UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA

PROYECTO FIN DE CARRERA

Ingeniería Técnica Industrial: Electrónica Industrial

SISTEMA DE GESTIÓN SANITARIA MEDIANTE TARJETAS JAVACARD

AUTOR: Pablo de Antonio Herrero

TUTOR: Raúl Sánchez Reíllo

Octubre 2013

Agradecimientos

Sin duda alguna este apartado merecería un proyecto aparte para poder dar las gracias a todas esas personas que me han ayudado a lo largo de todos estos años y me han acompañado hasta aquí. Estoy seguro de que me olvido de alguien; perdón de antemano ☺

A mis padres, Pablo e Isabel, por su infinita paciencia, su cariño, su apoyo, por siempre estar ahí, y por repetirme indefinidamente: *"¡Termina el proyecto!*". Si he llegado hasta aquí es solo gracias a vosotros: **Gracias**.

A Miriam, por todo lo que has sido y por lo que seguirás siendo: **Gracias**

A mi hermano Sergio y resto de la familia, por aguantarme y por estar ahí: **Gracias**.

A la pandilla del barrio, por acompañarme en todos esos momentos en los que estás hasta el gorro de todo y de todos: **Gracias**.

A la gente de VALID (Cortés, Guaje, Soto y compañía), por no solo ser compañeros de trabajo si no grandes amigos: **Gracias**.

A la gente de Galve de Sorbe, por esas tardes de botellines haciéndome olvidar el estrés de la ciudad y de los estudios: **Gracias**.

A Raúl, mi tutor, por su soporte, por su ayuda y por aguantar mis retrasos en las respuestas a sus mails: **Gracias**

Muchas GRACIAS a todos.

Contenido

Índice de acrónimos

- AID: Applet IDentifier.
- APDU: Application Protocol Data Unit.
- API: Application Program Interface.
- CIE: Código Internacional de Enfermedades.
- EEPROM: Electrically Erasable Programable Read Only Memory.
- GP: GlobalPlatform.
- ISD: Issuer Security Domain.
- JC: JavaCard.
- NFC: Near Field Communication
- NVM: Non Volatil Memory.
- ROM: Read Only Memory.
- SD: Secuirty Domain.
- SIM: Subscriber Identity Module
- SOTI: Sistema Operativo de Tarjeta Inteligente.
- TI: Tarjeta Inteligente

Índice de Figuras

Índice de tablas

1. Introducción

1.1 Objetivos

El objetivo de este proyecto es crear un nuevo sistema de gestión para la Seguridad Social, basado en tarjetas inteligentes.

Los componentes a desarrollar son un applet basado en Java Card, una herramienta capaz de gestionar y actualizar la información almacenada por dicho applet. Así como el establecimiento de un entorno de gestión de la Tarjeta Inteligente para facilitar las operaciones de borrado e instalación del applet.

Tras el desarrollo de todos los objetivos del proyecto, un usuario será capaz, con la ayuda de un equipo basado en Windows, un lector de tarjetas y la propia Tarjeta Inteligente, de instalar el applet en la tarjeta, así como personalizarla con la información deseada a través de la herramienta que hará de interfaz de usuario.

Tres tipos de usuarios que pueden interactuar sobre la tarjeta, de acuerdo a la Figura 1:

- *Centro médico*: Con acceso total a los datos personales, así como con permisos para modificar alguno de esos datos.
- *Farmacia*: Con acceso a los datos relacionados con las recetas introducidas por los centros médicos, pudiendo marcarlas como despachadas una vez se hayan entregado al ciudadano.
- *Ciudadano*: Solo tendrá acceso a ciertos datos y sin la posibilidad de modificarlos.

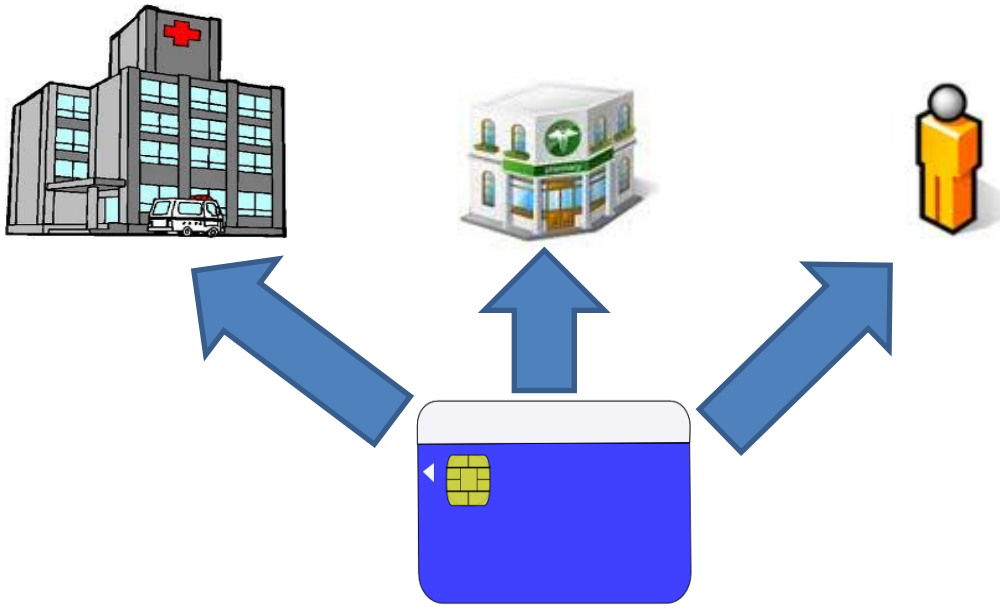

Figura 1. Presentación general de los usuarios

1.2 Motivación

Varias son las razones que motivan e impulsan este proyecto:

- Actualmente las tarjetas de banda magnética solo permiten accesos de lectura, a muy poca información. Las tarjetas inteligentes permiten modificar datos sin necesidad de fabricar una tarjeta nueva. Además, gracias a los sistemas criptográficos de que disponen estas tarjetas, el almacenamiento y acceso a dicha información siempre es de forma segura.

- Sería muy útil en caso de emergencia ya que cualquier ambulancia, con un simple ordenador y un lector de tarjetas podría acceder a información vital como puede ser el grupo sanguíneo del individuo, alergias a medicamentos, historial médico,…

- Se ahorrarían errores y costes a la hora de recetar, no será necesario escribir las recetas en papel.

- El ciudadano podrá conocer su propia información, muy útil por ejemplo para personas dependientes, ya que la persona responsable podrá saber en cualquier momento si padece alguna alergia importante, cada cuanto y en que dosis debe tomar el medicamento “x”,…

1.3 Estructura del documento

El proyecto se divide en una serie de capítulos para facilitar y estructurar su lectura. El primer capítulo (el actual) es una básica introducción con el objetivo de introducir al lector en las razones que han llevado a este proyecto, así como una breve descripción de los actores que intervendrán de aquí en adelante.

El capítulo 2 describe el actor central y eje del proyecto: la Tarjeta Inteligente. Haciendo un repaso a sus orígenes, su sistema operativo, los estándares internacionales que rigen su comunicación, su lenguaje de programación y el manejo de aplicaciones.

El siguiente capítulo (número 3), es el más importante del proyecto, ya que recoge todo el diseño del mismo, incluyendo las estructuras de datos que almacenará el applet.

El capítulo 4 expone toda la información referente a las herramientas usadas durante el desarrollo: GP Shell, Netbeans y los proyectos que contienen, tanto el applet, como la interfaz de usuario.

El capítulo 5, contiene la descripción de todos los comandos que pueden ser enviados a la tarjeta. Este API de comandos contiene tanto los comandos de seguridad, como el set de comandos dedicados a la gestión de la información almacenada en la tarjeta.

El capítulo 6 hace una descripción del applet de gestión sanitaria que se cargará en la Tarjeta Inteligente. Este capítulo detalla la jerarquía entre las distintas clases que componen el applet.

El siguiente capítulo, número 7, se centra en la herramienta que hará las veces de interfaz de usuario y que permitirá la visualización/modificación de los datos almacenados en el applet instalado en la tarjeta.

El último capítulo, número 8, recoge las conclusiones obtenidas así como las futuras líneas de desarrollo.

2. Tarjetas inteligentes y sus estándares

2.1 Pasado y presente de la Tarjeta Inteligente.

Se podría decir que el antepasado de lo que hoy conocemos como Tarjeta Inteligente fueron las tarjetas de visita, fabricadas en papel. Al fin y al cabo se trataba de almacenar cierta información y poder leerla.

Estas tarjetas evolucionaron para dar lugar a los primeros modelos que incorporaban una banda magnética, capaz de almacenar hasta 225 caracteres y que fueron usadas mayoritariamente por el sector bancario.

Estas tarjetas presentaban varios inconvenientes, como la escasa capacidad de almacenamiento, su fácil desmagnetización al ser introducidas en un campo magnético, y últimamente la facilidad que presentaban para ser clonadas, siendo un medio perfecto para defraudar a las entidades bancarias.

La principal ventaja de estas tarjetas es su escaso coste para producirlas, así como, la gran aceptación que han tenido por parte de los usuarios.

Todas las desventajas enumeradas anteriormente, llevaron a una evolución de las tarjetas de banda magnética, incorporando un chip a estas y dando lugar a lo que hoy en día conocemos como Tarjeta Inteligente.

Dependiendo del chip incorporado podemos tener Tarjetas Inteligentes con capacidad únicamente de almacenamiento (memorias), o con capacidad de procesamiento. Estas últimas son la base de este proyecto y de aquí en adelante el término Tarjeta Inteligente hará referencia a este tipo concreto de tecnología.

Las Tarjetas Inteligentes fueron inventadas en los años 70, aunque existen discrepancias acerca de dónde y quien las inventó. Lo que sí está claro es que su primer uso, de carácter público conocido, fue como medio de prepago telefónico en Francia en el año 1983.

Las principales ventajas que ofrecen este tipo de tarjetas son:

- Mayor capacidad de almacenamiento. Los últimos modelos llegan a ofrecer varios megabytes de capacidad de memoria no volátil.
- Capacidad de implementar lógica. No son solo memorias insertadas en un cuerpo de plástico, si no que tienen capacidad de procesamiento.
- Hardware criptográfico. La mayoría de los modelos incorporan aceleradores criptográficos capaces de soportar algoritmos del tipo DES, AES, RSA,…

2.2 Tipos de Tarjetas Inteligentes

Basándose en el modo de comunicación con el mundo exterior, se puede hacer una clasificación de las mismas en:

- Con contactos: requieren la inserción en un lector de tarjetas para ser procesadas. La comunicación se realiza mediante unos contactos metálicos que posee. Las tarjetas de contacto tienen ciertas limitaciones. Con el paso del tiempo estos contactos se desgastan, provocando un mal funcionamiento de la tarjeta.

  Este protocolo de comunicación esta estandarizado en la norma ISO/IEC 7816.

  Inicialmente estas tarjetas presentaban 8 contactos; pero la industria ha ido evolucionando y las últimos modelos solo tienen 6 contactos. En la siguiente figura podemos ver el típico esquema de contactos de una Tarjeta Inteligente:

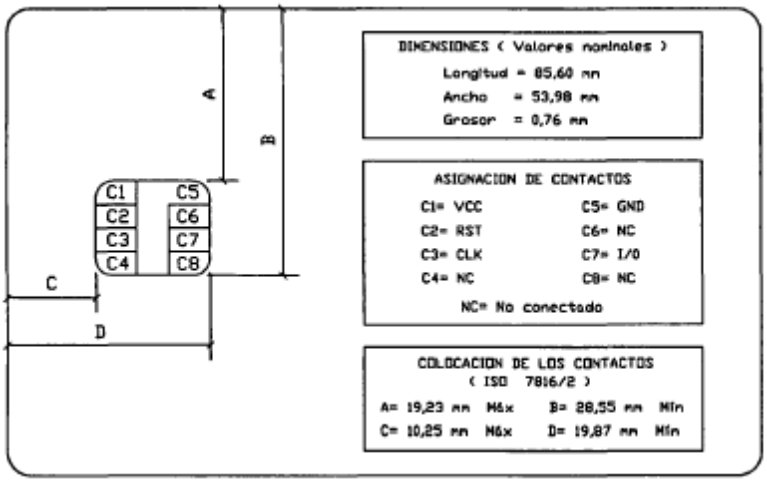

Figura 2: Esquema de una tarjeta de 8 contactos (Fuente [ISO/IEC 7816-1]).

- Contactless: Estas tarjetas incorporan una antena empotrada en el cuerpo de plástico de la propia tarjeta, haciendo posible la comunicación con el lector por aproximación de la tarjeta a este, sin necesidad de contacto físico entre ambos. Toda la comunicación está basada en radiofrecuencia (ISO/IEC 14443) haciendo que el desgaste de la tarjeta sea mucho menor que una basada en contactos.
- Dual interface: Son las últimas en aparecer en el mercado de las Tarjetas Inteligentes, y combinan ambos sistemas, unos contactos para poder introducir la tarjeta en un lector, así como, una antena embebida para posibilitar la comunicación contactless.

2.3 Componentes de una Tarjeta Inteligente.

La siguiente figura esquematiza los distintos componentes que integran una Tarjeta Inteligente:

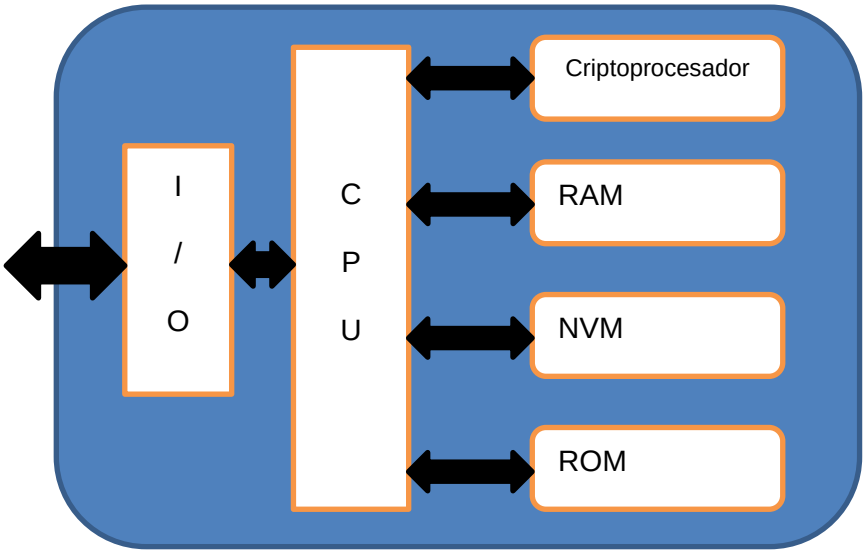

Figura 3: Componentes de una TI

Partiendo de este esquema, ya se puede observar que el acceso a las diferentes memorias no se puede efectuar de forma directa y siempre exige la participación de la CPU, lo que aporta un extra de seguridad, ya que la CPU puede hacer uso del criptoprocesador para securizar todos estos accesos.

