Universidad de las Ciencias Informáticas

Facultad 1

Implementación del Esquema de Autenticación del Applet Secure Data Manager

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

Autor: Abraham Armas Pérez

Tutor: Ing. Ander Sánchez Jardines

"Ciudad de La Habana. Junio, 2010"

Pensamiento

“NO SE PUEDE FORMAR EL CARÁCTER Y EL VALOR DEL HOMBRE QUITÁNDOLE SU INDEPENDENCIA, SU LIBERTAD Y SU INICIATIVA."

ABRAHAM LINCOLN.

Declaración de Autoría

Declaro que somos los únicos autores de este trabajo y autorizamos al Centro de Identificación y Seguridad Digital de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

_____

Abraham Armas Pérez

_____

Ing. Ander Sánchez Jardines

Dedicatoria

Este trabajo de diploma está dedicado especialmente a mi madre. Nadie se ha esforzado tanto ni le ha puesto tanto empeño a que yo haya llegado hasta aquí. Es como ella dice el titulo viene con dos nombre el suyo y el mío. Existe también otra persona que con agradecerle solo no bastaría, Javier López-Belio.

Agradecimientos

Son muchas las personas que podrían venir a la mente en este momento a las cuales estaría muy agradecido, pero la verdad que se haría de noche si intentara mencionarlas a todas. Primero que todo agradecer a Ander, mi tutor, mi compañero de tesis y tutor de todas las tesis del departamento de tarjetas. Agradecer a todos los miembros del tribunal, especialmente a Edistio que si lo molesto una vez más lo tengo que poner como tutor. Agradecer a personas muy importantes en mi transcurso por esta universidad, Marianela Orozco, Guillermo Zerquera y Yadira García Huelga. Agradecer a Sabina Rivera mi mama2 y gran amiga, son muchas las cosas que le pudiera decir. Al mejor actor de los últimos tiempos que sé a graduado en el ISA Yaser Rivero Ituria. Alberto Pérez (Stripper), como el quedan pocos. Agradecer a mi hermano, casi padre Jendry y a su esposa Cris. No por ultimo menos importante sino todo lo contrario, agradecer a alguien que representa mi mas grande tesoro, quien ha estado en todos los momentos mas difíciles de esta tesis, brindándome un gran apoyo y estabilidad psicológica, mi futura abuela de mis nietos Laura Tariche Arrinda. En fin al departamento de tarjetas inteligentes, y a todas las amistades y conocidos de la UCI.

Resumen

En la actualidad las tarjetas inteligentes son usadas como documento de identificación personal. Las mismas poseen características como la confidencialidad y la seguridad de la información, lo que ha llevado a que estas sean una solución verdaderamente efectiva contra la falsificación de dichos documentos.

Varios países en el mundo se han inclinado por el uso de esta tecnología, dentro de los que se encuentra la República Bolivariana de Venezuela, la cual tiene algunos convenios con la Universidad de Ciencias Informática.

El Centro de Identificación y Seguridad Digital (CISED) de la Universidad de Ciencias Informáticas tiene el objetivo de renovar el Sistema de Identificación, Migración y Extranjería venezolana, basado en el rediseño de procesos integrados con las más recientes tecnologías y orientado a la satisfacción de las necesidades. CISED ha implementado el Applet Secure Data Manager para la Cédula de Identificación Electrónica venezolana con el objetivo de almacenar información referente a la persona, y de interés para las instituciones o entidades que necesiten utilizarla y así automatizar sus propios procesos.

Se hace necesario implementar un esquema de autenticación con el objetivo de lograr la autenticación del portador de la tarjeta y validar las diferentes entidades gubernamentales que necesitan tener acceso a dicha información.

El documento recoge los resultados de la investigación realizada, describiéndose las principales características de la arquitectura y el diseño del sistema propuesto. Se describen las herramientas, tecnologías utilizadas y los artefactos generados en el proceso de desarrollo.

Palabras Claves:

Proceso; Sistema; Cédula; Entidades; Gestión de la Información.

Índice de contenido

Índice de Tablas.

Índice de Figuras.

Introducción

Los inicios de las tarjetas inteligentes se remontan a la década de los 70 en Francia, pero no se puede hablar de una tarjeta inteligente tal y como hoy día hasta los años 80. En esta década la banca francesa llevó a cabo un proyecto que incluía la prueba con 100 000 tarjetas inteligentes. En este proyecto participaron las empresas Bull, Philips y Schlumberger.

Una tarjeta inteligente contiene un microprocesador con CPU, RAM y ROM, su forma de almacenamiento puede ser EPROM o EEPROM, la ROM consta de un sistema operativo que maneja la asignación de almacenamiento de la memoria, la protección de accesos y maneja las comunicaciones. La interfaz de comunicación de las tarjetas inteligentes, están diseñadas para comunicarse a través de un conjunto de 8 pines.

Estas tarjetas, tienen además de capacidad de memoria, un microprocesador que les permite realizar procesamiento local de los datos y complejos cálculos algorítmicos. Esto posibilita que las tarjetas inteligentes se puedan utilizar para el intercambio de información encriptado y para implementar avanzados mecanismos de seguridad.

Las tarjetas inteligentes son consideradas almacenes de información; gracias al chip que contienen, pueden guardar datos de diferentes índoles: financiera, laborales, sanitaria o personales. Dentro de la tarjeta, el chip puede realizar las mismas funciones que un ordenador. Obteniendo como ventajas, entre otras, su capacidad, alta portabilidad, resistencia material y por encima de todo, la seguridad de acceso autorizado a los datos.

Desde las tarjetas más sencillas, incluyen mecanismos de control de acceso. El más clásico consiste en la utilización de un PIN (Personal Identification Number o Número de Identificación Personal) que se almacena en texto claro o en ocasiones cifrado. El sistema solicita al usuario que introduzca el valor de su PIN; realizando a continuación diversas operaciones con el mismo; como la identificación de la persona. Las tarjetas más avanzadas están sustituyendo las técnicas de control de acceso a través de PIN por identificadores biométricos, como la huella digital o el iris del ojo, que son mucho más difíciles de obtener y falsificar. Logrando una alta seguridad y confidencialidad en la autenticación entre la tarjeta y el portador de la misma.

Las tarjetas pueden incluir también métodos criptográficos. Esta técnica se utiliza para establecer una comunicación segura entre la tarjeta y el lector, y pueden emplearse, para brindar seguridad a aplicaciones dentro del sistema al que está asociada la tarjeta.

En algunas tarjetas se implementa criptografía de clave secreta como por ejemplo DES, y actualmente existen tarjetas que trabajan con criptografía de clave pública, como por ejemplo

RSA (Rivest, Shamir y Adleman). Estos tipos de tarjetas sirven de apoyo a diversas aplicaciones para realizar firma digital.

Las tarjetas que contienen criptografía asimétrica garantizan el almacenamiento de la clave secreta de forma segura, ya que la clave que normalmente es generada por un dispositivo externo, se almacena directamente en la tarjeta en una zona de memoria protegida donde no puede reescribirse y leerse desde el exterior de la misma, de forma que el proceso de cifrado y descifrado que utiliza esta clave secreta se realiza siempre dentro de la propia tarjeta. También, dependiendo de la capacidad de almacenamiento de la tarjeta, se puede almacenar en la misma llave, certificados y firma digital.

Hoy en día las tarjetas inteligentes son usadas como documento de identificación personal. Características como la confidencialidad y la seguridad de la información ha llevado a ser una solución verdaderamente efectiva contra la falsificación de estos documentos de identificación personal. Son muchas las ventajas que pueden traer las tarjetas inteligentes como documento de identificación personal, una de las ventajas seria el almacenamiento de información referente a los servicios que varias entidades pudieran guardar en ella y la seguridad al acceder a cada uno de estos datos.

Son varios los países que utilizan las tarjetas inteligentes como documentos de identificación nacional, dentro de los que se encuentran Malasia pionera en el uso de esta tecnología, España, Finlandia, etc.; dentro de los países que pretenden dar el paso de avance hacia el uso de esta tecnología se encuentra la República Bolivariana de Venezuela que a pesar de ser un país en vía de desarrollo lleva a cabo un gran avance en la informatización de la sociedad. Para la cual se ha implementado el Applet Secure Data Manager (SDM) en la Cédula de Identificación Electrónica (CIE), con el objetivo de almacenar información referente a la persona de interés para las instituciones o entidades que necesiten utilizar la CIE y así automatizar sus propios procesos.

El mecanismo de autenticación para validar el portador de la CIE es mediante el PIN, que posee baja seguridad ya que es un número que cualquier persona puede tener, siendo vulnerable a la hora de identificar el portador de la tarjeta. Por otra parte, una vez verificado el PIN cualquier entidad que interactúe con la tarjeta tiene acceso a la gestión de la información contenida dentro de la CIE.

Partiendo de todo lo anteriormente expuesto se tiene la siguiente **Situación Problémica,** resumida en**:**

Para acceder a la información que se almacena dentro del applet se necesita verificar al portador de la tarjeta y la validez del certificado emitido para la Entidad Externa, por lo que se

hace necesario implementar un esquema de autenticación que permita brindar una mayor seguridad y confidencialidad a la gestión de esta información.

Derivado de la situación anteriormente expuesta se plantea el siguiente **problema científico**

¿Cómo garantizar la ejecución del applet SDM solo por parte del portador autentico y de las entidades externas autorizadas para preservar la autenticidad e integridad de la información contenida en la Cédula de Identificación Electrónica de la República Bolivariana de Venezuela?

Como **objeto de estudio,** se define implementación de esquemas de autenticación para la ejecución de applets instalados en tarjetas inteligentes.

El **campo de acción** se enmarca en el proceso de implementación de esquemas de autenticación para la ejecución de los applets instalados en la Cédula de Identificación Electrónica de la República Bolivariana de Venezuela.

La presente investigación propone la siguiente **idea a defender:** Si se implementa un esquema de autenticación para la ejecución del Applet Secure Data Manager instalados en la Cédula de Identificación Electrónica de la República Bolivariana de Venezuela, entonces se logrará que el acceso a la información sea de forma segura y confidencial.

Para dar solución al problema existente se ha tomado como **objetivo genera**l: Implementar un esquema de autenticación que permita la ejecución el Applet Secure Data Manager instalado en la Cédula de identificación Electrónica de la República Bolivariana de Venezuela.

Para dar respuesta a la interrogante presentada en este trabajo y con los objetivos trazados se plantea el cumplimiento de las siguientes **tareas de la investigación**:

1. Realizar una caracterización sobre la arquitectura de tarjeta inteligentes como base para la implementación del applet.
2. Caracterizar las tecnologías y estándares relacionados con tarjetas inteligentes.
3. Seleccionar las herramientas que permitan implementar el esquema de autenticación.
4. Documentar la situación actual de Mecanismos de Autenticación.
5. Caracterizar los algoritmos para la autenticación asimétrica en applets.
6. Investigar sobre la implementación y utilización de las interfaces compartidas en JavaCard.
7. Implementar el esquema de autenticación en la CIE; con todos los estándares y normas establecidas.

8. Determinar las pruebas que se le deben realizar.
9. Realizar las pruebas de calidad definidas.

La investigación estará sustentada en los siguientes métodos científicos:

Métodos teóricos:

El empleo del **método histórico-lógico** posibilito la comprensión lógica del objeto de estudio haciendo un análisis riguroso de sus antecedentes y el proceso evolutivo por el cual han transitado todas las tecnologías relacionadas con las tarjetas inteligentes y los servicios en línea.

El analítico-sintético permitió la consulta de diversas fuentes bibliográficas y la extracción de los elementos más importantes que se relacionan con el objeto de estudio. Será de gran importancia en el estudio del estado del arte.

El método inductivo-deductivo fue valioso para poder hacer generalizaciones del conocimiento a partir del estudio de situaciones concretas y viceversa, lo que es importante en cualquier investigación científica.

La abstracción de diversas situaciones y la representación de sus características fundamentales desde determinadas perspectivas fue de gran importancia en el desarrollo de esta investigación, por lo que se usará el método modelación, muestra de ello serán todos los diagramas y modelos que se producirán a lo largo de la misma.

Métodos empíricos:

Como métodos empíricos se usará la entrevista con el propósito de obtener información, experiencias, ideas, puntos de vistas, que contribuyan al desarrollo de la investigación y aporten conocimientos específicos del tema.

La observación de la percepción planificada dirigida a un fin y relativamente prolongada de un hecho o fenómeno, permitirá analizar y crear un esquema de autenticación y la utilización de herramientas que existen para los mismos.

El trabajo se encuentra estructurado de la siguiente forma:

Capítulo 1: Fundamentación Teórica, Tecnologías, Herramientas para el desarrollo del esquema de autenticación del Applet SDM.

Este capítulo contiene una base teórica para entender el problema planteado, en él se describen los conceptos fundamentales relacionados con tarjetas inteligentes, autenticación asimétrica en applets, interfaces compartidas entre los applets y firma digital.

Capítulo 2: Propuesta y Diseño del esquema de autenticación del Applet SDM.

Se presentarán los requerimientos funcionales y no funcionales del esquema de autenticación a utilizar.

Capítulo 3: Implementación y diseño de las pruebas del esquema de autenticación en el Applet SDM.

Se da cumplimiento a los planes trazados a través de las fases: Producción y Mantenimiento, se codifica la solución diseñada y finalmente se realizan las pruebas de aceptación con el cliente.

Capítulo 1: Fundamentación Teórica, Tecnologías, Herramientas para el Desarrollo del Esquema de autenticación del Applet Secure Data Manager.

