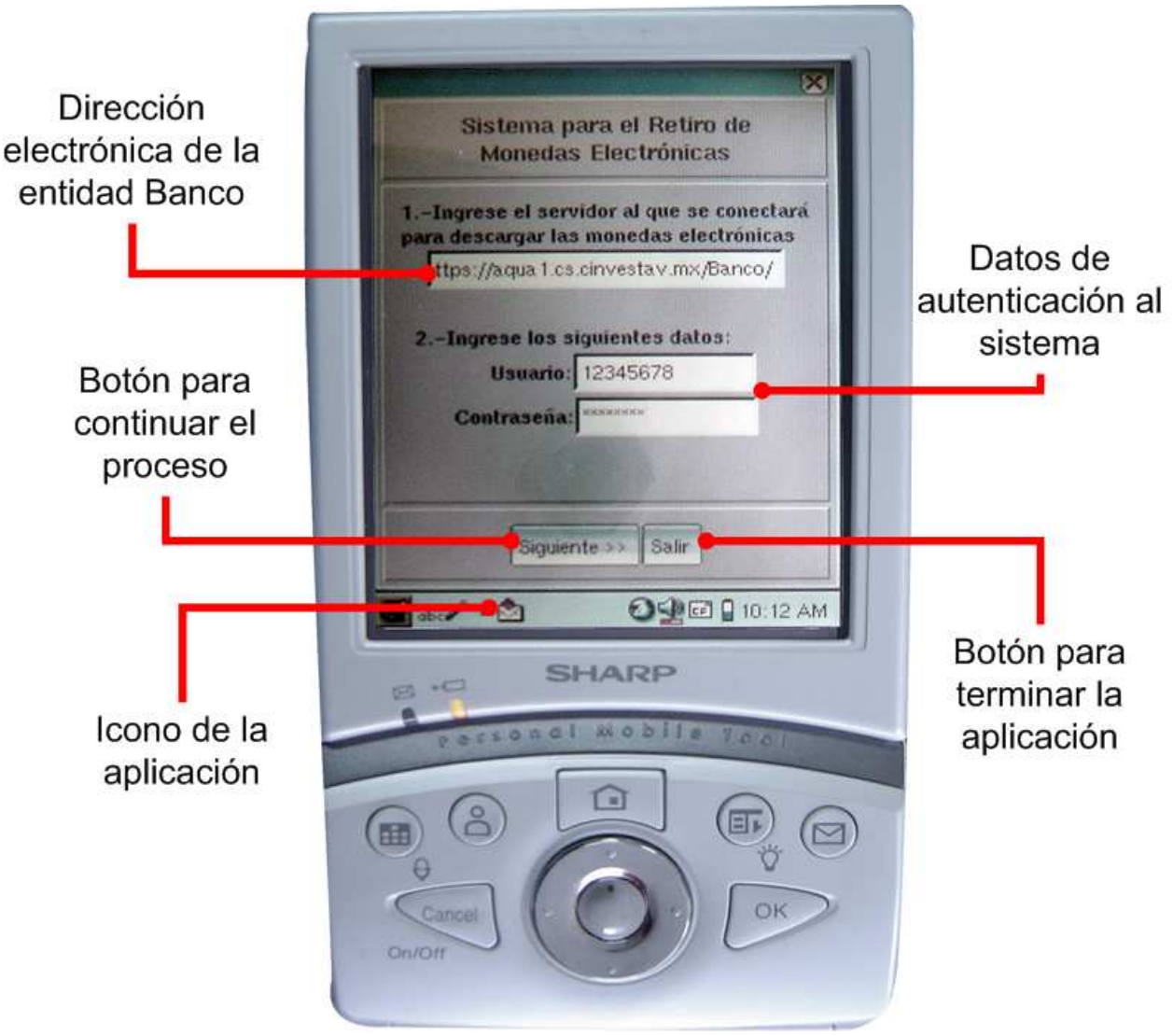

Figura B.4: Pantalla inicial de la aplicación retiro implementada para PDA

Al establecer los parámetros solicitados y presionar el botón de "Siguiente"será llevado a cabo el proceso de retiro de acuerdo al protocolo establecido (recordemos que pueden ser 3 diferentes). Al terminar la descarga de las monedas será mostrada una pantalla, la cual puede verse en la figura B.6, en donde se desplegará el contenido en monedas de nuestro archivo monedero.

