CONTENIDO

ANEXO F *GUIA DE USO DE LA PLATAFORMA*

LISTA DE FIGURAS

ANEXO F
GUIA DE USO DE LA PLATAFORMA

1. CONFIGURACION DE JAVACARD KIT

Sun Microsystems distribuye su herramienta de libre distribución para el desarrollo de Applets JavaCard denominada JavaCard Development Kit, para nuestro trabajo de grado utilizamos la versión java_card_kit-2_2_1. Esta herramienta se encuentra como un archivo .zip el cual debemos descomprimir en un lugar de nuestra elección, por ejemplo, C:\javacard\java_card_kit-2_2_1\. Realizado este paso obtenemos un grupo de ficheros como el mostrado en la figura 1-1.

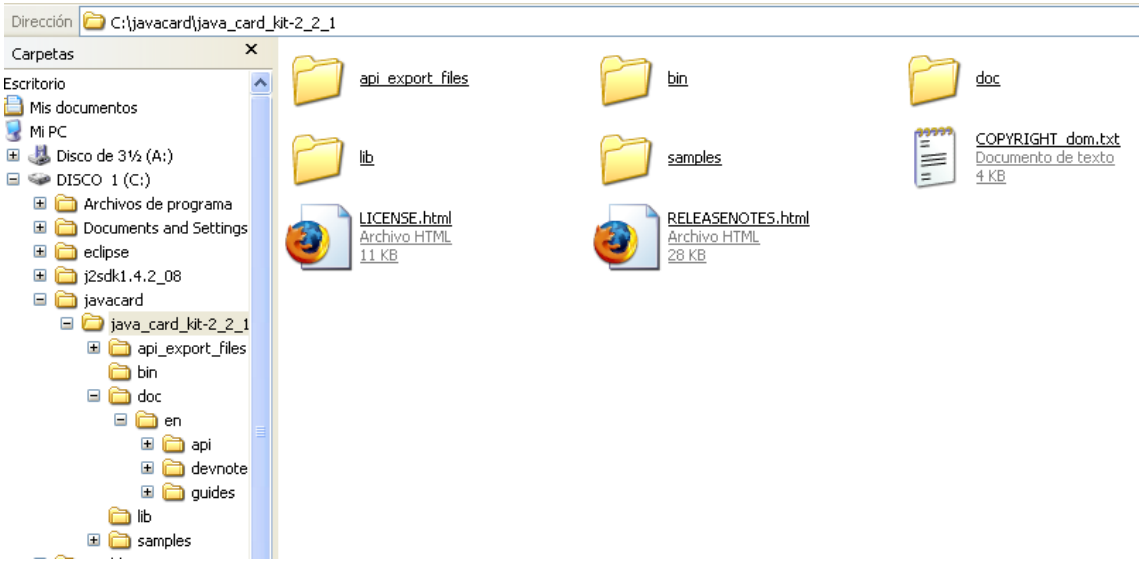

Figura 1-1. Sistema de archivos JavaCard kit

Terminado este proceso es necesario configurar las variables de entorno en nuestra máquina, estas variables son JAVA_HOME, JC_HOME, las cuales apuntan hacia el directorio principal de j2sdk1.4 y al directorio principal de java_card_kit-2_2_1 respectivamente.

La figura 1-2 muestra este proceso para el equipo utilizado en el desarrollo de la plataforma P3SIM.

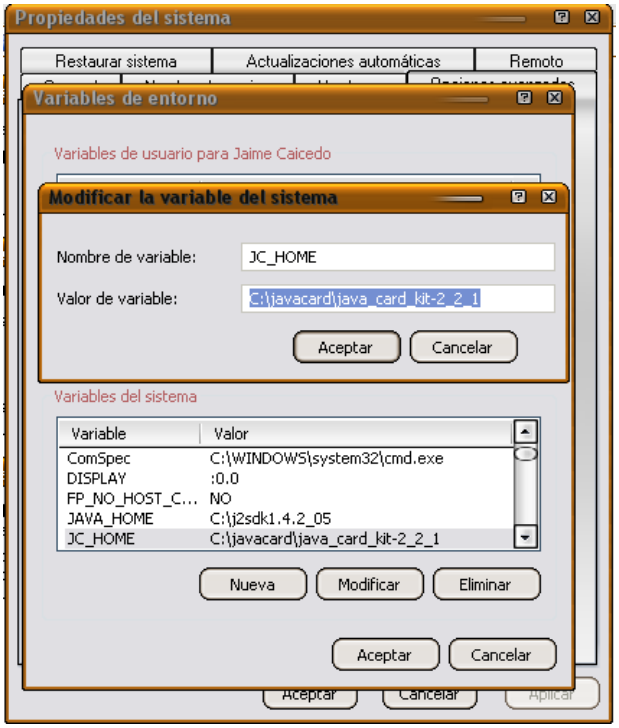

Figura 1-2. Variables de entorno

Es necesario tener en cuenta que el kit de desarrollo que nos proporciona Sun Microsystems posee una herramienta para simulación de tarjetas inteligentes JavaCard, pero no nos brinda soporte para tarjetas SIM JavaCard. Su utilidad principal consiste en la posibilidad de emulación de un dispositivo seguro al cual se puede acceder desde un emulador de dispositivos móviles J2ME como lo es el Sun Wireless Toolkit.

2. CONFIGURACION DE ASPECTS DEVELOPER

Aspects Developer es una herramienta para el desarrollo de aplicaciones sobre tarjetas SIM con soporte para JavaCard, este ambiente de desarrollo esta disponible en una versión de prueba de libre distribución. Para el desarrollo de nuestro trabajo de grado utilizamos Aspects Developer Evaluation Version Rev. 02.12.02.

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.

El proceso de instalación es automático puesto que se realiza mediante un Wizzard que nos guía a través de este proceso, como se muestra en la figura 2-1.

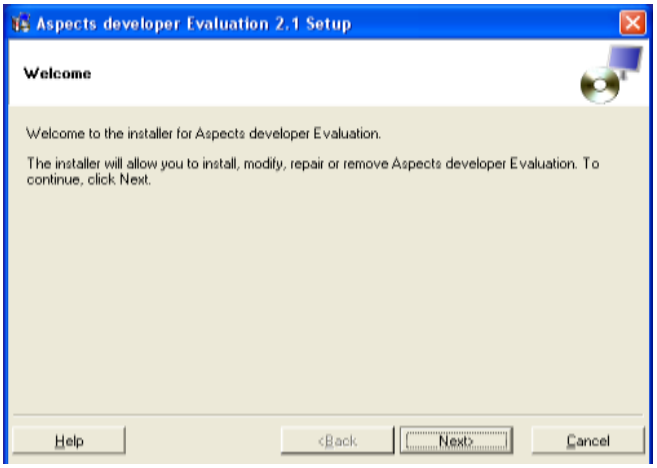

Figura 2-1. Wizzard Aspects Developer

Después del proceso de instalación tenemos un sistema de archivos como el mostrado en la figura 2-2.