1.1 Introducción.

El mayor auge de las tarjetas inteligentes fue en la década de los noventa, con la introducción de las tarjetas SIM utilizadas en la telefonía móvil GSM (Groupe Spécial Mobile) en Europa. El número de aplicaciones para SmartCards en general, va en un aumento constante, y abarca áreas muy diversas.

Algunos ejemplos típicos que se citan: es el ElectronicPurse o ElectronicWallet (ePurse y eWallet) esta aplicación se utiliza como dinero electrónico. Transacciones seguras ya sea a través de cajeros automáticos o de Internet. 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. Control de acceso y de asistencia a determinadas instalaciones. 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. El servicio puede variar desde telefonía celular hasta TV cable, pasando por acceso a sitios Web o transporte público.

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 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.

Los applets que gestionan la información, se ejecutan en el contexto del JavaCard Runtime – Environment (JCRE). Estos applets son los encargados de ejecutar las operaciones que

manejan, mediante los comandos APDU que le son enviados, retornando los resultados mediante APDU de respuestas.(thefreedictionary, 2011)

1.2.2 Applet Secure Data Manager.

Es una aplicación que se ejecuta dentro de la CIE de la República Bolivariana de Venezuela; para gestionar el acceso a la información, de los servicios que entidades venezolanas brindan a los ciudadanos de esa nación. (Almeida Sotolongo, 2009)

1.2.3 Middleware.

Es un software de conectividad que funciona como una capa de abstracción, situada entre la de aplicación y las inferiores (sistema operativo y red) haciendo posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. En general son usados para relacionar sistemas que necesitan intercambio de información, permitiendo realizar la conexión a través de interfaces de alto nivel. (thefreedictionary, 2011)

Algunas de sus funciones son: Transparencia de la heterogeneidad de los componentes de hardware, sistemas operativos y protocolos de comunicación. Proporciona un estándar de alto nivel de interfaces para los desarrolladores e integradores de aplicaciones, para que estas puedan ser fácilmente integradas, reutilizadas, adaptadas y hechas para inter-operar suministros de un conjunto de servicios comunes a diversas funciones de propósito general, a fin de evitar la duplicación de esfuerzos y facilitar la colaboración entre las aplicaciones. El papel del middleware es hacer más fácil el desarrollo de aplicaciones, ofreciendo abstracciones de programación común, mediante el enmascaramiento de la heterogeneidad, la distribución del hardware subyacente y sistemas operativos; además oculta los detalles de la programación de bajo nivel.

1.2.4 APDU.

El Application Protocol Data Unit (APDU) es la unidad de comunicación entre un lector de tarjetas inteligentes y una tarjeta inteligente. La estructura de un APDU está definida en los estándares ISO/IEC 7816**.** (ISO/IEC 7816, 2005.)

Hay dos tipos de APDUs: comandos y respuestas. Los comandos APDU los envía el lector a la tarjeta y contienen una cabecera obligatoria de 5 bytes“, y la longitud del APDU es de 0 a 255 bytes de datos. Las respuestas APDU las envía la tarjeta al lector y contienen una palabra de estado obligatoria de 2 bytes y la longitud del APDU respuesta es de 0 a 255 bytes de datos.

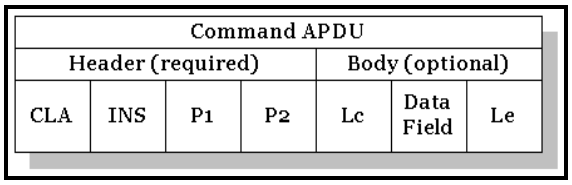

Figura 1: Estructura Comando APDU.

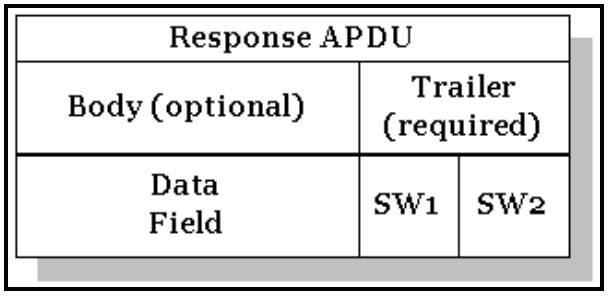

Figura 2: Estructura APDU respuesta

Comando APDU

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 byte	Número de bytes en el Data Field del APDU.
Le	1 byte	Cantidad máxima de bytes esperados como respuesta.

Tabla 1: Descripción del Comando APDU.

APDU Respuesta

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 2: Descripción del APDU Respuesta.

1.2.5 Cédula de Identidad Electrónica.

La Cédula de Identificación Electrónica 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. Tiene gran importancia

para responder a las necesidades que impone la creciente informatización del mundo moderno, concibe un nuevo documento de identificación, la cual está constituida por un chip que tiene alojado los applets que permiten la gestión de la información que se almacena.

1.3 Estándares relacionados con la Implementación del esquema de autenticación del Applet Secure Data Manager.

1.3.1 GlobalPlatform.

GlobalPlatform es una organización independiente enfocada a gestionar una infraestructura estandarizada para el desarrollo y despliegue de tarjetas inteligentes. Proporciona un conjunto de especificaciones universalmente reconocidas e implementadas, junto con configuraciones de mercado, aplicación de esas especificaciones y documentos de apoyo. Cubriendo toda la infraestructura de tarjetas inteligentes (las tarjetas, dispositivos y sistemas) estos documentos técnicos ofrecen una plataforma tecnológica dinámica y completa para el desarrollo de programas de tarjetas inteligentes, para poder establecer una conexión segura con la misma y administrar sus aplicaciones. (GlobalPlatform, 2010)

Las tarjetas, dispositivos y sistemas GlobalPlatform, son interoperables, independientemente de la tecnología del proveedor y la flexibilidad de su infraestructura técnica, garantizan que pueda responder a las necesidades básicas en el instante del despliegue inicial. Ofreciendo a los emisores la seguridad de que la infraestructura elegida será capaz de adaptarse y crecer a medida que cambian las condiciones de negocios.

1.3.2 PKCS

Se refiere a un grupo de estándares de criptografía de clave pública, concebidos y publicados por RSA Laboratories en California en cooperación con desarrolladores de sistemas de seguridad de todo el mundo, con el fin de acelerar el despliegue de la criptografía de clave pública. (RSA, 2000)

PKCS#11: Especifica una interfaz de programación para su uso con dispositivos criptográficos (tarjetas y HSM[1]). Define un API[2] genérico de acceso a dispositivos criptográficos.

[1]HSM son las siglas de Hardware Security Module (Módulo de Seguridad Hardware). HSM es un dispositivo criptográfico basado en hardware que genera, almacena y protege claves criptográficas y suele aportar aceleración hardware para operaciones criptográficas.

PKCS#15: Este PKCS surge para cubrir ciertos aspectos no contemplados por PKCS#11, dando uniformidad de estructura de directorios y ficheros con información criptográfica (por ejemplo las llaves simétricas y asimétricas, PIN, PUK y certificados digitales) en tarjetas inteligentes. Define un estándar que permite a los usuarios de dispositivo criptográficos identificarse con aplicaciones independientemente de la implementación del PKCS#11 u otro API

1.3.3 Estándar ISO/IEC- 7816.

La tarjeta inteligente más básica cumple los estándares de la serie ISO 7816. El objetivo de estos es lograr la interoperabilidad entre distintos fabricantes de tarjetas 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. Las características de comunicación de las tarjetas sin contacto son definidas en estándares como el ISO/IEC 14443. (ISO/IEC 7816, 2005.)

Descripción de algunas partes del estándar ISO 7816:

7816-1: Características físicas. 7816-2: Dimensiones y ubicaciones de los contactos. 7816-3: Señales electrónicas y protocolo de transmisión. 7816-4: Comandos de intercambio inter-industriales. 7816-5: Sistema de numeración y procedimientos de registro. 7816-6: Elementos de datos inter-industriales. 7816-7: Comandos inter-industriales y consultas estructuradas para una tarjeta. 7816-8: Comandos inter-industriales relacionados con seguridad. 7816-9: Comandos adicionales inter-industriales y atributos de seguridad. 7816-10: Señales electrónicas y respuesta para reiniciar una tarjeta inteligente síncrona.

1.3.4 Estándar PC/SC.

PC/SC (en inglés Personal Computer/Smart Card) es un conjunto de especificaciones para la integración de tarjetas inteligentes en ordenadores personales. En particular se define un API de programación, que permite a los desarrolladores trabajar de forma uniforme con lectores de tarjetas de distintos fabricantes (que cumplan con la especificación). (PC/SC, 2010)

El API de PC/SC está incorporada en sistemas Microsoft Windows 200x/XP y Microsoft Windows NT/9x. También hay una implementación libre, de código abierto, llamada PC/SC Lite (proyecto MUSCLE) para sistemas operativos GNU Linux.

---

[2]Una interfaz de programación de aplicaciones o API (Application Programming Interface) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción

La especificación se divide en 10 partes que contienen los requisitos detallados de interoperabilidad de dispositivos compatibles, información de diseño, interfaces de programación y otras.

Parte 1. Introducción y visión general de la arquitectura.

Parte 2. Requisitos de interoperabilidad para las tarjetas y los lectores.

Parte 3. Requisitos de interoperabilidad para los lectores conectados.

Parte 4. Consideraciones de diseño e información de referencia de los lectores.

Parte 5. Definición de la interfaz del Resource Manager.

Parte 6. Definición de la interfaz del Service Provider.

Parte 7. Consideraciones de diseño para el desarrollo de aplicaciones.

Parte 8. Recomendación para la implementación de servicios de seguridad y privacidad con tarjetas inteligentes.

Parte 9. Lectores con capacidades extendidas.

Parte 10. Lectores con capacidades de entrada de PIN de seguridad.

1.4 Análisis de otras soluciones

1.4.1 Tarjeta Criptográfica FNMT-RCM

Es el soporte físico empleado por el proyecto CERES (Certificación Española), que posibilita el uso del certificado de la FNMT-RCM para los servicios de la Administración Electrónica que los numerosos organismos oficiales están ofreciendo al ciudadano: Ministerios de Economía y Hacienda, Justicia, Administraciones Públicas, Defensa, Centro Nacional de Inteligencia, Guardia Civil, Consejo General del Notariado, Comisión Nacional de la Energía, Oficina Española de Patentes y Marcas, así como diversas Comunidades Autónomas, Ayuntamientos, Diputaciones, Colegios Profesionales, etc. Teniendo como características de seguridad la autenticación interna Tarjeta-Terminal, autenticación externa de usuario y de aplicación, validación de PIN de usuario, servicios de integridad mediante la generación y verificación de firmas digitales RSA. Generación de claves RSA en tarjeta, mecanismos de confidencialidad para el intercambio seguro de claves de cifrado. "Zona de espejo" para evitar pérdida de datos si la tarjeta es extraída durante una operación. (fnmt, 2010)

Soporta las técnicas criptográficas más avanzadas, como es el caso del algoritmo de cifrado simétrico Triple-DES, el algoritmo de cifrado asimétrico RSA con manejo de claves de 1.024 bits, y la generación de funciones unidireccionales hash mediante el algoritmo SHA-1. Permitiendo el almacenamiento y uso de claves de 1024 bits y con el componente ST19XL34

de hasta 2.176 bits. Generación y verificación de firmas digitales RSA. Cifrado y descifrado RSA. Generación de claves RSA. Cifrado y descifrado Triple DES. Cifrado hash SHA-1.

1.4.2 Implementación de Soluciones Biométricas (e-Passport) en Singapur.

Al considerarse a las fronteras como la primera línea de defensa contra amenazas a la seguridad exterior de los países, la Autoridad de Inmigración y Puestos de Control de Singapur, (ICA), requería una solución de seguridad de última generación que facilitará el traslado de los viajeros mediante la identificación biométrica basada en sus huellas dactilares y NEC, especialista con más de 30 años de experiencia en soluciones de identificación biométrica desarrolló e implementó una plataforma basada en pasaportes electrónicos apalancados con tecnología biométrica.(e-passport(singapur), 2010)

Esta tecnología futurista permite a los ciudadanos de Singapur hacer uso de sus pasaportes legibles por máquina para la limpieza automática, sin tener que solicitar una tarjeta inteligente adicional. El nuevo sistema avisa si un número de pasaporte o la huella dactilar no corresponde con los registrados, entonces los servicios de seguridad serán inmediatamente advertidos.

1.5 Tendencias Tecnológicas.

1.5.1 Tecnologías en Tarjetas Inteligentes.

1.5.1.1 SmartCard.

Es un dispositivo, que almacena y procesa información mediante un circuito de silicio embebido en el plástico de la tarjeta de acuerdo con el estándar ISO/IEC 7810 e ISO/IEC 7816 – 2.

