Universidad de las Ciencias Informáticas

Facultad 1

Solución para el control de acceso a la información de las entidades externas, en la cédula de identificación electrónica de la República Bolivariana de Venezuela.

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor(es): Joel Sáez Vilar.
Dayron Almeida Sotolongo.

Tutor(es): Ing. Adonis Cesar Legón Campos.

18 de Junio del 2009.

Ciudad de la Habana.

“*La ciencia y las letras doman las pasiones que engendra la política. Tiempo es ya de que el afecto reemplace en la ley del mundo al odio.****”***

José Julián Martí Pérez.

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado:
Solución para el control de acceso a la información de las entidades externas, en la cédula de identificación electrónica de la República Bolivariana de Venezuela
y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.
Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

_____ _____
<Nombre del estudiante 1> <Nombre del estudiante 2>

_____
<Nombre del Tutor>

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

AGRADECIMIENTOS

Joel:

Dayron:

DEDICATORIA

Joel:

Dayron:

RESUMEN

La República Bolivariana de Venezuela está inmersa en un complejo proceso de construcción, generación y perfeccionamiento del nuevo Estado Venezolano, al transformar y sustituir los viejos sistemas de orden político y social; por completas y modernizadas instituciones públicas que basan el flujo de sus procesos en los más novedosos adelantos tecnológicos.

Para comenzar con el nuevo proceso de cambios era necesario dar los primeros pasos y romper con las viejas costumbres que los venezolanos por mucho tiempo habían asumido, bajo estos principios surge la “Misión Identidad”; que posteriormente da lugar al Proyecto Identidad con el objetivo de automatizar el proceso de identificación de los venezolanos; surgiendo así el Servicio Autónomo de Identificación, Migración y Extranjería (SAIME).

Con el cambiante mundo tecnológico y aguardando tener una mayor seguridad en cuanto a identificación de cada ciudadano; el gobierno venezolano consideró la utilización de tarjetas inteligentes como documento de identificación (cédula de identificación electrónica) y de esta forma brindar la posibilidad a instituciones del país, guardar información referente a los servicios que prestan en la nueva cédula electrónica. En el presente trabajo se desarrolla una solución que permite gestionar el acceso a la información que pudieran almacenar instituciones externas a la Oficina Nacional de Identificación y Extranjería (ONIDEX) en la cédula de identificación electrónica, mediante la utilización de mecanismos de seguridad basados en certificados digitales y cumpliendo con los estándares internacionales definidos para el tema. A continuación le mostramos el trabajo y esperamos que les sea de interés.

PALABRAS CLAVES:

Proceso; Sistema; Subsistema; Misión; Cedulación; Entidades.

ÍNDICE DE CONTENIDO.

ÍNDICE DE TABLAS.

ÍNDICE DE FIGURA.

INTRODUCCIÓN

La automatización de los procesos que en la ONIDEX (*Oficina Nacional de Identificación y Extranjería*) se realizan, constituyó y constituye aún, el punto de partida de la transformación de la misma en una entidad con un alto sentido revolucionario, que tiene como perspectiva fundamental la eficiencia, seguridad y calidad de todos los servicios que se le brinden a la población.

Para comenzar el nuevo proceso de cambios era necesario dar los primeros pasos y romper con el viejo proceso de identificación que los venezolanos por mucho tiempo habían asumido, bajo estos principios tiene lugar la "Misión Identidad" como acción a corto plazo dirigida principalmente a propiciar la cedulación de la población con edad electoral que no estaba registrada.

El éxito de esta misión demostró que el pueblo y las instituciones de la República Bolivariana de Venezuela estaban listos para iniciar, a una mayor escala, el perfeccionamiento de su sistema de identificación. Para este entonces surge el Proyecto Identidad, encargado de desarrollar el Sistema SAIME (*Servicio Autónomo de Identificación, Migración y Extranjería*), cuyo objetivo fundamental es la reestructuración, modernización y automatización de todos los procesos que se desarrollan en la ONIDEX, para brindar servicios a los ciudadanos. Esta institución ha iniciado un ambicioso programa de trabajo para insertar las más seguras y modernas tecnologías en todas las áreas relacionadas con el Sistema de Identificación venezolano.

Con el avance de las tecnologías y en aras de elevar la seguridad del documento de identificación de Venezuela, a niveles sólo comparables con países altamente desarrollados, convirtiéndolo en la plataforma tecnológica que permitirá satisfacer las necesidades presentes y futuras de la sociedad, en materia de identificación, las instituciones competentes en el tema acuerdan la puesta en marcha de una segunda fase en el sistema que se había venido desarrollando, para adaptarlo a la asimilación de un nuevo tipo de documento de identificación. Esta etapa se basa en el empleo de tarjetas de identificación electrónica, las cuales cuentan con lo más novedoso en materia de identidad y con múltiples medidas de seguridad.

Las tarjetas inteligentes, como mecanismos de identificación, son utilizadas actualmente en varios países del mundo, en los cuales su forma de uso es diversa. La implantación de sistemas de identificación nacional basados en la utilización de tarjetas inteligentes, ha permitido que se reúnan una serie de experiencias, las cuales constituyen un buen punto de partida para aquellos países que pretendan aplicar esta tecnología.

Finlandia fue el primer país europeo en emitir tarjetas de identidad electrónica a finales de 1999. La tarjeta FINEID lleva impreso en el frente la información relativa a la tarjeta y el ente emisor de la misma y en el reverso: identificación de la entidad emisora, datos de la tarjeta y del ciudadano. El DNI electrónico de España es una tarjeta de identificación de tamaño ID-1 (definida en la norma ISO/IEC-7810), construida de policarbonato, que incorpora un chip con información digital. El documento de identificación puede ser usado de diferentes maneras, para identificarse presencialmente o de forma electrónica a través de los certificados contenidos en ella, además puede ser usado para la firma electrónica de documentos. La tarjeta también se utiliza en el sector privado para: acceder a sitios WEB, firma de contratos, verificación de autenticidad de documentos y en el sector público para la declaración anual de impuestos, interacción con la Administración Pública para la obtención de formularios, servicios en línea, y registros criminales.

En la República Bolivariana de Venezuela, este tipo de documento es en esencia una tarjeta inteligente o SmartCard, compuesta por una lámina de policarbonato y un chip con interfaz de comunicación por radio frecuencia o sin contacto como también se le conoce. Tiene incorporado un sistema operativo basado en JavaCard, que permite la instalación de varias aplicaciones en una misma tarjeta, soportado por un entorno de ejecución que administra y controla su seguridad e integridad. La información relacionada con el ciudadano está impresa en laser grabado en la superficie del documento y contiene además características de impresión de seguridad.

La nueva cédula de identificación electrónica, permite almacenar la información necesaria para comprobar, con alto grado de seguridad, la identidad de su portador y facilitará el desarrollo de servicios para la sociedad y el gobierno electrónico. Cuenta además con un mecanismo de verificación biométrico para garantizar la mayor unicidad posible en la

identificación y tendrá la capacidad de agrupar, en un solo documento, la posibilidad de utilizarse en diferentes servicios ya identificados, y algunos otros que se podrán proponer como parte de la infraestructura tecnológica que se pretende instalar para su uso, con el objetivo de minimizar la cantidad de trámites, reduciendo costos administrativos, garantizando la seguridad en las transacciones y el manejo de información. Todo lo cual contribuye a fortalecer la seguridad nacional.

El empleo de una tarjeta inteligente, brindará la posibilidad de contar con múltiples aplicaciones y servicios, por lo que además de ser un documento de identificación podrá usarse para almacenar información, referente a la persona, y de interés para las instituciones o entidades que necesiten utilizarla, con el objetivo de automatizar sus propios procesos. Tal es el caso de las historias clínicas, licencias de conducción, servicios bancarios, entre otros. Toda esta información deberá tener además, niveles y condiciones de acceso para garantizar su confiabilidad e integridad, protegiendo los intereses de la persona y la institución implicados, y fomentando el desarrollo de servicios electrónicos en el país, orientados al ciudadano o incluso a las propias instituciones.

Partiendo de estos acontecimientos surge una **Situación Problémica**, para la cual es necesario dar una solución: la cédula es el documento de identificación personal de la República Bolivariana de Venezuela, que mediante un proceso de modificación tecnológica, se ha convertido en una cédula de identificación electrónica, de esta forma es necesario garantizar a un grupo de entidades externas a la ONIDEX, la gestión de la información que necesiten almacenar en este documento, utilizando todas los mecanismos de seguridad existentes para este tipo de tecnología y teniendo en cuenta el uso de tecnologías multiplataforma, debido al nivel de heterogeneidad de los sistemas que existen en el país. Por estas razones es necesario proveer una solución de software multiplataforma, que permita el control de acceso a la información almacenada en la cédula de identificación electrónica para facilitar a las entidades externas su uso, en el desarrollo de aplicaciones y servicios que integren esta tecnología en sus procesos y garanticen al ciudadano un servicio confortable, de calidad y acorde con las intensiones de desarrollo tecnológico en el país.

De aquí la interrogante en la que nos enmarcamos para plantear el **Problema Científico**.

¿Cómo implementar el control de acceso a la información de las entidades externas a la ONIDEX, en la cédula de identificación electrónica de la República Bolivariana de Venezuela?

Como **campo de acción,** el proceso de gestión de la información almacenada en la cédula electrónica, de las entidades externas a la ONIDEX, en la República Bolivariana de Venezuela.

Para dar solución a la interrogante se plantea la siguiente **Hipótesis**:

Con la implementación de una solución integral multiplataforma se logrará controlar el acceso a la información almacenada en la nueva cédula electrónica de identidad de la República Bolivariana de Venezuela.

Dada la hipótesis planteada anteriormente se pueden definir como **variables de la investigación**:

Variable Independiente: implementación de una solución multiplataforma.

Variable Dependiente: control de acceso a la información.

Operacionalización de las Variables

Variable Conceptual	Dimensión	Indicadores	UM
Implementar una aplicación Applet y Middleware para la gestión de la información en la cédula electrónica de identificación.	Factibilidad	Tiempo de desarrollo	Extenso Moderado Breve
		Costo	Costoso Moderado Barato
		Esfuerzo	Alto Moderado Despreciable

Variable Conceptual	Dimensión	Indicadores	UM
Permitir el control de la información contenida en la cédula electrónica, a las entidades externas a la ONIDEX.	Mejor seguridad	Complejidad	Alta Media Baja
		Control	Bueno Malo
		Organización del trabajo	Bueno Malo
		Importancia	Alta Media Baja
		Dependencia	Dependiente Independiente
		Tiempo de ejecución	Rápido Medio Lento

Tabla 1: Operacionalización de las Variables.

Para dar solución al problema existente, se ha tomado como **Objetivo general**:

Desarrollar el análisis, diseño e implementación de componentes Applet y Middleware para permitir a entidades externas a la ONIDEX, acceder y gestionar la información que contienen en la cédula de identidad electrónica.

Además de **Objetivos específicos** como:

- Determinar aspectos teóricos conceptuales sobre aplicaciones Applet (según tecnología JavaCard).
- Determinar aspectos teóricos conceptuales sobre componente Middleware.
- Documentar la situación actual de Applet implementados en el mundo.
- Documentar la situación actual de Mecanismos de Autenticación sobre SmartCard.

- Documentar la propuesta de diseño de certificado digital para el control de acceso de las Entidades Externas a la cédula electrónica.
- Documentar la propuesta de desarrollo del Middleware para nuestro proyecto.
- Desarrollar el análisis, diseño e implementación de la aplicación Applet y del componente Middleware.
- Realizar las pruebas de calidad a la aplicación Applet y al componente Middleware.

Tareas de Investigación.

- Investigación sobre la arquitectura de la tarjeta (Ambiente de ejecución, JavaCard Runtime Environment, Administrador de tarjetas, Card Manager, Dominios de seguridad, API de Open Platform).
- Investigación sobre Arquitecturas de Seguridad PKI.
- Investigación sobre los estándares (JavaCard, Global - Platform, PKCS, PC / SC, Normas ISO).
- Implementación de la aplicación Applet con todos los estándares y normas establecidas.
- Implementación del componente Middleware para la gestión de los Applet.
- Desarrollo de la documentación y diagramas de ingeniería de software necesarios para el análisis y diseño de la solución.
- Estudio y selección de las herramientas que permitan obtener una solución óptima para la implementación de la solución tecnológica integral.
- Estudio de las pruebas que se le deben realizar a los componentes.

El presente documento consta de tres capítulos:

Capítulo 1 Fundamentación Teórica: Este capítulo contiene una base teórica para entender el problema planteado. Se describen los conceptos fundamentales para el dominio del problema, se muestra el estudio de aplicaciones Applet y Middleware además de hacer referencia a las tendencias y tecnologías actuales que se usaron.

Capítulo 2 Descripción de la Solución Propuesta: Se presentará el modelo de dominio, los requerimientos funcionales y no funcionales. Mostramos los principales procesos a través de casos de uso, los actores que intervienen, sus relaciones y una descripción de cada uno de ellos.

Capítulo 3 Implementación y Prueba: Se muestra el modelo de implementación, el modelo de despliegue y las técnicas usadas durante la implementación. Además del diseño de los casos de prueba.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

El surgimiento de las tarjetas inteligentes (SmartCards) se remonta a Europa en la década de los años 70. Los adelantos tecnológicos sucedidos a lo largo de estos años han propiciado un auge en el desarrollo y utilización de las tarjetas inteligentes, las cuales ya no sólo se utilizan en el campo de la telefonía celular o en el comercio electrónico. El número de aplicaciones para tarjetas inteligentes en general, va en un aumento constante, y abarca áreas muy diversas. Algunos ejemplos típicos se citan a continuación:

- *Electronic Purse o Electronic Wallet o Monedero Electrónica (ePurse y eWallet)*: esta aplicación se utiliza como dinero electrónico. Se puede fijar un monto de dinero inicial, sobre el cual se puede realizar operaciones de débito, crédito o consulta, y puede ser utilizado para el pago o cobro de servicios o bienes. Típicamente lleva asociado algún sistema de seguridad (por ejemplo un PIN), para evitar la posibilidad de fraude.
- *Transacciones Seguras:* Ya sea a través de cajeros automáticos o de Internet, las tarjetas inteligentes proveen un nivel de seguridad muy superior al de las tarjetas magnéticas comunes o los sistemas basados en contraseñas o cookies, ya que es normal que incluyan un API de criptografía fuerte.
- *Identificación Digital / Firma Digital*: este tipo de aplicaciones se utiliza para validar la identidad del portador de la tarjeta, o para poder certificar el origen de ciertos datos. Normalmente se basan fuertemente en las primitivas criptográficas del API y/o las que están implementadas en hardware.
- *Programas de Lealtad*: Este tipo de aplicación sirve a las empresas que ofrecen servicios preferenciales para clientes frecuentes para poder validar la identidad del cliente, y para descentralizar la información. Suponiendo que se tiene un sistema de puntos acumulables, en el cual participan varias empresas, esto simplifica mucho el

tratamiento de los datos, evitando tener que compartir una gran base de datos o tener que realizar réplicas de las distintas bases.

- *Sistemas de Prepago*: En estos sistemas, un cliente "carga" su tarjeta con una cierta "cantidad" de servicio, la cual va siendo decrementada a medida que el cliente hace uso del servicio. Este puede variar desde telefonía celular hasta TV cable, pasando por acceso a sitios web o transporte público.
- *Health Cards o Tarjeta de Salud*: En algunos hospitales ya se está implementando un sistema de identificación de pacientes y almacenamiento de los principales datos de la historia clínica de los mismos en tarjetas inteligentes para agilizar la atención. Actualmente la capacidad de almacenamiento es muy limitada, pero en un futuro se podría almacenar toda la historia clínica (incluidas radiografías y similares) en una Tarjeta Inteligente. La sección 4 presenta el caso de estudio desarrollado; el mismo consta de la implementación de una Health Card.
- *Control de Acceso y de Asistencia*: Ya hay varios sistemas de control de acceso y asistencia implementados en base a SmartCards o a iButtons.

En la actualidad algunos países han comenzado a utilizarla como documento de identificación debido a la seguridad tanto física como lógica que brindan las tarjetas inteligentes, almacenando en cada una de las tarjetas información personal referente a un individuo, así como, de algunos servicios que brindan dichas naciones.

Algunos fabricantes proclaman que sus tarjetas son físicamente inviolables, y que disponen de numerosas defensas contra ataques físicos, como lo son, detección de ciclos de reloj anormales en frecuencia, microprobing, retiro de la cubierta de resina epoxi o exposición del microprocesador a luz ultra violeta.

Sin embargo, hay claros ejemplos en los que se ha logrado obtener, a través de diversos medios, la información contenida en una Tarjeta Inteligente. En la Universidad de Cambridge, un grupo de investigadores han desarrollado varios métodos de extracción de información protegida dentro de una Tarjeta Inteligente, y los resultados que plantean no son muy alentadores. Según la clasificación de los posibles atacantes propuesta por IBM, un atacante de clase I (que podría ser un estudiante de grado de

Ingeniería Eléctrica) con menos de US$ 400 de equipamiento, podría lograr extraer información de una Tarjeta Inteligente.

De todos modos, los ataques registrados como exitosos sucedieron hace ya algún tiempo, y según los fabricantes, los componentes son cada vez más seguros.

En este capítulo se realiza un estudio acerca de implementaciones que actualmente manejan tecnologías que presentan puntos comunes a la nuestra. Se brinda información sobre aplicaciones (Applet) que actualmente corren en tarjetas inteligentes así como una conceptualización de la definición Middleware. Además, se exponen las tecnologías usadas para el desarrollo de la solución.

1.2 Conceptos Fundamentales Asociados al Dominio del Problema

1.2.1 Cédula de Identificación Electrónica

La cédula de identidad es un documento que acredita la identidad de una persona. Es de carácter personal e intransferible, y constituye el documento principal de identificación para los actos civiles, mercantiles, administrativos y judiciales, y para todos aquellos casos en los cuales su presentación sea exigida por la ley. El gobierno de la República Bolivariana de Venezuela, con el fin de mejorar la eficiencia de los servicios de identificación de ciudadanos y responder a las necesidades que impone la creciente informatización del mundo moderno, concibe un nuevo documento de identificación, la Cédula de Identidad Electrónica (CIE), la cual está constituida por un chip que tiene alojado los Applets que permiten la gestión de la información que se almacena.

1.2.2 Applet

Es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web. El Applet debe ejecutarse en un contenedor, que lo proporciona un programa anfitrión, mediante un plugin, o en aplicaciones como

teléfonos móviles que soportan el modelo de programación por Applets. (thefreedictionary Applet, 2008)

Los Applets, según el contexto del trabajo, son aplicaciones implementadas utilizando la tecnología JavaCard y son ejecutadas dentro de las tarjetas inteligentes. El Applet que gestiona la información de las entidades externas contenida en la CIE, se ejecuta en el contexto del JavaCard Runtime–Environment (JCRE). Este Applet es el encargado de ejecutar las operaciones que maneja, mediante los comandos APDU que le son enviados, retornando los resultados mediante APDU de respuestas.

1.2.3 APDU

Unidad de datos de protocolo de aplicación (Application Protocol Data Unit), es la unidad de comunicación entre un lector y una tarjeta. Su estructura está definida en el estándar ISO 7816, existiendo dos tipos de categorías de APDU, APDU Command (Comando APDU) y APDU Response (APDU Respuesta).

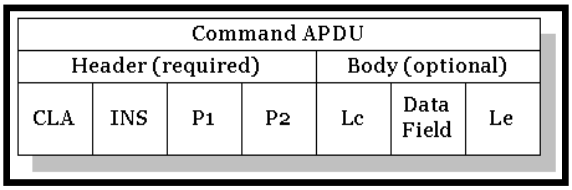


Figura 1: Estructura Comando APDU.

Command APDU		
Campo	**Tamaño**	**Descripción**
CLA	1 byte	Clase de instrucción. Indica la estructura y el formato.
INS	1 byte	Código de instrucción. Especifica la instrucción del comando.
P1	1 byte	Parámetros de la instrucción. Proveen más información sobre la instrucción.
P2	1 byte	
LC	1 byte	Número de bytes en el Data Field del APDU.

Data Field	LC bytes	Secuencia de bytes con información.
LE	1 byte	Cantidad máxima de bytes esperados como respuesta.

Tabla 2: Descripción del Comando APDU.

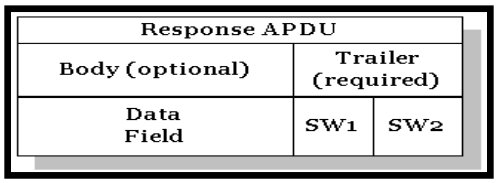


Figura 2: Estructura APDU Respuesta.

Response APDU		
Campo	**Tamaño**	**Descripción**
Data Field	Hasta LE bytes	Secuencia de bytes con información.
SW1	1 byte	Status Word (palabra de estado). Denotan el estado del procesamiento del comando en la tarjeta.
SW2	1 byte	

Tabla 3: Descripción del APDU Respuesta.

1.2.4 Middleware

Software que funciona como una conversión o capa de traducción. Soluciones middleware personalizadas se han desarrollado durante décadas para permitir a una aplicación comunicarse con otra, ya sea que se ejecuta en una plataforma diferente o viene de un proveedor distinto. (thefreedictionary Middleware, 2008)

El middleware implementado, sirve para aislar las operaciones de interaccion con el Applet contenido en la CIE, brindando una interpretación más adecuada de las funcionalidades que realiza.

1.3 Necesidad de la Implementación de una Aplicación Applet y Middleware

La introducción de la cédula electrónica de identificación en la República Bolivariana de Venezuela trae consigo, que Instituciones del país se interesen en almacenar información referente a los servicios que las mismas brindan a las personas. Por esta razón se hace necesario el desarrollo de una aplicación (Applet) que permita gestionar dicha información que estará contenida dentro de la tarjeta electrónica, además de un Middleware que permita la comunicación entre un ordenador y los distintos lectores de las tarjetas inteligentes.

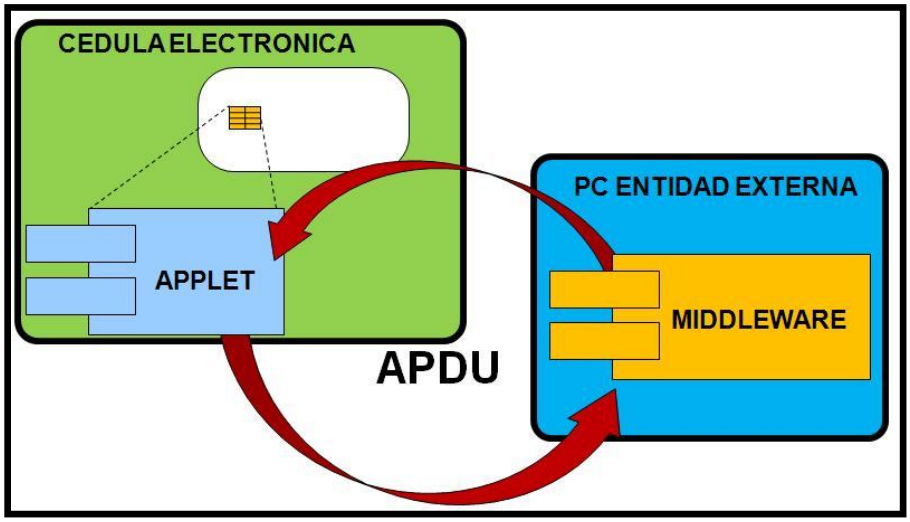


Figura 3: Diagrama Comunicación Middleware – Applet.

1.4 Tarjetas Inteligentes utilizadas en la identificación

Las tarjetas inteligentes como mecanismos de identificación son utilizadas actualmente en varios países del mundo, en los cuales su forma de uso es diversa. La implantación de sistemas de identificación nacional basados en la utilización de tarjetas inteligentes ha permitido que se reúnan una serie de experiencias, las cuales constituyen un buen punto de partida para aquellos países que pretendan aplicar esta tecnología. En este epígrafe analizaremos algunos de los ejemplos más significativos.

1.4.1 DNI Electrónico de Finlandia

Finlandia fue el primer país europeo en emitir tarjetas de identidad electrónica a finales de 1999. La tarjeta FINEID lleva impreso en el frente la información relativa a la tarjeta y el ente emisor de la misma y en el reverso: Identificación de la entidad emisora, datos de la tarjeta y del ciudadano en formato OCR, código de barras. Sin embargo, salvo el nombre, los otros datos personales no se almacenan electrónicamente en la tarjeta. Cada tarjeta tiene dos certificados, uno para autenticación y cifrado y el otro para firma electrónica. Cada uno se utiliza con un PIN diferente. Si el solicitante lo desea, puede incluir información médica en la tarjeta FINEID, con lo cual no necesita tener una tarjeta médica separada (denominada KELA).

A pesar de que la tarjeta FINEID lleva funcionando desde el 2000 y de la variedad de servicios a que pueden acceder con ella los ciudadanos (el Population Register Centre (PRC) ha establecido la marca para identificarlos fácilmente) y las empresas, su crecimiento está siendo muy lento. Por ejemplo, desde 2000 hasta mediados del 2003 sólo se habían emitido 16.000 tarjetas. (Como referencia, la población total del país en el 2003 era de 5,2 millones de personas).

Además de los certificados incluidos en las tarjetas de identidad el PRC emite certificados a los ciudadanos para utilizar en tarjetas Visa Electrón (estas tarjetas, que dan los mismos servicios que la de identidad y además servicios bancarios, se solicitan en una sucursal bancaria) y certificados para utilizar con la tarjeta SIM de un teléfono móvil (el ciudadano debe comprar la tarjeta SIM al operador móvil, y registrarla en el departamento de policía.

1.4.2 DNI Electrónico de España

La tarjeta de identidad de España es una tarjeta de tamaño ID-1 (definida en la ISO/IEC 7816-1), construido de policarbonato que incorpora un chip con información digital. El documento de identificación puede ser usado de diferentes maneras, para identificarse presencialmente o de forma electrónica a través de los certificados contenidos en ella,

también puede ser usado para la firma electrónica de documentos. La tarjeta puede ser usada en el sector privado para: acceder a sitios WEB, firma de contratos, verificación de autenticidad de documentos y en el sector público para la declaración anual de impuestos, interacción con la Administración Pública para la obtención de formularios, servicios en línea, y registros criminales.

La tarjeta contiene un certificado de firma electrónica cuya vigencia es de 2,5 años o hasta la renovación física de la tarjeta criptográfica. El plazo normal de validez de las tarjetas es 5 años, 10 años o permanente según la edad del solicitante. La renovación de los certificados de usuario, con o sin renovación de la tarjeta, implica el cambio de claves.

1.4.3 DNI Electrónico de Estonia

En enero de 2002, aprobado por el parlamento, comenzó la emisión de tarjetas de identidad electrónica. Son tarjetas de PC tamaño ID-1 construido de policarbonato, que cumplen con las normas ISO 7810, 7816 e ICAO.

Chip 16k, criptografía RSA 2048 bits. La tarjeta lleva impreso en el frente el nombre del ciudadano, código de identificación nacional, fecha de nacimiento, sexo, ciudadanía, número de serie de la tarjeta, fecha de validez, fotografía y la firma y en el reverso: lugar de nacimiento, fecha de emisión, permiso de residencia u otras informaciones si aplica.