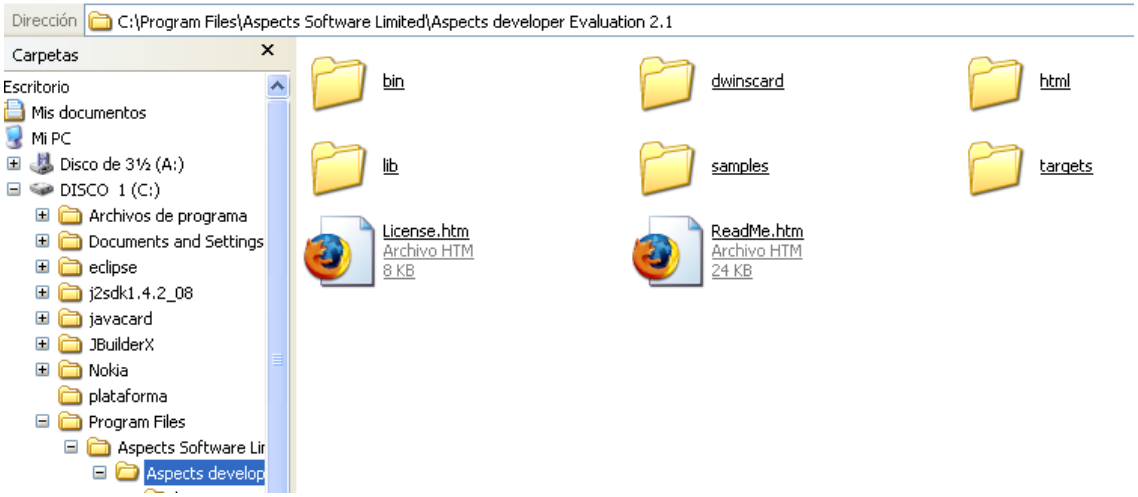

Figura 2-2. Sistema de archivos Aspects Developer

Es necesario tener en cuenta que este ambiente permite la simulación de las aplicaciones desarrolladas para la tarjeta SIM mediante un móvil y algunas otras herramientas destinadas a la comunicación con la aplicación.

Para poder simular las aplicaciones desarrolladas con este ambiente se necesita del Java Runtime Environment jre 1.3.1. De lo contrario obtendremos un mensaje de error pues es imposible trabajar con una versión superior a la descrita. La siguiente figura muestra la simulación de una aplicación SIM Application Toolkit.

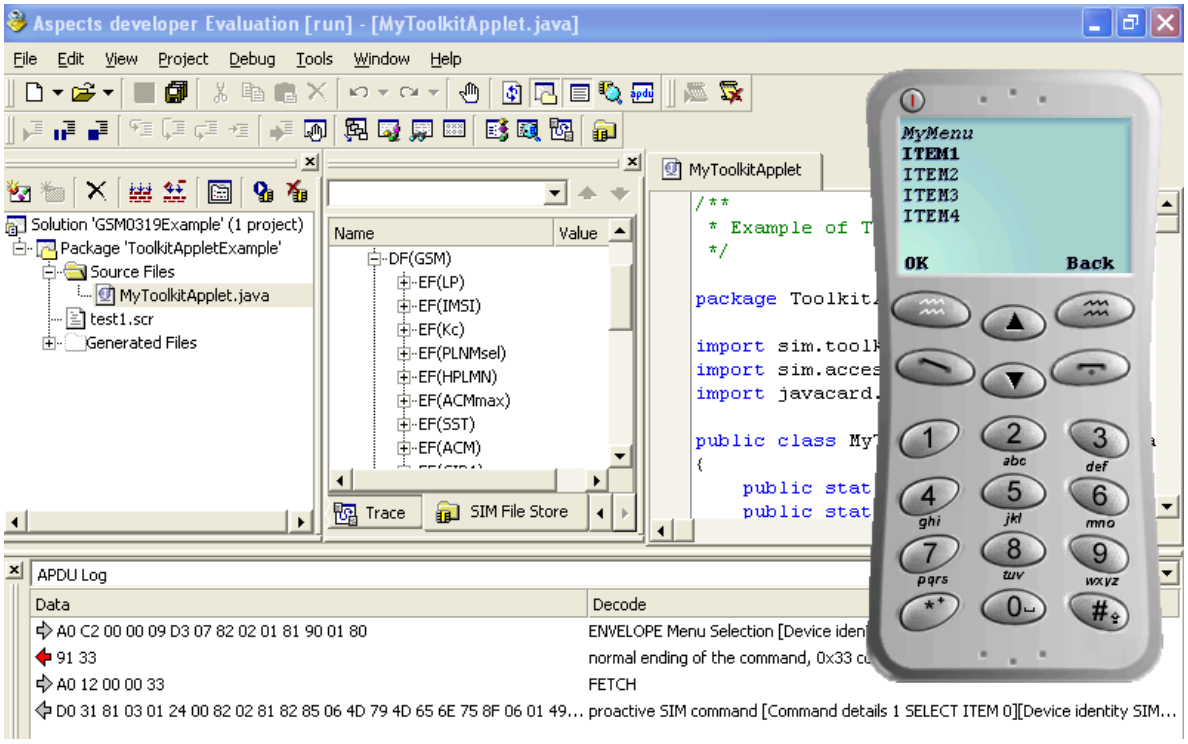

Figura 2-3. Aspects Developer

3. CONFIGURACION DE ECLIPSE

Eclipse es una herramienta de libre distribución y de código abierto distribuido por la comunidad de desarrollo Eclipse.org.

Es una herramienta versátil que posee una gran aceptación a nivel mundial por parte de los desarrolladores Java.

El proceso de instalación es muy sencillo, consiste en descomprimir el archivo de distribución en un lugar de nuestra preferencia y teniendo en cuenta la necesidad del Java Runtime Environment.

Para facilitar el proceso de compilación y generación de los scripts necesarios para la carga de Applets JavaCard en el emulador de tarjetas inteligentes del javacard_kit se desarrollo el archivo build.xml que consta de una serie de tareas Ant que permiten llevar a cabo estas tareas.

A continuación se muestra el proceso para la creación de un nuevo builder en un proyecto de desarrollo en Eclipse3.1, el ejemplo mostrado fue tomado del desarrollo

del SECApplet perteneciente a la plataforma P3SIM. La figura 3-1 muestra la creación del nuevo builder de Eclipse.