2.3.1 CPU

La CPU es el componente más importante de la Tarjeta Inteligente, se encarga de ejecutar las operaciones a bajo nivel, interconectando entre sí los distintos bloques funcionales de la tarjeta. El sistema operativo de la tarjeta permite controlar el conjunto de operaciones que se pueden realizar dentro de la tarjeta, de manera que el bloque de la CPU pasa prácticamente desapercibido para el usuario final.

Las primeras CPUs se han basado en microprocesadores de 8 bits, como la familia 8051 de Intel, o la familia 68HC05 de Motorola. En otros ámbitos como en los ordenadores personales los microprocesadores han sufrido grandes evoluciones, apareciendo procesadores de 64 bits. En el campo de las tarjetas inteligentes la

evolución no ha sido tan notoria, aunque han aparecido procesadores de 16 bits, e incluso algunos de 32 bits con tecnología RISC (Reduced Instruction Set Code), sobre todo en los últimos años. Sin embargo, hoy en día, la mayoría de las tarjetas siguen utilizando procesadores de 8 bits, dado que las aplicaciones que se instalan habitualmente en las tarjetas inteligentes no necesitan mayor potencia de cálculo para realizar satisfactoriamente su tarea, y el precio de cambio de adoptar procesadores más potentes resulta demasiado elevado. Por otra parte la evolución de la tecnología ha permitido la aparición de coprocesadores matemáticos en ciertas tarjetas, que permiten realizar los cálculos de forma mucho más rápida. También se han introducido bloques adicionales con funcionalidades criptográficas, así los algoritmos de cifrado requeridos para operaciones seguras pueden ser realizados en muy poco tiempo, una cantidad mucho menor que la necesaria si todo el cálculo lo tuviera que realizar directamente el microprocesador.

2.3.2 Bloque I/O

Es la parte encargada de comunicarse con los lectores basándose en una comunicación estándar serie.

Este bloque, básicamente, enrutará los datos de entrada a la CPU y enviará al lector la respuesta de ésta.

2.3.3 Memoria RAM

Este tipo de memoria es volátil, es decir, su contenido es reseteado al quitar la alimentación de la tarjeta. Esta memoria es ideal para almacenar información a la que se accede frecuentemente, ya que su acceso es mucho más rápido que el acceso a la memoria no volátil o información que no se deba almacenar de forma permanente, como las claves de sesión.

Actualmente el tamaño de la memoria RAM suele rondar el orden de unos pocos kilobytes, oscilando entre 4 kb de las tarjetas menos potentes hasta los 30 kb de los últimos prototipos.

2.3.4 Memoria NVM

La memoria NVM (Non Volatile Memory) representa el 80% de la memoria total disponible en el sistema siendo la más usada por el programador. Históricamente esta memoria era de tipo EEPROM, es decir, memoria no volátil escribible eléctricamente; pero en la actualidad el mercado ha evolucionado hacia las memorias FLASH, mucho más baratas y con mayor densidad de almacenaje que las memorias EEPROM. A cambio, esta memoria presenta problemas de durabilidad y “performance”, es decir, su acceso es 10 veces más lento que el acceso a memorias EEPROM.

El rango de tamaños para estas memorias oscila entre los 8 kb de los primeros modelos hasta los 1024 kb de los últimos modelos.

2.3.5 Memoria ROM

Esta memoria es de solo lectura y aquí reside el propio sistema operativo de la tarjeta, así como las rutinas de arranque de la Tarjeta Inteligente y el “bootloader” que permite la carga del sistema operativo.

El tamaño de estas memorias suele estar por debajo de los 300 kilobytes, llegando en los últimos modelos a unos pocos bytes, los justos para almacenar solamente las rutinas de arranque y el “bootloader”, ya que el sistema operativo se ha movido a la zona NVM.

La razón de esta migración la encontramos en los fabricantes de tarjetas, al mover el sistema operativo a la memoria NVM pueden corregir bugs y recargar el nuevo sistema operativo fácilmente, cosa que con el SO en ROM no se podía hacer.

2.3.6 Sistema Operativo de la Tarjeta Inteligente (SOTI)

El sistema operativo es lo que hace que una tarjeta sea inteligente o no. Básicamente el SOTI es un interfaz de alto nivel que simplifica el uso de la tarjeta. Son sistemas muy parecidos a los que nos podemos encontrar en un pc y sus funciones principales son:

- Inicializar la tarjeta.
- Gestionar el intercambio de datos con el exterior.
- Manejar el uso de las memorias de la tarjeta: Almacenando o borrando datos.
- Controlar toda la seguridad de la tarjeta: Almacenando claves criptográficas, evitando accesos no permitidos,…

2.4 ISO 7816

2.4.1 Introducción

La ISO 7816 es un estándar internacional que rige las tarjetas inteligentes. Esta norma está controlada por el Organismo Internacional de Estandarización, cuyo acrónimo en inglés se corresponde con ISO.

Este estándar se divide en varias partes; pero las que más impactan sobre este proyecto son las cuatro primeras:

- 7816-1: Características físicas.
- 7816-2: Tarjetas con contactos: Dimensiones y localización de los contactos.
- 7816-3: Características eléctricas.
- 7816-4: Organización, la seguridad y los comandos para el intercambio de la información

De estas cuatro partes la más importante para el desarrollo de este proyecto es la número cuatro que define la estructura básica de comunicación de la tarjeta con un

lector: el APDU (Application Protocol Data Unit), que se definirá en el siguiente capítulo.

2.4.2 APDU

2.4.2.1 Estructura de un APDU

Cada comando estará definido de acuerdo a la norma ISO 7816-4 como ilustra la siguiente tabla:

Campo	Descripción	Número de bytes	Dirección
Cabecera del comando	Byte de clase **CLA**	1	A la tarjeta
	Byte de instrucción **INS**	1	
	Bytes de parámetros **P1-P2**	2	
Lc	Longitud del campo de datos **Lc**	1	
Campo de datos	Datos	Lc bytes	
Le	Longitud de los datos de respuesta	1	

Tabla 1: Estructura básica de un APDU

Y las respuestas estarán compuestas por un campo de datos más un Status Word (SW) de 2 bytes.

Desglosando los bytes que componen la cabecera:

- CLA byte:

El byte de clase (CLA) es el primer byte de la cabecera y contiene información general del comando, como el canal al que va dirigido, la seguridad con la que ha sido generado,…

Todos los comandos descritos en este documento serán de carácter propietario por lo que el bit8 de todos los bytes de clase deberán tener su bit más significativo a 1.

La siguiente tabla muestra la codificación exacta del CLA byte, dónde además del bit8 son significativos:

- El bit3 que indica que el comando está securizado (como se describe más adelante).
- Los bits2 y bit1, que juntos forman el canal lógico sobre el que se envía el comando.

b8	b7	b6	b5	b4	b3	b2	b1	Significado
1								Comando propietario
	0							No usados
		0						
			0					
				0				
					1			Mensaje seguro propietario
						x	x	Canal lógico(entre 0 y tres)

Tabla 2: Codificación del CLA byte.

- INS byte

El byte de instrucción (INS byte) indica el comando a procesar, de acuerdo a la especificación ISO 7816-3, los valores ‘6x’ y ‘9x’ no serán usados, además no se utilizarán los valores indicados en la tabla 4.2 de la especificación ISO 7816-4.

El bit 1 indicará el formato del campo de datos:

- Si el bit b1 es 0 no se proporciona ninguna información sobre el formato.
- Si el bit b1 es 1, el campo de datos será codificado como BER-TLV.

- P1 y P2 bytes

Los bytes de parámetros P1 y P2 son específicos de cada comando y están definidos en el capítulo 5.

- Lc

Este byte indica la longitud del campo de datos, en caso de que haya datos entrantes.

2.4.2.2 Tipos de APDUs

Dado que Lc, el campo de datos y Le son campos opcionales, se pueden dar diferentes configuraciones para un APDU. Atendiendo a la presencia de estos campos, los tipos de APDU aparecen reflejados en la siguiente figura:

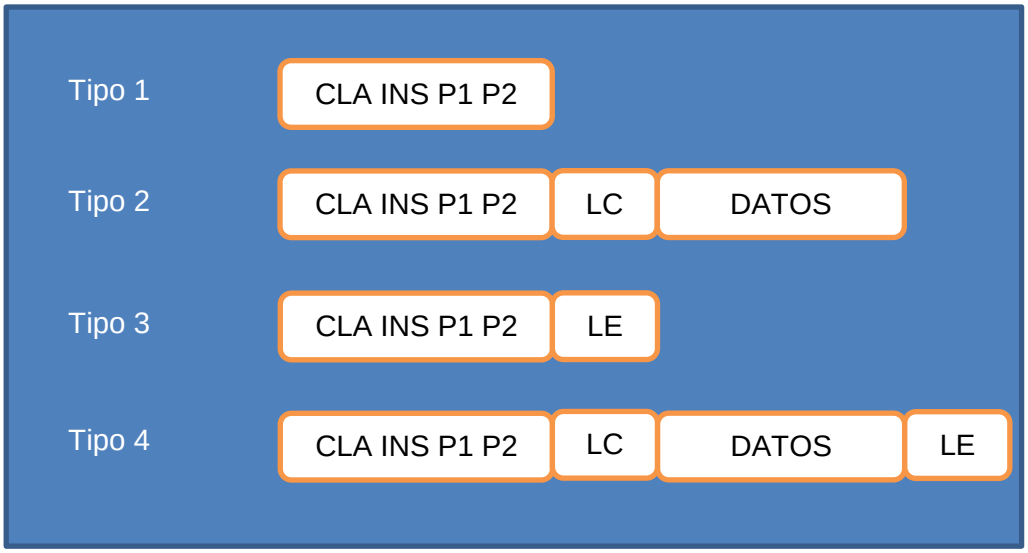

Figura 4: Tipología de un APDU

2.4.3 JavaCard

2.4.3.1 Introducción

JavaCard es una tecnología que permite ejecutar de forma segura pequeñas aplicaciones, denominadas “applets” en una Tarjeta Inteligente.

Básicamente, consiste en un subconjunto de Java, es decir, todas las instrucciones y métodos que componen JavaCard forman parte del conjunto de Java. De hecho un fichero de JavaCard se compila como se haría con cualquier fichero java, obteniendo un fichero .class que es el que se convierte para poder ser cargado en una Tarjeta Inteligente. Este último paso es el diferenciador, aunque la última versión de JavaCard 3.0.4 Connected Edition, directamente ofrece la posibilidad de cargar ficheros con extensión .class en una Tarjeta Inteligente. De momento ningún fabricante de tarjetas ofrece esta plataforma, aunque en un futuro no muy lejano se podrá hacer uso de ella.

Dos fueron las claves de la popularización de esta tecnología: su portabilidad y su seguridad.

La primera de ellas hace que cualquier applet JavaCard pueda ser ejecutado en cualquier plataforma compatible, independientemente del fabricante de la tarjeta.

La segunda se hereda de la propia filosofía de Java, donde se puede limitar el acceso entre clases. Y se ve potenciada por la gran cantidad de algoritmos criptográficos que ofrece JavaCard.

2.4.3.2 Paquetes y applets

Los applets son las aplicaciones que se ejecutan en una Tarjeta Inteligente, y estos applets están contenidos dentro de paquetes. Un paquete puede contener uno o varios applets y el mismo applet no puede estar contenido en dos paquetes distintos. Esta última condición es controlada a través del AID (“Application IDentifier”) de un applet.

El AID de un applet es único y en el proceso de carga e instalación (ver 2.4.4.4 y 2.4.4.5) de una aplicación la tarjeta comprueba que el AID que la identifica no coincida con el de los applets ya cargados en la tarjeta.

Todo applet implementa un método que le sirve de punto de comunicación con el exterior, este método es el `process (APDU apdu)`. Este método tiene como

parámetro de entrada el objeto APDU, que contiene la información descrita en el apartado 2.4.2.1.

2.4.4 GlobalPlatform

2.4.4.1 Introducción

GlobalPlatform es una organización internacional independiente que nació en 1999 recogiendo el testigo y las especificaciones de VISA OpenPlatform. Con el paso del tiempo ha ido creciendo y definiendo más especificaciones, no solo relativas a la propia Tarjeta Inteligente, si no que se han ido extendiendo hasta los dispositivos móviles, servidores de sms, etc.

En la actualidad cuenta con más de 100 miembros entre los que se encuentran los principales fabricantes de tarjetas, chips, teléfonos móviles,…

De todas las especificaciones definidas por GlobalPlatform la que más afecta a este proyecto es la “Card Specification” que define los componentes de una Tarjeta Inteligente, la relación entre ellos así como los mecanismos de seguridad que protegerán a la tarjeta y los datos que contiene a lo largo de la vida de esta última.

Además de definir la manera de manejar el contenido de la tarjeta de una manera segura e independiente del fabricante de la misma lo que proporciona una plataforma neutral e independiente.

2.4.4.2 Dominios de Seguridad

GlobalPlatform define una tipo de aplicación especial denominada Dominio de Seguridad, o SD (Security Domain). Estas aplicaciones se caracterizan por tener privilegios especiales que otras aplicaciones no pueden tener y que les permiten gestionar de forma segura la comunicación de la tarjeta con el mundo exterior, así como el manejo del contenido de esta.

De forma obligatoria, hay un SD que siempre está presente en la tarjeta ya que viene precargado por el fabricante de la misma. Este SD se denomina ISD (Issuer Security Domain) y es una aplicación que siempre estará presente en la tarjeta, ya que no puede ser borrado, y su ciclo de vida representa el de la tarjeta.

Además del ISD puede haber más SDs representando a los distintos proveedores de aplicaciones que operan sobre la tarjeta.

De forma genérica, los SDs son los poseedores de las claves necesarias para establecer una comunicación segura con el exterior. Estas claves solo son conocidas por el poseedor del SD, de modo que solo el podrá establecer una sesión segura para poder manejar el contenido de la tarjeta, ya sea instalando nuevos applets, como borrando antiguos, como cifrar la comunicación de estos para evitar poner en peligro información confidencial,…

2.4.4.3 Canal seguro

Este canal seguro consiste en la mutua autenticación de los actores: Por un lado está el SD y en el otro el “host” o usuario exterior.

Esta autenticación se inicia por el host cuando manda el primero de los comandos de autenticación, “INITIALIZE UPDATE” (ver 5.1.1) que es respondido por el SD. Con la respuesta del SD y las claves adecuadas el host puede componer el segundo de los comandos EXTERNAL AUTHENTICATE (ver 5.1.2), que es chequeado por el SD confirmando así la autenticación o no.