Cada tarjeta contiene dos certificados cualificados con códigos PIN diferentes. Uno se usa para autenticación y cifrado, el otro, para firmas digitales con validez legal equivalente a la firma manuscrita. Los certificados son válidos durante 1100 días (aproximadamente tres años). La única información personal que consta en los certificados es el nombre del titular y su número de identificación nacional único, ambos considerados de acceso público en Estonia.

El certificado de autenticación contiene, además del nombre y número de identificación del titular, una dirección de correo electrónico asignada por la Administración con el formato "nombre.apellido_NNNN@eesti.ee". Esta dirección está pensada para comunicaciones entre ciudadano y administración, aunque puede usarse también entre particulares. El titular puede cifrar sus correos electrónicos y/o firmarlos digitalmente con la clave correspondiente a su certificado de autenticación, aunque sin el compromiso legal que implica el certificado de firma. La tarjeta puede ser utilizada para: servicios públicos y privados, banca electrónica y pago de impuestos.

1.4.4 Proyecto de Taller V de la República del Uruguay "Programación de JavaCard"

El Instituto de Computación de la Facultad de Ingeniería de la Universidad de la República del Uruguay, se encuentra perfeccionando una solución ya desarrollada por ellos mismos, para brindar el control de la información referente a las "historias clínicas" en una Tarjeta inteligente, que por consenso denominan la HealthCard (Tarjeta de Salud). En principio se basan en conceptos muy similares a los usados en el presente documento.

Desarrollaron un Applet para brindar una estructura de la información que se encontraría almacenada en la aplicación, siguiendo para esto el estándar ISO 7816.

Desarrollaron una herramienta que le denominaron *proxy,* que actúa como un middleware el cual "wrappea" los comandos que le son enviados al Applet de la HealthCard, si viendo también como parser del árbol de información que es creado para brindar una estructura jerárquica, en formato Simple Markup Language (SML), con todos los elementos que se desean gestionar en el dominio de las historias clínicas.

En principio con esta solución encontramos una cierta similitud en el proceso y el flujo que proponemos para gestionar la información de las Entidades Externas en la Cédula de Identificación Electrónica.

1.5 Tendencias Tecnológicas

1.5.1 Tecnología en Tarjetas Inteligentes

1.5.1.1 SmartCard

Es un dispositivo del tamaño de una tarjeta de crédito, el cual almacena y procesa información mediante un circuito de silicio embebido en el plástico de la tarjeta de acuerdo con el estándar ISO / IEC 7810 y ISO / IEC 7816 – 2 (ver Figura 4).

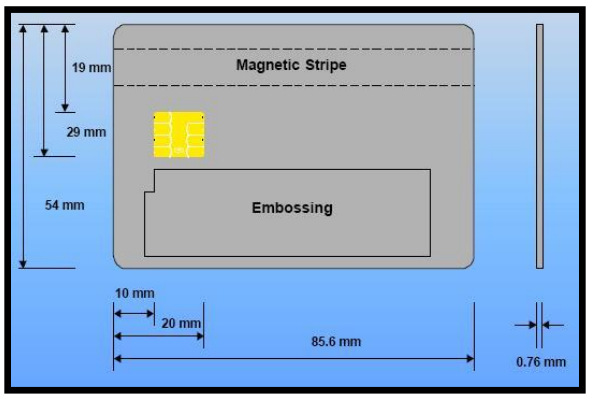


Figura 4: Características físicas de una SmartCard.

Una tarjeta inteligente contiene un microprocesador de 8 Bytes con su CPU, su RAM y su ROM, su forma de almacenamiento puede ser EPROM o EEPROM (ver Figura 5), el programa ROM consta de un sistema operativo que maneja la asignación de almacenamiento de la memoria, la protección de accesos y maneja las comunicaciones. El sendero interno de comunicación entre los elementos (BUS) es totalmente inaccesible desde afuera del chip de silicio mismo por ello la única manera de comunicar está totalmente bajo control de sistema operativo y no hay manera de poder introducir comandos falsos o requerimientos inválidos que puedan sorprender las políticas de seguridad. Las dimensiones y ubicación de los mismos están especificadas en el estándar ISO 7816 - 2.

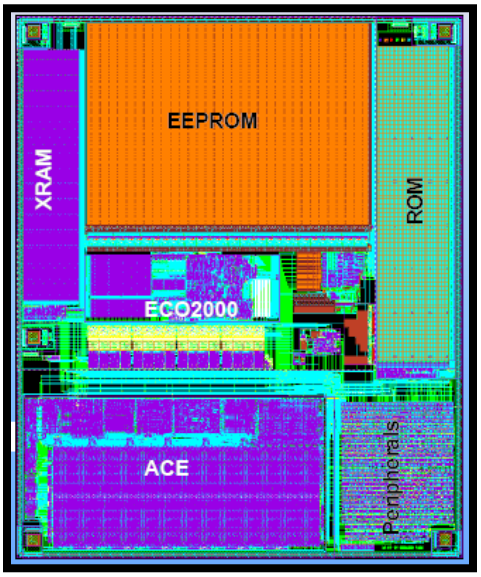


Figura 5: Distribución física de los tipos de memorias de un chip de SmartCard.

La interfaz de comunicación de las tarjetas inteligentes, está hecha para comunicarse con un dispositivo que acepte tarjetas (CAD - *Card Acceptance Device*) a través de un conjunto de 8 pines.

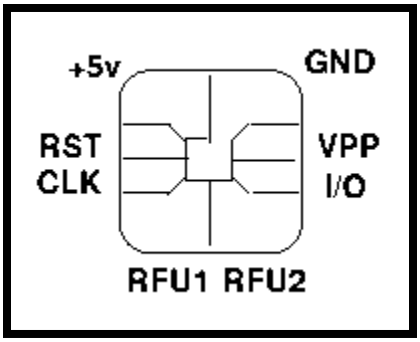


Figura 6: Interfaz de Comunicación de las SmartCard.

- **+5V** - **GND**: suministro de energía
- **I/O**: datos
- **RST**: reset
- **CLK**: señal del reloj (lo usual es < 5MHz)
- **VPP**: señal usada para suministrar energía a algún área en particular de la tarjeta, o para borrar la memoria no-volátil de la tarjeta. Por esencia la idea es que es controlable por software.
- **RFU1** - **RFU2**: reservados para uso futuro

La tarjeta sólo reacciona a los requerimientos de datos externos, nunca inicia por sí sola una comunicación. Todo el protocolo de comunicaciones, dimensiones, resistencia, etc, está claramente establecido en el estándar ISO-7816.

Dentro de la categoría de SmartCards con microprocesador se encuentran las llamadas JavaCards o Java SmartCards. Una JavaCard es una SmartCard capaz de ejecutar programas desarrollados en Java. En pocas palabras, una Java SmartCard es una tarjeta con microprocesador que puede ejecutar programas (llamados Applets) escritos en un subconjunto del lenguaje Java.

1.5.1.2 JavaCard

Es una tecnología que permite ejecutar de forma segura pequeñas aplicaciones Java (Applets) en tarjetas inteligentes y similares dispositivos empotrados. JavaCard da al usuario la capacidad de programar aplicaciones que se ejecutan en la tarjeta de modo que ésta tenga una funcionalidad práctica en un dominio de aplicación específico (pe. identificación, pago, etc.). Se usa ampliamente en las tarjetas SIM (utilizadas en teléfonos móviles GSM) y en tarjetas monedero electrónico.

A nivel de lenguaje, JavaCard es un subconjunto de Java: todas las construcciones del lenguaje JavaCard existen en Java y se comportan de la misma manera. Esto va hasta el punto de que, como parte de un ciclo estándar de desarrollo, un Applet JavaCard se compila en un archivo de clase Java (.class) por un compilador Java normal, sin ningún tipo de opción especial (aunque el fichero compilado será procesado posteriormente por herramientas específicas para la plataforma JavaCard).

1.5.1.3 JavaCard Runtime Environment (JCRE)

El JCRE comprende la máquina virtual de JavaCard (JCVM) junto a las clases y servicios definidos en el Application Programming Interface (API). Sobre este ambiente se ejecutan los Applets que se desarrollan. La JCVM se diferencia principalmente de una JVM normal en que el tiempo de vida de la misma es igual al tiempo de vida de la tarjeta, por lo cual los objetos mantienen sus estados entre dos sesiones con una terminal. Es responsabilidad del JCRE garantizar este comportamiento. Cuando se

retira la tarjeta del terminal, se asume que se está ejecutando en un ciclo de reloj de período infinito. Otras diferencias entre ellas son las limitaciones en los tipos de datos manejados y los requerimientos de hardware para la ejecución.

Para poder comprender cómo funciona una JavaCard, hay que tener en cuenta que al realizar la especificación de la plataforma, Sun se apegó al estándar ISO 7816, el cual establece, entre otras cosas, la forma de comunicación entre una tarjeta inteligente y una terminal. De acuerdo al ISO 7816, el intercambio de información y comandos entre la tarjeta y el terminal se realiza a través de APDUs (Application Protocol Data Units), los cuales son paquetes de información con un formato específico.

De acuerdo al estándar, las tarjetas inteligentes nunca inician la comunicación con el terminal, sino que sólo responden a los comandos que éste les envía.

Se puede decir que un Applet comienza su ciclo de vida al ser correctamente cargado en la memoria de la tarjeta, link – editada y preparada para su correcta ejecución. Una vez registrada en el JCRE un Applet está en condiciones de ejecutar. Este Applet normalmente existe durante el resto de la vida de la tarjeta.

La clase `javacard.framework.Applet`, es una clase abstracta provista en el framework de desarrollo utilizado (Developer Suite), donde se definen cuatro métodos públicos que son utilizados por el JCRE para hacer funcionar las aplicaciones.

- *Método install(byte[], short, byte).*

  Este método es invocado por el JCRE antes de crear una instancia del Applet en la tarjeta. La implementación usual de este método es llamar al constructor de la clase, que normalmente es privado, crear todos los objetos que el Applet necesitará para su ejecución, y por último registrar el Applet con el método `register`(). No es estrictamente necesario crear todos los objetos en el método `install`(). Sin embargo es una buena práctica de programación pues garantiza la obtención de toda la memoria necesaria, evitando quedar más adelante (tal vez una vez entregada al cliente) en un estado inválido por falta de memoria.

En caso de que se produzca una excepción durante la ejecución del método `install`(), el JCRE es responsable de realizar las actividades de limpieza pertinentes. Una vez finalizado el método, el JCRE marca al applet como listo para ser seleccionado (ver método `select`()).

- *Método select().*

Este método es invocado por el JCRE como consecuencia de la recepción a un SELECT APDU. Este APDU, cuyo formato está definido en el ISO 7816, contiene el Application IDentifier (AID) del Applet a seleccionar. El AID es una secuencia de entre 5 y 16 bytes, que identifica de forma única una aplicación para tarjetas inteligentes, de acuerdo al ISO 7816, y es la misma ISO la que otorga los AIDs. El formato de un AID se puede ver en la siguiente tabla:

Application Identifier (AID)	
National registered application provider **RID**	**Proprietary application identifier extension** **PIX**
5 bytes	Entre 0 y 11 bytes

Tabla 4: Descripción de los AID.

Cada empresa que produce Applets debe solicitar a la ISO su propio RID, y a su vez maneja sus PIX (en forma arbitraria) para identificar sus aplicaciones y packages. Una vez que el JCRE recibe un SELECT APDU, si hay algún Applet seleccionado, invoca a su método `deselect`() (ver método `deselect`()) y luego invoca al método `select`() del Applet cuyo AID fue especificado. El Applet puede, por distintas razones, declinar la selección, en cuyo caso el JCRE es responsable de responder adecuadamente al CAD.

En caso de que la selección se realice sin inconvenientes, se pasa el SELECT APDU al método `process`() (ver método `process`()) del Applet seleccionado para que lo procese y devuelva al CAD la información que sea pertinente.

- *Método process(APDU).*

Cuando llega un APDU el JCRE invoca este método del Applet seleccionado, pasándole como parámetro el COMMAND APDU recibido. Dentro de este método, el Applet identifica el comando asociado al APDU y los parámetros, si los hay, y los procesa de acuerdo al protocolo que se haya definido para la interacción entre el Applet y la aplicación terminal. En caso de que la ejecución finalice correctamente, el Applet sólo debe encargarse de cargar en el RESPONSE APDU la información que va a devolver, si la hay. El JCRE es responsable de resetear los SW del RESPONSE APDU al valor especificado para ejecución exitosa (0x9000, de acuerdo a lo especificado en el ISO 7816).

Durante el proceso de un APDU, el Applet puede levantar una ISOException con los SW apropiados, la cual, si no es atrapada por el código del Applet, es atrapada por el JCRE, quien se encarga de generar el RESPONSE APDU correspondiente.

- *Método deselect().*

Este método es invocado por el JCRE para avisar al Applet que está actualmente seleccionado, que va a dejar de estarlo. Esto sucede cuando el JCRE recibe un SELECT APDU (aún cuando el AID del Applet a seleccionar coincida con el del Applet seleccionado). Esto brinda al Applet la oportunidad de realizar las tareas de limpieza que sean necesarias para quedar en un estado consistente.

1.5.1.4 JavaCard Runtime Environment (JCRE): Shareable Interfaces

Uno de los aspectos claves a tener en cuenta en el desarrollo de aplicaciones JavaCard es la seguridad de los datos que las mismas manejan, debido a que los mismos son potencialmente muy sensibles (información de Entidades Externas, PINs, datos biométricos, entre otros).

Debido a ello, la especificación del JCRE brinda mecanismos de aislamiento de las aplicaciones que coexisten en una misma tarjeta. Esto permite a los desarrolladores programar con la tranquilidad de que sus datos están a salvo dentro de la tarjeta. (Sun Microsystems)

Sin embargo, hay algunas aplicaciones que necesitan compartir información y para ello el JCRE provee ciertos mecanismos de interacción entre Applets, para permitir que las mismas compartan ciertos datos o servicios, definidos por el desarrollador. De esta forma se logra una mayor flexibilidad en el desarrollo de Applets, y se abre el campo a nuevas aplicaciones basadas en las interacciones inter-Applet.

La especificación 2.1.1 del JCRE (la más actual a la fecha) define (al igual que la 2.1) interfaces para permitir que los Applets exporten ciertos objetos a través de las protecciones que impone el JCRE. Dichas interfaces presentan ciertas limitaciones y problemas de seguridad. ([SUN 2], Sun Microsystems, 1999.) (SmartCard Technology (Smartcard '99))

El mecanismo de *Object Sharing* propuesto implica que para exportar cierto servicio a través del firewall se debe definir una interfaz que extienda la interfaz Shareable y luego implementar esta nueva interfaz en la clase que ofrecerá los servicios (Figura 7). A una instancia de esta clase se le llamará *Shareable Interface Object* (SIO). Los métodos de un SIO que estén declarados en una *Shareable Interface* (SI) pueden ser invocados por objetos que existan en otros contextos.

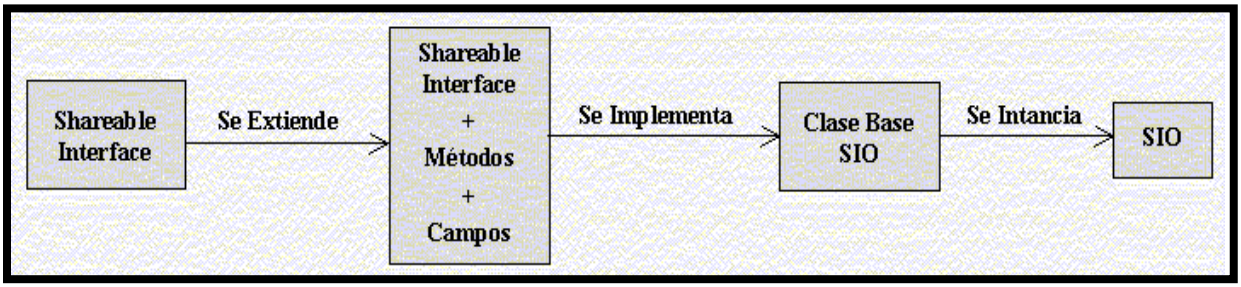


Figura 7: Interfaz Compartida.

El mecanismo de object sharing especificado en el JCRE 2.1.1 se basa en la definición de *shareable interfaces* (SIs). Se extiende la interfaz javacard.framework.Shareable, definiendo así nuevas shareable interface, y se desarrollan las clases que las implementan. Las instancias de estas clases se denominan *Shareable Interface Objects* (SIOs). Estos objetos son los que permiten la interacción entre Applets, ya que los métodos declarados en una SI pueden ser ejecutados a través del firewall, en el contexto del servidor, logrando de esta forma acceso a los datos y servicios que éste

brinda. En el caso de una aplicación de lealtad, por ejemplo, la SI podría definir métodos para verificar la identidad del mismo.

Los Applets que actúan como servidor implementan un método que permite exportar los SIOs necesarios para brindar servicios a otros Applets (clientes). El método a implementar es getShareableInterfaceObject() (Figura 8).

Dichos clientes acceden al SIO que requieren, a través de un método del JCRE que recibe como parámetros el AID del Applet servidor, y un byte para especificar opciones. El método a invocar es JCSystem.getAppletShareableInterfaceObject(). Este método invoca al método getShareableInterfaceObject() del Applet indicado, que devuelve una referencia al SIO solicitado, o NULL, en base al AID del cliente, el cual recibe como parámetro.

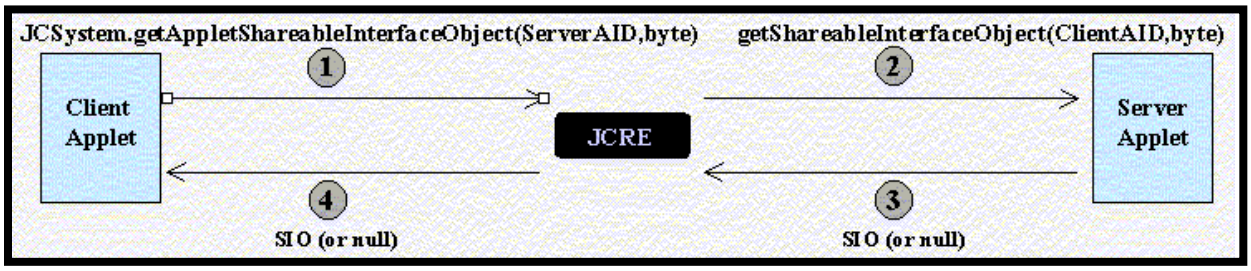


Figura 8: Método de la Interfaz Compartida.

Con la referencia al SIO, el cliente puede invocar métodos sobre el mismo, sin que el firewall se interponga, y de esta forma obtener los servicios o datos necesarios del servidor.

Este mecanismo presenta los siguientes problemas:

- La autenticación del cliente se basa en su AID, lo cual implica que:

  - La cantidad (e identidad) de clientes que atenderá un servidor en su vida útil está determinada al momento de embarcarlo en la tarjeta.

  - Si la tarjeta fuese comprometida, se podría utilizar un Applet malicioso instalado con el AID del cliente para extraer datos del servidor.

- Es común la implementación de varias interfaces en un mismo objeto (normalmente el propio Applet), lo que permite a un cliente malicioso realizar un casteo de una interfaz a otra, obteniendo así acceso ilícito a datos o servicios de la misma.
- Los métodos del SIO no pueden recibir objetos (que no sean a su vez SIOs como parámetros, ya que el firewall no permite la ejecución de ningún método sobre los mismos).

1.5.1.5 JavaCard Runtime Environment (JCRE): Mecanismos de Seguridad Lógica

Es una tecnología que refuerza la seguridad más allá de las protecciones que tiene la JCVM por sí misma. Los chequeos que ésta implica se realizan durante la ejecución de la aplicación.

Se llamará *contexto* de un Applet al conjunto de objetos que pertenecen a dicho Applet. El JCRE también tiene su propio contexto, el cual tiene la misma estructura que el de un Applet, teniendo además permisos especiales que le permiten realizar algunas operaciones a nivel de sistema que no son permitidas a los Applets.

El Applet firewall particiona el sistema de objetos de las JavaCards en los diferentes contextos de cada una de las Applets que hay instalados en el dispositivo. El firewall se puede visualizar como la barrera que existe entre un contexto y los demás (Figura 9). (Universidad de la República del Uruguay, 2004)

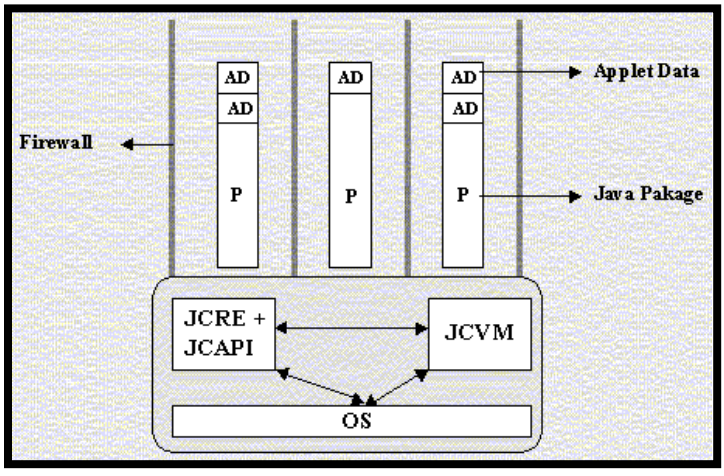


Figura 9: Descripción del Firewall.

1.5.2 Tecnología Biométrica. Match on Card (MoC)

Las tecnologías biométricas han ido fortaleciendo los mecanismos de autenticación al comparar la plantilla biométrica almacenada con la plantilla biométrica capturada al momento de la comparación (*plantilla en vivo*). En el caso de las tarjetas inteligentes, esta comparación se hace dentro de la tarjeta lo cual requiere una capacidad de procesamiento interno que dependerá de la complejidad de la información biométrica a comparar (huellas, iris, facial, etc.) y de los algoritmos usados.

Match on Card (MoC, *siglas en inglés):* El proceso biométrico utilizando la tecnología MoC se divide en dos funciones a realizar: *el enrolamiento y verificación* de las huellas digitales en la tarjeta. La plantilla biométrica es almacenada dentro de la tarjeta, que también realiza la comparación con la plantilla en vivo. Por tanto, se necesita una capacidad de procesamiento interno como la del microprocesador de una tarjeta inteligente, lo cual conlleva al uso de un sistema operativo que ejecute las aplicaciones de comparación necesarias. (Mendoza, 2008) (Precise Biometrics, 2005)

1.5.3 Estándares utilizados en Tarjetas Inteligentes

1.5.3.1 PC/SC

PC/SC es un workgroup cuyo objetivo es el de promover una especificación estándar, que asegure la interoperabilidad entre tarjetas inteligentes, lectores de tarjetas inteligentes y computadoras. PC/SC desarrolló una especificación independiente de la plataforma, que puede ser implementada sobre cualquier sistema operativo. Fue construido sobre los estándares actuales de SmartCard (ISO 7816), definiendo interfaces de bajo nivel para dispositivos y APIs independientes del dispositivo.

La especificación actual es la PC/SC Specification 1.0. La misma incluye los siguientes temas, entre otros:

- Provee la arquitectura del sistema y de sus componentes

- Detalla las características y requerimientos de compatibilidad de las tarjetas y los dispositivos
- Presenta una discusión con consideraciones de diseño para los dispositivos, y recomendaciones de implementación
- Describe los componentes con los que debe contar el sistema
- Presenta consideraciones de diseño para desarrolladores de aplicaciones, indicando cómo hacer uso de los componentes.

1.5.3.2 Estándares ISO/IEC 7816

Estándares internacionales para tarjetas con circuito integrado (tarjetas inteligentes). El objetivo de estos estándares es lograr la interoperabilidad entre distintos fabricantes de tarjetas inteligentes y lectores de las mismas, en lo que respecta a características físicas, comunicación de datos y seguridad. Estos estándares son basados en los ISO 7810 e ISO 7811, los cuales definen características físicas de tarjetas de identificación.

Los estándares ISO/IEC van desde el 7816-1 hasta el 7816-11, pero para la implementación de nuestra solución nos centramos en el estándar 7816-4 el cual define:

- El contenido de los pares comando-respuesta que se intercambian a nivel de interfaz.
- Estructuras para aplicaciones y datos en la tarjeta.
- La estructura y contenido de los caracteres históricos de la respuesta de Reset, los cuales describen las características de operación de la tarjeta inteligente.
- Métodos para extracción de objetos y elementos de datos de la tarjeta.
- Mecanismos para la identificación y direccionamiento de aplicaciones en la tarjeta.
- Estructuras de archivos y métodos de acceso.
- Arquitectura de seguridad para derechos de acceso a los archivos y datos en la tarjeta.
- Comandos orientados a objetos de datos.

- Métodos de acceso a los algoritmos que procesa la tarjeta inteligente (no describe los algoritmos).
- Métodos para el intercambio seguro de mensajes.

El estándar 7816-4 es independiente de la tecnología de interfaz física que se implemente en la tarjeta inteligente. (Estándar ISO/IEC 7816, 2006)

1.5.3.2.1 Estructura para aplicaciones y datos

Existen en la tarjeta dos categorías de estructura: los ficheros dedicados (DF siglas en inglés) y los ficheros elementales (EF siglas en inglés). Los DF están compuestos por grupos de ficheros, que pueden ser DF o EF y los EF son ficheros para almacenar datos y no contienen más ningún fichero. Existen dos categorías de EF.

- EF que almacena información interpretada por la tarjeta para su administración y control.
- EF que almacena información no interpretada por la tarjeta, información que se usa fuera de la tarjeta.

Existen dos tipos de organización lógica de los DF y los EF.

- La figura 10 muestra la jerarquía de los DF con su correspondiente arquitectura. En la organización que se presenta, el DF raíz, es llamado fichero maestro (MF siglas en inglés). Cualquier DF puede ser un grupo de ficheros con o sin jerarquía alguna.

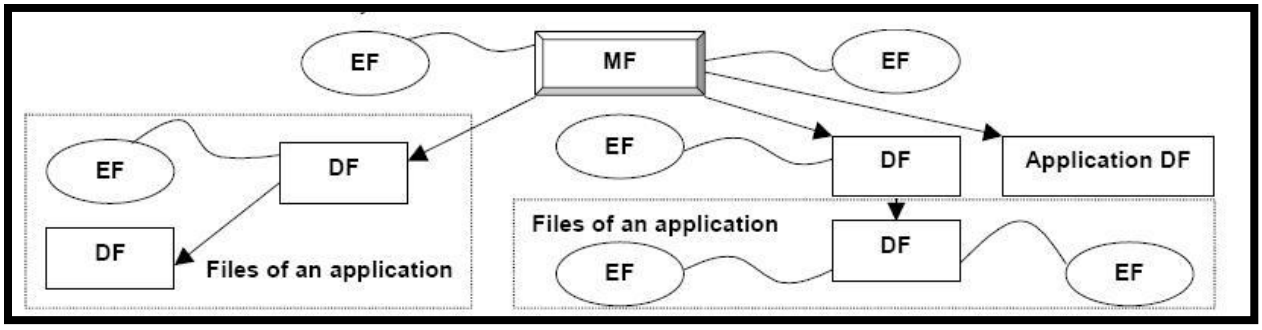


Figura 10: Jerarquía de ficheros.

