Plataforma para el Acceso a Servicios desde Dispositivos Móviles, utilizando parámetros de autenticación basados en SIM Card en redes GSM

JAIME CAICEDO GUERRERO

RODRIGO HERNÁNDEZ CUENCA

Monografía presentada como requisito para optar el titulo de Ingeniero en Electrónica y Telecomunicaciones

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Telemática

Popayán, Marzo de 2006

CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

RESUMEN

Este trabajo de grado se centra en una de las tecnologías mas usadas actualmente en el desarrollo de aplicaciones para dispositivos móviles: J2ME (*Java 2 Micro Edition*). La plataforma construida involucra tanto a J2ME como a las dos tecnologías más importantes en el desarrollo de aplicaciones para tarjetas SIM, a saber, SAT (SIM Application Toolkit) [1] y Java Card [2]. Es importante mencionar que la SIM es una tarjeta "inteligente", es decir, es una tarjeta que le impone al usuario una serie de "reglas y restricciones" para poder acceder a cierta información almacenada en ella y que además está capacitada para realizar operaciones criptográficas, lo que la hace ideal para desplegarse en entornos donde se requiere un alto nivel de seguridad.

La plataforma se concibe gracias a SATSA (*Security And Trust Services API*) [3], uno de los más recientes APIs (*Application Programming Interface*) que la tecnología J2ME ha proporcionado. El propósito de SATSA es brindar servicios de confianza y seguridad a las aplicaciones para dispositivos móviles, basándose (en gran medida) en la capacidad que tiene de integrarse a un Elemento de Seguridad, elemento que en la mayoría de los casos corresponde a una tarjeta SIM.

No sobra mencionar el gran trabajo realizado en la exploración de tecnologías como SAT y Java Card, lo cual nos condujo a la producción de la base de conocimiento inicial para el grupo de investigación W@pColombia y para la Universidad del Cauca en lo que se refiere a las tecnologías utilizadas para el desarrollo de aplicaciones sobre tarjetas SIM.

Finalmente se logró construir la plataforma inicialmente concebida, la cual fue validada mediante un prototipo de comercio móvil que trabaja en un entorno simulado. Aunque tecnologías como *SIM Application Toolkit* si se pudieron manipular sobre dispositivos reales, lastimosamente la plataforma como tal no se pudo probar en un entorno real debido a la inexistencia actual en el mercado de dispositivos con soporte a SATSA.

INTRODUCCIÓN

LOS SERVICIOS MÓVILES

Actualmente en el mundo se observa un constante crecimiento en el número de usuarios de la telefonía móvil y aunque en los países desarrollados es mínimo, debido a que el gran crecimiento lo tuvieron hace unos años, en otros países en vía de desarrollo como los latinoamericanos se observa un incremento abrumador. En Colombia las cifras sorprendieron a finales del 2004, cuando el número de usuarios de telefonía celular superó al de abonados de telefonía fija, y en enero del 2006 asombraron aún mas, cuando las cifras decían que en un país con una población cercana a los cuarenta y cuatro millones de habitantes, uno de cada dos colombianos poseía un teléfono celular.

La realidad dice que la telefonía celular se convirtió en una necesidad para todos los colombianos, y en ese ambiente comienzan a ser demandados una gran cantidad de servicios móviles además de los clásicos servicios de SMS (servicio de mensajes cortos) y voz. Ahora el punto es que en el mercado se está imponiendo un modelo de tres partes, en donde además de los típicos usuario y operador de red se suma un tercero, el proveedor de servicios también llamado proveedor de contenido. El proveedor de servicios surge en parte debido a la aparición de varias tecnologías orientadas a los dispositivos móviles, entre ellas las más conocidas son WAP (Wireless Application Protocol) y J2ME (*Java 2 Micro Edition*).

Los servicios móviles mejoran la calidad de vida de las personas al disminuir tiempo, dinero, distancias y facilitar medios de pago, lo cual es lo ideal, en especial cuando la disponibilidad y el tiempo de las personas es muy escaso. Estos servicios tienen una gran ventaja y es que se puede hacer uso de ellos a cualquier momento y en cualquier lugar. Los servicios más comunes incluyen:

- Telemetría.
- Banca móvil (*m-banking*) para realizar transferencias bancarias.
- Acceso a Internet en busca de contenido como noticias, juegos, etc.
- Comercio móvil (*m-commerce*) para comprar, ofrecer y vender productos.

Cabe recordar que la mayoría de los servicios móviles surgen como una migración de los servicios fijos y que en muchos de ellos la **seguridad** es un factor decisivo a la hora de la escogencia por parte del usuario. Efectivamente en la era digital se desarrollaron y se siguen desarrollando una gran cantidad de infraestructuras y procedimientos que buscan garantizar la seguridad de las actividades realizadas por las personas. Ejemplos de lo anterior son el cifrado simétrico DES (*Data Encryption Standard*) y la infraestructura de clave publica PKI (*Public Key Infrastructure*).

Uno de los primeros logros en la migración de los procedimientos de seguridad de las redes fijas a las redes móviles fue la aparición, en los años 90, de la tarjeta SIM (Modulo de Identificación del Suscriptor) como parte de la red GSM (*Global System for Mobile Communications*). Los objetivos iniciales de la tarjeta SIM eran autenticar al suscriptor ante el operador y cifrar cierto tipo de información crítica intercambiada cuando el usuario establecía una conversación en la red de su operador o de otro operador (*Roaming*).

Paralelo a la evolución de la telefonía móvil y su capacidad para el despliegue de nuevos servicios ha crecido el problema de la seguridad, puesto que ahora tenemos la capacidad de acceder a redes inherentemente inseguras como Internet desde nuestros dispositivos móviles. Debido a lo anterior, los usuarios requieren seguridad en el acceso a sus servicios móviles, seguridad que debe garantizar el proveedor de servicios. La buena noticia es que las tecnologías móviles (como J2ME) y la SIM han evolucionado a tal punto, que actualmente es posible el acceso seguro a dichos servicios.

IMPORTANCIA DEL TRABAJO DE GRADO

En esta monografía se encuentran las características relevantes del proceso de exploración y construcción de una plataforma para el acceso seguro a servicios desde dispositivos móviles utilizando parámetros basados en SIM, la cual involucró a las tecnologías más utilizadas actualmente para el desarrollo de aplicaciones móviles: J2ME, SAT (*SIM Application Toolkit*) y Java Card.

Además, el presente trabajo de grado contribuye positivamente a la convergencia de tecnologías modernas en el área de seguridad móvil y a la mitigación de las necesidades de seguridad de los servicios móviles de 2.5 y 3G, garantizando comunicaciones móviles seguras (autenticación, integridad, confidencialidad y no repudio) extremo a extremo, proporcionando utilidades y herramientas para facilitar el desarrollo de aplicaciones por parte de los proveedores de servicios.

OBJETIVOS DE LA PLATAFORMA

La plataforma tiene como objetivo primario brindar a los desarrolladores de servicios móviles sobre redes de 2.5G y 3G las facilidades para que estos implementen la seguridad que dichos servicios requieren.

Teniendo en cuenta que actualmente GSM es el estándar para telefonía móvil más ampliamente difundido a nivel mundial y que existe un módulo definido por la arquitectura GSM destinado a la gestión de seguridad de usuario, la plataforma explota las capacidades y facilidades que en materia de seguridad brinda este módulo.

El Módulo SIM es la única parte de la red de telefonía móvil que permanece con el usuario y físicamente consiste en una tarjeta inteligente, cuyo objetivo es ser un componente de seguridad e identidad, diseñado para fortalecer las relaciones entre los clientes y los operadores, ayudar a la evolución hacia mejores servicios de valor agregado y facilitar el comercio móvil en redes de 2G, 2.5G y 3G. Así pues, aprovechando que el Módulo SIM esta en la capacidad de almacenar información importante de forma portable y segura y el nacimiento de nuevos estándares que permiten la interoperabilidad de las aplicaciones desarrolladas para el Módulo SIM, otro objetivo de la plataforma es facilitar la gestión del acceso de los usuarios a los servicios móviles mediante la utilización de los parámetros que el Módulo SIM almacena.

1. MÓDULO DE IDENTIFICACIÓN DEL SUSCRIPTOR (SIM)

1.1. TARJETAS INTELIGENTES

1.1.1. Generalidades

Una tarjeta inteligente o s*mart card* representa una de las plataformas computacionales más pequeñas de uso en la actualidad. Una *smart card* es una tarjeta de plástico que contiene un circuito integrado con memoria no volátil y un sistema operativo de tarjeta que se acompaña de protocolos de comunicación.

Hay diferentes usos para las tarjetas inteligentes, y estas pueden servir como:

- Tarjetas seguras capaces de identificar al portador usando algoritmos de autenticación avanzados y almacenar con toda seguridad información secreta como claves privadas.

- "Billeteras" electrónicas que almacenan valor mediante diferentes aproximaciones y permiten una variedad de pagos electrónicos.

- En las transacciones cumplen el rol una vez jugado por las tarjetas de cinta magnética comúnmente encontradas al respaldo de las tarjetas de crédito o débito.

- Procesadoras que llevan a cabo cálculos necesarios para el portador en un estilo de caja negra.

- Tarjetas de memoria que actúan como altamente portátiles bases de datos.

Las tarjetas inteligentes de la actualidad son programables y multi-aplicación lo que les permite ser reprogramadas después de que son fabricadas y además le permiten al usuario manejar varias aplicaciones sin que esto le implique llevar simultáneamente varias tarjetas (situación que típicamente sucede cuando un usuario tiene varias tarjetas de crédito).

Las Tarjetas Inteligentes han sido por mucho tiempo asociadas con seguridad, son resistentes al saboteo, desde ellas se provee una solución parcial a las necesidades de autenticación personal, no repudio y confidencialidad. Debido a esto las tarjetas inteligentes hacen ideal el almacenamiento de información crítica como identificaciones de usuario, claves DES (*Data Encryption Standard*), claves privadas RSA (*Rivest, Shamir and Adleman*) para cifrar o descifrar información y hacer firmas digitales.

1.1.2. Tipos de tarjetas inteligentes

Las tarjetas inteligentes pueden ser de contactos, ver figura 1-1, y sin contactos.

1.1.2.1. DE CONTACTOS

Estas tarjetas deben ser insertadas en un dispositivo lector inteligente o CAD (*Card Acceptance Device*) para que por medio de contactos, los llamados contactos dorados, pueda ser leída. Son las más utilizadas en el mundo. La figura 1-1 muestra una tarjeta de crédito.

Figura 1-1. Tarjeta inteligente de contactos.

Un CAD a menudo viene con sus propias librerías y la mayoría luce como la figura 1-2. Hace unos años un grupo de compañías crearon una especificación que estandarizó a los CAD como periféricos de un computador. Esta especificación es llamada *Personal Computer/Smart Card* o de forma abreviada PC/SC [4]. La especificación PC/SC ha sido implementada sobre Windows y Linux.

Figura 1-2. CAD para tarjetas inteligentes.

1.1.2.2. SIN CONTACTOS

Son similares a las de contacto con respecto a lo que pueden hacer y a sus funciones pero utilizan diferentes protocolos de transmisión en capa lógica y física, no utilizan contacto galvánico sino de interfaz inductiva (una antena) lo que les permite ser de media distancia sin necesidad de ser introducidas en un CAD. Una de las ventajas que esta tarjeta tiene es que el no tener contactos externos la hace más resistente a los elementos exteriores. Están empezando a utilizarse de forma importante en la industria del trasporte.

1.1.3. Características Hardware

Las tarjetas inteligentes son tarjetas de plástico que pueden ser de tamaños pequeños (1.5 cm x 2.5 cm) como las tarjetas SIM o de tamaños mas grandes como las tarjetas de crédito. La tarjeta de plástico contiene un circuito integrado, en el caso de una tarjeta de contactos el circuito se encuentra detrás de los mismos, ver figura 1-3. Las Tarjetas Inteligentes más comunes usan de 5 a 8 contactos sobre un lado de la tarjeta los cuales les permiten comunicarse con un CAD.

Figura 1-3. Circuito integrado en las Tarjetas Inteligentes.

Lo que hace a la tarjeta "inteligente", comparada con una tarjeta de memoria (como la RAM, *Random Access Memory*, de los computadores), es la implementación de reglas de control de acceso a la memoria: por ejemplo algunas áreas (como la clave del dueño de la

tarjeta) solo pueden ser leídas después de que éstas hayan sido escritas primero. Este control de acceso es ejecutado por una CPU (*Central Process Unit*) de 8 o 16 bits.

La capacidad de memoria de una tarjeta inteligente puede ser del orden de 1KB de RAM, 8KB de *ROM (Read Only Memory*) y 16KB de EEPROM (*Electric Erase Programmable ROM*), aunque las mas avanzadas del mercado [5] pueden llegar a tener 8KB de RAM y 1000KB de Flash (equivalente a la EEPROM).

1.1.4. Características Software

Una tarjeta inteligente contiene dos módulos Software básicos: el *Card OS* o sistema operativo de la tarjeta y las APIs (Interfaces de Programación de Aplicaciones) específicas de la industria (financiera, telecomunicaciones, etc).

Sobre estos dos módulos se ejecutan las aplicaciones o *Applets* que van a ser los utilizados finalmente por el usuario. Actualmente estas aplicaciones pueden ser cargadas, actualizadas o borradas a cualquier hora, lo que no sucedía hace unos años, donde esto sólo se podía hacer durante la fase de fabricación de la tarjeta.

1.1.5. El estándar ISO 7816

El título formal del estándar ISO/IEC 7816 [6] es "Tarjetas de Circuito Integrado con contactos eléctricos". Es el estándar internacional más ampliamente usado y referenciado para tarjetas inteligentes de contactos, cualquiera que esté interesado en tener un conocimiento técnico de ellas, debe familiarizarse con el ISO 7816 [6].

1.1.5.1. ISO/IEC 7816 PARTE 4: COMANDOS INTER-INDUSTRIA PARA INTERCAMBIO

Describe los mensajes o unidades básicas de intercambio con una tarjeta inteligente, los llamados APDU's (*Application Protocol Data Units*). Los APDU's pueden ir en dos sentidos, mensajes *command* enviados hacia la tarjeta, y mensajes *response* retornados por la tarjeta. Un APDU puede ser considerado un paquete de datos que contiene una completa instrucción o una completa respuesta desde una tarjeta. Para proporcionar esta funcionalidad, los APDU's tienen una estructura bien definida.

- *Command APDU*

Encabezado obligatorio				**Cuerpo opcional**		
CLA	INS	P1	P2	Lc	*Data field*	Le

Figura 1-4. Estructura *Command APDU.*

- CLA (1 byte): Clase de instrucción – indica la estructura y formato para una categoría de *command* y *response* APDUs.
- INS (1 byte): código de instrucción, especifica la instrucción del *command*.
- P1 (1 byte), P2 (1 byte): parámetros 1 y 2 para la instrucción.
- Lc: la longitud en bytes del *Data field.*
- *Data field*: Datos que son enviados a la tarjeta para que ejecuten la instrucción especificada en el encabezado.
- Le: especifica el número de bytes esperados en la respuesta de la tarjeta.

- *Response APDU*

Cuerpo opcional	***Trailer* obligatorio**	
Data Field	SW1	SW2

Figura 1-5. Estructura *Response APDU.*

- *Data field* (longitud variable): Datos en forma de secuencia de bytes.
- SW1 (1 byte) y SW2 (1 byte): *Status Words* – denotan el estado de éxito o fracaso producido después de procesar el *command* en la tarjeta.

El ISO 7816 parte 4 además establece un conjunto de comandos a lo ancho de todas las industrias para proporcionar acceso y seguridad en la transmisión de los datos de la tarjeta. Dentro de este núcleo básico, por ejemplo, están los comandos de lectura, escritura y actualización de registros y archivos. Se aplica a tarjetas de o sin contactos.

1.5.1.2. ISO/IEC 7816 PARTE 3: INTERFAZ ELÉCTRICA Y PROTOCOLOS DE TRANSMISIÓN

Describe los protocolos de transporte para los APDU's. Las estructuras de datos intercambiadas entre un CAD y la tarjeta inteligente usando el protocolo de transporte son llamadas TPDU's (*Transmission Protocol Data Units*). Un APDU puede ser transportado por el protocolo de transmisión T=0 que es orientado a byte y *half duplex*, o por el T=1, que es

un protocolo orientado a bloque (secuencia de bytes) *half duplex* asíncrono. Otros protocolos pueden embeber una APDU en su propia estructura de transporte.

1.5.1.3. ISO/IEC 7816 PARTE 9: COMANDOS ADICIONALES INTER-INDUSTRIA Y ATRIBUTOS DE SEGURIDAD

Especifica los comandos inter-industria de las Tarjetas Inteligentes, tanto de contactos como sin contactos, para gestión de archivos, por ejemplo creación y borrado. Estos comandos cubren el ciclo de vida total y por consiguiente algunos comandos pueden ser usados antes de que la tarjeta sea emitida a los vendedores minoristas o después de que la tarjeta haya expirado.

1.1.6. Otros estándares

Las tarjetas inteligentes, dependiendo del ambiente en el que se desenvuelvan pueden usar estándares horizontales o estándares verticales. Los estándares Horizontales pueden ser usados por todas las aplicaciones, mientras que los estándares Verticales son específicos a un sistema o a un dominio.

1.1.6.1. ESTANDARES HORIZONTALES

- ISO 10536: estándar ISO para tarjetas inteligentes sin contactos.
- OCF (*OpenCard Framework*) [7]: *Framework* basado en Java cuyo objetivo es proporcionar a los desarrolladores de aplicaciones del lado *host* (por ejemplo, el lado de un computador) un API que trabaje independientemente del fabricante del CAD el cual les permita comunicarse con una aplicación en la tarjeta.
- Java Card [2]: Estándar para Applets Java que corren en una tarjeta inteligente
- PKCS#11 : *Cryptographic Token Interface Standard*
- CDSA (*Common Data Security Architecture*): una infraestructura de seguridad abierta desarrollada por Intel.

1.1.6.2. ESTANDARES VERTICALES

- Mondex, Proton: Estándares para dinero digital.

- VisaCash: Estándar para tarjetas débito que lleva el rastro de las tarjetas sobre el servidor.

- MPCOS: Tarjetas de propósito general que te permiten implementar tu propio tipo de moneda o símbolos.

- EMV: Especificación definida por Europay, Mastercard y Visa para pagos con tarjetas inteligentes en la industria financiera.

- GSM 11.11 y 11.14 [8]: Estándar de teléfonos celulares para tarjetas SIM, estandarizado por la ETSI [9].

- *Open Platform*: Conjunto de especificaciones para multi-funcionales tarjetas inteligentes y terminales, estandarizado por GlobalPlatform (una asociación establecida por organizaciones de varias industrias bajo el liderazgo de Visa *International* en 1999).

1.1.7. Entorno de las tarjetas inteligentes

Las tarjetas inteligentes al no tener teclado ni Display, necesitan de un terminal, usualmente llamado dispositivo receptor de tarjetas (CAD) o lector de tarjetas inteligentes.

Los dos tipos de CAD más encontrados son los lectores de tarjetas de crédito y los comúnmente encontrados por desarrolladores, CAD conectados directamente al puerto serial de un computador de escritorio. Mediante el uso de ambientes de desarrollo, los programadores pueden crear código para tarjetas inteligentes, descargarlo a una tarjeta prototipo, y de esta forma crear nuevas aplicaciones. Muchos de los grandes vendedores tienen Ambientes de desarrollo propietarios [10].

Hay muchas clases de conjuntos de comandos para tarjetas inteligentes. Efectivamente, hay muchos ambientes y ellos son tan especializados que el número de programadores de tarjetas inteligentes es muy pequeño. Considerando que las Tarjetas Inteligentes son tradicionalmente programadas en plataformas específicas de lenguajes ensambladores y

que cada vendedor tiene un lenguaje diferente, el número de programadores podría llegar a ser sorprendentemente pequeño.

La situación podría ser peor. Al menos la mayoría de vendedores cumple con el estándar ISO 7816 que proporciona una base común y asegura que las tarjetas inteligentes tienen pines terminales de salida similares, aceptan protocolos estándar de mensajes, y almacenan alguna información en bases de datos de diseño común. Debido a la existencia del ISO 7816, por citar un ejemplo, un CAD Gemplus puede leer una tarjeta inteligente Axalto.

1.2. LA TARJETA SIM

1.2.1. Generalidades

La tarjeta SIM (Módulo de Identificación del Suscriptor) es una clase particular de tarjeta inteligente que ha tenido gran aceptación dentro del mundo de la telefonía celular. Las principales razones por las que ha tenido este gran auge son las siguientes:

- Aunque es un módulo perteneciente a la arquitectura de la red de telefonía celular, es la única parte de ella que permanece con el usuario fortaleciendo así la relación del operador con el cliente.

- Su objetivo principal es ser un componente de seguridad e identidad, el cual a través de los años se ha ido mejorando y actualmente permite identificar al usuario no sólo ante el operador de telefonía sino también ante un gran número de proveedores de servicios de valor agregado.

- Brinda la capacidad de almacenar información importante (como contactos, mensajes de texto, etc) de forma segura y portable [11], lo cual le facilita al usuario el proceso de cambio de terminal, sin que esto implique la pérdida de dicha información o el cambio de su número telefónico.

- Es utilizada para el despliegue de servicios y aplicaciones donde la seguridad es un factor clave, ayudando a simplificar la agitada vida del ser humano. Ejemplos son actividades bancarias, m-commerce y servicios basados en PKI (*Public Key Infrastructure*) sumando la posibilidad de mantener un gran control tanto de las llamadas y los datos salientes como entrantes.

Actualmente este módulo se encuentra en la mayoría de redes celulares del mundo, tomando un nombre diferente dependiendo del tipo de red en el que se halle, a saber:

- SIM, para redes GSM (*Global System for Mobile Communications*) [11].
- RUIM (*Removable User Identity Module*), para redes CDMA (*Code Division Multiple Acces*) [12].
- USIM (*Universal Subscriber Identity Module*) o UICC (*Universal Integrated Circuit Card*), para redes UMTS (*Universal Mobile Telecommunications System*) [13].

1.2.2. La SIM y GSM

GSM (*Global System for Mobile Communications*) es un estándar Europeo para telefonía celular digital que ha sido adoptado en casi todo el mundo. Las especificaciones técnicas de GSM, hechas por la ETSI [9], definen las diferentes entidades que forman la red por la definición de sus funciones y requisitos de interfaz. La arquitectura de la red GSM se presenta en la figura 1-6.

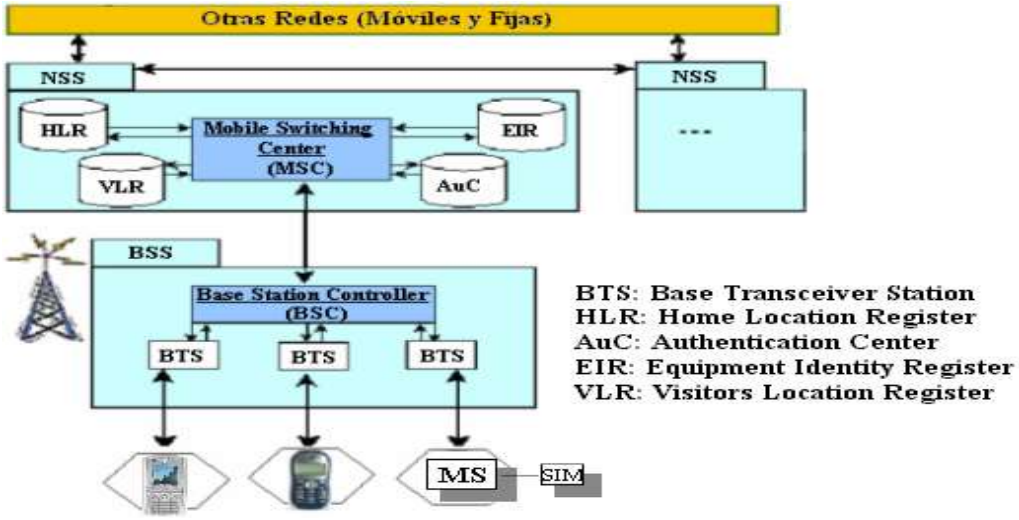

Figura 1-6. Arquitectura GSM.

La red GSM puede ser dividida en cuatro partes principales:

- La MS (*Mobile Station*)
- El BSS (*Base Station Subsystem*)
- El NSS (*Network Switching Subsystem*)
- El OSS (*Operation and Support Subsystem*).

Aunque GSM pertenece a la segunda generación de sistemas de telefonía Celular, es la base para UMTS [13], el sistema de tercera generación mas usado en el mundo.