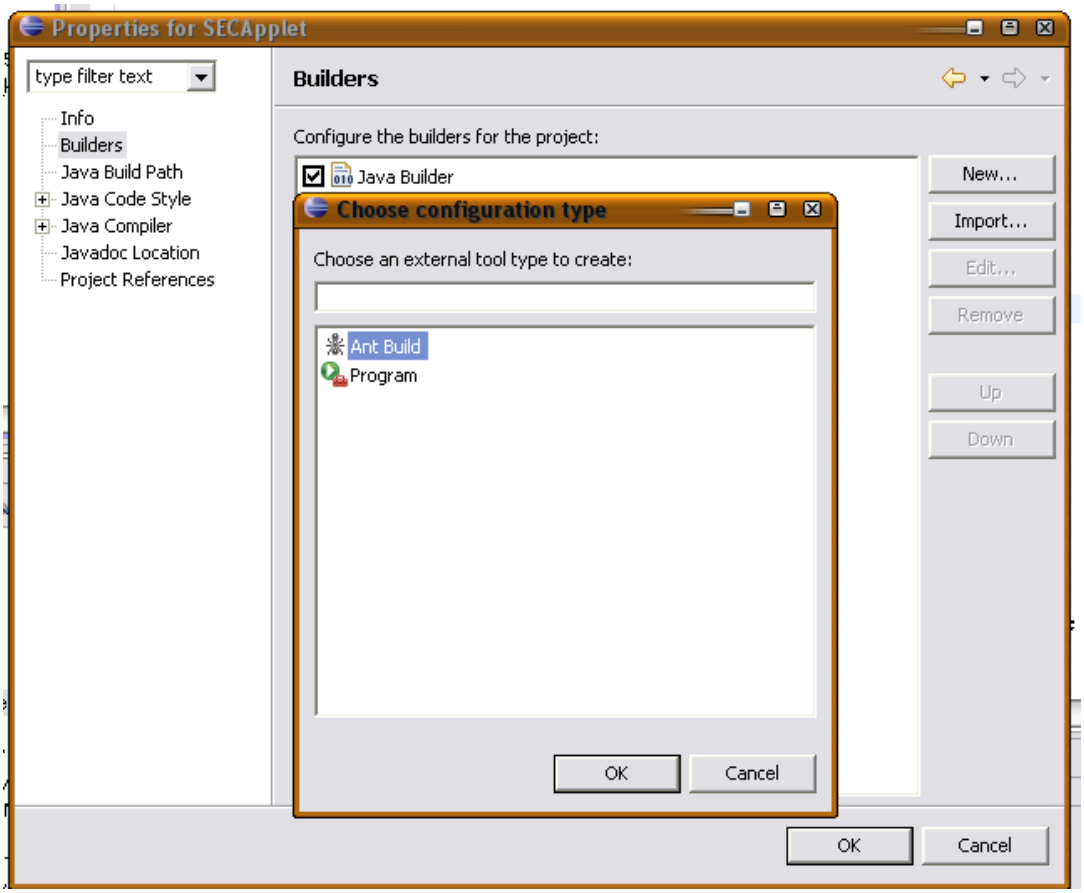

Figura 3-1. Creación de nuevo Builder de Eclipse

Luego seleccionamos un nombre para nuestro nuevo builder y seleccionamos el archivo *.xml que contiene las tareas Ant a llevar a cabo, en este caso el archivo es biuld.xml. El proceso se muestra en la figura 3-2.

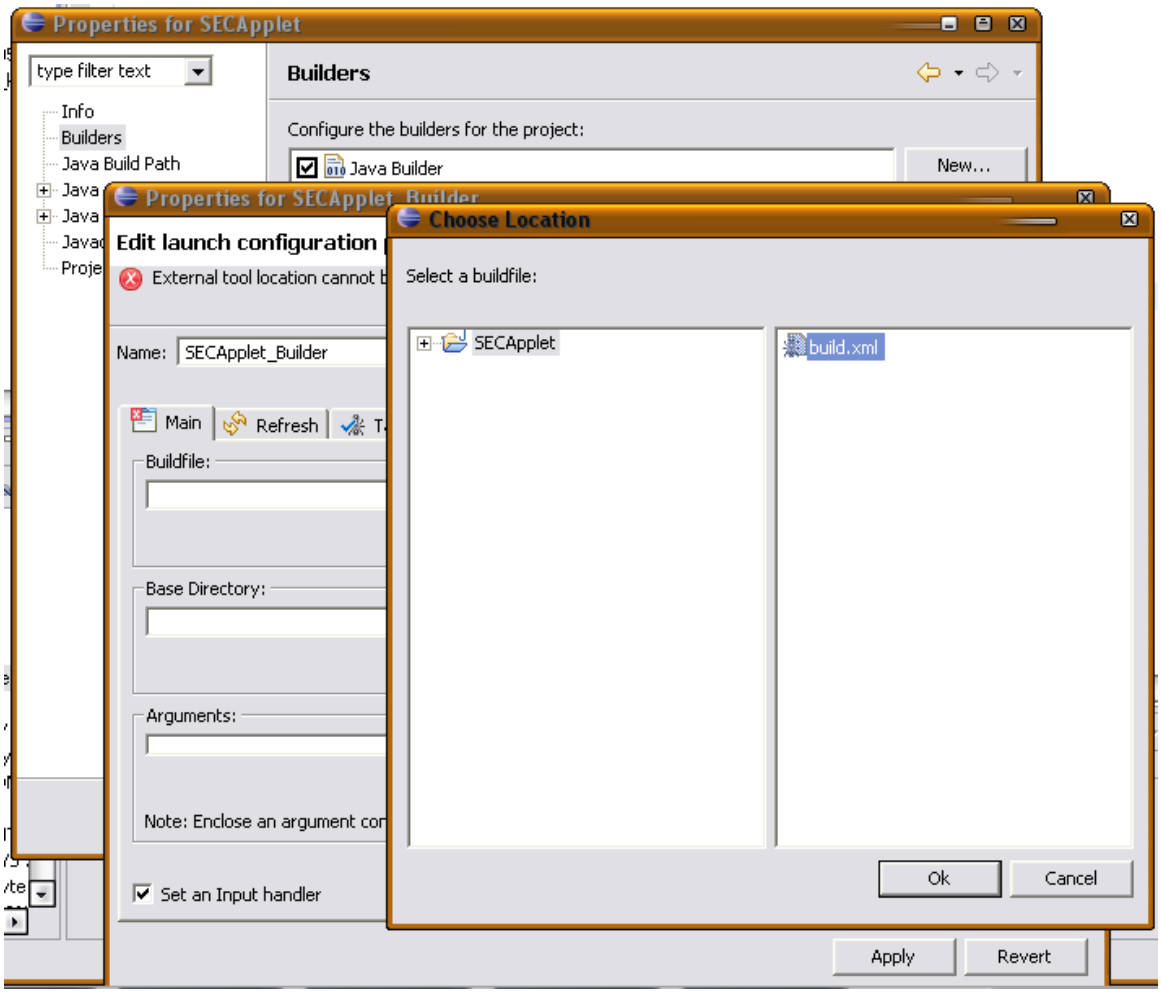

Figura 3-2. Selección del archivo build.xml

Terminado este paso ya tenemos configurado nuestro builder para el proyecto de Eclipse, ahora solo nos resta añadir las librerías que contienen las clases necesarias para el desarrollo de aplicaciones JavaCard. Estas librerías se encuentran en el JC_HOME\lib\api.jar.

