

Uso de Certificados - Programación

Práctica 6A

1. Objetivo

En esta práctica el alumno debe realizar pequeños programas que permitan acceder a un certificado de un almacén y utilizar sus claves asociadas con las protecciones apropiadas.

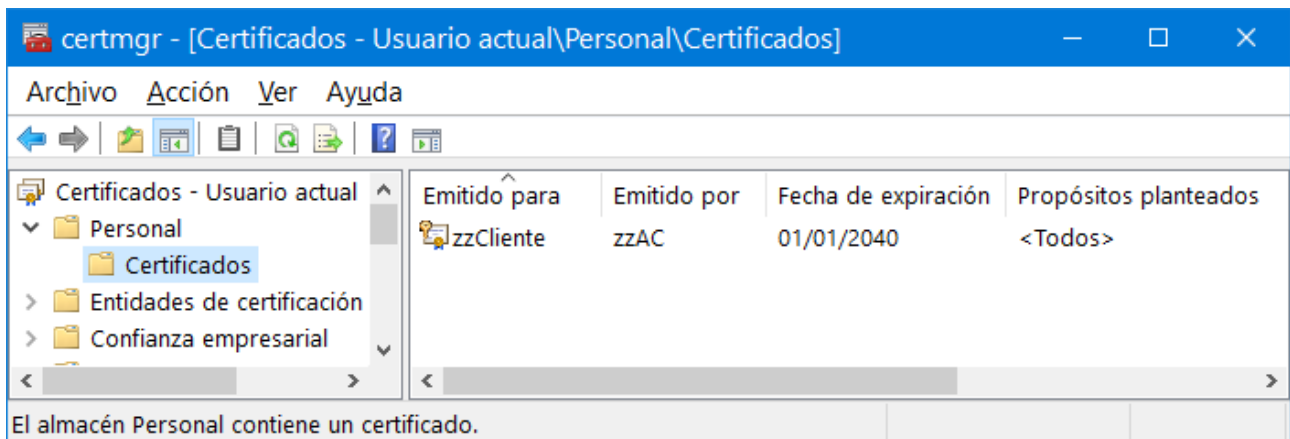
2. Integración del certificado en el almacén de certificados

Carga el certificado contenido en zzCLI.pfx en el almacén de certificados personales del usuario. Si ya estuviera cargado previamente, bórralo y cárgalo nuevamente como se indica en esta práctica. El certificado de la autoridad certificadora (zzAC.cer) que ha firmado el certificado del usuario debe estar cargado en el almacén de certificados "Entidades de certificación raíz de confianza".

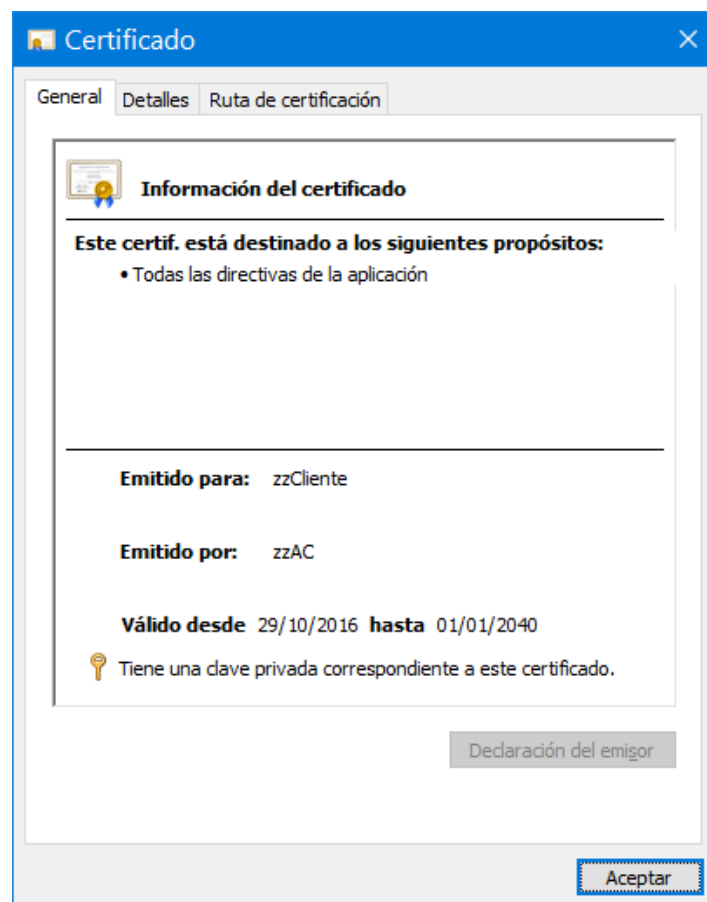
Haz doble clic en el fichero zzCLI.pfx. Se abre automáticamente el asistente para importación de certificados y se llega a la siguiente pantalla:

Selecciona las **dos** últimas opciones y proporciona la contraseña: conclipfx. Elige el almacén de certificados personales del usuario y termina el proceso de importación.

Abre la herramienta de gestión de certificados **certmgr.msc** y comprueba que el certificado está en el almacén de certificados personales.



Haz doble clic en el certificado para que se muestre la ventana de visualización de certificados:



Observa que en la ventana se indica "Tiene una clave privada correspondiente a este certificado".

CONCEPTO: Cuando se importa un certificado desde un fichero .pfx, además de importarse el certificado con su clave pública, también se importa la clave privada asociada al certificado en el almacén de claves privadas correspondiente.

3. Utilización del certificado y su clave privada asociada

Realiza un programa que permita acceder al almacén de certificados en el que Windows almacena los certificados del usuario, para extraer el certificado y usar sus claves. **Reutiliza el programa de la práctica anterior.**

Crea el array de bytes `Msg`, de 64 bytes, que contenga los bytes `0x00` a `0x3F`.

FASE.- ACCESO A UN ALMACÉN DE CERTIFICADOS

- 1) Crea el almacén de certificados `AlmaUser` usando como nombre `My` y como localización `CurrentUser`. Abre `AlmaUser` con el método `Open()` usando los flags `ReadOnly` y `OpenExistingOnly`. Después, usa la propiedad `Certificates` del objeto `AlmaUser` para cargar los certificados que contiene en la colección `CertsUser` de la clase `X509Certificate2Collection`.
- 2) Muestra en la consola el número de elementos de `CertsUser` y luego el sujeto (`Subject`) de cada certificado recorriendo los certificados con un bucle `foreach`.
- 3) Cierra el almacén de certificados invocando al método `Close()`.
- 4) Declara el objeto `Cert` de la clase `X509Certificate2` y asigne `CertsUser[i]`. Donde `i` es el número que corresponde al certificado de `zzCliente`. Si es el tercer certificado que se ha mostrado, entonces `i` debe ser dos.

ALTERNATIVA: También puedes utilizar el método `Find()` de la colección `CertsUser` para buscar un certificado y luego extraerlo. Este método devuelve una colección de certificados. Se puede comprobar que la colección contiene un solo elemento y asignarlo entonces al objeto `Cert`. Lo correcto es usar `Find()` para buscar el certificado deseado sin depender de los certificados que estén cargados en el almacén de certificados y el orden en el que aparecen en la colección.

FASE.- USO DEL CERTIFICADO

Primeramente, muestra algunas propiedades de `Cert`, como `Subject` y `HasPrivateKey`. Solamente si `HasPrivateKey` es `true`, muestra el algoritmo de la clave privada y su longitud.

- 1) Declara un objeto `ProvRSA1` de la clase `RSACryptoServiceProvider` y asígnale en la misma declaración la clave pública contenida en `Cert`, haciendo el cast apropiado:

```
= (RSACryptoServiceProvider) Cert.PublicKey.Key;
```

Muestra los parámetros de la clave llamando al método `VerParam()` suministrado con la práctica.

El proveedor `ProvRSA1` tiene **1** clave (pública solamente).