B.3. Proceso de pago/compra utilizando una PC

Cuando se utiliza una PC para realizar el pago de un pedido realizado en una tienda virtual utilizando monedas electrónicas, es necesario realizar primeramente un pedido dentro

Figura B.5: Pantalla para la realización del proceso de retiro en la aplicación para PDA

de la tienda virtual. Después de haber establecido el pedido solicitando los productos o servicios requeridos y haber establecido el monto de la compra se procede de la siguiente manera para realizar el proceso de pago:

1. Se establecen los datos para la entrega del pedido.

2. Se descarga un applet firmado desde el servidor de la entidad *vendedor* tal y como se muestra en la figura B.7. En dicho applet se deberán establecer el archivo de identificación, el archivo monedero y el número de monedas con el cual se realizará el pago. Hecho lo anterior se procede a presionar el botón de "Efectuar de pago"para que se realice el proceso de pago de acuerdo al protocolo de dinero electrónico establecido.

Figura B.6: Pantalla final del proceso de retiro en la aplicación para PDA

3. Si el proceso de pago/compra se ha llevado a cabo con éxito se muestra una pagina, tal como lo muestra la figura A.8, dándole a conocer al usuario el estado del pedido.

B.4. Proceso de retiro utilizando una PDA

Cuando se utiliza una PDA para realizar el pago de un pedido realizado en una tienda virtual utilizando monedas electrónicas, es necesario iniciar la aplicación "Pago"dentro del menú "Monedero", al hacerlo, se abrirá entonces la aplicación para efectuar la descarga de monedas electrónicas.

El proceso de retiro en la aplicación de la PDA consta de dos sencillos pasos:

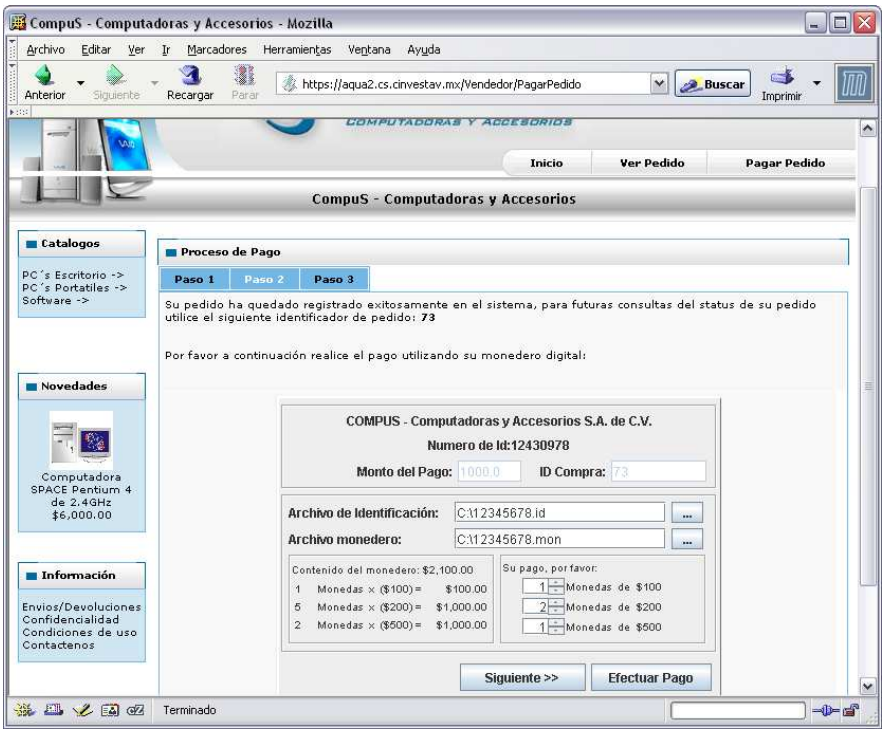

Figura B.7: Pantalla que muestra el applet con el cual se realizará el proceso de pago/compra para PC