El proceso de adición de las librerías para el desarrollo de aplicaciones JavaCard se muestra en la figura 3-3.

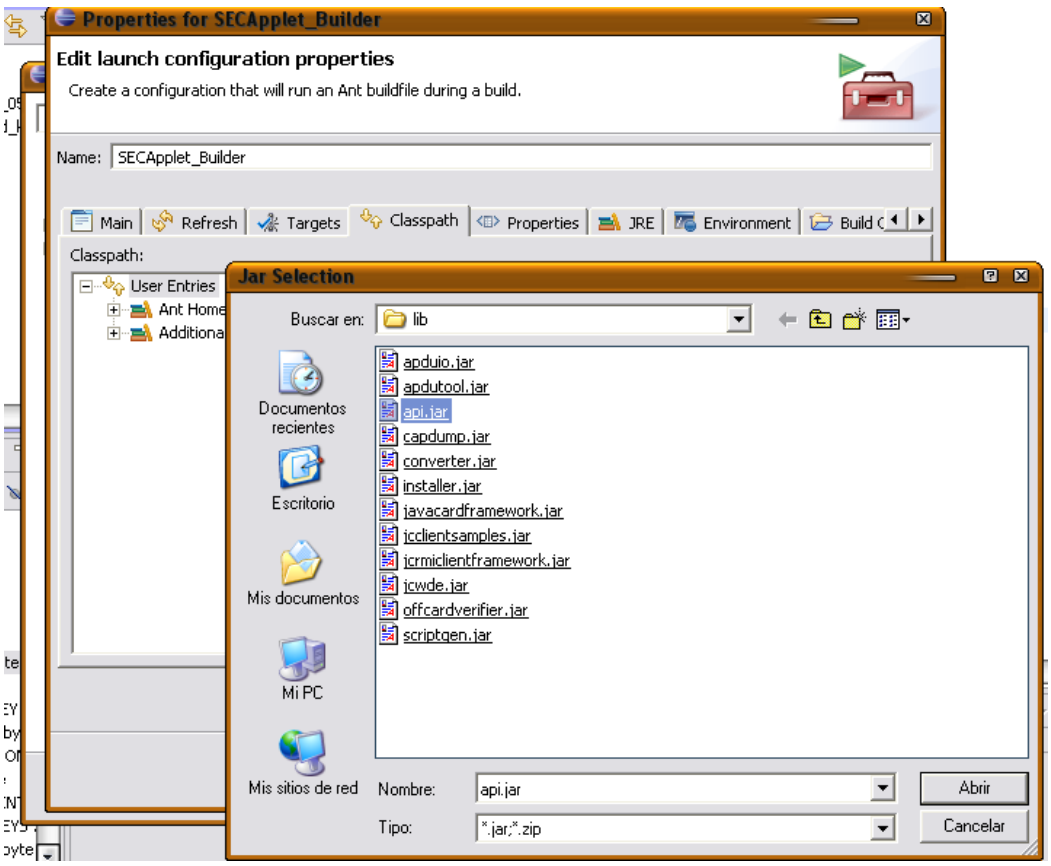

Figura 3-3. Adición de librerías JavaCard

Una vez alcanzado este punto ya tenemos una herramienta para la automatización de los procesos de compilación, verificación, conversión, y generación de scripts en el proceso de desarrollo de aplicaciones JavaCard.

Luego de la ejecución del builder tenemos un sistema de archivos como se muestra en la figura 3-4 donde se destaca el archivo *_script.txt. El cual contiene las APDU necesarias para la instalación del Applet desarrollado en el simulador de tarjetas inteligentes del javacard kit.

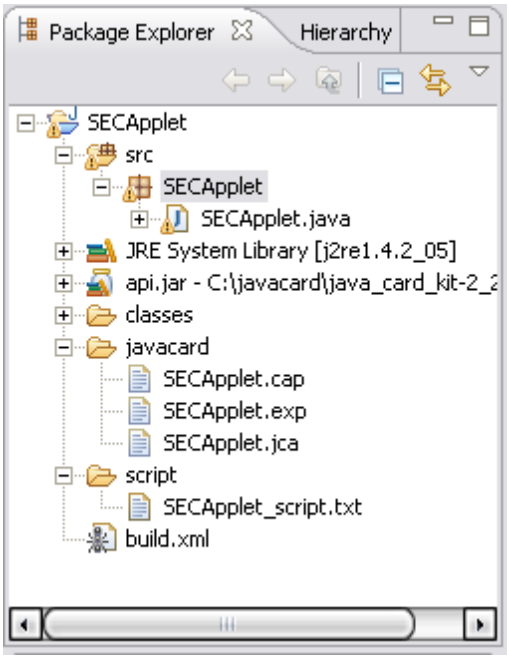

Figura 3-4. Archivos generados por el builder en Eclipse

4. UTILIZACION DE P3SIM

La utilización de los métodos proporcionados por la plataforma P3SIM se realiza de forma directa, puesto los métodos son del tipo *static*. No se hace necesario la declararon e instancia de objetos del tipo P3SIM, sino, que se utilizan de forma directa como se muestra a continuación.

```
    /**
     * get SIM parameters
     */
    byte[] iccid = P3SIM.getICCID();
    byte[] imsi = P3SIM.getIMSI();

    /**
     * cipher some data
     */
    byte[] iccidEncrypted = P3SIM.encryptData(iccid,
P3SIM.DESKEY);
    byte[] imsiEncrypted = P3SIM.encryptData(imsi,
P3SIM.RSA_MY_PRIVATE_KEY);

    /**
     * calculate a signature
     */
    String dataToSign = "this data need signature!!";
    byte[] signature2 =
P3SIM.calculateSignature(dataToSign.getBytes());

    /**
     * verification of signatures
```

```
     */
    boolean validity =
P3SIM.verifySignature(dataToSign.getBytes(), signature2,

P3SIM.RSA_PUBLIC_KEY);
    validity = P3SIM.verifySignature(dataToSign.getBytes(),
signature2,

P3SIM.RSA_MY_PUBLIC_KEY);
```

5. CARGA DE APPLETS SOBRE TARJETAS AXALTO

Gracias a la organización “simagine” http://www.simagine.org se pudo descargar la herramienta de Axalto VIEWS Professional.