2) Declara un objeto ProvRSA2 de la clase [RSACryptoServiceProvider](#) y asígnale en la misma declaración la clave privada asociada a Cert, haciendo el cast apropiado:

```
= (RSACryptoServiceProvider) Cert.PrivateKey;
```

El proveedor ProvRSA2 tiene **2** claves (pública + privada)

3) Comprueba si ProvRSA2 == null. En este caso muestra un mensaje de error y termina la ejecución. En caso contrario muestra un mensaje indicando que se ha creado un proveedor con una clave privada (=completa, pública + privada) y muestra los parámetros de la clave llamando al método VerParam() suministrado con la práctica.

Cifra con la clave pública

4) Cifra el mensaje Msg usando el método [Encrypt\(\)](#) de ProvRSA1 y muestra el texto cifrado en la consola. Comprueba que también puedes cifrar usando ProvRSA2. ¿por qué?

Descifra con la clave privada

5) Descifra el texto cifrado usando el método [Decrypt\(\)](#) de ProvRSA2 y muestra el texto descifrado en la consola. Comprueba que NO puedes descifrar usando ProvRSA1. ¿por qué?

Firma con la clave privada

6) Crea el array de bytes Firma e inicialízalo con lo que devuelve el método [SignData\(\)](#) del proveedor ProvRSA2. Usa el algoritmo "SHA1". Muestra el contenido de Firma en la consola. Comprueba que NO puedes firmar usando ProvRSA1. ¿por qué?

7) Inserta un par de instrucciones que permitan "estropear" Msg y/o Firma. Inicialmente comenta ambas instrucciones.

```
// Msg[0] = 0xFF;  
// Firma[0] = 0xFF;
```

Verifica con la clave pública

8) Crea la variable booleana Verifica e inicialízala con lo que devuelve el método [VerifyData\(\)](#) del proveedor ProvRSA1. Muestra el contenido de Verifica en la consola. Comprueba que también puedes verificar usando ProvRSA2. ¿por qué?

4. Pruebas variando la importación del certificado en el almacén

PRUEBA 1

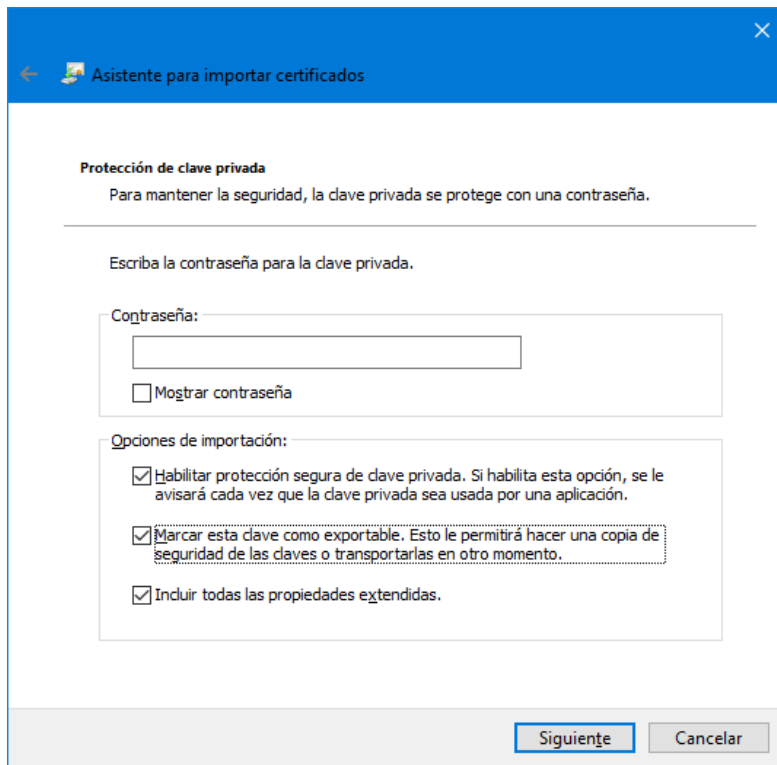
Borra el certificado de zzCiente del almacén y vuelve a importarlo a partir del fichero zzCLI.pfx. En el asistente de importación deja solo marcada la última opción, tal como se muestra a continuación.

En este caso, la clave privada no es exportable.

Vuelve a ejecutar el programa. Comprueba que se genera una excepción en el método VerParam() cuando se intenta exportar la clave privada. Comenta la llamada al método VerParam(). Comprueba que el programa funciona, esto es, que se puede usar la clave privada, pero no exportarla (verla).

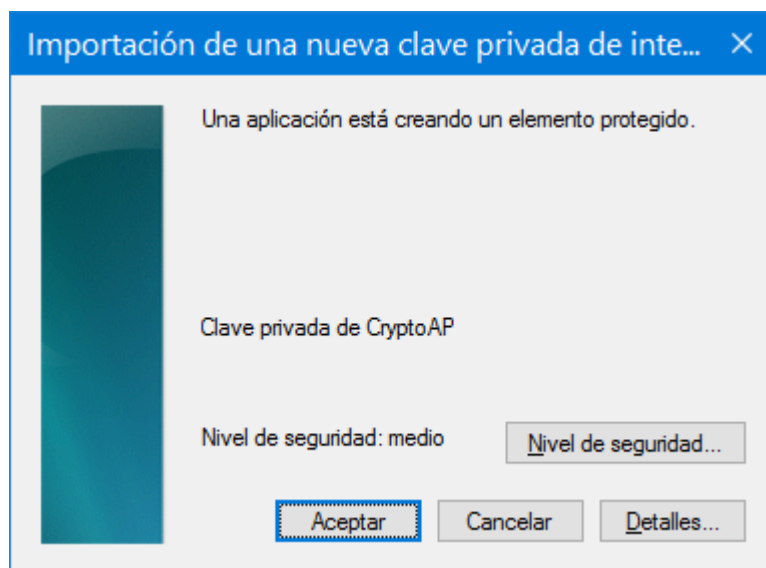
PRUEBA 2

Borra el certificado de zzCiente del almacén y vuelve a importarlo a partir del fichero zzCLI.pfx. En el asistente de importación selecciona las tres opciones, tal como se muestra a continuación.

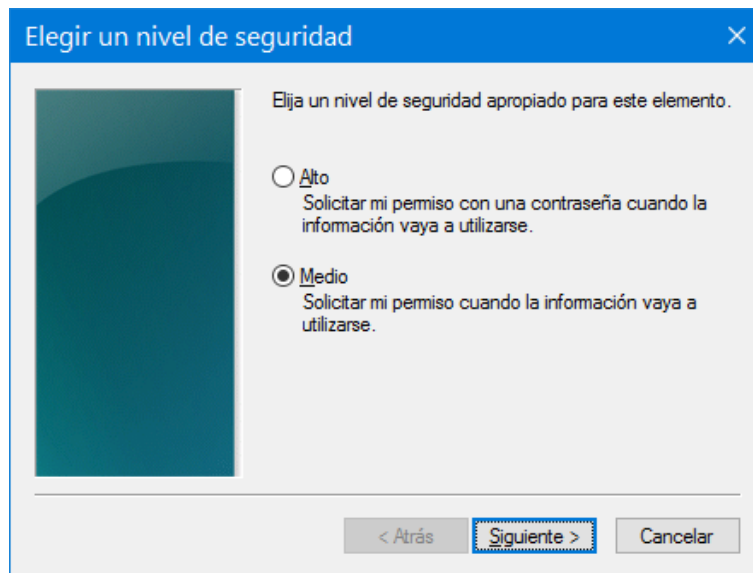


Se está solicitando habilitar la protección (de la utilización) de la clave privada.

Elige el almacén personal y antes de terminar la importación Windows muestra la ventana que permite elegir el nivel de seguridad con el que se desea proteger el uso de la clave privada.