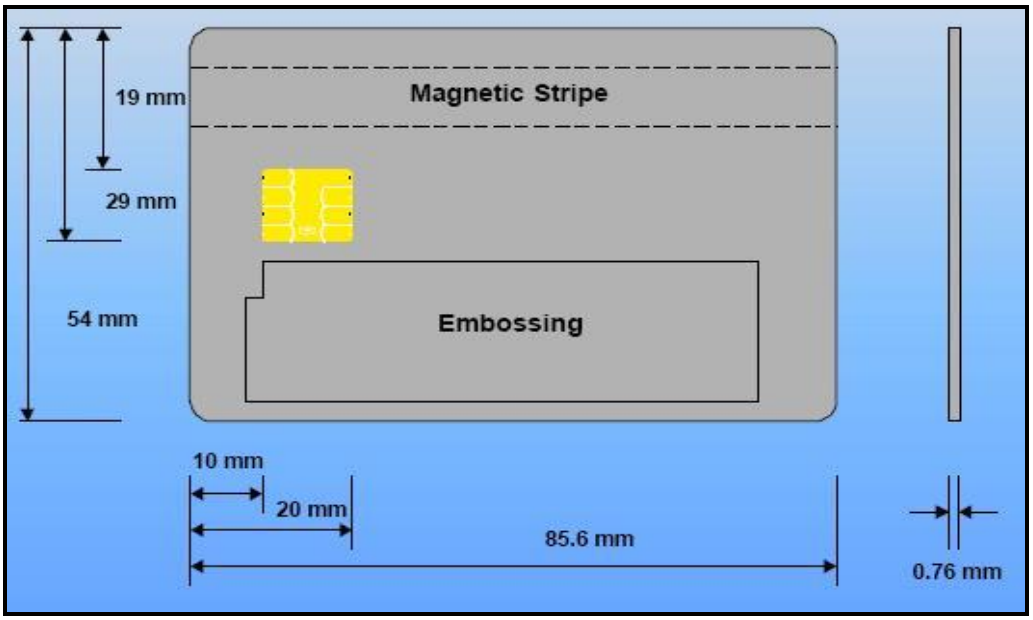

Figura 3. Características físicas de una SmartCard.

Una tarjeta inteligente contiene un microprocesador con su CPU, RAM y ROM, su forma de almacenamiento puede ser EPROM o EEPROM, 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 total mente inaccesible desde afuera del chip 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.

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

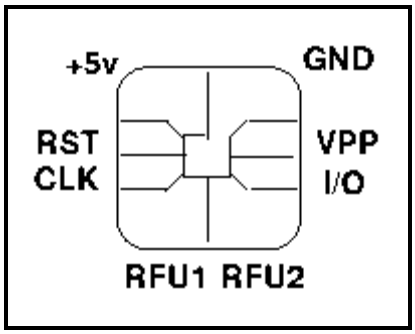

Figura 4: Interfaz de Comunicación de las SmartCard.

- +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 alguna 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, todo el protocolo de comunicaciones, dimensiones, resistencia, y 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 JavaCard. 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 RuntimeEnvironment (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 (Java Virtual Machine) 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 como 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 SmartCard 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 le 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 define 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 APDU6. 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 SmartCards, 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	Proprietary application identifier extension
RID	PIX
5 bytes	Entre 0 y 11 bytes

Tabla 3: 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 paquetes. 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 (Card Acceptance Device).

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 (aun 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.

1.5.1.2.1 JavaCard Runtime Environment (JCRE): shareable interfaces

El mecanismo de ObjectSharing 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 5). A una instancia de esta clase se la 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**.**

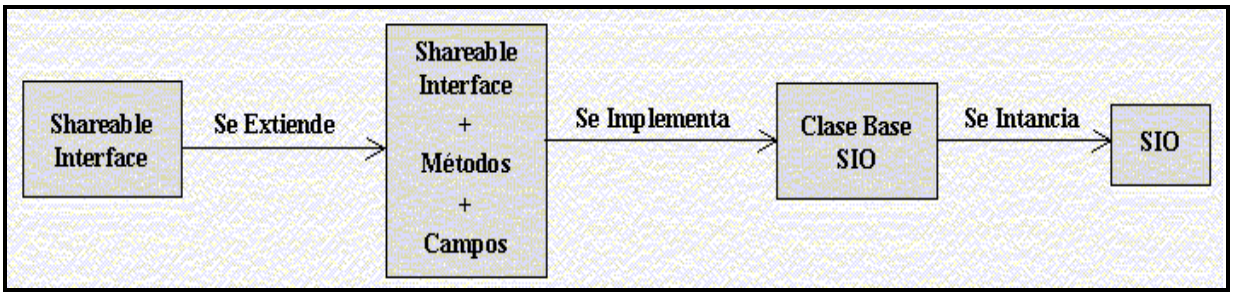

Figura 5: Interfaz Compartida.

El mecanismo de objectsharing 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 desarrolla las clases que las implementa. 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 6).

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.

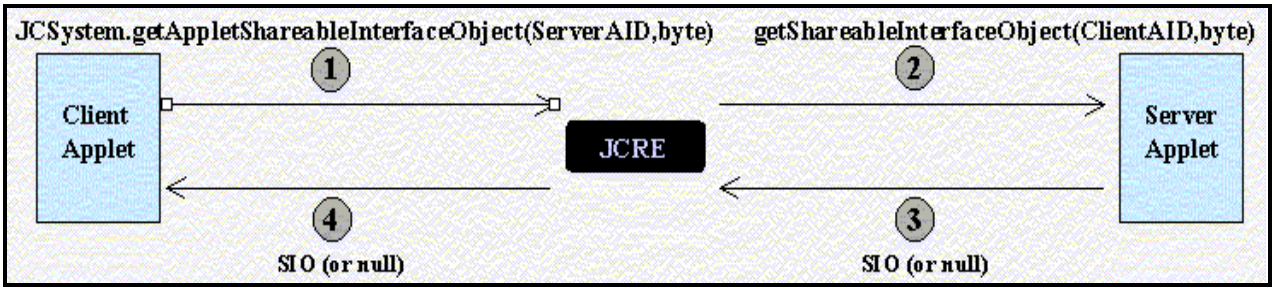

Figura 6: Método de la Interfaz Compartida.

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).

1.5.1.2.2 JavaCard RuntimeEnvironment (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 7).

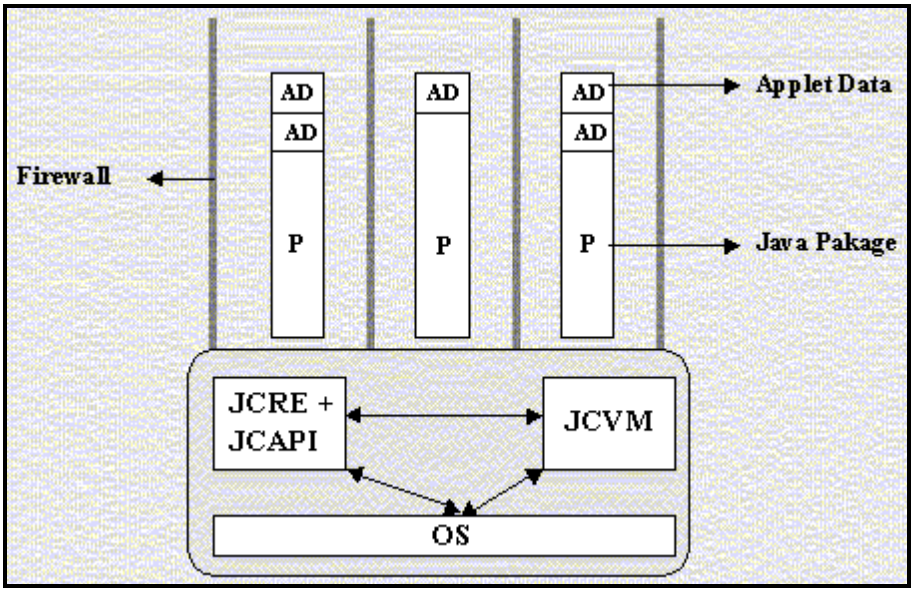

Figura 7: Descripción del Firewall.

1.5.1.3 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.). Esta tecnología se usa ampliamente en las tarjetas SIM (utilizadas en teléfonos móviles GSM) y en tarjetas monedero electrónico. (Java Card™ , 2005)

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).

1.5.1.4 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.(Sintaxis ASN.1, 2008)

1.5.1.5 Certificados X509.

En criptografía, X.509 es un estándar ITU-T para infraestructuras de claves públicas (PKI). X.509 especifica, entre otras cosas, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación. (Sintaxis ASN.1, 2008)

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 8), 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 el Applet SDM.

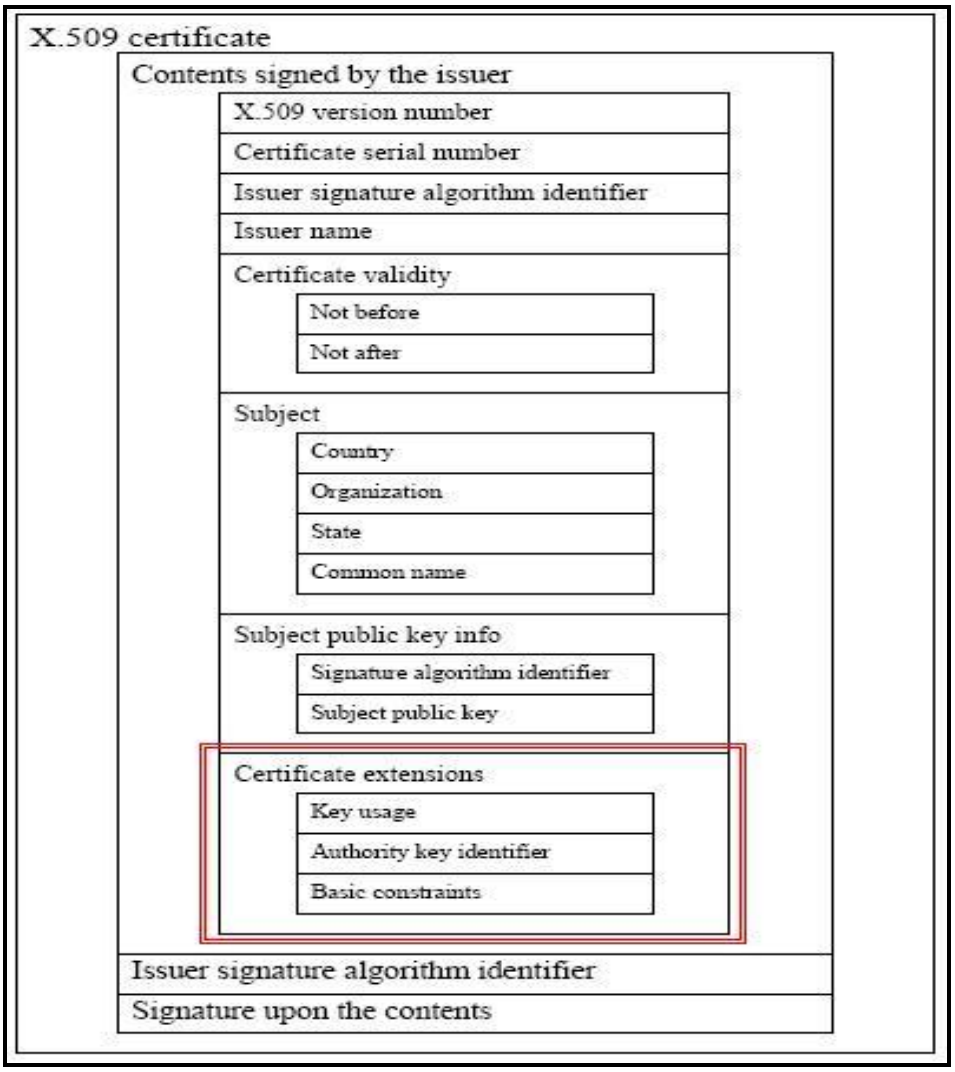

Figura 8: Estructura de Certificado versión 3.

1.5.2 Tecnología Biométrica. Match onCard.

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 interna que dependerá de la complejidad de la información biométrica a comparar (huellas, iris, facial, etc.) y de los algoritmos usados. (Precise Biometrics, 2005)

Match onCard (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

interna como la del microprocesador de una tarjeta inteligente, lo cual conlleva el uso de un sistema operativo que ejecute las aplicaciones de comparación necesarias.

1.5.3Tecnologías de Desarrollo.

1.5.3.1 Developer Suite Gemalto.

Herramienta que brinda un ambiente favorable para el diseño y la implementación de applets, además posibilita simular las funcionalidades de los applets antes de ser instalados en las tarjetas inteligentes**.** (Gemalto. Developer Suite., 2010)

1.5.3.2Metodologías de desarrollo.

Dentro de las metodologías de desarrollo existen dos grandes grupos, las conocidas metodologías tradicionales y las metodologías ágiles. Las primeras se centran en el uso absoluto de documentación durante todo el ciclo de vida del proyecto y es recomendada para los proyectos con grandes equipos de desarrollo. La metodología más utilizada dentro de este grupo es RUP, resultado de varios años de desarrollo y uso práctico en el que se han unificado varias técnicas de desarrollo. Mientras que las ágiles enfatizan las comunicaciones cara a cara con el cliente en vez de la documentación.

1.5.3.3 XP como metodología a utilizar.

Está diseñada para entregar el software que el cliente necesita, en breve periodo de tiempo. Además promueve el uso de prácticas para aumentar la productividad del equipo de desarrollo y mejorar la adaptabilidad a los frecuentes cambios dentro del ciclo de vida del proyecto.
Dentro de las ventajas que brinda la metodología XP podemos encontrar que es la más apropiada para entornos volátiles, equipos de desarrollo pequeños (de 2 a 10 desarrolladores) y proyectos de alto riesgo. También permite una mejor adaptabilidad a los cambios, que se traduce en una reducción de costos, la planificación es a corto plazo y es más transparente para los clientes, ya que conocen las fechas de entrega de funcionalidades vitales para su negocio y permite tener retroalimentación continua de los usuarios a través de las entregas frecuentes. Esta metodología también posee desventajas dentro de las que se encuentra la dificultad en muchas ocasiones para delimitar el alcance del proyecto con el cliente.