La MS está conformada por la SIM y el ME (*Mobile Equipment*). El objetivo inicial de la tarjeta SIM es proporcionar un ambiente seguro y resistente al saboteo (extremadamente difícil de violar bajo una variedad de ataques) que identifique a un usuario móvil particular, lo cual se logra manejando claves digitales cifradas (de uso de los operadores GSM) que autentican a los usuarios en la conexión a la red y rastrean aquellas actividades que realizan una vez están "al aire". La SIM mantiene conexión permanente con la red desde que el ME se enciende, sin la SIM el ME no se conecta a la red. Esta conexión autenticada y conciente de la ubicación del usuario es la que permite efectuar el *Roaming*.

1.2.3. Características Hardware

El tamaño de la tarjeta SIM es conocido como "ID-1" (ver figura 1-7). Las dimensiones físicas de la ranura para conectar la SIM al ME son: 15 mm de ancho y 25 mm de largo.

Figura 1-7. Tamaño tarjeta SIM

La CPU (*Central Process Unit*) es de 8/16 bit y funciona con una frecuencia de reloj de 1MHz a 10 MHz. Las primeras SIM ejecutaban 1/3 de millón de instrucciones por segundo (MIP) con 16K de memoria. Las actuales tarjetas SIM poseen 32/64 KBytes de EEPROM, necesitan una fuente única de voltaje de 5V ± 10% (en muchos casos es 3V ± 10%) y pueden funcionar en un rango de temperatura de -25 a 75 ºC.

Según el estándar GSM 11.11 [8] una tarjeta SIM tiene 8 contactos (C1, C2,…, C8), pero en el ME (Equipo Móvil) se deja opcional el soporte a los contactos C4 y C8. Los contactos tienen las siguientes características:

C1: es Vcc, su valor puede variar entre 4.5 V – 5.5 V, $I_{C\,Max}$ es 10 mA
C2: Reset (RST)
C3: Clock, puede tener una frecuencia de 1 MHz – 5 MHz, proporcionado por el ME.
C5: Ground
C6: Vpp

C7: I / O, a un baudiaje = (frecuencia del Clock) / 372.

1.2.4. Ciclo de vida de la SIM

Para muchos operadores, el ciclo de vida comienza durante la pre-personalización de la SIM, ver figura 1-8. Desafortunadamente, el tiempo de acción desde el punto de fábrica (generalmente en Francia u Holanda) al punto de venta puede ser muy prolongado. Sólo a través de mejores gestiones, tales tiempos iniciales de la SIM pueden ser disminuidos. Después de que la tarjeta ha sido pre-personalizada, la logística procede a distribuir las tarjetas a "minoristas" (vendedores de cada ciudad) dentro de unos pocos días. Aún dentro del dominio al por menor, los minoristas compiten dentro de un agitado clima de negocios (recordando que vender una SIM es vender un línea de telefonía celular) y muchos poseen en los puntos de venta modernas e independientes herramientas de gestión de la SIM, que benefician a los altos vendedores (operadores) y brindan oportunidades a vendedores cruzados. El contenido de las tarjetas no es estático después de la personalización, está modificándose continuamente, es decir, la SIM puede ser manejada a través del ciclo de vida. De este modo, la tarjeta puede siempre contener los datos correctos en los momentos justos, con los requerimientos individuales del usuario tomados en consideración.

Además de esto, el usuario puede requerir asistencias del centro de atención al cliente y desde los servicios de auto-provisión del operador que utilizan mecanismos OTA (*Over The Air*). Estos aseguran que la tarjeta permitirá efectuar *Roaming*, mantendrá certificados para un tipo de uso particular y brindará los servicios correctos.

El operador para no obtener bajas rentabilidades de dominio, debe tomar control y optimizar el manejo del ciclo de vida de la SIM a través de la gestión de los usuarios que la controlan a cualquier momento y en cualquier lugar. El ciclo de vida de las SIM [14] se basa en la estrategia del operador de utilizar componentes como: herramientas de puntos de venta, Repositorios SIM, gestión de la interfaz OTA (sobre el aire), tarjetas con facilidades pre-personalizadas y proveedores de contenido en un camino unificado.

1.2.4.1. JUGADORES EN EL CICLO DE VIDA DE LA SIM

Para captar los requerimientos individuales de los usuarios, el operador necesita actualizar constantemente su infraestructura y preguntarse ¿A dónde se dirigirá el usuario cuando nuevos servicios sean requeridos?

La figura 1-8 muestra cómo una tarjeta es manejada por diferentes jugadores a lo largo de todo su ciclo de vida. Históricamente el contenido de la tarjeta podía sólo ser controlado en la pre-personalización (a la izquierda en la figura). Los requerimientos del mercado han causado la introducción de soluciones que extienden el control de la tarjeta también durante y después de la emisión al usuario (mirar a la derecha de la figura). Esto significa que el contenido de la tarjeta no es muy estático después de la personalización, pero preferentemente que la tarjeta puede ser modificada durante todo el ciclo de vida.

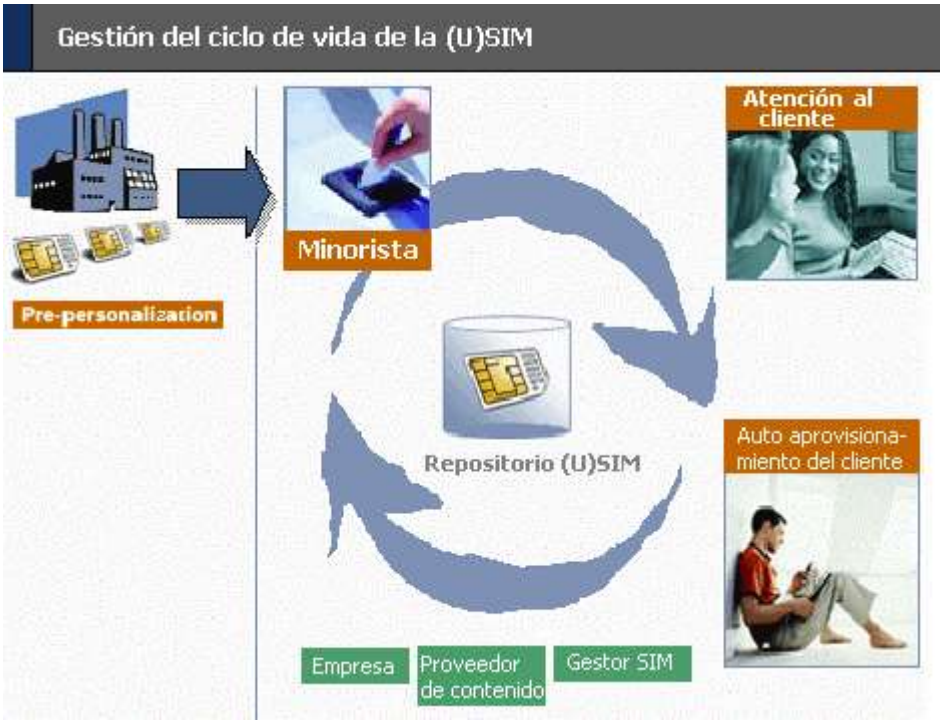

Figura 1-8. Ciclo de vida de la SIM

Diferentes individuos tienen diferentes requerimientos. Algunos usuarios pueden tener acceso a un computador, y pueden entrar a una página Web para iniciar una descarga OTA. Algunas veces el usuario necesita asistencia personal, caso en el cual los puntos de venta son la mejor solución.

Muchas partes tienen un interés en el contenido de la tarjeta y pueden beneficiarse con la habilidad de controlar remotamente este contenido, por vía OTA o leyendo la tarjeta. Como las características y funciones de las tarjetas así como también el equipo del usuario evolucionan, habrá más allá aspectos de control, y nuevas partes interesadas en acceder a la tarjeta.

- El Gestor SIM

El Gestor SIM es la persona(s) encargada de los contenidos y configuraciones de bloques (conjuntos) de tarjetas. Una de las muchas tareas para un manejador SIM es monitorear las existencias y control logístico para obtener cortos tiempos iniciales y asegurar la calidad de las tarjetas. Adicionalmente, el gestor SIM define el perfil de personalización de las tarjetas e inicia una serie de actualizaciones sobre el aire de gran parte de la base de usuarios. Dentro de la SIM, algunos operadores colocan sus propios servicios de valor agregado. No existe contacto directo con los usuarios.

- El Minorista (vendedor)

El punto de venta es lugar donde el usuario compra el Equipo Móvil y recibe una suscripción. Por el establecimiento de una relación con el usuario, el Minorista trata de construir un mercado y conocer lo que el usuario necesita, quiere y desea configurar. Este es el lugar para preguntarle a un usuario acerca de sus requerimientos personales hacia la tarjeta y además es el primer y único sitio en donde el servicio personal puede ser provisto con potencialidad para ventas cruzadas o ventas superiores. Una mutua relación entre el Minorista y sus usuarios puede ayudar al operador a simplificar requerimientos y formular estrategias consecuentemente.

En el punto de venta el vendedor a menudo tiene una alta velocidad de conexión a Internet lo cual hace que la entrega del proceso de configuración o despliegue de nuevos servicios sea mucho mas atractiva que sobre el aire.

- El cliente

El usuario es el más valioso elemento para el operador. Sin embargo, dentro del actual clima de negocios la batalla para la obtención de usuarios es eventualmente dura. Para el operador, la SIM puede ser la clave para retener la satisfacción y lealtad del cliente. Al manipular el concepto de Gestión del ciclo de vida, el operador podría ser capaz de usar una plataforma estandarizada que ofrezca al usuario la personalización libre que el merece y requiere, lo que como consecuencia le permitirá al operador incrementar sus ingresos promedio por usuario (ARPU). El usuario requiere auto-aprovisionamiento, soporte de atención al cliente y soporte cara a cara por parte del minorista.

- El centro de atención al cliente

El centro de atención al cliente se mantiene por fiabilidad, sensibilidad y es vital en las relaciones con el cliente durante un rango de indagatorias y problemas del usuario final. La comunicación es principalmente llevada vía telefónica, e-mail o chat.

Debido a la evolución tecnológica, el centro de atención al cliente se apoya e impulsa en software que unifica los disparejos datos del cliente al operador, da soporte a ventas cruzadas y presta ayudas a clientes autosuficientes mediante un auto-aprovisionamiento sobre un portal inalámbrico.

El centro de atención al cliente es capaz de iniciar descargas sobre el aire y chequeos del estado de una tarjeta específica. El uso de más datos informativos desde el Repositorio SIM podría presentar una continua vista de la información crítica acerca del usuario – esto podría ser un diferenciador competitivo y crítico del éxito.

- Empresa

Las empresas tienen requerimientos específicos. Por ejemplo, hay empresas que podrían querer manejar los contactos (guías telefónicas) de sus empleados, bloquear las llamadas a ciertos números y decidir que todos sus empleados tengan un cierto conjunto de servicios sobre su tarjeta, como acceso a la guía telefónica de la corporación, más aún, las empresas podrían usar el Equipo Móvil cuando se "use afuera" la firma digital o funcionalidades de autenticación de personal sin inversiones adicionales en tarjetas inteligentes y lectores. El operador podría usar soluciones accesibles sin adicionar capas de complejidad que podrían aumentar el tiempo de desarrollo, retardar los tiempos líderes, decrementar el rendimiento o incrementar los costos de manejo.

Otra área de progreso que atrae a operadores y empresas es el campo máquina-a-máquina (M2M). El número de módulos M2M inalámbricos excedió los 100 millones en el 2004 y un típico servicio incluye la Telemetría. El servicio de gestión OTA juega un significante rol en la configuración y despliegue de servicios seguros. Para asegurar la satisfacción de la empresa, el operador debe simplificar el procedimiento de introducir nuevas series de tarjetas SIM, mejorando la conectividad, ofreciendo integración al dominio empresarial y mejorando la calidad del servicio.

- Proveedor de contenido

Los proveedores de contenido son compañías que proveen servicios a los usuarios finales. Un ejemplo es una agencia de contenido como un periódico o una institución financiera. El contenido es a menudo suministrado y personalizado por el usuario mediante la vía de auto-suministro. Más aún, el proveedor de contenido necesita iniciar el servicio de descargas, llaves de seguridad y configuraciones. Esto es a menudo hecho en una serie de actualizaciones para una gran cantidad de usuarios.

1.2.4.2. IMPORTANTES ELEMENTOS EN EL CICLO DE VIDA

- El Repositorio SIM

El repositorio SIM es un instrumento vital dentro del concepto de Gestión del ciclo de vida. Para toda empresa, el control de los negocios es de extremada importancia y el camino de conocer y/o controlar la SIM es el método que menos disgusta. ¿A quien no le gustaría tener un simple punto de entrada para todos los perfiles SIM, servicios y datos? El repositorio es el punto clave para la gestión centralizada y reducida del número de bases de datos y clientes "desconectados" (por ejemplo, celulares robados) requeridos dentro de varios dominios del operador. Más aún, el Repositorio le permite al operador construir potentes y personalizados servicios de valor agregado, tomando ventaja de características disponibles sin adicionar capas de complejidad que podrían prolongar el tiempo de despliegue, retardar los tiempos líderes y disminuir el rendimiento o incrementar los costos de manejo.

Esto es deseable para un repositorio que intercambia información con otros tipos de sistemas y trabaja en un ambiente integrado dentro del dominio del operador. Por ejemplo, el Repositorio debe capturar el sistema completo que llevará la logística de las tarjetas SIM, desde ordenar la producción, personalización e integración dentro de sistemas como el AuC (Centro de Autenticación) figura 1-6, HLR (*Home Location Register*), fábricas de manufactura de tarjetas, portales de provisión, CRM (Gestión relacionada al cliente), ERP (Planeación de Recursos de la Empresa), atención al cliente y pago, así como también las plataformas OTA. Por consiguiente los repositorios son frecuentemente basados sobre una plataforma abierta y tienen, por defecto, interfaces bien definidas y *Shell* adaptadores para despliegue de servicios, integración a otras aplicaciones, directorios y sistemas.

El operador puede obtener grandes ventajas competitivas a través de un mejor entendimiento de los usuarios. El intento del proceso CRM que mejora el servicio al usuario y expande el diálogo entre el operador y su usuario, es a menudo frustrado. Al utilizar el Repositorio SIM, CRM y manejadores de minas de datos, se podría tener la habilidad de diferenciar lucrativos usuarios, lucrativos servicios y aplicaciones que están fracasando. La precisión de los datos acerca de las demandas del usuario podría ofrecer una nueva dimensión de auto-aprovisionamiento, puntos de venta y mecanismos de gestión de OTA.

- Sistema de pre-personalización

Habilita en el operador la personalización de tarjetas "en casa". Por medio de un sistema de pre-personalización el operador puede en gran medida reducir tiempos iniciales para introducir nuevas versiones de tarjetas, y además la posibilidad de emitir cantidades pequeñas de tarjetas (por ejemplo al mercado corporativo).

- A cualquier hora, en cualquier lugar y sobre cualquier tarjeta SIM

Habilita una plataforma OTA (sobre el aire) que desempeña actualizaciones sobre el contenido total de la tarjeta después de que esta ha sido expedida al usuario, brindándole la libertad de elegir el más apropiado servicio. La plataforma actualiza configuraciones, servicios y todo tipo de contenido sobre la tarjeta.

1.2.5. Rol de la SIM

Históricamente, la SIM ha jugado un papel centrado en los ingresos por el tráfico de voz y el desarrollo de servicios un poco más avanzados, pero en la actualidad se percibe un fuerte desarrollo en los servicios que generan ingresos por el tráfico de datos, cosa que ha tomado mayor significado a partir del 2002 gracias a los SMS, especialmente en países en vía de desarrollo.

En cada teléfono móvil la tarjeta SIM es la única parte de la red de telefonía celular que permanece con el cliente, por tal razón el operador puede usar este componente para interactuar directamente con el móvil del usuario con imágenes o mensajes SMS, brindando una eficiente, económica e invaluable forma de obtener un máximo impacto con esfuerzos mínimos. Pero además, se pueden aprovechar las características únicas de la SIM para obtener datos muy importantes de una forma segura y confidencial habilitando el ingreso a servicios más avanzados y personalizados para cada uno de los usuarios.

Una de las características más destacadas de la SIM es su rol bien definido y estandarizado, característica que se convierte en una fortaleza al brindar una total independencia entre su funcionamiento y el teléfono móvil. Un usuario puede cambiar su teléfono móvil sin perder sus datos personales almacenados en la tarjeta SIM y sin necesidad de cambiar su suscripción ante el operador de red, reduciendo los costos logísticos y administrativos por parte del operador y brindando libertad y total movilidad al usuario.

La tarjera SIM también ayuda a la red a identificar las capacidades SIM Application Toolkit [1] de un nuevo equipo móvil, por ejemplo, cuando un usuario actualiza su tradicional

teléfono móvil GSM hacia un avanzado teléfono móvil MMS con pantalla a colores, la SIM puede identificar el nuevo terminal y ofrecer nuevos servicios a la medida según el perfil de este teléfono móvil. La SIM puede mostrar un simple menú de descubrimiento de servicios para guiar a los usuarios a través de las funcionalidades del nuevo terminal y facilitar así la apropiación de las nuevas características y servicios, asegurando que los clientes puedan de una forma simple y descomplicada optimizar el uso de su teléfono móvil.

El creciente desarrollo de la tecnología ha llevado a la creación de nuevos teléfonos móviles que parecen mini-computadores y para poder aprovechar sus características hardware con un máximo rendimiento e interoperabilidad, las tecnologías como Java [2] también evolucionan permitiendo el despliegue de aplicaciones avanzadas como juegos interactivos y en red, juegos de apuestas y una innumerable gama de posibilidades multimediales. Lo anterior puede ser realizado por el operador tomando ventaja de la interoperabilidad y seguridad que brinda la SIM.

La SIM es por lo tanto una herramienta que le permite al operador y a los proveedores de contenido desplegar aplicaciones orientadas a datos. En particular, la SIM soporta firmas digitales y robustos esquemas de autenticación como los definidos en los requisitos de la Unión Europea para el despliegue de casos de negocio viables en el comercio móvil para operadores móviles, instituciones financieras y comerciantes.

1.2.6. Características Software

Se puede decir que la tarjeta SIM posee tres módulos Software:

- El sistema operativo (*Card OS*): maneja el acceso a los datos almacenados en diferentes archivos. El Card OS se almacena en la memoria ROM de la tarjeta. Su implementación está muy ligada al Hardware, no lográndose aún manejar un Sistema Operativo de disco (como el de los computadores) que se pueda borrar e instalar en cualquier momento.

- El conjunto de comandos, procedimientos y APIs usados durante la fase de operación de red GSM que además permiten la ejecución de Aplicaciones.

- Aplicaciones: consisten de un conjunto de mecanismos de seguridad, archivos, datos y protocolos que se almacenan en la memoria EEPROM de la tarjeta. Las aplicaciones se ejecutan por petición del usuario.

1.2.7. Modelo Lógico de la SIM

Los archivos están organizados en una estructura jerárquica y pueden ser de tres tipos: MF (archivo maestro), DF (archivo dedicado) o EF (archivo elemental). La figura 1-9 muestra las relaciones estructurales generales que pueden existir entre archivos. Los archivos están compuestos de un encabezado, el cual es internamente manejado por la SIM, y opcionalmente de un cuerpo. La información del encabezado está relacionada a la estructura y atributos del archivo, esta información es fijada durante la fase administrativa. El cuerpo contiene los datos del archivo.

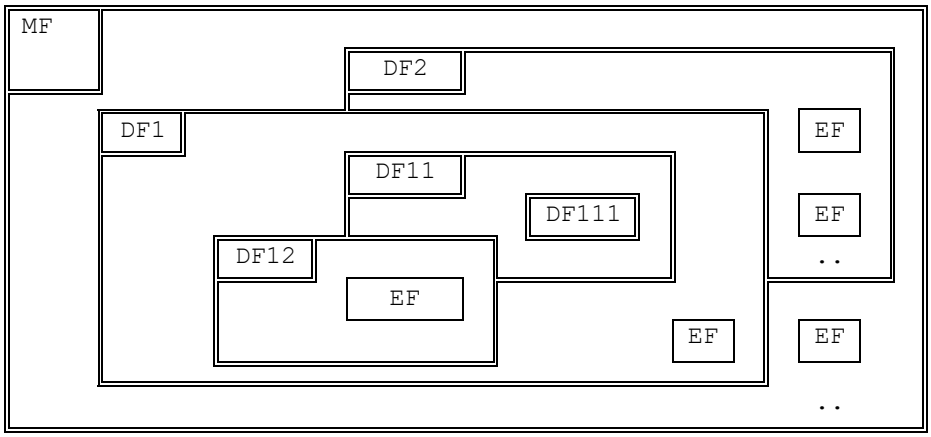

Figura 1-9. Relaciones estructurales de los archivos en la SIM

1.2.7.1. CONTENIDO DE LOS ARCHIVOS ELEMENTALES (EF)

Todos los archivos pueden ser leídos por un usuario con el equipo apropiado, pero sólo algunos EFs pueden ser escritos (el operador si puede escribir todos los EFs). No sobra decir que el nombré del archivo está directamente relacionado con su contenido.

- EFs del nivel MF

  En este nivel, hay dos EF: EF_{ICCID} y el EF_{ELP}. Es decir se puede acceder a los parámetros:

  - ICCID (Identificación ICC)
  - ELP (*Extended Language Preference*)

- EFs del nivel de aplicación GSM

  En este nivel (DF_{GSM}) están los siguientes archivos elementales: EF_{LP} (*Language Preference*), EF_{IMSI} (IMSI), EF_{Kc} (*Ciphering key Kc*), $EF_{PLMNsel}$ (*PLMN selector*), EF_{HPLMN} (*HPLMN search period*), EF_{ACMmax} (*ACM maximum value*), EF_{SST} (*SIM service table*), EF_{ACM} (*Accumulated call meter*), EF_{GID1} (*Group Identifier Level 1*), EF_{GID2} (*Group Identifier Level 2*), EF_{SPN} (*Service Provider Name*), EF_{PUCT} (*Price per unit and currency table*), EF_{CBMI} (*Cell broadcast message identifier selection*), EF_{BCCH} (*Broadcast control channels*), EF_{ACC} (*Access control class*), EF_{FPLMN} (*Forbidden PLMNs*), EF_{LOCI} (*Location information*), EF_{AD} (*Administrative data*), EF_{Phase} (*Phase identification*), EF_{VGCS} (*Voice Group Call Service*), EF_{VGCSS} (*Voice Group Call Service Status*), EF_{VBS} (*Voice Broadcast Service*), EF_{VBSS} (*Voice Broadcast Service Status*), EF_{eMLPP} (*enhanced Multi Level Pre-emption and Priority*), EF_{AAeM} (*Automatic Answer for eMLPP Service*), EF_{CBMID} (*Cell Broadcast Message Identifier for Data Download*), EF_{ECC} (*Emergency Call Codes*), EF_{CBMIR} (*Cell broadcast message identifier range selection*), EF_{DCK} (*De-personalization Control Keys*), EF_{CNL}(*Co-operative Network List*), EF_{NIA}(*Network's Indication of Alerting*), EF_{KcGPRS} (*GPRS Ciphering key KcGPRS*), $EF_{LOCIGPRS}$ (*GPRS location information*).

- EFs del nivel Telecom

  En este nivel ($DF_{TELECOM}$) se encuentran los siguientes archivos elementales: EF_{ADN} (*Abbreviated dialling numbers*), EF_{FDN} (*Fixed dialling numbers*), EF_{SMS} (*Short messages*), EF_{CCP} (*Capability configuration parameters*), EF_{MSISDN} (MSISDN), EF_{SMSP} (*Short message service parameters*) , EF_{SMSS} (*SMS status*) , EF_{LND} (*Last number dialled*), EF_{SDN} (*Service Dialling Numbers*), EF_{EXT1} (*Extension1*), EF_{EXT2} (*Extension2*), EF_{EXT3} (*Extension3*), EF_{BDN} (*Barred Dialling Numbers*), EF_{EXT4} (*Extension4*), EF_{SMSR} (*Short message status reports*).

1.2.7.2. CONDICIONES DE ACCESO A ARCHIVOS

Cada archivo tiene su propia condición de acceso. La condición de acceso relevante del último archivo seleccionado debe ser cumplida a cabalidad antes de que la acción pedida pueda tomar lugar. Los niveles de condiciones de acceso son definidos en la tabla 1-1.