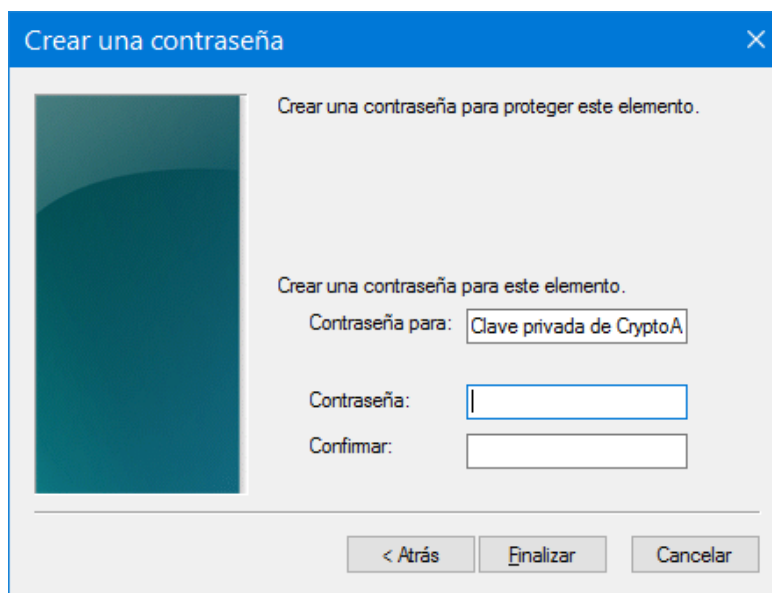
Pulsa el botón "Nivel de seguridad..." para realizar la selección y aparece la ventana que se muestra a continuación:



Si se selecciona Medio, cada vez que se utilice la clave privada se abre una ventana que pide autorización para utilizar la clave privada.

Si se selecciona Alto, cada vez que se utilice la clave privada se abre una ventana que pide una contraseña que hay que introducir para utilizar la clave privada.

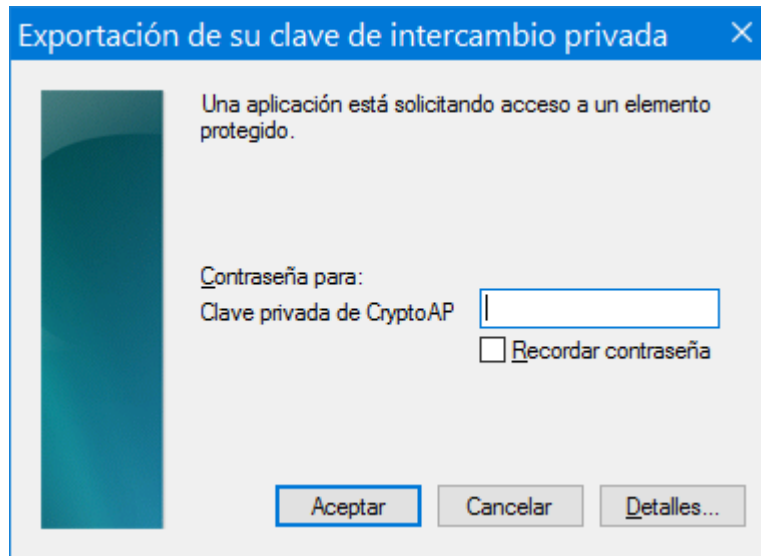
SELECCIONA el nivel Alto. Aparece la ventana siguiente:



Introduce una contraseña y la importación termina correctamente.

Ejecutar nuevamente el programa de la práctica.

Cuando el programa necesita exportar la clave muestra la ventana siguiente:



Se puede comprobar que Windows pide la contraseña en el primer método que la utiliza la clave; en este caso el método `ExportParameters()` del proveedor RSA en el método `VerParams()`.

Tras suministrarle la contraseña, el programa se ejecuta hasta el final sin volver a pedir la contraseña, aunque la clave privada se utiliza en otros sitios del programa.

Comentar la llamada a `VerParams()` y volver a ejecutar el programa. Ahora se puede comprobar que la contraseña se pide al invocar al método `Decrypt()`. Luego no se vuelve a pedir.

PRUEBAS ADICIONALES

Puedes realizar cualquier variación con las opciones de importación del certificado para comprobar el comportamiento del programa en función de la importación.

5. Pruebas con el certificado de la FNMT

Carga tu certificado de la FNMT (usuario.pfx) en el almacén de certificados personales del usuario. No habilites la protección segura de la clave privada y marca la clave como exportable.

El fichero .pfx generalmente contiene una cadena de tres certificados: el del usuario, el certificado intermedio "AC FNMT Usuarios" y el certificado raíz de la Fábrica Nacional de Moneda y Timbre, denominado "AC RAZI FNMT-RCM".

Si en el asistente de importación de certificados se elige "Seleccionar automáticamente el almacén de certificados según el tipo de certificado", el asistente carga el certificado del usuario en el almacén "Personal", el certificado AC FNMT Usuarios en el almacén "Entidades de certificación intermedias" y el certificado AC RAZI FNMT-RCM en el almacén "Entidades de certificación raíz de confianza". Esto es lo correcto.

Si en el asistente de importación de certificados se elige "Colocar todos los certificados en el siguiente almacén" y se elige el almacén "Personal" coloca todos los certificados en el almacén "Personal". Es incorrecto que los certificados de entidades de certificación estén en el almacén "Personal". Hay que exportarlos a ficheros y luego cargarlos en el almacén apropiado. Después se deben eliminar del almacén "Personal".

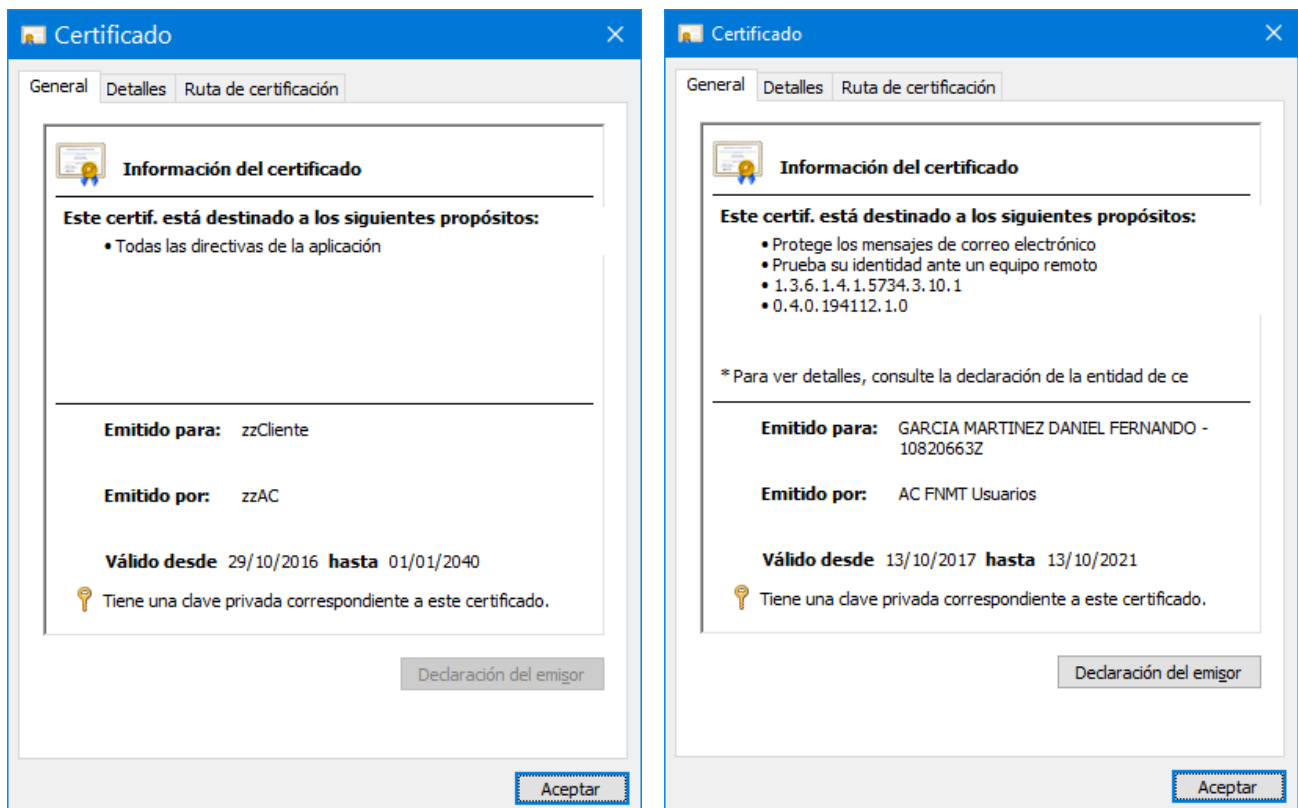
Otra opción es descargar los certificados necesarios de la FNMT de:

<https://www.sede.fnmt.gob.es/descargas/certificados-raiz-de-la-fnmt>

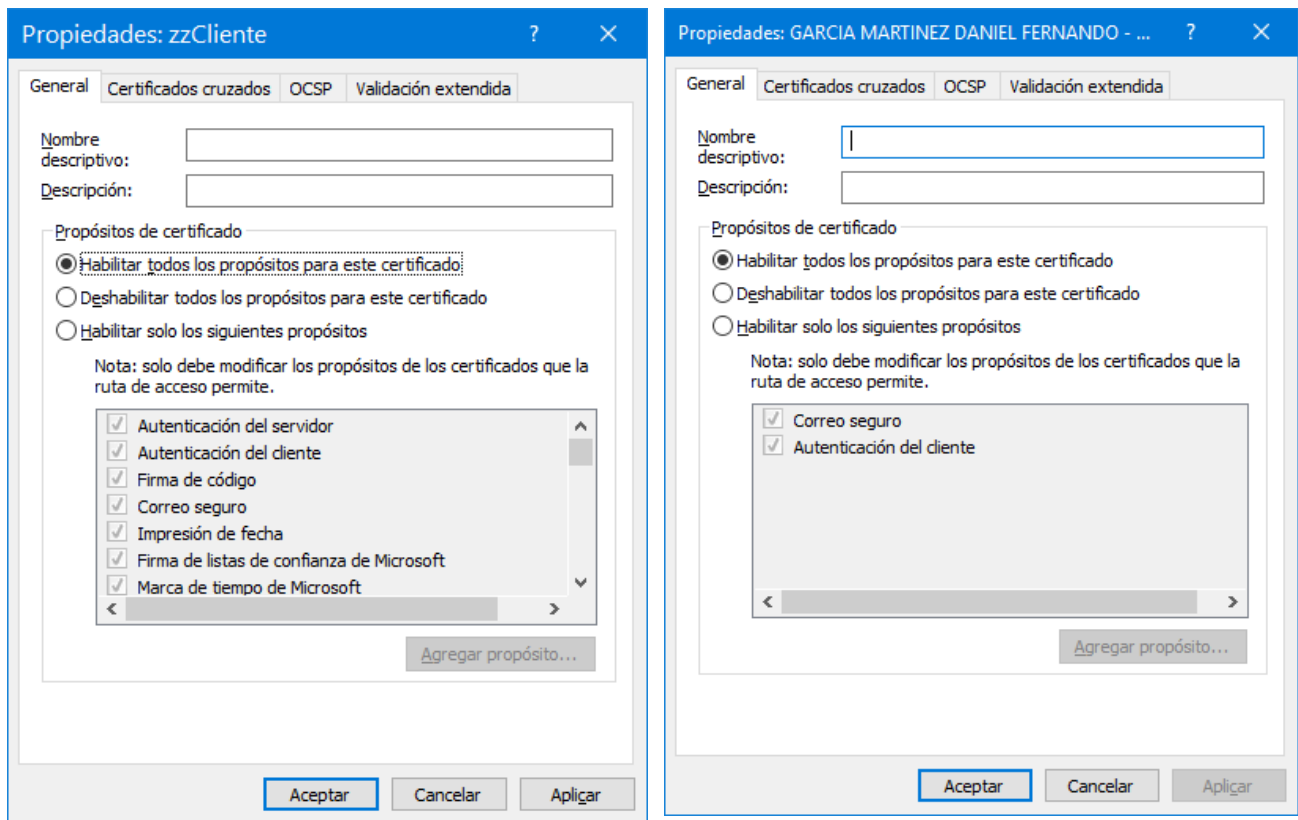
Cargarlos directamente en los almacenes apropiados y luego borrarlos del almacén "Personal".

¿Puedes hacer las mismas operaciones que con el certificado del cliente generado con MakeCert?

Si no puedes, el problema puede deberse a una limitación de los propósitos del certificado. En la ventana de certificados personales de **certmgr.msc** haz clic sobre el certificado del cliente y mira los propósitos. Luego mira los propósitos del certificado de la FNMT.



Después, selecciona el certificado del cliente en la ventana de certmgr y pulsa el botón derecho del ratón. En el menú emergente selecciona Propiedades y observa nuevamente los propósitos del certificado con más detalle y con la posibilidad de habilitar y deshabilitar algunos de los propósitos. Haz lo mismo con el certificado de usuario de la FNMT.



Parece que la creación de un proveedor criptográfico RSA con la siguiente línea de código:

```
RSACryptoServiceProvider ProvRSA2 = (RSACryptoServiceProvider)Cert.PrivateKey;
```

Puede limitar la utilización de las claves del proveedor RSA.

Una forma posible de eliminar las limitaciones consiste en exportar las claves del certificado a una cadena y luego usar la cadena para inicializar un proveedor RSA. Para un certificado de la FNMT que utiliza una clave RSA de 2048 y proveedor RSA de la clase RSACryptoServiceProvider, el código puede ser el siguiente:

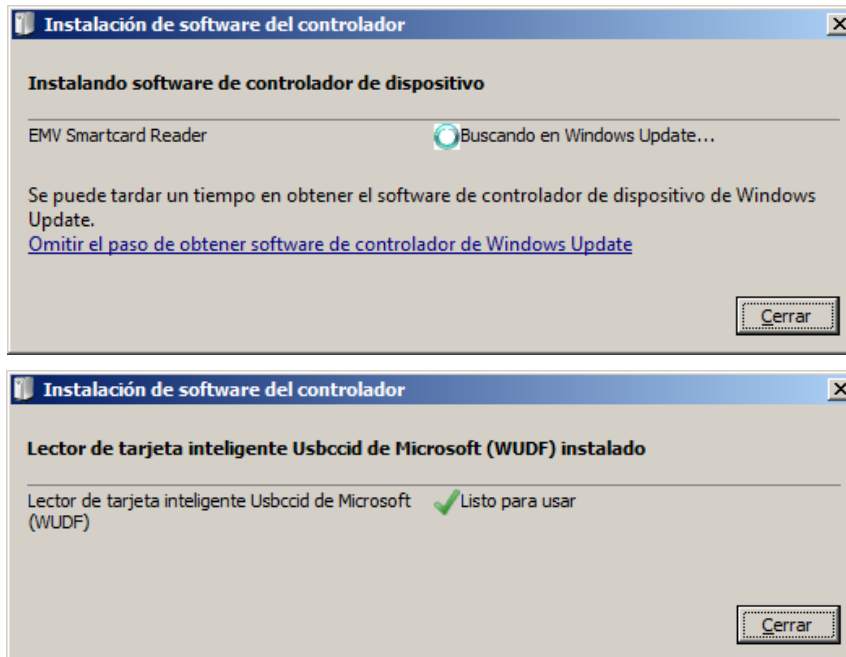
```
string ClavePriXML = Cert.PrivateKey.ToXmlString(true);
Console.WriteLine("\n\nClave Privada en XML: " + ClavePriXML);
RSACryptoServiceProvider ProvRSA2 = new RSACryptoServiceProvider(2048);
ProvRSA2.FromXmlString(ClavePriXML);
```

Para que este código funcione al importar el certificado en el almacén de certificados desde el fichero .pfx hay que marcar la clave privada como EXPORTABLE.