1.5.3.4 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; aun 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.

1.5.3.5 UModelAltova.

UModel se utiliza para crear e interpretar diseños software mediante la potencia del estándar UML 2.1. Se 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 de 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.

1.6 Plataforma .NET.

1.6.1 Microsoft .NET.

Es el conjunto de 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.

Agrupa las funcionalidades del sistema operativo en un espacio de nombres jerárquico 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.

1.6.2 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 la 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.

1.6.3 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 procesamiento muy grandes.

1.7 Propuesta y selección de herramientas.

El Applet SDM es una solución para gestionar información, referente a los servicios, que entidades administrativas les brindan a los ciudadanos de sus respectivas naciones que utilicen tarjetas inteligentes. El acceso a la información debe ser lo más seguro posible; para ello se va a desarrollar un esquema de autenticación para validar que la persona o institución sean las autorizadas a leer o escribir información en la CIE.

El esquema de autenticación se va a desarrollar utilizando tecnología JavaCard, la cual permite que se desarrollen aplicaciones (applets), que se ejecutan dentro de una tarjeta inteligente el cual tenga el mismo sistema operativo JavaCard, y componentes para estas aplicaciones.

Además, se utilizaran estándares, definidos internacionalmente, para el desarrollo con este tipo de técnica.

Por otra parte, se va a efectuar la implementación de una capa intermedia (Middleware), para permitir la comunicación con el esquema de autenticación. El desarrollo del mismo va a ser en C#, como parte de los lenguajes de .Net, el cual brinda una gran versatilidad frente a estándares establecidos para los lectores de tarjetas inteligentes PC/SC.

1.8 Conclusiones

- Durante el desarrollo del presente capítulo, se realizó un profundo estudio de los estándares relacionados con tarjetas inteligentes, seleccionando así los referentes a la implementación del esquema de autenticación para el Applet SDM.
- Se analizaron y seleccionaron las herramientas que permitan obtener una solución óptima para la implementación del esquema de autenticación.
- Se hizo un exhaustivo estudio sobre los algoritmos para la autenticación asimétrica en applets, como también la implementación y utilización de las interfaces compartidas en JavaCard.
- Se identificaron las metodologías de desarrollo, concluyendo que XP será la metodología que guiará los procesos de desarrollo del software, debido a sus características es la que mejor se adaptan al entorno de desarrollo.
- Por lo que en este capítulo se aborda ampliamente el porqué de la necesidad de la implementación del esquema de autenticación para el Applet SDM.

Capítulo 2: Propuesta y Diseño del Esquema de Autenticación del Applet Secure Data Manager.

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. 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 las historias de usuarios con sus respectivas descripciones, los requerimientos no funcionales del sistema, la arquitectura de la solución, plan de entrega, estimación de tiempo y el plan de iteraciones para la realización del esquema de autenticación del Applet SDM.

2.2 Propuesta de solución

2.2.1 Metáfora

Cada proyecto XP es guiado por una metáfora global. Estas ayudan a cualquier persona a entender el objetivo del programa, principalmente al equipo ya que aporta un contexto para entender los elementos básicos y sus relaciones proporcionando integridad conceptual. Es de vital importancia que el cliente y los desarrolladores estén de acuerdo y conozcan la metáfora a usar para así poder trabajar y discutir en los mismos términos de una forma más precisa y de dominio de todos.

El Applet SDM es una solución para gestionar información, referente a los servicios, que entidades administrativas les brindan a los ciudadanos de sus respectivas naciones que utilicen tarjetas inteligentes. El acceso a la información debe ser lo más seguro posible; para ello se va a desarrollar un esquema de autenticación para validar que la persona o institución sean las autorizadas a leer o escribir información en la CIE.

El Applet SDM a través del Esquema de Autenticación permitirá comprobar que el portador de la tarjeta sea el usuario de la misma a través de la verificación del PIN y de la comparación de

la huella dactilar, comparando la huella almacenada con la huella obtenida por el lector, lo cual le otorgará mayor confiabilidad al proceso disminuyendo considerablemente las posibilidades de falsificación. El Esquema de Autenticación también verificará la integridad del certificado digital que la autoridad certificadora emite a los centros o instituciones que tendrán acceso a la información de la CIE. Este proceso es de vital importancia ya que la autenticación establece el control de niveles de acceso a la gestión de la información de la CIE, estableciendo cuales son las instituciones con permiso de lectura-escritura de la información y cuáles son las que solo tienen permiso de lectura, así como se determina a que información dentro de la tarjeta se va a poder acceder.

2.2.2 Modelo de dominio.

Teniendo en cuenta que no está determinado el problema científico que el esquema de autenticación pretende resolver y debido a la ausencia de procesos bien definidos que permitan modelar el funcionamiento del mismo, se ha procedido a crear un modelo de domino. El mismo permitirá identificar los principales conceptos sociales y tecnológicos del entorno en el cual se desarrolla el esquema de autenticación, así como las relaciones entre ellos.

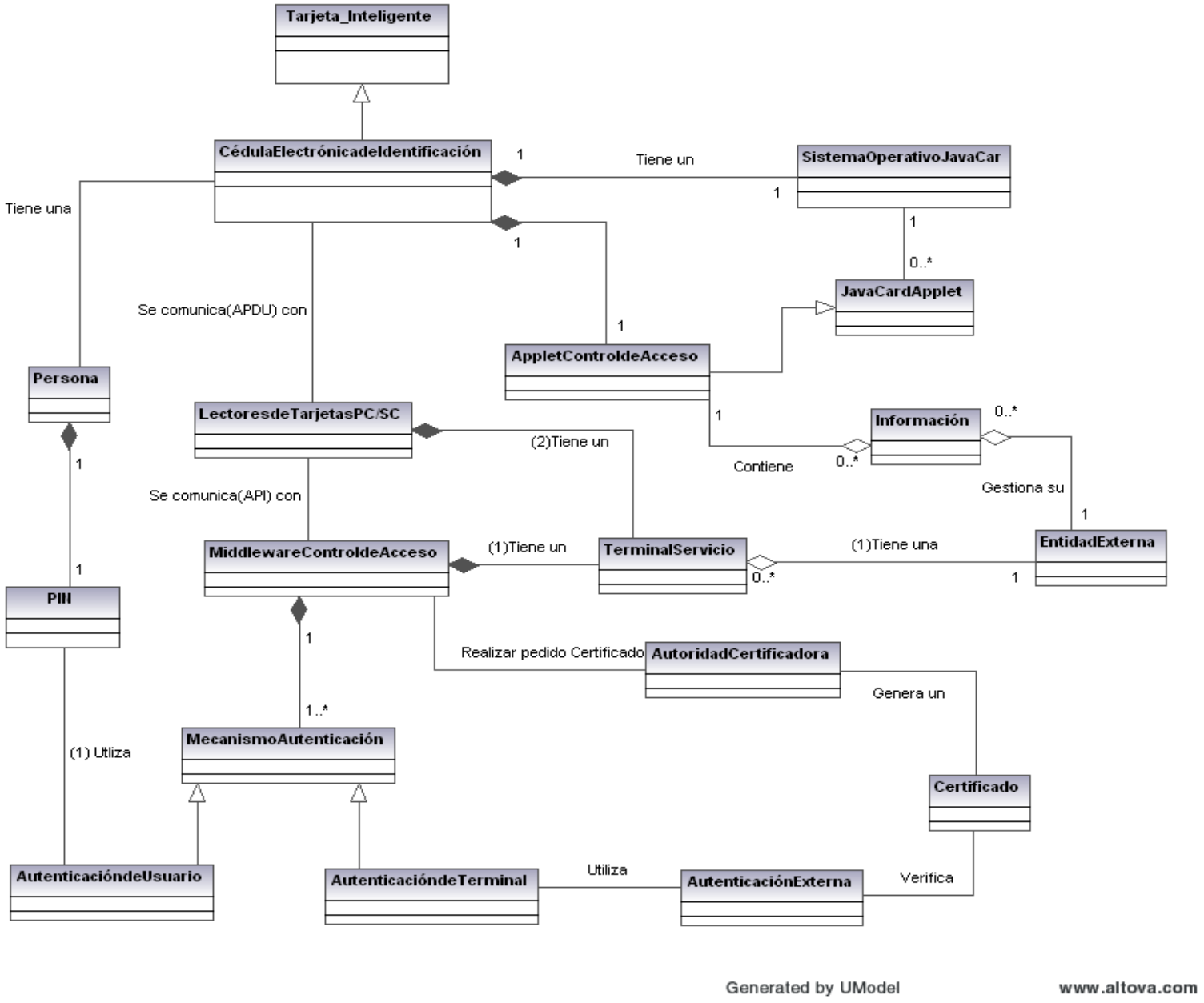

Figura 9: Diagrama de dominio de la solución.

2.2.3 Glosario de conceptos del modelo de dominio.

Tarjeta Inteligente: Es un dispositivo 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.

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 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.

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 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 la CIE (applet).

Entidad Externa: Es cualquier entidad, 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, 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 inglés) garantiza la vinculación entre la identidad del portador de la Cédula de Identificación

Electrónica 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.

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. **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.

2.2.4 Historias de usuario.

Las historias de usuario son utilizadas en XP para especificar los requisitos del *software* desde el punto de vista del cliente. Estas se caracterizan por establecer la descripción de los requisitos funcionales del cliente, describir pequeños trozos de funcionalidad que aportan valor al desarrollo de la aplicación, son asignadas a la persona o desarrollador encargado de la programación con un número de horas de desarrollo estimado, establecen la prioridad en el desarrollo del sistema de la funcionalidad descrita, son descritas entre el cliente y el analista, las historias de usuario guían la construcción de los test de aceptación (casos de prueba).

	Historia de usuario
Número: HU_1	**Nombre de Historia de Usuario:** Inicializar comunicación con la tarjeta.
Modificación de Historia de Usuario Número: Ninguna	
Usuario:	**Iteración asignada:** 1
Prioridad en negocio: Media	**Puntos estimados:** 2
Riesgo en desarrollo: Media	**Puntos reales:** 2
Descripción: Se reconoce los lectores que están disponibles, se selecciona uno con el cual se va a establecer la comunicación con la CIE.	
Observaciones: Se muestra al usuario se a inicializado la comunicación con el Applet de la CIE.	

Tabla 4: HU_1Inicializar comunicación con la tarjeta.

	Historia de usuario
Número: HU_2	**Nombre de Historia de Usuario:** Finalizar comunicación con la tarjeta.
Modificación de Historia de Usuario Número: Ninguna	
Usuario:	**Iteración asignada:** 1
Prioridad en negocio: Alto	**Puntos estimados:**1
Riesgo en desarrollo: Alto	**Puntos reales:**1
Descripción: Consiste en realizar la desconexión entre la CIE y el lector.	
Observaciones: Se muestra al usuario el fin de la comunicación con el Applet de la CIE.	

Tabla 5: HU_2 Finalizar comunicación con la tarjeta.

	Historia de usuario
Número: HU_3	**Nombre de Historia de Usuario:** Establecer Canal Seguro.

Modificación de Historia de Usuario Número:	
Usuario:	**Iteración asignada:** 1
Prioridad en negocio: Alto	**Puntos estimados:** 2
Riesgo en desarrollo: Alto	**Puntos reales:** 2
Descripción: 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.	
Observaciones: Se muestra al usuario el fin de la comunicación con el Applet de la CIE.	

Tabla 6: HU_3 Establecer Canal Seguro.

	Historia de usuario
Número: HU_4	**Nombre de Historia de Usuario:** Verificar la autenticación del usuario por PIN.
Modificación de Historia de Usuario Número: Ninguna	
Usuario:	**Iteración asignada:** 1
Prioridad en negocio: Media	**Puntos estimados:** 1
Riesgo en desarrollo: Media	**Puntos reales:** 1
Descripción: Consiste en realizar la autenticación del Usuario portador de la CIE, introduciendo este su número de PIN para validar la autenticidad de posesión de la CIE.	
Observaciones: Siempre se le mostrará al usuario el resultado de la operación en curso.	

Tabla 7: HU_4 Verificar la autenticación del usuario por PIN.

Historia de usuario	
Número: HU_5	**Nombre de Historia de Usuario:** Obtener la respuesta de la verificación de la huella dactilar.
Modificación de Historia de Usuario Número: Ninguna	
Usuario:	**Iteración asignada:** 2
Prioridad en negocio: Alto	**Puntos estimados:** 2
Riesgo en desarrollo: Alto	**Puntos reales:**2
Descripción: Obtener el resultado de la comparación de la huella dactilar a través de la interfaz compartida que brinda el Applet del CryptoManager.	
Observaciones: Siempre se le mostrará al usuario el resultado de la operación en curso.	

Tabla 8: HU_5 Obtener la respuesta de la verificación de la huella dactilar.

Historia de usuario	
Número: HU_6	**Nombre de Historia de Usuario:** Verificar el token de acceso.
Modificación de Historia de Usuario Número: Ninguna	
Usuario:	**Iteración asignada:** 2
Prioridad en negocio: Alto	**Puntos estimados:**3
Riesgo en desarrollo: Alto	**Puntos reales:** 3
Descripción: El Middleware de Control de Acceso envía el token de acceso al Applet para su verificación en un TLV. Dándole acceso a la información de acuerdo con los permisos que la entidad externa posee. Se obtiene los datos enviados por el middleware y se verifica la firma.	
Observaciones: Siempre se le mostrará al usuario el resultado de la operación en curso.	