- La figura 11 muestra DF en paralelo, sin una MF y sin una jerarquía entre ellos. Esta organización soporta que cada DF tenga su propia jerarquía con su correspondiente arquitectura.

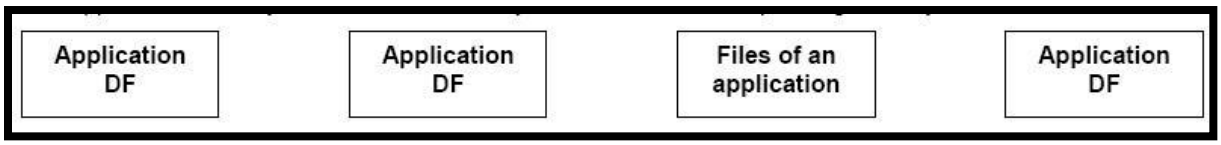


Figura 11: Ficheros.

1.5.3.2.2 Estructura de ficheros

Se definen tres tipos de estructura de los EF.

- Estructura Transparente: El EF es visto como una única secuencia continua de datos accesibles por comandos para el manejo de las unidades de datos. EL tamaño de la unidad de datos depende del EF.
- Estructura de Registros: El EF es visto como una única secuencia continua de registros de identificación individual accesibles por comandos de manejos de registro. El EF define dos atributos:
  - El tamaño de los atributos, que puede ser fijo o variable.
  - La organización de los registros puede ser una estructura lineal o una estructura cíclica.
- Estructura TLV: El EF es visto como un conjunto de objetos de datos accesibles por comandos para el manejo de objetos de datos. El tipo de objetos de datos en el EF están en formato Simple-TLV o Ver-TLV.

Para la referencia de los datos en el EF, la tarjeta soporta al menos una de las cinco estructuras que se muestran en la Figura 12.

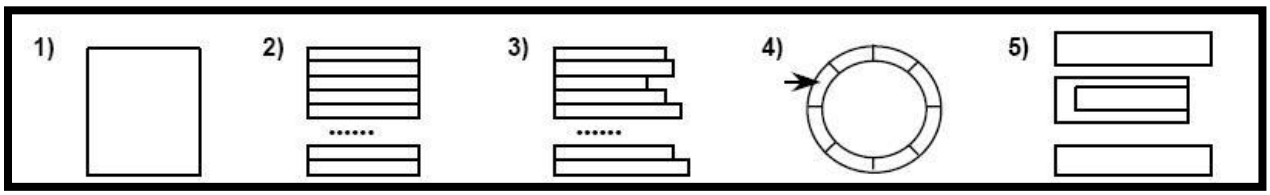


Figura 12: Estructura de los EF.

1) Estructura Transparente.
2) Estructura Lineal con registros de tamaño fijo.
3) Estructura Lineal con registros de tamaño variable.
4) Estructura cíclica con registros de tamaño fijo (la flecha referencia al último registro escrito).
5) Estructura TLV.

1.5.3.2.3 Atributos de Seguridad

Los atributos de seguridad de un archivo depende de su categoría (DF o EF) y definen cuáles acciones se pueden realizar y bajo qué condiciones. Los atributos de seguridad pueden:

- Especificar el estado de seguridad que la tarjeta debe tener antes de acceder a los datos.
- Restringir el acceso a los datos para cierta función (ejemplo, sólo escritura) si la tarjeta está en un estado particular.
- Definir cuáles funciones de seguridad deben ser realizadas para obtener un estado de seguridad específico. (Estándar ISO/IEC 7816, 2006)

1.5.4 Tecnología PKI

1.5.4.1 PKI Infraestructura de Clave Pública

Combinación de hardware y software, políticas y procedimientos de seguridad que permiten la ejecución con garantías de operaciones criptográficas como el cifrado, la firma digital o el no repudio de transacciones electrónicas. El término PKI se utiliza para referirse tanto a la autoridad de certificación y al resto de componentes, como para referirse, de manera más amplia y a veces confusa, al uso de algoritmos de clave pública en comunicaciones electrónicas. Este último significado es incorrecto, ya que no se requieren métodos específicos de PKI para usar algoritmos de clave pública.

Permite a los usuarios autenticarse frente a otros usuarios y usar la información de los certificados de identidad (por ejemplo, las claves públicas de otros usuarios) para cifrar

y descifrar mensajes, firmar digitalmente información y garantizar el no repudio de un envío. (Wikipedia PKI, 2008)

1.5.4.2 PKCS

En criptografía, se refiere a un grupo de estándares de criptografía de clave pública, concebidos y publicados por los laboratorios de RSA en California en cooperación con desarrolladores de sistemas de seguridad en todo el mundo con el fin de acelerar el despliegue de la criptografía de clave pública. A RSA Security se le asignaron los derechos de licenciamiento para la patente de algoritmo de clave asimétrica RSA y adquirió los derechos de licenciamiento para muchas otras patentes de claves. Publicado por primera vez en 1991 como resultado de las reuniones con un pequeño grupo de los primeros adoptantes de tecnologías de clave pública, los documentos PKCS se han convertido en referencia y ampliamente aplicados. Contribuciones de la serie PKCS se han convertido en parte de notas académicas y normas, incluyendo ANSI X9 documentos, PKIX, SET, S / MIME y SSL. (technology, 2002)

Los estándares PKCS van desde el número uno hasta el número quince. Realizaremos un resumen de los utilizados en el desarrollo de nuestro sistema.

- **PKCS 7.**

  Es el estándar de la sintaxis de los mensajes criptográficos, define la sintaxis de varios tipos de mensajes criptográficos protegidos, incluyendo mensajes encriptados y mensajes con firmas digitales. Originalmente la Internet Privacy-Enhanced Mail convirtió a este estándar en especificación segura de correos electrónicos. Pero este no solo fue limitado al correo, sino que se convirtió en la base de los mensajes seguros en sistemas tan diversos como el Secure Electronic Transaction (SET) especificado para transacciones bancarias utilizando pagos por tarjetas. Tras realizar un pedido PKCS# 10 a una Autoridad Certificadora (CA), esta devuelve el certificado en formato PKCS# 7.

- **PKCS 10.**

  Formato de los mensajes enviados a una Autoridad de certificación para solicitar la certificación de una clave pública. Describe la sintaxis de los pedidos

de certificados. Un pedido de certificado consiste de un nombre distinguido, una llave pública, y opcionalmente un conjunto de atributos, todos firmados por una entidad de pedidos de certificación. Los pedidos de certificados son enviados a una autoridad certificadora, la cual transforma el pedido en un certificado X509.

- **PKCS 11.**

  Interfaz de dispositivo criptográfico ("Cryptographic Token Interface" o cryptoki). Define un API genérico de acceso a dispositivos criptográficos. Constituye la definición de un interfaz de acceso a los denominados Tokens criptográficos. Definición de Cryptoki en lenguaje ANSI C.

- **PKCS 15.**

  Estándar de formato de información de dispositivo criptográfico. Define un estándar que permite a los usuarios de dispositivos criptográficos identificarse con aplicaciones independientemente de la implementación del PKCS#11 (cryptoki) u otro API. Constituye una definición de cómo son almacenadas en los Tokens abstracciones criptográficas de alto nivel tales como claves y certificados.

1.5.4.3 ASN.1

Fue desarrollado como parte de la capa 6 (presentación) del modelo de referencia OSI (esta capa define la forma en que los datos serán almacenados en los nodos). Esta notación proporciona un nivel de abstracción similar al ofrecido por lenguajes de programación de alto nivel. La notación ASN.1 fue publicada en la recomendación ITU-T X.208 | ISO/IEC 8824 (diciembre/1987). En 1995 se hicieron revisiones para corregir errores, ambigüedades e incluir nuevas capacidades. Los documentos revisados están contenidos en las recomendaciones de la serie X.680. Es una notación que ofrece un rico conjunto de *tipos de datos* y *constructores* que permiten definir estructuras de datos complejas a partir de tipos simples o primitivos. Al igual que cualquier lenguaje de programación, la notación es especificada utilizando gramática BNF. (technology, 2002)

1.5.4.4 Certificados X509

En criptografía, X.509 es un estándar ITU-T para infraestructuras de claves públicas (PKI). X.509 especifica, entre otras cosas, formatos estándares para certificados de claves públicas y un algoritmo de validación de la ruta de certificación.

En el sistema X.509, una autoridad certificadora (AC) emite un certificado asociando una clave pública a un nombre distinguido particular en la tradición de X.500 o a un nombre alternativo tal como una dirección de correo electrónico o una entrada de DNS. X.509 es la pieza central de la infraestructura de clave pública y es la estructura de datos que enlaza la clave pública con los datos que permiten identificar al titular. Su sintaxis se define empleando el lenguaje ASN.1.

Los certificados x509, pueden estar especificados en varias versiones. En particular la versión 3 define un conjunto de atributos extras, que son denominados *Atributos Extendidos* (ver figura 13), que son definidos como una secuencia de uno o más certificados extendidos, en donde se puede portar información sobre el usuario y proveedor del certificado, llaves públicas, administración de la jerarquía del certificado, entre otros. Cada extensión en el certificado puede ser definida como crítica o no crítica, debe especificar el tipo de extensión, el cual puede ser uno de los estandarizados para este tipo de atributos u otro adicional, y por último el valor de la extensión. Esta triada de información es la que brinda la posibilidad de estandarizar los atributos extendidos a las necesidades del usuario. Es aquí donde se portarán las condiciones de acceso que se necesitan para gestionar la información que almacena la Cédula de Identificación Electrónica, de la República Bolivariana de Venezuela.

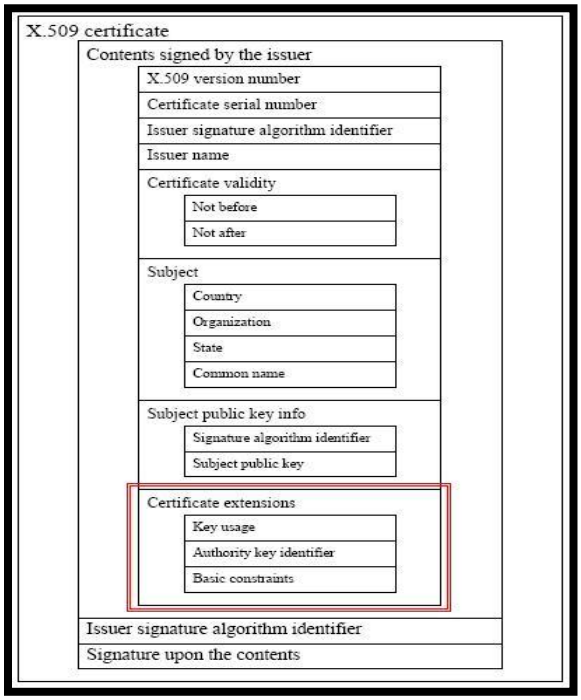


Figura 13: Estructura de Certificado versión 3.

1.5.4.5 Online Certificate Status Protocol (OCSP)

Es un protocolo de Internet utilizado para la obtención de la condición de revocación de certificados digitales X.509. Se describe en el RFC 2560 y está en el seguimiento de normas de Internet. Fue creado como alternativa a la listas de revocación de certificados (CRL), que tratan específicamente de ciertos problemas relacionados con el uso de una infraestructura de clave pública (PKI). La comunicación a través de mensajes OCSP se codifica en ASN.1 y suele ser transmitida a través de HTTP.

1.5.5 Tecnologías de Desarrollo

1.5.5.1 Developer Suite Gemalto

Herramienta que nos brinda un ambiente favorable para el diseño y la implementación de applets, además nos posibilita simular las funcionalidades de los applets antes de ser instalados en las tarjetas inteligentes.

1.5.5.2 UML como lenguaje de modelación visual

El **Lenguaje Unificado de Modelado** (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Actualmente UML es el estándar para el diseño orientado a objetos, ya que es el resultado de la unión de las mejores cualidades de los tres lenguajes existentes que le dieron paso por el trabajo en conjunto de sus autores.

A partir del surgimiento de UML, muchas de las metodologías existentes han sido adaptadas para utilizar este lenguaje, como es el caso de la Metodología de Análisis y Diseño Orientado a Objetos de Sistemas Informáticos en su versión 5.0 y en otras como el Proceso Unificado de Desarrollo se concibió desde sus inicios utilizar UML.

1.5.5.3 UModel Altova

UModel se utiliza para crear e interpretar diseños software mediante la potencia del estándar UML 2.1. Dibuja el diseño de la aplicación y puede generar código para Java o C# a partir de planos, así como que permite realizar ingeniería inversa de programas existentes a diagramas UML claros y precisos para abarcar rápidamente su arquitectura software. Incluso, con la utilización de UModel se puede corregir el código generado o los modelos y completar la ronda produciendo automáticamente nuevos diagramas o regenerando el código.

1.5.5.4 RUP como metodología de desarrollo

El **Proceso Racional Unificado** (RUP, por sus siglas en inglés *Rational Unified Process*) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de

Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico proceso unificado. Proporciona una guía en el orden de las actividades de un equipo, dirige las tareas individuales de los desarrolladores, especifica qué productos deberían ser desarrollados y ofrece criterios para monitorear y medir los productos y actividades del proyecto.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales, los 6 primeros son conocidos como flujos de ingeniería (Modelado del negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba, Instalación) y los tres últimos de apoyo (Administración del Proyecto, Administración de Configuración y Cambios, Gestión de Entorno).

Un proyecto realizado siguiendo RUP se divide en cuatro fases: Inicio (puesta en marcha), Elaboración (definición, análisis, diseño), Construcción (implementación) y Transición (fin del proyecto y puesta en producción). En cada fase se ejecutará una o varias iteraciones (de tamaño variable según el proyecto), dentro de cada una de ellas seguirá un modelo de cascada para los flujos de trabajo con sus respectivas actividades.

RUP se caracteriza por ser un proceso dirigido por casos de uso (o sea que avanza a través de los flujos de trabajo que parten de los casos de uso y estos son el instrumento para validar la arquitectura del software y extraer los casos de prueba), está centrado en la arquitectura (los modelos son proyecciones del análisis y el diseño, constituye la arquitectura del producto a desarrollar) y es iterativo e incremental (durante todo el proceso de desarrollo se producen versiones incrementales del producto en desarrollo, que se acercan al producto terminado).
Sus principales características son:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace, qué, cuándo y cómo).

- Pretende implementar las mejores prácticas en Ingeniería de Software:
  - Desarrollo iterativo.
  - Administración de requisitos.
  - Uso de arquitectura basada en componentes.
  - Control de cambios.
  - Modelado visual del software.
  - Verificación de la calidad del software.

1.5.5.5 Rational Rose

Es la herramienta CASE (Ingeniería de Software Asistida por Ordenador - Computer Aided Software Engineering) desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson) para realizar gráficamente la modelación del sistema. Dicha herramienta cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

Rational Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción. Además, cada integrante del equipo puede modelar sus componentes e interfaces de forma individual y luego unirlos con otros componentes del proyecto, gracias a que cada cual tiene sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue).

1.5.6 Plataforma .NET

1.5.6.1 Microsoft .NET

Es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados

remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados.

El Framework de .Net es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables.

Organiza toda la funcionalidad del sistema operativo en un espacio de nombres jerárquicos de forma que a la hora de programar resulta bastante sencillo encontrar lo que se necesita.

Posee un conjunto de ventajas entre las que se destacan:

- *Código administrado*: El Tiempo de ejecución del Lenguaje Común (CLR, por sus siglas en inglés Common Language Runtime) realiza un control automático del código para que este sea seguro, es decir, controla los recursos del sistema para que la aplicación se ejecute correctamente.
- *Interoperabilidad multilenguaje*: El código puede ser escrito en cualquier lenguaje compatible con .Net ya que siempre se compila en código intermedio o Microsoft Intermediate Lenguaje (MSIL).
- *Compilación just-in-time*: El compilador JIT (Just In Time, nombre que recibe ese tipo de compilación porque se realiza en tiempo de ejecución) incluido en el Framework compila el código intermedio (MSIL) generando el código máquina propio de la plataforma. Se aumenta así el rendimiento de la aplicación al ser específico para cada plataforma.
- *Despliegue:* Por medio de los ensamblados resulta mucho más fácil el desarrollo de aplicaciones distribuidas y el mantenimiento de las mismas. El Framework realiza esta tarea de forma automática mejorando el rendimiento y asegurando el funcionamiento correcto de todas las aplicaciones.

1.5.6.2 Mono .NET

Mono es la implementación libre del CLI (Common Language Infraestructure) y C#, de acuerdo a las especificaciones enviadas a la ECMA para su estandarización.

El Mono incluye el CLI, el cual contiene la máquina virtual que se encarga de cargar las clases, el compilador JIT (Just-in-time) y el garbage collector; todo esto escrito desde cero de acuerdo a las especificaciones Ecma-334.

Adicionalmente Mono cuenta con un catálogo de librerías compatibles con las librerías del .Net Framework, pero además cuenta con una serie de librerías no existentes en el .Net Framework de Microsoft; como el GTK# que permite crear interfaces gráficas nativas del toolkit GTK+, Mono.LDAP, Mono.Posix, etc.

Los objetivos iniciales del proyecto Mono eran implementar en un entorno de software libre para el mundo Unix las especificaciones ECMA, para lo cual se incluye un compilador para C#, un entorno de ejecución CLR y un conjunto de librerías de clase que incluyen las FCL, así como otras añadidas.

1.5.6.3 Lenguaje de programación C#

Aunque para la plataforma .NET es prácticamente posible programar en cualquier lenguaje, el C# es el lenguaje de propósito general diseñado por Microsoft para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros.

Entre sus principales características se destacan:

- *Sencillez*: C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET.
- El código escrito en C# es auto contenido, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera.
- El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile, lo que facilita la portabilidad del código.

- *Orientación a componentes*: La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros).
- *Eficiente*: En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. Para ello basta marcar regiones de código como inseguras (modificador unsafe) y podrán usarse en ellas punteros de forma similar a cómo se hace en C++, lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad de procesamiento muy grandes.

1.6 Propuesta y selección de herramientas

La Cédula de Identidad Electrónica (CIE) es una tarjeta de circuitos integrados el cual tiene especificaciones estandarizadas a nivel internacional. La ISO/IEC 7816 – 4 define cómo serán organizados los comandos que se enviarán a la tarjeta en ámbitos de seguridad, intercambio de datos, niveles de acceso a ficheros, entre otros.

Las entidades externas a la ONIDEX antes de acceder a la información que pueden gestionar en la CIE, deben presentar un certificado X509 generado por una *autoridad certificadora* en el cual se integran datos tanto de la entidad externa como de la autoridad certificadora, todos firmados bajo una llave privada, permitiendo demostrar la autenticidad de la entidad externa.

Para la gestión de la información en la CIE, se va a realizar la implementación de una aplicación (applet) la cual permitirá a entidades externas a la ONIDEX escribir y leer datos dentro de la CIE. El applet se va a desarrollar utilizando tecnología JavaCard, la cual permite implementar aplicaciones que se ejecutan dentro de las tarjetas inteligentes de modo que estas tenga funcionalidades prácticas.

Para implementar el Middleware se utilizará C#, como parte de la familia de los lenguajes de .Net, el cual brinda una gran versatilidad frente a estándares establecidos para los lectores de tarjetas inteligentes PC/SC. La utilización de MONO .NET brinda la posibilidad de ejecución de la solución en múltiples plataformas, cumpliendo de esta forma con los requisitos que se hacen necesarios para la implementación de la actual solución para el control de acceso a la información de las entidades externas a la ONIDEX en la República Bolivariana de Venezuela.

1.7 Conclusiones

- El desarrollo de este capítulo entrega un amplio y profundo estudio, sobre los países que actualmente utilizan como medios de documentación a nivel nacional las tarjetas inteligentes, marcando un amplio desarrollo en el dominio de la identificación personal, brindando además diferentes servicios que requieran de la autenticidad de las personas con la utilización de certificados digitales.
- Se identificaron las numerosas tecnologías y estándares que permiten realizar el modelado y la implementación de la solución propuesta, profundizando en las potencialidades que brindan.
- En este capítulo se aborda ampliamente el por qué de la necesidad de la implementación de un Applet como aplicación establecida para realizar la gestión de la información que estará almacenada en el chip de la tarjeta, dando cumplimiento a las necesidades que las entidades externas a la ONIDEX han desarrollado ante la inserción de la cédula de identificación electrónica, así como la implementación de un Middleware que sea capaz de comunicarse con el Applet y realizarle consultas en el orden de la información que se persista en la tarjeta.

CAPÍTULO 2: CARACTERÍSTICAS DE LA SOLUCIÓN

2.1 Introducción

La solución a desarrollar, debe ser fruto de un correcto análisis y una amplia comprensión de todos los elementos que se relacionan en correspondencia al tema de Applet y de Middleware, profundizándose en el estudio de las características que posibilitan desarrollar los mismos.

En este capítulo se interpretan las necesidades del sistema especificándolas mediante los requerimientos funcionales y los no funcionales. Además, se hace un estudio del negocio en que se enmarca el problema concluyendo que se debe realizar una modelación del dominio, identificando para esto las entidades principales que se tendrán y las relaciones entre ellas.

El capítulo además expone el diagrama de casos de uso con sus respectivas descripciones, el diseño de la solución; mostrando el diagrama de clases del Middleware y Applet de Control de Acceso, así como la reseña de cada uno de ellos.

2.2 Modelo de Dominio

Actualmente Venezuela no cuenta con una infraestructura dedicada a los procesos de gestión de la información que se podrá contener en las Cédulas de Identificación Electrónica (CIE), por tal motivo se determina que no se podían identificar procesos del negocio que enmarcaran nuestro problema. Sin embargo conceptos tecnológicos que definen correctamente cada eslabón de la solución se fueron observando, determinándose entonces la creación de un modelo de dominio el cual deja bien claro cómo funciona el entorno en el cual está enmarcado el problema. El modelo de dominio representa cosas del mundo real y para poder identificar los conceptos se hace necesario investigar el dominio del problema. (Proenza, 2005)

A continuación, se muestra el modelo de dominio que conceptualiza los elementos principales y sus relaciones entre sí.

2.3 Diagrama de Clases del Modelo de Dominio

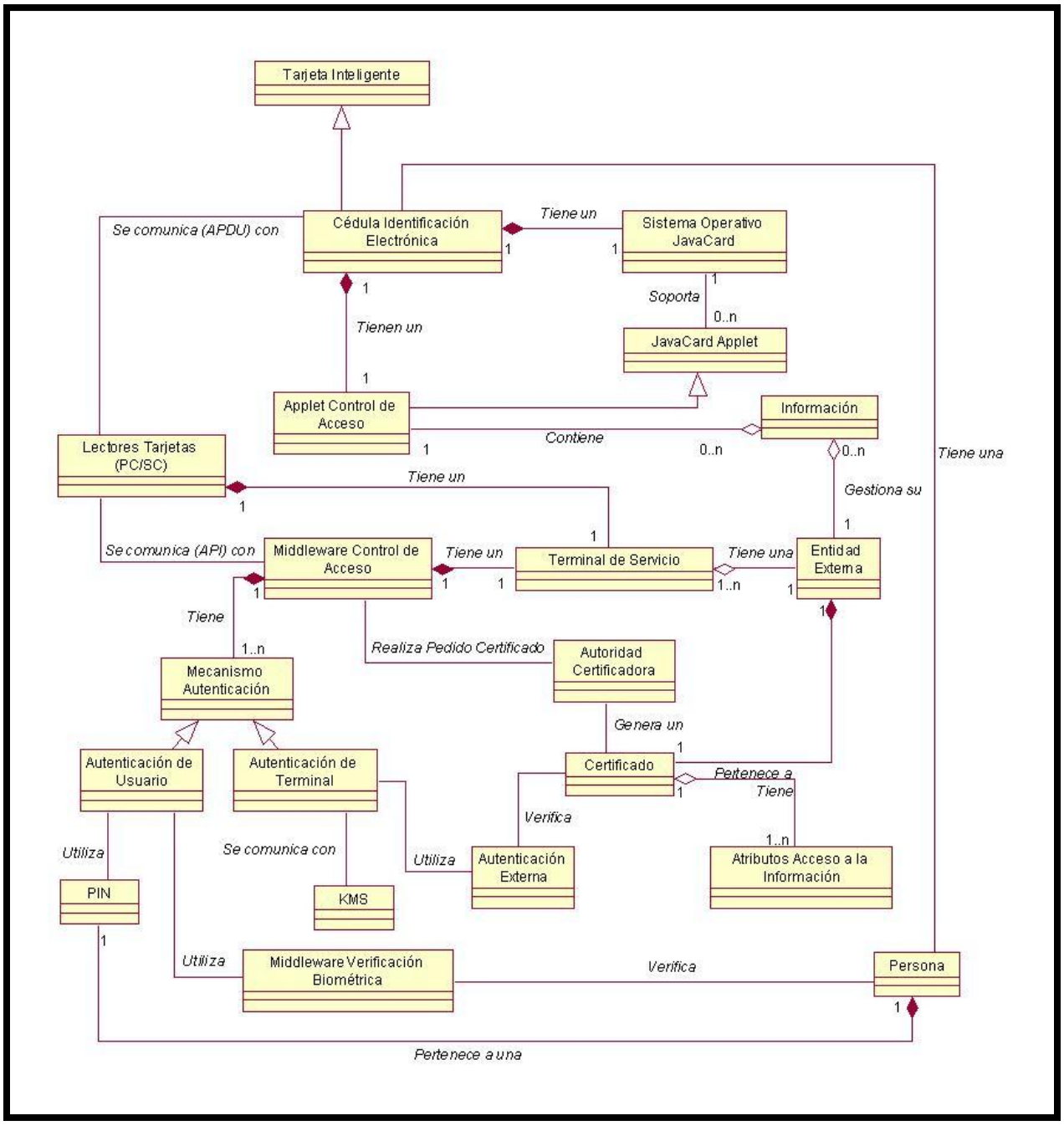


Figura 14: Diagrama de Clases del Dominio.

2.4 Glosario de Conceptos del Modelo de Dominio

A continuación los conceptos tratados en el modelo de dominio serán definidos dentro del contexto del problema a resolver.

- **Tarjeta Inteligente:** Es un dispositivo de plástico similar en tamaño y otros estándares físicos a las tarjetas de crédito, presentan un circuito integrado, el mismo puede ser de sólo memoria o contener un microprocesador(CPU) con un sistema operativo que le permita una serie de funcionalidades:
    - Almacenar Información.
    - Encriptar Información.
    - Leer y escribir datos, similar a un ordenador.
- **Sistema Operativo JavaCard:** La tecnología JavaCard combina parte del lenguaje de programación Java con un entorno de ejecución optimizado para tarjetas inteligentes y similares. El objetivo de la tecnología JavaCard es llevar los beneficios del desarrollo de software en Java al mundo de las tarjetas inteligentes.
- **Cédula Electrónica de Identificación (CIE):** Es una tarjeta inteligente que contiene tecnología JavaCard, la cual contiene datos de identificación de su portador, así como información referente a servicios que brindan distintas entidades externas a la ONIDEX en la República de Venezuela.
- **JavaCard Applet:** Los Applets son las aplicaciones que corren embebidas en una JavaCard. Dichas aplicaciones interactúan en todo momento con el JCRE utilizando los servicios que éste brinda, e implementan la interfaz definida en la clase abstracta `javacard.framework.Applet`.
- **Applet Control de Acceso:** Es una aplicación implementada en JavaCard, instalada dentro de la tarjeta inteligente la cual permite gestionar la información que se almacene dentro de la CIE.
- **Información:** Son datos referentes a los servicios que prestan distintas entidades externas a la ONIDEX en la República Bolivariana de Venezuela y son almacenados en la CIE.
- **Middleware Control de Acceso:** Es un componente que funciona como capa de traducción entre el applet de control de acceso y otro sistema, permitiendo una mejor comprensión de las respuestas obtenidas por la comunicación establecida con la aplicación instalada en CIE (Applet).