Esta herramienta permite compilar Applets, convertirlos y cargarlos sobre tarjetas SIM Java Card. Además tiene exploradores para los archivos GSM y para los paquetes y Applets cargados sobre dicha SIM. La figura 5-1 muestra la vista principal de esta herramienta.

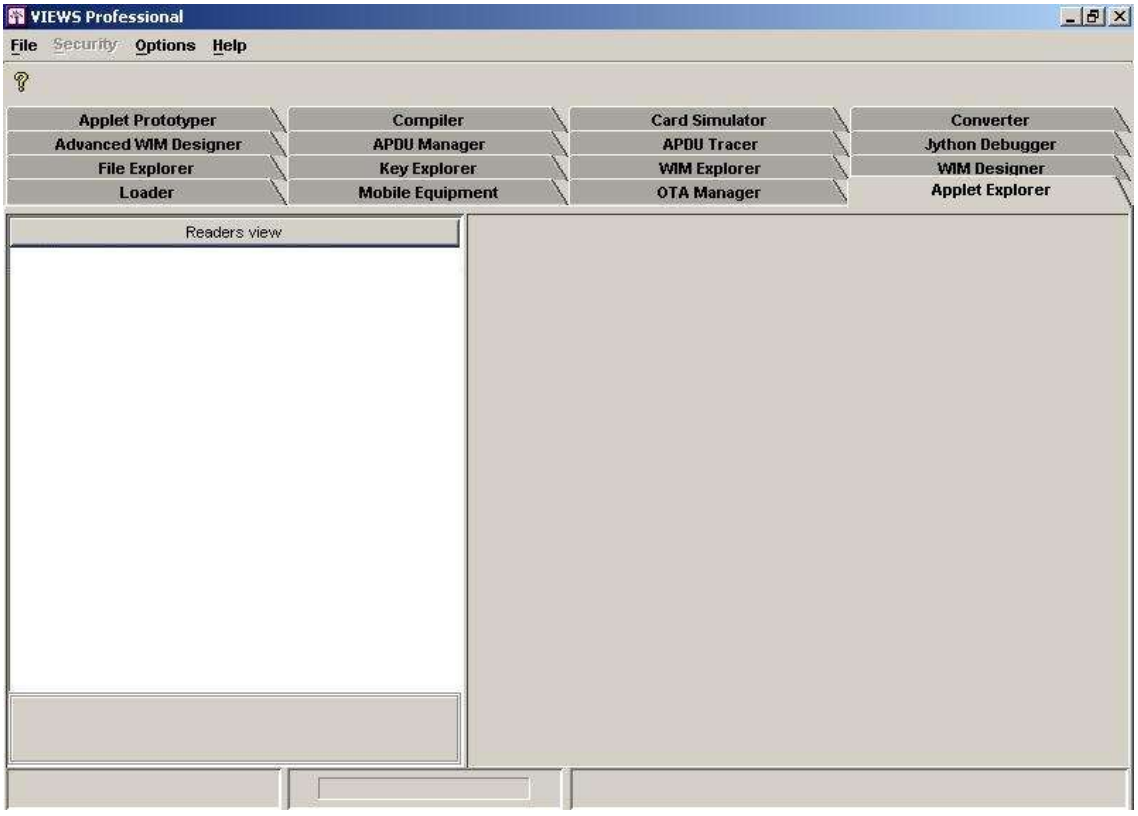

Figura 5-1. VIEWS Professional

Una vez se han instalado los drivers del lector de tarjetas inteligentes, se procede a configurar el conjunto de claves que tiene la tarjeta en particular, para que se puedan cargar applets sobre la misma. Las tarjetas que se manejaron son las “Usimera Classic 3 Linear 128K” las cuales tienen el siguiente conjunto de claves:

```
Key Set: 01
KLA: 0101010101010101
KIC: 01010101010101010101010101010101
KID: 01010101010101010101010101010101
```

La figura 5-2 muestra como configurar estas claves en el VIEWS Professional, después de seleccionar el reader “SchlumbergerSema Reflex 72 v2.0”.