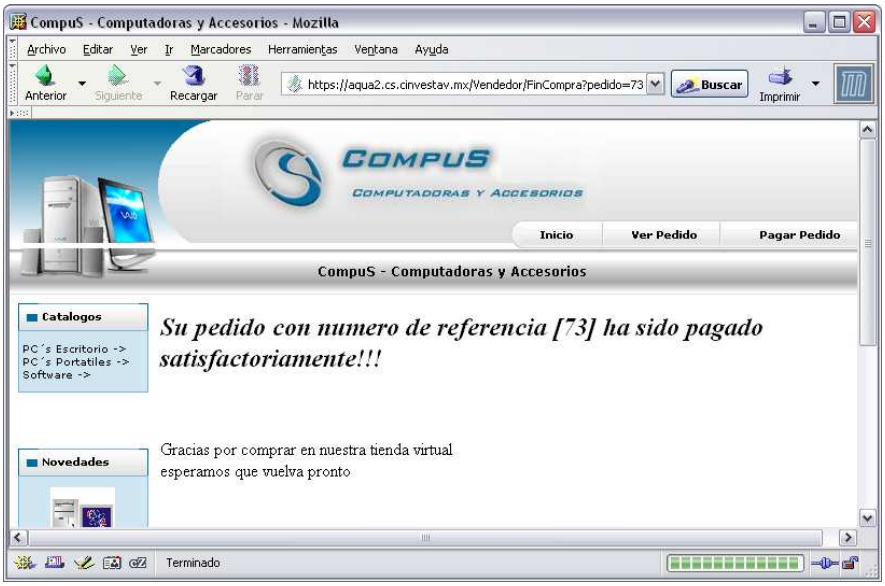

Figura B.8: Pantalla final del proceso de pago/compra en la aplicación para PC

1. Se establece la dirección del servidor de la entidad Vendedor a la cual se conectará para pagar con las monedas electrónicas, de igual manera son establecidos el número del

pedido a pagar, la ruta del archivo de identificación y la ruta del archivo monedero. En la figura B.9 podemos apreciar la pantalla inicial de la aplicación, en donde se establecen los parámetros previamente mencionados.

Figura B.9: Pantalla inicial que muestra el proceso de pago/compra en la aplicación para PDA

Una vez que se han establecido los parámetros ya mencionados se debe presionar el botón de "Siguiente"para continuar con el proceso. Si el vendedor encuentra el pedido solicitado entonces la aplicación continua con el proceso, en caso contrario la aplicación le enviará un mensaje al usuario indicándole que el pedido solicitado no existe o ya ha sido pagado.

2. Si todo va bien la aplicación muestra la pantalla que se puede ver en la figura B.10 en donde le indicará al usuario el monto del pedido a pagar, y el usuario deberá establecer el

pago de dicho pedido indicando el número exacto de las tres diferentes denominaciones con las cuales el monto del pedido será cubierto.

Figura B.10: Pantalla que muestra el proceso de pago/compra en la aplicación para PDA

Finalmente se presiona el botón de "Siguiente"y entonces se realiza el proceso de pago/compra dependiendo del protocolo seleccionado. Al realizar el pago del pedido será mostrada una pantalla, la cual puede verse en la figura B.10, en donde se desplegará el estado del pedido (pagado, no pagado) dependiendo del resultado del proceso de pago y de igual forma se mostrará el contenido final del archivo monedero.

Figura B.11: Pantalla que muestra fin del proceso de pago/compra en la aplicación para PDA

Apéndice C

Funcionamiento Casino en Línea

En esta sección se explican los detalles más importantes de la adaptación realizada a un sistema de dinero electrónico para que funcione para el establecimiento de las apuestas en un casino en línea.

En la figura C.1 se muestra la página inicial del casino en línea implementado. En esta página se puede ver del lado derecho el menú principal del sitio WEB en donde las opciones son las siguientes:

Figura C.1: Pagina WEB inicial de la entidad *casino*

- Inicio: Muestra la pagina principal del sitio WEB
- BlackJack: Muestra las reglas y las instrucciones principales del juego BlackJack implementado
- Activación: Página que realiza el proceso de activación de la cuenta del usuario en el casino.
- e-Fichas: Le permite a cualquier usuario ingresar a su cuenta en el casino para realizar ya sea un retiro de fichas electrónicas (para posteriormente apostar en el juego) o bien realizar un depósito de fichas (para incrementar su saldo en la cuenta del casino).
- Jugar: Se desarrolla el juego de blackjack en donde se realizan las apuestas con las fichas electrónicas se ganan y se pierden las mismas.