- **Entidad Externa:** Es cualquier entidad con jurisdicción del Gobierno Bolivariano de Venezuela, que se encuentra externa a la Oficina Nacional de Identificación y Extranjería (ONIDEX) y muestra interés en guardar información de los servicios que brinda en la CIE.
- **Terminal Servicio:** Es un punto en una computadora donde se instalan todas las condiciones para poder interactuar con la solución para el control de acceso a la información de las Entidades Externas en la República Bolivariana de Venezuela. Entre los elementos indispensables también podemos citar, los lectores PC/SC, lectores de huellas, entre otros.
- **Lectores de Tarjetas PC/SC:** Es un lector compatible con el estándar PC/SC, el cual define la comunicación entre el applet de control de acceso y el middleware de control acceso, sin realizarle modificaciones a ambas aplicaciones.
- **Autoridad Certificadora:** Entidad de confianza, responsable de emitir y revocar los certificados digitales o certificados, utilizados en la firma electrónica y autenticación de las entidades externas, para lo cual se emplea la criptografía de clave pública.
- **Certificado:** Es un documento digital mediante el cual la Autoridad Certificadora (CA siglas en ingles) garantiza la vinculación entre la identidad del portador de la Cédula de Identificación Electrónica de la República Bolivariana de Venezuela o las Entidades Externas y la llave pública del documento. El formato del certificado digital está regido por el estándar UIT-T X.509.
- **Mecanismo Autenticación:** Es el mecanismo que se puede definir para realizar la autenticación tanto de las personas como de los terminales de servicio que requieran de este servicio. Este puede ser dentro de nuestro dominio de problema de dos tipos, *Autenticación de Usuario* y *Autenticación de Terminal.*
- **Autenticación de Usuario:** Consiste en autenticar a un usuario que se le desee comprobar que exista un lazo seguro entre este y su Cédula de Identificación Electrónica. Los tipos de autenticación de usuario pueden ser por *PIN* y por medio de un *Middleware de Verificación Biométrica*.

- **PIN:** (Personal Identification Number o Número de Identificación Personal en castellano) es un valor numérico usado para identificarse y poder acceder a la información gestionada en la CIE.
- **Middleware de Verificación Biométrica:** Es un middleware encargado de gestionar todo un proceso de captación y verificación de las huellas dactilares de la persona portadora de la CIE.
- **Autenticación de Terminal:** Consiste en autenticar a una terminal de servicio de la cual se desee comprobar que está apta para cumplir los roles especificados en los contextos de utilización de la solución propuesta, para la gestión de la información de las Entidades Externas. Esta autenticación puede ser de dos tipos, mediante un *Sistema de administración de Llaves (KMS)* o mediante una *Autenticación Externa.*
- **Sistema de administración de Llaves (KMS):** Es el encargado de proteger las llaves simétricas para la autenticación del terminal, y proveer funcionalidades criptográficas, a través de la generación del criptograma de autenticación.
- **Autenticación Externa:** Es un proceso usado por la CIE para autenticar al host, y determinar el nivel de seguridad requerido para todas las subsecuencias de comandos que se pueden desencadenar entre estos.
- **Persona:** Portador de la Cédula de Identificación Electrónica en la República Bolivariana de Venezuela.

2.5 Especificaciones de los Requerimientos del Software

2.5.1 Requerimientos funcionales

Los Requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. Es decir, especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. (Software, 2004)

RF 1. Comunicar con lector de tarjetas inteligentes seleccionado utilizando estándar PC / SC.

RF 1.1 Permitir obtener lectores disponibles conectados a la terminal de servicio.

RF 1.2 Permitir establecer conexión con la tarjeta inteligente.

RF 1.3 Permitir cerrar conexión con la tarjeta inteligente.

RF 1.4 Permitir enviar comandos APDU a la tarjeta inteligente.

RF 1.5 Permitir obtener APDU de respuestas de la tarjeta inteligente.

RF 2. Permitir obtener el contenido de los *atributos de acceso a la información* de la Cédula de Identificación Electrónica (CIE) del certificado emitido por Autoridad Certificadora (CA) en un formato de fácil interpretación.

RF 2.1 Permitir especificar el formato en el que se desea obtener la información referente a los atributos de acceso. Ejemplo de estos son formato XML y formato HTML.

RF 3. Establecer canal seguro de comunicación con la tarjeta, siguiendo GlobalPlatform.

RF 3.1. Permitir autenticación de Terminal de Servicio On - Line.

RF 3.1.1 Enviar criptograma recibido del sistema de administración de llaves (KMS) a la tarjeta y realizar la autenticación.

RF 3.2 Mostrar información devuelta por la tarjeta.

RF 4. Permitir verificar la validez del Certificado de la Entidad Externa.

RF 4.1 Permitir la verificación en la tarjeta del certificado digital.

RF 4.1.1 Permitir enviar certificado de la tarjeta inteligente.

RF 4.2 Permitir la verificación on-line contra la CA a través del protocolo Online Certificate Status Protocol (OCSP).

RF 4.2.1 Permitir elaborar un pedido de verificación de certificado a través del protocolo OCSP.

RF 4.3 Mostrar información devuelta por la tarjeta.

RF 5. Permitir autenticación del usuario por Número de Identificación Personal (PIN).

RF 5.1 Enviar a la tarjeta el PIN del usuario para realizar la autenticación.

RF 5.2 Mostrar información devuelta por la tarjeta.

RF6. Permitir autenticación del usuario por Mach on Card (MoC).

RF 6.1 Realizar la obtención de minucias de la huella dactilar.

RF 6.2 Enviar a la tarjeta los datos de las minucias obtenidas luego de la captura de huellas dactilares del usuario para realizar la autenticación.

RF 6.3 Mostrar información devuelta por la tarjeta.

RF 7. Gestionar la información contenida en la cédula electrónica.

RF 7.1 Verificar condiciones de acceso a la información dentro del Applet de Control de Acceso.

RF 7.2. Escribir información.

RF 7.2.1 Enviar datos a la tarjeta inteligente.

RF 7.2.2 Mostrar información devuelta por la tarjeta.

RF 7.3. Leer información.

RF 7.3.1 Obtener los datos almacenados en la tarjeta inteligente.

RF 7.3.2 Mostrar información devuelta por la tarjeta.

2.5.2 Requerimientos no funcionales

Los requerimientos no funcionales son las propiedades o cualidades que el producto debe tener para que este sea atractivo, usable, rápido y confiable. (Software, 2004)

- *Usabilidad*
  - El middleware debe ser de fácil utilización para lograr una mayor comodidad en su integración con aplicaciones existentes.

- *Rendimiento*
  - El Applet debe ser capaz de realizar sus operaciones de manera eficiente, garantizando su funcionalidad en un corto intervalo de tiempo.
- *Soporte*
  - Manual de usuarios. Sistema de ayuda. Manual de procedimientos.
- *Portabilidad*
  - El middleware se debe desarrollar sobre una tecnología multiplataforma, que permita su utilización en distintos Sistemas Operativos.
  - El middleware debe ser compatible con cualquier lector de tarjetas que cumpla con el estándar PC/SC.
- *Seguridad*
  - *Confiabilidad*
    - Debe recuperarse en el menor tiempo posible en caso de producirse una falla.
    - La información almacenada en el Applet estará protegida de ataques externos a través de la seguridad que define el proveedor de tarjetas, su Sistema Operativo y la tecnología JavaCard.
  - *Confidencialidad*
    - La información de las Entidades Externas, estará protegida de acceso no autorizado, mediante el uso de los mecanismos de seguridad estándares definidos por GlobalPlatform y PKI.
    - Los datos transmitidos y recibidos de la tarjeta, podrán ser cifrados con criptografía simétrica, según define el estándar GlobalPlatform.
  - *Integridad*
    - La información contenida en la tarjeta, será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.

- Los datos transmitidos y recibidos de la tarjeta, podrán ser verificados utilizando Código de Autenticación de Datos (MAC[1] siglas en ingles), según se define en el estándar GlobalPlatform.

- *Interfaz interna.*
  - Comunicación con lectores de tarjetas inteligentes.
  - Comunicación con escáner de huella.
  - Interfaz con el middleware de verificación biométrica.
  - Interfaz con otras aplicaciones (API).
  - Interfaz de comunicación con una Autoridad Certificadora (CA).
  - Interfaz de comunicación con un Sistema de Administración de Llaves (KMS siglas en ingles).

2.6 Modelación de la Solución

2.6.1 Definición de actores

La solución para el control de acceso a la información de las entidades externas cuenta con un actor que será el encargado de inicializar todas acciones que se llevan a cabo en el Middleware y con los demás sistemas para proveer condiciones estables en la comunicación, seguridad y legitimidad de la gestión de la información.

Actor	Justificación
Terminal de Servicio	Es el responsable de comunicarse con el Middleware de Control de Acceso, inicializar y terminar la comunicación con la Cédula de Identificación Electrónica (CIE), proveer los datos para realizar los procesos de autenticación del terminal y el usuario, y es el encargado de inicializar las acciones de lecturas o escrituras en la CIE.

Tabla 5: Descripción del Actor del Sistema.

[1] MAC: Es una transformación de criptografía simétrica de los datos que provee autenticación original e integridad en los datos.

2.6.2 Diagrama de casos de uso del sistema

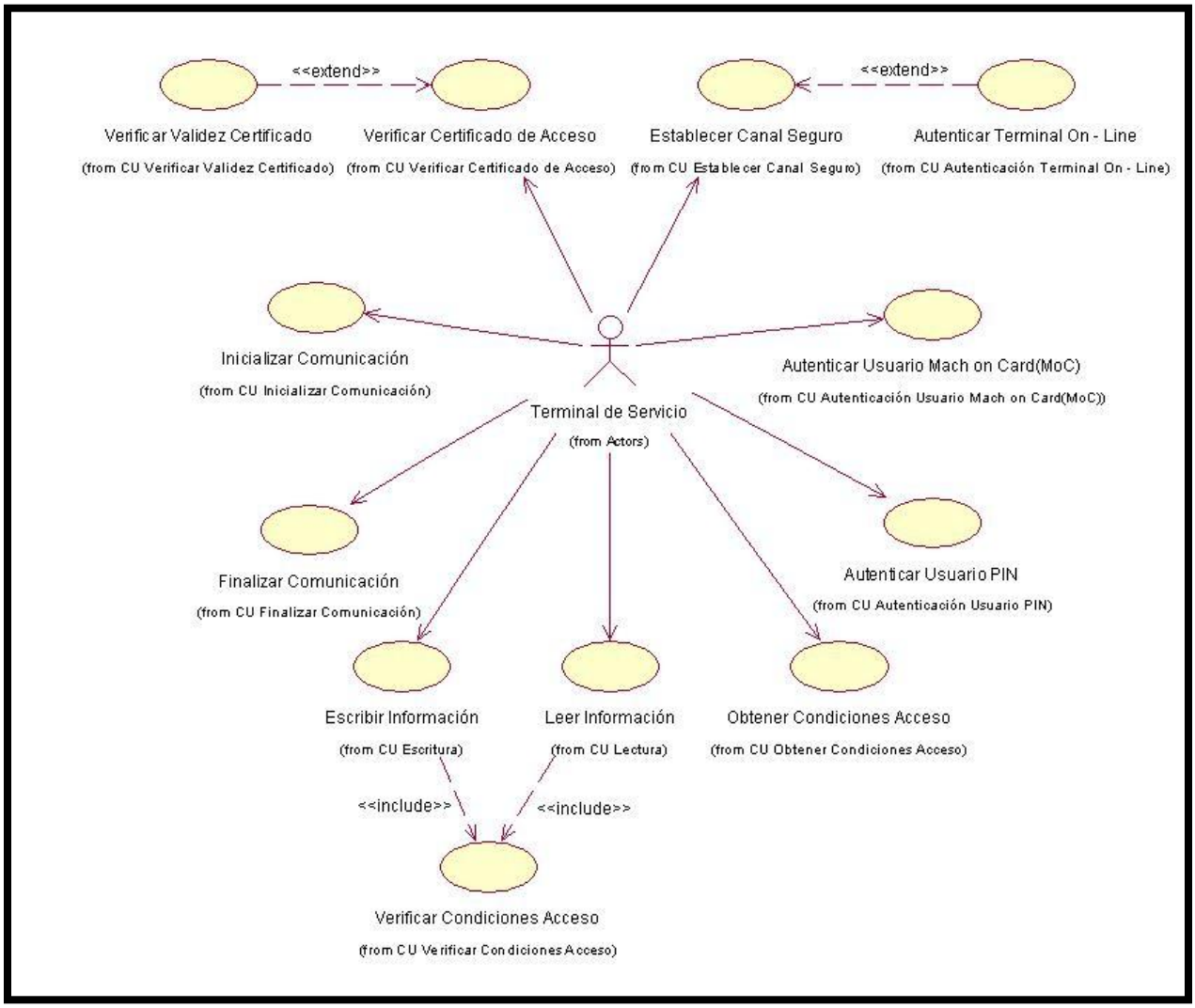


Figura 15: Diagrama de Casos de Uso.

2.6.3 Descripción de los casos de uso del sistema

2.6.3.1 Descripción caso de uso "Establecer Canal Seguro"

Caso de Uso:	Establecer Canal Seguro
Actores:	Terminal de Servicio
Resumen:	Se establece un canal de intercambio de información de forma segura entre el Middleware y el Applet de Control de Acceso, utilizando Protocolo de Canal Seguro "01" según especificaciones de GlobalPlatform.

<table>
<tr><td>Precondiciones:</td><td colspan="2">Inicializar Comunicación</td></tr>
<tr><td>Referencias</td><td colspan="2">RF – 3, RF – 3.1, RF – 3.2.</td></tr>
<tr><td>Prioridad</td><td colspan="2">Alta</td></tr>
<tr><td colspan="3">Flujo Normal de Eventos</td></tr>
<tr><td colspan="2">Acción del Actor</td><td>Respuesta del Sistema</td></tr>
<tr><td colspan="2">1. El caso de uso se inicia cuando el Actor desea establecer un canal seguro de comunicación con la Cédula de Identificación Electrónica (CIE), especificando los parámetros de configuración que son necesarios para inicializar el mismo.</td><td>2. El Middleware de Control de Acceso (MCA), solicita al Applet de la CIE, inicializar una autenticación mutua con el mismo.
3. El Applet de la CIE responde al MCA con una información necesaria para comprobar la veracidad y confiabilidad del Applet.
4. El MCA verifica los parámetros de configuración que le fueron suministrado por el actor. Si especifica que desea realizar un pedido de autenticación de la terminal de servicio de forma on – line, ir al flujo alterno 1. Envía al Applet de la CIE información que identifica a la terminal donde se encuentra instalado para lograr la una autenticación mutua.
5. El Applet de la CIE responde una confirmación en la cual se notifica que se estableció un canal seguro.
6. El MCA informa al actor que el canal seguro se ha establecido o que ha ocurrido algún error en la operación.</td></tr>
<tr><td colspan="3">Flujo Alterno 1</td></tr>
</table>

	5. Se solicita pedido de información a un sistema on – line encargado de generar información para autenticar al terminal de servicio con el Applet.
Puntos de Extensión	
• **Punto de Extensión 1** CU Extendido "Autenticar Terminal on - line".	
Prototipo de Interfaz	
Poscondiciones	El Canal Seguro queda establecido.

Tabla 6: Caso de uso "Establecer Canal Seguro".

2.6.3.2 Descripción caso de uso "Autenticar Terminal on - line".

Caso de Uso:	Autenticar Terminal on - line
Caso de Uso Base:	Establecer Canal Seguro
Actores:	Terminal de Servicio
Resumen:	Se autentica la terminal de servicio de una Entidad Externa, obteniendo para esta un criptograma de identificación de forma on – line.
Precondiciones:	Inicializar Comunicación
Referencias	RF – 3.1, RF – 3.1.1.
Prioridad	Media
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso se inicia cuando el actor solicita establecer un canal seguro utilizando la opción de autenticación de de la terminal de forma on – line.	2. El Middleware le solicita a un Sistema de Administración de Llaves (KMS), que genere la información necesaria para autenticar a la terminal de servicio. 3. El Middleware obtiene el criptograma

	de autenticación. 4. Se utiliza el criptograma obtenido como información que se le envía al Applet de la Cédula de Identificación Electrónica (CIE) para establecer un canal seguro.
Prototipo de Interfaz	
Poscondiciones	Se obtiene el criptograma de autenticación para una terminal en específico.

Tabla 7: Caso de uso "Autenticar Terminal on - line".

2.6.3.3 Descripción caso de uso "Verificar Condiciones de Acceso".

Caso de Uso:	Verificar Condiciones de Acceso.
Actores:	
Casos de Uso Base:	Leer Información. Escribir Información.
Resumen:	Se verifican las condiciones de acceso de un certificado emitido por una autoridad certificadora referente a una terminal de trabajo de una Entidad Externa dentro del Applet de Control de Acceso.
Precondiciones:	Inicializar Comunicación, Establecer Canal Seguro, Verificar Certificado de Acceso.
Referencias	RF – 7.1
Prioridad	Alta
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso se inicia cuando se solicita realizar alguna operación de gestión de la	2. El Applet comprueba que las condiciones de acceso son las

información que se encuentra almacenada en la Cédula de Identificación Electrónica (CIE).	correctas para realizar la solicitud de gestión de la información referente a las Entidades Externas. *Información: En los atributos extendidos del certificado se encuentran los tipos de Modos de Acceso que se pueden tener:* - *Solo lectura de la información.* - *Solo escritura de la información.* - *Lectura / Escritura de la información.* 3. Si las condiciones de acceso son cumplidas el Applet procede a realizar la operación de gestión especificada por el terminal de servicio. Si no ir al Flujo Alterno 1.
Flujo Alterno 1 "Condiciones de Acceso no cumplidas"	
	4. Si las condiciones de acceso no son cumplidas el Applet retorna un mensaje APDU notificando el error el cual está definido dentro del estándar GlobalPlatform.
Prototipo de Interfaz	
Poscondiciones	Se hizo efectiva la verificación de los atributos de acceso a la información correspondientes a un certificado X.509 de una terminal de servicio.

Tabla 8: Caso de uso "Verificar Condiciones de Acceso".

2.6.3.4 Descripción caso de uso "Leer Información".

Caso de Uso:	Leer Información.
Actores:	Terminal de Servicio
Resumen:	Se lee la información de la Entidad Externa en cuestión, que están en la

	Cédula de Identificación Electrónica (CIE) perteneciente a un ciudadano cedulado de la República Bolivariana de Venezuela.
Precondiciones:	Inicializar Comunicación, Establecer Canal Seguro, Verificar Certificado de Acceso, Verificar Condiciones de Acceso.
Referencias	RF – 7.1, RF – 7.3, RF – 7.3.1, RF – 7.3.2
Prioridad	Alta

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso se inicia cuando se solicita leer la información que se encuentra almacenada en la Cédula de Identificación Electrónica (CIE) de un ciudadano, referente a una entidad externa.	2. El Middleware envía la solicitud de lectura de la información. 3. El Applet realiza una Verificación de Condiciones de Acceso. Si las condiciones son las necesarias para realizar la lectura de la información, el Applet retorna un cuerpo de datos (DATA), donde se encuentra la información deseada. Si no cumple las condiciones necesaria ir Flujo Alterno 1. 4. El Middleware recibe la información, y la transforma para mostrársela a la terminal de servicio.
Flujo Alterno1: "Condiciones no Lectura"	
	3. Si se recibe del Applet un mensaje APDU de error (ISOException), se le notifica a la terminal, que las condiciones de acceso a la información presentadas en el certificado emitido por la CA no cumple el *Modo de Acceso* de Lectura de la

	Información.
Prototipo de Interfaz	
Poscondiciones	Se hizo efectiva lectura de la información solicitada por la entidad externa.

Tabla 9: Caso de uso "Leer Información".

2.6.3.5 Descripción caso de uso "Escribir Información".

Caso de Uso:	Escribir Información.
Actores:	Terminal de Servicio
Resumen:	Se escribe información referente a una Entidad Externa en la Cédula de Identificación Electrónica (CIE) perteneciente a un ciudadano cedulado de la República Bolivariana de Venezuela.
Precondiciones:	Inicializar Comunicación, Establecer Canal Seguro, Verificar Certificado de Acceso, Verificar Condiciones de Acceso.
Referencias	RF – 7.1, RF – 7.2, RF – 7.2.1, RF – 7.2.2
Prioridad	Alta
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso se inicia cuando se solicita escribir información en la Cédula de Identificación Electrónica (CIE) de un ciudadano, referente a una entidad externa, enviando la información que se desea persistir en la CIE.	2. El Middleware envía la solicitud de escritura de la información y la información que se desea guardar en la CIE, al Applet de Control de Acceso. 3. El Applet realiza una Verificación de Condiciones de Acceso. Si las condiciones son las necesarias para realizar la escritura de la información, el

	Applet procede a almacenar la información deseada. Si no cumple las condiciones necesaria ir Flujo Alterno 1. 4. El Middleware recibe mensaje APDU emitido por el Applet de operación realizada sin errores.
Flujo Alterno1: "Condiciones no Escritura"	
	3. Si la condición de acceso a la información no es de escritura, entonces se devuelve una excepción que especifique el problema.
Prototipo de Interfaz	
Poscondiciones	Se hizo efectiva escritura de la información deseada por la entidad externa.

Tabla 10: Caso de uso "Escribir Información".

2.6.3.6 Descripción caso de uso "Autenticar Usuario PIN".

Caso de Uso:	Autenticar Usuario PIN
Actores:	Terminal Servicio
Resumen:	Consiste en realizar la autenticación del Usuario portador de la Cédula de Identificación Electrónica (CIE), introduciendo este su número de PIN para validar la autenticidad de posesión de la CIE.
Precondiciones:	Inicializar Comunicación, Establece Canal Seguro.
Referencias	RF – 5, RF – 5.1, RF – 5.2.
Prioridad	Alta
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando se solicita realizar la autenticación del ciudadano poseedor de la CIE, enviando el PIN del poseedor de la Cédula de Identificación Electrónica, como parámetro de configuración para realizar la autenticación.	2. El Middleware envía el PIN al Applet de Control de Acceso, retornando si la autenticación tuvo éxito o no. - Si la autenticación es correcta se notifica que la autenticación ha sido comprobada. - Si la autenticación es incorrecta ir al Flujo Alterno1.
Flujo Alterno1: "Autenticación incorrecta"	
	3. Se notifica que la autenticación ha sido incorrecta.
Prototipo de Interfaz	
Poscondiciones	Se autentica el usuario portador de la CIE.

Tabla 11: Caso de uso "Autenticar Usuario PIN".

2.6.3.7 Descripción caso de uso "Autenticar Usuario MoC".

Caso de Uso:	Autenticar Usuario MoC
Actores:	Terminal Servicio
Resumen:	Consiste en realizar la autenticación del Usuario portador de la Cédula de Identificación Electrónica (CIE), captando la huella dactilar de este para validar la autenticidad de posesión de la CIE.
Precondiciones:	Inicializar Comunicación.
Referencias	RF – 6.1, RF – 6.2, RF – 6.3.
Prioridad	Alta
Flujo Normal de Eventos	

Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando se solicita realizar la captura de la huella dactilar para la obtención de las minucias y autenticarse contra la CIE.	2. El Middleware de Control de Acceso obtiene las minucias de la huella a través del Middleware de Verificación Biométrica.
	3. El Middleware de Control de Acceso envía las minucias al Applet de Verificación Biométrica en la CIE, para realiza la comparación. - Si la comparación de las minucias es correcta se notifica que la autenticación ha sido comprobada. - Si la comparación de las minucias es incorrecta ir al Flujo Alterno1.
Flujo Alterno1: "Autenticación incorrecta"	
	4. Se notifica que la comparación ha sido fallida por lo tanto la autenticación es incorrecta.
Prototipo de Interfaz	
Poscondiciones	Se autentica el usuario portador de la CIE.

Tabla 12: Caso de uso "Autenticar Usuario MoC".

2.6.3.8 Descripción caso de uso "Verificar Validez Certificado".

Caso de Uso:	Verificar Validez Certificado
Caso de Uso Base:	Verificar Certificado de Acceso.
Actores:	
Resumen:	Se válida que el certificado que esta emitido para una terminal de servicio de

	una Entidad Externa presenta condiciones de validez.
Precondiciones:	Inicializar Comunicación.
Referencias	RF – 4.2, RF – 4.2.1
Prioridad	Media

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso se inicia cuando se solicita Verificar Certificado de Acceso, pudiéndose verificar si el certificado en cuestión cumple con las normas de valides de: el tiempo de validez y que no se encuentre en lista de revocación.	2. El Middleware envía el certificado seleccionado a una Autoridad Certificadora (CA) encargada de realizar la comprobación de la validez de dicho certificado, utilizando para esto el protocolo Online Certificate Status Protocol (OCSP).
	3. El Middleware obtiene la información de respuesta de la CA y se la notifica al actor. Si notifica que el certificado es inválido ir al Flujo Alterno1.
Flujo Alterno1: "Certificado inválido"	
	4. Se notifica a la terminal de servicio que el certificado seleccionado es incorrecto.
Prototipo de Interfaz	

Poscondiciones	Se realiza la comprobación de la validez del certificado.

Tabla 13: Caso de uso "Verificar Validez Certificado".

2.6.3.9 Descripción caso de uso "Obtener Condiciones Acceso".

Caso de Uso:	Obtener Condiciones Acceso.

<table>
<tr><td>Actores:</td><td colspan="2">Terminal de Servicio</td></tr>
<tr><td>Resumen:</td><td colspan="2">Se obtiene las condiciones de Accesos que fueron emitidas en un certificado, por una autoridad certificadora, transformando los mismos en un formato de fácil interpretación, así como brindando la posibilidad de seleccionar la extensión en la que se desee interpretar dichas condiciones.</td></tr>
<tr><td>Precondiciones:</td><td colspan="2">Ninguna</td></tr>
<tr><td>Referencias</td><td colspan="2">RF – 2, RF – 2.1</td></tr>
<tr><td>Prioridad</td><td colspan="2">Alta</td></tr>
<tr><td colspan="3">Flujo Normal de Eventos</td></tr>
<tr><td colspan="2">Acción del Actor</td><td>Respuesta del Sistema</td></tr>
<tr><td colspan="2">1. El caso de uso se inicia cuando se solicita obtener las condiciones de acceso que están presentes dentro del certificado que fue emitido por una Autoridad Certificadora (CA), pasándole las opciones de configuración del formato en la que se desee obtener las condiciones de acceso.</td><td>2. El middleware extrae del certificado X.509, las extensiones referentes a las condiciones de acceso a la información que se encuentra en la Cédula de Identificación Electrónica (CIE).
3. Se verifica las opciones de configuración de la extensión que fueron enviadas por el terminal.
4. Se transforman las condiciones de acceso de la información en el formato deseado por el Terminal.
5. Se envía el documento generado al Terminal para que este pueda gestionarlo.</td></tr>
<tr><td colspan="3">Prototipo de Interfaz</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td>Poscondiciones</td><td colspan="2">Se hizo efectiva la obtención de las condiciones de acceso en un formato de</td></tr>
</table>

	fácil interpretación para el Terminal

Tabla 14: Caso de uso "Obtener Condiciones Acceso".