Tabla 9: HU_6 Verificar el certificado del token de acceso.

2.3 Requerimientos no funcionales.

Requerimientos del hardware:

- Lector de tarjetas incorporado a la PC que cumpla con el estándar PC/SC versión 1.0 o superior.
- Escáner de huella.

Requerimientos del software:

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.

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).

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 siglas en ingles), según se define en el estándar GlobalPlatform.

2.4 Arquitectura.

2.4.1 Arquitectura del middleware.

Para la confección de la arquitectura se pensó en el estilo Arquitectura en Capas a continuación se muestra dividido en dos partes. En el diagrama de componentes se muestra la arquitectura del middleware, relacionando componentes y paquetes. En la solución se implementó la DLL <<SecureDataManager.dll>>, el paquete <<SmartCard.Net Framework>> y el paquete <<WinPCSCWrapper>>.

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 wrapperWinPCSC.

El WinPCSCWrapper 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.

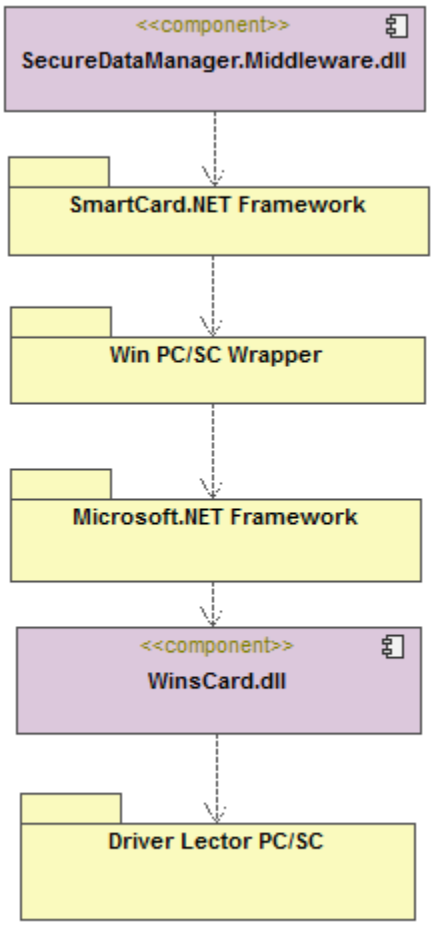

Figura 10: Arquitectura del middleware.

2.4.2 Arquitectura del Applet.

El SDM es un applet que actúa como cliente el cual se comunica con el CryptoManager a través de una API interna (interfaz compartida) siendo este último el applet servidor. El Applet SDM contendrá como componente el esquema de autenticación el cual verificara el valor del PIN, obtendrá el resultado de la verificación de la huella dactilar a través de la interfaz compartida del Applet CryptoManager y validará el token de acceso para las entidades externas que vayan hacer uso de la información almacenadas en el Applet SDM.

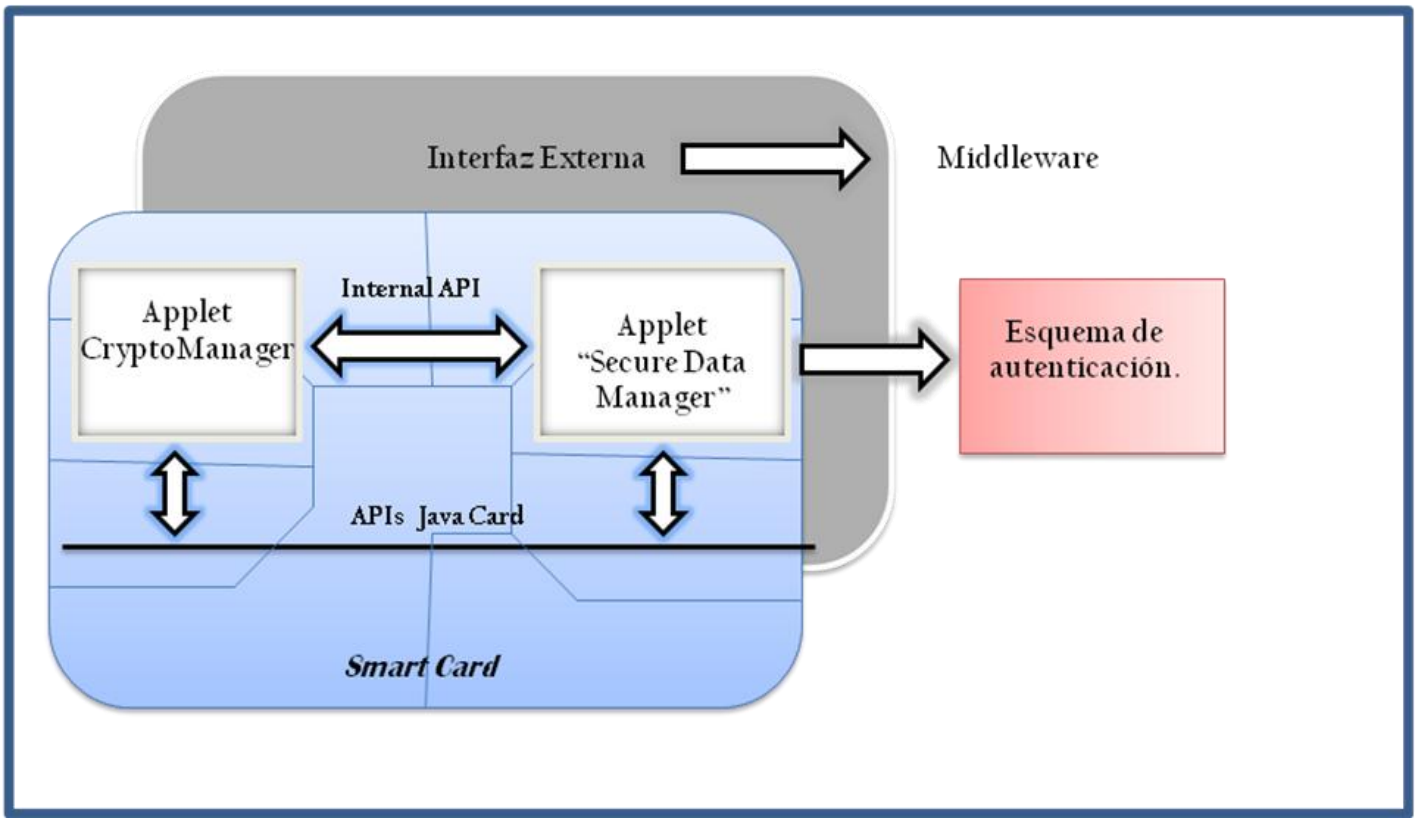

Figura 11: Arquitectura del Applet SDM.

2.5 Patrones de diseño.

Se utilizaron para el diseño los patrones GRASP los cuales describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable.

Creador: Permite decidir cuáles serán las clases creadoras de otras clases. La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia, es útil contar con un principio general para la asignación de las responsabilidades de creación. Este patrón se utiliza en las clases entidades, cuando estas se instancian y crean instancias de clases contenidas en las mismas, ejemplo: Configurar.cs.

Alta cohesión: Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable. Una clase cohesionada facilita el cambio. Al realizar un cambio en una clase muy cohesionada, todos los métodos que pueden verse afectados, toda la información que necesitamos controlar, estará a la vista, en el mismo fichero. Se pone de manifiesto: Certificado.cs.

Bajo acoplamiento: Uno de los principales síntomas de un mal diseño y alto acoplamiento es una herencia muy profunda. Debe haber pocas dependencias entre las clases. Uno de los principios para proteger al software frente al cambio es mantener bajo el acoplamiento entre clases. Ejemplo, (ComandoEnviarToken.cs -> APDUCommand.cs).

2.6 Plan de entrega.

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. (Ver Anexo 1)

2.7 Estimación de tiempo.

Los programadores estiman el tiempo que necesitan para desarrollar cada historia de usuario. El valor del tiempo se expresa en semanas. (Ver Anexo 2)

2.8 Plan de iteraciones.

Como parte del ciclo de vida de un proyecto usando la metodología XP se crea el plan de duración de cada una de las iteraciones que se han definido, que tiene como objetivo mostrar la duración y el orden en que serán implementadas las historias de usuario dentro de cada iteración. Para la solución se han definido 6 historias de usuario divididas en 2 iteraciones, de acuerdo a los intereses del cliente, para una duración total del proyecto de 10 semanas. (Ver Anexo 3)

2.9 Conclusiones.

- En el capítulo 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 definieron los requerimientos no funcionales que brindarán las cualidades que se deben de tener en cuenta para desarrollar una solución adecuada.
- Se define la arquitectura con la cual será implementada el middleware y el applet del esquema de autenticación para el Applet SDM.
- Se precisaron el plan de entrega, la estimación del tiempo y el plan de iteraciones.

Capítulo 3: Implementación y Prueba del Esquema de Autenticación del Applet Secure Data Manager.

3.1 Introducción

Seguido de la fase de Exploración y Planificación, XP define las fases Iteraciones a primera liberación y Producción. En la Planificación se definieron las iteraciones y en cada iteración se diseñan, prueban y codifican cada una de las historias de usuario.

Con la elaboración de este capítulo se propone explicar el diseño de la solución, los principales componentes y las relaciones que existen entre ellos. Mostrándolo a través de diferentes diagramas que ofrece una vista de los principales flujos de procesos.

3.2 Iteraciones a primera liberación.

En esta fase ocurren varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. Al principio de cada iteración se realizan las tareas necesarias de análisis, arreglando con el cliente todos los datos que sean necesarios. Al final de la última iteración el sistema estará listo para entrar en producción. En esta fase es donde se da cumplimiento al Plan de Iteración, los elementos que se deben tener en cuenta para la confección de este artefacto son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

3.2.1 Tareas de ingeniería

Para comenzar a codificar la solución es necesario saber qué codificar. Las historias de usuarios no ofrecen el nivel de detalles requerido para llevar a cabo la solución, es por eso que son divididas en tareas de ingeniería. Una historia de usuario generalmente se divide en más de una tarea de ingeniería y a partir de estas tareas comienza el ciclo de la fase de Iteraciones a primera liberación. Según el Plan de iteraciones las historias de usuario se agruparon en dos iteraciones. A continuación se muestran las tareas de ingeniería derivadas de cada historia de usuario.

Iteración	Historias de Usuarios	Tareas
1	Inicializar comunicación con la tarjeta.	-Listar lectores conectados. -Establecer una conexión con la tarjeta a través del lector escogido.
	Finalizar comunicación con la tarjeta.	- Finalizar comunicación con la tarjeta.
	Establecer el canal seguro.	- Establecer un canal de intercambio de información de forma segura entre el Middleware y el Applet de Control de Acceso. - Utilizar Protocolo de Canal Seguro “01”.
2	Verificar la autenticación del usuario por PIN.	-Enviar el Número de Identificación Personal a través de un comando APDU para su verificación. - Verificar la autenticación del usuario por PIN en el applet. -Notificar al usuario el resultado de la operación a través del comando respuesta.
	Verificar la autenticación del usuario por Mach onCard.	- Obtener el resultado de la verificación biométrica a través de la interfaz compartida que brinda el CryptoManager. -Notificar al usuario el resultado de la operación a través del comando respuesta.
	Verificar el token de acceso.	-Enviar el token de acceso a través del comando APDU. -Verificar el token de acceso en el applet. -Notificar al usuario el resultado de la operación a través del

		comando respuesta.

Tabla 10: Distribución de las tareas de ingeniería por iteraciones.

3.3 Diseño de la solución:

Una vez desglosadas las historias de usuario en tareas, se comienzan a generar ideas de cómo ejecutarlas. Estas ideas se unifican en las sesiones de diseño previas a cada iteración. En estas sesiones, XP propone la puesta en práctica de ciertos principios, descritos a continuación, para garantizar la agilidad en el proceso de desarrollo.
Según Wells, J. Donovan (Wells, 1999) estos principios son:
Simplicidad: Un diseño simple siempre se termina más rápido y es más fácil de entender que uno complejo.

Uso de tarjetas CRC: (clase, responsabilidad, colaboración), estas tarjetas son manejadas por el equipo de desarrollo durante la codificación de la solución; generalmente cada tarjeta representa una clase diferente en la codificación y tienen como ventaja que todo el equipo contribuye a la elaboración del diseño de la solución.

No adicionar funcionalidades tempranamente: Mantener el sistema lo más separado de las funcionalidades extras que no sean imprescindibles. Solo el 10% de las funcionalidades extras son utilizadas y hacen perder el 90% del tiempo.

MiddlewareSDM	
Responsabilidades	**Colaboradores**
- Listar lectores conectados. - Establecer comunicación con la tarjeta. - Seleccionar Applet del esquema de autenticación SDM. - Configurar el canal seguro por GlobalPlatform para el SDM. - Establecer el canal seguro con el applet SDM. - Permitir introducir el PIN.	- SmartCard.Client - SmartCard.Core - SmartCard.Devices - SmartCard.Devices.CardReaders.PCSLite - SmartCard.Devices.CardReaders.Win32PCSCWrapper - SmartCard.GlobalPlatform - SmartCard.ISO7816 - SmartCard.Security - SmartCard.VenezuelaIDCardManager

- Verificar el PIN en el esquema de autenticación del Applet SDM. - Enviar la huella dactilar al Applet CryptoManager para su verificación. - Obtener certificado del token de Acceso. - Obtener el token de Acceso y enviarlo al Applet SDM.	- ComandoAlmacenarHuella - ComandoVerificarPIN - ComandoVerificarHuella - ComandoVerificarCertificado

Tabla 11: Tarjeta CRC: MiddlewareSDM.