Los proceso de retiro y de depósito de fichas electrónicas son análogos a los proceso mostrados en el apéndice B en donde se muestran el proceso de retiro y el proceso de pago/compra de monedas electrónicas para computadoras. Por esto mismo aquí solo se muestra como quedó conformado el sistema de juego de Blackjack que implementa, en el manejo de apuestas, el uso de un sistema de dinero electrónico.

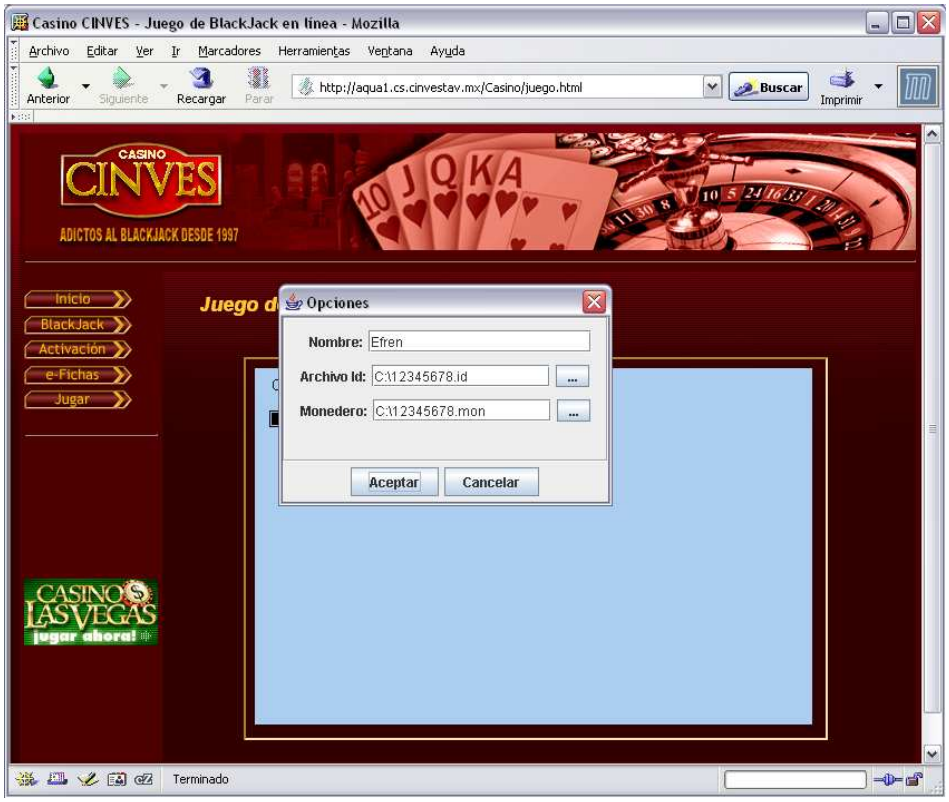

Figura C.2: Pagina WEB inicial para el incio del juego de BlackJack

Cuando se ingresa al juego de Blackjack lo primero que se le pide al usuario es que especifique la ruta del archivo de identificación y de igual forma la ruta del archivo donde se encuentran las fichas electrónicas, tal y como lo muestra la figura C.2.

Una vez realizado lo anterior, el juego de blackjack es cargado como un applet. Y se activa un único botón "Apostar". Con dicho botón se establece la apuesta realizando un proceso de pago/compra entre la entidad Apostador y la entidad Casino. Para esto, se debe previamente especificar la apuesta, en el cuadro de dialogo que aparecerá justo cuando se presiona el botón de "Apostar". Tal y como se muestra en la figura C.3.