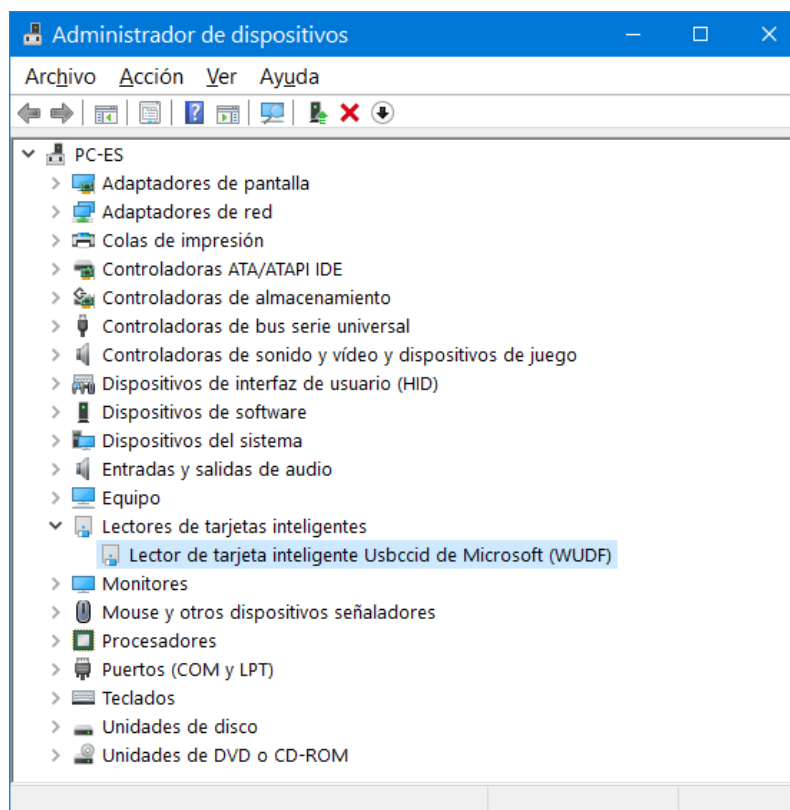
Para realizar estas pruebas, no generes un nuevo proyecto. Introduce en el código las dos formas de crear el proveedor RSA y comenta una u otra según la prueba a realizar.

6. Pruebas con el DNle

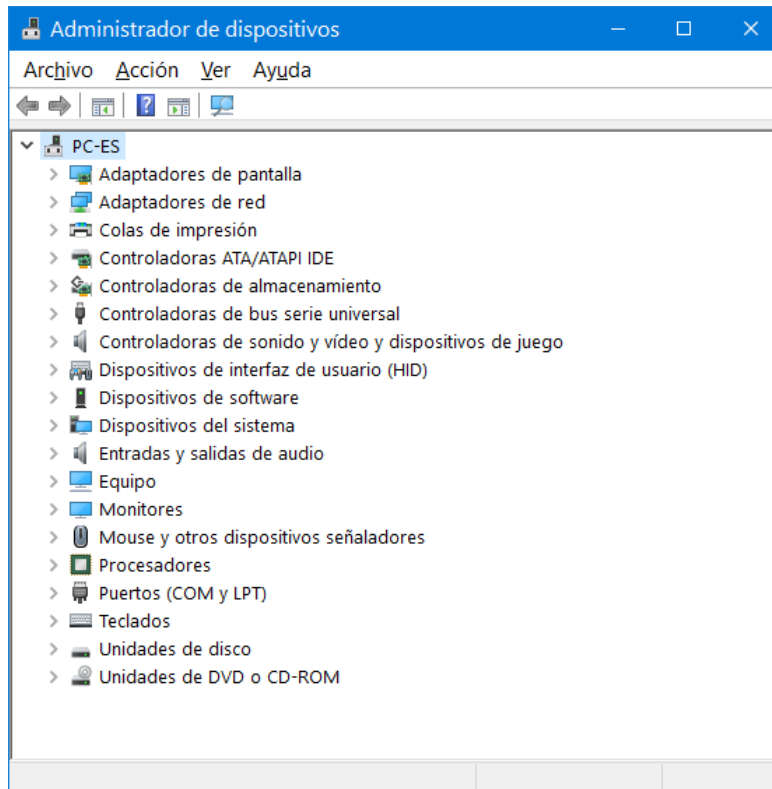
Inserta el lector de tarjetas inteligentes Woxter en un puerto USB. Al hacerlo por primera vez, en Windows 7, aparecen la ventanas siguientes:



Estas ventanas de instalación no aparecen si los drivers del lector ya están instalados en el sistema operativo. En Windows 10 están instalados por defecto. Abre el Administrador de dispositivos y observa que aparece el tipo de dispositivos “Lectores de tarjetas inteligentes” y en ese tipo “Lector de tarjeta inteligente Usbccid de Microsoft (WUDF)”.

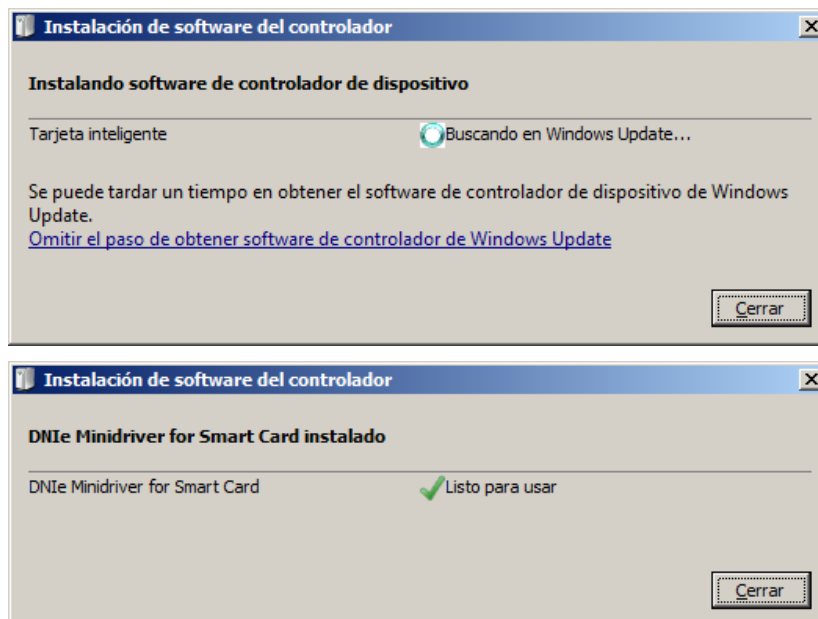


Si se extrae el lector del puerto USB desaparece el tipo Lectores de tarjetas inteligentes:



Posteriormente, al volver a insertar el lector en el puerto USB ya no reinstalan drivers. Se puede saber que el lector está activo porque en los iconos que hay en el extremo derecho de la barra de tareas, al seleccionar el icono “Expulsar hardware de forma segura y expulsar el medio” aparece “Expulsar EMV Smartcard Reader”.

Inserta el DNle en el lector de tarjetas. Es necesario instalar un driver para que el chip integrado en el DNle se comuniquen con el lector. La instalación es automática.

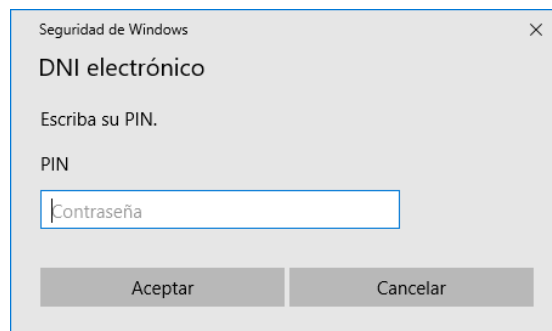


Estas ventanas de instalación no aparecen si los drivers del DNle ya están instalados en el sistema. Por defecto, aparecen en Windows 7, pero no en Windows 10.