Nivel	**Condición de Acceso**
0	ALWays
1	CHV1

2	CHV2
3	Reservado para uso futuro de GSM
4 a 14	ADM
15	NEVer

Tabla 1-1. Niveles de condiciones de acceso

Los significados de las condiciones de acceso a los archivos son los siguientes:

- **ALWays:** La acción puede ser ejecutada sin alguna restricción

- **CHV1** (Card Holder Verification 1) o PIN: La acción sólo será posible si una de las siguientes tres condiciones es cumplida a cabalidad:

  - Un correcto valor del PIN ya ha sido presentado a la SIM durante la sesión actual
  - El indicador CHV1 *enabled/disabled* está fijado a "*disabled*"
  - UNBLOCK CHV1 ha sido exitosamente ejecutado durante la sesión actual

- **CHV2:** La acción sólo será posible si una de las dos siguientes condiciones es cumplida a cabalidad:

  - Un correcto valor CHV2 ya ha sido presentado a la SIM durante la sesión actual
  - UNBLOCK CHV2 ha sido exitosamente ejecutado durante la sesión actual

- **ADM:** La asignación de estos niveles y los respectivos requerimientos para su cumplimiento son la responsabilidad de la apropiada autoridad administrativa (el operador). La definición de la condición de acceso ADM no excluye a la autoridad administrativa de usar ALW, CHV1, CHV2 y NEV si se requiere.

- **NEVer:** La acción no puede ser ejecutada sobre la interfaz SIM/ME. La SIM puede ejecutar la acción internamente.

1.2.8. Características de Seguridad

Los aspectos de seguridad de GSM son descritos en las referencias normativas GSM 02.09 [15] y GSM 03.20 [16]. Las características de seguridad soportadas por la SIM habilitan, entre otras, lo siguiente:

- Autenticación de la identidad del suscriptor ante la red
- Confidencialidad de los datos sobre la interfaz de radio
- Condiciones de acceso a archivos.

Muchas tarjetas SIM vienen provistas de un coprocesador que les permite cifrar información con claves DES (cifrado simétrico) o con claves RSA (cifrado asimétrico).

1.3. ESPECIFICACIONES DEL 3GPP

1.3.1. Generalidades

Existen centenares de estándares para las redes móviles, ver Figura 1-10. Muchos de ellos están sobre Internet y son proporcionados por organizaciones dedicadas a fijar y estructurar estándares. Afortunadamente la mayoría se puede descargar gratuitamente, pero para obtener otros se tiene que pagar una tarifa (el caso del ISO 7816).

Los estándares cubren todo desde el tamaño físico y características del chip hasta la forma en que maneja y almacena la información entrante. Desde hace unos años los estándares actualizados de Telefonía Celular en Europa dejaron de ser divulgados por la ETSI [9] y pasaron a ser divulgados por el 3GPP [17].

Figura 1-10. Estándares que gobiernan a la SIM

1.3.2. Estándares de uso común

Los estándares más importantes para desarrollar aplicaciones sobre la SIM son:

- 3GPP 11.14 [1]: *SIM Application Toolkit* (SAT), para la interfaz SIM - ME.

- 3GPP 03.48 [19]: Mecanismos de seguridad para SAT, escenario 2.

- 3GPP TS 03.19 [20]: SIM API para Java Card™. Integra los estándares 3GPP TS 11.11, 11.14 y 03.48 en un API para la plataforma JavaCard [2].

Los dos estándares que describen la plataforma UICC (la evolución de la SIM) son:

- ETSI TS 102.221 (análogo a GSM 11.11 o 3GPP 31.101) [8]: UICC - interfaz terminal; características físicas y lógicas. Este es un estándar básico que describe la comunicación de bajo nivel con una tarjeta UICC, la estructura de APDUs y la seguridad de la tarjeta inteligente.

- ETSI TS 102.222 [9]: comandos administrativos para aplicaciones de telecomunicaciones. Son definidos en este estándar los APDUs tales como CREATE FILE para la administración de la UICC. La idea es que estos comandos sean sólo usados por los fabricantes de las tarjetas.

Cuando se construye una aplicación para la tarjeta USIM, nos debemos remitir al:

- ETSI TS 102.223 (igual a 3GPP 31.111) [21]: *USIM Application Toolkit*.

Las aplicaciones para tarjetas SIM a menudo son diseñadas para trabajar con mensajería SMS (Servicio de Mensajes cortos).

Los estándares que cubren la codificación de los mensajes SMS que se entregan a los dispositivos móviles destino y la codificación de las instrucciones a la red GSM y al SMSC (Centro SMS) son:

- 3GPP 23.040 [22] ó 3GPP 03.40: realización Técnica de SMS

- 3GPP 24.011 [23]: soporte PP SMS sobre la interfaz de radio móvil

Para verificar la mayoría de los anteriores estándares, tan solo se necesita establecer una comunicación serial con el teléfono celular y a continuación enviarle comandos AT (para manipular el MODEM GSM), los cuales son descritos en el estándar:

- 3GPP 27.007 [24]: Conjunto de comandos AT para 3G UE

2. ESTÁNDARES PARA EL DESARROLLO DE APLICACIONES SOBRE MÓDULOS DE IDENTIFICACIÓN DE SUSCRIPTOR

2.1. SIM APPLICATION TOOLKIT

SAT (*SIM Application Toolkit*) nace en los inicios de los años noventa como una respuesta a la necesidad de los operadores de telefonía móvil de ofrecer más y mejores servicios con el objetivo de capturar nuevos suscriptores y mantener la lealtad de los actuales.

SAT se encuentra estandarizado por el comité T3 del 3GPP [17] y permite el desarrollo de aplicaciones que residen en el módulo SIM mediante un conjunto bien definido de acciones que se llevan a cabo bajo un modelo maestro-esclavo.

2.1.1. Generalidades

En agosto de 1994 dos ingenieros de BT Cellnet, Colin Hamling y Kristian Woodsend, tuvieron la idea de definir un conjunto de comandos de relativo bajo nivel de propósito general residentes en el ME, que permitan al módulo SIM ensamblarlos de formas diferentes para crear servicios específicos del operador [25]. Esta idea es ahora un estándar del 3GPP.

Puesto que la comunicación entre ME y módulo SIM se realiza mediante un modelo maestro-esclavo, para mantener esta relación, SAT agrega una nueva palabra de respuesta de estado que el módulo SIM envía al ME diciendo "El comando que me enviaste se ejecutó satisfactoriamente y tengo un comando para ti". De esta forma, el módulo SIM puede solicitar al ME la ejecución de comandos.

SAT se puede describir completamente mediante los mecanismos SIM Application Toolkit y las características SIM Application Toolkit.

2.1.2. Mecanismos SIM Application Toolkit

Dentro de las características SIM Application Toolkit nos encontramos con los mecanismos SAT. A continuación se listan los mecanismos SIM Application Toolkit y posteriormente se describe en detalle cada uno de ellos.

- *Profile Download.*
- *Proactive SIM.*
- *Data Download to SIM.*
- *Menu Selection.*
- *Call Control y MO SMS Control by SIM.*
- *Event Download.*
- *Security.*
- *Multiple Card.*
- *Time Expiration.*
- *Bearer Independent Protocol.*

2.1.2.1. *PROFILE DOWNLOAD*

Es el mecanismo que provee SIM Application Toolkit para que el módulo SIM pueda conocer las capacidades SAT que posee el ME.

Mediante este mecanismo el ME informa al módulo SIM sobre las capacidades SAT que soporta. Por otro lado, el ME puede saber las capacidades del modulo SIM a través del archivo elemental EF_{SST} y el archivo elemental EF_{PHASE}.

La importancia de este comando radica en que mediante él, el módulo SIM puede saber las capacidades SIM Application Toolkit del ME y por tanto limitar el rango de instrucciones de acuerdo a las capacidades declaradas.

La descripción detallada de la estructura, campos y significado de cada uno de los bits que constituyen la respuesta TERMINAL PROFILE enviada por el ME se encuentra especificada en TS11.14 [1] sección 5.2.

2.1.2.2. *PROACTIVE SIM*

La especificación TS11.11 [8] define que la comunicación entre el ME y el modulo SIM debe realizarse utilizando el protocolo de comunicación T=0, especificado en ISO/IEC 7816-3 [6]. Bajo este contexto, el ME actúa siempre como "maestro" y envía comandos al módulo SIM, por tanto, no hay un mecanismo para que el módulo SIM inicie una comunicación con el ME. Para solucionar esto, SIM Application Toolkit incorpora el servicio *Proactive SIM* que es un mecanismo para mantenerse dentro del protocolo T=0 y permitir al módulo SIM enviar comandos proactivos al ME para solicitar que éste realice una operación concreta.

Para que el ME pueda identificar un módulo SIM que soporte comandos proactivos, el módulo SIM debe tener activado el servicio *Proactive SIM* en su tabla se servicios SIM del archivo elemental EF_{SST}. Para que el módulo SIM pueda identificar a un ME que soporte *Proactive SIM*, el ME debe enviar esta información utilizando el comando TERMINAL PROFILE durante el procedimiento de inicialización especificado en el TS11.11 [8] sección 11.2.1.

En la respuesta que envía el ME a la solicitud de ejecución de comandos proactivos se incorpora una nueva palabra de respuesta de estado SW1. Esta respuesta de estado tiene el mismo significado de la palabra de terminación normal '90 00' pero que permite al módulo SIM decir al ME "tengo información para enviarle", entonces, el ME usa la función FETCH para buscar cuál es esta información.

Los comandos proactivos mas importantes definidos para SIM Application Toolkit según la especificación TS 11.14 [1] se listan a continuación ordenados alfabéticamente.

- *DISPLAY TEXT*: Mediante el cual se solicita al ME desplegar un texto o un icono en pantalla reemplazando el contenido actual de ella.

- *GET INKEY*: Mediante el cual se envía texto o iconos a la pantalla del ME y se espera una respuesta de entrada consistente en un simple carácter. Este tipo de comando es utilizado para mantener diálogos entre el usuario y la aplicación SIM Application Toolkit, mediante la selección de un ítem o acción determinada en un menú de opciones.

- *GET READER STATUS*: Mediante este comando es posible obtener información sobre el estado de lectores y módulos SIM adicionales que posea el ME.

- *LANGUAGE NOTIFICATION*: Mediante el cual el módulo SIM puede informar al ME del lenguaje utilizado en las cadenas de texto que están siendo enviadas por la aplicación SIM Application Toolkit.

- *LAUNCH BROWSER*: Mediante el cual se solicita al ME la ejecución de un navegador, si es que este ME soporta alguno, para que procese una URL determinada.

- *OPEN CHANNEL*: Mediante el cual se solicita al ME la apertura de un canal de comunicaciones con los parámetros especificados en el comando.

- *PERFORM CARD APDU*: Mediante el cual se solicita al ME el envío de una APDU hacia alguna tarjeta adicional que este disponga.

- *PLAY TONE*: Mediante el cual se solicita al ME la reproducción de tonos en alguno de los dispositivos de salida de sonido que este disponga.

- *POWER ON CARD*: Mediante el cual se inicia la sesión de comunicación entre el ME y algún módulo SIM adicional que este posea.

- *PROVIDE LOCAL INFORMATION*: Mediante el cual se solicita al ME que transfiera información local al módulo SIM, información como el MCC (*Mobile Country Code*), el MNC (*Mobile Network Code*), el LAC (*Location Area Code*), el CID (*Cell ID*) de la celda de servicio actual, el IMEI del ME y otros datos.

- *SEND DATA*: Mediante el cual se solicita al ME la transmisión de datos proporcionados por la tarjeta SIM a través de un canal de comunicación determinado.

- *SEND DTMF*: Mediante el cual se solicita al móvil el envío de tonos DTMF en una llamada establecida.

- *SEND SHORT MESSAGE*: Mediante el cual se puede enviar un mensaje SMS o un comando SMS a la red. Para este comando se definen dos tipos:

- Un mensaje SMS se envía a la red en un mensaje SMS-SUBMIT o un mensaje SMS-COMMAND donde los datos de usuario son pasados de forma transparente.

- Un mensaje SMS se envía a la red en un mensaje SMS-SUBMIT donde el texto necesariamente es empaquetado por el ME.

Cualquiera que sea el tipo, la cadena de texto resultante no debe exceder los 160 caracteres. La codificación de este texto debe estar acorde a la especificación TS23.038 [26].

- *SET UP CALL*: se usa para establecer una llamada.

- *SET UP MENU*: Utilizado cuando el módulo SIM brinda al ME una lista de ítems que deben ser incorporados a la estructura de menús que este maneja.

Las interacciones del ME con el módulo SIM a través de los comandos y el manejo de excepciones y flujos alternos en caso de fallo en la ejecución de algunos comandos se encuentran a cargo de la aplicación SIM Application Toolkit, sin embargo, la aplicación SAT necesita conocer el resultado del procesamiento de un comando y para esto se utilizan respuestas que se agrupan en 3 tipos principales:

- *OK*: Para indicar la ejecución exitosa de un comando.

- *Temporary problem*: Este resultado indica que la ejecución del comando ha fallado por problemas temporales pero que vale la pena intentarlo de nuevo.

- *Permanent problem*: Este resultado indica que la ejecución del comando ha fallado y que no debe intentarse nuevamente su ejecución durante la sesión actual.

2.1.2.3. *DATA DOWNLOAD TO SIM*

La descarga de datos al módulo SIM a través de SMS es un servicio que debe ser desplegado y activado por el operador de red y permite que los datos lleguen a las aplicaciones SIM Application Toolkit directamente desde la red de telefonía a través de SMS de dos formas:

- SMS-PP**:** Descarga de datos mediante SMS punto a punto, o sea, la estación base envía mensajes SMS solo a un ME determinado.

- *Cell Broadcast Data Download***:** Descarga de datos mediante mensajes SMS a todos los ME que se encuentran bajo la cobertura de una celda determinada.

Estos servicios como se mencionó anteriormente dependen en gran medida del operador, de que tan sofisticada es la red de telefonía móvil que este posee y es él quien se encarga del manejo y control de estos servicios mediante OTA.

2.1.2.4. *MENU SELECTION*

Cuando el usuario solicita una acción en alguno de los ítems de un menú en el ME, este debe informar al módulo SIM de tal evento y será la aplicación SIM Application Toolkit quien se encargará del procesamiento de esta entrada. La información y detalles sobre procedimientos y respuestas de este mecanismo se encuentran en la especificación TS11.14 [1] sección 8.

2.1.2.5. *CALL CONTROL Y MO SMS CONTROL BY SIM*

El control de llamada por la SIM es un servicio que debe ser desplegado y activado por el operador de red, mediante estos mecanismos se permite al módulo SIM el manejo de detalles de las llamadas que se realicen en el móvil. La especificación TS 11.14 [1] sección 9 define de forma detallada este procedimiento.

2.1.2.6. *EVENT DOWNLOAD*

El módulo SIM puede especificar al ME que tipo de eventos desea que el monitoree, así pues, cuando un evento de los especificados ocurra, el ME debe proveer esta información al módulo SIM intentando toda una serie de mecanismos y flujos alternos dependiendo de las condiciones del ME y de la interfaz ME-SIM. Los detalles están en la especificación TS11.14 [1] sección 11.

2.1.2.7. *SECURITY*

Las aplicaciones diseñadas utilizando las características SIM Application Toolkit requieren de métodos que aseguren confidencialidad de los datos, integridad de los datos y validación

del origen de los datos o un subconjunto de estos. Por tal razón, las especificaciones de seguridad se encuentran en la especificación del 3GPP TS03.48 [19].

2.1.2.8. *MULTIPLE CARD*

Este mecanismo se encuentra disponible únicamente si un ME tiene soporte para lectores y/o módulos SIM adicionales y si además manifiesta conformidad con SIM Application Toolkit clase "a". Un ME que manifiesta conformidad con SIM Application Toolkit clase "a" esta obligado a implementar los siguientes comandos proactivos:

- *GET READER STATUS*
- *PERFORM CARD APDU*
- *POWER ON CARD*
- *POWER OFF CARD*

2.1.2.9. *TIME EXPIRATION*

El módulo SIM puede iniciar un temporizador en el ME. Cuando un temporizador ha expirado, el ME debe notificar al módulo SIM este hecho utilizando el mecanismo TIME EXPIRATION. Los detalles se encuentran en la especificación TS11.14 [1] sección 10.

2.1.2.10. *BEARER INDEPENDENT PROTOCOL*

El conjunto de comandos proactivos *OPEN CHANNEL, CLOSE CHANNEL, SEND DATA, RECEIVE DATA y GET CHANNEL STATUS* permiten al módulo SIM establecer un canal de datos con el ME, y a través del ME a un servidor remoto en la red. El módulo SIM provee información para la selección de un portador (bearer) en el momento del establecimiento del canal. El ME por tanto permite que el módulo SIM y el servidor remoto puedan intercambiar información de una forma transparente sobre el canal de datos establecido.

2.1.3. Características de seguridad

Las características de seguridad para SIM Application Toolkit son encaminadas a brindar seguridad punto a punto en ambientes de aplicaciones con mensajería SMS (*Short Message Service*), ya que el SMSC (*Short Message Service Center*) o el Servidor de aplicaciones puede estar ubicado fuera de la seguridad de la red móvil, por ejemplo, puede estar en Internet.

Según la especificación Técnica del 3GPP TS 23.038 [26], la estructura de un SMS del tipo SMS_SUBMIT debe ser la mostrada en el figura 2-1.

Figura 2-1. Estructura de mensaje SMS_SUBMIT.

Para gestionar la seguridad del paquete se puede utilizar dentro del campo de cabeceras de usuario del mensaje SMS_SUBMIT, la cabecera de SIM Toolkit Security, la cual es codificada como un TLV (*Tag-Length-Value*) y cuya función es indicar que el *payload* contiene una descripción denominada *Command Header.* Este *Command Header* nos dice como se encuentra asegurado el resto del *payload.* La estructura de un SMS_SUBMIT asegurado se puede ver en la figura 2-2.

Figura 2-2. SMS_SUBMIT con SAT Security.

2.1.4. Gestión de Seguridad

Para gestionar la seguridad de las aplicaciones basadas en SMS desarrolladas con SAT, utilizamos los mecanismos de seguridad de la especificación técnica del 3GPP TS 03.48 [19], los apartes más importantes se describen a continuación.

2.1.4.1. AUTENTICACIÓN

La autenticación es la verificación de la identidad demandada por una entidad a otra entidad. El propósito de la autenticación es el de proteger las entidades y las aplicaciones contra el uso no autorizado. La autenticación asegura que solo las partes autorizadas puedan ejecutar acciones en el módulo SIM, y esto evita que partes no autorizadas tengan acceso a entidades de la red por medio de algún mecanismo SAT.

2.1.4.2. INTEGRIDAD DEL MENSAJE

La integridad del mensaje asegura que el contenido de ese mensaje no ha sido cambiado accidental o intencionalmente en su paso por una red determinada. Su propósito es detectar alguna corrupción del mensaje de aplicación o de todo el paquete asegurado.

2.1.4.3. DETECCIÓN DE REPETICIÓN E INTEGRIDAD DE SECUENCIA

La detección de repetición permite reconocer cuando un paquete asegurado recibido ha sido recibido previamente, con el propósito de evitar ataques de repetición y la duplicidad de paquetes.

2.1.4.4. ACUSE DE RECIBO Y PRUEBA DE EJECUCIÓN

Mediante el acuse de recibo y la prueba de ejecución se confirma que un paquete asegurado se ha recibido correctamente y se han ejecutado los chequeos de seguridad necesarios con el propósito de evitar la ambigüedad en dicho paquete. Esto permite la detección de paquetes no entregados debido a errores de red, corrupción del mensaje, validación fallida u otros problemas que pudieran ocurrir.

2.1.4.5. CONFIDENCIALIDAD DEL MENSAJE

Mediante la confidencialidad del mensaje se asegura que el mensaje intercambiado no se ha revelado a individuos, entidades o procesos sin autorización, para evitar que se extraiga información sensible de un paquete asegurado.

2.1.4.6. MECANISMOS DE SEGURIDAD Y COMBINACIONES RECOMENDADAS

- Mecanismos no-criptográficos: A continuación se presenta una lista de mecanismos de seguridad los cuales se basan en mecanismos no-criptográficos. Estos mecanismos no ofrecen seguridad contra algún ataque enviado, sólo contra detección de corrupción accidental.

    - Chequeo de redundancia
    - Reconocimiento inseguro
    - Contador simple.

- Mecanismos criptográficos: El encabezado de seguridad, excepto la suma de chequeo criptográfica o la firma digital, siempre será incluido en el cálculo de la suma de chequeo criptográfica o la firma digital.

- Suma de chequeo criptográfica (d1) o firma digital (d2): Este mecanismo de seguridad va dirigido a los siguientes requerimientos de seguridad: autenticación, integridad del mensaje, detección de repetición e integridad de secuencia.

- Reconocimiento como suma de chequeo criptográfica (f1) o firma digital (f2): Este mecanismo de seguridad satisface el requerimiento de seguridad de acuse de recibo.

- Cifrado de los datos de aplicación y partes posibles del encabezado de seguridad (g): El cifrado de los datos de aplicación y partes posibles del encabezado de seguridad corresponden al requerimiento de confidencialidad del mensaje.

2.2. JAVA CARD

Recientemente, Sun Microsystems Inc. publicó la definición de un nuevo miembro de las tecnologías Java, llamada Java Card, que está orientada a la programación de tarjetas inteligentes. Java Card fue diseñada de tal forma que ciertas construcciones de Java consideradas como demasiado complejas o no aplicables para la programación de tarjetas inteligentes no son incorporadas y por otro lado se agregan facilidades específicas para el manejo de transacciones con tarjetas inteligentes (atomicidad de un grupo de operaciones, objetos persistentes, etc.).

2.2.1. Generalidades

La tecnología Java Card adapta la plataforma Java para que pueda ser utilizada en tarjetas inteligentes y otros dispositivos cuyos ambientes de operación son altamente especializados y cuyas restricciones de memoria y capacidad de procesamiento son aun mayores que las presentes en los dispositivos J2ME (*Java 2 Micro Edition*).

La tecnología Java Card es la primera plataforma estándar abierta que ofrece una completa interoperabilidad entre las aplicaciones basadas en tarjetas inteligentes y se encuentra regulada y estandarizada por organismos como:

- JCF (*Java Card Forum*): Quien trabaja con el patrocinio de Sun Microsystems Inc. para definir la especificación de JavaCard.

- ETSI (*European Telecom Standards Industry*): Quien es responsable de definir y sostener los estándares de la telefonía móvil GSM y de implementar una interfaz común *Open OS* para las tarjetas SIM.

- Visa: Quien define el VOP (*Visa Open Platform*) para tarjetas inteligentes y servicios de pago.

2.2.1.1. ESPECIFICACION JAVA CARD

La especificación de la tecnología JavaCard, al momento de redacción de este documento, se encuentra en la versión 2.2 [2] y consta de tres partes principales:

- *Java Card Virtual Machine Specification*: Esta especificación define un subconjunto del lenguaje de programación Java y una máquina virtual para tarjetas inteligentes.

- *Java Card Runtime Environment Specification*: Esta especificación define el comportamiento en tiempo de ejecución de las tarjetas JavaCard.

- *Java Card API Specification*: Esta especificación define el framework, los paquetes de extensión y las clases Java para el desarrollo de aplicaciones para tarjetas inteligentes.

Una tarjeta inteligente que cumpla y soporte estas especificaciones es referida como una plataforma Java Card ya que en esta tarjeta pueden coexistir de forma segura múltiples aplicaciones desarrolladas por diferentes proveedores.

2.2.1.2. ELEMENTOS DE UNA APLICACIÓN JAVA CARD

Los principales elementos que componen a una aplicación Java Card se pueden apreciar en la figura 2-3.

- *Back-End Application*: Las aplicaciones Back-End proveen servicios que soportan a los applets internos de la tarjeta, por ejemplo, una aplicación Back-End puede

proveer conectividad a un sistema de seguridad que, junto con las credenciales internas en la tarjeta, proveen alta seguridad.

- *Host Application*: La aplicación Host es una aplicación que maneja las comunicaciones entre el usuario, el Java Card applet y la aplicación Back-End. Reside en un PC, un Terminal de pago electrónico, un teléfono celular o un subsistema de seguridad. Tradicionalmente, la aplicación Host es escrita utilizando lenguaje C, pero con el reciente desarrollo del API opcional de J2ME SATSA, esta aplicación puede ser escrita en lenguaje Java.

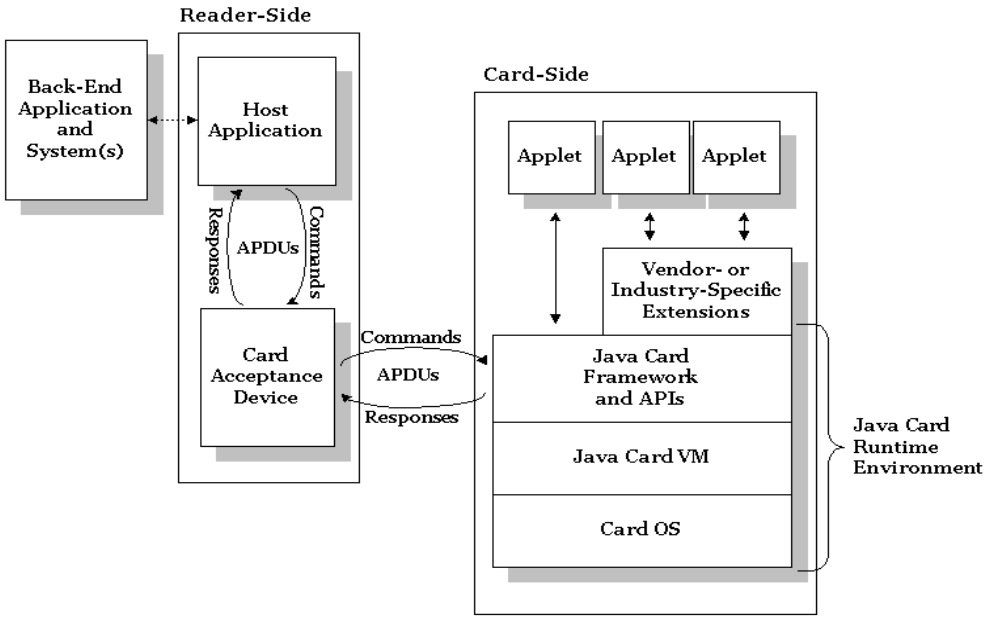

Figura 2-3. Arquitectura de una aplicación Java Card.