Applet SDM	
Responsabilidades	**Colaboradores**
- Establecer el canal seguro con el middleware. - Permitir obtener el PIN. - Verificar el PIN. - Obtener el resultado de la comparación de la huella dactilar. - Obtener los datos del token de Acceso. - Verificar la firma del token de Acceso.	- No tiene.

Tabla 12: Tarjeta CRC: Applet SDM.

3.3.1 Descripción del flujo de proceso: esquema de autenticación.

El proceso fundamental que ejecuta el esquema de autenticación es la confirmación del portador de la CIE y la validez del certificado digital, mediante la verificación del token de Acceso.

A continuación una breve descripción:

- Con la tarjeta ya conectada el usuario introduce el PIN.
- Después que el PIN es capturado es enviado a la tarjeta y verificado por el Esquema de Autenticación.
- El usuario coloca su dedo en el escáner de huellas, se obtiene la huella y se envían las minucias al Applet CryptoManager.
- El Esquema de Autenticación, mediante la interfaz compartida, obtiene la respuesta de la verificación de la huella.
- El Middleware SDM obtiene y envía el token de Acceso al Esquema de Autenticación.

- Seguidamente el Esquema de Autenticación obtiene los datos firmados y verifica la firma.
- Una vez realizadas todas estas operaciones, si su ejecución dio los resultados esperados, se procede a escribir o leer de un fichero determinado dentro de la tarjeta inteligente.

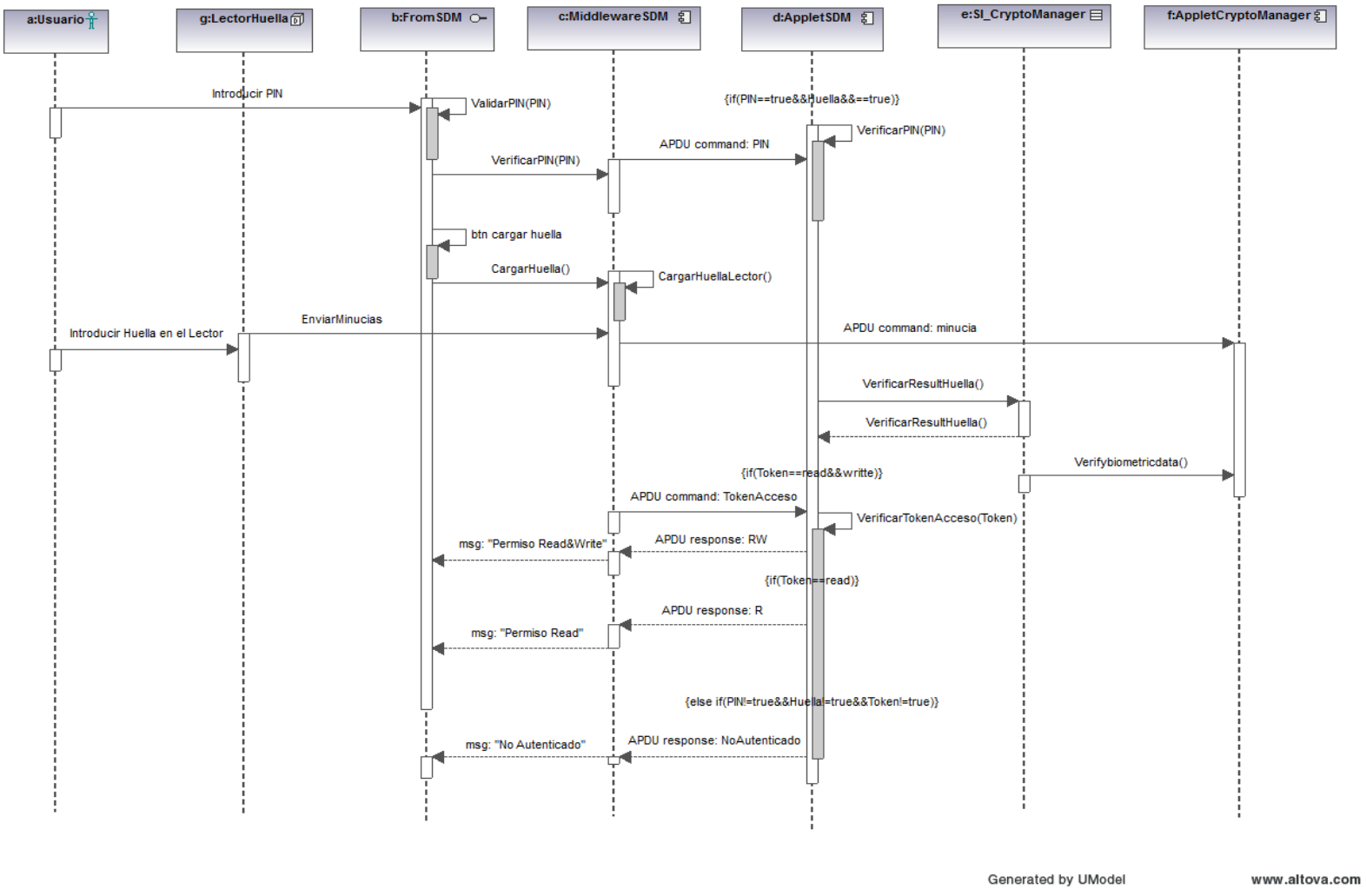

Figura 12: Diagrama de secuencia del esquema de autenticación.

3.3.2 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. (Ver Tabla 12)

Nombre del comando.	Composición de los comandos.					Descripción
	CLA	INS	P1	P2	Data	
	APDU Estándares de GlobalPlatform					
Initialize Update	80	50	00	00	-	Comando que inicializa el proceso para el canal seguro para la comunicación entre el Applet SDM, 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 Applet SDM y determina el nivel de seguridad requerido por todos los comandos que le siguen.
	APDU Propios					
Verificar PIN.	90	10			PIN	Comando para verificar el PIN.
Cambiar PIN.	90	12			PIN	Comando para cambiar el PIN.
Enviar Minucias.	90	13			Minucias	Comando para enviar las minucias de la huella.
Verificar resultado de la comparación de la huella.	90	14				Comando para saber el resultado de la comparación de la huella.
Enviar token de Acceso	90	16			token	Comando para enviar el token de Acceso.
Verificar token	90	18				Comando para Verificar el token de Acceso.

Tabla 13: Descripción de los Comandos APDU.

Si el comando respuesta es 90 00 significa que el comando fue procesado, de lo contrario ocurrió algún error. (Ver tabla 13)

Data	SW1	SW2
	90	00

Tabla 14: Descripción del comando respuesta.

3.3.3 Descripción del resultado de la comparación de la huella.

El resultado de la comparación de la huella se obtiene a través de una interfaz compartida que brinda el applet de biometría, siendo el CryptoManager el applet de biometría que se utilizó para la implementación del esquema de autenticación. El applet del CryptoManager implementa una interfaz compartida donde se tiene acceso al método GET PROPERTIES el cual devuelve un arreglo de byte de 25 posiciones. A continuación se muestra una tabla donde se ve las propiedades de cada valor en el arreglo.

Data Type Description	**Data Type Description**
BYTE Applet major version	BYTE Applet major version
BYTE Applet minor version	BYTE Applet minor version
BYTE Algorithm major version	BYTE Algorithm major version
BYTE Algorithm minor version	BYTE Algorithm minor version
BYTE Current life cycle state	BYTE Current life cycle state
BYTE AdminKey retries	BYTE AdminKey retries
BYTE StartKey retries	BYTE StartKey retries
BYTE Currently authenticated keys	BYTE Currently authenticated keys
BYTE Number of template containers	BYTE Number of template containers
BYTE[10] Template container #01--#10 retries	BYTE[10] Template container #01--#10 retries
SHORT Currently authenticated template containers	SHORT Currently authenticated template containers
SHORT Currently locked template containers	SHORT Currently locked template containers
SHORT Currently used template containers	SHORT Currently used template containers

Tabla 15: Descripción del método GetProperties.

3.3.4 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 pública 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.

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.
5 + X	53h	Cabecera de la llave pública.
6 + X	L	Largo de la llave pública.
7 + X	54h	Cabecera del Módulo de la llave pública.
8 + X	L	Largo del Módulo de la llave pública.
9+ X	Módulo	Valor del Módulo de la llave pública.
9 + X	55h	Cabecera del exponente de la llave pública.
10 + X	L	Largo del exponente de la llave pública.
11 + X	Exponente	Valor del Exponente de la llave pública.
11 + X	56h	Cabecera de la firma de los datos
12 + X	L	Largo de la firma de los datos
13 + X	Firma	Valor de la firma de los datos

Tabla 16: Descripción del token de Acceso.

La posición donde se encuentra el valor se calcula sumando la posición que aparece en la tabla más la X en el caso donde aparece. Significando la X, el valor de la suma de las longitudes anteriores al valor que desea buscar.

3.3 Fase de Producción

El equipo de desarrollo trabajó solo en los pedidos del cliente, refinando el diseño y el código continuamente para lograr un sistema limpio y capaz de evolucionar. El perfeccionamiento de código sólo fue posible a través de un grupo de pruebas unitarias automatizadas que aseguraron la ejecución correcta del sistema en todo el período de desarrollo.
La metodología XP divide las pruebas del sistema en dos grupos: pruebas unitarias y pruebas de aceptación, la primera es la encargada de verificar el código y diseñada por los programadores y la última es las destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

3.3.1Pruebas Unitarias

Las pruebas unitarias son una herramienta muy útil en el desarrollo y diseño del *software* ya que ayudan a garantizar que el programa hace justo lo que se especifica en los códigos de pruebas que lo definen. Herramientas de pruebas unitarias están implementadas en casi

cualquier lenguaje de programación. Los códigos de pruebas son útiles en cualquier momento del desarrollo del software aunque se cambie la implementación, siempre que se mantenga la interfaz.

Las pruebas de unidad se realizaron en el middleware con Visual Studio 2010 y en el applet con el JCardManager. Se construyó un caso de prueba para cada método de la clase especificada, con soporte de prueba para los métodos críticos de cada clase. De esta manera, se logró la disminución del tiempo en el ciclo compilación-ejecución, permitiendo al equipo de desarrollo concentrarse en la codificación de la solución. (Ver Anexo 4 y 5).

3.3.2 Pruebas de aceptación

Las pruebas del cliente también conocidas como pruebas de aceptación son definidas por este y su objetivo es probar que las funcionalidades del sistema corresponden con las historias de usuarios definidas por él. Una historia de usuario puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento y no se considera completa hasta que no supera sus pruebas de aceptación. Es responsabilidad del cliente verificar la corrección de las pruebas y tomar decisiones acerca de las mismas.

Caso de prueba de aceptación	
Código de caso de prueba: HU1_CP1	**Nombre de la historia de usuario:** Inicializar comunicación con la tarjeta.
Responsable de la prueba: Abraham Armas Pérez	
Descripción de la prueba: Prueba de funcionalidad para inicializar comunicación con la tarjeta.	
Condiciones de ejecución: Debe estar la tarjeta conectada al lector.	
Entrada/Pasos de ejecución: • Se lista todos los lectores los cuales están conectados con la PC. • Una vez escogido el lector se establece comunicación con la tarjeta.	
Resultado esperado: Se establece satisfactoriamente la comunicación con la tarjeta.	
Evaluación de la prueba: Prueba Satisfactoria.	

Tabla 17: HU1_CP1 Inicializar comunicación con la tarjeta.

Caso de prueba de aceptación

Código de caso de prueba: HU2_CP2	**Nombre de la historia de usuario:** Finalizar comunicación con la tarjeta.
Responsable de la prueba: Abraham Armas Pérez	
Descripción de la prueba: Prueba de funcionalidad para finalizar comunicación con la tarjeta.	
Condiciones de ejecución: Debe estar la tarjeta conectada al lector.	
Entrada/Pasos de ejecución: • Se finaliza la comunicación con la tarjeta.	
Resultado esperado: Se finaliza satisfactoriamente la comunicación con la tarjeta.	
Evaluación de la prueba: Prueba Satisfactoria.	

Tabla 18: HU2_CP2Finalizar comunicación con la tarjeta.

	Caso de prueba de aceptación
Código de caso de prueba: HU3_CP3	**Nombre de la historia de usuario:** Establecer el canal seguro.
Responsable de la prueba: Abraham Armas Pérez	
Descripción de la prueba: Prueba de funcionalidad para establecer el canal seguro.	
Condiciones de ejecución: Debe estar la tarjeta conectada al lector.	
Entrada/Pasos de ejecución: • Realizar operaciones en la aplicación.	
Resultado esperado: Se establece satisfactoriamente el canal seguro.	
Evaluación de la prueba: Prueba Satisfactoria.	

Tabla 19: HU3_CP3 Establecer el canal seguro.

	Caso de prueba de aceptación
Código de caso de prueba: HU4_CP4	**Nombre de la historia de usuario:** Verificar la autenticación del usuario por PIN.
Responsable de la prueba: Abraham Armas Pérez	
Descripción de la prueba: Prueba de funcionalidad para verificar la autenticación del usuario por PIN.	
Condiciones de ejecución: Debe estar la tarjeta conectada al lector.	

Entrada/Pasos de ejecución: • El usuario introduce el PIN • Se envía el PIN a la tarjeta. • El esquema de autenticación verifica el PIN. • Se notifica el resultado de la operación.
Resultado esperado: Se verifica satisfactoriamente la autenticación del usuario por PIN.
Evaluación de la prueba: Prueba Satisfactoria.