Las claves necesarias para establecer este canal seguro son dinámicas, es decir, se recalculan cada vez que se inicia un canal nuevo a partir de las claves estáticas que almacena el SD y que conoce el host. El proceso para generar estas claves está especificado en [GP spec].

Aplicado a este proyecto, la siguiente figura representa el flujo de comunicaciones a través de un canal seguro entre el host o usuario y el applet residente en la tarjeta:

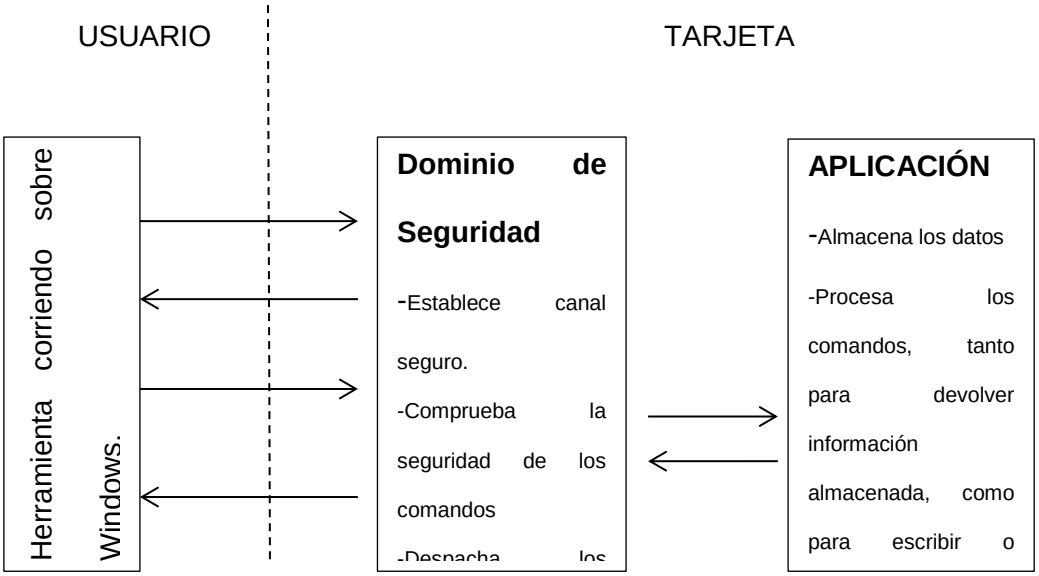

Figura 5: Flujo de un canal seguro

2.4.4.4 Carga y borrado de paquetes

GlobalPlatform define un mecanismo para la carga y borrado de paquetes, de forma que cualquier paquete JavaCard puede ser cargado en cualquier Tarjeta Inteligente compatible con GlobalPlatform.

La secuencia para dicha carga empieza con la apertura de un canal seguro, como define el apartado anterior. Tras esto es enviado un comando INSTALL for LOAD (ver [GPCS] capítulo 9.5), y luego la secuencia de comandos LOAD (ver [GPCS] capítulo 9.6) que contienen el fichero .cap que contiene toda la información del paquete.

De forma similar a la carga de paquetes funciona el borrado de los mismos, comenzando por la apertura de un canal seguro, seguido de un comando DELETE (ver [GPCS] capítulo 9.2).

2.4.4.5 Instalación y desinstalación de applets.

Tras la correcta carga de un paquete, surge la necesidad de poder instalar alguno de los applets que pueda contener el paquete. Esto se consigue, siempre tras la apertura

de un canal segura, con otro comando GlobalPlatform, concretamente el INSTALL for INSTALL (ver [GPCS] capítulo 9.5).

La desinstalación de un applet es básicamente el borrado del mismo, usando el comando DELETE (ver [GPCS] capítulo 9.2).

3. Diseño de la solución y estructuras de datos

3.1 Descripción de componentes

La solución completa se compone de estos actores:

- Herramienta de gestión: Es una interfaz gráfica capaz de correr sobre cualquier máquina con Windows, encargada de intermediar entre los distintos usuarios y el applet residente en la tarjeta, a través de comandos enviados a la tarjeta vía un lector. Los detalles de esta herramienta están descritos en el capítulo 7 de esta memoria.
- Applet sanitario: Es la aplicación instalada en la tarjeta y que se encargará de almacenar toda la información necesaria, así como de chequear los permisos del usuario para modificar o acceder a dicha información (ver capítulo 6).
- Lector de tarjetas: Es el nexo de unión entre la herramienta de gestión y la Tarjeta Inteligente. Para este proyecto se ha usado, en concreto, el “SmartCard Reader ACR38”; pero, cualquier lector de tarjetas PC/SC (Personal Computer / Tarjeta Inteligente) podría ser usado en su lugar.

3.2 Tarjeta

Los requisitos que debe cumplir la Tarjeta Inteligente son:

- JavaCard: Evidentemente, es necesario que la tarjeta sea compatible con JavaCard, aunque no es necesaria ninguna versión especial. Por tanto cualquier Tarjeta Inteligente, cuyo sistema operativo sea compatible con JavaCard 2.1.1, JC 2.2, JC 2.2.2 o JC 3.0.4 es válida.
- GlobalPlatform: Básicamente sólo es necesario que esté presente el Issuer Security Domain (ISD), una entidad que es obligatoria para cualquier tarjeta GlobalPlatform (GP). Este modelo permite cargar 3 claves distintas para poder

diferenciar el tipo de usuario (doctor, farmacia o usuario normal) que accede a la tarjeta y controlar el acceso a la información almacenada.

En base a los APIs (Application Program Interface), está claro que es más completo y ofrece más métodos el de GP v2.2; pero los métodos expuestos por el API GP v2.1.1 son suficientes, ya que cubren todos los métodos necesarios para la apertura de un canal seguro.

Con todo esto cualquier tarjeta GP 2.x del mercado sería válida; pero por simplicidad, economía y facilidad de obtención se usarán tarjetas GP v2.1.1.

- Capacidad: Este es el requisito más relevante a la hora de elegir una tarjeta para esta solución, ya que de él depende la cantidad de información que podrá almacenar el applet. Aunque para definir este valor es necesario primero saber qué información guardará la tarjeta, como se expone a continuación.

3.3 Información almacenada

3.3.1 Datos Personales

Campo	Descripción	Máximo tamaño (bytes)
Nombre	Datos personales del usuario	30
Apellidos		60
Provincia		20
Localidad		30
Calle/ Avenida/ Plaza		40
Número		2
Piso/ Escalera /Puerta		2
Teléfono		10

Teléfono emergencia	Teléfono de la persona que será avisada en caso de emergencia	10
Nº Seguridad Social	Numero de afiliación a la Seguridad Social	8

Tabla 3: Datos personales

Tanto el usuario final como la farmacia solo tendrán acceso de lectura sobre toda esta información, salvo el campo “Teléfono de emergencia” que podrá ser modificado por el propio usuario.

Todos estos datos pueden ser tanto leídos como modificados por el usuario “Hospital”.

3.3.2 Información alérgica

Cada alergia es codificada con tres bytes, el primero de ellos indica el tipo o grupo al que pertenece la alergia, y los otros dos definen un código único dentro de ese grupo, de forma que cada alergia es fácilmente codificable con solo tres bytes.

Partiendo de datos estadísticos, se dimensionará el tamaño de toda esta información acorde a un máximo de 200 alergias por individuo, lo que lleva a un tamaño máximo de 600 bytes para almacenar la información relativa a las alergias.

Tipo de alergia	Codificación tipo (1 byte)	Código alergia (2 bytes)
Medicamentos	‘01’	‘xxxx’
Polen	‘02	
Ácaros	‘03’	
Animales	‘04’	
Alimentos	‘05’	
Picaduras de insectos	‘06’	

Tabla 4: Alergias

3.3.3 Recetas

Este apartado está dedicado a los medicamentos que serán recetados por el usuario "hospital", despachadas por el usuario "farmacia" y consultadas por el usuario "ciudadano". Atendiendo a esto solo el "hospital" será capaz de modificar cualquiera de los campos descritos, la "farmacia" solo podrá marcar el campo de "receta despachada" cuando el usuario acuda a por ella; y el "ciudadano" solo tendrá permiso de lectura sobre todos estos datos, de forma que pueda consultar la dosis, frecuencia, duración del tratamiento…

La tabla 5 muestra toda la información necesaria para cada receta, empezando por el código de medicamento. Éste campo debe estar preparado para los más de 30.000 medicamentos que pueden ser recetados en España; así que con tres bytes será suficiente dejando además la puerta abierta a que esta lista siga creciendo.

El resto de los campos, junto a su significado, se encuentran descritos en la siguiente tabla:

Campo	Descripción	Codificación
Medicamento	Código hexadecimal del medicamento.	3 bytes.
Cantidad	Formato y cantidad del medicamento a despachar.	4 bytes en total. Los dos primeros indican el formato de la tarjeta (ej. '0004': comprimidos de 650 mg.). Los otros dos bytes indican la cantidad.
Dosis	Dosis de medicamento en cada toma.	1 byte. Ya que el formato viene indicado en el

		campo anterior.
Elementos por toma	Número de elementos por toma.	1 byte.
Tomas	Dosis y frecuencia de cada toma.	2 bytes. El primer byte indica la cantidad de dosis de cada toma y el segundo hará referencia a la frecuencia entre tomas. (ej.: '020A': 2 dosis cada día).
Duración	Duración del tratamiento.	2 bytes. El primer byte representa el número y el segundo la unidad de tiempo (día, semana, mes, etc.)
Renovación automática	El flag de receta crónica o renovación automática indica que el tratamiento requiere de más de una receta igual. Este flag posibilita que la herramienta de gestión instalada en la "farmacia" calcule de forma automática la fecha de la siguiente receta y la almacene en el siguiente campo.	1 byte.
Fecha siguiente receta	Almacena la fecha en que el "ciudadano" deberá acudir de	3 bytes. (día/mes/año).

	nuevo a la farmacia.	

Tabla 5: Recetas

Las siguientes tablas definen la codificación del formato de los medicamentos, por ejemplo, los valores contenidos en los dos primeros bytes del campo “Cantidad”:

Valor	Significado
‘01’	Mililitros
‘02’	Miligramos
‘03’	Gramos
‘04’	Comprimidos
‘05’	Inyecciones

Tabla 6: Codificación del formato

Las unidades de tiempo están especificadas en la siguiente tabla:

Valor	Significado
‘01’	Horas
‘02’	Días
‘03’	Semanas
‘04’	Meses
‘05’	Años

Tabla 7: Codificación de las unidades de tiempo

En base a todo lo anterior se puede concluir que para gestionar cada receta son necesarios 18 bytes, asumiendo un máximo de 30 recetas en cada visita al doctor, con 540 bytes es suficiente para gestionar las recetas.

3.3.4 Cartilla de vacunación

Cada vacuna es codificada con tres bytes; el primero de ellos contiene el código de la vacuna, lo que implica que se podrán codificar hasta 255 vacunas. El segundo byte representa el número total de dosis que han de ser suministradas y el tercer byte indica cuantas dosis han sido aplicadas ya sobre el paciente.

Campo	Descripción	Codificación
Vacuna	Código de la vacuna	1 byte
Dosis totales	Número de dosis a aplicar	1 byte
Dosis suministradas	Número de dosis ya aplicadas al paciente	1 byte

Tabla 8: Codificación de las vacunas

En España hasta los 16 años es obligatoria la vacunación de 13 enfermedades, evidentemente con 13 vacunas no es posible cubrir todas las necesidades teniendo en cuenta que hay gente que se vacuna cada año de enfermedades como la gripe; así se estima un máximo de 150 vacunas por paciente a lo largo de su vida, lo que implica 450 bytes para poder almacenar toda esta información.

3.3.5 Historial médico

La última información que almacenará la tarjeta; aunque no por ello menos importante, es el historial médico del paciente, de manera que pueda ser consultado rápidamente; por ejemplo tras un accidente de tráfico.

La información almacenada, simplemente contendrá la enfermedad pasada y la fecha de diagnóstico. Hay catalogadas unas 9.000 enfermedades así que con dos bytes es suficiente para gestionarlas.

Campo	Descripción	Codificación
Enfermedad	Código de la enfermedad	2 bytes

Fecha	Fecha de diagnóstico de la enfermedad	3 bytes (día/mes/año)

Tabla 9: Codificación del historial médico

Asumiendo que un individuo no sufre más de 100 enfermedades a lo largo de su vida, con 500 bytes quedan cubiertas las necesidades de almacenamiento.

3.3.6 Requerimiento de memoria

La cantidad de memoria que es requerida para almacenar toda la información en la tarjeta está reflejada en la siguiente tabla:

Información	Bytes requeridos
Datos personales	212
Alergias	600
Recetas	540
Vacunas	450
Historial médico	500
Total	**2302**

Tabla 10: Total de memoria

Sin embargo, esta cantidad no representa la realidad ya que esta información es almacenada en objetos JavaCard que requieren de un cierto “overhead” dependiente del sistema operativo. De forma aproximada se puede concluir que serán necesarios unos **4 Kilobytes de memoria no volátil** para guardar toda la información.

4. Herramientas de desarrollo

4.1 NetBeans

4.1.1 Introducción

NetBeans es un entorno de desarrollo libre, gratuito y sin restricciones de uso; hecho principalmente para el lenguaje de programación Java. Existen, además, un número importante de módulos para extenderlo.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun MicroSystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans.

Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

4.1.2 Instalación

Lo primero es descargarse de la siguiente página web https://netbeans.org/ el fichero de instalación, ejecutarlo y seguir las instrucciones del wizard que aparece. La versión más actual de NetBeans es la 7.3.1; pero no es necesario instalarse esta versión, de hecho, este proyecto se realizó con la versión 6.9.