- *Card Acceptance Device*: El CAD (*Card Acceptance Device*) es un dispositivo de interfaz que se ubica entre la aplicación Host y la tarjeta Java Card. El CAD se encarga de proveer de energía a la tarjeta. El CAD puede ser un dispositivo lector conectado a un PC utilizando algún tipo de comunicación serial, una Terminal de pago electrónico o la interfaz para tarjetas SIM de los teléfonos Celulares. El CAD es el encargado de enviar APDU's del tipo *Command* desde la aplicación Host hacia la tarjeta Java Card y enviar las respuestas desde la tarjeta Java Card hacia la aplicación Host.

- *Java Card Applets Environment*: Como puede verse en la figura 2-3, Java Card es una plataforma multi-aplicación, uno o más applets Java Card pueden residir en la

tarjeta junto con el software de soporte, el sistema operativo de la tarjeta y JCRE (*Java Card Runtime Environment*).

2.2.1.3. COMUNICACIÓN CON UN APPLET JAVA CARD

Existen dos modelos para la comunicación con una tarjeta Java Card.

- Modelo de paso de mensajes: Es la base de todas las comunicaciones Java Card y se basa en el intercambio de APDU's entre el CAD y el *Java Card Framework*. El *Java Card Framework* recibe una APDU entrante enviada por el CAD y la reenvía al applet Java Card apropiado. El applet procesa la APDU entrante y retorna una respuesta (*Response APDU*). Estas APDUs están definidas por el ISO/IEC 7816-4[6].

- Modelo JCRMI: El modelo JCRMI se apoya en un subconjunto del modelo de objetos distribuidos de J2SE, RMI (*Remote Method Invocation*).

2.2.2. Arquitectura Java Card

Los componentes principales de la arquitectura Java Card se muestran en la figura 2-4.

- *Card OS*: controla el acceso físico a la memoria de la tarjeta, chequea la integridad de los objetos constituyentes del applet, se ocupa de los detalles de I/O a nivel de transporte con el protocolo T=0, integra elementos básicos de cifrado de información y de la detección de intentos de sabotaje mediante sensores hardware.

- *Java Card VM*: Similar a otras versiones de Java, la máquina virtual de Java Card interpreta el código y es responsable de la ejecución de la aplicación. La Java Card VM stá dividida en dos partes, una dentro de la tarjeta y otra parte fuera de ella.

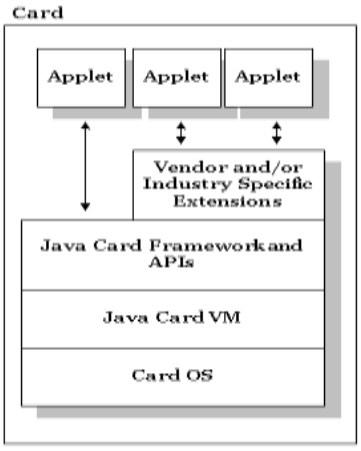

Figura 2-4. Arquitectura Java Card

- *Java Card Framework y APIs*: Estas APIs proporcionan un conjunto de objetos y métodos predefinidos y estandarizados, los que simplemente podemos reutilizar.

- *JavaCard Runtime Environment* (JCRE): El JCRE sustenta y administra todos los datos y servicios globales en la tarjeta y maneja el ciclo de vida del a*pplet*.

- *Applet*: El *applet* es la aplicación JavaCard almacenada en la SIM como un conjunto de archivos *bytecodes* generados a partir de la compilación del código fuente donde se resuelve la lógica de la aplicación.

2.2.3. Java Card Virtual Machine JCVM

Los principales servicios que presta la máquina virtual son:

- Gestión de la seguridad
  - Aislamiento entre los applets utilizando mecanismos de *firewall*.
  - Análisis de la sintaxis, los *bytecodes* incorrectos son eliminados.

- Gestión de objetos
  - Posicionamiento de los objetos (ubicación en memoria / liberación de memoria de los objetos).
  - Chequear y autorizar el acceso o referencia a los objetos de una aplicación.

- Independencia del hardware.

La máquina virtual para la plataforma Java Card esta implementada en dos partes, una parte que es externa a la tarjeta y otra parte que corre dentro de la tarjeta. La parte dentro de la tarjeta es quien se encarga de interpretar los *bytecodes*, gestión de objetos y clases, etc. La máquina virtual fuera de la tarjeta es una herramienta de desarrollo, comúnmente definida como *Java Card Converter tool*, quien se encarga de cargar, verificar y luego dejar listas las clases en un applet para su posterior ejecución dentro de la tarjeta. La herramienta *Java Card Converter tool* verifica que las clases que constituyen el *applet* sigan las especificaciones Java Card y su salida es un archivo CAP (*Converted Applet*), que contiene todas las clases Java con una representación de paquete binario cargable y ejecutable.

La Java Card VM no para cuando se interrumpe la alimentación puesto que su estado es almacenado en la memoria no volátil de la tarjeta. Arrancando, la JCVM inicializa el JCRE y crea todos los objetos del *framework* del JCRE, los cuales viven a lo largo del ciclo de vida de la JCVM. Luego que la JCVM se ha inicializado, todas las interacciones con la tarjeta son en principio controladas por un *applet* en la tarjeta. Cuando la alimentación es removida, cualquier dato contenido en la RAM de la tarjeta se pierde, pero cualquier estado almacenado en la memoria no volátil de la tarjeta se mantendrá. Cuando la potencia es aplicada nuevamente la máquina virtual se activa, momento en el cual el estado de la máquina virtual, los objetos son realmacenados y la ejecución reinicia esperando por futuras entradas.

El rol de la JCVM es el mejor entendimiento en el contexto del proceso de producción y despliegue del software para la plataforma Java Card. Hay varios componentes que constituyen un sistema Java Card, incluyendo la JCVM, el *Converter* para la plataforma Java Card (*Java Card Converter*), una herramienta de instalación terminal, y un programa de instalación que corre sobre el dispositivo, como se muestra en la figura 2-5 y figura 2-6.

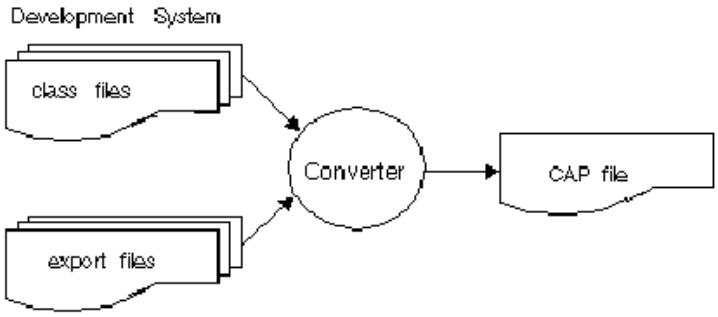

Figura 2-5. Conversión del paquete Java Card API

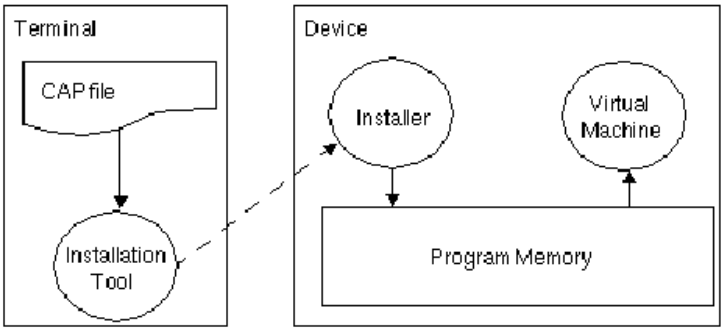

Figura 2-6. Instalación del paquete Java Card API

El desarrollo de un applet Java Card es como el de cualquier otro programa Java: un desarrollador escribe una o más clases Java, y compila el código fuente con un compilador

Java, produciendo uno o más archivos *class*. El *applet* se corre, prueba y se depura sobre una estación de trabajo usando herramientas de simulación para emular el ambiente del dispositivo. Entonces, cuando un *applet* está listo para ser descargado a un dispositivo, el archivo *class* que comprende el applet es convertido a un archivo CAP usando la herramienta *Java Card Converter*.

Después de la conversión, el archivo CAP es copiado a un Terminal de tarjeta, tal como un computador de escritorio con un periférico lector de tarjetas. Entonces una herramienta de instalación sobre el Terminal carga el archivo CAP y lo transmite al dispositivo habilitado con tecnología Java Card. Un programa de instalación sobre el dispositivo recibe el contenido del archivo CAP y prepara al applet para ser corrido por la JCVM.

2.2.4. Java Card Runtime Environment JCRE

El JCRE (*Java Card Runtime Environment*) como se mencionó anteriormente sustenta y administra todos los datos y los servicios globales de la tarjeta, su composición se puede ver en la figura 2-7.

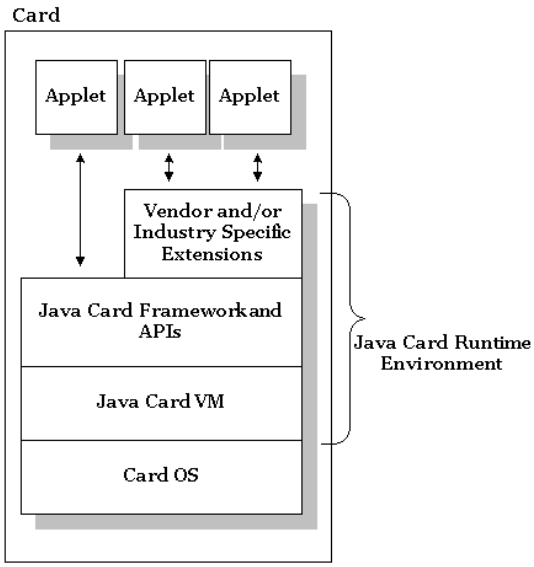

Figura 2-7. Componentes del Java Card *Runtime Environment*

Las funciones más relevantes del JCRE son:

- Manejo de datos como el PIN.
- Mantenimiento del ciclo de vida de la tarjeta y el applet

- Gestionar y supervisar tareas administrativas alrededor del ciclo de vida de un applet (instalación, registro, selección, etc.).
- Despachar las instrucciones que sean enviadas en los comandos APDU al applet correspondiente.
- Manejar todas las funciones necesarias para garantizar la carga segura de nuevos applets.

2.2.5. Java Card API

El API de Java Card ofrece a los desarrolladores un conjunto de objetos y métodos predefinidos y estandarizados, los que simplemente se pueden reutilizar. EL API es a su vez un elemento de seguridad puesto que solamente se puede realizar la invocación de objetos y métodos en un applet a través del API.

La especificación del API Java Card define un pequeño subconjunto del tradicional API J2SE, pero además de este pequeño subconjunto del API tradicional, el Java Card Framework define un conjunto de clases específicas para el soporte de aplicaciones Java Card y dichas clases se agrupan en los siguientes paquetes [2]:

- java.lang.
- javacard.framework.
- javacard.security.
- java.rmi.
- javacardx.crypto.

2.2.6. Ciclo de vida de las aplicaciones Java Card

Cada applet Java Card que se encuentra dentro de una tarjeta Java Card se identifica inequívocamente mediante un AID (Application ID). Un AID, como se define en el estándar ISO 7816-5 [6], es una secuencia de 5 a 16 bytes. Todos los applets deben heredar de la clase abstracta javacard.framework.Applet que define los métodos utilizados por el JCRE para controlar el ciclo de vida del applet como se muestra en la figura 2-8.

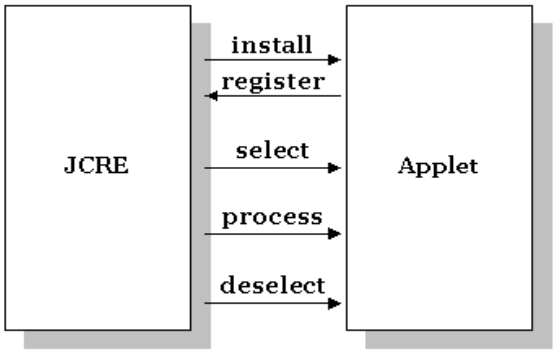

Figura 2-8. Métodos del ciclo de vida del applet Java Card.

El ciclo de vida del applet inicia cuando el applet es descargado a la tarjeta y el JCRE invoca el método estático del applet *Applet.install()* y el applet se registra a el mismo ante el JCRE invocando el método *Applet.register()*. Una vez el applet se ha instalado y registrado se encuentra en el estado deseleccionado, o dicho de otra forma, listo para ser seleccionado y realizar el procesamiento de APDU's como se muestra en la figura 2-9.

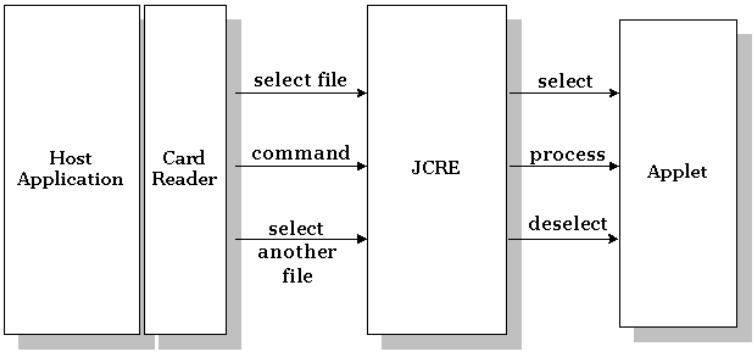

Figura 2-9. Uso de los métodos del applet Java Card.

2.3. J2ME Y SATSA

La especificación de SATSA (*Security and Trust Services API*) define paquetes opcionales para la plataforma J2ME. Esta especificación ha sido producida en respuesta al JSR-177 (*Java Specification Request* 177) [3]. El propósito de la JSR-177 es especificar una colección de API's que proporcionen servicios de confianza y seguridad por medio de la integración de un Elemento de Seguridad denominado SE (*Security Element*). Un SE es un componente en un dispositivo J2ME, que provee los siguientes beneficios:

- Almacenamiento seguro de datos sensitivos protegidos, tales como llaves privadas de usuario, certificados de claves públicas, credenciales, información personal y otros.

- Operaciones criptográficas que soportan protocolos de pagos, integridad de datos, y confidencialidad de datos.

- Un ambiente seguro de ejecución para desplegar características de seguridad acostumbradas. Las aplicaciones J2ME podrían basarse sobre estas características para tomar muchos servicios de valor agregado, tales como identificación y autenticación de usuarios, banca, pagos, aplicaciones de lealtad y otras.

Un SE puede presentarse en una variedad de formas. Las tarjetas inteligentes son las mas usadas comunmente para implementar un SE. Puesto que están ampliamente desplegadas en los teléfonos inalámbricos, tales como las tarjetas SIM en teléfonos GSM, tarjetas UICC en teléfonos 3G, y tarjetas RUIM en teléfonos CDMA. Alternativamente, un SE puede estar totalmente implementado en software. Esta especificación no excluye cualquiera de las posibles implementaciones de un SE, sin embargo algunos de los paquetes son optimizados para las implementaciones en tarjetas inteligentes.

2.3.1 Alcance de SATSA

Los SE pueden tener diversas características software y hardware. En vez de dirigirse a cada posible tipo de SE y sus capacidades, la especificación JSR 177 [3] considera las funciones del API basadas en los siguientes criterios:

- Tamaño de requerimientos para recursos obligados de dispositivos de clientes.
- Amplitud de uso del SE.
- Flexibilidad y extensibilidad del API.

Basados en estos criterios, esta especificación define un API que proporciona las siguientes capacidades, con base a las diferentes necesidades que las aplicaciones J2ME pueden tener cuando interactúan con un SE.

- Comunicación con la tarjeta inteligente: Las tarjetas inteligentes proporcionan un ambiente programable seguro. Ellas son los mas ampliamente desplegados SE que entregan un amplio rango de servicios de seguridad y confianza. Estos servicios

pueden ser continuamente actualizados con nuevas o mejoradas aplicaciones que pueden ser instaladas sobre una tarjeta inteligente. Dos métodos de acceso están definidos en esta especificación basados en el modelo de paso de mensajes y el modelo JCRMI. Estos métodos de acceso permiten que una aplicación J2ME se comunique con una tarjeta inteligente apalancando los servicios de seguridad desplegados sobre ella.

- Servicio de firma digital y gestión de credenciales de usuario básico: El servicio de firma digital permite a una aplicación J2ME generar firmas digitales que conforman el formato CMS (*Cryptographic Message Syntax*). Las firmas digitales son usadas para autenticar usuarios finales o transacciones autorizadas usando llaves públicas de cifrado. La identidad del usuario está usualmente ligada a la llave pública a través de un certificado de llave pública. La gestión de credenciales de usuario le permite a aplicaciones J2ME manejar credenciales de usuario, tales como certificados, sobre una representación de usuario.

  Para cada uso, la complejidad de generar una firma digital formateada es reducida a través del diseño de una interfaz de alto nivel. La implementación es responsable por pedir las operaciones criptográficas requeridas, así como también, es responsable de ejecutar el formateo apropiado de los resultados.

  El servicio de firma digital y gestión de credenciales relevado sobre un SE, proporciona un almacenamiento seguro para credenciales de usuario y llaves criptográficas, así como también ejecuta una segura computación que envuelve las llaves criptográficas que son seguramente almacenadas sobre el SE.

- Librerías criptográficas de propósito general: Las librerías criptográficas proporcionan un subconjunto del API de criptografía de la plataforma J2SE 1.4.2. Esta soporta operaciones criptográficas básicas, tales como Hash o Digest de mensajes, verificación de firma digital, cifrado, y descifrado. Las operaciones criptográficas le permiten a una aplicación J2ME proporcionar comunicación segura de datos, protección de datos, y gestión de contenido.

2.3.2 Visión global del API

El API está definido en cuatro paquetes opcionales que pueden ser implementados independientemente, estos paquetes son:

- SATSA-APDU: Define un API que soporta comunicación con aplicaciones de tarjetas inteligentes que usen el protocolo APDU.

- SATSA-JCRMI: Define un API de cliente JCRMI que le permite a una aplicación J2ME comunicarse con la tarjeta inteligente utilizando la invocación de métodos de un objeto Java Card remoto.

- SATSA-PKI: Define un API que soporta aplicaciones de nivel de firma digital, firma y gestión de credenciales de usuario básicas. Para habilitar la reutilización este API es independiente del tipo de SE que sea utilizado por el dispositivo J2ME.

- SATSA-CRYPTO: Define un subconjunto del API criptográfico J2SE. Este proporciona operaciones criptográficas básicas que soportan asimilación de mensajes, verificación de firma, cifrado y descifrado.

3. DESCRIPCIÓN DE LA PLATAFORMA

3.1 METODOLOGÍA DE DESARROLLO

Las tecnologías involucradas en la plataforma son J2SE, J2ME y Java Card. Todas ellas orientadas a objetos, por lo cual la metodología de desarrollo escogida fue el MCS [27] (Modelo para construcción de soluciones). La figura 3-1 establece las fases de referencia del MCS.

Figura 3-1. Fases de referencia del MCS

La plataforma consta de dos partes, una parte Host alojada en el dispositivo móvil y otra parte, el Applet, alojado en la tarjeta SIM.

3.1.1. La aplicación Host

El modelo utilizado para la parte Host es el MVC (Modelo-Vista-Control). Como característica particular se tiene que la plataforma como tal no posee interfaces gráficas de usuario.

3.1.2. El Applet

Para desarrollar un Applet se deben seguir en general los siguientes pasos:

1. Especificar (narrar) las funciones del applet.
2. Solicitar y asignar AIDs al applet y al paquete que contiene las clases del Applet.
3. Diseñar la estructura de clases y métodos del programa (Diagrama de clases, de secuencia, …).
4. Definir la interface (conjunto de APDUs) entre el applet y la aplicación host.

3.2. ANÁLISIS

La plataforma que finalmente se concibe permite:

- Un acceso seguro a servicios móviles basado en parámetros SIM.
- Cifrar o descifrar información tanto con algoritmos simétricos como con asimétricos.
- Gestionar (fijar, recalcular u obtener) de forma segura en la SIM claves simétricas y asimétricas.
- Manejo de funciones Hash, con el objetivo de generar y verificar firmas digitales.

Se ha denominado a la plataforma como "**P3SIM**" acrónimo de "Plataforma de Seguridad para Servicios móviles basada en SIM".

3.2.1. Diagrama de casos de uso

El actor denominado "Servicio" (ver figura 3-2) corresponde al servicio o aplicación(es) que haciendo uso de la plataforma brindará el nivel de seguridad apropiado para cada usuario.

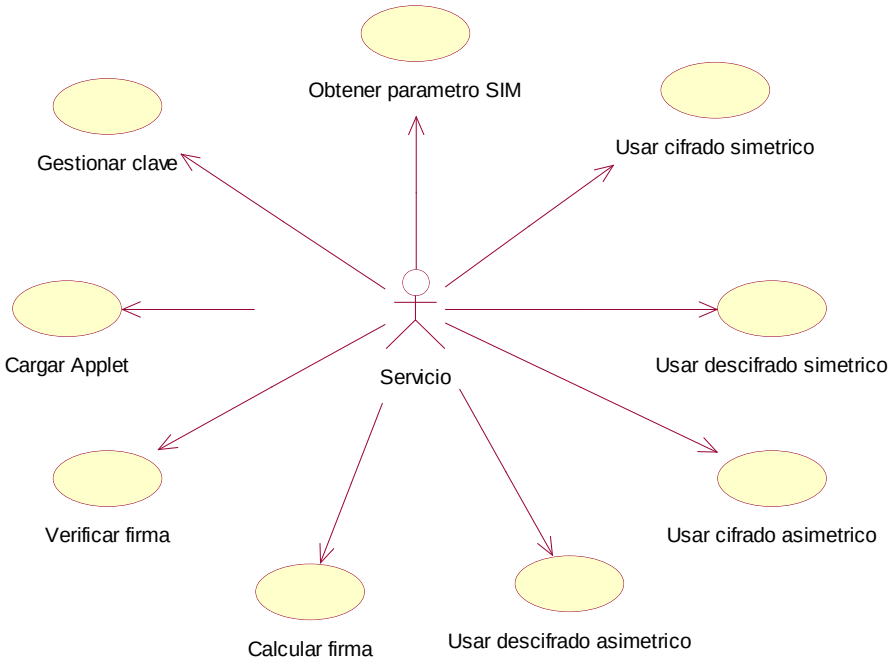

Figura 3-2. Diagrama de casos de uso de la plataforma

3.2.2. Descripción de los casos de uso

Información General	
Caso de uso:	**Cargar AppletSIM**
Actores:	Servicio
Propósito:	Permitir que el servicio instale un Applet en la tarjeta SIM.
Resumen:	El servicio le pasa a la plataforma un archivo que representa un Applet el cual será cargado en la SIM. Una vez se carga el applet, se procede a inicializarlo.
Tipo:	Primario.
Referencias cruzadas:	No hay.
Precondiciones	
- El archivo contiene APDUs que deben cumplir con el estándar GSM 03.48.	

Flujo Principal
– El servicio le indica a la plataforma cual es el archivo que contiene los APDUs para que

posteriormente la plataforma le envíe las APDUs a la SIM.
- Luego la plataforma trata de seleccionar el Applet.
- Si la selección es exitosa, entonces la plataforma le envía al servicio el mensaje de instalación exitosa.

Flujos de Excepción

E1: El Applet no pudo ser seleccionado.
- La plataforma tuvo algún problema al cargar al Applet en la SIM y por eso el applet no puede ser seleccionado.
- La plataforma le envía al servicio el respectivo mensaje de error.

Información General	
Caso de uso:	**Obtener parámetro SIM**
Actores:	Servicio
Propósito:	Permitir que el servicio obtenga algún parámetro SIM.
Resumen:	La plataforma le envía al Applet en la SIM un command APDU que sirve para obtener el parámetro que especificó el servicio, luego la SIM le retorna el parámetro SIM a la plataforma.
Tipo:	Primario.
Referencias cruzadas:	No hay.