Tabla 20: HU4_CP4Verificar la autenticación del usuario por PIN.

	Caso de prueba de aceptación
Código de caso de prueba: HU5_CP5	**Nombre de la historia de usuario:** Verificar la autenticación del usuario por Mach onCard.
Responsable de la prueba: Abraham Armas Pérez	
Descripción de la prueba: Prueba de funcionalidad para verificar la autenticación del usuario por Mach onCard.	
Condiciones de ejecución: Debe estar la tarjeta conectada al lector y debe haber macheado con el lector de huella.	
Entrada/Pasos de ejecución: • El Esquema de Autenticación, mediante la interfaz compartida, obtiene el resultado de la comparación de la huella dactilar en el Applet CryptoManager. • Se notifica el resultado de la operación.	
Resultado esperado: Se verifica satisfactoriamente la autenticación del usuario por Mach onCard.	
Evaluación de la prueba: Prueba Satisfactoria.	

Tabla 21: HU5_CP5 Verificar la autenticación del usuario por Mach onCard.

	Caso de prueba de aceptación
Código de caso de prueba: HU6_CP6	**Nombre de la historia de usuario:** Verificar el token de acceso.
Responsable de la prueba: Abraham Armas Pérez	
Descripción de la prueba: Prueba de funcionalidad para verificar el certificado del token de acceso.	
Condiciones de ejecución: Debe estar la tarjeta conectada al lector y debe haber macheado con el	

lector de huella.
Entrada/Pasos de ejecución: • Se envía el token de acceso al esquema de autenticación. • El esquema de autenticación obtiene los datos firmados y verifica la firma del mismo. • Se notifica el resultado de la operación.
Resultado esperado: Se verifica satisfactoriamente el certificado del token de acceso.
Evaluación de la prueba: Prueba Satisfactoria.

Tabla 22: HU6_CP6Verificar el certificado del toquen de acceso.

3.4 Conclusiones

Luego de terminadas las fases de Iteraciones a primera liberación y Producción, se concluye que:

- El desglose de las historias de usuario en tareas de la ingeniería fue una buena práctica que mostró a los programadores las funcionalidades específicas a implementar.
- El desarrollo guiado por pruebas aseguró la ejecución correcta de la solución en todo el período de implementación, disminuyendo el tiempo invertido en el ciclo compilación-ejecución.
- Las pruebas de aceptación concluyeron de manera exitosa demostrando la satisfacción del cliente con la solución.

Conclusiones

Con el desarrollo de este proyecto se logró cumplir con todos los objetivos, tanto generales como específicos, trazados para la implementación del esquema de autenticación del Applet SDM el cual permite la gestión de la información por las Entidades Externas a la ONIDEX, en la CIE de la República Bolivariana de Venezuela. Después de haber implementado el Esquema de Autenticación del Applet SDM se arribó a las siguientes conclusiones:

- Durante el desarrollo de la investigación se realizó un profundo estudio de los estándares relacionados con tarjetas inteligentes, se analizaron y seleccionaron las herramientas necesarias para obtener una solución óptima en la implementación del esquema de autenticación, así como una exhaustiva revisión sobre los algoritmos para la autenticación asimétrica en applets. Se identificaron las metodologías existentes y se seleccionó XP por ser la más pertinente para el desarrollo de este software.
- Se definió el diseño y la arquitectura con la que se implementó el Middleware y el esquema de autenticación para el Applet SDM y se definieron los requerimientos no funcionales que se tuvieron en cuenta para un correcto funcionamiento de la solución.
- Se implementó un esquema de autenticación para el Applet SDM cumpliendo con todos los estándares y tecnologías definidas a lo largo del trabajo de diploma, verificando a través de mecanismos de autenticación que la CIE pertenezca al ciudadano correspondiente y la verificación de la entidad que hace uso de la misma.
- Las pruebas realizadas a la solución permitieron corregir los errores que se fueron dando al terminar cada iteración, logrando así que los resultados finales hayan sido satisfactorios.

Recomendaciones

Continuar la implementación del protocolo “01” del canal seguro según las especificaciones de GlobalPlatform con los niveles de seguridad “01” y “03”, es decir con MAC y cifrado con MAC para lograr una mayor seguridad en los datos enviados y recibidos en la tarjeta.

Bibliografía

Almeida Sotolongo, D. y. (2009). *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. .* Ciudad de La Habana, Cuba: Universidad de las Ciencias Informáticas.

encyclopedia2.thefreedictionary.com/applet. (s.f.).

encyclopedia2.thefreedictionary.com/middleware. (s.f.).

e-passport(singapur). (2010). *necmexico.wordpress.com/2010/08/05/implementacion-de-soluciones-biometricas-e-passport-en-singapur/*.

fnmt. (2010). *www.fnmt.es/index.php?cha=companies&scha=19&page=71&spage=147*.

Garcia, F. (2004). *Recomendaciones Metodológicas para la elaboración de trabajos de tesis.*

Gemalto. Developer Suite. (2010). *www.gemalto.com.*

GlobalPlatform. (2010). *www.globalplatform.org/specifications.asp.*

ISO/IEC 7816. (2005.). *ISO/IEC. info_isoiec7816-4{ed2.0}en.pdf.*

Java Card™ . (2005). *Application Programming Interface.*

Joskowicz, I. J. (s.f.). *Reglas y Prácticas en eXtreme Programming.*

Larman, C. (s.f.). *UML y Patrones.*

PC/SC, W. (2010). *www.pcscworkgroup.com.*

Pereda Viñolo, K. y. (2010). *Plataforma para el desarrollo de servicios en línea utilizando tarjetas inteligentes.* Ciudad de La Habana, Cuba: Universidad de las Ciencias Informáticas.

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

Rosas, H. G. (s.f.). *El método científico aplicado a las ciencias experimentales.*

RSA, L. (2000). *pkcs-1v2-0a1.pdf. .*

Sanpieri, R. H. (s.f.). *METODOLOGIA DE LA INVESTIGACION.*

Sintaxis ASN.1. (2008).

Wells, J. D. (1999). *www.extremeprogramming.org.*

Bibliografía Referenciada.

Almeida Sotolongo, Dayron y Sáez Vilar, Joel. 2009.*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. .* Ciudad de La Habana, Cuba: Universidad de las Ciencias Informáticas. : s.n., 2009.

encyclopedia2.thefreedictionary.com/applet. [En línea]

encyclopedia2.thefreedictionary.com/middleware. [En línea]

e-passport(singapur). 2010. necmexico.wordpress.com/2010/08/05/implementacion-de-soluciones-biometricas-e-passport-en-singapur/. [En línea] 2010.

fnmt. 2010. www.fnmt.es/index.php?cha=companies&scha=19&page=71&spage=147. [En línea] 2010.

Gemalto. Developer Suite. 2010. www.gemalto.com. [En línea] 2010.

GlobalPlatform. 2010. www.globalplatform.org/specifications.asp. [En línea] 2010.

ISO/IEC 7816. 2005..*ISO/IEC. info_isoiec7816-4{ed2.0}en.pdf.* 2005.

Java Card™ . 2005.*Application Programming Interface.* 2005.

PC/SC, Workgroup. 2010. www.pcscworkgroup.com. [En línea] 2010.

Pereda Viñolo, Katerina y Fernández Santana, Vismar. 2010.*Plataforma para el desarrollo de servicios en línea utilizando tarjetas inteligentes.* Ciudad de La Habana, Cuba: Universidad de las Ciencias Informáticas. : s.n., 2010.

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

RSA, Laboratorios. 2000.*pkcs-1v2-0a1.pdf. .* 2000.

Sintaxis ASN.1. 2008. 2008.

Wells, J. Donovan. Extreme Programming: A gentle introduction. 1999.
www.extremeprogramming.org. [En línea] 1999.

Glosario de términos

Middleware: Es una librería de *software* que media entre las aplicaciones que corren en la tarjeta inteligente y las que corren en una computadora.

APDU: Protocolo de Unidad de Datos de Aplicaciones.

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

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

SDM: Secure Data Mangar

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

Framework: Es una estructura de soporte definida en la cual un proyecto de *software* puede ser organizado y desarrollado.

MAC: Código de Autenticación de Datos.

Multiplaforma: Término usado para referirse a los *software* que pueden funcionar en diversas plataformas o sistemas operativos.

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.

Anexos.

Anexos 1: Plan de entrega.

Entregable	Fin Iteración 1	Fin Iteración 2
Esquema de autenticación	Febrero 2011	Mayo 2011

Tabla 23: Plan de entregas.

Anexos 2: Estimación de tiempo.

Historias de usuarios	Estimación
Inicializar comunicación con la tarjeta.	1
Finalizar comunicación con la tarjeta.	1
Establecer el canal seguro.	2
Verificar la autenticación del usuario PIN.	1
Verificar la autenticación del usuario por Mach onCard.	2
Verificar el certificado del token de acceso.	3

Tabla 24: Estimación de tiempo de las historias de usuario.

Anexos 3: Plan de iteraciones.

Iteración	**No. HU**	**Historia de Usuario**	**Duración Estimada(semanas)**
Iteración 1	1	Inicializar comunicación con la tarjeta.	1
	2	Finalizar comunicación con la tarjeta.	1
	3	Establecer el canal seguro.	2
Iteración 2	4	Verificar la autenticación del usuario por PIN.	1

	5	Verificar la autenticación del usuario por MoC.	2
	6	Verificar el certificado del token de acceso.	3

Tabla 25: Plan de iteraciones.

Anexo 4: Pruebas Unitaria en el applet.

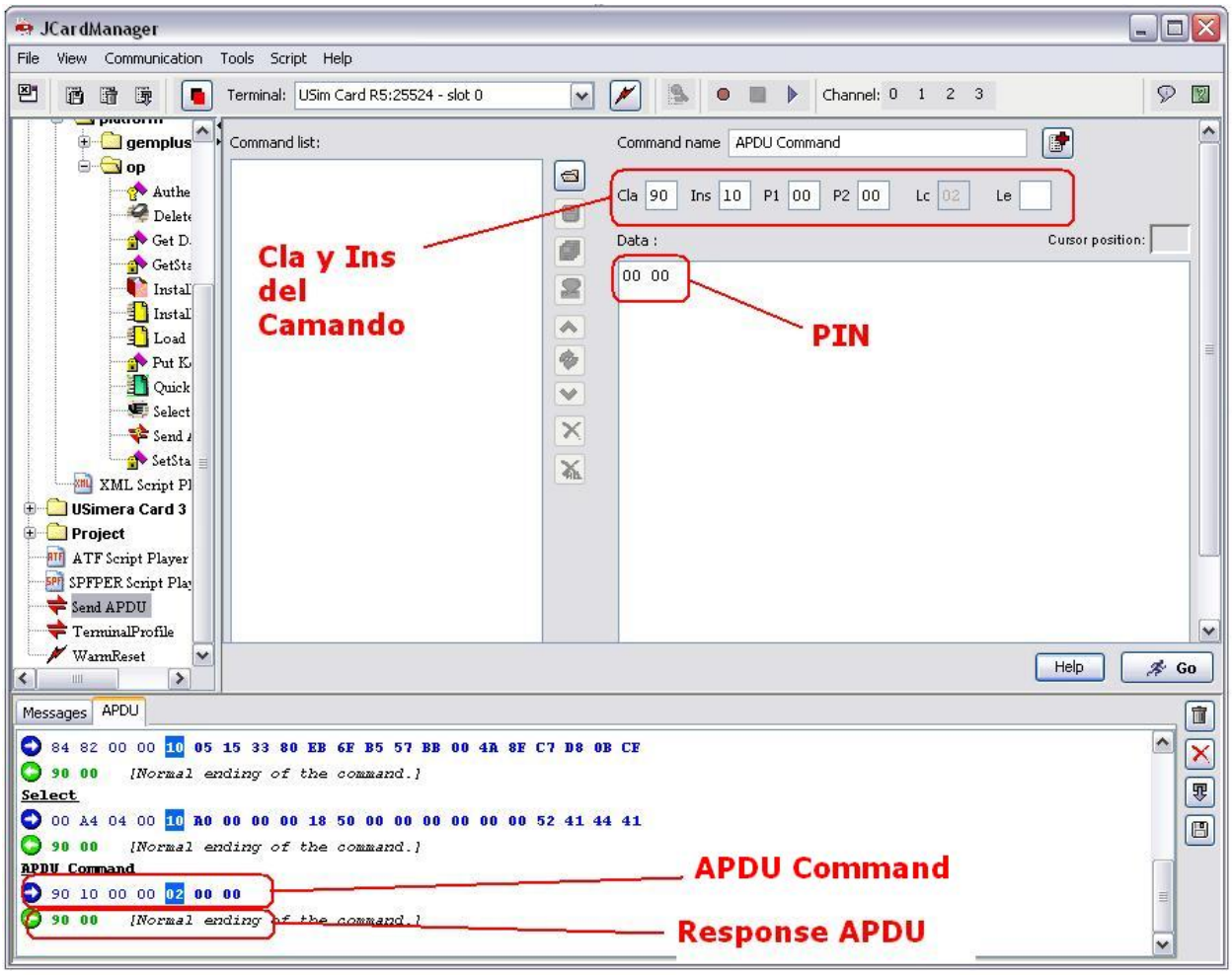

Figura 12: PU-1 "Verificar PIN"