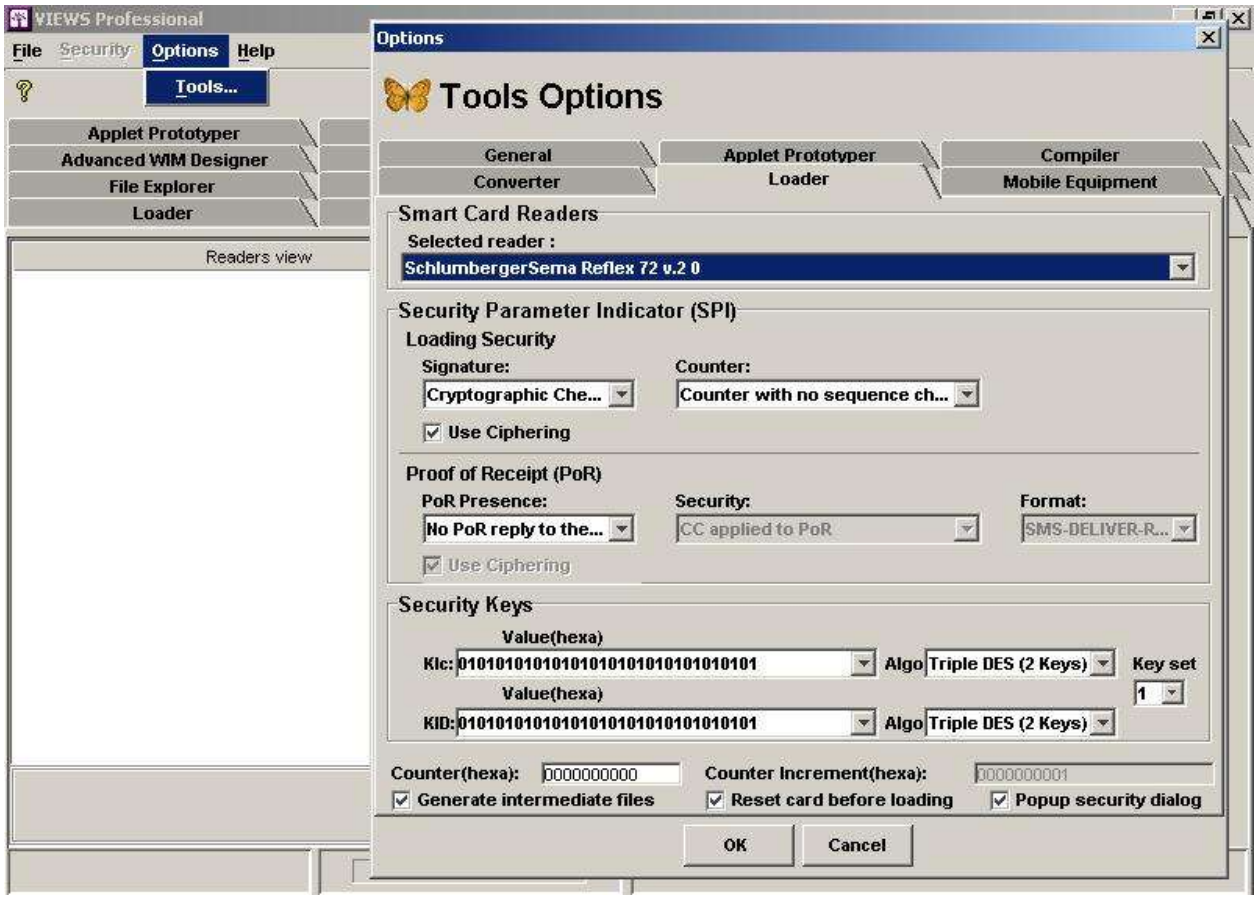

Figura 5-2. Configuración de claves de la tarjeta

A continuación se debe insertar la tarjeta en el lector. Una vez se inserta la tarjeta, el VIEWS Professional la detecta. La figura 5-3 muestra esta detección.

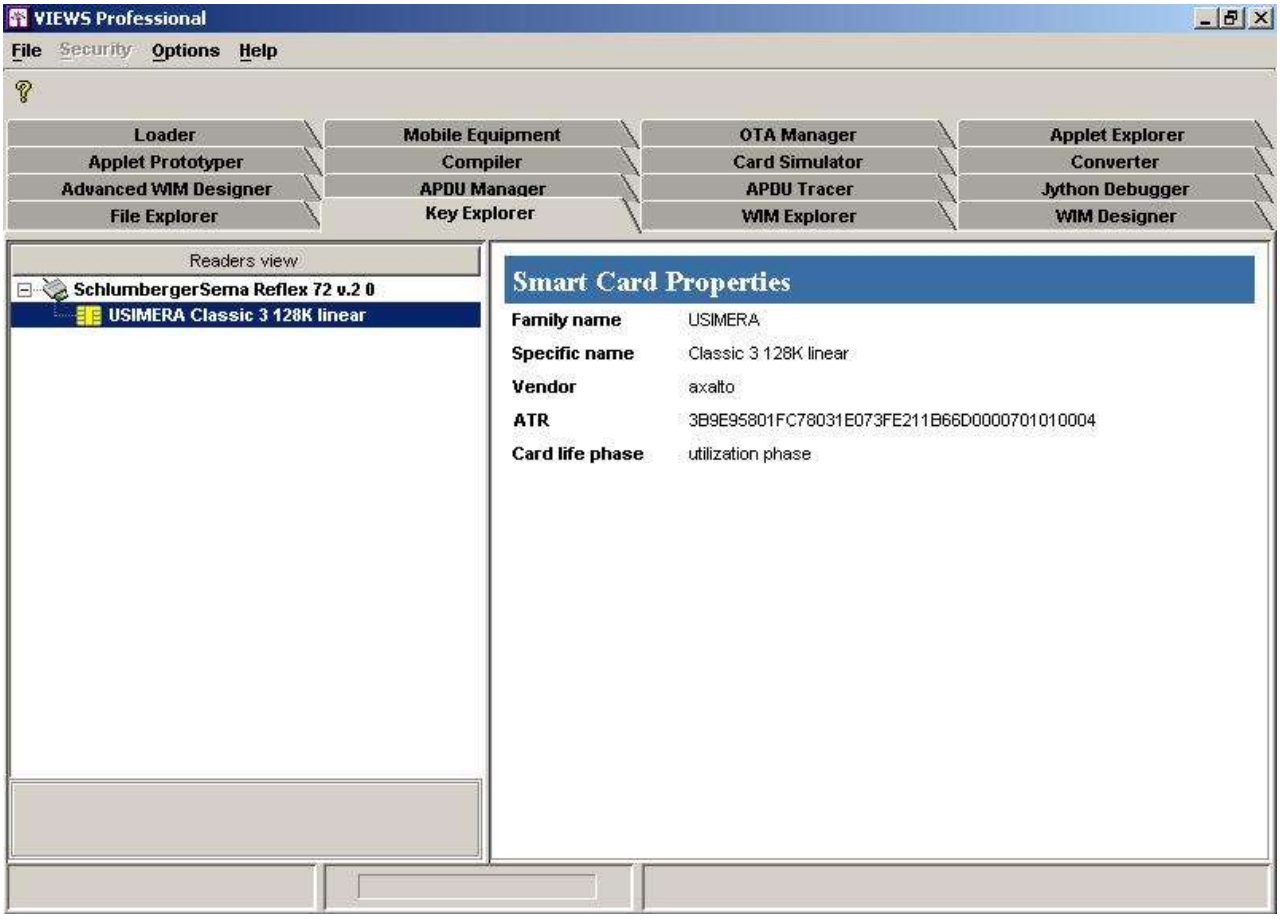

Figura 5-3. Detección de la tarjeta

Luego se puede realizar la exploración de los paquetes y applets instalados en la tarjeta. Para realizar esto de debe seleccionar la pestaña Applet Explorer y posteriormente conectarse a la tarjeta. Para poder conectarse se debe introducir la clave KLA "0101010101010101". La figura 5-4 muestra este procedimiento.