Precondiciones

- El applet debe estar instalado en la tarjeta SIM.

Flujo Principal
- El servicio le especifica la plataforma que parámetro SIM quiere obtener.
- La plataforma le pide el parámetro específico al Applet en la SIM
- La SIM le retorna el parámetro a la plataforma.
- La plataforma le envía el parámetro al servicio.

Flujos de Excepción

E1: El parámetro no existe.
- Debido a que algunos parámetros de la SIM puede estar opcionalmente en las tarjetas reales, se puede presentar el caso se que el parámetro no exista.
- La plataforma le envía al servicio el respectivo mensaje de error.

Información General	
Caso de uso:	**Gestionar clave**
Actores:	Servicio
Propósito:	Permitir que el servicio almacene, actualice u obtenga una clave en la tarjeta SIM.
Resumen:	El servicio puede crear una clave, obtenerla, actualizarla o almacenarla. Todo ello la plataforma lo realiza sobre la tarjeta SIM.
Tipo:	Primario.
Referencias cruzadas:	No hay.
Precondiciones	
- El applet debe estar instalado en la tarjeta SIM.	

Flujo Principal

- El servicio especifica lo que quiere hacer: generar, obtener o almacenar una clave. Subflujos S1, S2 ó S3.
- Se le informa al servicio del éxito o fracaso de la operación.

Subflujos

S1: Generar una clave

- La plataforma le ordena a la tarjeta SIM que genere el tipo de clave que el servicio especifica.
- Luego esa clave se almacena en la SIM.

S2: Obtener una clave

- El servicio le informa a la plataforma qué clave en particular quiere obtener.
- La plataforma obtiene la clave desde la tarjeta SIM y se la retorna al servicio.

S3: Almacenar una clave.

- El servicio le envía a la plataforma una nueva clave, para que sea almacenada en la tarjeta SIM. El servicio debe informar las características de la clave que ha enviado: si es simétrica o asimétrica (publica o privada), el algoritmo de cifrado para el que es utilizada.

Flujos de Excepción
E1: La Clave no es una clave válida.
– La plataforma no soporta el tipo de clave especificada por el servicio.
– No se almacena ni se actualiza ninguna clave

Información General	
Caso de uso:	**Verificar firma**
Actores:	Servicio
Propósito:	Garantizar la integridad de la información y el no repudio.
Resumen:	El servicio obtiene una información con su respectiva firma digital. Este caso de uso permite verificar la validez de dicha firma digital con respecto a la información recibida.
Tipo:	Primario.
Referencias cruzadas:	- Gestionar claves - Usar descifrado asimétrico
Precondiciones	
– El applet debe estar instalado en la tarjeta SIM.	

Flujo Principal
– El servicio le pasa a la plataforma tres parámetros: la información recibida, la firma digital y la clave pública de la entidad que envió la información.
– La plataforma descifra con la clave pública la firma digital, obteniendo un hash.
– Luego calcula el hash de la información recibida.
– Se comparan los dos hash.
– Si los hash son iguales, se le informa al servicio de la validez de la firma. Si no son iguales se le informa lo contrario.

Flujos de Excepción
E1: No se puede descifrar la firma digital.
– La plataforma lanza una excepción debido a tres posibilidades: la firma no es válida, la clave pública no es válida, el algoritmo con que se obtuvo la firma no es soportado.

Información General	
Caso de uso:	**Calcular firma**
Actores:	Servicio
Propósito:	Este caso de uso permite calcular la firma digital de cualquier información.
Resumen:	El servicio debe especificar los datos de entrada (información) y el tipo de algoritmo usado para obtener la firma. Internamente la plataforma también hará uso de las claves asimétricas del usuario.
Tipo:	Primario.
Referencias cruzadas:	Gestionar claves.

Precondiciones
- El applet debe estar instalado en la tarjeta SIM. - Deben estar almacenadas la clave pública y privada del usuario.

Flujo Principal
– La plataforma le calcula el hash a la información proporcionada por el servicio. – La plataforma le envía a la SIM el hash para que la tarjeta lo cifre con la clave privada del usuario. La SIM retorna a la plataforma el hash cifrado. – La plataforma obtiene la clave pública del usuario, la cual está almacenada en la SIM. – La plataforma le retorna al servicio el hash cifrado y la clave pública del usuario.

Flujos de Excepción
E1: El algoritmo para generar la firma no está soportado. – Se le informa al servicio, del fracaso al generar la firma con dicho algoritmo.

Información General	
Caso de uso:	**Usar descifrado asimétrico**
Actores:	Servicio
Propósito:	Permitirle conocer al servicio si la información que le llegó no ha sido modificada. Dependiendo de si la cifraron con una llave pública o una privada permite garantizar que el usuario es el destino real o que la entidad que lo envió es realmente quien dice ser.
Resumen:	El servicio le envía a la plataforma la información cifrada, el

	algoritmo utilizado y con que clave asimétrica quiere que la descifre.
Tipo:	Primario.
Referencias cruzadas:	Gestionar claves.

Precondiciones

- El applet debe estar instalado en la tarjeta SIM.
- Debe estar almacenada la clave privada del usuario, en el caso correspondiente.

Flujo Principal

- El servicio le envía a la plataforma la información cifrada, el nombre del algoritmo y como quiere que la descifre. Subflujos S1ó S2.
- La plataforma le retorna al servicio la información descifrada.

Subflujos

S1: Descifrar con la clave privada del usuario

- La plataforma le ordena a la tarjeta SIM que descifre una información con la clave privada del usuario.
- Luego la SIM le retorna la información descifrada a la plataforma.

S2: Descifrar con la clave pública de una entidad conocida

- La plataforma almacena la clave pública de la entidad conocida en la tarjeta SIM.
- La plataforma le ordena a la tarjeta SIM que descifre una información con la clave pública de la entidad.
- Luego la SIM le retorna la información descifrada a la plataforma.

Flujos de Excepción

E1: No se puede descifrar la información.

- La plataforma lanza una excepción debido a tres posibilidades: la información ha sido modificada, la clave no es válida, el algoritmo no es soportado

Información General

Caso de uso:	**Usar cifrado asimétrico**
Actores:	Servicio
Propósito:	Le permite al servicio cifrar una información con la clave pública o

	privada del usuario, o cifrar con la clave pública de una entidad.
Resumen:	El servicio le envía a la plataforma una información para que la cifre con alguna de las claves asimétricas almacenadas en la SIM.
Tipo:	Primario.
Referencias cruzadas:	Gestionar clave.

Precondiciones

- El applet debe estar instalado en la tarjeta SIM.
- Debe estar almacenada la clave asimétrica correspondiente.

Flujo Principal

- El servicio le envía a la plataforma la información a cifrar, el nombre del algoritmo y con que clave quiere que la cifre.
- La tarjeta SIM cifra la información con el algoritmo y la clave especificada. La SIM le retorna a la plataforma la información cifrada.
- La plataforma le retorna al servicio la información cifrada.

Flujos de Excepción

E1: No se soporta el algoritmo.

- La plataforma lanza una excepción debido a que no puede cifrar con el algoritmo especificado.

Información General

Caso de uso:	**Usar cifrado simétrico**
Actores:	Servicio
Propósito:	Permitirle al servicio cifrar una información con una clave simétrica. Así se logra garantizar la integridad y confidencialidad de la información.
Resumen:	El servicio le envía a la plataforma la información a cifrar, el algoritmo utilizado y con que clave simétrica quiere que la cifre.
Tipo:	Primario.
Referencias cruzadas:	Gestionar claves.

Precondiciones

- El applet debe estar instalado en la tarjeta SIM.

- Debe estar almacenada la clave simétrica correspondiente.

Flujo Principal
– El servicio le envía a la plataforma la información, el nombre del algoritmo y con que clave quiere que la cifre. – La plataforma obtiene de la tarjeta SIM la respectiva clave. Subflujos S1ó S2. – La plataforma cifra la información con el respectivo algoritmo. – La plataforma le envía al servicio la información cifrada.

Subflujos
S1: Obtener la clave simétrica del usuario – La plataforma le pide a la tarjeta SIM la clave simétrica del usuario. – La SIM le envía la clave a la plataforma. *S2: Obtener la otra clave simétrica* – La plataforma le pide a la tarjeta SIM la otra clave simétrica. – La SIM le envía la clave a la plataforma.

Flujos de Excepción
E1: No se puede cifrar la información. – La plataforma lanza una excepción debido a que el algoritmo no es soportado

Información General	
Caso de uso:	**Usar descifrado simétrico**
Actores:	Servicio
Propósito:	Permitirle conocer al servicio si la información que le llegó no ha sido modificada, así se garantiza la integridad y confidencialidad de la información.
Resumen:	El servicio le envía a la plataforma la información cifrada, el algoritmo utilizado y con que clave simétrica quiere que la descifre.
Tipo:	Primario.
Referencias cruzadas:	Gestionar claves.

Precondiciones
- El applet debe estar instalado en la tarjeta SIM. - Debe estar almacenada la clave simétrica correspondiente.

Flujo Principal
– El servicio le envía a la plataforma la información cifrada, el nombre del algoritmo y con que clave quiere que la descifre. – La plataforma obtiene de la tarjeta SIM la respectiva clave. Subflujos S1ó S2. – La plataforma descifra la información cifrada con el respectivo algoritmo. – La plataforma le envía al servicio la información descifrada.

Subflujos
S1: Obtener la clave simétrica del usuario – La plataforma le pide a la tarjeta SIM la clave simétrica del usuario. – La SIM le envía la clave a la plataforma. *S2: Obtener la otra clave simétrica* – La plataforma le pide a la tarjeta SIM la otra clave simétrica. – La SIM le envía la clave a la plataforma.

Flujos de Excepción
E1: No se puede descifrar la información. – La plataforma lanza una excepción debido a que el algoritmo no es soportado

3.3. DISEÑO

3.3.3. Diagramas de secuencia

A continuación se muestra el diagrama de secuencia de los casos de uso Obtener parámetro SIM y Verificar firma.

Caso de uso: Obtener parámetro SIM

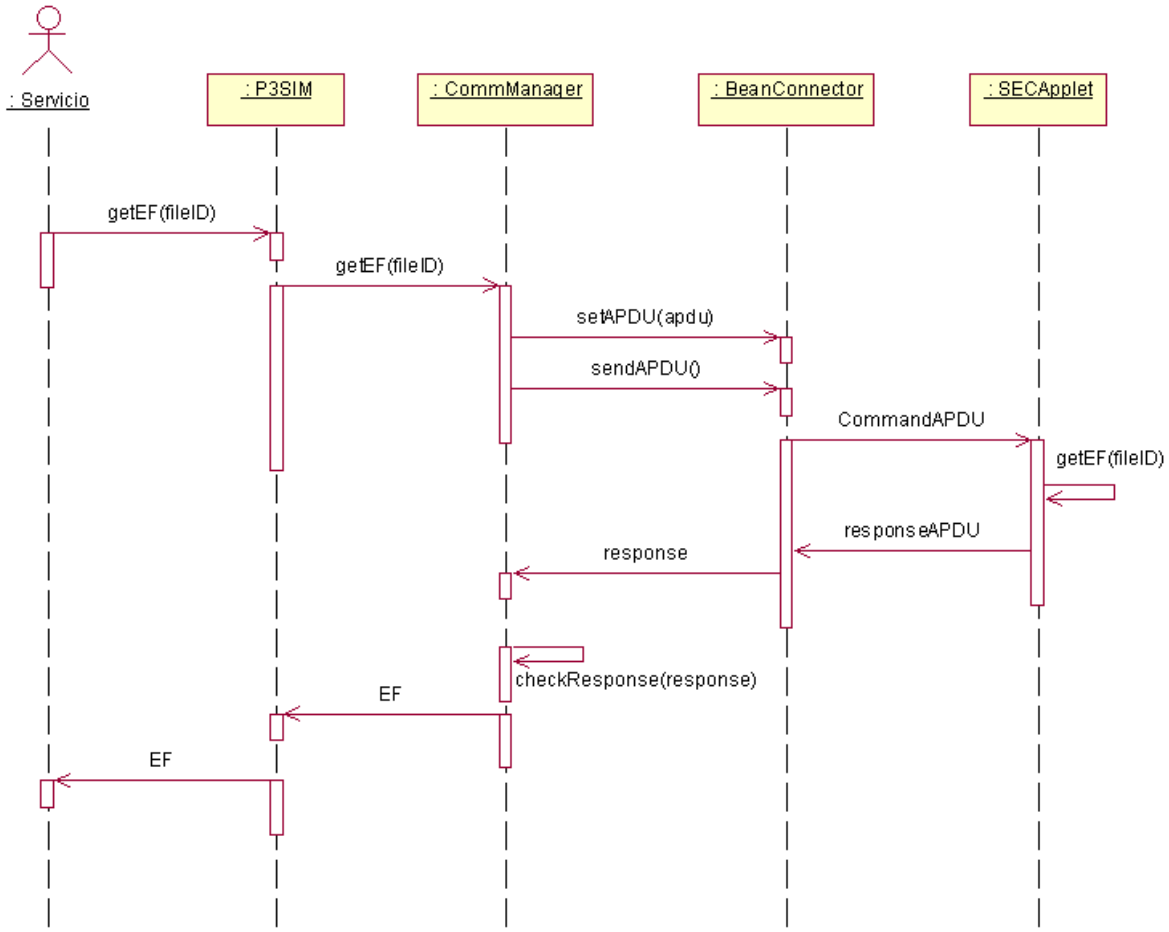

Figura 3-3. Diagrama de secuencia para Obtener parámetro SIM

Caso de uso: Verificar firma

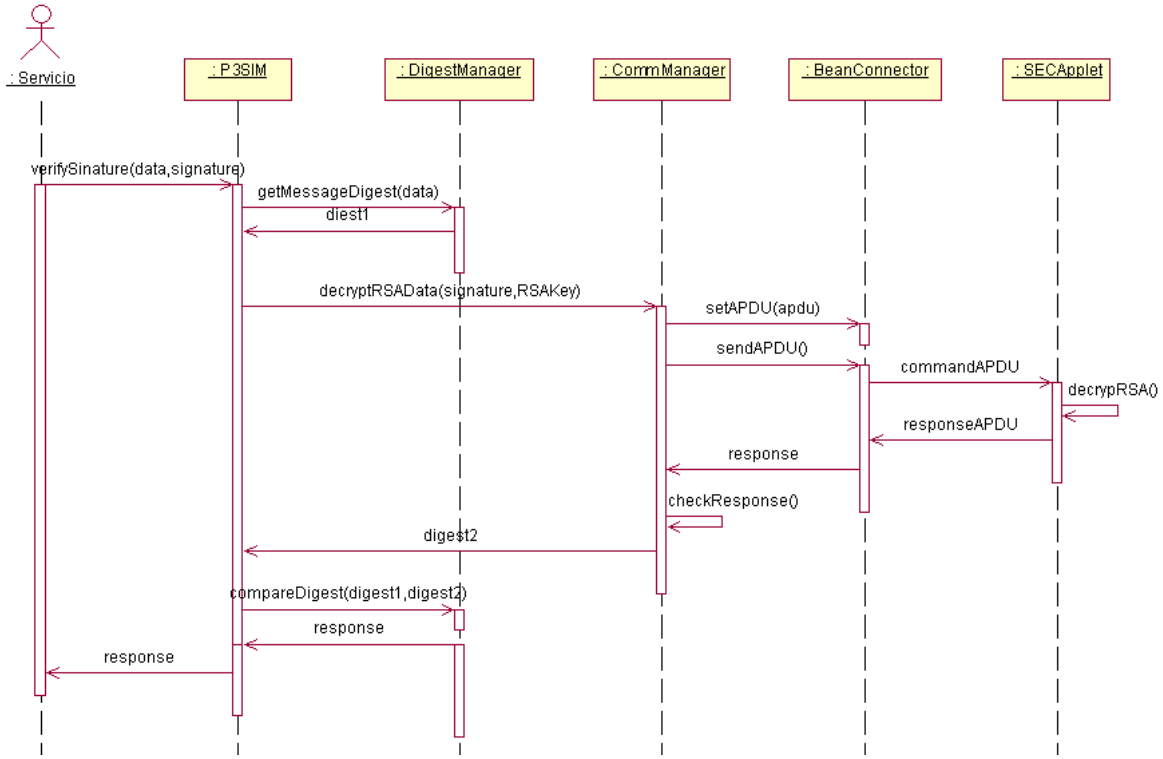

Figura 3-4. Diagrama de secuencia para verificar firma

3.3.4. Diagrama de clases

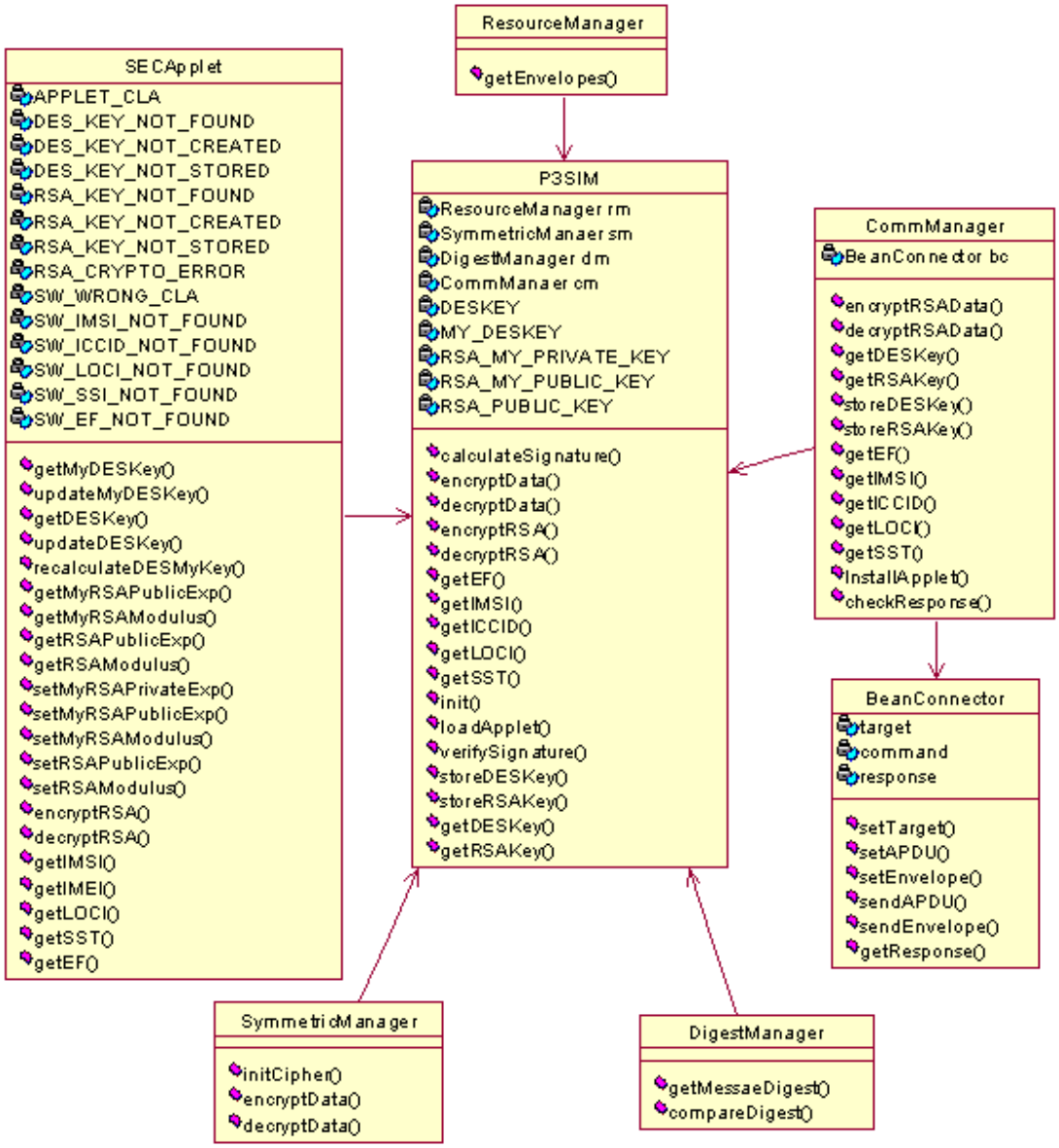

Figura 3-5. Diagrama de clases de la plataforma

3.3.5. Descripción de las clases

- P3SIM: Clase principal de la plataforma, esta provee los métodos estáticos que pueden ser invocados por los desarrolladores que deseen hacer uso de la plataforma.

- ResourceManager: Clase encargada del manejo de los recursos, provee la funcionalidad necesaria para recuperar las envelopeAPDU necesarias para la instalación del SECApplet.

- CommManager: Clase encargada del manejo de las comunicaciones entre la plataforma P3SIM y el Applet de seguridad SECApplet instalado en la tarjeta SIM del móvil, para realizar sus tareas se soporta en la clase BeanConnector.

- SymmetricManager: Clase encargada del manejo de la criptografía simétrica, posee métodos necesarios para el cifrado y descifrado utilizando el algoritmo DES.

- DigestManager: Clase encargada de la generación de *MessageDiggest* utilizando para esto el algoritmo SHA-1, además, permite la comparación de dos *MessageDiggest* a fin de verificar la validez de una firma.

- BeanConnector: Clase utilitaria quien realiza procesos de comunicación con el Applet de seguridad a bajo nivel, manipula los datos y los bytes necesarios para la generación de las CommandAPDU con las cuales se comunican la plataforma P3SIM y el Applet de seguridad instalado en la tarjeta SIM

- SECApplet: Clase correspondiente a JavaCard, en esta clase se implementan todas las características JavaCard y SAT que serán utilizadas, es decir, implementa las facilidades para seguridad, almacenamiento de claves, generación de claves, cifrado asimétrico y todas las facilidades SAT, es decir, acceso a parámetros SIM, actualización de parámetros SIM.

3.4. CONSTRUCCIÓN DE LA PLATAFORMA

La construcción de la plataforma involucró el manejo de los APIs de las tecnologías: J2SE, J2ME y Java Card.

3.4.1. Entorno de desarrollo

El entorno de desarrollo utilizado para manejar las tecnologías J2SE y J2ME fue "Eclipse", un IDE (*Integrated Development Environment*) de libre distribución. Se debe aclarar que el

manejo de J2ME por parte de Eclipse es logrado gracias a la adición de un Plug-in. En vista de que la única herramienta que permitió simular la conexión de un dispositivo J2ME con una tarjeta inteligente fue el "Sun Wireless Toolkit 2.3 Beta", se opto por utilizar el plug-in llamado "EclipseME" [28] el cual es distribuido bajo *Academic Free License*.

Para desarrollar con la tecnología Java Card se manejaron los entornos "*Aspects Developer*" [29] y "Eclipse". Se debe mencionar que las herramientas disponibles en Internet para manejar la tecnología Java Card son muy escasas y generalmente muy costosas. El entorno "Aspects Developer" fue uno de los mejores entornos que encontramos, a través de una versión de evaluación.

3.4.1.1. *ASPECTS DEVELOPER*

Las características de la versión utilizada de este entorno de desarrollo, para la construcción de Applets Java Card, son:

- Se integra con el Sun Java SDK 1.2 o 1.3 y con el Java Card Kit 2.1.2
- Tiene la capacidad de encontrar errores cuando se construye el sistema.
- Cuenta con un Wizard para la creación de Applets y paquetes.
- Posee decodificadores para archivos Java Card CAP y Export.
- Permite ejecutar los archivos estándar CAP.
- Soporta totalmente el API Java Card 2.1.2, el API GlobalPlatform 2.1, el API Visa OpenPlatform y los APIs ETSI GSM 03.19 y 03.48.
- Permite construir un Applet SIM Toolkit y simularlo sobre un dispositivo móvil de una forma muy sencilla. (Ver figura 3-6).
- Permite gestionar los archivos elementales GSM.
- Soporte criptográfico total (lo cual fue fundamental para la construcción de la plataforma).
- *Wizard* para generación de pares RSA y DSA.
- Un *debugger* que permite conocer el estado transitorio en memoria de todas las variables que pueda contener el Applet que se está simulando.
- Permite enviar archivos .scr de APDUs a Applets cargados sobre una tarjeta real o una simulada, con soporte a los estándares más utilizados: el ETSI, el de SUN y el ISO. Genera el respectivo reporte de los Command y Response APDUs intercambiados con el entorno de desarrollo.