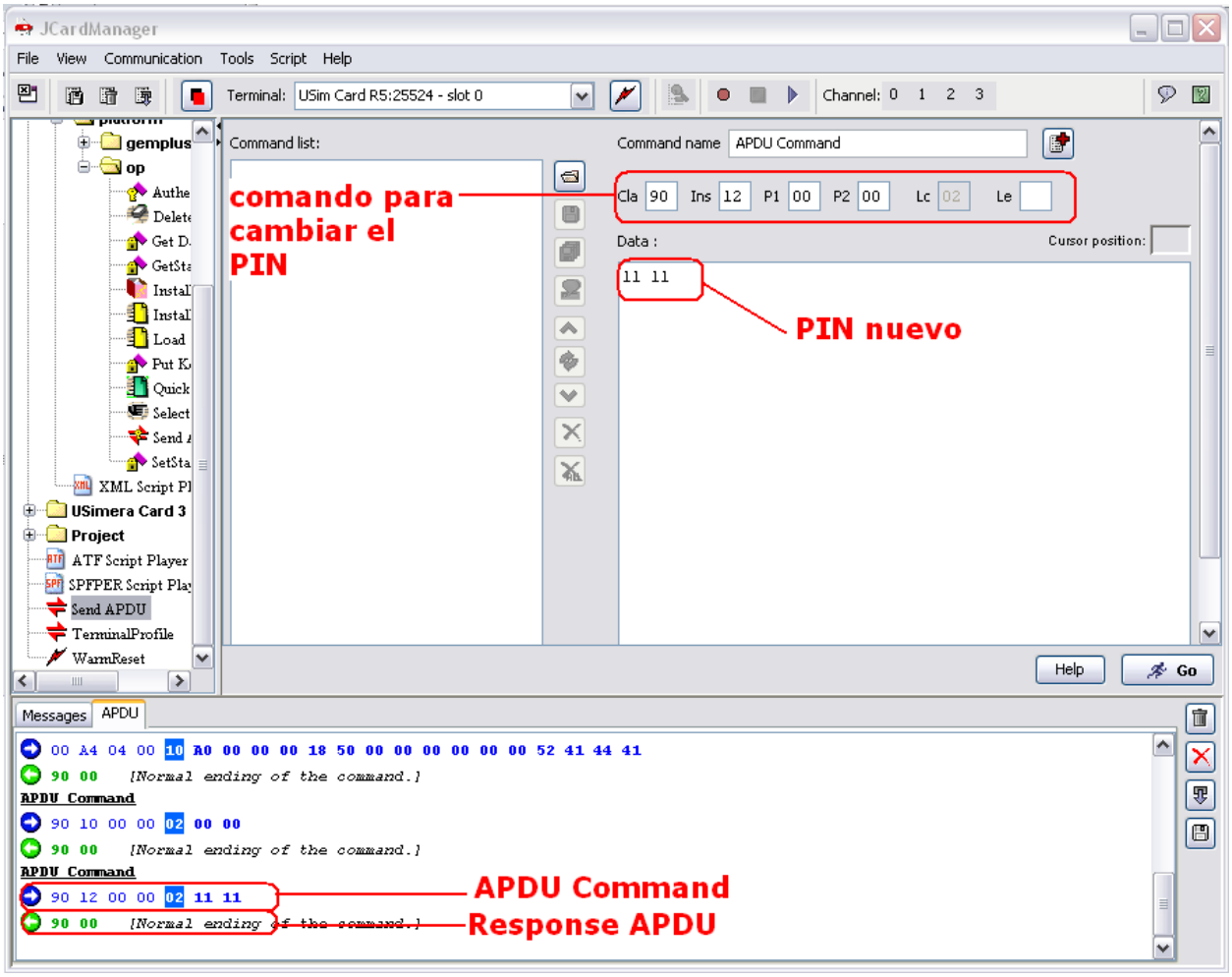

Figura 13: PU-2 "Cambiar PIN"

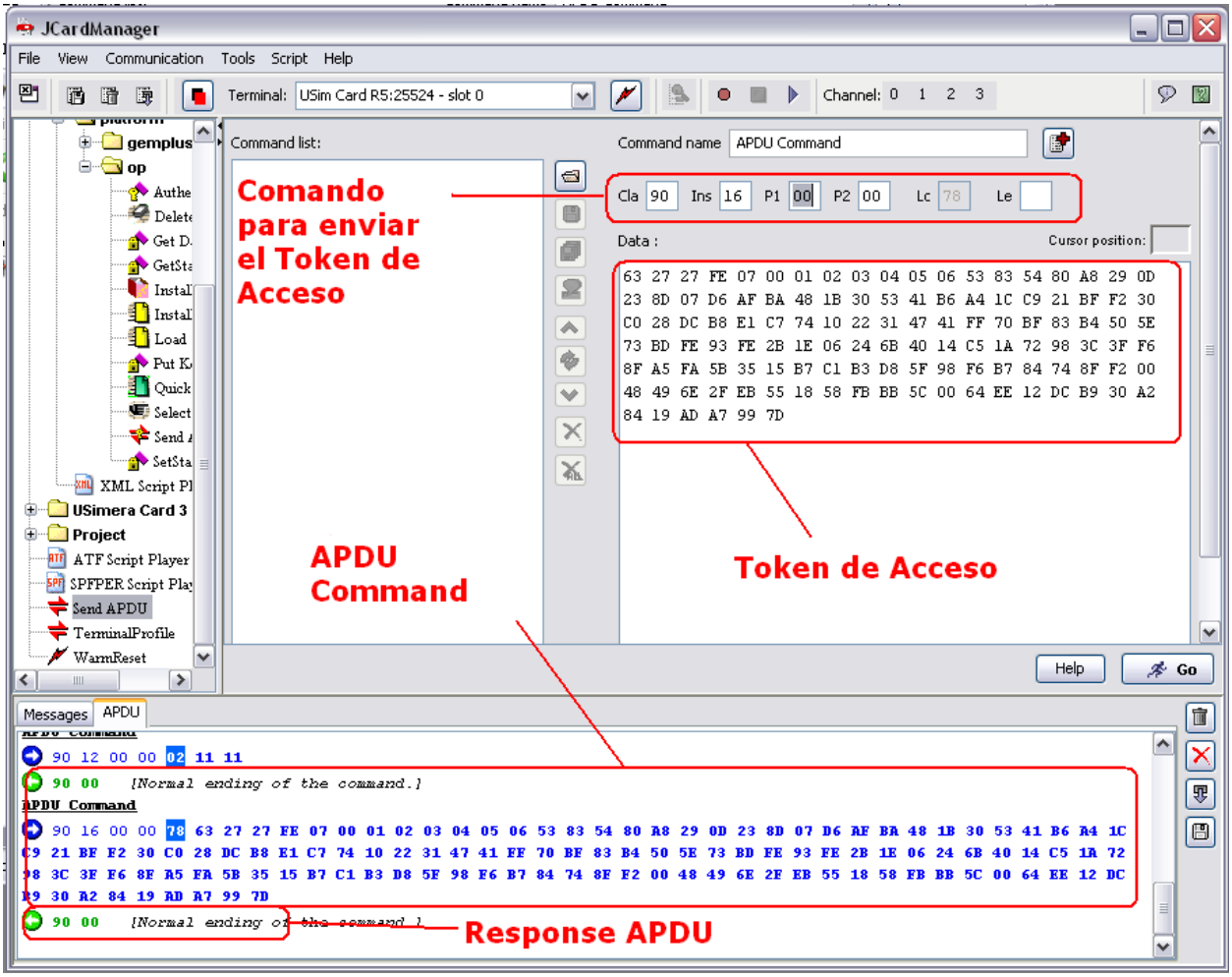

Figura 14: PU-3 "Enviar token de Acceso".

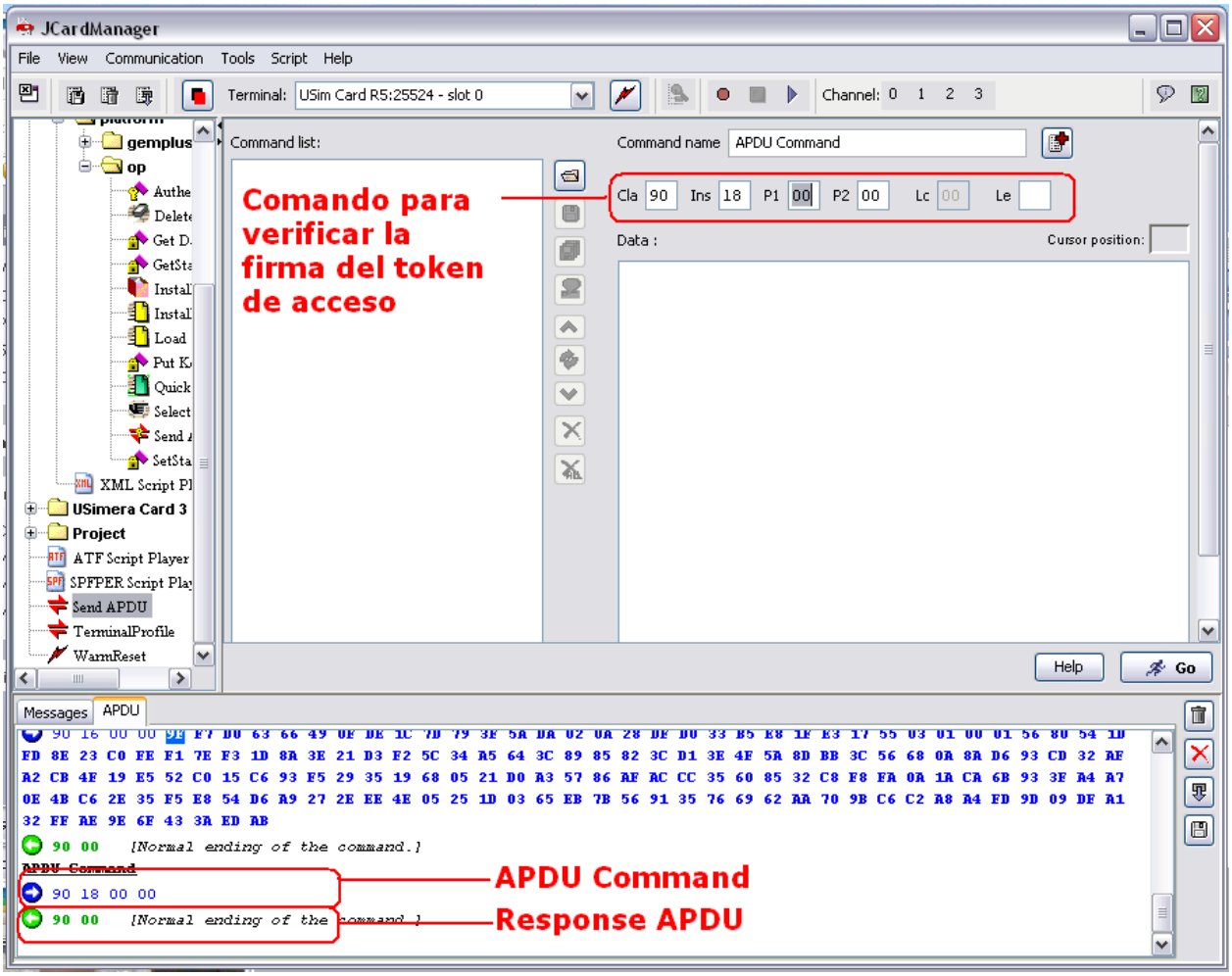

Figura 15: PU-3 "Verificar token de Acceso".

Anexo 4: Pruebas Unitaria en el Middleware.

Prueba de Unidad		
Nombre de prueba: VerificarPinTest.		
Estado: Satisfactoria	Tipo: Caja Blanca	**Última ejecución:**
Ejecutado por: Abraham Armas Pérez	**Verificado por:**	
Descripción: Para poder ejecutar la prueba previamente se deben haber asignado los valores del PIN (*0000*) si los datos son correctos la verificación se realiza de forma satisfactoria, de lo contrario, se lanza un mensaje mostrando el error que se originó.		
Criterio de Aceptación : Verificar Pin		

Resultados:

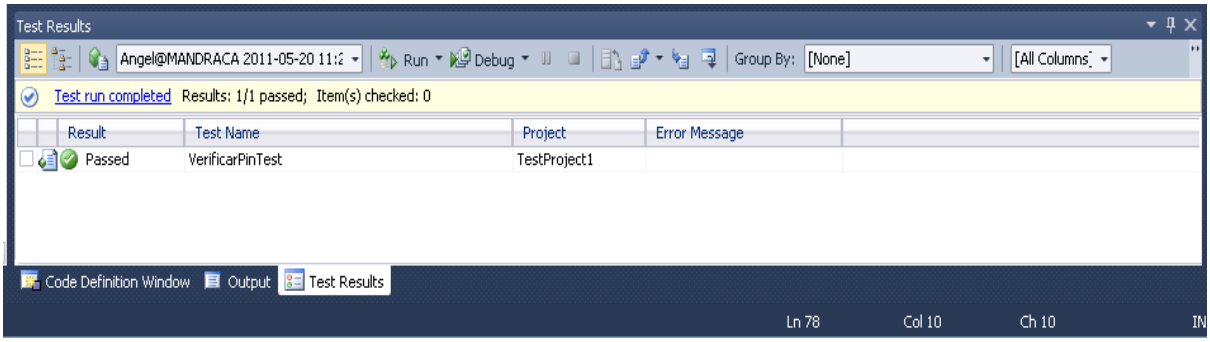

Tabla 26: Prueba Unitaria: VerificarPinTest.