En la esquina inferior derecha de la pantalla aparece la siguiente ventana:



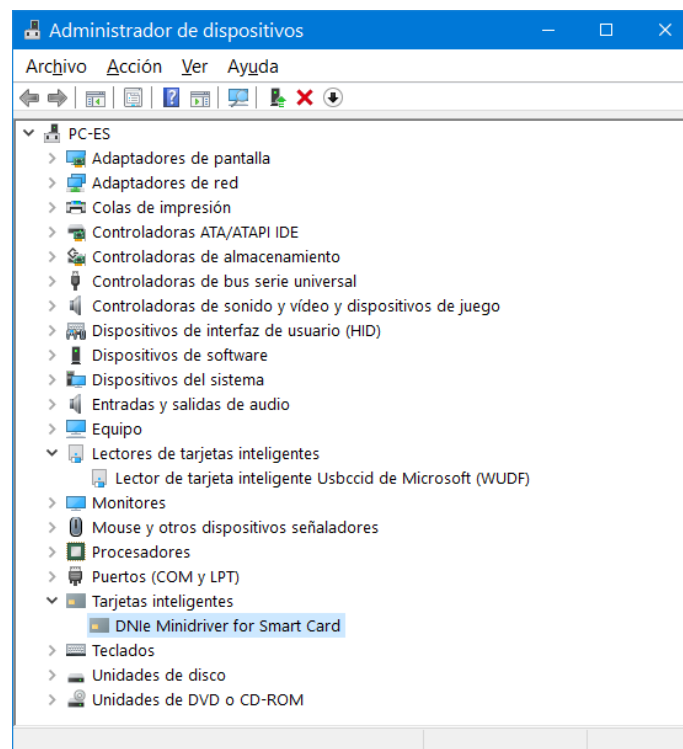
En ese momento el sistema operativo muestra la ventana siguiente:



Tras suministrar el PIN correcto, el DNle es utilizable. Las dos ventanas anteriores solo aparecen la primera vez que se utiliza un DNle en un computador.

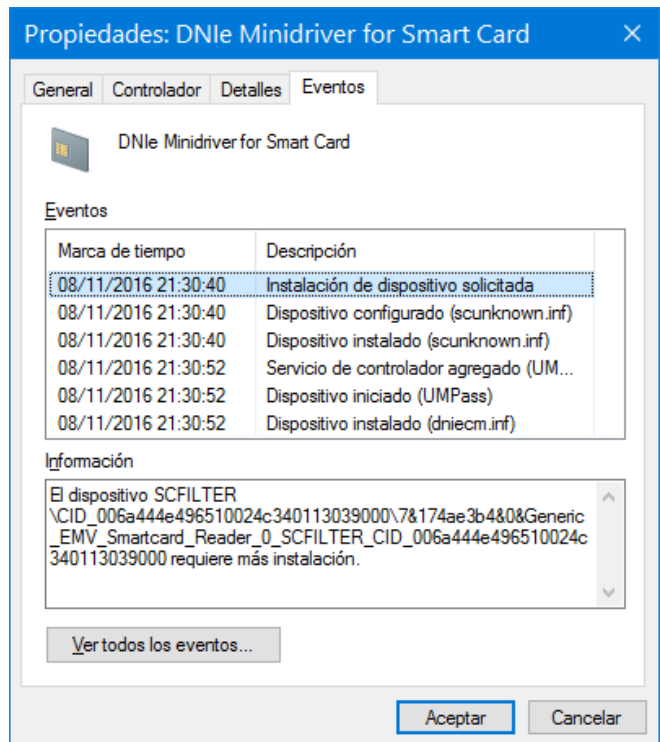
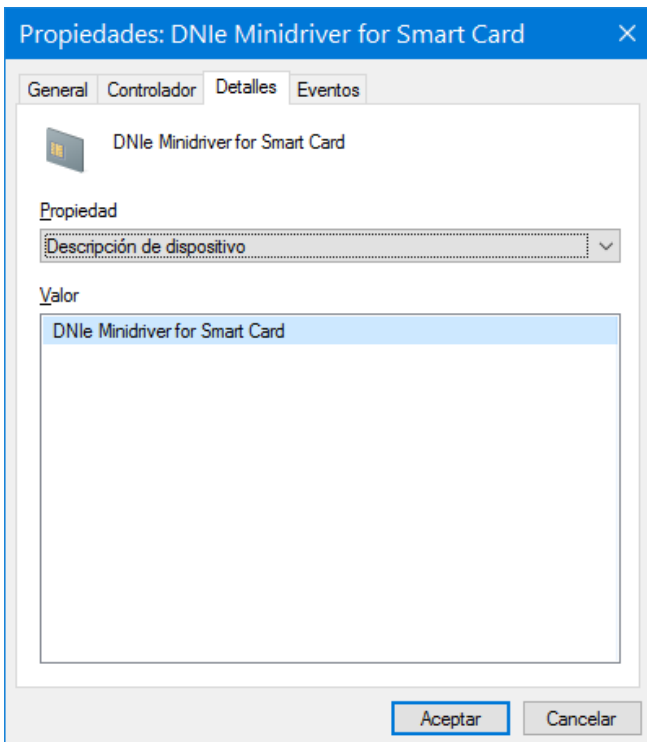
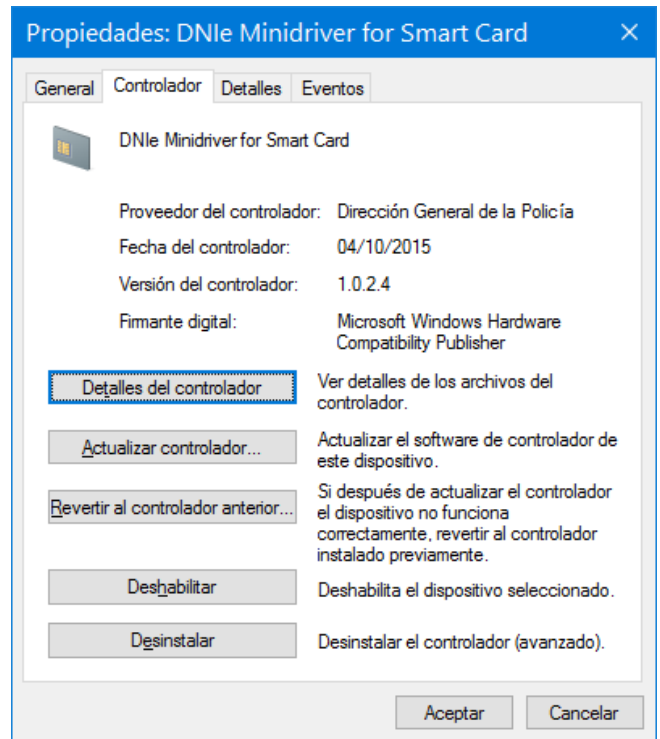
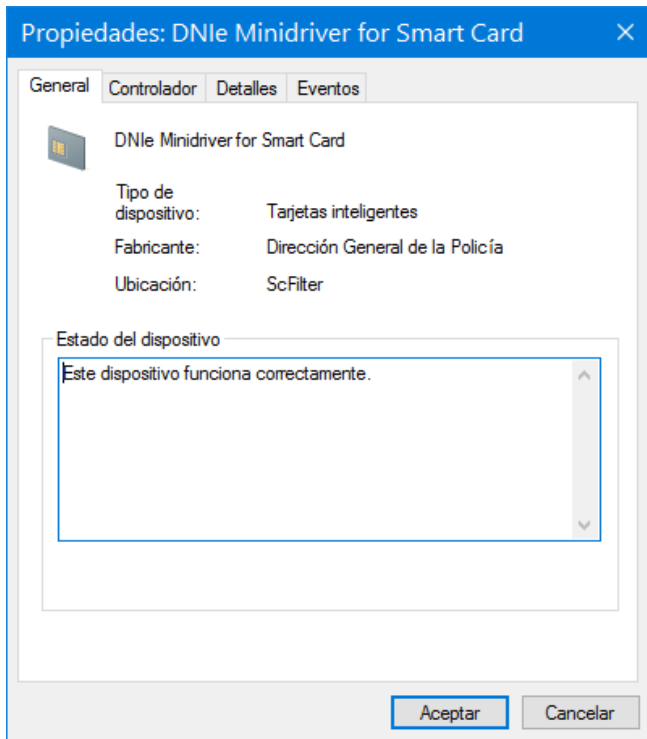
En las últimas versiones de Windows 10 no se pide el PIN al insertar el DNle, sino solo al usarlo.