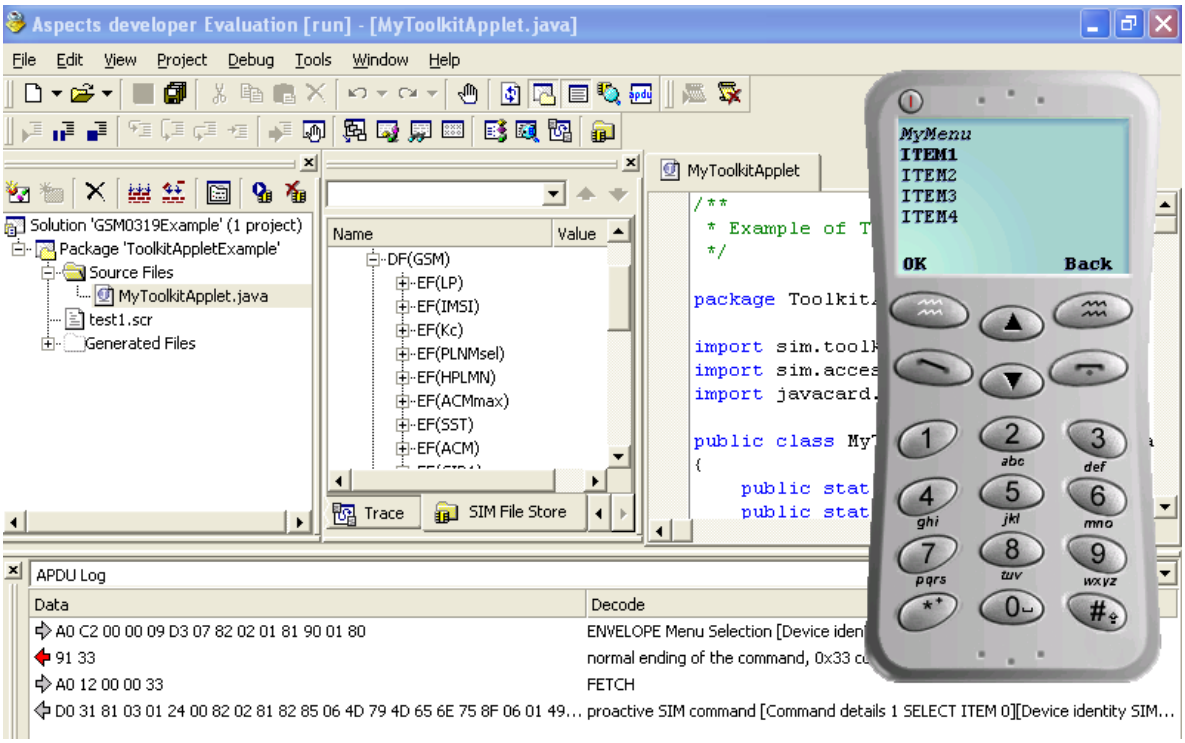

Figura 3-6. Aspects Developer

3.4.1.2. *ECLIPSE*

Para "Eclipse" se creo un archivo xml que permite compilar, empaquetar y generar los scripts de los Applets Java Card de una forma muy fácil, similar a como lo hace el "Aspects Developer". Se debe tener en cuenta que el archivo xml hace uso del Java Card Kit 2.2.1 y del Sun JSDK 1.4.x. A continuación se muestra este archivo.

```
<?xml version="1.0" encoding="UTF-8"?>
<project default="makeAll" name="JavaCard Builder" basedir=".">

  <property name="scr" value="src" />
  <property name="classes" value="classes" />
  <property name="script" value="script"/>
  <property name="package" value="SECApplet"/>
  <property name="AID" value="0xa0:0x00:0x00:0x00:0x62:0x01:0x0d 1.0"/>
  <property name="Applet"
   value="0xa0:0x00:0x00:0x00:0x62:0x01:0x0d:0x01 ${package}.SECApplet"/>

  <property name="JC_HOME" value="C:\javacard\java_card_kit-2_2_1"/>

<target name="clean" description="clean project directories">
  <delete dir="${classes}" />
</target>

<target name="compile" depends="clean" description="Compile code">
```

```
  <mkdir dir="${classes}"/>
  <javac srcdir="${scr}" destdir="${classes}" debug="yes" verbose="yes"/>
</target>
<target name="converter" depends="compile" description="Generate EXP, CAP files">
  <exec executable="${JC_HOME}/bin/converter.bat">
            <arg line="-classdir ${classes}" />
            <arg line="-exportpath ${JC_HOME}/api_export_files"/>
            <arg line="-out EXP JCA CAP"/>
            <arg line="-applet ${Applet}"/>
            <arg line="${package} ${AID}"/>
  </exec>
<!--<copydir dest="javacard" src="${classes}/${package}/javacard"></copydir>-->
  <copy todir="javacard" overwrite="yes">
      <fileset dir="${classes}/${package}/javacard"></fileset>
  </copy>
  <delete dir="${classes}/${package}/javacard" />
</target>
<target name="scriptgen" depends="converter" description="Gener the script file">
  <mkdir dir="${script}"/>
  <exec executable="${JC_HOME}/bin/scriptgen.bat">
      <arg line="-o script/${package}_script.txt"/>
      <arg line="javacard/${package}.cap"/>
  </exec>
</target>

<target name="makeAll" depends="scriptgen" description="make all targets"/>
</project>
```

Como se puede observar primero se definen las propiedades y luego se realizan cuatro tareas. La primera es limpiar el directorio, la segunda es compilar el Applet, la tercera es convertir los .class en un paquete .cap, y la cuarta es generar el script que contiene los APDUs que crearán el Applet en la implementación de referencia de Java Card, conocida como "cref". Como se puede observar en el archivo xml, la cuarta tarea depende de la tercera, la tercera de la segunda y la segunda de la primera. Es decir si no se puede convertir al .cap (tercera tarea) entonces no se puede crear el script de APDUs (cuarta tarea).

3.4.2. Documentación

Adicional a la plataforma se generaron dos documentos: el manual de usuario y el API doc que describe todos los métodos y atributos de las clases que hacen parte de la plataforma.

3.5. CARACTERÍSTICAS FINALES DE LA PLATAFORMA

P3SIM es una plataforma para brindar seguridad a los servicios móviles única en su género, debido a que sólo se necesita del dispositivo móvil y de la tarjeta SIM. Sus características finales son:

- Permite acceder a los parámetros SIM: IMSI, ICCID, SIM Service Table y LOCI.
- Genera y almacena dos claves simétricas DES y tres claves asimétricas (dos claves públicas y una privada) RSA.
- Permite cifrar o descifrar información con cualquiera de las claves, teniendo en cuenta el algoritmo criptográfico.
- Genera y verifica la firma digital de alguna información, especificado el algoritmo que se quiere utilizar.

3.5.1. Ventajas de la plataforma

Las ventajas que se encontraron en P3SIM son:

- Le permite al proveedor de servicios acceder fácilmente a parámetros SIM, lo que anteriormente sólo era privilegio del operador.
- El dispositivo J2ME puede generar claves RSA, lo cual actualmente no se puede hacer con ninguno de los APIs de J2ME.
- Genera, obtiene y almacena claves en la tarjeta SIM, lo cual le brinda al usuario el máximo nivel de seguridad que existe en la actualidad.
- Al utilizar las facilidades proporcionadas por P3SIM y compararlo con el uso de un protocolo seguro como HTTPS u otro basado en XML, se observa un mejor aprovechamiento del ancho de banda por parte de la plataforma.

3.5.2. Características técnicas de la plataforma

La plataforma consta de un archivo .jar el cual empaqueta a seis clases java y un archivo de texto que contiene las APDUs que instalarán un Applet en la tarjeta SIM.

El dispositivo móvil sobre el cual se instala el MIDlet que hace uso de la plataforma, debe soportar dos paquetes opcionales de SATSA: SATSA-APDU y SATSA-CRYPTO.

La SIM debe ser una tarjeta con soporte Java Card 2.1.1 en adelante.

4. DESCRIPCIÓN DEL PROTOTIPO

4.1. ANÁLISIS Y DISEÑO

El prototipo ideado para validar la plataforma es una aplicación enmarcada en el servicio de m-commerce, debido a que las características de seguridad brindadas por P3SIM pueden ser mostradas en toda su magnitud en un tipo de servicio como este, en donde la seguridad de las transacciones realizadas es el aspecto mas importante tanto para el proveedor de servicio como para el usuario móvil.

La aplicación le permite al usuario:

- Suscribirse (proceso realizado una sola vez) ante el proveedor de servicio.
- Ofrecer un producto, para que otros usuarios lo puedan observar y si les parece lo puedan comprar.
- Buscar un producto mediante unos criterios de búsqueda como el nombre del producto.
- Comprar/Pagar un producto.

Este prototipo consta de dos partes: una parte cliente (un MIDlet J2ME) y una parte servidor que es manejada por el Proveedor del servicio.

4.1.1. Diagrama de casos de uso

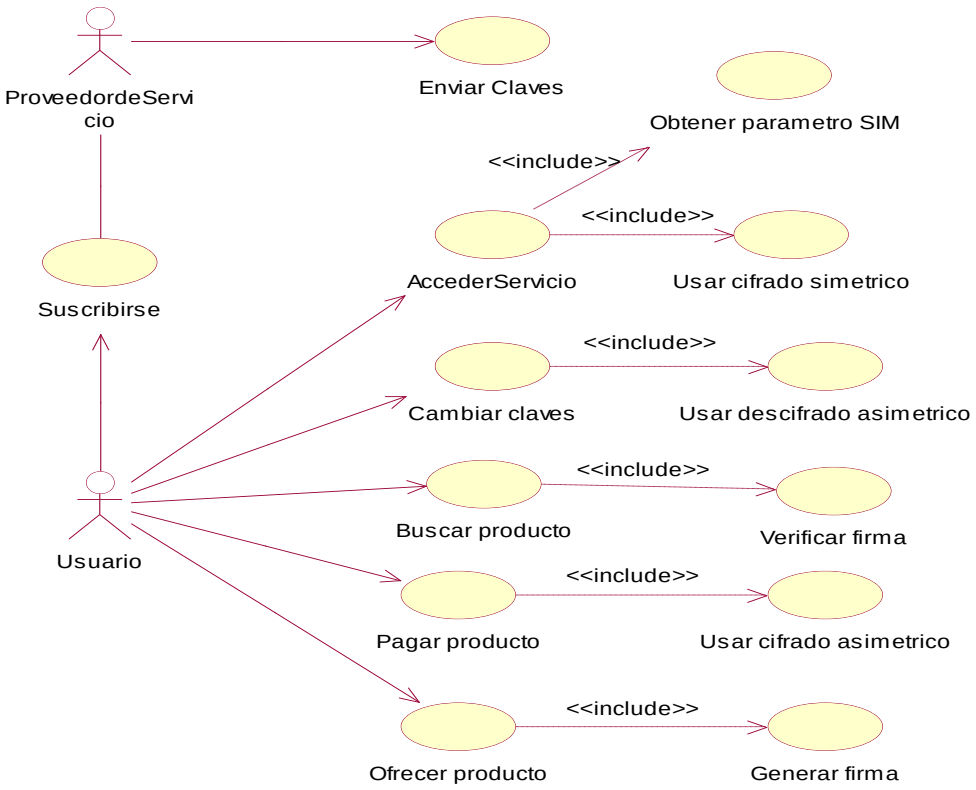

Figura 4-1. Diagrama de casos de uso del prototipo

4.1.2. Descripción de los casos de uso

Información General	
Caso de uso:	**Suscribirse**
Actores:	Usuario
Propósito:	Permitirle al proveedor de servicio conocer los parámetros SIM del usuario.
Resumen:	El Usuario digita su nombre y presiona la opción "suscribirse". El MIDlet adicionalmente al nombre, envía los parámetros SIM IMSI y el ICCID al servidor.
Tipo:	Primario.
Referencias cruzadas:	No hay.

Precondiciones
- El SECApplet debe estar instalado en la tarjeta SIM.

Flujo Principal
– El usuario ingresa a la opción suscribirse del MIDlet.
– El MIDlet le muestra al usuario un espacio para que se digite el nombre.
– El usuario presiona la opción "suscribirse"
– El MIDlet haciendo uso de P3SIM obtiene el ICCID y el IMSI del usuario.
– El MIDlet envía al servidor el nombre, el ICCID y el IMSI del usuario.
– El servidor retorna un mensaje informando del éxito en la suscripción del usuario.
– El MIDlet le muestra al usuario el mensaje de éxito.

Flujos de Excepción
E1: Parámetros SIM no accesibles.
– Por alguna razón no se pudo obtener el ICCID o el IMSI del usuario.
– Se le muestra al usuario el mensaje de fracaso en la suscripción.

Información General	
Caso de uso:	**Acceder servicio**
Actores:	Usuario
Propósito:	Permitirle al usuario acceder al servicio.
Resumen:	El MIDlet envía el ICCID y el IMSI cifrado del usuario al servidor.
Tipo:	Primario.
Referencias cruzadas:	No hay.
Precondiciones	
- El Usuario tiene que estar registrado (suscribirse previamente) ante el proveedor de servicio. - El SECApplet debe estar instalado en la tarjeta SIM.	

Flujo Principal
– El usuario selecciona la opción "ingresar".
– El MIDlet obtiene una clave DES (conocida por el proveedor del servicio) y cifra el IMSI del usuario, luego lo envía junto al ICCID.
– El servidor valida estos datos y le envía al usuario la respuesta OK.

- El MIDlet le permite al usuario acceder a todas las opciones del prototipo: comprar producto, …

Flujos de Excepción
E1: Parámetros SIM o clave DES no accesible.
- Se le muestra al usuario el mensaje de fracaso en el ingreso.

Información General	
Caso de uso:	**Cambiar claves**
Actores:	Usuario
Propósito:	Permitirle al usuario cambiar sus claves.
Resumen:	El Servidor le envía las nuevas claves al MIDlet y luego se almacenan en la tarjeta SIM.
Tipo:	Primario.
Referencias cruzadas:	Enviar clave.
Precondiciones	
- El Usuario tiene que estar registrado (suscribirse previamente) ante el proveedor de servicio. - El SECApplet debe estar instalado en la tarjeta SIM.	

Flujo Principal
- El usuario selecciona la opción "cambiar claves".
- El servidor le envía al MIDlet las nuevas claves cifradas con la clave pública del usuario.
- El MIDlet descifra dichas claves con la clave privada del usuario.
- El MIDlet almacena las nuevas claves.
- El MIDlet le muestra al usuario el mensaje "claves cambiadas exitosamente"

Flujos de Excepción
E1: Error criptográfico.
- Si en el recorrido fue alterada la información enviada del servidor al cliente, se produce una excepción de seguridad.
- Se le muestra al usuario el mensaje "no se pudieron cambiar las claves".

Información General	
Caso de uso:	**Enviar claves**
Actores:	Proveedor de servicio
Propósito:	Permitirle al usuario cambiar las claves.
Resumen:	El servidor le envía al usuario una clave nueva, esta clave es enviada en un formato cifrado.
Tipo:	Primario.
Referencias cruzadas:	No hay.

Precondiciones

- El Usuario tiene que estar registrado (suscribirse previamente) ante el proveedor de servicio.
- El SECApplet debe estar instalado en la tarjeta SIM.

Flujo Principal

- El usuario selecciona la opción "cambiar claves".
- El proveedor de servicio genera una nueva clave y luego la cifra con la clave pública del usuario. Se envía al cliente la nueva clave cifrada.
- El MDIet recibe la información y la descifra con la clave privada del usuario.
- El MIDlet almacena la nueva clave en la tarjeta SIM.
- El MIDlet le muestra al usuario el mensaje de éxito

Flujos de Excepción

E1: Error criptográfico.

- Si en el recorrido fue alterada la información enviada del servidor al cliente, se produce una excepción de seguridad.
- Se le muestra al usuario el mensaje "error en el cambio de claves".

Información General	
Caso de uso:	**Buscar producto**
Actores:	Usuario
Propósito:	Permitirle al usuario buscar un producto.
Resumen:	El servidor le envía al usuario la descripción de un producto junto con su firma digital.
Tipo:	Primario.

Referencias cruzadas:	No hay.

Precondiciones

- El Usuario tiene que estar registrado (suscribirse previamente) ante el proveedor de servicio.
- El SECApplet debe estar instalado en la tarjeta SIM.

Flujo Principal

- El usuario entra a la opción "Buscar producto".
- El servidor le envía al cliente la lista de los productos disponibles.
- El usuario selecciona algún producto disponible.
- El servidor le envía al cliente la descripción de un producto junto con su firma digital
- El MIDlet le muestra al usuario la información del producto, siempre y cuando la firma obtenida sea válida.

Flujos de Excepción

E1: Error criptográfico.

- Si en el recorrido fue alterada la información enviada del servidor al cliente, se produce una excepción de seguridad.
- Se le muestra al usuario el mensaje "no se puede observar la información del producto".

Información General

Caso de uso:	**Pagar producto**
Actores:	Usuario
Propósito:	Permitirle al usuario pagar un producto.
Resumen:	El usuario le envía al servidor el número de su cuenta bancaria cifrada asimétricamente.
Tipo:	Primario.
Referencias cruzadas:	No hay.

Precondiciones

- El Usuario tiene que estar registrado (suscribirse previamente) ante el proveedor de servicio.
- El SECApplet debe estar instalado en la tarjeta SIM.

Flujo Principal
– El usuario entra a la opción "pagar producto".
– El usuario digita su número de cuenta bancaria y selecciona la opción "pagar".
– El MIDlet cifra el número de cuenta bancaria con la clave pública del proveedor de servicio.
– El MIDlet le envía el número de cuenta bancaria cifrado.
– El Servidor descifra esta información con la clave privada del proveedor de servicio. Si este proceso es exitoso se le envía al cliente el mensaje de éxito.
– El MIDlet le muestra al usuario el mensaje "producto pagado"

Flujos de Excepción
E1: Error criptográfico.
– Si en el recorrido fue alterada la información enviada del cliente al servidor, se produce una excepción de seguridad.
– Se le muestra al usuario el mensaje "no se puede pagar el producto".

Información General	
Caso de uso:	**Ofrecer producto**
Actores:	Usuario
Propósito:	Permitirle al usuario ofrecer un producto en el servidor.
Resumen:	El usuario le envía al servidor la descripción del producto y la firma digital de dicha descripción.
Tipo:	Primario.
Referencias cruzadas:	No hay.
Precondiciones	
- El Usuario tiene que estar registrado ante el proveedor de servicio.	
- El SECApplet debe estar instalado en la tarjeta SIM.	

Flujo Principal
– El usuario entra a la opción "ofrecer producto".
– El usuario digita la descripción del producto.
– El MIDlet genera la firma digital de la descripción del producto.
– El MIDlet le envía al servidor la descripción del producto y la firma digital.

- El servidor verifica la firma digital. Si la firma es válida, se le envía al cliente el mensaje de éxito.
- El MIDlet le muestra al usuario el mensaje "producto ofrecido"

Flujos de Excepción

E1: Error criptográfico.

- Si en el recorrido fue alterada la información enviada del cliente al servidor, se produce una excepción de seguridad.
- Se le muestra al usuario el mensaje "no se puede ofrecer el producto".