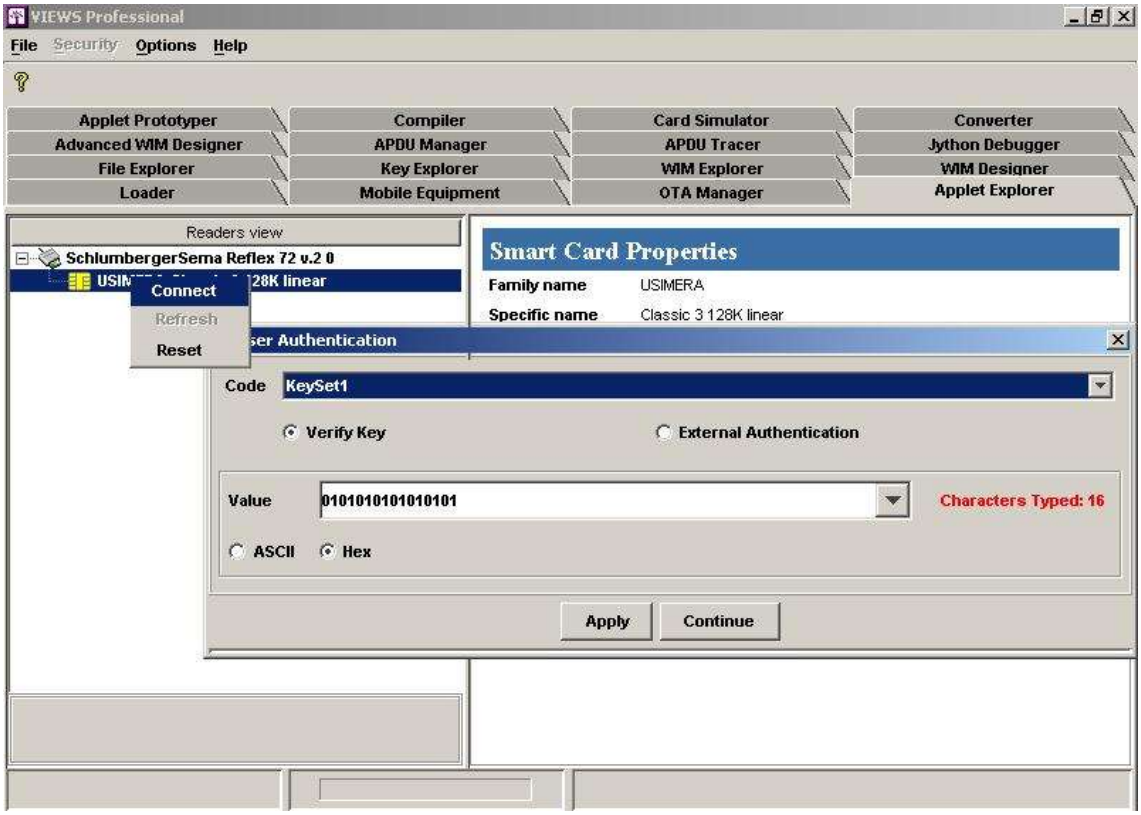

Figura 5-4. Applet Explorer

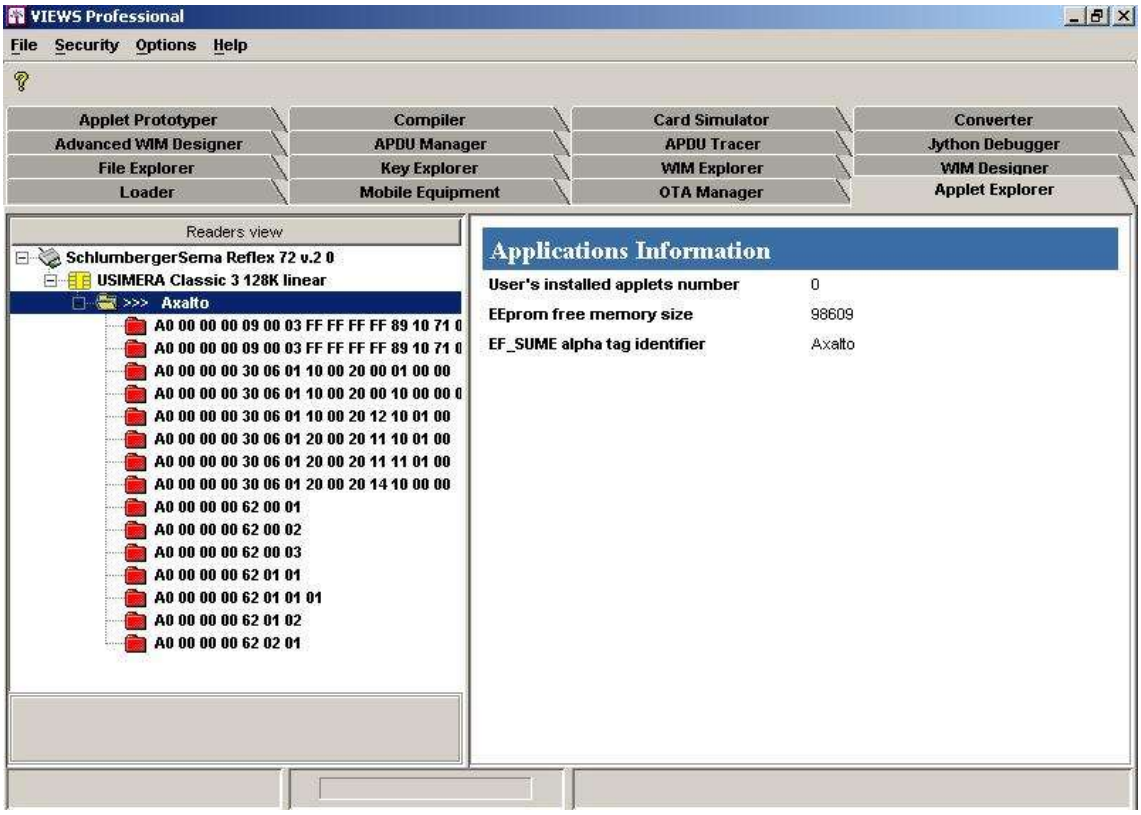

Figura 5-5. Paquetes instalados en la tarjeta

La figura 5-5 muestra el resultado de la conexión con el Applet Explorer. Como se puede observar solo hay carpetas rojas. Estas carpetas corresponden a los paquetes del API Java Card y los APIs de extensión. Si hubiera algún applet instalado, saldría una carpeta de color verde.

Lo que sigue es cargar un applet en la tarjeta. Para hacer esto se debe seleccionar la pestaña Loader y posteriormente seleccionar el archivo .cap que contiene al applet. Finalmente se debe dar clic en el icono verde de la esquina superior izquierda, con lo que se pide la clave KLA. La figura 5-6 muestra este procedimiento.