El siguiente paso, una vez instalado el NetBeans, es añadir los plugins necesarios para poder compilar y convertir applets JavaCard. Dentro de la pestaña “Tools”, menú “Plugins” se deberán agregar los que aparecen en la siguiente pantalla:

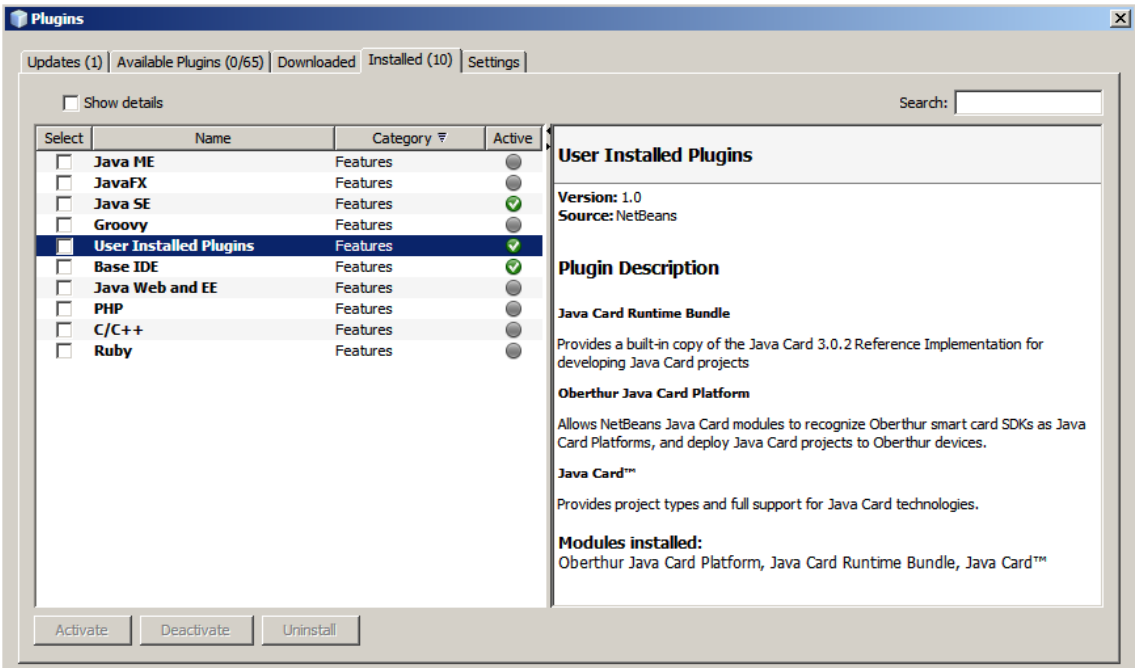

Figura 6: Plugins necesarios

4.1.3 Proyecto de applet javacard

Para crear el proyecto del applet de gestión sanitaria, se debe empezar por desplegar la pestaña “File”, opción “New Project”, categoría “Java Card”, proyecto “Classic Applet Project” (ver Figura 7). Una vez aquí pulsar la opción “Next” y elegir la ruta donde se almacenará el proyecto, el nombre del mismo, así como el AID de paquete y de clase, como se vio en el apartado 2.4.3.2.

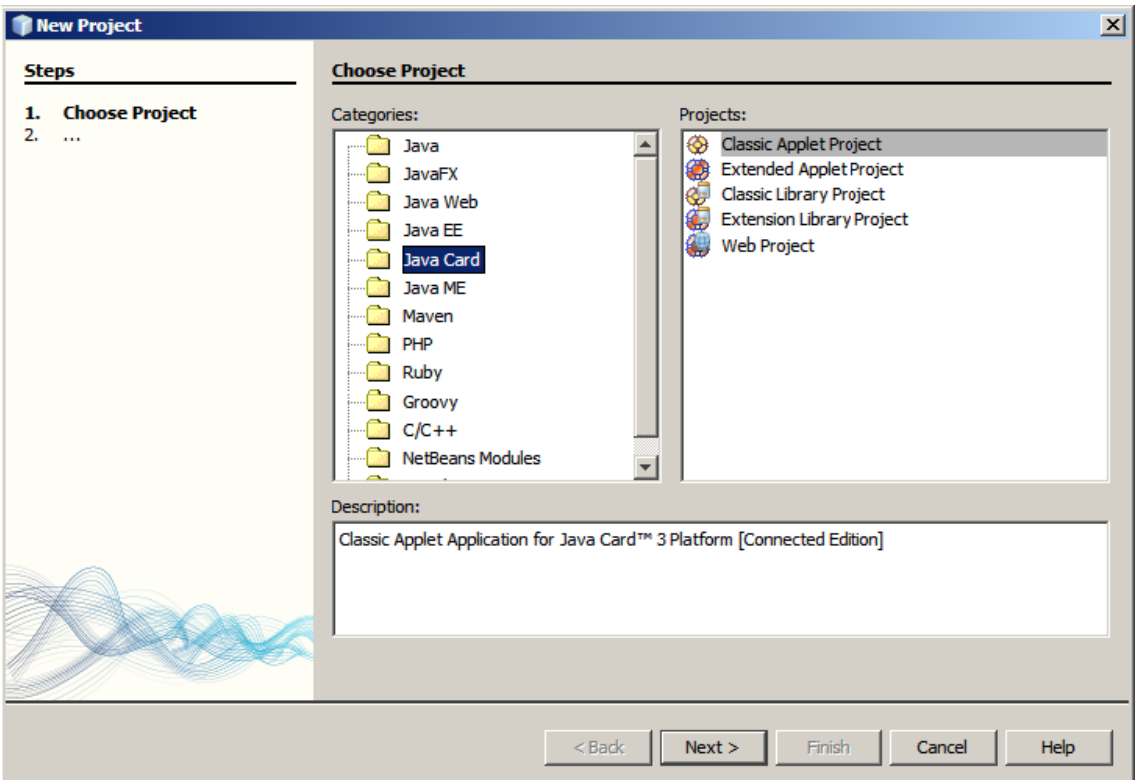

Figura 7: Creación del proyecto para el applet

Una vez creado el proyecto, se pueden empezar a añadir los ficheros .java que formarán el paquete.

4.1.4 Proyecto de herramienta de gestión

La creación del proyecto para la herramienta de gestión es más sencilla que la del applet, ya que no conlleva tener instalado ningún plugin especial, ni tampoco necesita ningún tipo de AID. Simplemente habrá que elegir la categoría “Java” y el proyecto “Java Desktop Application” del menú “New Project”, como muestra la siguiente figura:

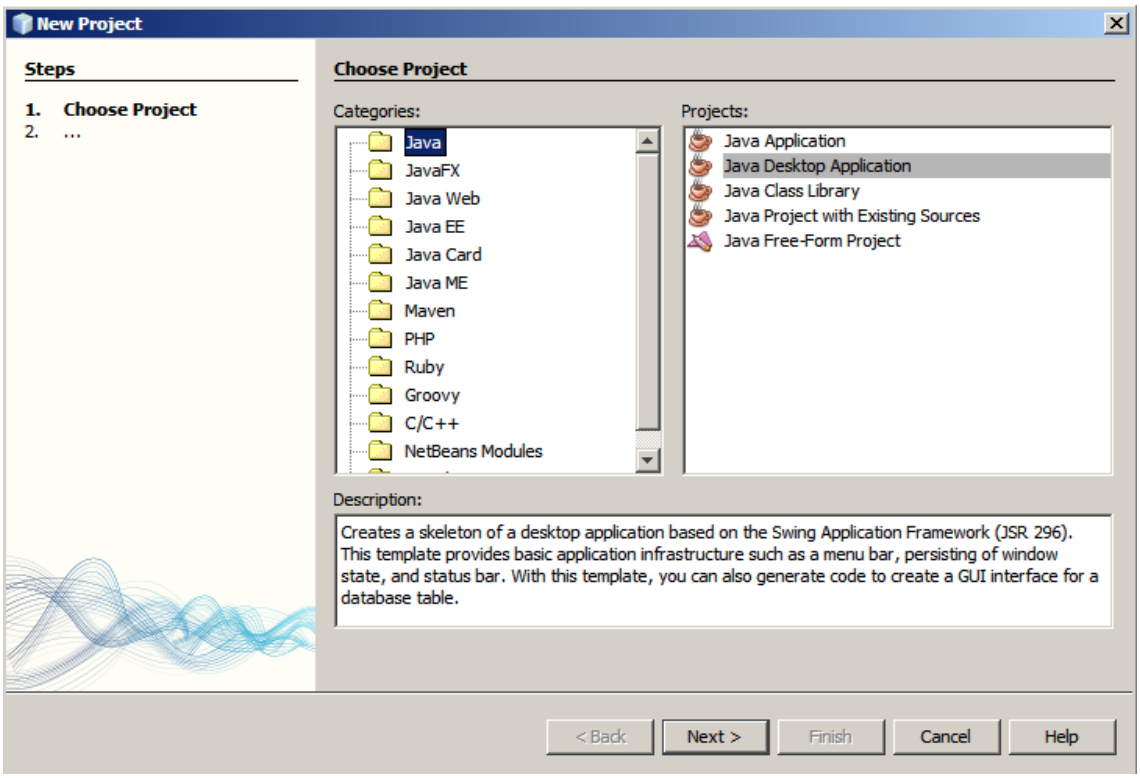

Figura 8: Creación del proyecto para la herramienta

4.2 GP Shell

4.2.1 Introducción

GP Shell surge de la necesidad de cargar, instalar y borrar applets de una tarjeta compatible con GlobalPlatform; así se creó este Shell compuesto por una librería de C y una línea de comandos.

Es posible descargar la última versión de GP Shell de esta dirección http://sourceforge.net/projects/globalplatform/files/ y se puede encontrar toda la información necesaria para conocer el funcionamiento de este Shell en http://sourceforge.net/p/globalplatform/wiki/Home/ .

4.2.2 Ejecución

La carpeta del GP Shell contiene todos los ficheros necesarios para poder operar sobre la tarjeta, incluyendo el fichero de arranque “run.bat” y los distintos scripts con

extensión .txt, así como las dll necesarias para la ejecución, como las que incluyen la implementación de los comandos GlobalPlatform.

Es importante resaltar que si tenemos varios lectores de tarjeta conectados al mismo tiempo, no habrá posibilidad de elegir el destinatario de los comandos, simplemente se usará el primero de la lista de la librería PC/SC de Windows. Así, es recomendable tener solo un lector conectado al equipo.

4.2.3 Menú principal

Tras ejecutar el fichero "`run.bat`" aparece el siguiente menú principal:

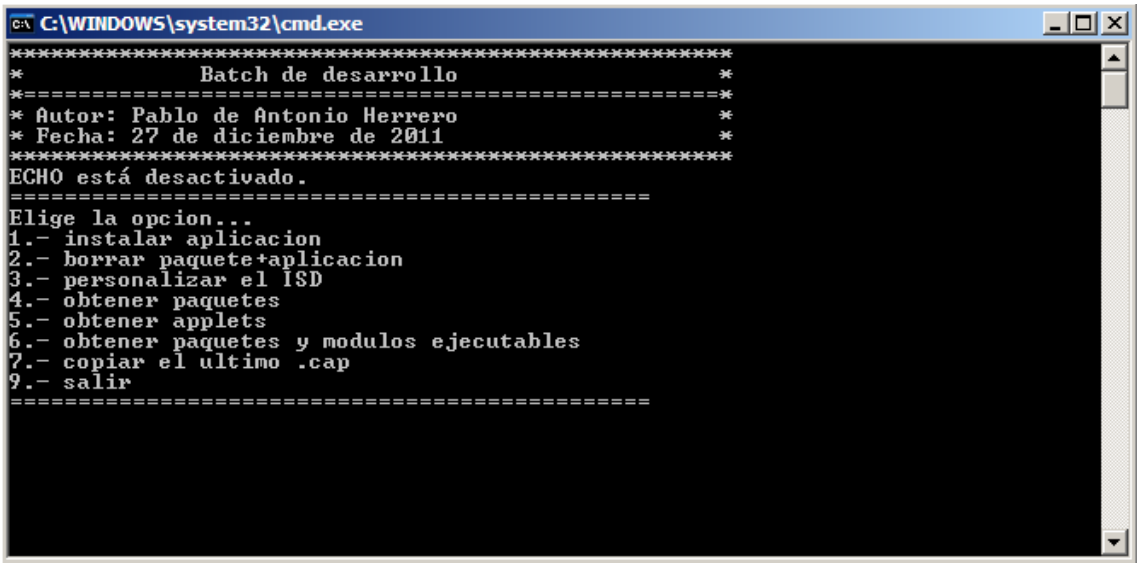

Figura 9: Menú principal del Shell

- Opción 1: ejecuta el script contenido en el fichero "`instalaAppSanitaria.txt`" que abre un canal seguro, carga el paquete e instala el applet de gestión sanitaria con el AID: 'A000000000F0' quedando la tarjeta lista para ser usada.
- Opción 2: ejecuta el script del fichero "`borraAppSanitaria.txt`" que abre un canal seguro y borra el paquete de gestión sanitaria desinstalando el applet también. Esta opción es muy útil durante la etapa de desarrollo, ya que con

frecuencia hay que corregir errores en el applet y es necesario borrar el paquete entero para poder a volver a cargar la versión corregida.

- Opción 3: corre el script definido en el fichero "`putKey.txt`" que tras abrir canal seguro mete tres juegos de claves en la tarjeta, uno para cada usuario. Esta opción es irreversible y solo se hará una vez en la vida de la tarjeta, generalmente se hace en la fase de producción de la misma.
- Opción 4: ejecuta el script del fichero "`getStatusPaq.txt`" que primero establece un canal seguro y después manda comandos get status de GP para obtener los AIDs de los paquetes cargados en la tarjeta.
- Opción 5: corre el script definido en el fichero "`getStatusApp.txt`", igual que la opción anterior; pero en este caso se obtienen los AIDs de los applets instalados en la tarjeta.
- Opción 6: Es una mezcla de las dos anteriores, ya que tras ejecutar los comandos contenidos en el fichero "`getStatusPaqEM.txt`", se obtienen los AIDs de los paquetes y los applets que contiene cada paquete, estén o no instalados.
- Opción 7: Esta opción realmente no ejecuta ningún comando sobre la tarjeta, simplemente es por optimización de la fase de desarrollo, ya que este Shell solo puede cargar ficheros .cap que estén en su misma carpeta. Esta opción mueve el fichero convertido del applet de la ruta de NetBeans y lo pega dentro de la carpeta del Shell para que pueda ser cargado e instalado con la opción 1.
- Opción 9: Cierra la ventana de comandos y termina la ejecución.

5. Implementación y comandos

5.1 Comandos de seguridad

Como se describió en el apartado 2.4.4.3, para la apertura de un canal seguro son necesarios un par de comandos, de modo que tanto el host como la tarjeta puedan autenticarse mutuamente.

5.1.1 Initialize update

5.1.1.1 Estructura del comando

Código	Valor	Significado
CLA	“80”	Ver 2.4.2.1
INS	“50”	Initialize update
P1	“xx”	Número de versión de las claves
P2	“00”	No usado
LC	“08”	Longitud del Host Challenger
Datos	“xx…xx”	Host Challenge

Tabla 11: Cabecera Initialize Update

El host Challenge es un número aleatorio generado por la entidad externa que quiere autenticarse con la tarjeta.

5.1.1.2 Respuesta del comando

La correcta ejecución de este comando se traduce en una serie de datos, como se describe en la tabla inferior, y un SW ‘9000’.

Nombre	Longitud
Datos de diversificación	10 bytes
Información de clave	2 bytes
Contador	2 bytes
Card Challenge	6 bytes
Criptograma de la tarjeta	8 bytes

Tabla 12: Respuesta Initialize Update

Los datos de diversificación son los usados típicamente para obtener las claves estáticas de la tarjeta.