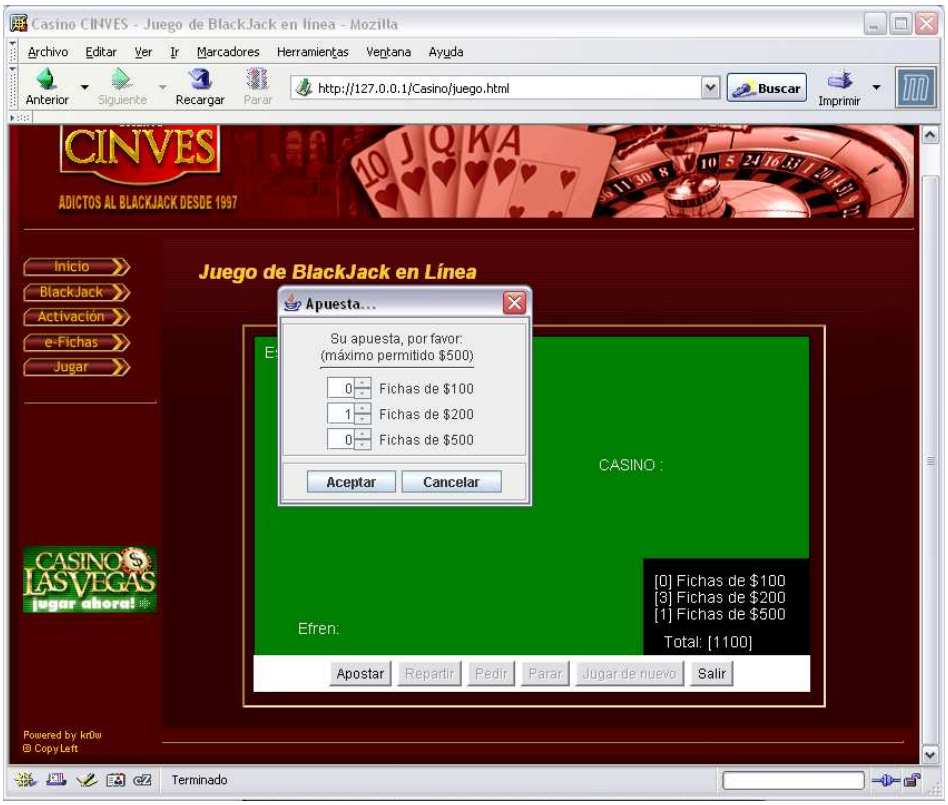

Figura C.3: Establecimiento de la apuesta para el juego en línea de BlackJack

Una vez establecida la apuesta, se activará el botón de "Repartir" con el cual dará comienzo el juego de blackjack entre el servidor de la entidad casino y el usuario, repartiendo las cartas iniciales con las que se comienza el juego, tal y como se muestra en la figura C.4

Si el usuario pierde el juego las monedas apostadas se pierden. Pero si gana se realiza un proceso de retiro con el cual el usuario obtendrá las fichas electrónicas que pagaran la apuesta realizada al principio del juego, para dar la opción de comenzar otro juego o de salir del applet del jeugo de Blckjack.

Figura C.4: Pagina WEB que muestra el desarrollo del juego en línea de BlackJack

Figura C.5: Fin del juego y pago de apuesta en el juego de BlackJack

Bibliografía

[1] Michael O. Rabin. *Digitalized Signatures*. Foundations of Secure Computation, ed. by R.A. DeMillo, D.P. Dobkin, A.K. Jones, R.J. Lipton; Academic Press, N.Y. 1978, 155-166.

[2] D. Chaum. *Blind signatures for untraceable payments*. Proc. of CRYPTO'82, pag. 199-203. Springer-Verlag(1982)

[3] D. Chaum, A. Fiat y M. Naor. *Untraceable electronic cash*. LNCS 403, Proc. Of CRYPTO'88 pag. 319-327. Springer-Verlag(1988).

[4] T. Okamoto y K. Otha. *Universal electronic cash*. LNCS 576, Proc. Of CRYPTO'91, pag. 324-337, Sprnger-Verlag(1991).

[5] C.P. Schnorr. *Efficient signature generation by smard cards*. Journal of Cryptology 4(3) pag. 161-174, (1991).

[6] S. Brands, D. Chaum, R. Cramer, N. Ferguson y T Pedersen. *Transaction systems with observers*. Manual no publicado. URL: http://homepages.cwi.nl/~cramer/

[7] D. Chaum y T. Pedersen. *Wallet databases with observers*. LNCS Proc. Of CRYPTO'92 pag. 90-106 Springer-Verlag

[8] R. Cramer y T. Pedersen. *Improved privacy in wallets with observers.* LNCS 765. Advances in Cryptology EUROCRYPT'93, pag 329-343 Springer-Verlag (1993)

[9] S. Brands. *Untraceable off-line cash in wallet with observer*. LNCS 773, Proc. Of CRYPTO'93, pag. 302-318. Springer-Verlag(1993).

[10] N. Ferguson. *Single term off-line coins.* LNCS 765 Advances in Cryptology EUROCRYPT'93 pag. 318-328. Springer-Verlag(1993).