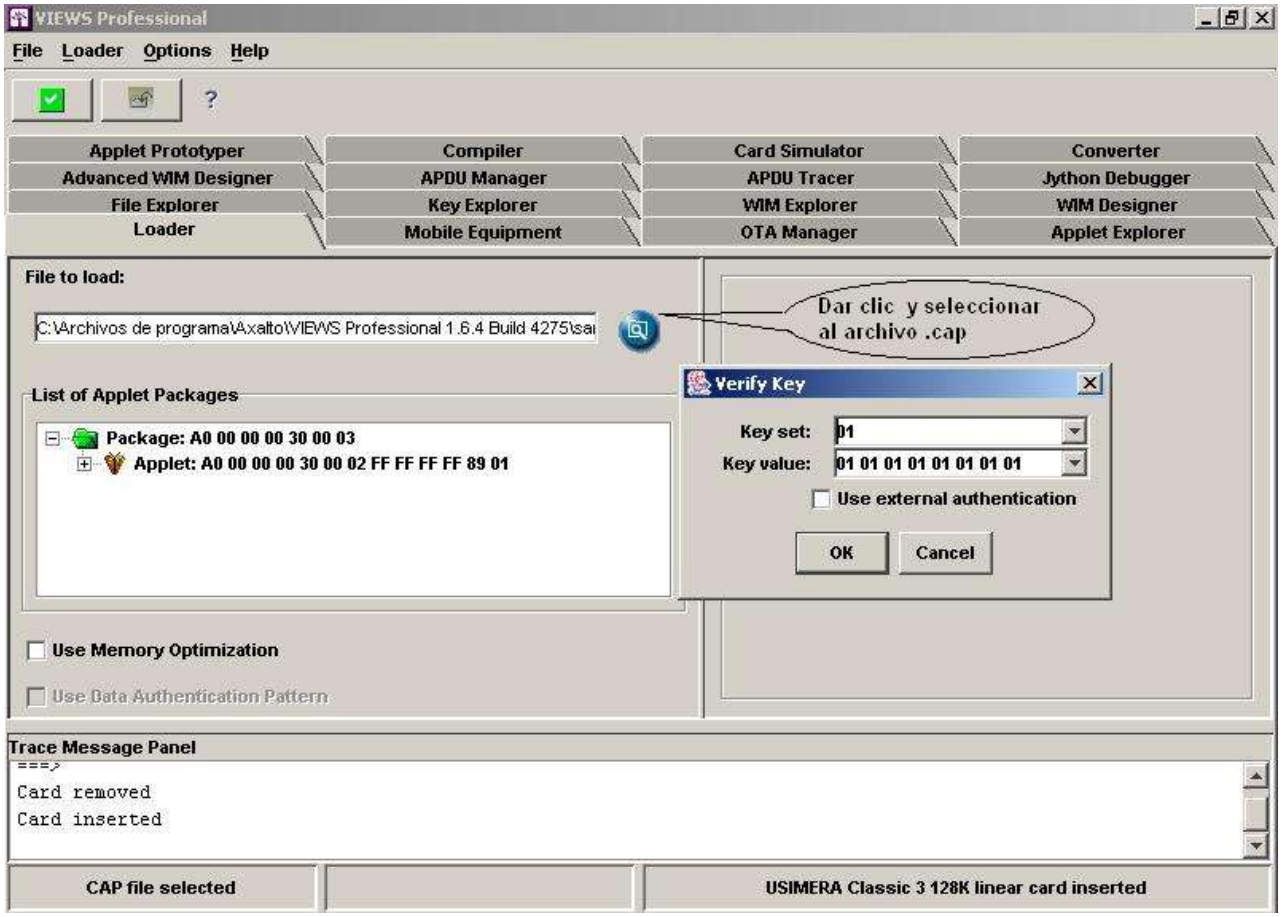

Figura 5-6. Procedimiento para cargar un applet

Una vez se carga un applet sobre la tarjeta, se puede seguir cargando otros o actualizar alguno si se quiere.

Finalmente se puede seleccionar nuevamente la pestaña Applet Explorer y verificar que efectivamente se han cargado los applets. La figura 5-7 muestra los applets cargados.

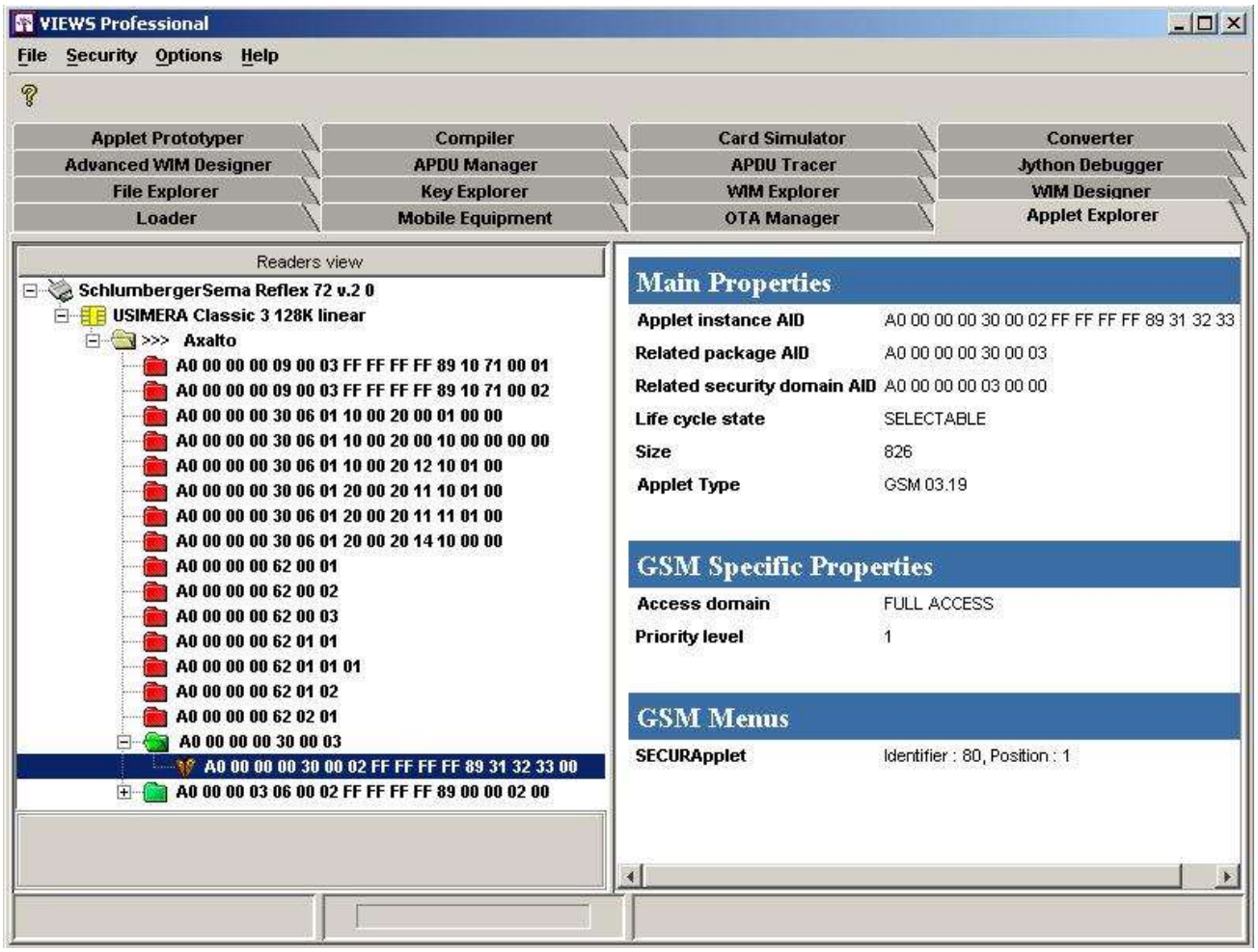

Figura 5-7. Applets cargados sobre la tarjeta

Como se puede ver en la figura anterior, los applets cargados aparecen como carpetas de color verde y las instancias con un icono de una mariposa.