La información de clave está compuesta por el número de versión del juego de claves con el que se abre sesión más el protocolo usado (en este caso 02).

El contador enumera el número de veces que se abrió un canal seguro con esta versión de clave.

El Card Challenge es un número aleatorio generado internamente por la tarjeta.

El criptograma es un criptograma de autenticación calculado por la tarjeta en base al host Challenger.

5.1.1.3 Status Word

Los SW que puede devolver este comando están representados en la siguiente tabla:

SW1	SW2	Significado
“6A”	“88”	Datos referenciados no encontrados
“90”	“00”	Comando procesado sin errores

Tabla 13: Status Word Initialize Update

5.1.2 External authenticate

5.1.2.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“82”	External authenticate
P1	“xx”	Nivel de seguridad
P2	“00”	No usado
LC	“10”	Longitud del criptograma del host y el MAC
Datos	“xx…xx”	Criptograma del host y MAC del comando

Tabla 14: Cabecera External Authenticate

El nivel de seguridad puede tomar los siguientes valores:

- “00” : Comandos sin seguridad
- “01” : Comandos con MAC
- “03” : Comandos con MAC y encriptados

5.1.2.2 Respuesta del comando

Este comando no presenta datos de respuesta.

5.1.2.3 Status Word

Los SW que puede devolver este comando están representados en la siguiente tabla:

SW1	SW2	Significado
“63”	“00”	Incorrecto criptograma del host
“90”	“00”	Comando procesado sin errores

Tabla 15: Status Word External Authenticate

5.2 Comandos de obtención de información

Este primer set de comandos tiene como funcionalidad poder obtener toda la información del paciente almacenada por el applet de gestión.

5.2.1 Obtener alergias

5.2.1.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“4E”	Obtener alergias
P1	“xx”	Parámetro de control P1
P2	“00”	No usado
LC	“08”	Longitud del MAC
Datos	“xx…xx”	MAC del comando

Tabla 16: Cabecera "Obtener alergias"

El parámetro de control P1, puede tomar los siguientes valores:

- “00”: Primer comando. Inicio de secuencia.
- “01”: Siguiente comando. Continúa secuencia.

5.2.1.2 Respuesta del comando

Si el comando es ejecutado correctamente, el applet responderá la información solicitada de acuerdo a la siguiente estructura:

Nombre	Valor	Longitud
Alergia 1	Grupo	1 byte
	Código de alergia	2 bytes

Alergia 2	Grupo	1 byte
	Código de alergia	2 bytes
Alergia N	Grupo	1 byte
	Código de alergia	2 bytes

Tabla 17: Respuesta "Obtener alergias"

5.2.1.3 Status Word

Los SW que puede devolver este comando están representados en la siguiente tabla:

SW1	SW2	Significado
“63”	“10”	Más datos disponibles
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver.
“6A”	“86”	P1 incorrecto
“69”	“85”	Secuencia incorrecta

Tabla 18: Status Words "Obtener alergias"

En caso de que los datos de respuesta no puedan ser devueltos en un simple APDU, se devolverá el SW ‘6310’ junto a la mayor información posible, posibilitando así que se envíe de nuevo el mismo comando con P1=’01’ para obtener los siguientes datos. Se deberá encadenar este mecanismo hasta la obtención de un SW ‘9000’ que indicará el final de la secuencia.

5.2.2 Obtener vacunas

5.2.2.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“52”	Obtener vacunas
P1	“xx”	Parámetro de control P1
P2	“00”	No usado
LC	“08”	Longitud del MAC
Datos	“xx…xx”	MAC del comando

Tabla 19: Cabecera "Obtener vacunas"

El parámetro de control P1, puede tomar los siguientes valores

- “00”: Primer comando. Inicio de secuencia.
- “01”: Siguiente comando. Continúa secuencia.

5.2.2.2 Respuesta del comando

Si el comando es ejecutado correctamente, el applet responderá la información solicitada de acuerdo a la siguiente estructura:

Nombre	Valor	Longitud
Vacuna 1	Identificador de vacuna	1 byte
	Dosis totales	1 byte
	Dosis suministradas	1 byte
Vacuna N	Identificador de vacuna	1 byte
	Dosis totales	1 byte
	Dosis suministradas	1 byte

Tabla 20: Respuesta "Obtener vacunas"

5.2.2.3 Status Word

En caso de que los datos de respuesta no puedan ser devueltos en un simple APDU, se devolverá el SW '6310' junto a la mayor información posible, posibilitando así que se envíe de nuevo el mismo comando con P1='01' para obtener los siguientes datos. Se deberá encadenar este mecanismo hasta la obtención de un SW '9000' que indicará el final de la secuencia.

SW1	SW2	Significado
"63"	"10"	Más datos disponibles
"90"	"00"	Comando procesado sin errores. No hay más datos para devolver.

Tabla 21: Status Words "Obtener vacunas"

5.2.3 Obtener historial médico

5.2.3.1 Estructura del comando

Código	Valor	Significado
CLA	"84"	Ver 2.4.2.1
INS	"54"	Obtener historial médico
P1	"xx"	Parámetro de control P1
P2	"00"	No usado
LC	"08"	Longitud del MAC
Datos	"xx…xx"	MAC del comando

Tabla 22: Cabecera "Obtener historial médico"

El parámetro de control P1, puede tomar los siguientes valores:

- "00": Primer comando. Inicio de secuencia.
- "01": Siguiente comando. Continúa secuencia.

5.2.3.2 Respuesta del comando

Si el comando es ejecutado correctamente, el applet responderá la información solicitada de acuerdo a la siguiente estructura:

Nombre	Valor	Longitud
Enfermedad 1	Código de enfermedad	2 bytes
	Fecha de diagnostico	3 bytes
Enfermedad 2	Código de enfermedad	2 bytes
	Fecha de diagnostico	3 bytes
Enfermedad N	Código de enfermedad	2 bytes
	Fecha de diagnostico	3 bytes

Tabla 23: Respuesta "Obtener historial médico"

5.2.3.3 Status Word

En caso de que los datos de respuesta no puedan ser devueltos en un simple APDU, se devolverá el SW ‘6310’ junto a la mayor información posible, posibilitando así que se envíe de nuevo el mismo comando con P1=’01’ para obtener los siguientes datos. Se deberá encadenar este mecanismo hasta la obtención de un SW ‘9000’ que indicará el final de la secuencia.

SW1	SW2	Significado
“63”	“10”	Más datos disponibles
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver.

Tabla 24: Status Words "Obtener historial médico"

5.2.4 Obtener datos personales

5.2.4.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“55”	Obtener datos personales
P1	“xx”	Parámetro de control P1
P2	“00”	No usado
LC	“08”	Longitud del MAC
Datos	“xx…xx”	MAC del comando

Tabla 25: Cabecera "Obtener datos personales"

El parámetro de control P1 toma el valor del tag cuya información se desea obtener, como indica la siguiente tabla:

Tag	Significado
“10”	Nombre
“20”	Apellidos
“30”	Provincia
“40”	Localidad
“50”	Calle/Avenida/Plaza
“60”	Número
“70”	Piso/Escalera/Puerta
“80”	Teléfono
“90”	Teléfono de contacto en caso de emergencia
“A0”	Número de la seguridad social

Tabla 26: Valores posibles de P1

5.2.4.2 Respuesta del comando

El comando responderá en formato TLV (Tag+Longitud+Valor), dependiendo de la información solicitada (P1).

Ejemplo:

- *COMANDO: 80 93 10 00 00*
- *RESPUESTA: 10 05 50 41 42 4C 4F (Tag=10, Long.=05, Valor=PABLO(ASCII)*

5.2.4.3 Status Word

Los distintos SW que devolverá este comando vienen especificados en la siguiente tabla:

SW1	SW2	Significado
“6A”	“86”	P1 incorrecto. Tag desconocido.
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver.
“6A”	“88”	Tarjeta sin información. No personalizada

Tabla 27: Status Words "Obtener datos personales"

5.2.5 Obtener recetas

5.2.5.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“56”	Obtener recetas
P1	“xx”	Parámetro de control P1
P2	“00”	No usado
LC	“08”	Longitud del MAC
Datos	“xx…xx”	MAC del comando

Tabla 28: Cabecera "Obtener recetas"

El parámetro de control P1, puede tomar los siguientes valores:

- “00”: Primer comando. Inicio de secuencia.
- “01”: Siguiente comando. Continúa secuencia.

5.2.5.2 Respuesta del comando

La correcta ejecución del comando implica la devolución de todas las recetas almacenadas por el applet de acuerdo a la siguiente estructura:

Nombre	Valor	Longitud
Receta 1	Todos los datos de la receta.	18 bytes
Receta 2	Todos los datos de la receta.	18 bytes
Receta N	Todos los datos de la receta.	18 bytes

Tabla 29: Respuesta "Obtener recetas"

5.2.5.3 Status Word

En caso de que los datos de respuesta no puedan ser devueltos en un simple APDU, se devolverá el SW ‘6310’ junto a la mayor información posible, posibilitando así que se envíe de nuevo el mismo comando con P1=’01’ para obtener los siguientes datos. Se deberá encadenar este mecanismo hasta la obtención de un SW ‘9000’ que indicará el final de la secuencia.

SW1	SW2	Significado
“63”	“10”	Más datos disponibles
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver.

Tabla 30: Status Words "Obtener recetas"

5.3 Comandos de actualización de información

Los comandos descritos en esta sección son usados para añadir o modificar la información almacenada por el applet.

El uso de estos comandos está asociado al usuario que los envía, de tal forma que su correcta ejecución dependerá de quien envía el comando. Por ejemplo, un usuario "farmacia" no podrá modificar los datos personales de la tarjeta. Así, la siguiente tabla presenta la relación entre comandos y usuarios:

Usuario/Comando	Hospital	Farmacia	Ciudadano
Borrar alergia	X	---	---
Añadir alergia	X	---	---
Añadir vacuna	X	---	---
Incrementar dosis vacuna	X	---	---
Borrar enfermedad	X	---	---
Añadir enfermedad	X	---	---
Borrar datos personales	X	---	---
Añadir datos personales	X	---	X
Borrar receta	X	---	---
Añadir receta	X	---	---
Marcar receta despachada	X	X	---

Tabla 31: Comandos soportados por cada usuario

Dónde:

- X, comando soportado
- ---, comando **no** soportado

5.3.1 Borrar alergia

Este comando permite el borrado de una alergia de forma individual o el borrado de todo un grupo de alergias.

5.3.1.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“72”	Borrar alergia
P1	“xx”	Parámetro de control P1
P2	“00”	No usado
LC	“xx”	Longitud de los datos
Datos	“xx…xx”	Datos

Tabla 32: Cabecera "Borrar alergia"

El parámetro de control P1, puede tomar los siguientes valores:

- “00”: Borrar alergia.
- “01”: Borrar grupo.

5.3.1.2 Campo de datos

- En el caso del borrado de una alergia de forma individual, el campo de datos estará compuesto por el byte de grupo más los dos bytes del identificador de alergia.
- En el caso de borrado de un grupo simplemente se enviará el byte identificador de grupo.

5.3.1.3 Respuesta del comando

Este comando no presenta datos de respuesta

5.3.1.4 Status Word

SW1	SW2	Significado
“6A”	“88”	Alergia no encontrada
“6A”	“86”	Grupo incorrecto
“69”	“85”	Usuario sin capacidad de usar esta función
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver.

Tabla 33: Status Words "Borrar alergia"

5.3.2 Añadir alergia

Este comando permite añadir alergias de forma individual.

5.3.2.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“73”	Añadir alergia
P1	“00”	No usado
P2	“00”	No usado
LC	“xx”	Longitud de los datos
Datos	“xx…xx”	Datos

Tabla 34: Cabecera "Añadir alergia"

Datos:

El campo de datos estará compuesto por el byte de grupo más los dos bytes del identificador de alergia.

5.3.2.2 Respuesta del comando

Este comando no presenta datos de respuesta.

5.3.2.3 Status Word

SW1	SW2	Significado
“6A”	“89”	Alergia ya presente
“6A”	“80”	Grupo incorrecto
“69”	“85”	Usuario sin capacidad de usar esta función
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver.

Tabla 35: Status Words "Añadir alergia"

5.3.3 Añadir vacuna

Este comando permite añadir una vacuna de forma individual, indicando además el número de dosis de la misma.

5.3.3.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“74”	Añadir vacuna
P1	“xx”	Identificador de vacuna
P2	“xx”	Número de dosis totales
LC	“08”	Longitud del MAC
Datos	“xx…xx”	MAC del comando

Tabla 36: Cabecera "Añadir vacuna"

5.3.3.2 Respuesta del comando

Este comando no presenta datos de respuesta.

5.3.3.3 Status Word

SW1	SW2	Significado
“6A”	“89”	Vacuna ya presente
“69”	“85”	Usuario sin capacidad de usar esta función
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver.

Tabla 37: Status Words "Añadir vacuna"

5.3.4 Incrementar dosis de vacuna

Este comando permite incrementar el contador de dosis suministradas de una vacuna ya almacenada en la tarjeta.

5.3.4.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“76”	Incrementar dosis vacuna
P1	“xx”	Identificador de vacuna
P2	“00”	No usado
LC	“08”	Longitud del MAC
Datos	“xx…xx”	MAC del comando

Tabla 38: Cabecera "Incrementar dosis vacuna"

5.3.4.2 Respuesta del comando

Este comando no presenta datos de respuesta.

5.3.4.3 Status Word

SW1	SW2	Significado
“6A”	“80”	Incremento por encima del máximo
“69”	“85”	Usuario sin capacidad de usar esta función
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver.
“6A”	“88”	Vacuna no presente

Tabla 39: Status Words "Incrementar dosis vacuna"

5.3.5 Borrar vacuna

Comando para el borrado de vacunas de forma individual.

5.3.5.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“78”	Borrar vacuna
P1	“xx”	Identificador de vacuna
P2	“00”	No usado
LC	“08”	Longitud del MAC
Datos	“xx…xx”	MAC del comando

Tabla 40: Cabecera "Borrar vacuna"

5.3.5.2 Respuesta del comando

Este comando no presenta datos de respuesta.

5.3.5.3 Status Word

SW1	SW2	Significado
“6A”	“88”	Vacuna no existente
“69”	“85”	Usuario sin capacidad de usar esta función
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver.

Tabla 41: Status Words "Borrar vacuna"

5.3.6 Añadir enfermedad

Este APDU permite añadir enfermedades al historial médico del paciente.

5.3.6.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“7A”	Añadir enfermedad
P1	“00”	No usado
P2	“00”	No usado
LC	“xx”	Longitud de los datos y el MAC
Datos	“xx…xx”	Datos más MAC del comando

Tabla 42: Cabecera "Añadir enfermedad"

Datos:

- El primer byte indicará el tipo de enfermedad.