Comprueba que en el administrador de dispositivos aparece un nuevo tipo de dispositivos: “Tarjetas inteligentes”, y que dentro de este tipo hay una tarjeta concreta: “DNle Minidriver for Smart Card”.

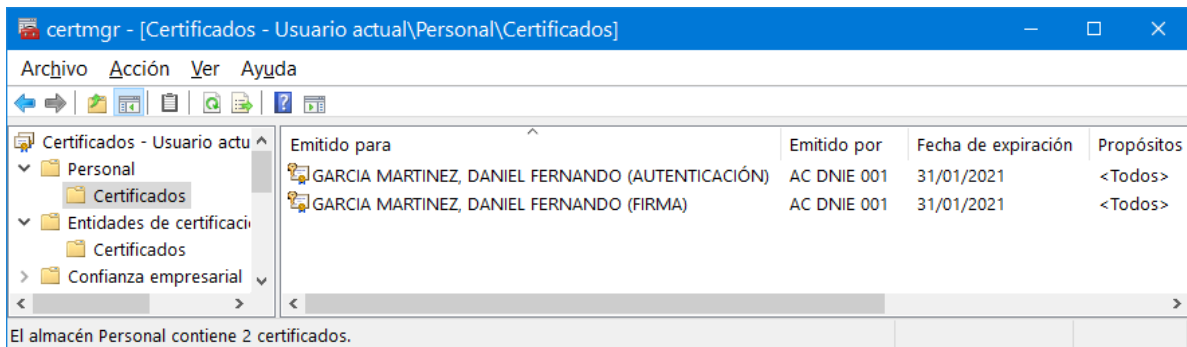


El DNle Minidriver, al detectar un DNle por primera vez en un equipo, solicitará el PIN para leer la parte pública de los certificados de ciudadano y mantener una cache cifrada en el perfil del usuario. Las siguientes veces que se inserta el DNle se lee la parte pública de los certificados de dicha caché cifrada en lugar de la tarjeta para evitar la solicitud de PIN. Posteriormente solicitará el PIN únicamente cuando se realicen operaciones con el DNle.

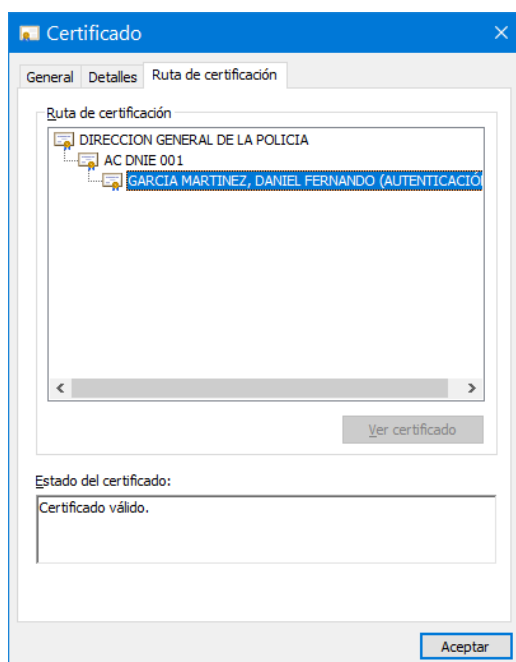
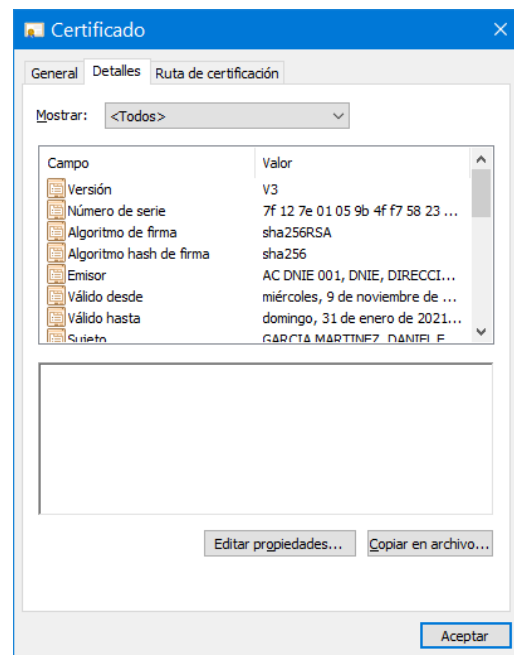
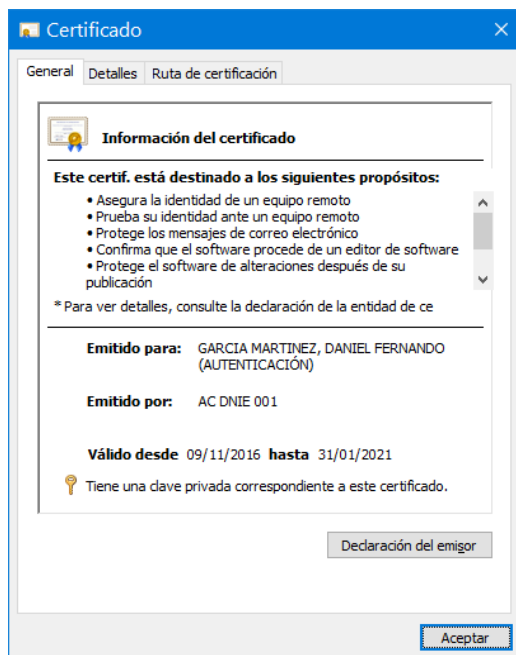
Haz doble clic en “DNle Minidriver for Smart Card” para ver los detalles del controlador, tal como se muestra en las cuatro figuras siguientes:



Usando la aplicación certmgr.msc comprueba que en el almacén de certificados Personal aparecen dos nuevos certificados automáticamente, uno para AUTENTICACIÓN y otro para FIRMA, tal como se muestra en la figura siguiente:



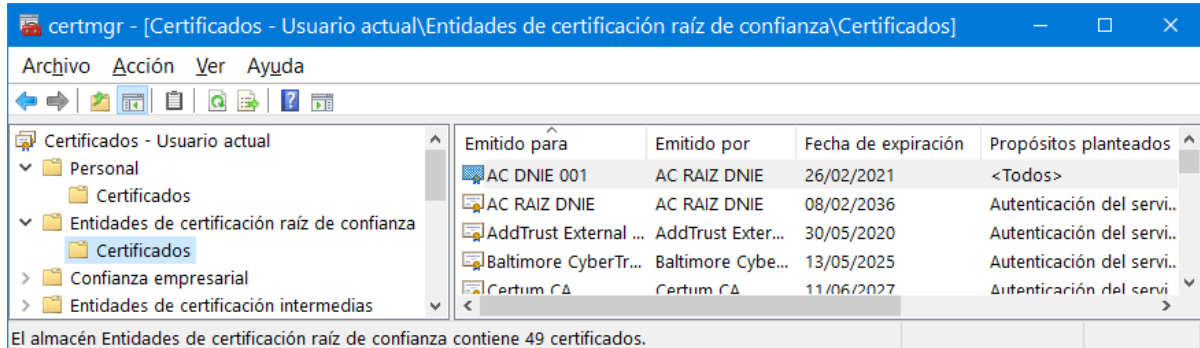
Haz doble clic sobre un certificado para ver la información sobre el propio certificado.