4.1.3. Diagramas de secuencia

Caso de uso: Suscribirse

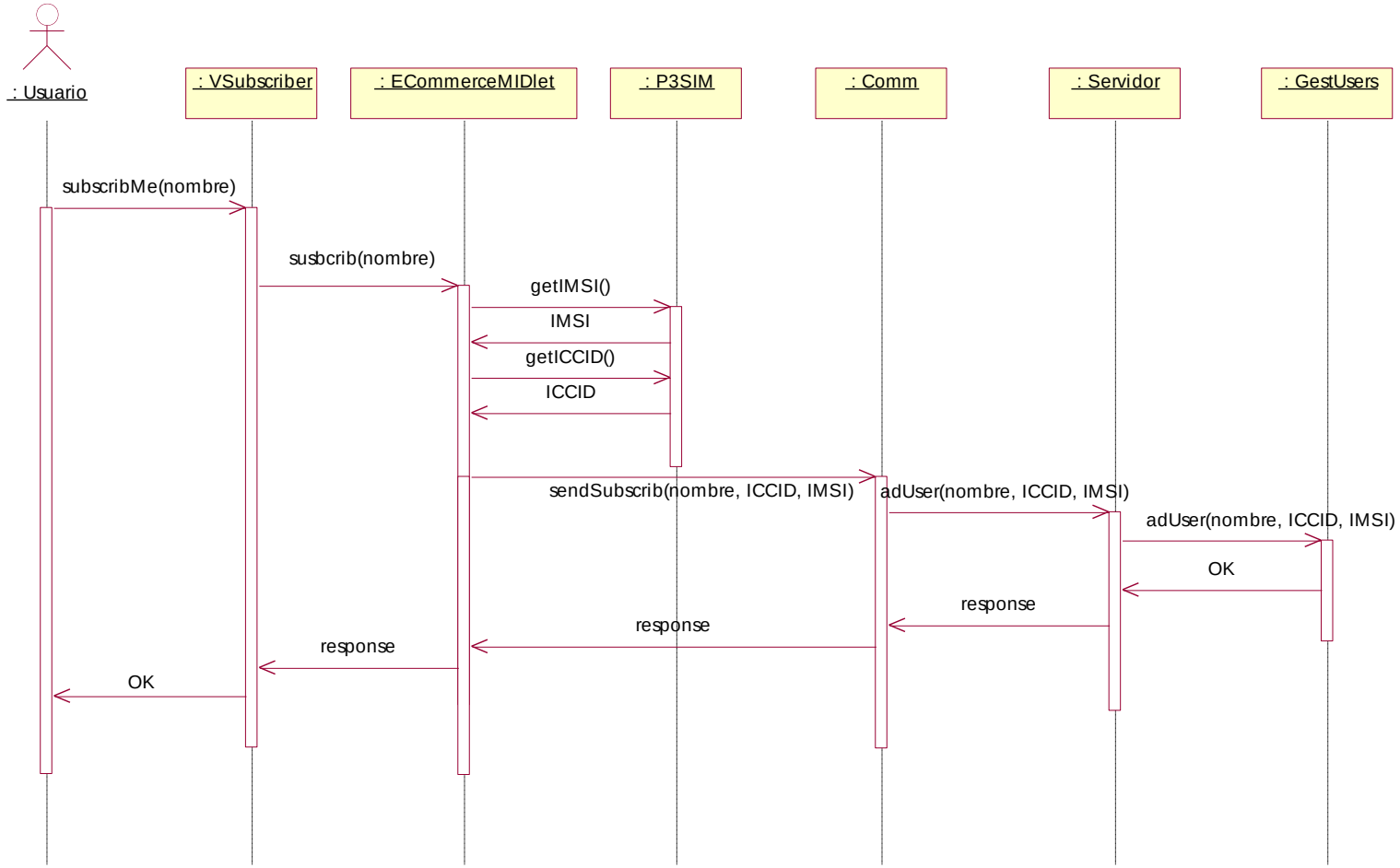

Figura 4-2. Diagrama de secuencia para Suscribirse

Caso de uso: Acceder servicio

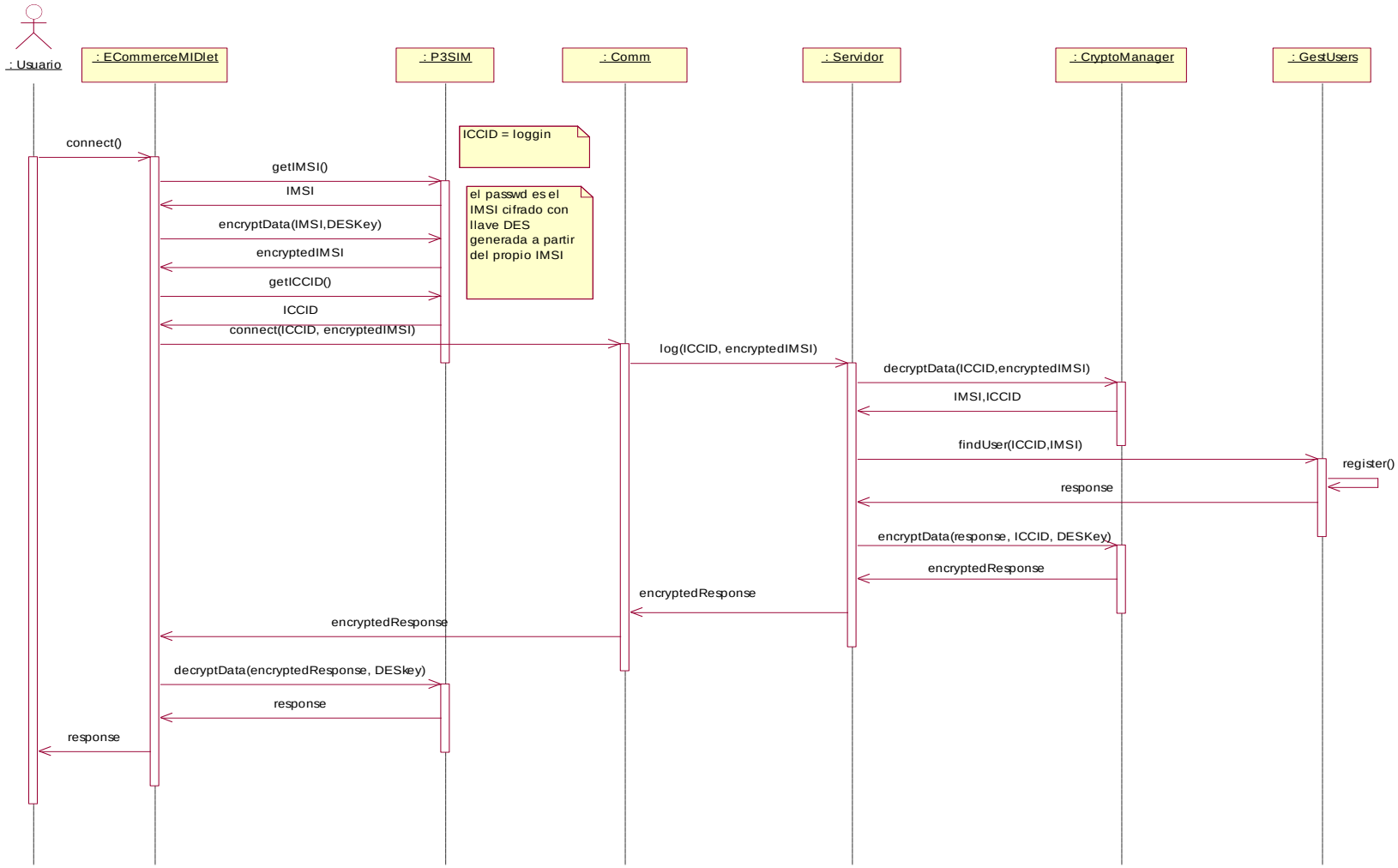

Figura 4-3. Diagrama de secuencia para Acceder servicio

Casos de uso: Cambiar claves

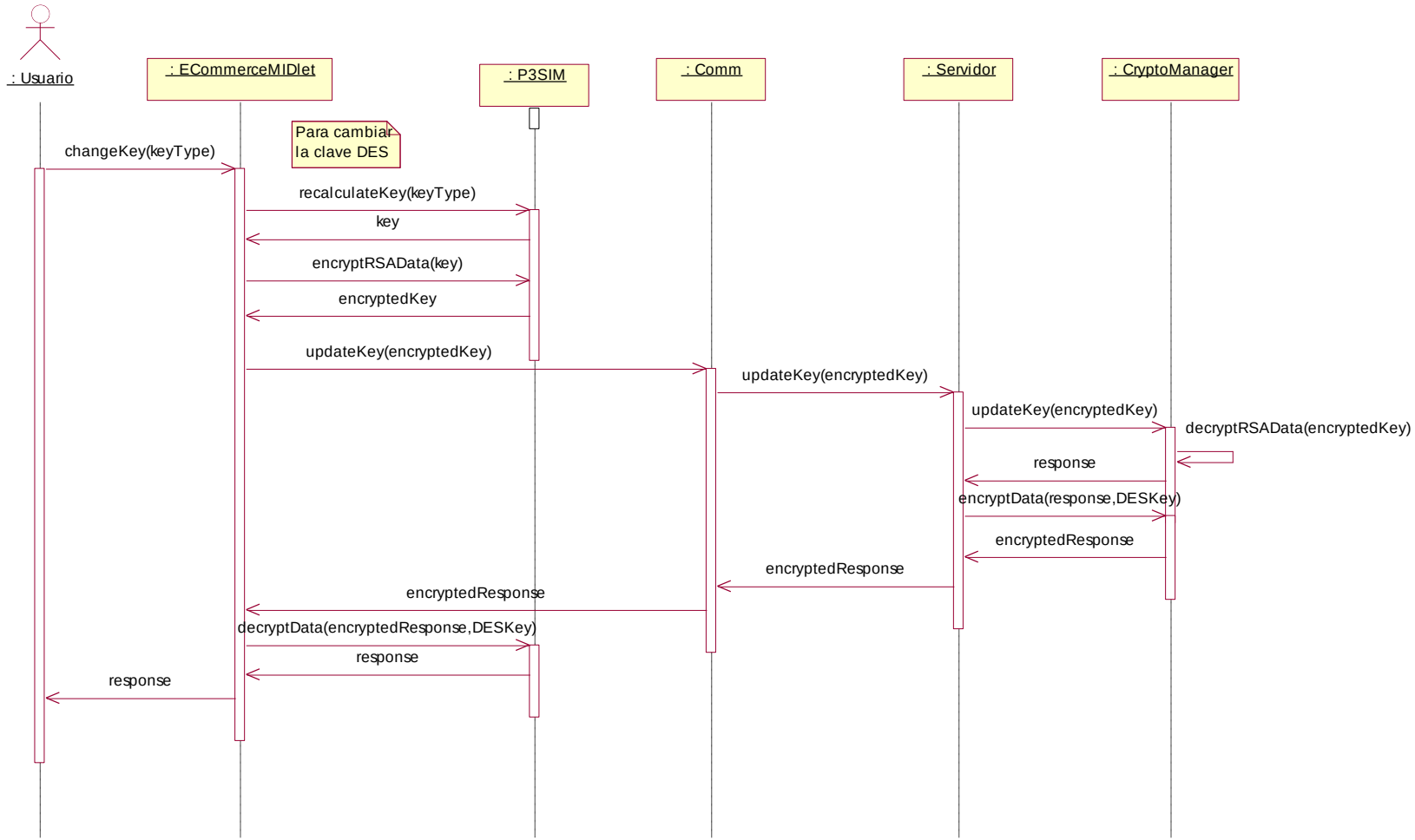

Figura 4-4. Diagrama de secuencia para Cambiar claves

Caso de uso: Enviar claves

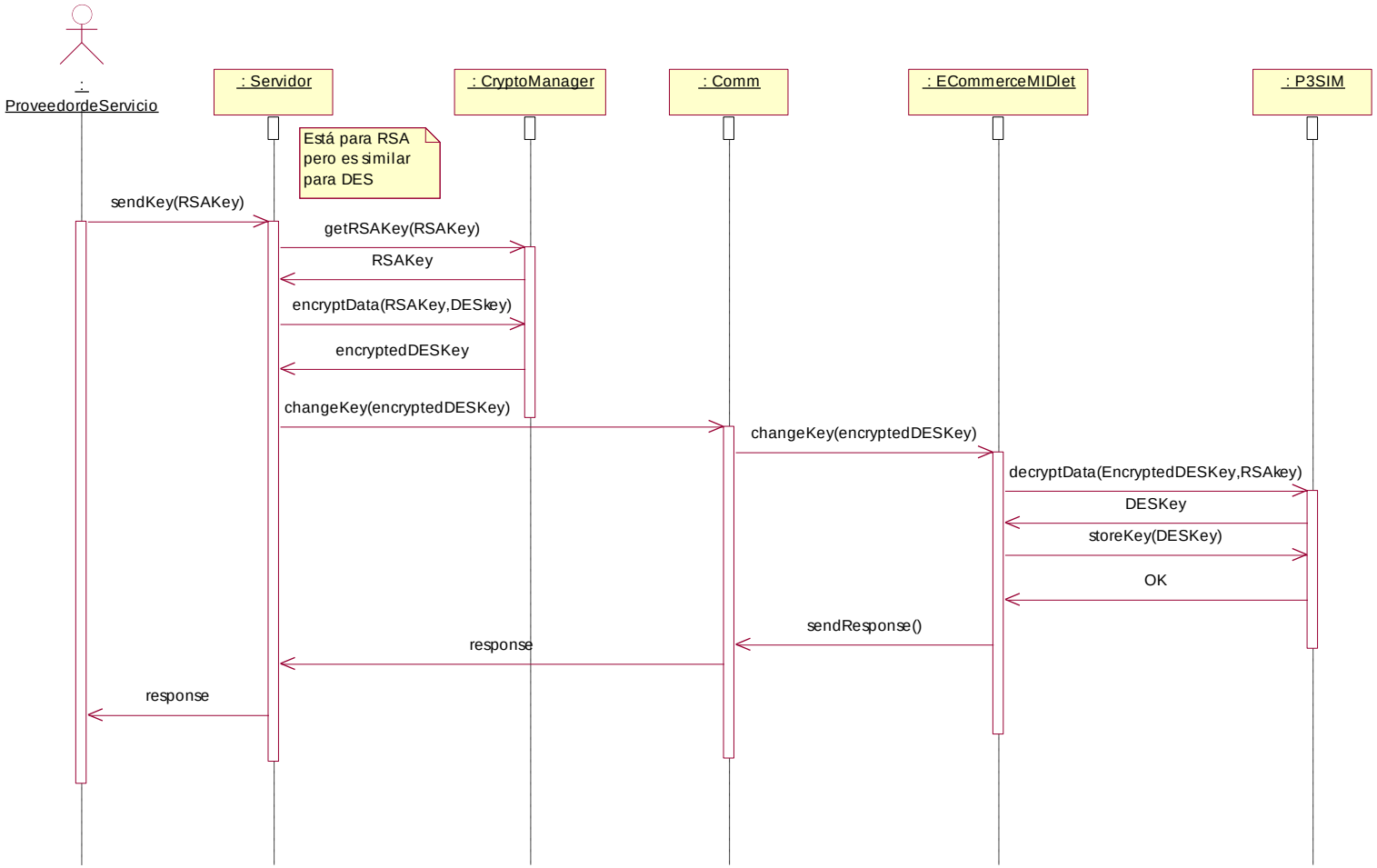

Figura 4-5. Diagrama de secuencia para Enviar claves

Caso de uso: Ofrecer producto

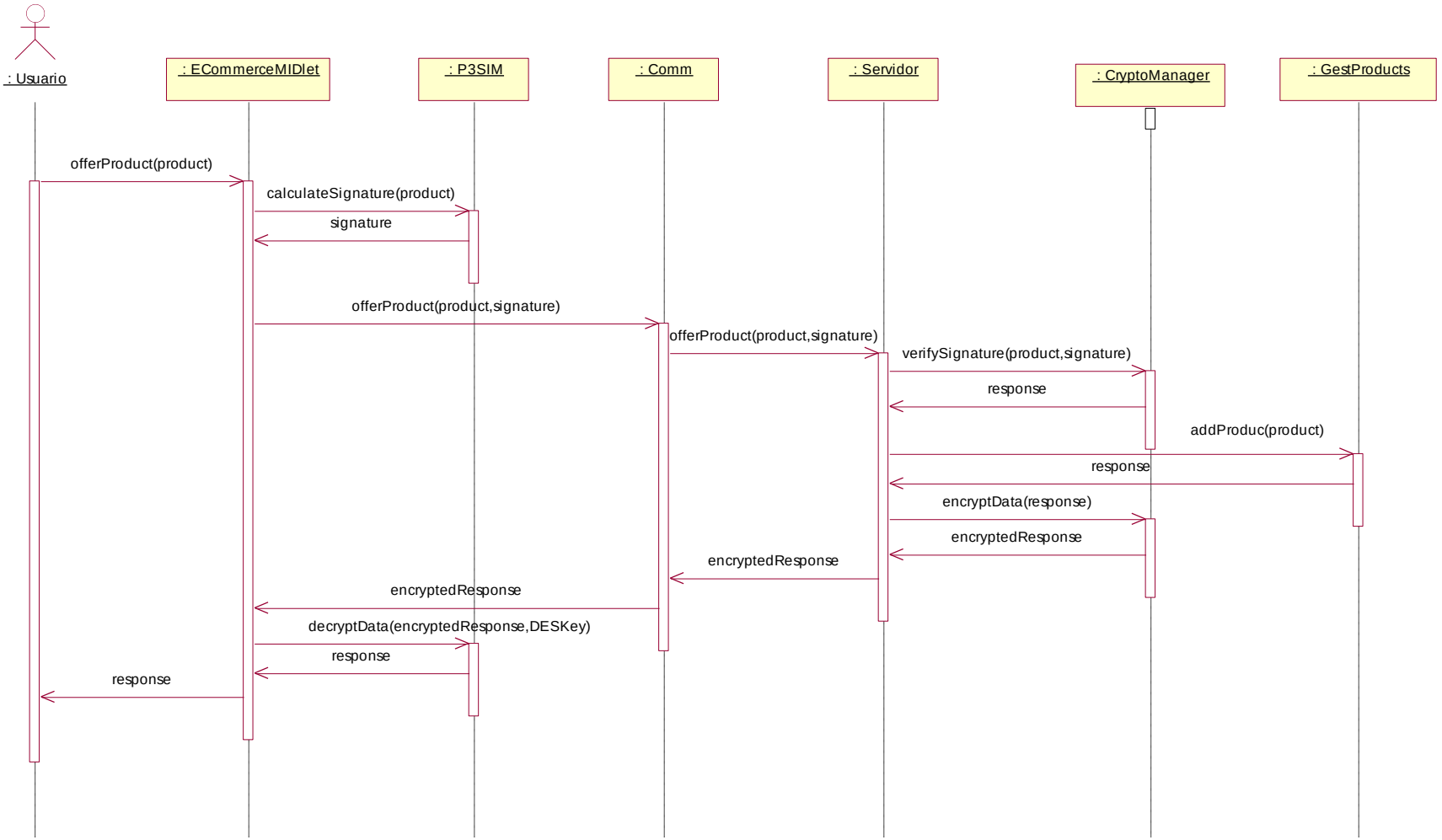

Figura 4-6. Diagrama de secuencia para Ofrecer producto

4.1.4. Diagrama de clases

La arquitectura del sistema como se mencionó anteriormente está compuesta de dos partes: el cliente (figura 4-7) y el servidor (figura 4-8).

La plataforma sólo puede ser accedida a través de la clase P3SIM mediante la invocación de sus métodos estáticos.

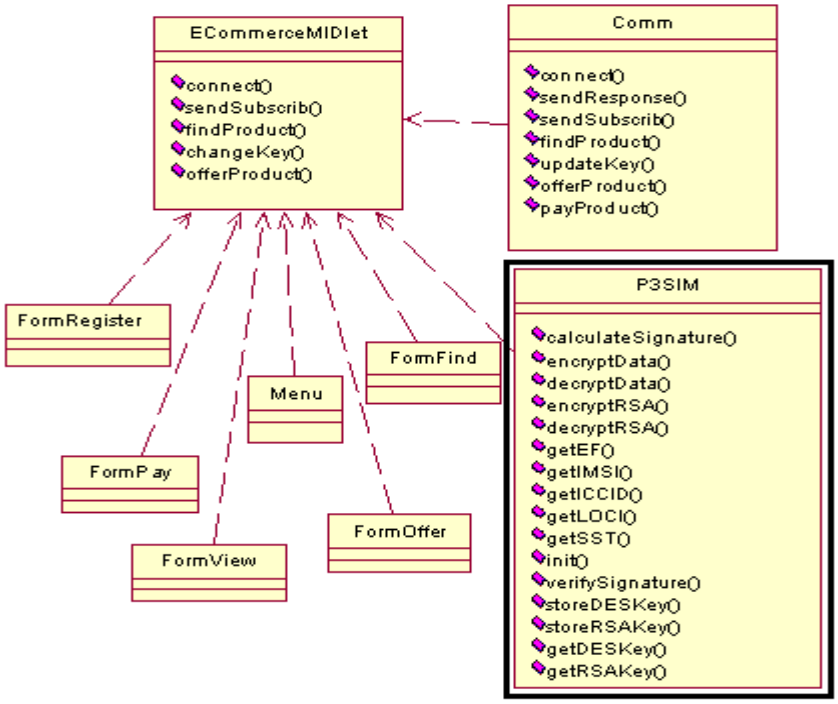

Figura 4-7. Diagrama de clases del cliente del prototipo

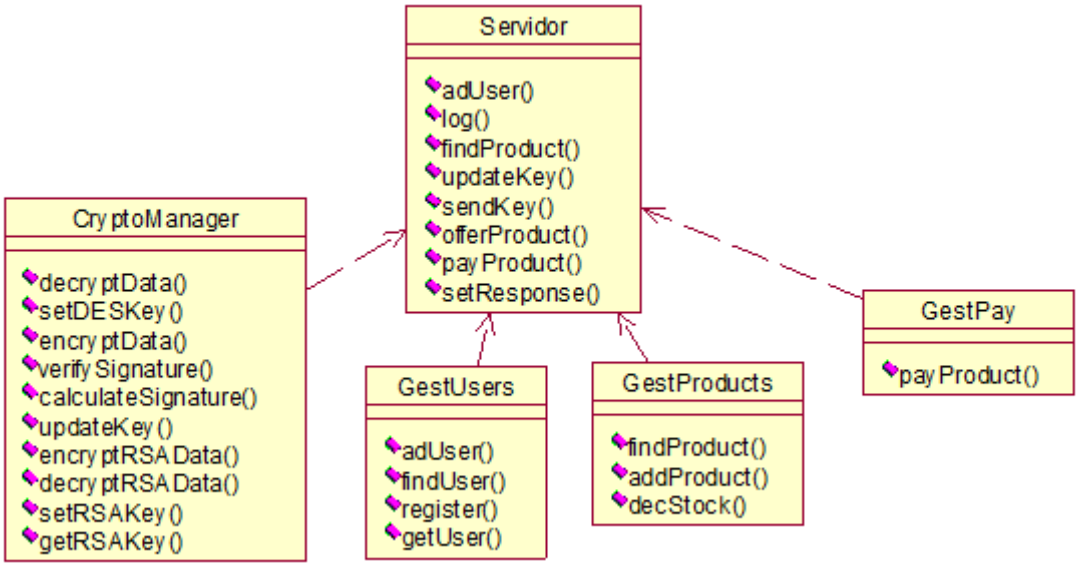

Figura 4-8. Diagrama de clases del servidor del prototipo

4.2. CARACTERÍSTICAS FINALES

4.2.1. Diagrama de despliegue

El diagrama de despliegue se muestra en la figura 4-9. En el servidor se manejan dos conceptos:

- Gestión de la seguridad: lo cual es efectuado por las clases CryptoManager y GestUsers.
- Gestión de comercio: lo cual es realizado por las clases GestProducts y GestPay.

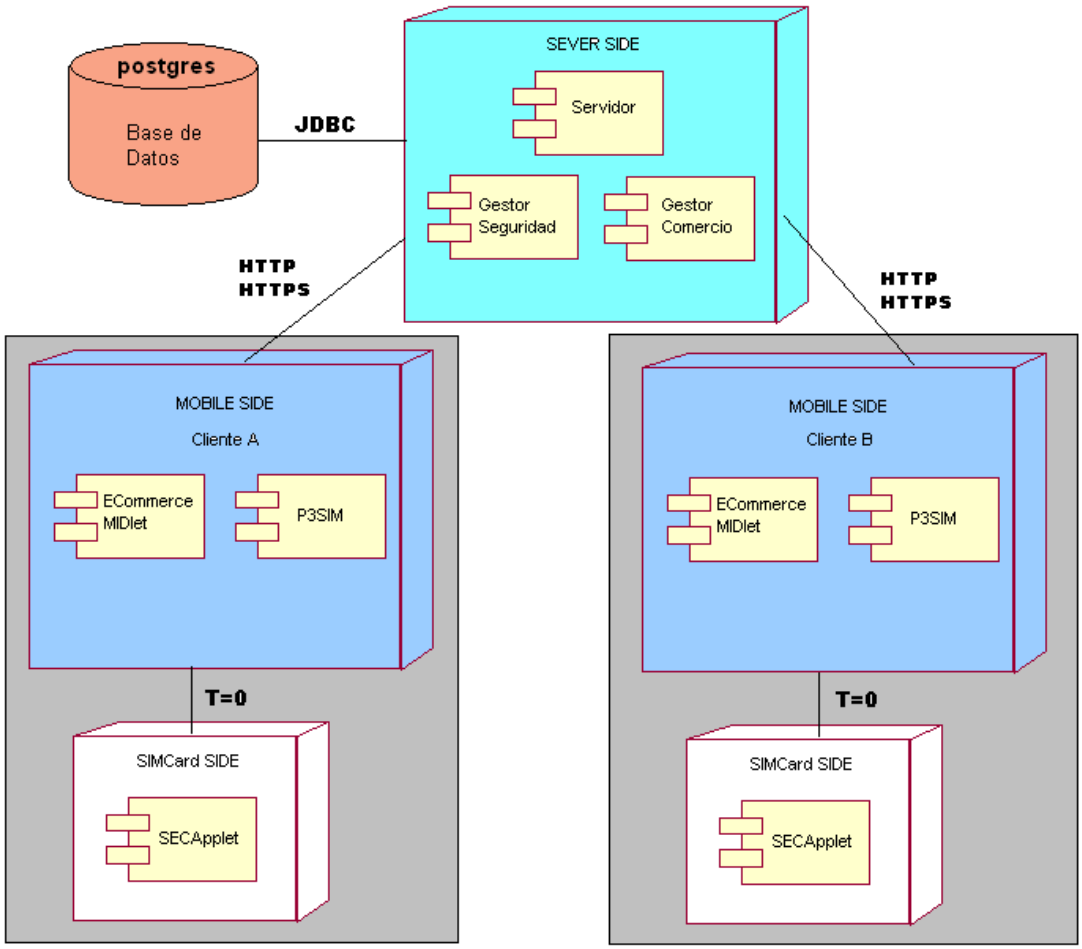

Figura 4-9. Diagrama de despliegue del prototipo

El prototipo utilizado para la validación de la plataforma P3SIM consta de dos partes principales: el servidor del comercio electrónico y el cliente de comercio electrónico.

El lado del servidor se encuentra implementado bajo la especificación J2EE prestando sus servicios como Servicios Web.

El lado del cliente es un MIDlet que utiliza las facilidades que brinda la plataforma P3SIM.

La plataforma P3SIM brinda al MIDlet de validación la posibilidad de identificar a los usuarios mediante la utilización de los parámetros SIM, utilizamos el parámetro IMSI como semilla para la generación de la clave DES utilizada para las comunicaciones iniciales entre el cliente y el servidor y el parámetro ICCID como identificador único de cada usuario.

4.2.2. Características reales del prototipo

El prototipo finalmente desplegado consta de un MIDlet que se simula con el Wireless Toolkit 2.3 y un archivo war que se desplegó en un servidor Apache Tomcat.

Adicionalmente se maneja una base de datos PostgreSQL, que maneja la información de los usuarios.

Se necesita del java_card_kit-2_2_1. Finalmente para poder simular la creación de claves RSA en java card y acceder a los parámetros SIM se necesita del Aspects Developer.

4.2.2.1. Interfaces gráficas del prototipo


NOKIA
Applications
ECommerceMIDlet
NOKIA
Menu
Registrar
Options
Close
NOKIA
Menu
Connectar
Options
Close
NOKIA
Message
Message
Registration
successful
completed
OK
NOKIA
Menu
Buscar Producto
Ofrecer Producto
Options
Close
NOKIA
Register
Name :
Jaime
Surname :
Coaicedo
Company :
Unicauca
Country :
Options
Close

Figura 4-10. Interfaz gráfica del prototipo de validación

5. PRUEBAS DE LA PLATAFORMA

5.1. PRUEBAS DE UNIDAD

5.1.1. Diseño de las pruebas de unidad

Lo primero que se hizo fue construir el Applet para la tarjeta SIM. Para probarlo se construyeron unos archivos de texto que contenían los APDUs que manejaban el applet. A continuación se muestra uno de estos archivos:

```
powerup;

echo "Select SECApplet";
0x00 0xA4 0x04 0x00 0x08 0xa0 0x0 0x0 0x0 0x62 0x1 0xd 0x1 0x7F;

echo "Solicitud de mi clave DES";
0x70 0x01 0x00 0x00 0x00 0x08;

echo "Solicitud de la otra clave DES";
0x70 0x02 0x00 0x00 0x00 0x08;

echo "actualizacion de mi clave DES";
0x70 0x11 0x00 0x00 0x08 0x55 0x54 0x53 0x52 0x51 0x50 0x58 0x59 0x7F;

echo "Solicitud de mi clave DES";
0x70 0x01 0x00 0x00 0x00 0x08;

echo "Solicitud de la otra clave DES";
0x70 0x02 0x00 0x00 0x00 0x08;

echo "actualizacion de la otra clave DES";
0x70 0x12 0x00 0x00 0x08 0x79 0x78 0x76 0x75 0x74 0x73 0x72 0x71 0x7F;

echo "Solicitud de mi clave DES";
0x70 0x01 0x00 0x00 0x00 0x08;

echo "Solicitud de la otra clave DES";
0x70 0x02 0x00 0x00 0x00 0x08;

echo "recalcula mi clave DES";
0x70 0x21 0x00 0x00 0x00 0x7F;

echo "Solicitud de mi clave DES";
0x70 0x01 0x00 0x00 0x00 0x08;

powerdown;
```

Cuando se crearon los métodos del applet para acceder a los parámetros SIM, también se diseñaron las respectivas pruebas. Las pruebas consistieron en acceder a los parámetros: IMSI, ICCID, LOCI y SST.