- Los siguientes 2 bytes son el identificador de la enfermedad
- Los siguientes 3 bytes son la fecha de diagnóstico

5.3.6.2 Respuesta del comando

Este comando no presenta datos de respuesta.

5.3.6.3 Status Word

SW1	SW2	Significado
“6A”	“80”	Fecha de diagnóstico no presente en el campo de datos
“69”	“85”	Usuario sin capacidad de usar esta función
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver
“67”	“00”	Longitud incorrecta

Tabla 43: Status Words "Añadir enfermedad"

5.3.7 Borrar enfermedad

Este comando posibilita el borrado de alguna de las enfermedades almacenadas en el historial médico del paciente.

5.3.7.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“7C”	Borrar enfermedad
P1	“E1”	Primer byte del identificador de

		enfermedad
P2	“E2”	Segundo byte del identificador de enfermedad
LC	“xx”	Longitud del MAC
Datos	“xx…xx”	MAC del comando

Tabla 44: Cabecera "Borrar enfermedad"

5.3.7.2 Respuesta del comando

Este comando no presenta datos de respuesta.

5.3.7.3 Status Word

SW1	SW2	Significado
“6A”	“86”	Enfermedad no encontrada
“69”	“85”	Usuario sin capacidad de usar esta función
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver

Tabla 45: Status Words "Borrar enfermedad"

5.3.8 Borrar datos personales

Este comando faculta borrar cualquier información guardada como dato personal del usuario.

5.3.8.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“8A”	Borrar datos personales

P1	“xx”	Parámetro de control P1
P2	“00”	No usado
LC	“08”	Longitud del MAC
Datos	“xx…xx”	MAC del comando

Tabla 46: Cabecera "Borrar datos personales"

El parámetro de control P1 toma el valor del tag cuya información se desea borrar de acuerdo a la siguiente tabla:

Tag	Significado
“10”	Nombre
“20”	Apellidos
“30”	Provincia
“40”	Localidad
“50”	Calle/Avenida/Plaza
“60”	Número
“70”	Piso/Escalera/Puerta
“80”	Teléfono
“90”	Teléfono de contacto en caso de emergencia
“A0”	Número de la seguridad social

Tabla 47: Posibles valores de P1

5.3.8.2 Status Word

SW1	SW2	Significado
“6A”	“86”	Tag incorrecto.
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver.

Tabla 48: Status Words "Borrar datos personales"

5.3.9 Añadir datos personales

Este comando posibilita añadir cualquiera de los campos que componen los datos personales del paciente.

5.3.9.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“8C”	Añadir datos personales
P1	“xx”	Parámetro de control P1
P2	“00”	No usado
LC	“xx”	Longitud de los datos más el MAC
Datos	“xx…xx”	Datos + MAC del comando

Tabla 49: Cabecera "Añadir datos personales"

El parámetro de control P1 toma el valor del tag cuya información se desea almacenar:

Tag	Significado
“10”	Nombre
“20”	Apellidos
“30”	Provincia
“40”	Localidad
“50”	Calle/Avenida/Plaza
“60”	Número
“70”	Piso/Escalera/Puerta

“80”	Teléfono
“90”	Teléfono de contacto en caso de emergencia
“A0”	Número de la seguridad social

Tabla 50: Posibles valores de P1

Datos:

El campo de datos contendrá la información a almacenar en la tarjeta.

5.3.9.2 Status Word

SW1	SW2	Significado
“6A”	“86”	Tag incorrecto.
“6A”	“89”	El tag ya contiene información.
“6A”	“8A”	Formato incorrecto
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver

Tabla 51: Status Words "Añadir datos personales"

5.3.10 Borrar receta

5.3.10.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“8D”	Borrar receta
P1	“00”	Parámetro de control P1
P2	“00”	No usado
LC	“0B”	Longitud del código de receta más el MAC
Datos	“xx…xx”	Código de receta y MAC

		del comando

Tabla 52: Cabecera "Borrar receta"

Datos:

El campo de datos contendrá únicamente los 3 bytes del identificador de medicamento de la receta a borrar.

5.3.10.2 Status Word

SW1	SW2	Significado
“6A”	“88”	Identificador no encontrado
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver

Tabla 53: Status Words "Borrar receta"

5.3.11 Añadir receta

Este comando permite añadir una receta de forma individual.

5.3.11.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“8E”	Añadir receta
P1	“00”	No usado
P2	“00”	No usado
LC	“xx”	Longitud de los datos más el MAC
Datos	“xx…xx”	Datos + MAC del comando

Datos:

Campo	Longitud
Código de medicamento	3 bytes
Cantidad	4 bytes
Dosis de cada elemento	1 byte
Número de dosis por toma	1 byte
Número de tomas	2 bytes
Duración del tratamiento	2 bytes
Receta crónica	1 byte
Fecha siguiente receta	3 bytes

Tabla 54: Campo de datos

5.3.11.2 Status Word

SW1	SW2	Significado
“6A”	“89”	Medicamento ya presente.
“6A”	“80”	Campo de datos incorrecto. Falta alguno de los tags.
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver

Tabla 55: Status Words "Añadir receta"

5.3.12 Marcar receta despachada

Este APDU posibilita a un usuario “farmacia” marcar una receta como despachada.

5.3.12.1 Estructura del comando

Código	Valor	Significado
CLA	“84”	Ver 2.4.2.1
INS	“8F”	Despachar receta
P1	“00”	Parámetro de control P1
P2	“00”	No usado
LC	“0B”	Longitud del código de receta más el MAC
Datos	“xx…xx”	Código de receta y MAC del comando

Tabla 56: Cabecera "Marcar receta despachada"

Datos:

El campo de datos contendrá únicamente los 3 bytes del identificador de medicamento de la receta que será marcada como despachada.

5.3.12.2 Status Word

SW1	SW2	Significado
“6A”	“88”	Identificador no encontrado
“69”	“85”	Usuario sin capacidad de usar esta función
“90”	“00”	Comando procesado sin errores. No hay más datos para devolver

Tabla 57: Status Words "Marcar receta despachada"

6. Applet

6.1 Introducción

Toda la información descrita en el capítulo 3 se debe almacenar de forma estructurada en la tarjeta y además debe ser accesible desde el exterior. La entidad encargada de toda esta gestión es el applet de gestión sanitaria.

Este applet fue desarrollado íntegramente en NetBeans de acuerdo al proyecto creado en el apartado 4.1.3.

Tras la codificación del mismo, lo siguiente es compilar y convertir el applet haciendo clic con el botón derecho sobre el proyecto del applet y seleccionando la opción de compilar y convertir.

El resultado es un fichero .cap, que como se explicó en el apartado 4.1.5 se cargará e instalará en la tarjeta usando el GP Shell.

6.2 Applet de gestión sanitaria

Como todo paquete JavaCard está compuesto por varias clases, en función de su utilidad dentro del applet. Empezando por la clase principal que contiene la llamada al método “install” de JavaCard y el “process” de la clase applet.

6.2.1 Clases

6.2.1.1 Clase AppSanidad

Esta es la clase principal del applet y contiene los dos métodos básicos que debe implementar un applet JavaCard:

- Método “install”: Este método es llamado durante la instalación del applet, y en él se crean todos los objetos que usará el applet durante su ciclo de vida. Se trata de arrays de referencias estáticos que contienen los objetos que modelan cada estructura. Estos objetos están detallados a lo largo de este capítulo.

- Método “process”: Este método es el punto de entrada al applet de gestión sanitaria. El SO de la tarjeta llama a este método pasándole como parámetro un objeto APDU que contendrá el comando a procesar. Tras recibir el comando, el applet debe invocar los métodos de su dominio de seguridad para eliminar la capa del protocolo de seguridad (canal seguro, apartado 2.4.4.3) invocando el método “processSecurity”. Esta capa de seguridad es transparente para el applet, ya que el encargado de descifrar y chequear la integridad de los comandos recibidos es el dominio de seguridad al que está asociado el applet.

  Una vez, el applet recibe el comando en claro, comienza el análisis de la cabecera. En concreto los bytes de CLA e INS, que determinará que comando está entrando. Si estos chequeos son correctos, se invoca alguno de los métodos de la siguiente clase.

6.2.1.2 Clase GestorComandos

Todos los comandos definidos en el capítulo 5, encuentran su correspondiente método en esta clase, que representa un “administrador de comandos”.

Todos estos métodos siguen una estructura similar, ya que comienzan analizando los bytes de parámetros P1 y P2 comprobando que tienen los valores válidos que define el API de comandos.

Tras pasar este primer filtro, estos métodos analizan el campo de datos (en caso de que exista) y ejecutan la acción.

6.2.1.3 Clase Constantes

Esta clase contiene todas las constantes usadas por las distintas clases que componen el paquete de gestión sanitaria.

6.2.1.4 Clase Alergia

Esta clase define el objeto “alergia” compuesto por dos campos virtuales:

- Clase: Un byte que identifica el grupo al que pertenece la alergia.
- Identificador: 2 bytes que señalan de forma inequívoca la alergia dentro del grupo.

6.2.1.5 Clase DatosPersonales

Esta clase define dos objetos, uno es el principal que contiene toda la información de los datos personales del usuario:

```
package appSanidad;

/**
 *
 * @author pablo.deantonio
 */
public class DatosPersonales {



    byte[] nombre=new byte[30];
    byte[] apellidos=new byte[60];
    Direccion direccion=new Direccion();
    byte[] telefono=new byte[9];
    byte[] telefonoEmergencia=new byte[9];
    byte[] seguridadSocial=new byte[12];
}

class Direccion{
    byte[] provincia=new byte[20];
    byte[] localidad=new byte[30];
    byte[] calle=new byte[40];
    byte[] numero=new byte[4];
    short piso;

}
```

Figura 10: Campos de la clase DatosPersonales

6.2.1.6 Clase Enfermedad

Esta clase define los objetos de tipo enfermedad y está compuesto por los siguientes campos virtuales:

- Tipo: Byte para identificar el grupo al que pertenece la enfermedad almacenada.
- Identificador: De la enfermedad dentro del grupo, compuesto por dos bytes.
- Día: De la fecha de diagnóstico.
- Mes: De la fecha de diagnóstico.
- Año De la fecha de diagnóstico.

6.2.1.7 Clase Receta

Esta clase se compone del objeto público Receta y dos objetos privados, Cantidad y Fecha, que forman parte de la estructura del objeto principal:

```
/**
 *
 * @author pablo.deantonio
 */
public class Receta {
    byte[] medicamento=new byte[3];
    Cantidad cantidad=new Cantidad();
    byte dosisElemento;
    byte elementosToma;
    byte[] numeroTomas=new byte[2];
    byte[] duracion=new byte[2];
    byte flag_cronico;
    Fecha fecha=new Fecha();
    byte despachada;
}

class Cantidad{
    short formato;
    short cantidad;
}

class Fecha{
    byte dia;
    byte mes;
    byte ano;
}
```

Figura 11: Clase "Receta"

6.2.1.8 Clase Vacuna

Por último, esta clase define el objeto usado para almacenar la información relativa a cada vacuna, estando compuesto por los siguientes campos:

- Identificador: de la vacuna almacenada.
- Dosis suministradas: Número de dosis de la vacuna que ya se le han aplicado al paciente.
- Dosis totales: Número de dosis totales que requiere la vacuna.

7. Herramienta de gestión

7.1 Instalación

Esta herramienta está desarrollada puramente en java, por tanto se deberá tener instalado el entorno de desarrollo adecuado proporcionado por Oracle y que puede ser descargado desde su página web: http://java.com/es/download/index.jsp . La versión del JRE (Java Runtime Enviroment) debe ser la 1.6 o superior.

Una vez instalado el JRE simplemente hay que copiar el fichero Gestión Sanitaria.jar junto a la carpeta `lib` en el directorio de destino, no es necesaria ninguna instalación especial.

Estos ficheros han sido obtenidos tras compilar el proyecto completo y extraerlo en la carpeta `dist` del mismo.

Para ejecutar la herramienta se debe abrir una línea de comandos y teclear:

`java -jar "Gestion_Sanitaria.jar"` y la herramienta arrancará sin problemas.

7.2 Estructura del proyecto

El proyecto está compuesto por varios paquetes de acuerdo a su funcionalidad, como se puede apreciar en la siguiente figura:

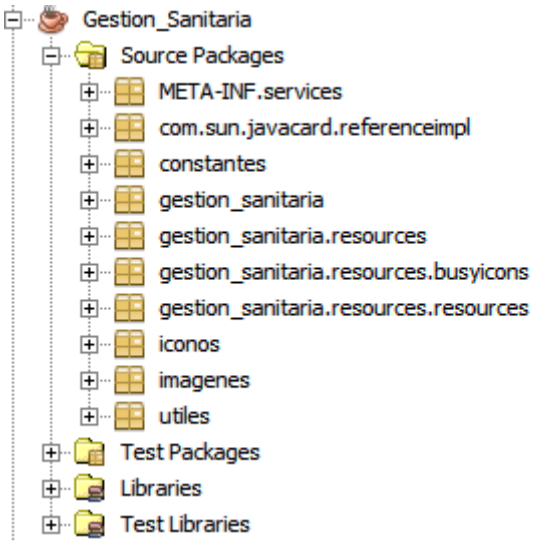

Figura 12: Estructura del proyecto

7.2.1 Paquete implementación de referencia

El paquete `com.sun.javacard.referenceimpl` contiene la librería de Windows que permite la comunicación con lectores PC/SC (`WinsCardParaJava.dll`), así como las clases que implementan los métodos nativos que invocan las funciones de la librería.

Toda esta capa es el interfaz entre la herramienta de gestión sanitaria y el lector de tarjetas, implementando toda la capa de comunicación, tanto en el sentido desde la herramienta al lector como desde el lector a la herramienta.

Lo primero que hace esta capa es la conexión con el lector, para ello escanea todos los lectores conectados al equipo, siendo decisión del usuario elegir uno u otro. Una vez elegido lo enciende para chequear si hay alguna tarjeta insertada en el lector.

Esta capa también se encarga de añadir la capa de seguridad a los comandos salientes que lo requieran, es decir, todos aquellos enviados en dirección al lector y cuya byte de CLA indique que el comando debe llevar MAC, se les calculará dicha firma y se les añadirá al final del comando, modificando también la longitud del comando al tener que añadirle estos 8 bytes de MAC.

7.3 Pantalla principal

En la vida real tendríamos tres herramientas distintas, una para cada usuario, médico – paciente – proveedor de servicios, cada una con las claves adecuadas para comunicarse con la tarjeta.