En la ventana detalles observa el campo clave pública. Comprueba que cada certificado del DNIE incluye una clave pública RSA de 2048 bits. Lógicamente las dos claves públicas son diferentes. Aunque los primeros 9 bytes y los últimos 5 bytes son diferentes.

Análisis de la cadena de 3 certificados del DNIE

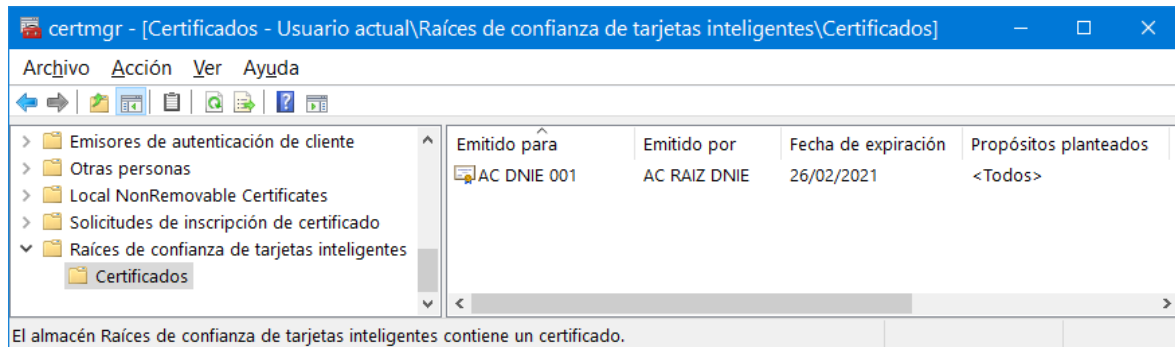
Haz doble clic sobre el certificado de autenticación del DNIE para verlo. Selecciona la pestaña “Ruta de certificación” y observa que el certificado ha sido emitido por la autoridad certificadora intermedia “AC DNIE 001”. Busca este certificado en el almacén “Entidades de certificación raíz de confianza” y comprueba que efectivamente está en ese almacén.



Aunque parecería lógico que el certificado AC DNIE 001 estuviese en el almacén “Entidades de certificación intermedias”, no es así. Comprueba que no está en ese almacén.

El certificado intermedio de “AC DNIE 001” ha sido emitido por “AC RAZ DNIE”, que en la ruta de certificación tiene el nombre “DIRECCION GENERAL DE LA POLICIA”. Comprueba que este certificado raíz también se ha cargado automáticamente en el almacén “Entidades de certificación raíz de confianza”.

Comprueba que el certificado intermedio “AC DNIE 001” también se ha cargado en el almacén “Raíces de confianza de tarjetas inteligentes”, tal como muestra la figura siguiente:



Extrae el DNIE del lector de tarjetas. Comprueba que los dos certificados del usuario desaparecen del almacén de certificados “Personal” pero que el certificado AC DNIE 001 permanece en el almacén de certificados “Raíces de confianza de tarjetas inteligentes” y que los certificados AC RAZ DNIE y AC DNIE 001 también permanecen en el almacén de certificados “Entidades de certificación raíz de confianza”.

Si una segunda persona (con su DNIE diferente del utilizando anteriormente) utiliza el mismo computador en la misma sesión de usuario e inserta su DNIE en el lector, se genera un nuevo certificado intermedio “AC DNIE 002” en los almacenes de certificados “Entidades de certificación raíz de confianza” y “Raíces de confianza de tarjetas inteligentes”.

Los certificados AC DNIE nnn los gestiona el sistema operativo y el usuario no los puede eliminar. Pero se eliminan automáticamente al cerrar la sesión o al apagar el computador.

Utilización del programa con los certificados del DNle

En general el comportamiento del programa será similar al observado con un certificado creado con Makecert o con el certificado de la FNMT. No obstante hay algunas limitaciones con los dos certificados del DNle. Independientemente de que se utilice el certificado de FIRMA o de AUTENTICACIÓN del DNle, tener en cuenta las siguientes consideraciones:

a) El uso de la sentencia siguiente crea un proveedor RSA con dos claves: pública y privada, a partir de uno de los certificados del DNle: FIRMA ó AUTENTICACIÓN.

```
RSACryptoServiceProvider ProvRSA2 = (RSACryptoServiceProvider)Cert.PrivateKey;
```

b) Pero el uso de la sentencia siguiente para extraer la clave privada genera una excepción.

```
string ClavePriXML = Cert.PrivateKey.ToXmlString(true);
```

Por tanto no se puede crear un proveedor RSA a partir de una clave privada almacenada en formato XML extraída de un certificado del DNle.

c) Cuando un proveedor RSA se ha creado a partir de un certificado del DNle usando Cert.PrivateKey, esto es, cargándoles las claves pública y privada, no permite la exportación de los parámetros de la clave privada. El uso de la sentencia siguiente genera una excepción.

```
ParamRSA = ProvRSA2.ExportParameters(true);
```

Utilizando el certificado de FIRMA del DNle:

Comprueba que NO puedes cifrar con la clave pública ejecutando el método Encrypt() de ProvRSA2, creado cargando la clave pública y la privada del certificado de FIRMA del DNle, porque se genera una excepción: clave incorrecta.

Comprueba que SI puedes cifrar con la clave pública ejecutando el método Encrypt() de ProvRSA1, creado cargando solo la clave pública del certificado de FIRMA del DNle.

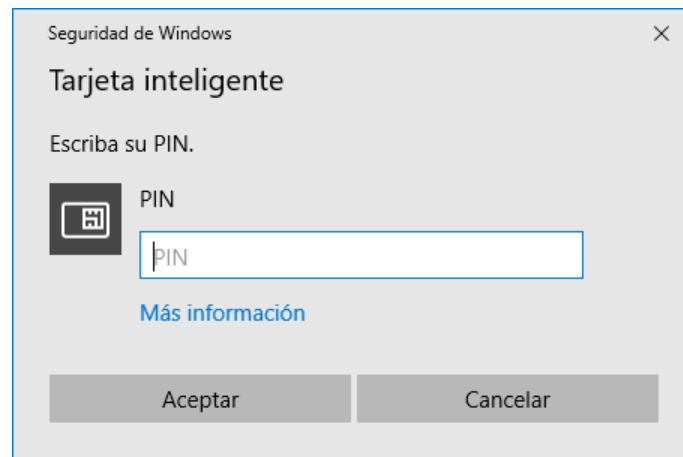
Pero NO puedes descifrar con la clave privada ejecutando el método Decrypt() de ProvRSA2, creado cargando la clave pública y la privada del certificado de FIRMA del DNle, porque se genera una excepción: clave incorrecta.

Este comportamiento se debe a limitaciones en los usos posibles del certificado.

Ahora analiza el proceso de Firma y su Verificación. Para no utilizar la sección del código de “Cifrado con clave pública y Descifrado con clave privada” insertálo en un bloque de comentario: /* ... código ... */

Comprueba que SI se puede firmar con la clave privada ejecutando el método SignData() de ProvRSA2 y verificar la firma con la clave pública ejecutando el método VerifyData() de ProvRSA1. Comprueba que puedes usar varios algoritmos de resumen, por ejemplo SHA1 y SHA256.

Al intentar acceder a la clave privada el sistema operativo muestra una ventana solicitando la contraseña que protege la utilización de la clave privada, tal como se muestra a continuación:



Tras introducir la contraseña correcta, el sistema operativo (dependiendo de la versión) puede mostrar una ventana en la esquina inferior derecha de la pantalla solicitando el permiso para realizar la operación de firma, como se muestra a continuación:



Utilizando el certificado de AUTENTICACIÓN del DNle:

Este certificado permite cifrar con la clave pública de ProvRSA1 y descifrar con la clave privada de ProvRSA2. También permite firmar con la clave privada de ProvRSA2 y verificar con la clave pública de ProvRSA1. El sistema operativo pide la contraseña al descifrar y al firmar con la clave privada de ProvRSA2.