[11] M. Jakobsson y M. Yung. *Revokable and Versatile Electronic Money*. (1996).

[12] Y. Frankel, Y. Tsiounis y M. Yung. *Indirect Discourse Proofs: Achieving Efficient Fair Off-Line E-Cash*. Advances in Cryptology - Proceedings of Asiacrypt 1996, pp. 286-300.

[13] A. Chan, Y. Frankel and Y Tsiounis. *How to break and repair e cash protocols based on the representation problem.* Research Report NU CCS Northeastern University Boston Massachussets, 1996.

[14] Alfred Menezes, P van Oorschot, y S. Vanstone, *Handbook of Applied Cryptography*. Ed. CRC Press, 1996.

[15] B. Schneier. *Applied Cryptography. Protocols, Algorithms, and Source Code in C*. Ed. Wiley, segunda edición, 1996.

[16] Yiannis S. Tsiounis. *Efficient Electronic Cash: New Notations and Techniques*. Tesis de Doctorado, Northeastern University Boston, Massachusetts(1997).

[17] Mandana Jahanian Farsi. Digital Cash. Tesis de Maestría. Departament of mathematics and computing science, Göteborg University (1997)

[18] A. Chan, Y. Frankel and Y Tsiounis. *Easy come - easy go divisible cash.* LNCS 1403, pag. 561. Springer-Verlag(1998).

[19] G. Ateniese, J. Camenisch, M. Joye y G. Tsudik. *A Practical and Provably Secure Coalition-Resistant Group Signature Scheme*, Proceedings of CRYPTO 2000, pp. 255-270, 2000.

[20] Greg Maitland and Colin Boyd. *Fair Electronic Cash Based on a Group Signature Scheme*, Information and Communications Security, ICICS'01, LNCS 2229, pp.461-465. Springer-Verlag (2001).

[21] Wooseok Ham. *Design of Secure and efficent E-commerce protocols using criptographic primitives*. Tesis de Maestría, School of engineering Information and Comunications University, Daejeon Korea(2002).

[22] Wade Trappe y Lawrence C. Whashington. *Introduction to Cryptography with coding Theory*. Ed. Prentice Hall (2002).

[23] Popescu Constantin. *An Off-line Electronic Cash System with Revokable Anonymity.* Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference, 12-15 May 2004.

[24] Cryptology Pointers by Helger Lipmaa. URL: http://www.tcs.hut.fi/~helger/crypto/link/protocols/ecash.php

[25] Miquel Soriano. *Sistemas de Pago electrónicos: Presente y Futuro.* Universidad Politécnica de Cataluña, España (2003). URL: http://www.criptored.upm.es/guiateoria/gt_m008a.htm

[26] Gonzalo Álvarez Marañón (Criptonomicon). *Dinero electrónico.* URL: http://www.iec.csic.es/criptonomicon/comercio/dineroe.html

[27] *Secure Electronic Transaction Specification*. Book 1: Business Description.V 1.0. Mayo 1997. http://www.setco.org/set_specifications.html

[28] Ray Byler. *What is the money?*. Mid-South College Computing Conference, pag 200-209, 2004.

[29] Stephen Lea et al. *The Individual in the Economy- A Survey of Economic Psychology*. Press Syndicate of the University of Cambridge, U.S.A. 1987.

[30] Ian Sommerville. *Ingeniería de Software*. Ed. Adison Wesley, 6a Edición - 2002.

[31] Marty Hall. *Core Servlets y Java Server Pages*. Ed. Prentice Hall, 1a Edición - 2001.

[32] *J2ME PersonalJava*, http://java.sun.com/products/personaljava/

[33] *Java Programming on the Sharp Zaurus*, http://developers.sun.com/techtopics/mobility/personal/articles/ztutorial

[34] *Whitepaper: An Introduction to Smart Cards*, http://www.sspsolutions.com/solutions/whitepapers/introduction_to_smartcards/

[35] *EMV: Integated Circuit Card Application*, http://international.visa.com/fb/paytech/productsplatforms/downloads.jsp

[36] *EMV CPA 1.0 - Draft*, http://www.emvco.com/

[37] *Wallet Phones Report*. Gartner Research issued. http://www.eurotechnology.com/store/walletphone/index.html

[38] *e-Cash System* http://esp.ecashdirect.net/about.html