Para el proyecto se han integrado las tres herramientas en una sola, pudiendo elegir el usuario en la pantalla principal presentada al arrancar la aplicación como se aprecia en la siguiente figura:

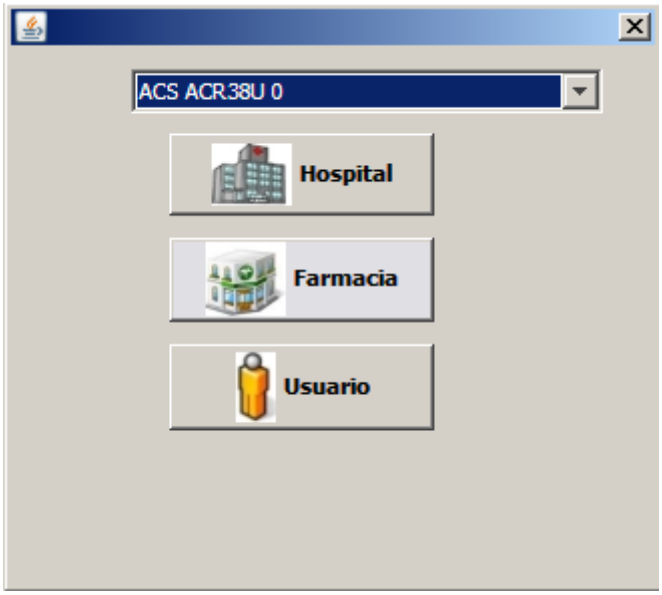

Figura 13: Pantalla de elección de usuario

Una vez elegido lector y el usuario, la herramienta establece un canal seguro de comunicación con la tarjeta usando las claves específicas de cada usuario, lo que garantiza que el usuario en cuestión solo tiene acceso a modificar cierta información.

La siguiente pantalla que se despliega, tras esta primera selección, es la pantalla de elección de datos, como se ve en la figura de abajo:

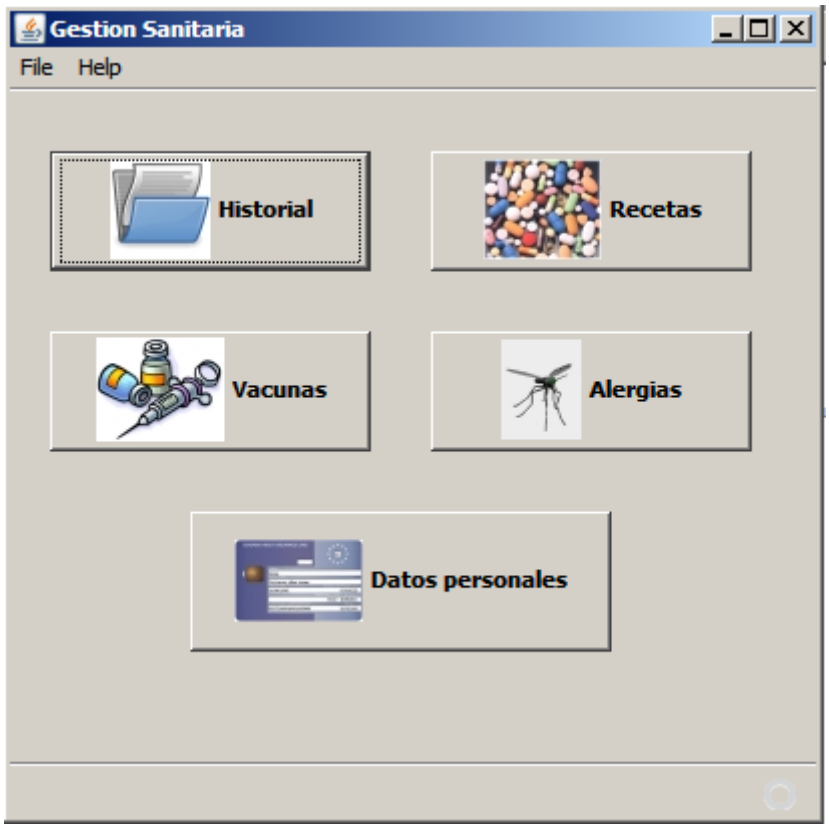

Figura 14: Pantalla de selección de información

En los siguientes apartados se desglosan todas las pantallas, describiendo la funcionalidad de todos los elementos que las integran.

7.4 Pantalla datos personales

Esta pantalla permite la consulta y modificación de los datos personales del paciente, aunque esta última acción solo puede ser llevada a cabo por el centro médico. Salvo el teléfono del usuario y el de la persona a la que llamar en caso de emergencia que pueden ser modificadas también por el propio paciente.

En la siguiente figura se puede observar, como tras iniciar la herramienta en modo usuario, solo los botones de modificación del campo del teléfono de emergencia y del teléfono del paciente son accesibles:

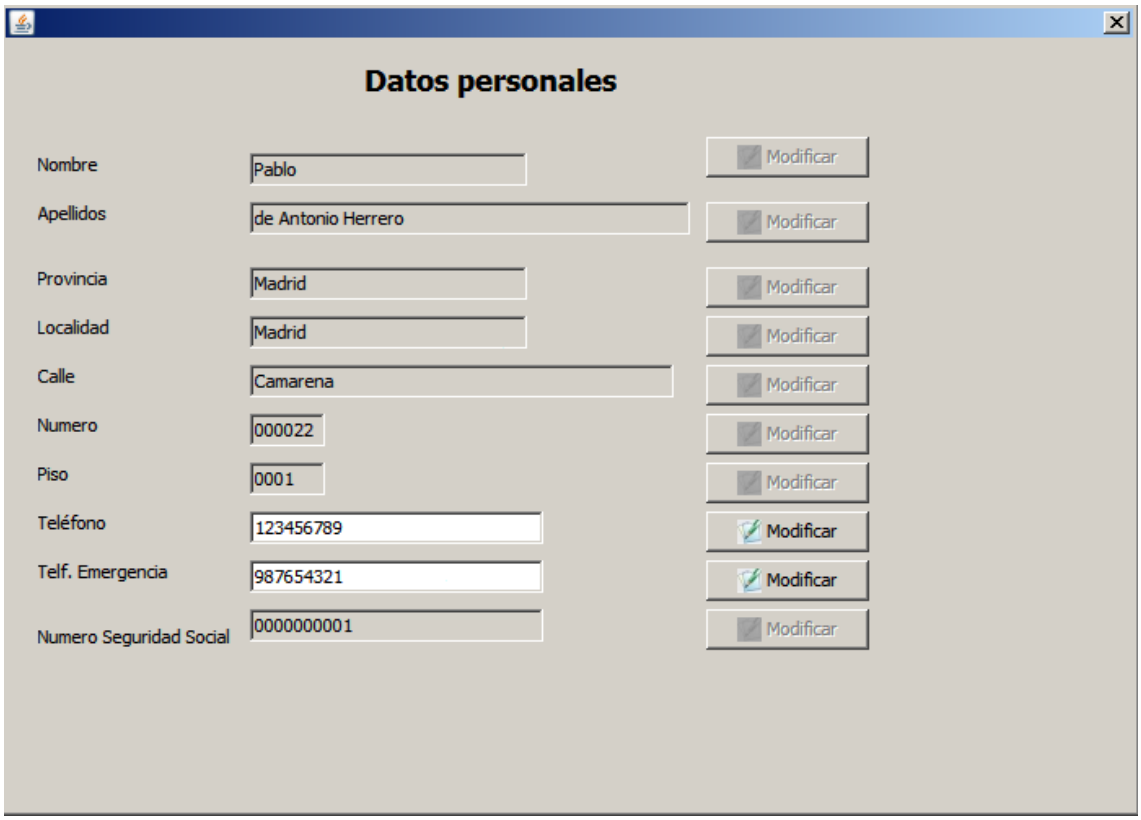

Figura 15: Pantalla de datos personales, vista de paciente

7.5 Pantalla historial médico

Esta pantalla permite al médico actualizar el historial médico del paciente y al paciente consultarlo.

El médico dispone de varios menús desplegables:

- Tipo: Permite elegir el grupo al que pertenece la enfermedad diagnosticada, de acuerdo al código internacional de enfermedades (CIE).
- Enfermedad: Desplegable para seleccionar la enfermedad.
- Fecha: Permite seleccionar el día, mes y año en que fue diagnosticada la enfermedad.

Una vez establecidos estos campos ya se puede añadir la enfermedad al historial pulsando el botón correspondiente.

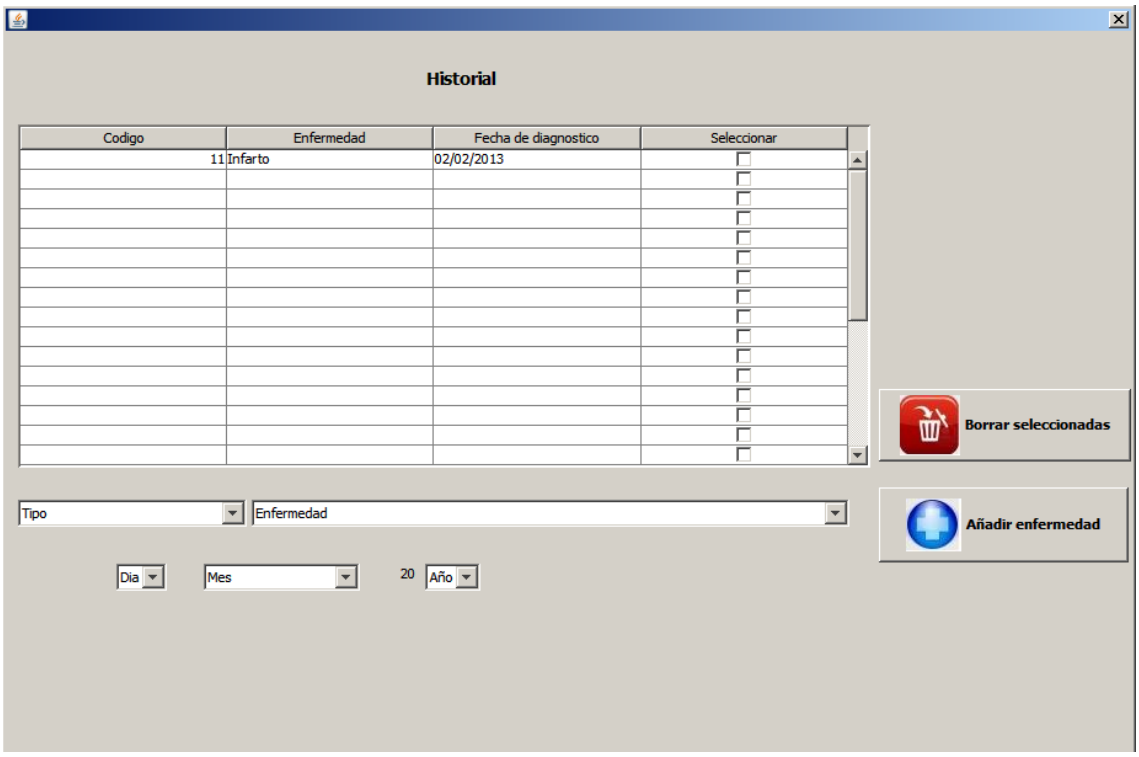

Figura 16: Pantalla del historial medico

Si quedara algún campo sin establecer cuando se pulsa el botón de añadir enfermedad, la herramienta desplegará la siguiente ventana de error:

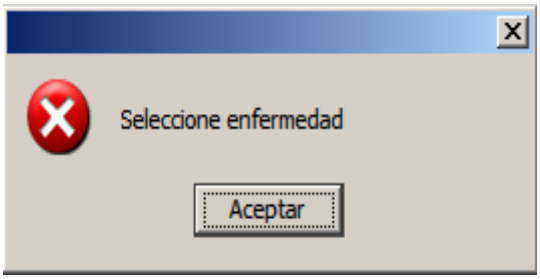

Figura 17: Mensaje de error por falta de selección

También es posible eliminar una o varias enfermedades del historial médico, usando la columna de selección para marcarlas y pulsando el botón de borrado.

7.6 Pantalla vacunas

Esta pantalla permite consultar la cartilla de vacunación del paciente, ofreciendo la posibilidad de hacer un seguimiento de las dosis suministradas de cada vacuna.

A nivel de usuarios, el centro médico tiene acceso a todos los campos pudiendo añadir nuevas vacunas, borrar existentes o incrementar la dosis de alguna vacunación ya iniciada.

El paciente sólo puede acceder a esta pantalla a modo informativo sin capacidad de modificación de ninguno de los campos, y la farmacia puede incrementar la dosis de una vacuna ya empezada; pero no tiene permiso para borrar o añadir vacunas nuevas.

La siguiente figura muestra la pantalla de vacunación:

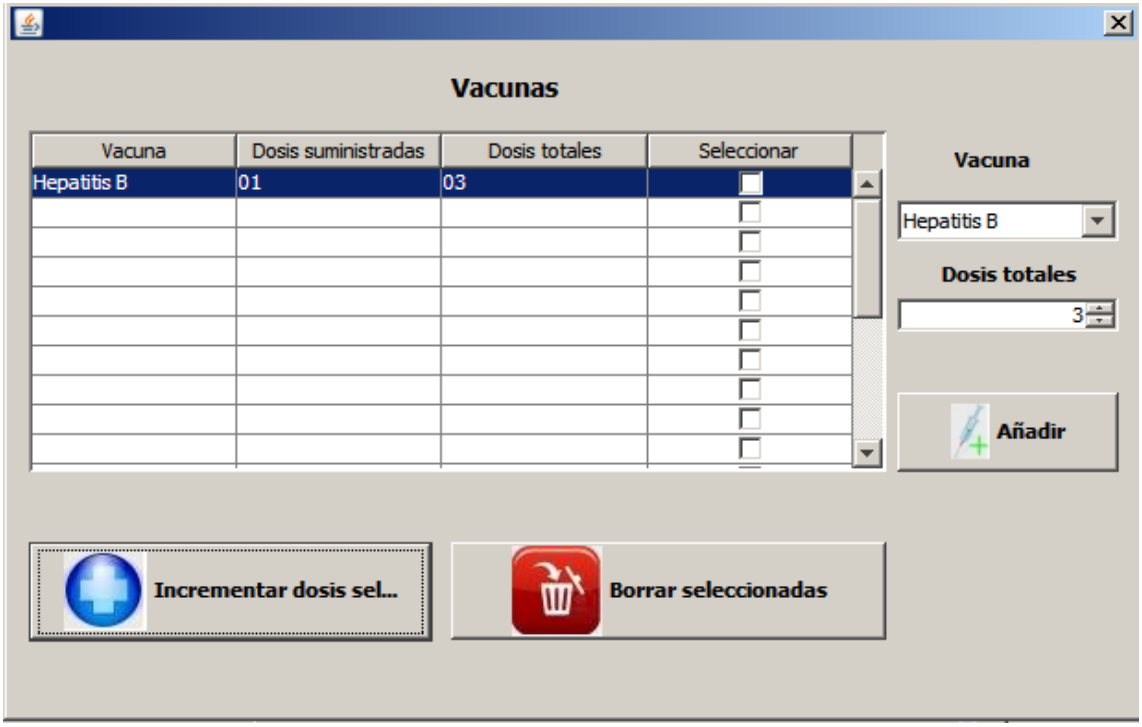

Figura 18: Pantalla de vacunación

7.7 Pantalla recetas

La pantalla más compleja de la herramienta es la encargada de gestionar los medicamentos recetados por el centro médico. Esta pantalla muestra:

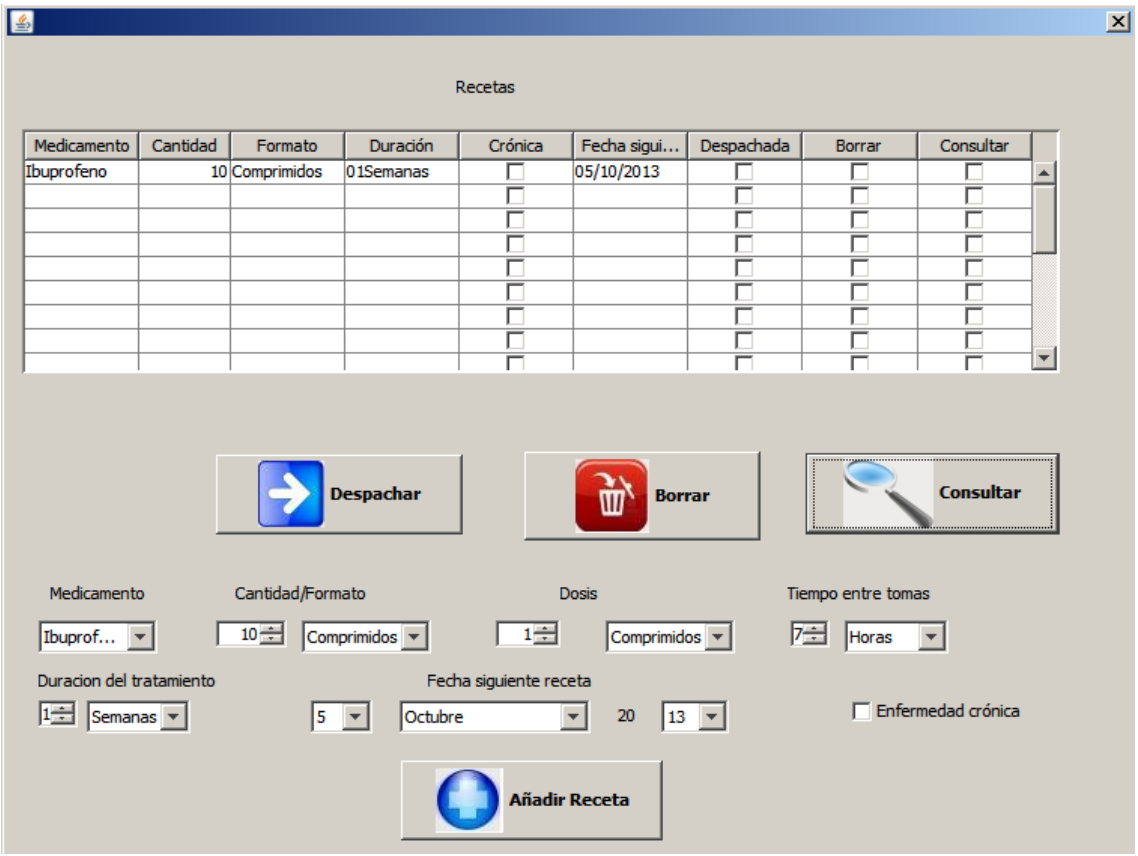

Figura 19: Pantalla de recetas

- El cuadro principal muestra los medicamentos ya recetados, exponiendo la siguiente información: Medicamento, cantidad recetada del mismo, formato de dicha cantidad, duración del tratamiento, indicador de enfermedad crónica que requiere actualización automática de la receta, fecha de la siguiente receta, indicador de que la receta fue despachada ya, casilla para seleccionar las recetas a borrar y las recetas a consultar.
- Botones que sirven para modificar o consultar recetas ya almacenadas en la tarjeta:

- Despachar: Solo accesible para el médico y la farmacia, es usado para marcar una receta como despachada y que no sea dispensada dos veces por dos farmacias.
- Borrar: Solo accesible por el centro médico, borra las recetas marcadas en la columna “borrar”.
- Consultar: Accesible por el paciente permite acceder de forma sencilla a la información del tratamiento, de modo que el paciente pueda saber, por ejemplo, cada cuanto debe tomar la medicina o en qué cantidad. La siguiente figura muestra un ejemplo de cómo se despliega esta información:

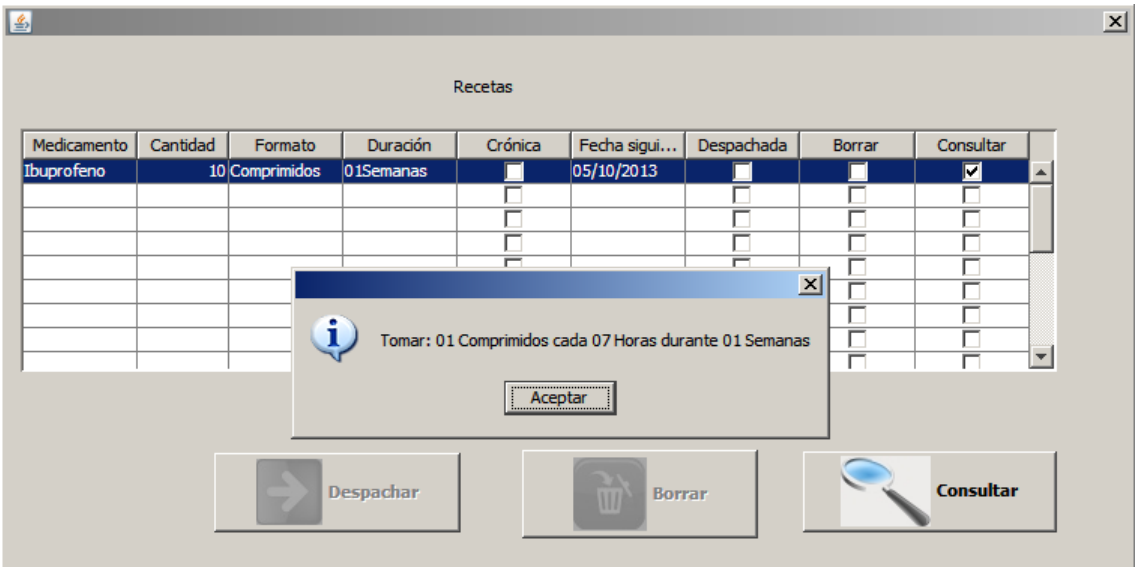

Figura 20: Ejemplo de consulta de tratamiento

- Desplegables para introducir una nueva receta en la tarjeta: Permite al médico almacenar recetas en la tarjeta del paciente, debiendo introducirse todos los campos antes de pulsar el botón de añadir receta; en caso contrario la herramienta mostrará el correspondiente mensaje de error, como muestra la siguiente figura:

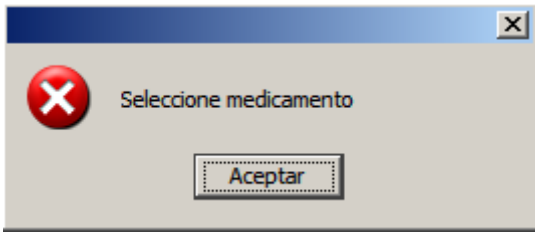

Figura 21: Mensaje de error al introducir una receta incompleta

- Botón de añadir receta: Para almacenar la receta en la tarjeta del paciente.

7.8 Pantalla alergias

La última de las pantallas, mostrada en la figura inferior, permite acceder a la información relativa a las alergias del paciente:

Figura 22: Pantalla de alergias

Los campos que componen la pantalla son:

- Tabla de alergias: Con el nombre y el grupo de las alergias diagnosticadas al paciente, así como la columna que permite seleccionar las alergias a borrar.
- Menús desplegables: Para poder seleccionar el grupo alérgico y la alergia dentro de ese grupo.
- Botones: Para añadir una nueva alergia o borrar una existente.

8. Conclusiones y líneas de futuro

8.1 Conclusiones

La realización de este proyecto ha demostrado que es posible implementar un sistema de gestión sanitaria basada en Tarjetas Inteligentes. Lo que conlleva una serie de ventajas:

- Ahorro económico: Al no necesitar imprimir las recetas, ya que toda la información necesaria está almacenada en la tarjeta.
- Resolución de emergencias sanitarias: Los médicos de urgencias tendrán acceso inmediato al historial médico del paciente, incluyendo las alergias del mismo, así como a su grupo sanguíneo, lo que puede salvar vidas.
- Mejora de la calidad de vida del paciente: Por varios motivos, el primero en caso de recetas crónicas no tendrá que acudir a su centro médico a por dicha receta, ya que se renovarán automáticamente y solo deberá acudir a la farmacia. Segundo, podrá consultar cualquier tratamiento en su domicilio, teniendo acceso a las dosis de medicamento recetados así como a la duración del tratamiento, dosis,…
- Personas dependientes: La persona al cargo personas dependientes solo tendrá que consultar los tratamientos almacenados en la tarjeta de cada paciente, evitándose confundir expedientes o recetas.

8.2 Líneas de futuro

Actualmente hay una nueva tecnología, NFC o “contactless” que está de moda en varios sectores, como por ejemplo, transporte, pago,…

Se podría evolucionar este proyecto hacia esa tecnología, en dos pasos, el primero sería hacer funcionar este mismo modelo sobre tarjetas dual-interface que permitan la

comunicación sin contactos, lo que implicaría que el usuario no debería introducir la tarjeta en un lector, bastaría con acercarla.

Y el segundo paso, mucho más ambicioso, sería poder instalar la aplicación de gestión sanitaria en una tarjeta SIM insertada en un teléfono con NFC. Esto permitiría, no solo, usar el teléfono como una tarjeta dual-interface como se describe en el párrafo anterior, sino que además se podría usar cualquiera de los APIS existentes que definen la comunicación teléfono-SIM para poder usar la capacidad del teléfono para mostrar información. Un ejemplo podría ser que el teléfono despliegue una alarma cada vez que haya que tomar el medicamento “X” del tratamiento actual.

Otra gran ventaja, es que al estar la aplicación instalada en la tarjeta SIM, se podrían usar los servicios del operador de telefonía para proporcionar funcionalidades extra, como por ejemplo poder pedir citas médicas.

A.1 Presupuesto

Descomposición de las actividades:

- Fase 1: Documentación para realizar el proyecto y estudio del arte.
- Fase 2: Aprendizaje de los lenguajes de programación y familiarización con las TI de los distintos fabricantes.
- Fase 3: Implementación del applet y del interfaz gráfico.
- Fase 4: Testeo y depuración del código fuente.
- Fase 5: Documentación y memoria

Las siguientes tablas recogen el presupuesto ofrecido por el desarrollo del PFC, se destaca que el precio final sería el precio que ofrecería una consultoría, no el precio real gastado en el desarrollo.

PRESUPUESTO DEL PROYECTO

1. Autor: **Pablo de Antonio Herrero**
2. Departamento: **Dpto. De Tecnología Electrónica**.
3. Descripción del proyecto.

Se diseñará y desarrollará un sistema de gestión sanitaria, médico – paciente – proveedor de servicios, mediante el uso de tarjetas inteligentes JavaCard. Se implementará una herramienta a modo de interfaz gráfica para poder gestionar la información almacenada por el applet de gestión sanitaria residente en la Tarjeta Inteligente.

- Título : **Sistema de Gestión Sanitaria mediante tarjetas JavaCard**
- Duración (meses): **20**

Tasa de costes indirectos: **20%**

4. Presupuesto total del proyecto (en Euros).

66.837,02 Euros

5. Desglose presupuestario (costes indirectos).

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar – solo a título personal)	Categoría	Dedicación (hombres / mes)	Coste hombres mes	Coste (Euro)
Sánchez Reíllo, Raúl		Ingeniero Sénior	1	4.289,54	4.289,54
De Antonio Herrero, Pablo		Ingeniero	19	2.694,39	51.193,41
				Total	**55.482,95**

EQUIPOS Y SOFTWARE

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación (meses)	Coste imputable
Lector de tarjetasACR38	19.75	100	15	60	4.94
PC de mesa con Windows XP	600	100	20	60	200
				Total	**204,94**

Fórmula de Cálculo de la Amortización:

A/B x C x D

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto (habitualmente 100%)

OTROS COSTES

Descripción	Coste (Euro)	Cantidad	Coste imputable
SmartCafe 3.2 72KB	5.50	2	11.00
		Total	**11,00**

6. Resumen de costes.

Descripción	Presupuesto Costes Totales
Personal	55.482,95
Amortización	204,94
Costes Indirectos	11.138,13
Otros costes	11,00
Total	**66.837,02**

El presupuesto total de este proyecto asciende a la cantidad de SESENTA Y SEIS MIL OCHOCIENTOS TREINTA Y SIETE EUROS CON DOS CENTIMOS.

Leganés, Octubre de 2013

El ingeniero proyectista

Fdo. Pablo de Antonio Herrero

A.2 Bibliografía

Doc Ref	Título del documento	Versión	Autor
[ISO/IEC 7816-1]	ISO/IEC 7816-1:2005 Identification cards — Integrated circuit cards — Part 1: Physical Charcteristics of Integrated Circuit Cards	Second edition	ISO
[ISO/IEC 7816-4]	ISO/IEC 7816-4:2005 Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange	Second edition	ISO
[GPCS]	GlobalPlatform – Card Specification	2.1.1	GlobalPlatform
[EMV]	EMV 4.3. Book 3. Application Specification	4.3	EMV
[CIE]	Wiki con todos los códigos internacionales de enfermedades http://cie10.tiddlyspot.com/	--	Universidad Católica de Perú
[ACR38]	Data sheet del lector ACR38 : http://www.acs.com.hk/index.php?pid=product&id=ACR38	--	ACS
[GPSHELL]	Wiki con toda la información del GP Shell: http://sourceforge.net/p/globalplatform/wiki/Home/	--	--
[SMARTCAFE]	Información sobre la Smart Café Expert 72k de G&D: http://www.Tarjeta Inteligentefocus.com/ilp/id~45/The_SmartCafe_Expert_Range/p/cards.shtml	--	G&D
[JAVACARD]	JavaCard Platform Specification v2.2.2: http://www.oracle.com/technetwork/java/javacard/specs-138637.html	2.2.2	Oracle
[NETBEANS]	Tutorial de NetBeans: https://netbeans.org/kb/docs/java/quickstart.html	--	Oracle