Luego se procedió a crear los métodos del applet que gestionaban las dos claves DES. La pruebas consistieron en generar, actualizar y obtener cualquiera de las dos claves DES.

Posteriormente se procedió a crear los métodos del applet que gestionaban las tres claves RSA y que permitían cifrar/descifrar información con dichas claves. Las pruebas consistieron en generar, fijar y obtener los parámetros de las tres claves RSA, también se diseñaron las pruebas que permitieran cifrar con una clave y descifrarla con su clave par.

Una vez probado el applet, se procedió a construir las clases J2ME que completan a la plataforma. Las pruebas diseñadas consistían en imprimir unos mensajes de texto que indicaban si los métodos funcionan bien o mal.

5.1.2. Resultados de las pruebas de unidad

Para poder probar el acceso a los parámetros SIM se tuvo que trabajar con el Aspects Developer, debido a que el Java Card Kit de Sun no soporta este tipo de acceso. Los resultados fueron favorables, es decir, se pudo acceder al IMSI, ICCID, LOCI y SST de la SIM emulada por el Aspects Developer.

Para probar la gestión de claves simétricas, si se pudo trabajar con el Java Card Kit. El proceso realizado consistió en simular por medio del "cref" una tarjeta que tenia cargado el SECApplet y por medio del "apdutool" se le envió el archivo mostrado en la sección anterior, lo cual produjo un archivo de salida con los correspondientes response APDUs relacionados en los parámetros "Le", "SW1" y "SW2". Los resultados fueron favorables, es decir, se pudo generar, almacenar y obtener cualquiera de las dos claves DES. El archivo obtenido fue el siguiente:

```
Select SECApplet
CLA: 00, INS: a4, P1: 04, P2: 00, Lc: 08, a0, 00, 00, 00, 62, 01, 0d, 01,
Le: 00, SW1: 90, SW2: 00

Solicitd de mi clave DES
CLA: 70, INS: 01, P1: 00, P2: 00, Lc: 00, Le: 08, 01, 02, 03, 04, 03, 02,
01, 00, SW1: 90, SW2: 00
```

```
Solicitud de la otra clave DES
CLA: 70, INS: 02, P1: 00, P2: 00, Lc: 00, Le: 08, 01, 01, 01, 01, 01, 01,
01, 01, SW1: 90, SW2: 00

actualizacion de mi clave DES
CLA: 70, INS: 11, P1: 00, P2: 00, Lc: 08, 55, 54, 53, 52, 51, 50, 58, 59,
Le: 00, SW1: 90, SW2: 00

Solicitud de mi clave DES
CLA: 70, INS: 01, P1: 00, P2: 00, Lc: 00, Le: 08, 55, 54, 53, 52, 51, 50,
58, 59, SW1: 90, SW2: 00

Solicitud de la otra clave DES
CLA: 70, INS: 02, P1: 00, P2: 00, Lc: 00, Le: 08, 01, 01, 01, 01, 01, 01,
01, 01, SW1: 90, SW2: 00

actualizacion de la otra clave DES
CLA: 70, INS: 12, P1: 00, P2: 00, Lc: 08, 79, 78, 76, 75, 74, 73, 72, 71,
Le: 00, SW1: 90, SW2: 00

Solicitud de mi clave DES
CLA: 70, INS: 01, P1: 00, P2: 00, Lc: 00, Le: 08, 55, 54, 53, 52, 51, 50,
58, 59, SW1: 90, SW2: 00

Solicitud de la otra clave DES
CLA: 70, INS: 02, P1: 00, P2: 00, Lc: 00, Le: 08, 79, 78, 76, 75, 74, 73,
72, 71, SW1: 90, SW2: 00

recalcula mi clave DES
CLA: 70, INS: 21, P1: 00, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00

Solicitd de mi clave DES
CLA: 70, INS: 01, P1: 00, P2: 00, Lc: 00, Le: 08, 59, 5d, a0, 5e, 61, 8d,
a5, a6, SW1: 90, SW2: 00
```

Para probar la parte de gestión de las claves RSA y el cifrado/descifrado de información de nuevo se tuvo que trabajar con el Aspects Developer. Los resultados fueron favorables, es decir, se pudo generar, fijar y obtener los parámetros de las tres claves RSA, también se logró cifrar con una clave y descifrar con su clave par sobre una tarjeta SIM emulada por el Aspects Developer.

5.2. PRUEBAS DEL SISTEMA

5.2.1. Diseño del sistema

Las pruebas del sistema consistieron en probar a la plataforma como un todo. Las pruebas diseñadas fueron:

- La instalación del Applet en la SIM simulada, por parte de un MIDlet.

- Crear una clave DES con base en un parámetro SIM.
- Obtener una clave simétrica de la tarjeta simulada, y luego cifrar/descifrar información con dicha clave.
- Obtener un Hash de cualquier información, con los algoritmos soportados por SATSA.
- Generar y verificar una firma digital, para ello el Wireless Toolkit debe acceder a la tarjeta SIM simulada.

5.2.2. Resultados de las pruebas del sistema

Los resultados obtenidos fueron favorables, aunque se tuvieron algunos inconvenientes en aquellos casos en donde se necesitaba de la tarjeta SIM emulada por el Aspects Developer.

Vale la pena mencionar el hecho de que se trabajó con tarjetas Java Card reales. Las tarjetas utilizadas fueron las tarjetas "Axalto USIMERA 128k" [30]. Para poder instalar un Applet en una tarjeta como esta, además de tener el Software apropiado se deben configurar unas claves que permitan superar unas restricciones de seguridad. En el caso de las claves, estas tarjetas manejan dos conjuntos de claves:

- Conjunto de claves 01
  - KLA: 0101010101010101
  - KIC: 01010101010101010101010101010101
  - KID: 01010101010101010101010101010101

- Conjunto de claves 05
  - KID: 2AAA3D9F97C5B849

Para tratar de instalar el Applet se trabajó con dos herramientas: Una fue el "*Interoperable Loader*" (proporcionada por la organización "SIMAlliance" [31]) y otra el "*VIEWS Professional*" en versión Trial (válida solo por unos pocos días). Con el *Interoperable Loader* no se pudo cargar ningún Applet sobre las tarjetas. El *VIEWS Professional* (ver figura 5-1) es una herramienta propietaria de Axalto por lo que si permitió cargar applets sobre las tarjetas USIMERA del mismo fabricante.

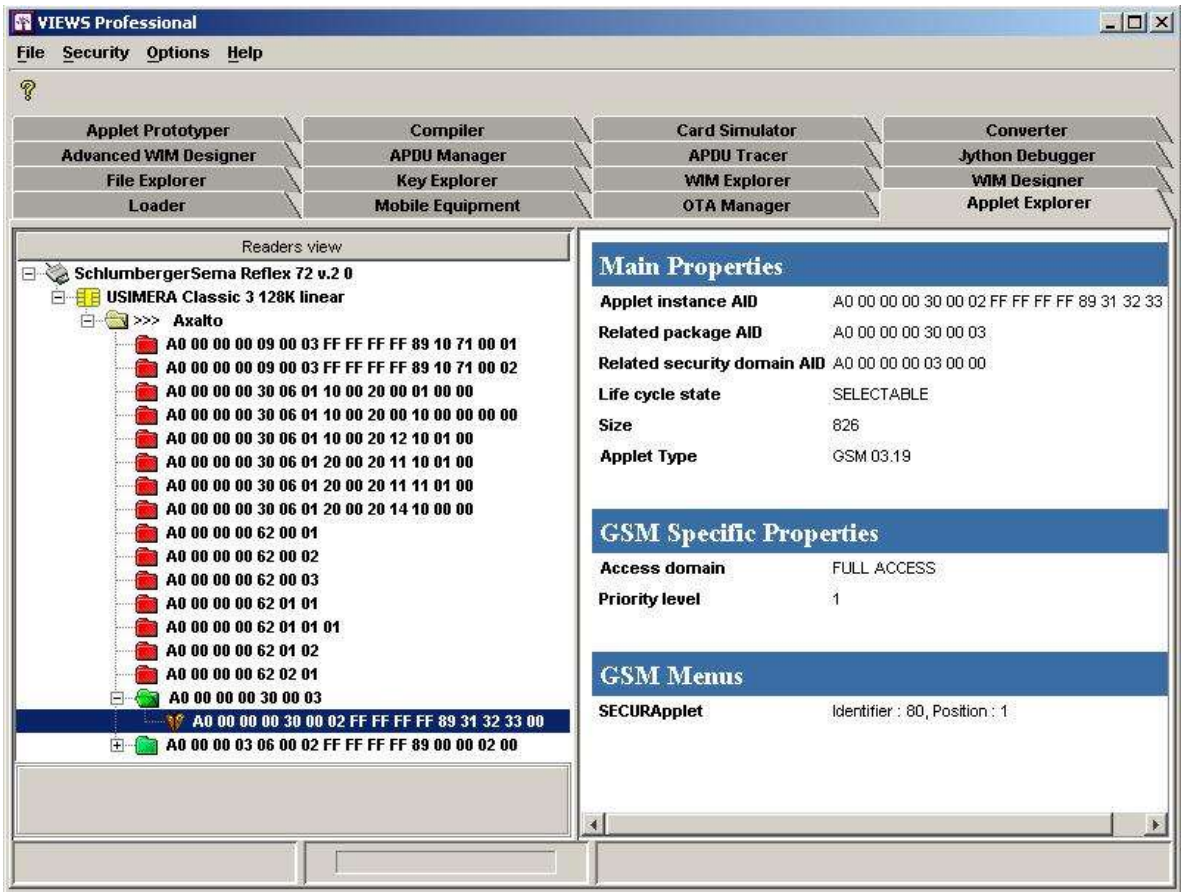

Figura 5-1. VIEWS Professional

Lamentablemente no se pudo instalar el Applet de seguridad con toda su funcionalidad, porque se presentó un problema con la parte de RSA. Sin embargo la parte de acceso a parámetros SIM y a claves simétricas no presentó ningún problema.

6. PRUEBAS DEL PROTOTIPO

6.1. DESCRIPCIÓN DE LAS PRUEBAS

6.1.1. Prueba de registro

Mediante esta prueba se realiza el registro de un cliente móvil con el MIDlet ECommerceMIDlet ante el servidor de comercio electrónico.

En el desarrollo normal de los eventos, el portador del dispositivo móvil con el MIDlet ECommerceMIDlet instalado inicia la aplicación. La aplicación detecta la falta de registro por tanto muestra la interfaz para la realización del registro. El usuario llena el formulario de registro y esta información se envía al servidor para realizar el registro en la base de datos, el resultado de la operación llega al cliente desde el servidor en forma cifrada. Si el cliente es capaz de descifrar la información obtendrá la respuesta del proceso de registro.

6.1.2. Prueba de validación

Mediante esta prueba se realiza la verificación del acceso a los parámetros SIM utilizados para la validación de usuario y la efectiva comprobación y confrontación de los parámetros enviados por el cliente con los parámetros almacenados en el servidor.

Realizamos el procedimiento normal esperando recibir una respuesta positiva y luego deliberadamente cambiamos la información almacenada en el servidor para imposibilitar autenticación exitosa del cliente.

6.1.3. Prueba de adición

Mediante esta prueba se realiza la verificación de la capacidad del cliente para ingresar productos al servicio de comercio electrónico. Para realizar este proceso se diligencia el formulario de adición de productos, luego se firma la información recolectada para posteriormente enviar la información al servidor.

Realizamos el procedimiento normal esperando recibir una respuesta positiva luego deliberadamente cambiamos la clave RSA pública utilizada para descifrar la firma para que de esta forma la integridad de la información se encuentre comprometida.

6.1.4. Prueba de búsqueda

Mediante esta prueba se realiza la verificación de la capacidad del cliente para buscar productos en el servicio de comercio electrónico. Para realizar este proceso se diligencia el formulario de búsqueda de productos, luego se firma la información encontrada para posteriormente enviar la información al cliente.

Realizamos el procedimiento normal esperando recibir una respuesta positiva luego deliberadamente cambiamos la clave RSA privada utilizada para cifrar la firma de tal forma que la integridad de la información se encuentre comprometida.

6.1.5. Prueba de pago

Mediante esta prueba se realiza la verificación de la capacidad del cliente para pagar productos que se desea comprar en el servicio de comercio electrónico. Para realizar este proceso se diligencia el formulario para pago de productos, se cifra esta información para ser enviada al servidor y realizar la transacción.

Realizamos el procedimiento normal esperando recibir una respuesta positiva luego deliberadamente cambiamos la clave RSA pública utilizada para descifrar la información en el lado del servidor para que de esta forma la validez de la misma se encuentre comprometida.

6.1.6. Prueba de gestión de claves

Mediante esta prueba se realiza la verificación de la capacidad del cliente para cambiar su clave de cifrado de información

Realizamos el procedimiento normal esperando recibir una respuesta positiva.

6.1.7. Prueba de acceso a parámetros

Esta prueba se realiza de forma implícita en el desarrollo de las pruebas anteriores.

Pero deliberadamente intentamos realizar el acceso a un parámetro no permitido.

6.2. RESULTADOS DE LAS PRUEBAS DEL SISTEMA

6.2.1. Prueba de registro

Figura 6-1. Prueba de registro.

6.2.2. Prueba de validación

Figura 6-2. Prueba de validación.

6.2.3. Prueba de adición

Figura 6-3. Prueba de adición

6.2.4. Prueba de búsqueda

Figura 6-4. Prueba de búsqueda

6.2.5. Prueba de pago

Figura 6-5. Prueba de pago

6.2.6. Prueba de gestión de clave

Figura 6-6. Prueba de gestión de clave

6.2.7. Prueba sobre teléfono real

Finalmente mencionar que se lograron manejar los comandos SAT en un dispositivo real, gracias a una aplicación cargada sobre las tarjetas Axalto USIMERA.

Figura 6-7. Comando proactivo SIM Application Toolkit

7. CONCLUSIONES Y TRABAJOS FUTUROS

- La implementación de la plataforma P3SIM nos permitió verificar la idoneidad de los parámetros SIM para realizar la validación y autenticación de usuarios ante proveedores de servicios.

- Se logró incursionar con éxito en el desarrollo de aplicaciones para tarjetas SIM.

- La tecnología SAT sigue siendo utilizada ampliamente en redes GSM 2+ pero el despliegue de soluciones bajo este ambiente se encuentra dominado por los operadores, debido a que ellos son lo únicos conocedores de los códigos de seguridad de las tarjetas de sus usuarios móviles.

- Actualmente las dos tecnologías mas usadas en el desarrollo de aplicaciones para tarjetas SIM, SAT y Java Card han llegado a una convergencia. Es decir las aplicaciones SAT se desarrollan utilizando el lenguaje Java Card, concretamente con el API GSM 03.19.

- A pesar de los esfuerzos de los organismos internacionales por lograr una estandarización que permita una total interoperabilidad de las aplicaciones desarrolladas para tarjetas inteligentes y una alta portabilidad sobre dispositivos de diferentes fabricantes, la tecnología Java Card posee algunas limitaciones que dependen del fabricante.

- Las herramientas de libre distribución que se encuentran en el mercado para el desarrollo de aplicaciones sobre tarjetas inteligentes Java Card y SIM Java Card tienen grandes limitaciones.

- Los fabricantes de tarjetas SIM Java Card aprovechan la cercanía con los operadores de telefonía móvil GSM para hacer acuerdos de negocio para el desarrollo de aplicaciones a la medida del operador creando un ambiente cerrado de negocio restringiendo la participación de otros miembros en ciclo de vida de la SIM.

- La utilización de una metodología como el MCS, la programación orientada a objetos y la aplicación de patrones de diseño son una gran ayuda para el desarrollo de proyectos de calidad, con altos niveles de eficiencia y bajo un cronograma controlado.

- Para trabajos futuros se recomienda trabajar con la herramienta propietaria de Gemplus, debido a que el costo en que se incurre inicialmente sopesa con el tiempo invertido y resultados obtenidos posteriormente en la Simulación y despliegue sobre tarjetas SIM reales en Colombia.

- La seguridad es una necesidad básica del usuario de las redes móviles actuales, por lo que estos tipos de trabajo son muy necesarios y comercialmente muy viables.

Desarrollar una plataforma para el acceso seguro a servicios móviles utilizando parámetros SIM no fue nada fácil. Se tuvieron varios inconvenientes de gran envergadura, como lo fueron:

- Trabajar con una tecnología tan reciente como SATSA, que aún no ha entrado a su etapa de madurez, que presenta poca documentación y de la cual se consiguen implementaciones en versiones Beta con posibles errores.

- Imaginar una plataforma funcional que involucraba tecnologías que nadie había manejado en la Universidad del Cauca, y que actualmente en el mundo muy pocas personas son capaces de integrar.

- La falta de herramientas que permitan trabajar de forma eficiente con tecnologías como SAT y Java Card.

Se plantea como trabajo futuro:

- La implementación de un prototipo que además de utilizar la autenticación basada en parámetros SIM, use HTTPS o esquemas de seguridad basados en XML, con el consecuente análisis del ancho de banda utilizado y tiempos de respuesta preferentemente sobre dispositivos y redes reales.

- La ampliación del número de algoritmos manejados por parte del framework P3SIM, es decir hasta ahora sólo se da soporte a DES para el cifrado simétrico y a RSA para el asimétrico, pero Java Card y SATSA brindan la posibilidad de manejar también 3DES, AES, DSA y EC (Elliptic Curve).

- Otra falencia es un manejo más robusto de la firma digital por parte del framework, lo cual es tecnológicamente posible. Es decir manejar más funciones hash, mas mecanismos de padding (relleno) y, si se considera económicamente viable, certificados digitales para cada usuario.

BIBLIOGRAFIA

[1] (Especificación Técnica) 3GPP 11.14. SIM Application Toolkit [doc]. Disponible en http://www.3gpp.org/ftp/Specs/html-info/1114.htm

[2] Java Card, documentación relacionada a este estándar disponible en http://java.sun.com/products/javacard

[3] JSR 177 SATSA, documentación relacionada a este API disponible en http://java.sun.com/products/j2me/satsa

[4] Especificación PC/SC y sus múltiples partes. Puede ser obtenida en: http://www.pcscworkgroup.com/

[5] SHARP e IBM Japan, http://www.sharpsma.com/part.php?PartID=4875

[6] Estándar ISO 7816, http://www.ttfn.net/techno/smartcards/iso7816_4.html

[7] OCF (*Open Card Framework*). http://www.opencard.org

[8] Estándar GSM 11.11 ó 3GPP TS 11.11: Descripción de la interfaz SIM-ME [doc]. Disponible en http://www.3gpp.org/ftp/Specs/html-info/1111.htm

[9] ETSI (European Telecommunications Standardization Institute). http://www.etsi.org

[10] Ambientes de desarrollo Java propietarios: GemXpreso de Gemplus (www.gemplus.com) y Cyberflex de Axalto (www.cyberflex.slb.com).

[11] (Especificación Técnica) 3GPP 11.10 (1999). *Mobile Station Conformity Specification* [doc]. Disponible en http://www.3gpp.org/ftp/Specs/html-info/1110.htm

[12] (Especificación Técnica) 3GPP2 C.S0023-B_v1.0. *Removable User Identity Module for Spread Spectrum Systems* [pdf]. Disponible en: http://www.3gpp2.org/Public_html/specs/C.S0023-B_v1.0_040426.pdf

[13] (Especificación Técnica) 3GPP 22.038 (2004). *Technical Specification Group Services and System Aspects; USIM Application Toolkit (USAT)* [doc]. Disponible en: http://www.3gpp.org/ftp/Specs/html-info/22038.htm

[14] Ciclo de vida de la tarjeta SIM [pdf]. SIMLifeCycleManagement.pdf

[15] (Especificación Técnica) GSM 02.09: Sistema de telecomunicaciones celular digital (fase 2+); aspectos de seguridad. Disponible en http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_11_Mainz/Docs/PDF/S3-000142.pdf

[16] (Especificación Técnica) GSM 03.20 (igual a 3GPP TS 43.020): Funciones de red relacionadas a seguridad. Disponible en http://www.3gpp.org/ftp/Specs/html-info/43020.htm

[17] 3GPP (3rd Generation Partnership Project). http://www.3gpp.org

[18] (Especificación Técnica) ETSI TS 02.19 (similar a 3GPP 22.038): USIM Application Programming Interface (USIM API), Descripción del servicio, escenario 1. Disponible en http://www.3gpp.org/ftp/Specs/html-info/22038.htm

[19] (Especificación Técnica) 3GPP 03.48: Arquitectura de seguridad para SAT, escenario 2. Disponible en http://www.3gpp.org/ftp/Specs/html-info/0348.htm

[20] (Especificación Técnica) 3GPP TS 03.19: SIM API para java card. Disponible en http://www.3gpp.org/ftp/Specs/html-info/0319.htm

[21] (Especificacion Técnica) ETSI TS 102.223 (igual 3GPP 31.111): Smart Cards, USIM application toolkit. Disponible en www.3gpp.org/ftp/Specs/html-info/31111.htm

[22] (Especificación Técnica) 3GPP 23.040: realización Técnica de SMS. Disponible en http://www.3gpp.org/ftp/Specs/html-info/23040.htm

[23] (Especificación Técnica) 3GPP 24.011: soporte PP SMS sobre la interfaz de radio móvil. Disponible en http://www.3gpp.org/ftp/Specs/html-info/24011.htm

[24] (Especificación Técnica) 3GPP 27.007: Conjunto de comandos AT para el Equipo de Usuario. Disponible en http://www.3gpp.org/ftp/Specs/html-info/27007.htm

[25] Scott Guthery – Mary Cronin. Ed McGraw-Hill. Mobile Application Development with SMS and the SIM Toolkit. Diciembre de 2001.

[26] (Especificación Técnica) 3GPP 23.038 Tales tales. Disponible en http://www.3gpp.org/ftp/Specs/html-info/23038.htm

[27] Carlos Enrique Serrano. Universidad del Cauca. Conferencias Modelos para la Construcción de Soluciones. Diciembre de 2002.

[28] EclipseME. Plug-in para Eclipse. Disponible en http://eclipseme.sourceforge.net

[29] Aspects Developer. IDE Java Card. Mayor información en http://www.aspectssoftware.com

[30] Axalto USIMERA 128K. Información de esta tarjeta disponible en: http://www.axalto.com/wireless/usimera.asp

[31] SIMAlliance. http://www.simalliance.org

GLOSARIO

A

ADM: Administrator
APDU: Application Protocol Data Unit
API: Application Programming Interface
APPLET Aplicación para Tarjetas Inteligentes.
ARPU Average Revenue per User
AuC Authentication Center

B

BSS Base Station Subsystem

C

CAD Card Acceptance Device
CAP Converted Applet
CARD OS Card Operating System
CDC Connected Device Configuration
CDMA Code Division Multiple Acces
CDSA Common Data Security Architecture
CHV1 Card Holder Verification 1
CHV2 Card Holder Verification 2
CID Cell ID
CLCD Connected Limited Device configuration
CPU Central Process Unit
CRM Costumer Relationship Management

D

DES Data Encryption Standard
DF Dedicated File
DTMF Dual Tone Multi-Frequency

E

EEPROM Electrically Erasable Programmable Read Only Memory
EF Elementary File
ELP Extended Language Preference
ETSI European Telecommunications Standard Institute

F

FLASH
Firewall

G

GSM Global System for Mobile Communications

H

HALF DUPLEX
HLR Home Local Register

I

IMEI International Mobile Equipment Identifier
ISO International Standard Organization

J

JCRE Java Card Runtime Environment
JCRMI Java Card Remote Method Invocation
JCVM Java Card Virtual Machine

K

L

M

MCC Mobile Country Code
ME Mobile Equipment
MF Master File
MIDP Mobile Information Device Profile
MMS Multimedia Message Service
MNC Mobile Network Code
MS Mobile Station
M2M Machine to Machine

N

NMR Network Measurement Results
NSS Network Switching Subsystem

O

OCF Open Card Framework
OSS Operation and Support Subsystem
OTA Over the Air

P

PIN Personal Identification Number
PS/SC Personal Computer / Smart Card
PKCS#11 Cryptographic Token Interface Standard
PKI Public Key Infrastructure

Q

R

RAM Random Access Memory
Roaming
ROM Read Only Memory
RSA algoritmo Rivest-Shamir-Adleman
RUIM Removable User Identity Module

S

SATSA Security and Trust Services API
SIM Subscriber Identity Module
SMS Short Message Service
SMS-PP Short Message Service Point to Point
Smart Card tarjeta inteligente

SS Suplementary Service

T
TA Time Advance)
TPDU Transport Protocol Data Unit
TS Technical Specification

U
UICC Universal Integrated Circuit Card
USIM Universal Subscriber Identity Module
USSD Unstructured Supplementary Service Data

V
VLR Visitors Location Register

W

X

Y

Z