2.6.3.10 Descripción caso de uso "Inicializar Comunicación".

Caso de Uso:	Inicializar Comunicación
Actores:	Terminal Servicio
Resumen:	Se muestran los lectores que están disponibles, se selecciona uno con el cual se va a establecer la comunicación con la Cédula de Identificación Electrónica (CIE).
Precondiciones:	Ninguna
Referencias	RF – 1.1, RF –1.2, RF –1.4, RF –1.5.
Prioridad	Alta

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso se inicia mostrando los lectores de tarjetas conectados al terminal de servicios.	
2. El actor selecciona el lector de tarjetas con el que va desea comunicarse.	3. El Middleware de Control de Acceso inicializa la comunicación con la CIE mediante el envió de APDU.
Prototipo de Interfaz	
Poscondiciones	Queda inicializada la comunicación con la CIE

Tabla 15: Caso de uso "Inicializar Comunicación".

2.6.3.11 Descripción caso de uso "Finalizar Comunicación".

Caso de Uso:	Finalizar Comunicación
Actores:	Terminal Servicio
Resumen:	Consiste en realizar la desconexión entre la Cédula de Identificación Electrónica (CIE) y el lector.
Precondiciones:	Ninguna
Referencias	RF – 1.3
Prioridad	Alta
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso se inicia cuando el actor selecciona la opción desconectar la CIE del lector.	2. El Middleware de Control de Acceso ejecuta la funcionalidad de desconectar la CIE del lector.
Prototipo de Interfaz	
Poscondiciones	Queda finalizada la comunicación de la CIE y el lector.

Tabla 16: Caso de uso "Finalizar Comunicación".

2.6.3.12 Descripción caso de uso "Verificar Certificados de Acceso".

Caso de Uso:	Verificar Certificados de Acceso
Actores:	Terminal Servicio
Resumen:	Consiste en realizar la validación, en la Cédula de Identificación Electrónica (CIE), del certificado digital presentado por la Entidad Externa.

Precondiciones:	Inicializar Comunicación.
Referencias	RF – 4.1, RF – 4.1.1.
Prioridad	Alta
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso se inicia cuando el actor solicita verificar el certificado de acceso.	2. El Middleware de Control de Acceso envía a la CIE el certificado digital de la Entidad Externa.
	3. El Applet de Control de Acceso verifica la validez del certificado a través de la firma digital de sus datos. - Si la verificación es correcta se informa. - Si la verificación es incorrecta ir al Flujo Alterno1.
Flujo Alterno1: "Verificación incorrecta"	
	4. Se notifica que el certificado digital presentado no es válido.
Prototipo de Interfaz	
Poscondiciones	Queda Verificado el Certificado de Acceso.

Tabla 17: Caso de uso "Verificar Certificados de Acceso".

2.7 Diseño

2.7.1 Descripción de los Principales Flujos de Procesos

2.7.1.1 Proceso de Autenticación de Usuario

Uno de los procesos a realizarse es el de autenticación de usuario, él mismo se efectúa con el objetivo válidar las condiciones de seguridad existentes a la hora de escribir o leer información en la Cédula de identificación Electrónica (CIE). Este proceso consiste en: la Terminal de Servicio de la entidad externa a través del Middleware de Control de Acceso solicita la captura de la huella dactilar y/o número de identificación personal (PIN) del portador de la CIE, el portador introduce lo solicitado y el middleware envía los datos de autenticación al Applet de Control de Acceso para que sean verificados; luego el applet envia, la respuesta obtenida, al middleware y este a su vez se la muestra a la Terminal de Servicios. (Ver Anexo 8)

2.7.1.2 Proceso de Gestión de Información

El proceso de gestión de la información comienza cuando la Terminal de Servicio, presenta el certificado digital emitido por la autoridad certificadora correspondiente a la entidad externa.; luego, el Middleware de Control de Acceso obtiene las extensiones que presenta el certificado y genera el token de acceso, dicho token es enviado al Applet de Control de Acceso donde se verifica y se obtiene los atributos de seguridad necesarios para realizar gestión de la información. Los atributos de seguridad se verifican para saber qué operación, de lectura o escritura, se puede efectuar; se procede a realizar una de ellas y se le informa al Terminal de Servicio según la operación realizada. (Ver Anexo 8)

2.7.2 Características del Applet de Control de Acceso

El Applet de Control de Acceso incluye un sistema de ficheros (*File System*) que se responsabiliza por la administración de los ficheros que se almacenen. La instanciación inicial del Applet activa una única vez la construcción de una jerarquía de ficheros dedicados, manipulados en el nivel global de la estructura general (*ver Figura 16 Diagrama de estructura de ficheros*), creándose un fichero raíz (*Master File*) que indica

el comienzo de la estructura de ficheros y un fichero dedicado (*Dedicated File*), el cual es el contenedor a nivel global de todos los ficheros elementales (*Elementary File*) que son generados a partir de un token de acceso específico para cada Entidad Externa.

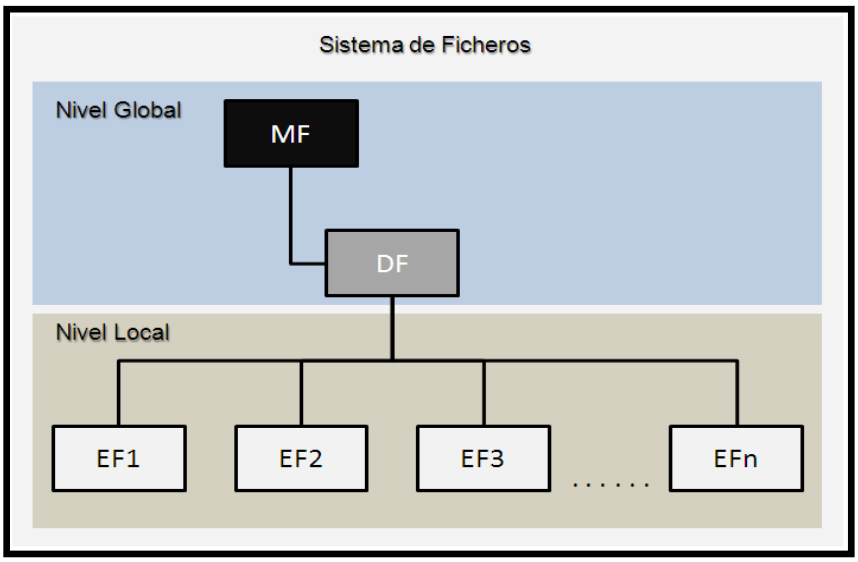


Figura 16: Diagrama de Estructura de Ficheros.

Cada fichero dentro del sistema de ficheros cumple con el estándar ISO/IEC 7816 – 4, donde están identificados por un archivo de información de control (*File Control Information / FCI*) *ver Anexo 4 y Anexo 5*, los cuales están definidos para cada fichero independientemente de su tipo, dedicado y elemental. Cada FCI de un fichero elemental presenta un byte de descripción de fichero el cual puede ser variable, especificando el tipo de estructura de almacenamiento que contendrá dicho fichero, el recomendable por su intuitiva funcionalidad es el transparente, definido por un arreglo de byte capaz de proporcionar linealmente toda la información almacenada. El Applet manipula el tipo de estructura de datos transparente, pero cuenta con una fábrica de estructuras de datos dando extensibilidad en la manipulación de las dichas estructuras.

El Applet de Control de Acceso interpreta los comandos APDU que le son proporcionados desde el exterior según su instrucción. Los que están definidos dentro del estándar ISO/IEC 7816 – 4 son tratados por un intérprete definido para este tipo de APDU en el cual se llevan los procesos de gestión de los ficheros, incluyendo la

verificación de las condiciones de acceso y de seguridad, definidos en los FCI de los ficheros. . Los comandos APDU que no estén especificados bajo el estándar, son gestionados por el mismo Applet que es capaz de procesarlos y generar respuestas acorde a la acción realizada. Todos estos comandos APDU cambian el estado en los que el Applet pueda estar (*ver Anexo 6*) los cuales están definidos en los requerimientos funcionales de la solución.

2.7.3 Diagrama de Clases del Applet de Control de Acceso.

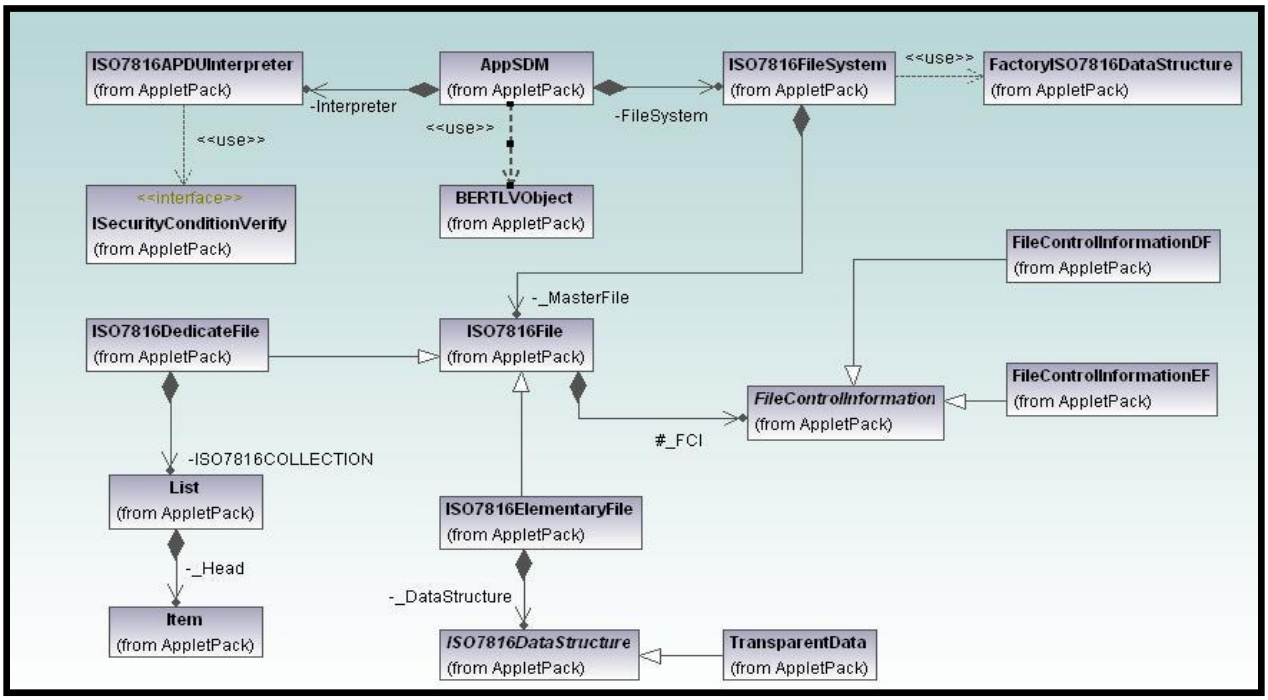


Figura 17: Diagrama de Clases del Applet.

Para tener una vista detalla de cada clase que compone el diagrama del Applet de Control de Acceso puede ver el anexo 6 donde se encuentra cada una con sus atributos y métodos.

2.7.4 Descripción del Token de Acceso.

El token de acceso es un TLV (formato para representar información, de forma que haya información que pueda tener presencia opcional y longitud variable), el cual contiene los atributos de accesos compuestos por los FCI de los ficheros que son generados dentro de la tarjeta para crear los ficheros de información de las Entidades

Externas, incluyendo también la llave publica de la Entidad Externa, así como una firma de los datos anteriormente expresados. A continuación se muestra una tabla con la estructura del Token de Acceso.

Token de Acceso

Posición	Valor	Descripción
0	63h	Cabecera del Token de Acceso
1 - 2	Le	Longitud del Token de Acceso
3	FEh	Cabecera de los Atributos de Seguridad
4	L	Largo de los Atributos de Seguridad
5	Atributos Seguridad	Datos de los Atributos de Seguridad
6 + X	53h	Cabecera de la llave publica
7 + X	L	Largo de la llave publica
8 + X	54h	Cabecera del Módulo de la llave publica
9 + X	L	Largo del Módulo de la llave publica
10 + X	Módulo	Valor del Módulo de la llave publica
11 + X	55h	Cabecera del exponente de la llave publica
12 + X	L	Largo del exponente de la llave publica
13 + X	Exponente	Valor del Exponente de la llave publica
14 + X	56h	Cabecera de la firma de los datos
15 + X	L	Largo de la firma de los datos
16 + X	Firma	Valor de la firma de los datos

Tabla 18: Descripción del Token de Acceso.

2.7.5 Descripción de los APDU.

Los comandos APDU que son enviados a la tarjeta, presentan la estructura especificada en el estándar IEC / ISO 7816 – 4, utilizándose algunos predefinidos y utilizados internacionalmente bajo dicho estándar, y otros que son creados para operaciones especificas del Applet de Control de Acceso. A continuación se muestra

una tabla donde se explican los comandos APDU y se brinda una breve descripción de sus significados.

Nombre	Comando APDU					Descripción
APDU Estándares de Global Platform						
	Composición					
	CLA	**INS**	**P1**	**P2**	**DATA**	
Initialize Update	80	50	00	00	-	Comando que inicializa el proceso para el canal seguro para la comunicación entre el Secure Data Manager Applet, que se encuentra en la CIE y el Terminal de Servicio.
External Authenticate	84	82	00	00	-	Comando para la autenticación del Terminal de Servicio por el Secure Data Manager Applet y determina el nivel de seguridad requerido por todos los comandos que le siguen.
APDU Estándares del ISO / IEC 7816						
	Composición					
	CLA	**INS**	**P1**	**P2**	**DATA**	
Select File	00	A4	00	00	-	Comando para seleccionar el fichero especificado, ya sea DF o EF.
Create File	00	E0	00	00	FCI	Comando para crea un EF debajo de un EF o del MF, o crear un DF debajo del MF. Aunque se interprete este comando en el Applet no estará disponible para ejecutarse desde fuera de la tarjeta, solo de forma interna.
Read File	00	B0	00	00 - FF	-	Comando para leer la información contenida en un EF.
Write File	00 / 10	D0	00	00	Deed	Comando para escribir información dentro de un EF.
APDU Propios						
	Composición					
	CLA	**INS**	**P1**	**P2**	**DATA**	
Send Access Token	00	E9	00	00	-	Comando que permite enviar el token de acceso a la tarjeta.
PIN Authentication	00	E7	00	00	PIN	Comando para realizar la autenticación del usuario con la CIE

						mediante la utilización de un PIN.
PIN Change	00	E8	00	00	PIN	Comando para realizar el cambio del PIN que presenta el portador de la CIE.

Tabla 19: Descripción de los APDU.

2.7.6 Características del Middleware de Control de Acceso

El Middleware de Control de Acceso, es un componente que actúa como una capa (wrapper), que aísla al humano de la comunicación directa con las operaciones que realiza el Applet de Control de Acceso. El middleware es capaz de incluir la definición de los comandos APDU que son enviados al Applet los cuales están definidos en el estándar IEC / ISO 7816 – 4 y los comandos APDU definidos para la utilización especifica de dicho Applet. La comunicación del middleware con el Applet se realiza a través del middleware SmartCard Framework, el cual implementa las especificaciones de los estándares ISO 7816 – 4, y Global Platform para los procesos de encriptación de los datos a enviar, también implementa a nivel de API la comunicación con los lectores de tarjetas electrónicas inteligentes.

El Middleware de Control de Acceso implementa los procesos definidos en los requerimientos de la solución, utilizando además las operaciones que son necesarias para gestionar el certificado que es emitido por una autoridad certificadora (*ver Anexo 7*) para una Entidad Externa específica, construyendo posteriormente el Token de Acceso y enviándolo al Applet de Control de Acceso. La comunicación entre el Middleware y el Applet está determinada bajo capas que son necesarias para establecer un canal correcto de transmisión de los datos de información. El siguiente diagrama demuestra dichas capas de comunicación.

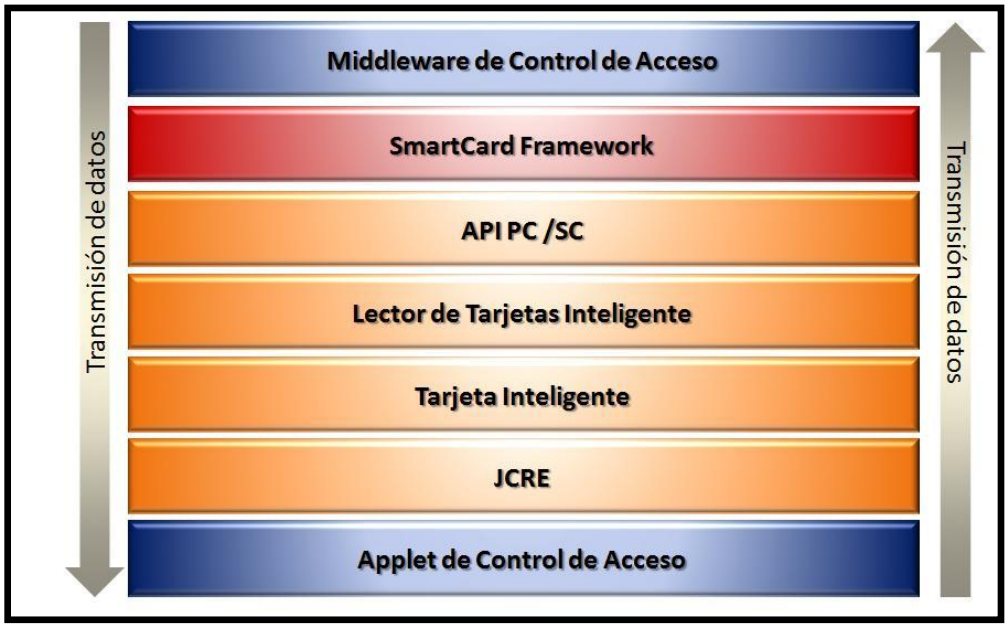


Figura 18: Diagrama de capas de comunicación.

2.7.7 Diagrama de Clases del Middleware

El Middleware de Control de Acceso presenta el siguiente diagrama de clases.

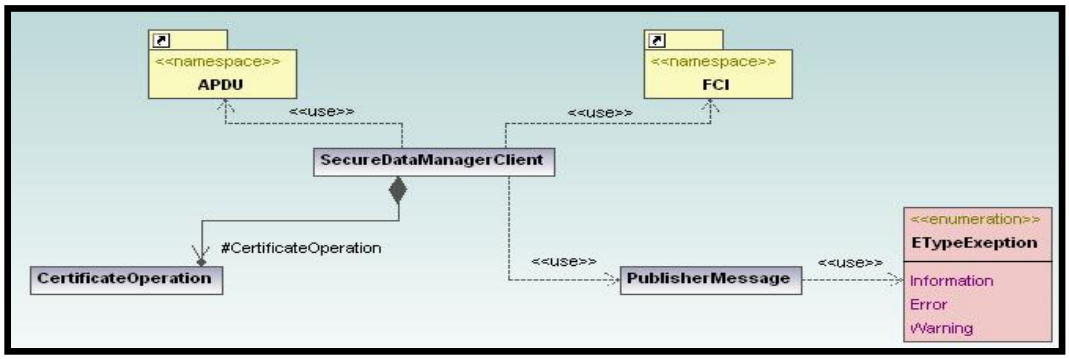


Figura 19: Diagrama de clases del Middleware.

La clase <<*CertificateOperation*>> es la encargada de gestionar los procesos asociados a los certificados X509 versión 3, que son emitidos por las Autoridades Certificadoras. Esta gestión consiste en la extracción de las extensiones que son incluidas dentro de estos certificados, conformando seguidamente el *Token de Acceso* que será enviado al Applet de Control de Acceso. Los certificados emitidos cuentan con una estructura estándar (ver Anexo 5 Figura 33), contando con tres extensiones críticas; los atributos de accesos a los ficheros que se encuentran en la tarjeta; la llave

pública de la entidad externa que realizó el pedido de certificado; así como la firma de estas dos extensiones anteriormente expuestas.

El Middleware usa dos paquetes que le proveen información y manipulación de los requerimientos.

El paquete APDU contiene las clases que se describen en el siguiente diagrama.

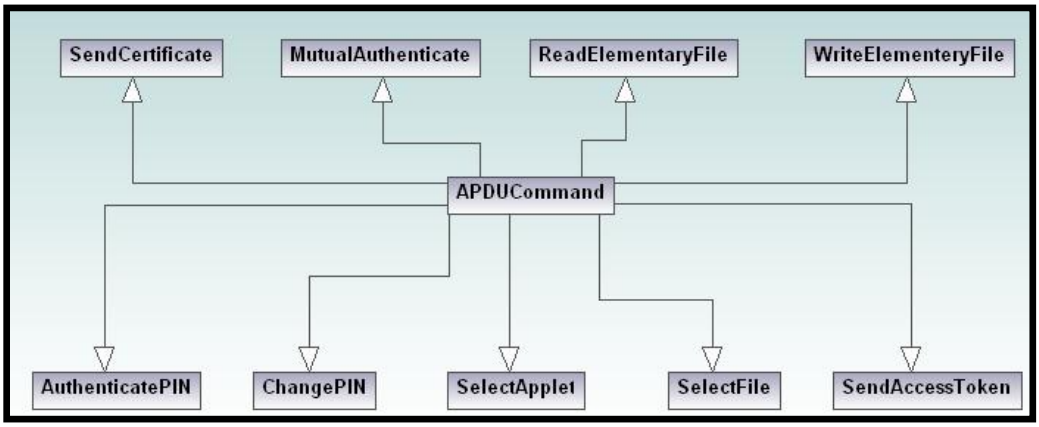


Figura 20: Diagrama de clases del paquete APDU.

Las clases que se describen en el paquete APDU heredan de la clase APDUCommand la cual especifica la estructura de los comandos APDU según el estándar IEC/ISO 7816 – 4. Estas clases son utilizadas por la clase <<*SecureDataManagerMiddleware* > > para la transmisión de los comandos APDU correspondientes a cada una de las acciones que se llevan a cabo.

El paquete FCI es el encargado del proceso de interpretación de un certificado emitido por una Autoridad Certificadora, donde en sus extensiones están contenidos los atributos de accesos descritos en formato FCI del estándar IEC/ISO 7816 – 4.

El siguiente diagrama muestra las clases de este paquete.

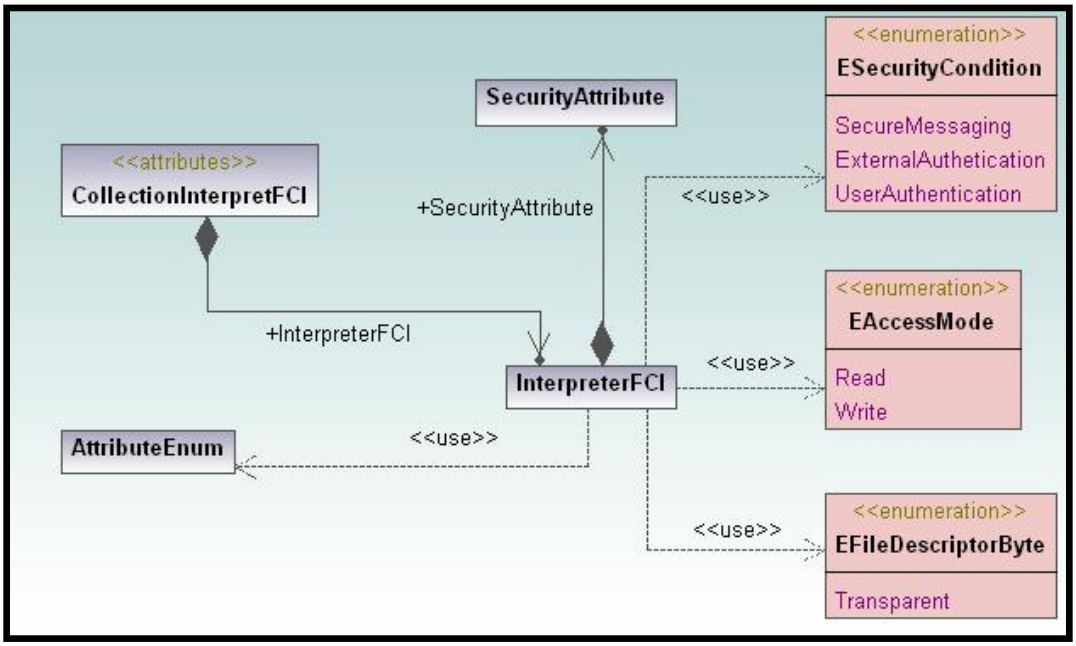


Figura 21: Diagrama de clases del paquete FCI.

Para tener una vista detalla de cada clase que compone el diagrama de clases del Middleware de Control de Acceso puede ver el anexo 7 donde se encuentra cada una con sus atributos y métodos.

2.7.8 Diagramas de Secuencia

Muestran las interacciones entre objetos, ordenadas en secuencia temporal durante un escenario concreto. Si los casos de uso tienen varios flujos o subflujos distintos, suele ser útil crear un diagrama de secuencia para cada uno de ellos.

2.7.8.1 Diagrama de Secuencia “CU Establecer Canal Seguro”.

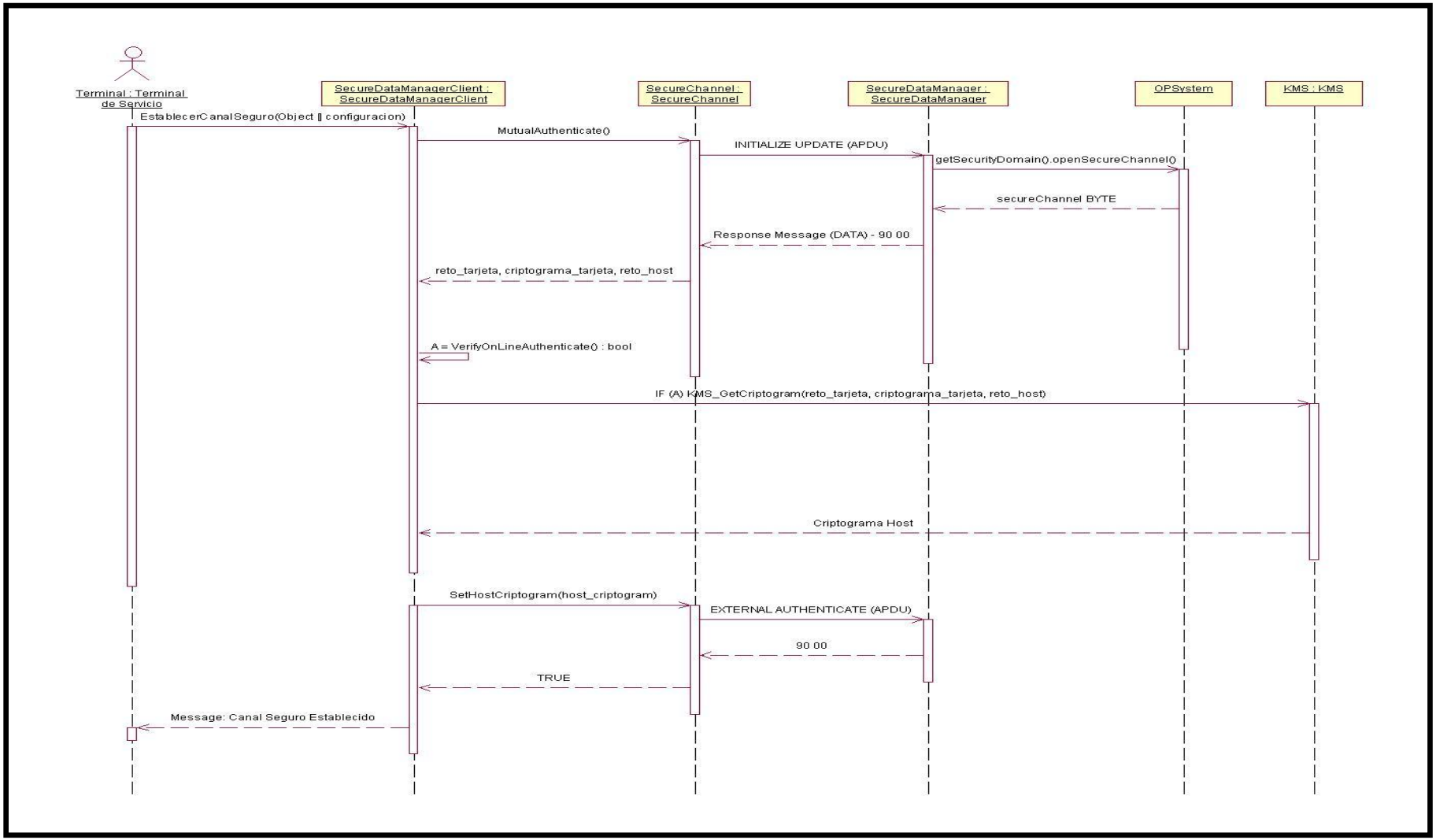


Figura 22: Diagrama de Secuencia “Establecer Canal Seguro”.

2.7.8.2 Diagrama de Secuencia “CU Autenticar Usuario PIN”

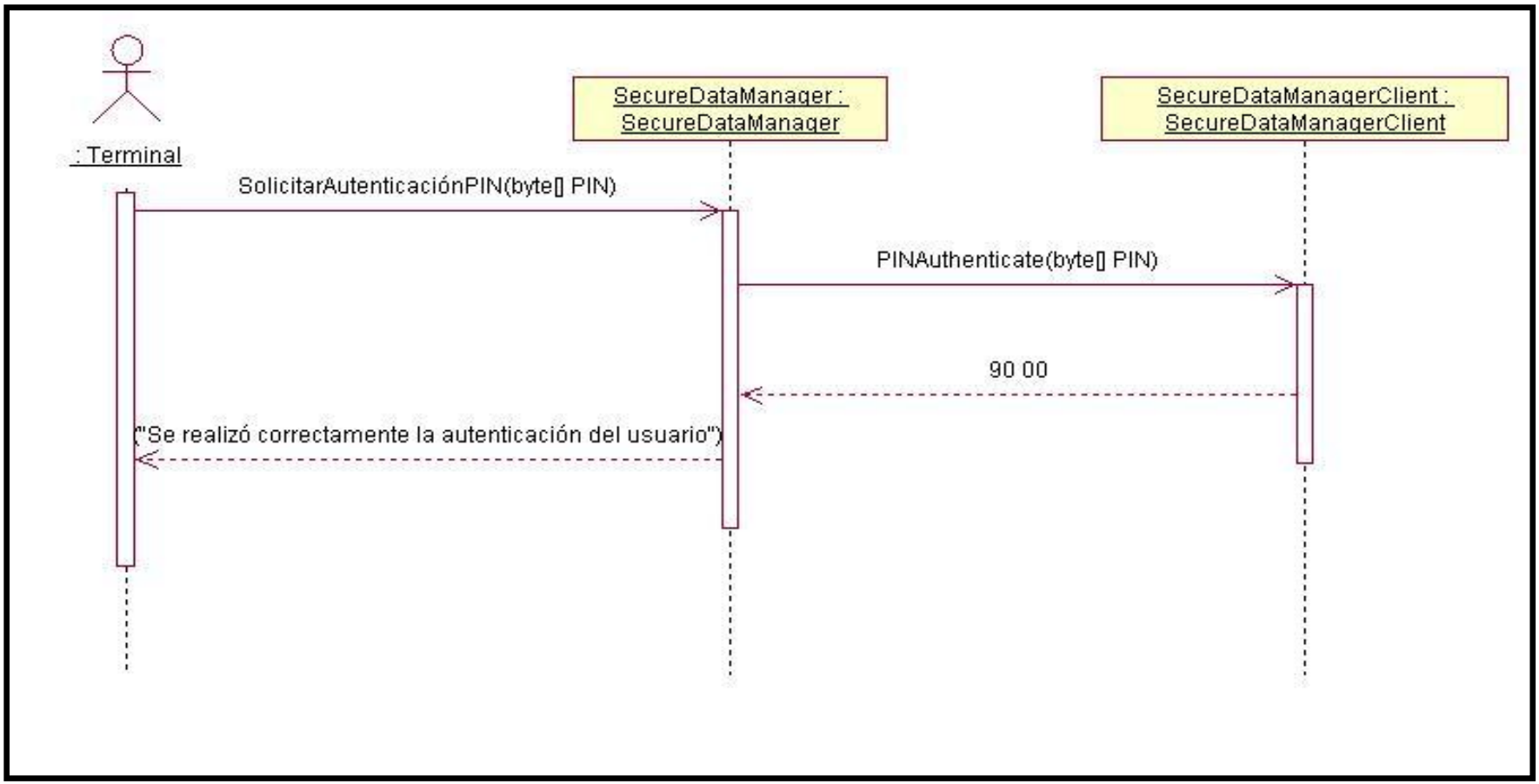


Figura 23: Diagrama de Secuencia "Autenticar Usuario PIN".

2.7.8.3 Diagrama de Secuencia "CU Escribir Información"

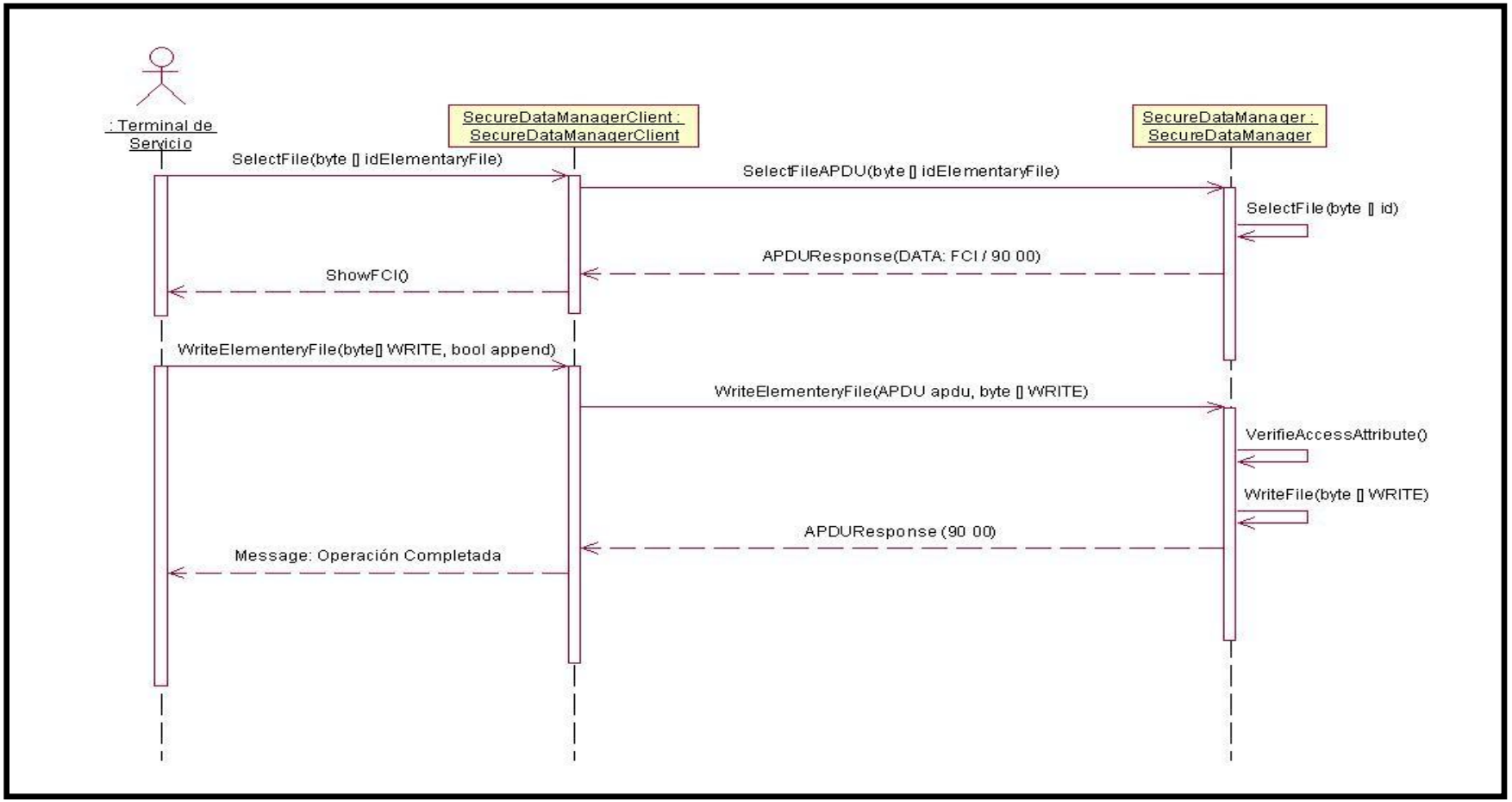


Figura 24: Diagrama de Secuencia "CU Escribir Información".

Los demás Diagramas de Secuencia se pueden observar en los anexos (ver anexo 7).

2.8 Conclusiones

- En el concluyente capitulo se determinaron las bases del porque la elección de un modelo de dominio, el cual agruparía todos los conceptos asociados a nuestra solución así como las relaciones entre estos conceptos, que son determinados luego del estudio de las herramientas, tecnologías y elementos tangibles asociados al dominio de la solución.
- Se determinó el actor del sistema que estará encargado de brindar la interacción con la solución, realizándose una descripción del mismo, determinándose también los casos de usos del sistema con los cuales este actor se relaciona.
- Se definieron los requerimientos funcionales que caracterizan a la solución así como los requerimientos no funcionales que brindarán las cualidades que se deben de tener en cuenta para desarrollar una solución adecuada.
- El Token de Acceso está estructurado en TLV que es un formato de fácil interpretación el cual es enviado a la tarjeta para gestionar todos los procesos de acceso y creación de ficheros que son específicos para cada Entidad Externa.
- Los comandos APDU que son enviados al Applet de Control de Acceso están definidos bajo la estructura del estándar ISO / IEC 7816 – 4, manipulándose también comandos definidos dentro de este estándar internacional, así como otros comandos que son específicos de la solución que en su totalidad son el medio de transporte de información comunicativa con el Applet de Control de Acceso.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

3.1 Introducción

En el siguiente capítulo se verá un representación física de cómo se implementó la solución, a través de representaciones gráficas, enfocándose en el diagrama de componente y diagrama de despliegue de la solución desarrollada. Además de que se validará y verificará el cumplimiento de los requerimientos estipulados, mediante la aplicación de métodos de pruebas que garanticen la calidad del sistema.

3.2 Diagrama de Despliegue

El diagrama de despliegue muestra las relaciones que se establecen entres los componente software y hardware, representado mediante nodos estrechamente conectados.

3.2.1 Descripción del Diagrama de Despliegue

La aplicación para gestionar la información en la Cédula de Identificación Electrónica (CIE) de la República Bolivariana de Venezuela debe estar instalada en cada CIE que emita la Oficina Nacional de Identificación y Extranjería (ONIDEX), y el Middleware para realizar la comunicación con la CIE, debe estar en cada uno de los Terminales de Servicio que existan en las Entidades Externas. Esta comunicación se realizará mediante el envio de unidad de datos de protocolo de aplicaciones (APDU siglas en inglés) con los lectores de tarjetas sin contacto; los lectores se conectan a cada uno de los Terminales de Servicio mediante el estándar USB.

En la Terminal de Servicio se encuentra instalado un lector biométrico de huellas dactilares, para la captura de las minucias de cada individuo, opcionalmente se puede incluir en el proceso de autenticación. Este hardware va conectado por el estándar de comunicación USB.

El Middleware también efectúa la comunicación entre el Terminal de Servicio, la autoridad certificadora encargada de comprobar que el certificado perteneciente a una

Entidad Externa está activo (no se encuentre en una lista de revocación de certificados, la fecha de expiración no se ha cumplido), y un sistema de administración de llaves (KMS siglas en inglés) para administrar las llaves del proceso de autenticación. La comunicación con la CA de verificación de certificados se establece mediante el protocolo para el estado de certificados en línea (OCPS siglas en inglés). La comunicación con el KMS se establece mediante el estándar: protocolo de acceso de objetos simples (SOAP siglas en inglés) utilizado en los servicios Web.

La comunicación con la CA y el KMS se realiza mediante la utilización de una red privada con un nivel de seguridad para proteger los datos.

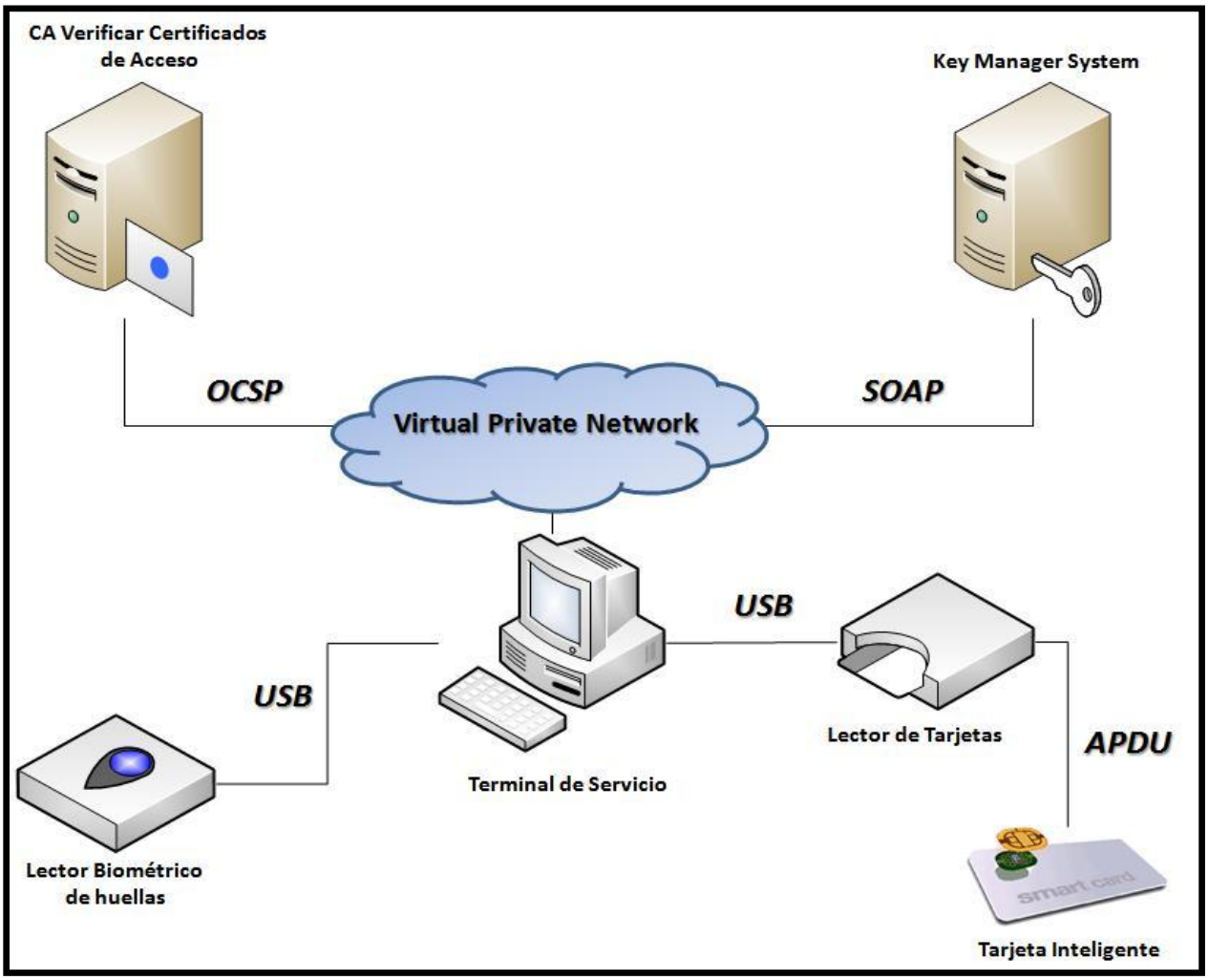


Figura 25: Diagrama de Despliegue.

3.3 Diagrama de Componentes

Se muestra el Diagrama de Componentes del Middleware donde se relacionan componentes y paquetes que brindan la arquitectura del Middleware. En la solución se implementó la DLL <<SecureDataManager.dll>>, el paquete <<SmartCard.Net Framework>> y el paquete <<WinPCSC Wrapper>>.

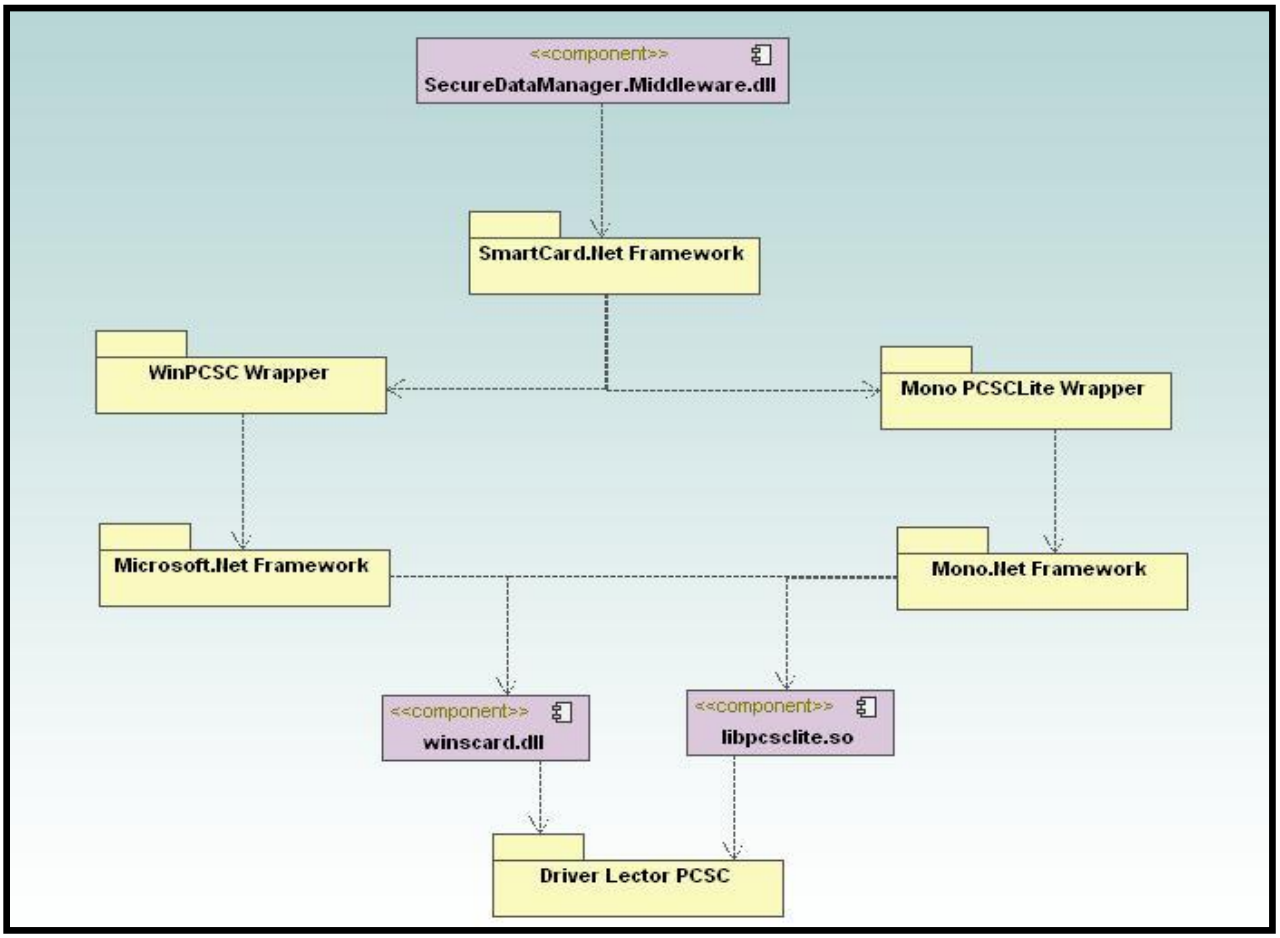


Figura 26: Diagrama de Componentes.

3.3.1 Descripción del Diagrama de Componentes

El Diagrama de Componentes representa cómo el Middleware de Control de Acceso fue desarrollado. El mismo está compuesto por la DLL <<SecureDataManagerCliente.dll>> la cual contiene todas las funcionalidades que

presenta el middleware, Este a su vez utiliza el SmartCard .Net Framework que es un paquete contenedor de los wrapper WinPCSC y Mono PCSCLite

El WinPCSC Wrapper es la solución para la ejecución de aplicaciones de SmartCard que cumplan las especificaciones PC/SC para el sistema Operativo Windows, la cual utiliza el paquete Microsoft.Net Framework que contiene todas las librerías bases que provee la tecnología .Net, así como de la DLL <<winscard.dll>> que permite la comunicación con cualquier equipo que cumpla con las especificaciones PC/SC.

El MonoPCSCLite es la solución para la ejecución de aplicaciones de SmartCard que cumplan las especificaciones PC/SC para software libre, este wrapper incluye todas las características para ser ejecutado bajo ambiente Linux o Windows, está desarrollado sobre el Mono.Net Framework y utiliza la biblioteca <<winscard.dll>>para implementar PC/SC para Windows y de la librería <<libpcsclite.so>> para implementar PC/SC para entorno Linux.

3.4 Prueba

El principal objetivo de esta disciplina es de evaluar la calidad del producto que se está desarrollando a través de las diferentes fases por las cuales este pasa, mediante la aplicación de pruebas concretas para validar que las suposiciones hechas en el diseño y los requerimientos se estén cumpliendo satisfactoriamente, esto quiere decir que se verifica que el producto funcione como se diseñó y que los requerimientos son satisfechos cabalmente. (Centro Nacional de Tecnologías de Información , 2009)

Como forma de verificar los objetivos trazados, se llevó a cabo un proceso de pruebas que validará y le dará un nivel de calidad a la solución implementada. Para ello Para ello se llevó a cabo las pruebas de caja blanca.

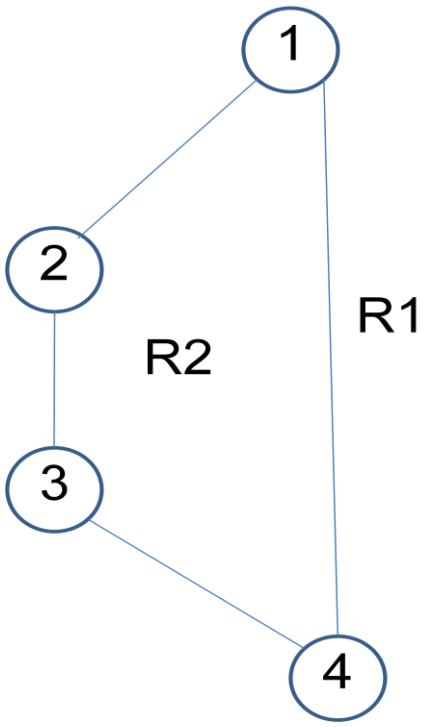
3.4.1 Prueba de Caja Blanca

La puesta en práctica de este método requiere del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones del diseño o el código. Se basa en la comprobación de los caminos lógicos del software dado un código específico.

3.5 Casos de Prueba

De acuerdo al segmento de código correspondiente al caso de uso "*Inicializar Comunicación*"; se le realiza la prueba de caja blanca.

```
public void APDUCommandSelectApplet ()
{
    if (_Conect) (1)
    {
        SelectApplet apdu = new SelectApplet(); (2)
        APDUResponse resp = _CardReader.Transmit(apdu); (3)
    }
} (4)
```



Complejidad Ciclomática.

V (G): *Número de regiones del grafo.*

V (G) = A – N + 2

V (G) = P + 1

A: *Número* de aristas del grafo.

N: *Número de n*odos.

P: *Número* de nodos predicados.

V (G) = 2

Caminos: 1-2-3-4, 1-4.

Camino: 1-2-3-4.

Caso de prueba: Seleccionar el Applet; SecureDataManager (SDM).

Entrada: Conexión entre el lector de tarjetas inteligentes y la Cédula de Identificación Electrónica (CIE).

Resultados: Se selecciona el Applet (SDM).

Condiciones: Estar conectado.

Camino: 1-4.

Caso de prueba: Seleccionar el Applet; SecureDataManager Applet.

Entrada: No hay conexión entre el lector de tarjetas inteligentes y la CIE.

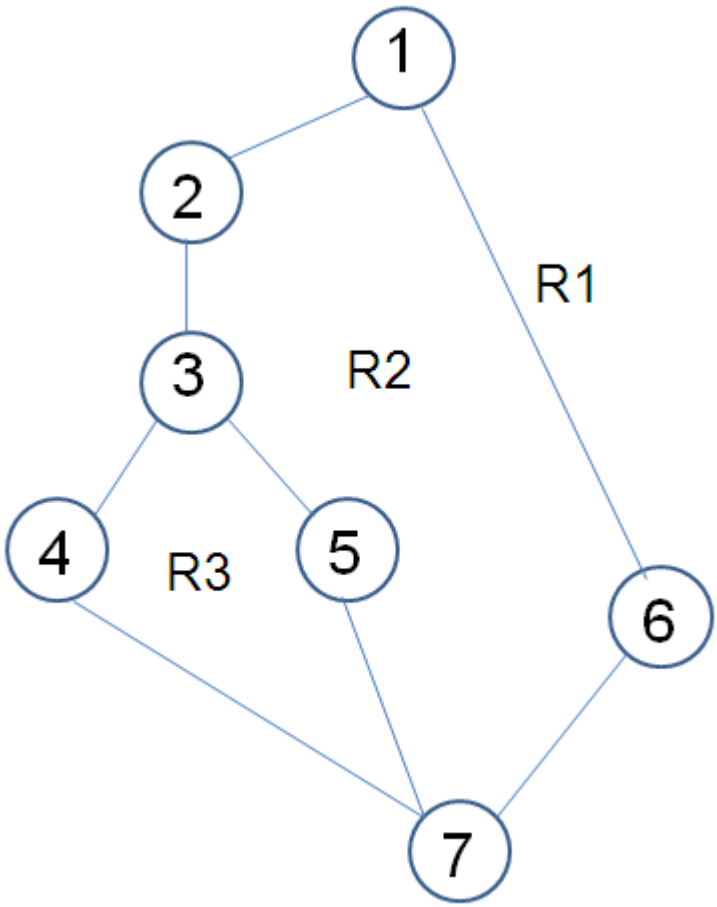
Resultados: No se efectúa el proceso de selección.

De acuerdo al segmento de código correspondiente al caso de uso ***"Autenticar Usuario PIN"***; se le realiza la prueba de caja blanca.

```
public string APDUCommandAuthenticationPIN (byte[] PIN)
{
    if (_Conect) (1)
    {
        AuthenticatePIN apdu = new AuthenticatePIN(PIN);(2)

        APDUResponse response = _CardReader.Transmit(apdu);(3)
        if (response.SW1 == (byte)0x90 && response.SW2 == (byte)0x00)(4)
        {
            _Authenticate_PIN = true;(4)
                return "Operación realizada satisfactoriamente.";(4)
        }
        else (5)
        {
```

```
        return "Error autenticando, chequee su PIN y trate de nuevo.";(5)
      }
    }
   return "You Se requiere estar conectado para realizar la operación.";(6)
} (7)
```



Complejidad Ciclomática.

V (G): *Número de regiones del grafo.*

V (G) = A – N + 2

V (G) = P + 1

A: *Número* de aristas del grafo.

N: *Número de n*odos.

P: *Número* de nodos predicados.

V (G) = 3

Caminos: 1-2-3-4-7, 1-2-3--5-7, 1-6-7.

Camino: 1-2-3-4-7.

Caso de prueba: Realizar la Autenticación por PIN.

Entrada: Se verifica que la conexión entre el Terminal de Servicios y la Cédula de Identificación Electrónica (CIE); se introduce el valor del PIN que el portador de la CIE posee.

Resultados: Se verifica el PIN; si la verificación es correcta se muestra el mensaje “La Autenticación es correcta.”.

Condiciones: Estar conectado e Introducir el valor del PIN.

Camino: 1-6-7.

Caso de prueba: Realizar la Autenticación por PIN.

Entrada: Se verifica que la conexión entre el Terminal de Servicios y la Cédula de Identificación Electrónica (CIE); se introduce el valor del PIN que el portador de la CIE posee.

Resultados: Si no se ha establecido la conexión entre el Terminal de Servicio y la CIE se muestra el mensaje “Necesita estar conectado para procesar.”.

Condiciones: Estar conectado.

3.6 Conclusiones

- El diagrama de despliegue muestra la distribución física que tendrá la solución para el control de acceso a la información de las entidades externas, brindando una distribución completa del acople de los distintos componentes de hardware que van conectados a la terminal de servicio.
- El diagrama de componentes muestra la distribución de los paquetes y componentes que están implementados para dar solución al tema de la presente tesis, este diagrama se basa en brindar una solución multiplataforma en la cual tanto en sistemas operativos propietarios como libres sea capaz de ejecutarse el Middleware y utilizar los componentes y API necesarios para la comunicación con los lectores de las tarjetas electrónicas inteligentes.
- Se determinaron las pruebas a realizarle a la solución validando y verificando las funcionalidades descritas; y dándole al sistema un nivel de calidad importante.

CONCLUSIONES

- Con el desarrollo del trabajo se logró cumplir con todos los objetivos, tanto generales como específicos, trazados para la implementación de una solución multiplataforma que permita la gestión de la información, de Entidades Externas a la ONIDEX, en la Cédula de Identificación Electrónica (CIE) de la República Bolivariana de Venezuela.
- El sistema desarrollado cumple con estándares definidos para el trabajo con tarjetas inteligentes; permitiendo contar con mecanismos de seguridad necesarios para gestionar de forma eficiente y segura el acceso a la información; brindando así confidencialidad e integridad a los servicios que Entidades Externas proveen a través de la CIE.
- La definición de un token de acceso el cual es elaborado por el Middleware de Control de Acceso y enviado por este hacia el Applet de Control de Acceso, posibilitando tener los datos que son contenidos en las extensiones de los certificados que son emitidos por la Autoridad Certificadora, en formato TLV, el cual es de fácil interpretación para el Applet.
- El Middleware de Control de Acceso cuenta con un grupo de clases que le posibilitan la creación de comandos APDU para cada uno de los requerimientos de la solución. Es importante reconocer que este Middleware utiliza el SmartCardFramework para gestionar la comunicación con las tarjetas electrónicas.
- El Applet de Control de Acceso cuenta con un sistema de ficheros el cual está definido por dos niveles lógicos: el global y el local, donde en el local se contiene los ficheros dedicados que sirven para organizar y brindar una estructura acorde a los requerimientos de la solución, y en el local se contienen los fichero elementales que son los contenedores de información referente a las Entidades Externas.
- El Applet de Control de Acceso cuenta con un proceso completo de verificación de las condiciones de acceso a los ficheros elementales, en donde se comienza comprobando el modo de acceso para la gestión de la información dentro de

este tipo de fichero, así como la verificación de las condiciones de seguridad que se necesitan cumplir para poder realizarlo.

- La solución cumple con los requerimientos establecidos para el desarrollo del mismo, brindando la posibilidad de ser utilizada en diferentes plataformas y brindando siempre una cómoda comunicación entre el cliente y el Applet.

RECOMENDACIONES

En las siguientes fases de la solución que en el presente trabajo se plantea; se exponen como recomendaciones:

- Realizar la modelación e implementación de la autenticación asimétrica entre la Cédula de Identificación Electrónica (CIE) y el Terminal de Servicio, para garantizar que tanto la CIE como el terminal de servicio se reconocen como válidos y poder intercambiar información de forma segura.
- Realizar la Autenticación del portador de la CIE mediante Match on Card (MoC) a través del mecanismo de interfaz compartida que provee el CryptoManager Applet.
- Continuar con la implementación del estándar ISO/IEC 7816 para soportar diversos mecanismos de almacenamiento y elementos de seguridad.

BIBLIOGRAFÍA

1. **ITU – T Grupo de Estudio.** ASN.1 Specification of basic notations.pdf. 2002.
2. **ITU – T Grupo de Estudio.** ASN.1 Encoding Rules.pdf. 2002.
3. **ISO/IEC.** info_isoiec7816-4{ed2.0}en.pdf. 2005.
4. **RSA, Laboratorios.** pkcs-1v2-0a1.pdf. 2000.
5. **RSA, Laboratorios.**. pkcs-1v2-1.pdf. 2002.
6. **Gemalto.** IAS Classic Applet. 2008.
7. RMI_Client_API_2_2_2.pdf.
8. Biometría en las Tarjetas Inteligentes.
9. MOC_FAQs.
10. PGuide.
11. [En línea] http://asn1.elibel.tm.fr/en/uses/index.htm.
12. [En línea] http://en.wikipedia.org/wiki/Abstract_Syntax_Notation_One.
13. [En línea] http://en.wikipedia.org/wiki/Generic_String_Encoding_Rules.
14. [En línea] http://en.wikipedia.org/wiki/Distinguished_Encoding_Rules.
15. [En línea] http://en.wikipedia.org/wiki/Packed_Encoding_Rules.
16. [En línea] http://en.wikipedia.org/wiki/Basic_Encoding_Rules.
17. [En línea] http://www.dnielectronico.es.
18. [En línea] http://www.fineid.fi/vrk/fineid/home.nsf/pages/index_eng.
19. [En línea] http://www.id.ee.
20. [En línea] http://www.vaestorekisterikeskus.fi/vrk/home.nsf/pages/index_eng.
21. [En línea] http://tataware.com/tesis_maestria/13%20-%20Anexo%20B%20-%20Est%E1ndares%20ISO_IEC%207816.pdf.
22. **Toyserkani, Ali.** Java Card Technology. 2007.
23. **Sharykin, Raman.** Java Card Programming. 2006.

REFERENCIAS BIBLIOGRÁFICAS

[SUN 2], Sun Microsystems, 1999. . *Java Card 2.1.1 Runtime Environment (JCRE) Specification.* [En línea] http://java.sun.com/products/javacard/javacard21.html .

Centro Nacional de Tecnologías de Información . 2009. MeRinde. [En línea] 2009. http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=139&Itemid=194.

CodeProject. 2008. *CodeProject.* [En línea] 10 de Noviembre de 2008. http://CodeProject/PKCS_Standards_NET Framework_help.com.

Estándar ISO/IEC 7816. 2006. *Estándares ISO/IEC 7816.* 2006.

Mendoza, Yurdik Cervantes. 2008. *BIOMETRÍA EN LAS TARJETAS INTELIGENTES.* Ciudad de La Habana : s.n., 2008.

Precise Biometrics. 2005. *Precise BioMatch™ J 3.0 Manual.* 2005.

Proenza, Y. 2005. *Introducción al modelo conceptual, .* Ciudad de la Habana, Universidad de Ciencias Informáticas : s.n., 2005.

SmartCard Technology (Smartcard '99), USENIX, May 1999. *[SUN3]Secure Object Sharing in Java Card.*

Software, Departamento Central de Ingeniería de Softwera. 2004. *Software, E.d.p.d.I.d., Flujo de trabajo Captura de requisitos. Modelo de Negocio.* Ciudad de la Habana, Universidad de Ciencias Informáticas : s.n., 2004.

STANDARD, INTERNATIONAL. 2006. *INTERNATIONAL STANDARD ISO/IEC 7816-4.* 2006.

Sun Microsystems, 1999. *Java Card 2.1 Runtime Environment (JCRE) Specification.* [En línea] http://java.sun.com/products/javacard/javacard21.html.

technology, Information. 2002. *Information technology – Abstract Syntax Notation One (ASN.1).* 2002.

thefreedictionary Applet. 2008. *thefreedictionary.* [En línea] 25 de Marzo de 2008. http://www.thefreedictionary.com/applet.

thefreedictionary Middleware. 2008. *thefreedictionary.* [En línea] 25 de Octubre de 2008. http://encyclopedia2.thefreedictionary.com/middleware.

Universidad de la República del Uruguay. 2004. *Uso de los circuitos integrados sin contacto.* 2004.

Wikipedia PKI. 2008. *Wikipedia.* [En línea] 2 de Noviembre de 2008. http://en.wikipedia.org/wiki/PKI.

ANEXOS

ANEXO 1 PKSC #15

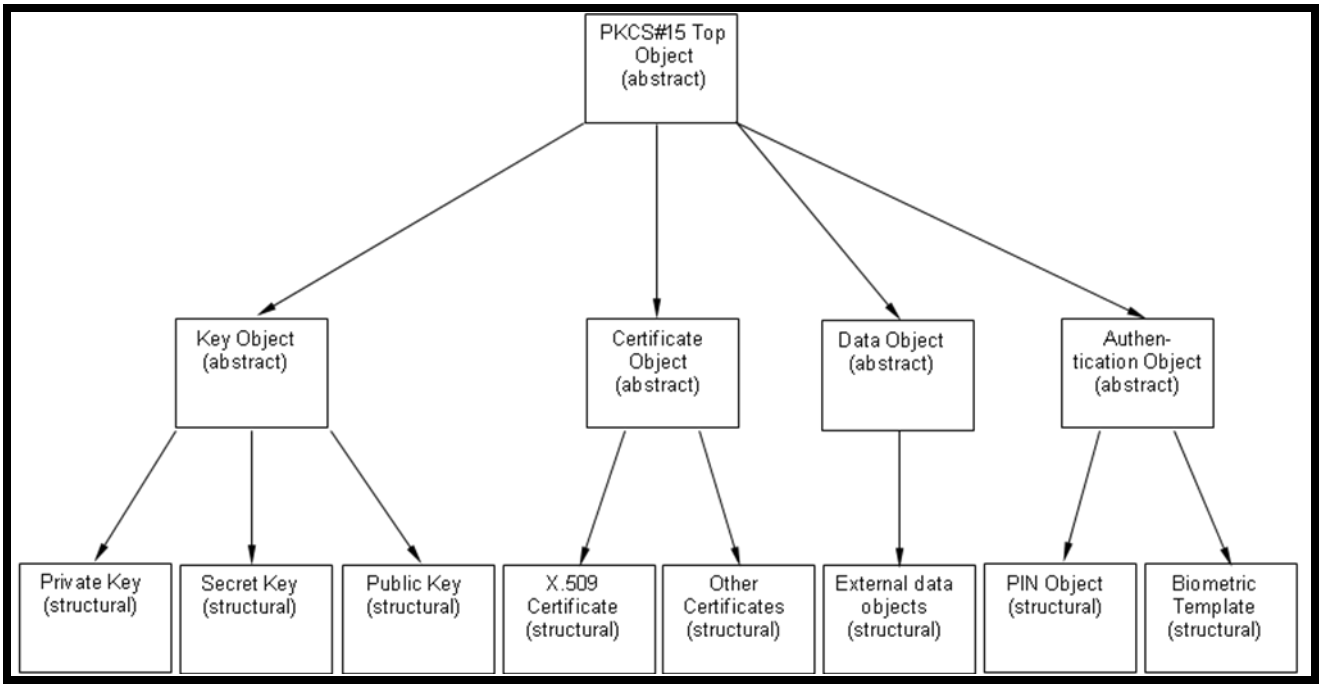


Figura 27: Diagrama Estructural PKCS #15.

ANEXO 2 Verificación Biométrica

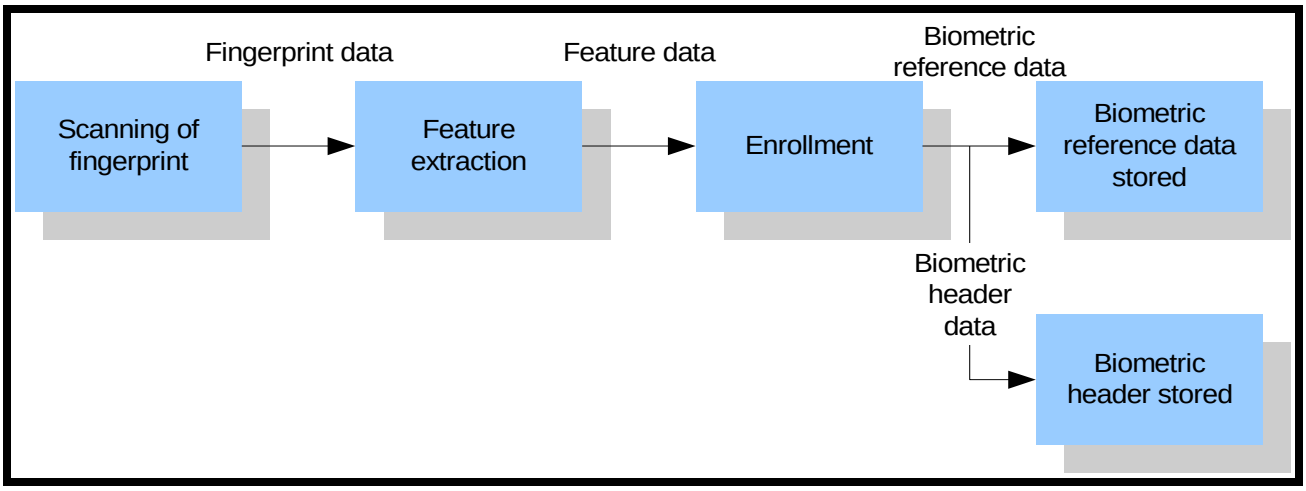


Figura 28: Proceso de enrolamiento de huellas.

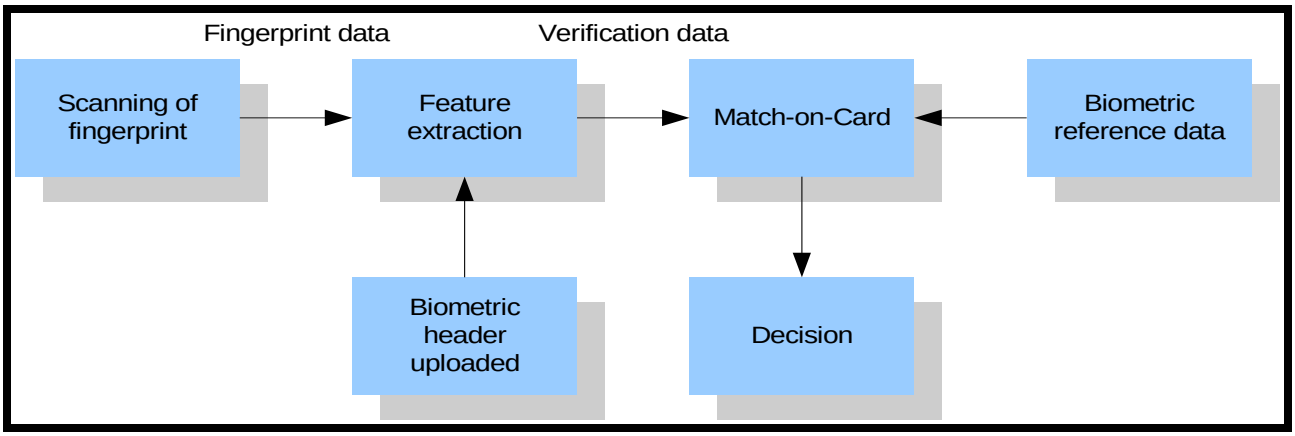


Figura 29: Proceso de verificación de huellas dentro de la tarjeta.

ANEXO 3 Plantillas del archivo de control de información para ficheros.

Offset	Data	Description
0	6Fh	Tag of FCI template
1	L	Length of FCI data
2	83h	Tag of File ID
3	02h	Length of File ID
4–5	File ID	Value of File ID
6	8Ch	Tag of security attributes
7	L	Length of security attributes
8	AM	Access mode byte
9–(8+X)	SC	Security condition bytes (X)
9+X	84h	Tag of DF Name
10+X	L	Length of DF Name
11+X –	DF name	Value of DF name (up to 16 bytes)

Figura 30: Plantilla del archivo de control de la información (File Control Information) para ficheros dedicados.

Offset	Data	Description
0	6Fh	Tag of FCI template
1	L	Length of FCI data
2	81h	Tag of File Size
3	02h	Length of File Size
4–5	File Size	Value of File Size
6	82h	Tag of FDB
7	01h	Length of FDB
8	FDB	Value of FDB
9	83h	Tag of File ID
10	02h	Length of File ID
11–12	File ID	Value of File ID
13	8Ah	Tag of Life Cycle Status byte for file
14	01h	Length of Life Cycle Status byte for file
15	Var	Value of Life Cycle Status byte for file
16	8Ch	Tag of security attributes
17	L	Length of security attributes
18	AM	Access mode byte
19–(18+X)	SC	Security condition bytes (X)

Figura 31: Plantilla del archivo de control de la información (File Control Information) para ficheros elementales.

ANEXO 4 Estados del Applet de Control de Acceso

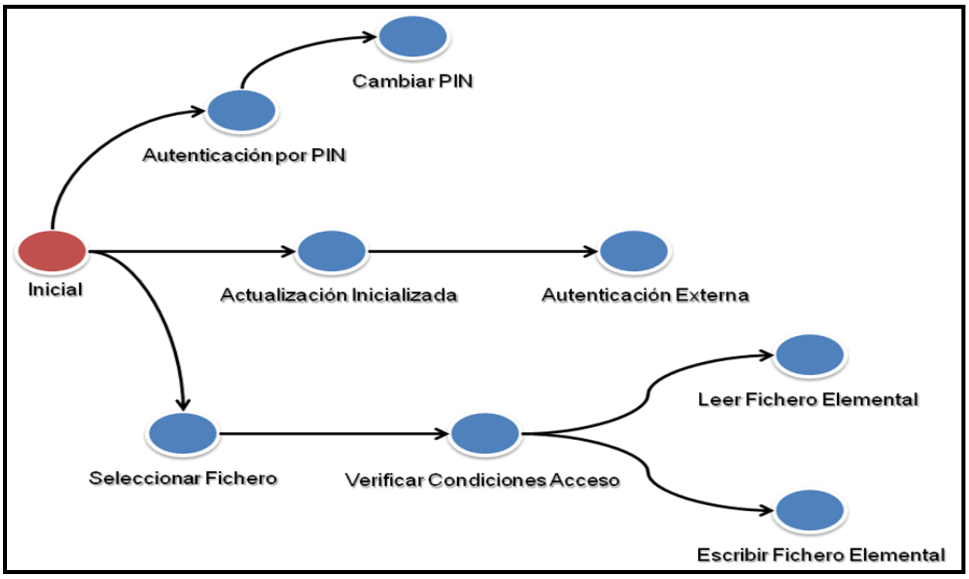


Figura 32: Diagrama de estados del Applet de Control de Acceso.

ANEXO 5 Certificado Digital

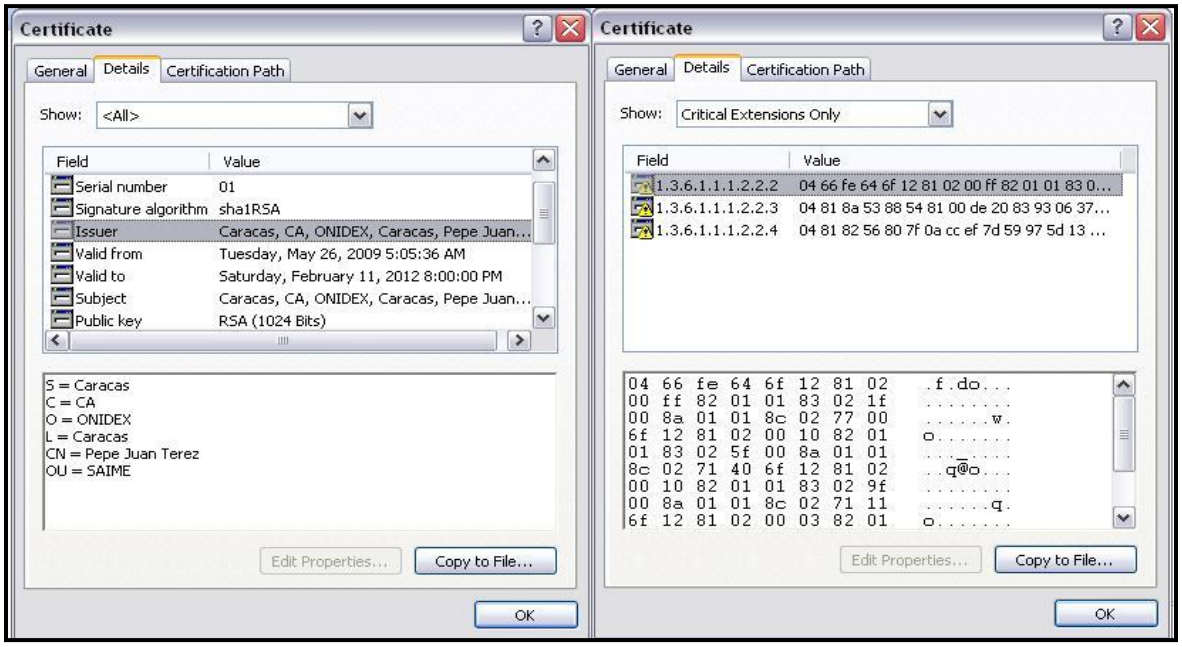


Figura 33: Detalles del Certificado.

Anexo 6 Detalles de las principales clases del Diagrama de Clases del SDM

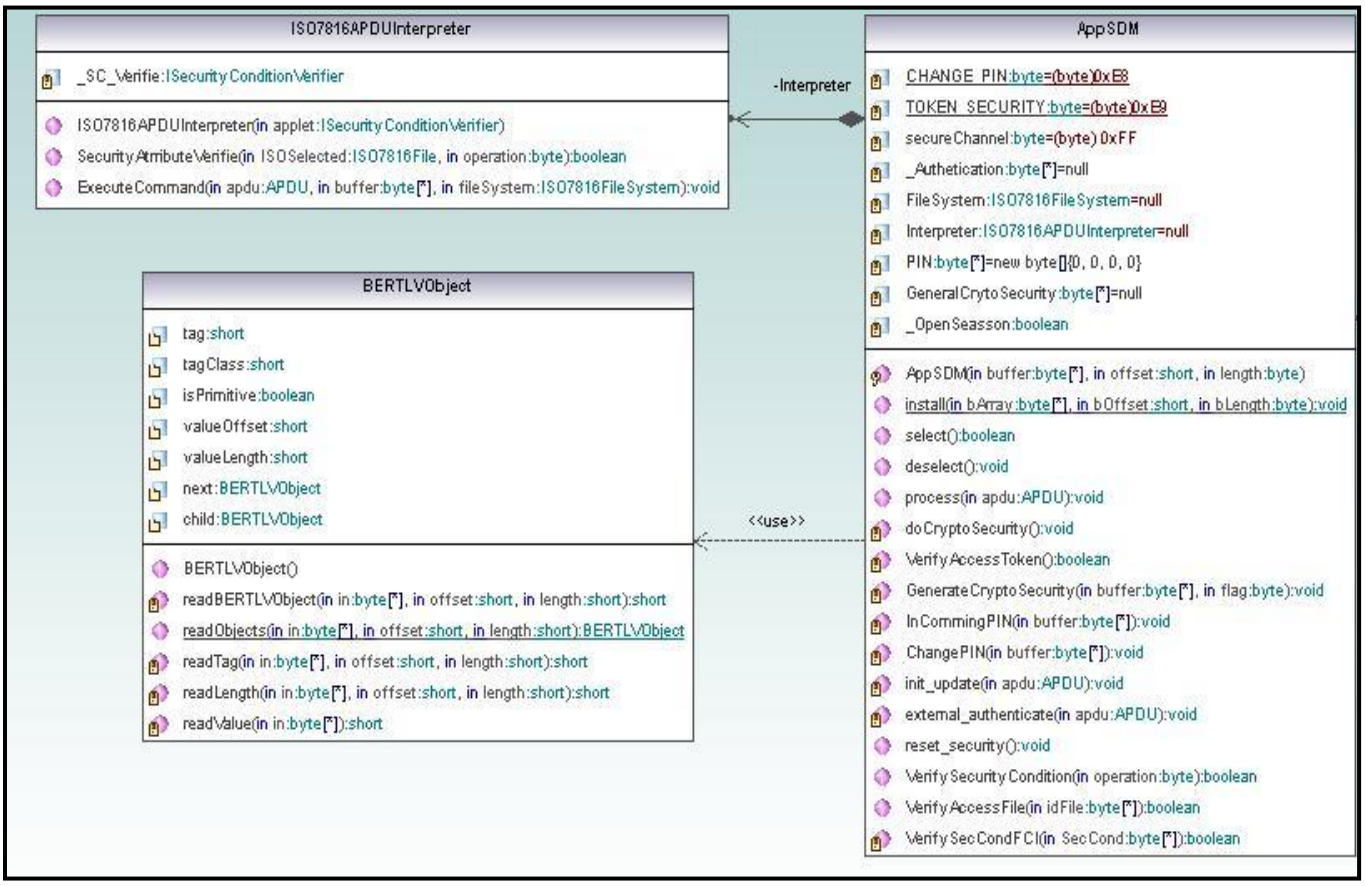


Figura 34: Detalles del Diagrama de Clases del Applet.

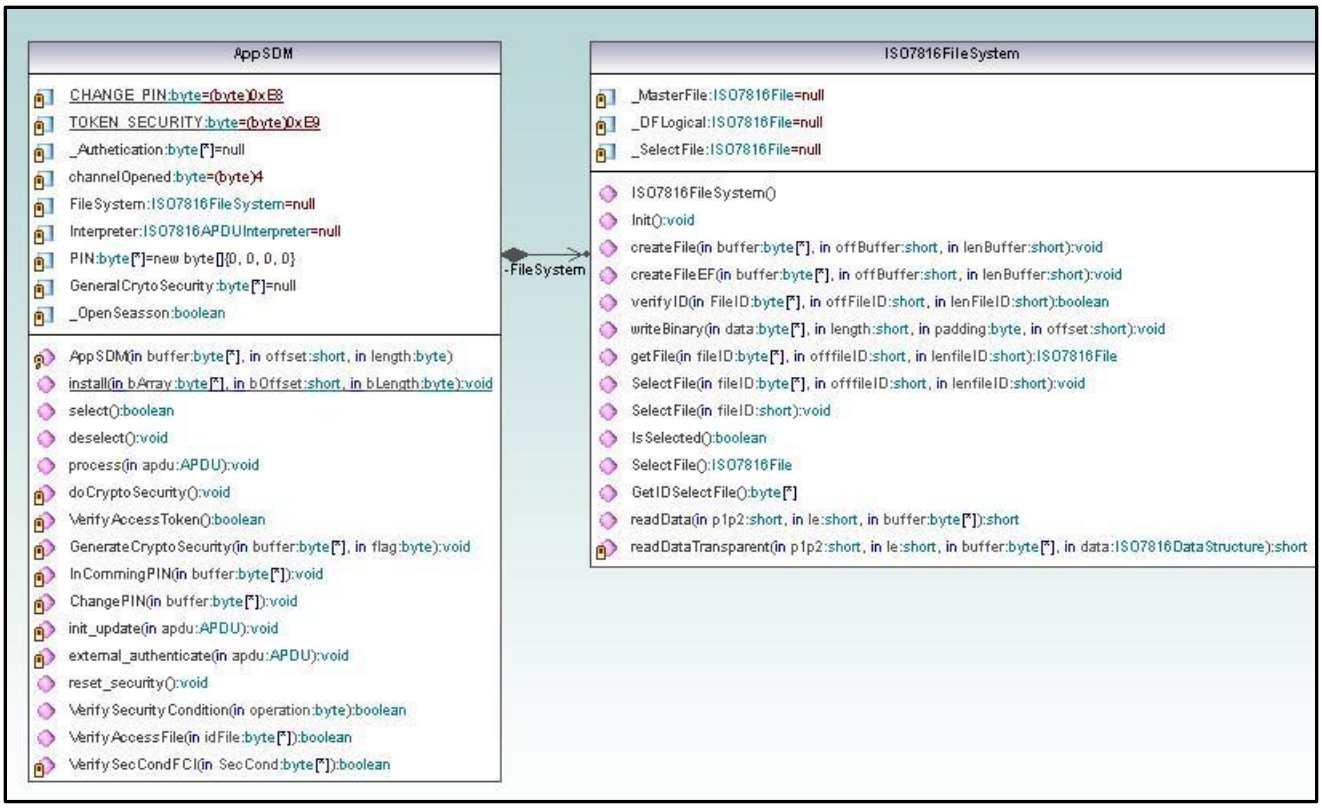


Figura 35: Detalles del Diagrama de Clases del Applet.

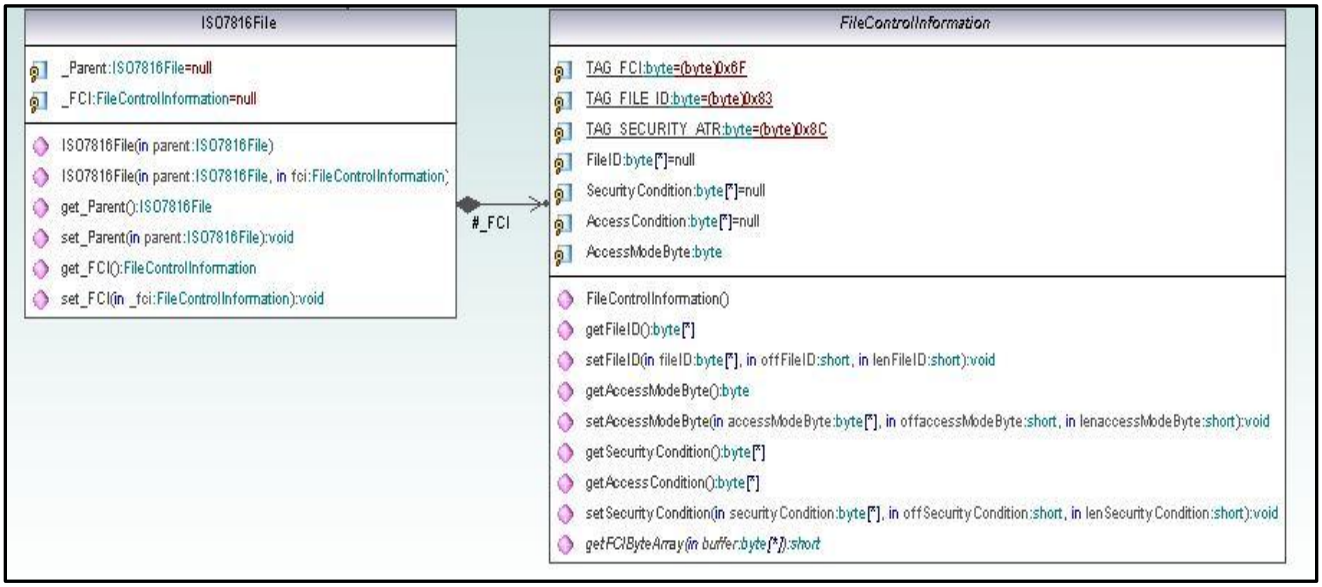


Figura 36: Detalles del Diagrama de Clases del Applet.

Anexos 7 Diagramas de Secuencia

Diagrama de Secuencia “CU Autenticar Usuario MoC”

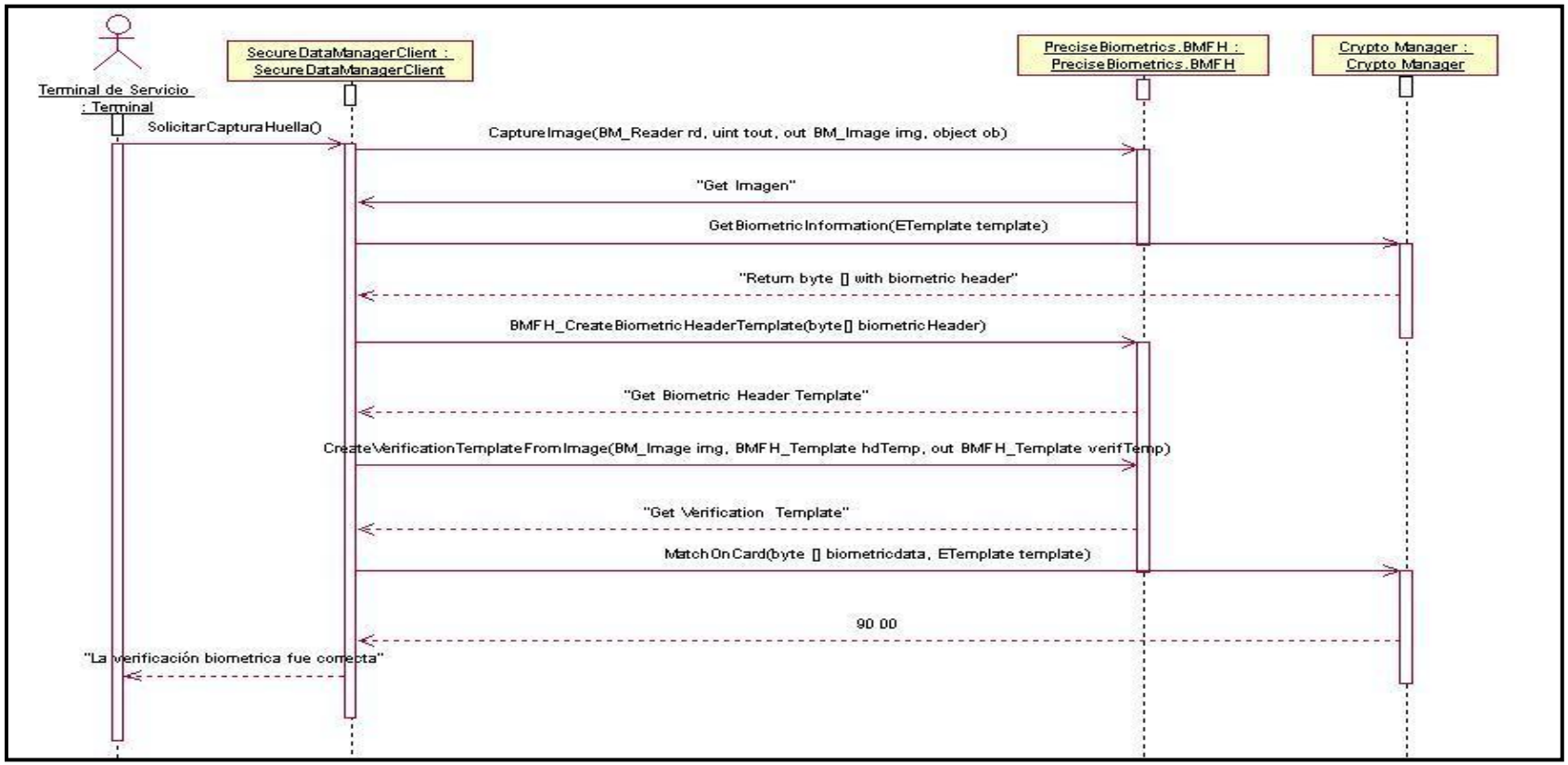


Figura 37: Diagrama de Secuencia "Autenticar Usuario MoC".

Diagrama de Secuencia "Inicializar Comunicación"

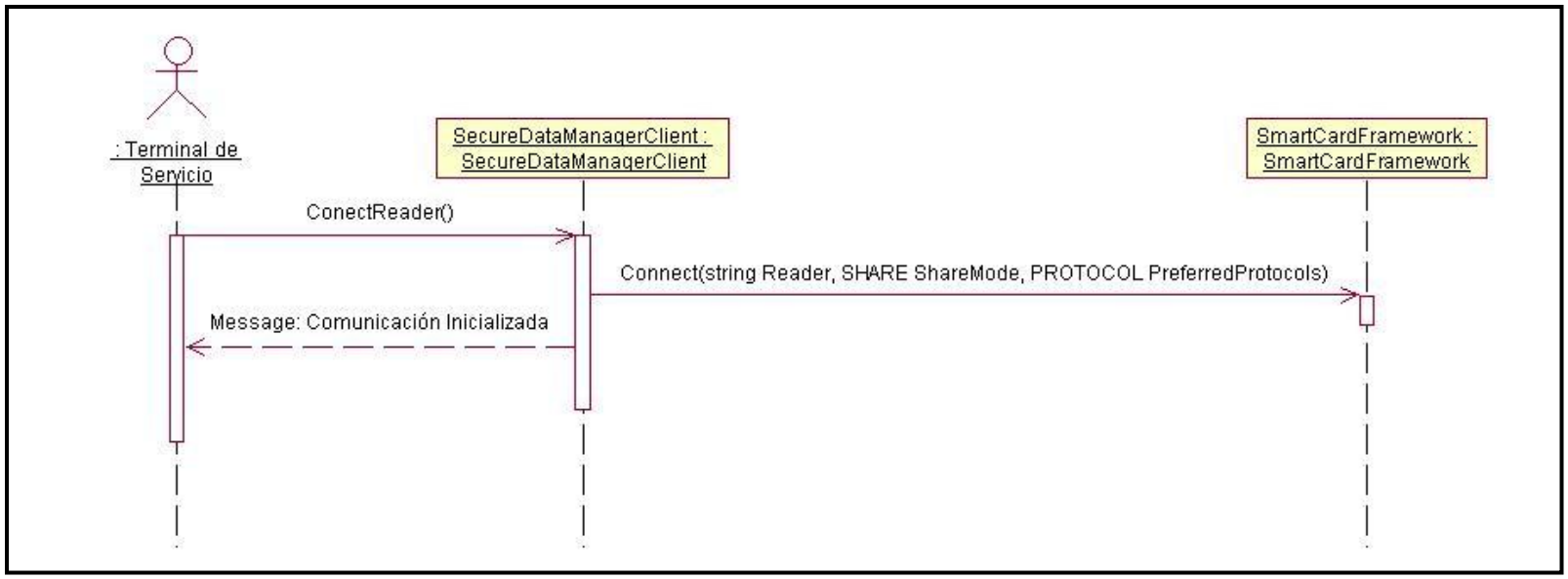


Figura 38: Diagrama de Secuencia "Inicializar Comunicación".

Diagrama de Secuencia "Finalizar Comunicación"

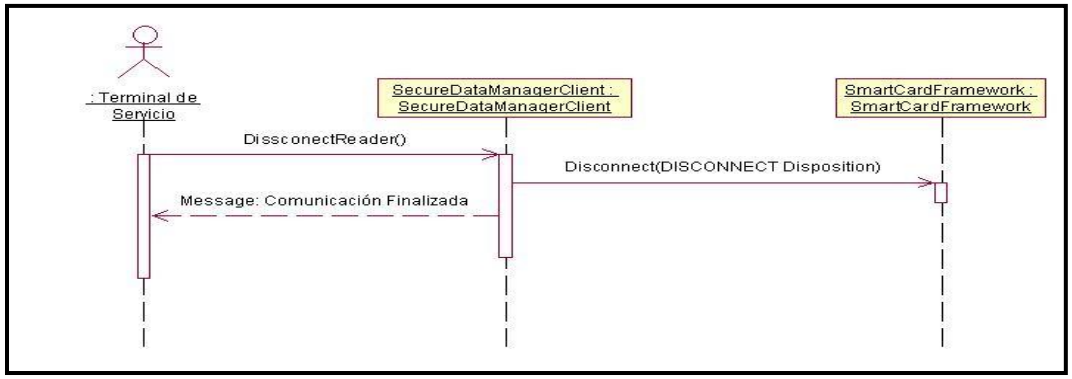


Figura 39: Diagrama de Secuencia "Finalizar Comunicación".

Diagrama de Secuencia "Verificar Certificado de Acceso"

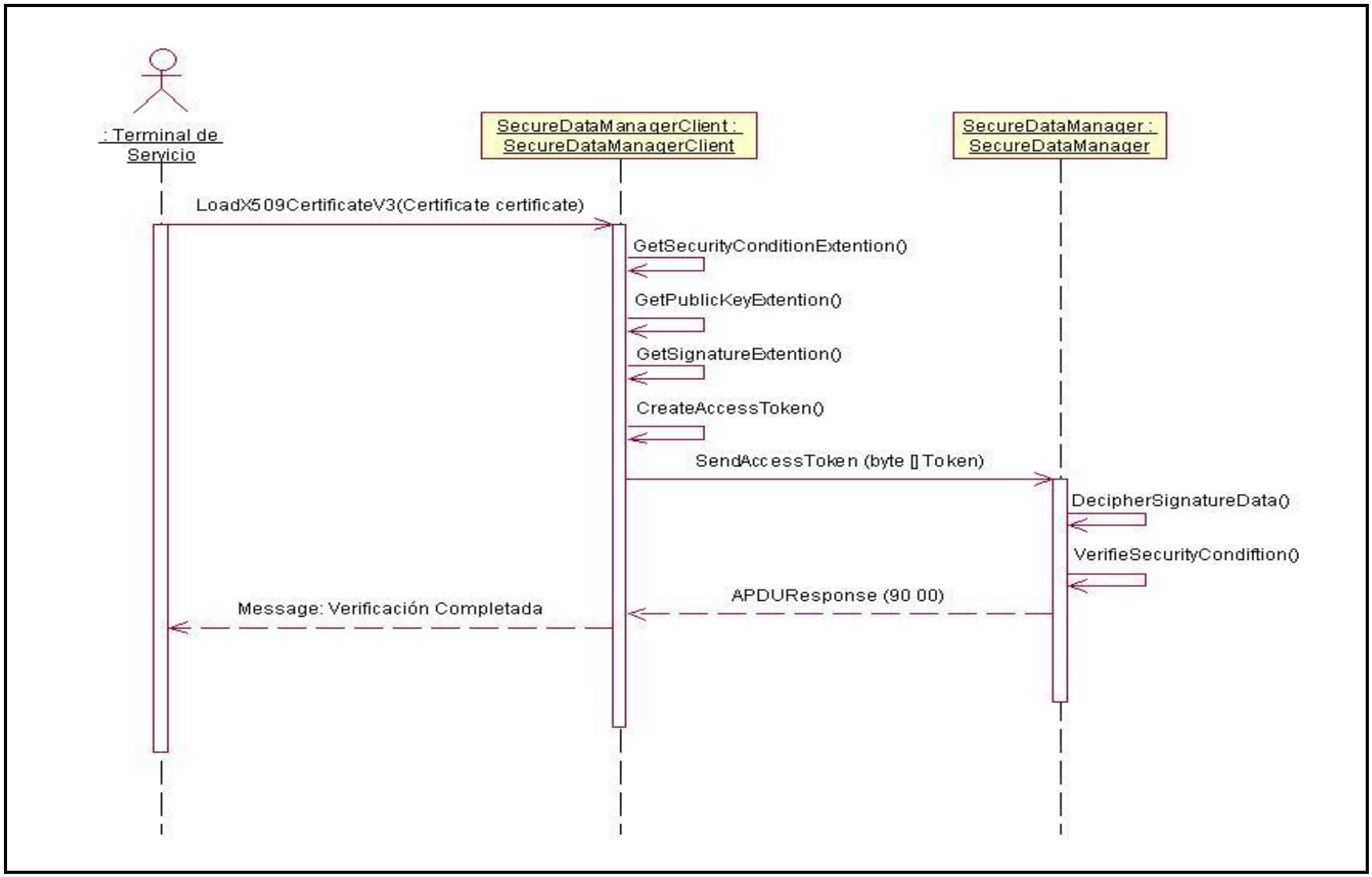


Figura 40: Diagrama de Secuencia "Verificar Certificado de Acceso"

Diagrama de Secuencia "Obtener Condiciones Acceso"

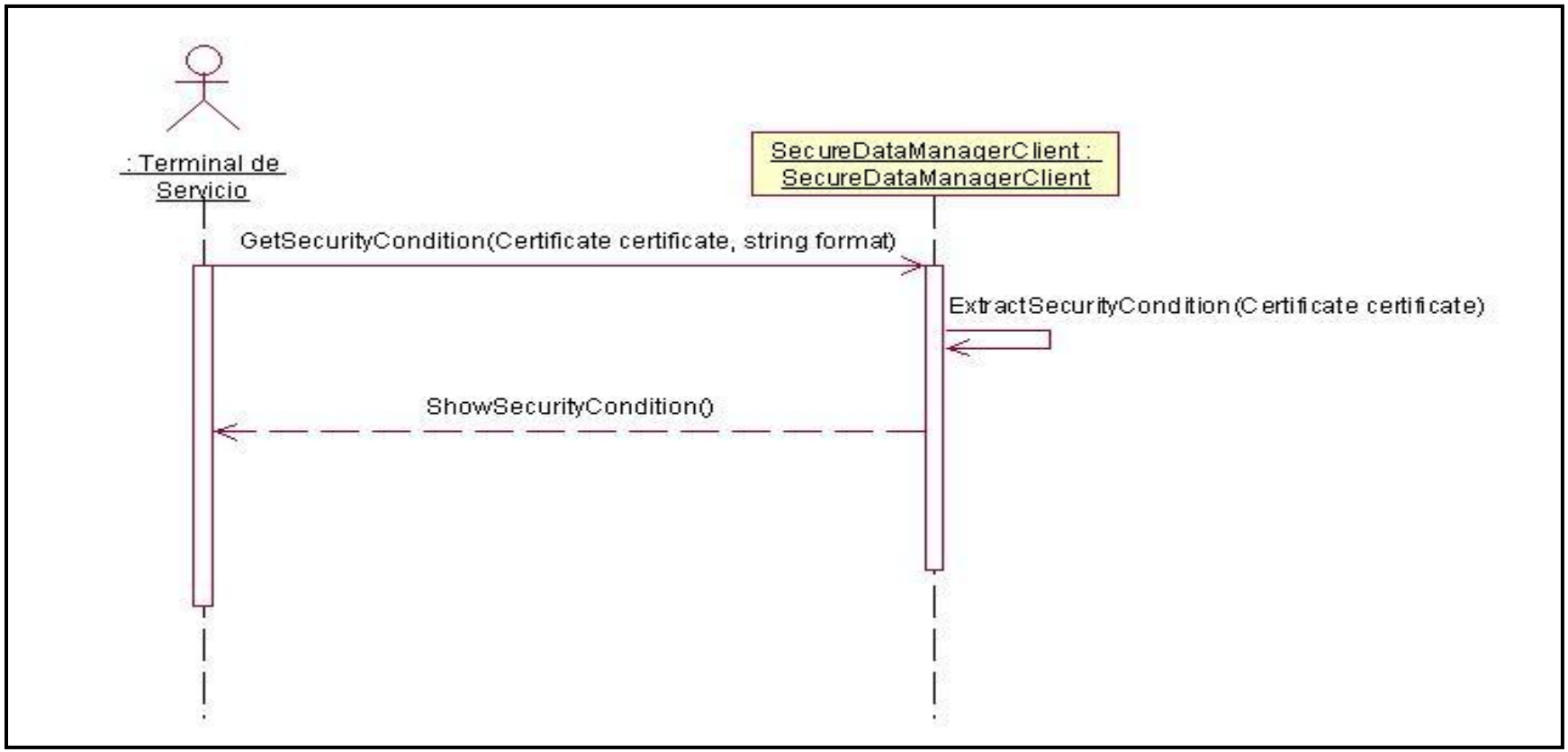


Figura 41: Diagrama de Secuencia "Obtener Condiciones Acceso".

Diagrama de Secuencia "Leer Información"

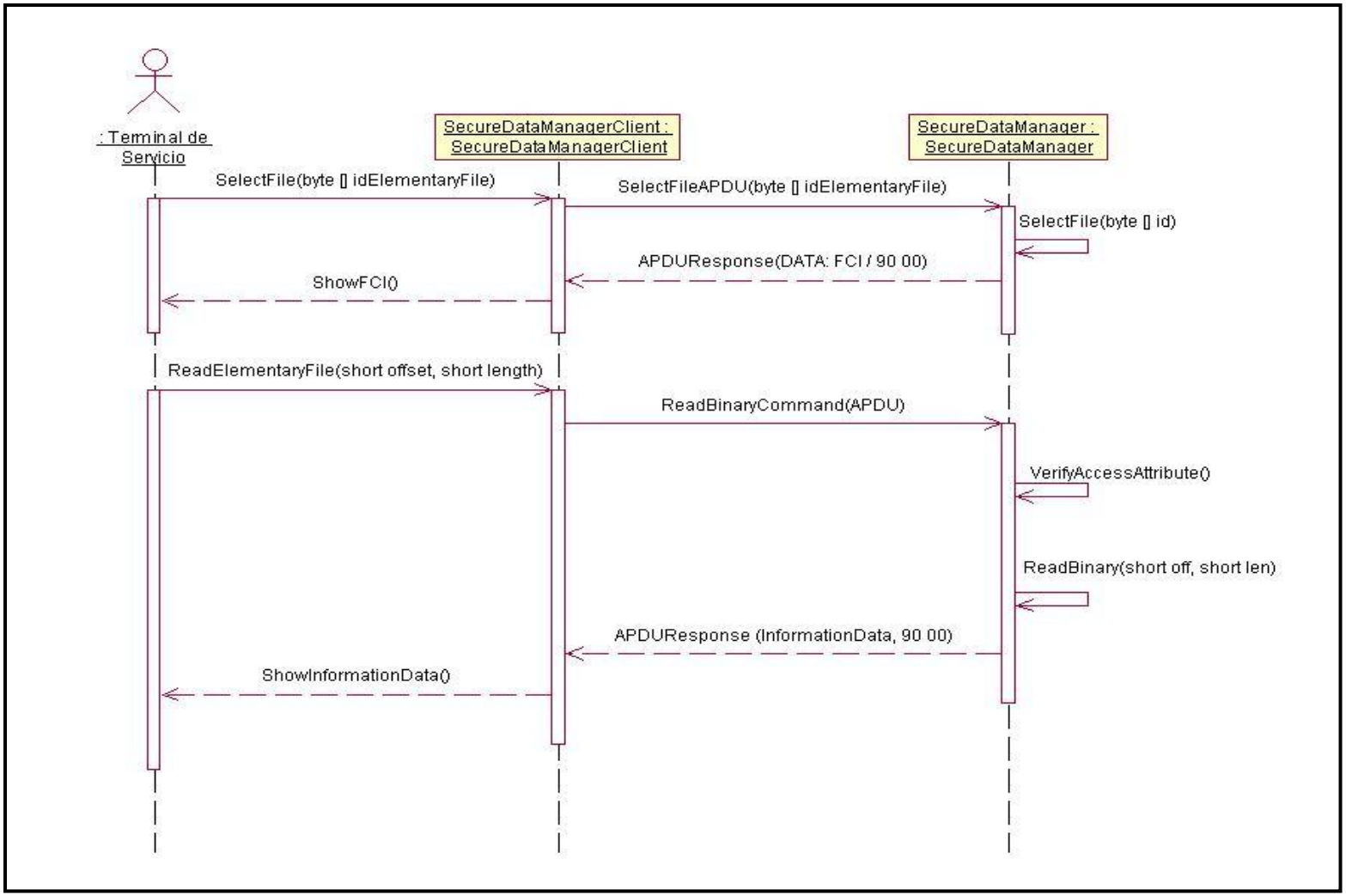


Figura 42: Diagrama de Secuencia "Leer Información".

Anexo 8 Descripción de Principales Flujo de Procesos

Proceso de Autenticación de Usuario.

Figura 43: Proceso de Autenticación de Usuario.

Proceso de Gestión de Información.

Figura 44: Proceso de Gestión de Información.

GLOSARIO DE TÉRMINOS

ONIDEX: Oficina Nacional de Identificación y Extranjería.

EE: Entidades Externas a la ONIDEX que tienen interés en guardad información de los servicios que prestan en la CIE.

APDU: Protocolo de Unidad de Datos de Aplicaciones.

CIE: Cédula de Identificación Electrónica, documento de identificación en la República Bolivariana de Venezuela.

Applet: Aplicación que se ejecutan dentro de las tarjetas inteligentes y gestiona la información almacenada en ella.

SDM: Aplicación para la Gestión Segura de Datos en la CIE.

EF: Fichero elemental.

DF: Fichero dedicado.

MF: Fichero maestro.

FCI: Información de control de ficheros.

MoC: Comparación de huellas en la tarjeta inteligente.

PIN: Número de Identificación Personal.

SW: Palabra de Estado.

TLV: Etiqueta, Longitud y Valor.

PKI: Infraestructura de Llave Pública.

PKCS: Estándar Criptográfico de Llave Pública.

OCSP: Protocolo de Estado de Certificados en Línea.