Acceso a Almacenes de Certificados
Práctica 5B

1. Objetivo

En esta práctica el alumno debe realizar pequeños programas que permitan acceder a los almacenes de certificados del sistema operativo Windows, con el objetivo de buscar certificados, extraerlos, eliminarlos, cargar nuevos certificados, etc.

2. Acceso al almacén de certificados de las Autoridades de Certificación: buscar certificados

Realiza un programa que permita acceder al almacén de certificados en el que Windows almacena los certificados de las Autoridades de Certificación. Para evitar borrados accidentales en un almacén importante como éste, se accederá en modo de solo lectura.

Para trabajar con certificados, y almacenes de certificados, es necesario incluir en el programa el espacio de nombres **System.Security.Cryptography.X509Certificates**.

Debajo del método Main(), escribe el método ExtraeCertificado() que devuelve un objeto de la clase **X509Certificate2** y usa como parámetro de entrada un **string** con el **Nombre** del sujeto del certificado que se desea extraer.

El método a realizar tiene dos fases principales:

FASE.-1 ABRIR EL ALMACEN Y EXTRAER SUS CERTIFICADOS

1) Crea un objeto **Almacen** de la clase **X509Store**. En el constructor utiliza el valor de la enumeración **StoreName** que corresponde al almacén de las "Entidades de certificación raíz de confianza" y para la enumeración **StoreLocation** el valor que corresponde al "Usuario actual". Buscar los valores apropiados de las enumeraciones en la ayuda. Toma notas en el programa sobre las equivalencias entre los valores de la enumeración StoreName y los almacenes de certificados a los que corresponde.

2) Como comprobación, muestra las propiedades **Name** y **Location** de **Almacen** en la consola.

3) Abre **Almacen** con el método **Open()**. Usa dos flags, uno para abrirlo en modo de solo lectura y otro para que solo se pueda abrir almacenes ya existentes. Debes combinar los flags con el operador OR bit a bit: |.

4) Coloca a todos los certificados del almacén en un objeto de clase **X509Certificate2Collection**. Para ello, declara una variable **ColeCert** de la clase **X509Certificate2Collection** y asígnale, en la misma declaración, la propiedad **Certificates** del objeto **Almacen**.

5) Cierra **Almacen** llamando al método **Close()**.

6) Muestra los certificados que contiene la colección **ColeCert** con un bucle **foreach**. Para ello muestra para cada certificado solo el nombre del sujeto, usando la propiedad **Subject** del certificado. Este bucle se puede comentar una vez comprobado el buen funcionamiento del método, pero no lo borres, déjalo para depurar posibles fallos en el futuro.

7) Como resumen, muestra en la consola el número total de certificados que contiene el almacén, o lo que es lo mismo, el número de elementos de **ColeCert**, usando su propiedad **Count**.

FASE.-2 BUSCAR UN CERTIFICADO

1) Buscar los certificados que cumplan una determinada condición. Para ello se utiliza el método **Find()** del objeto **ColeCert** de la clase X509Certificate2Collection. Este método devuelve otra colección de la clase **X509Certificate2Collection** que se debe declarar, por ejemplo, en la misma sentencia en la que se invoca el método Find(), y que se puede denominar **CertsEncontrados**.

Como primer parámetro, pasar a Find() un valor de la enumeración **X509FindType** que permita encontrar todos los certificados de la colección, que a la fecha y hora actuales, tales que su periodo de validez ya haya comenzado y todavía no haya expirado. Mostrar en la consola el número de certificados encontrados que están dentro de su periodo de validez.

Repetir la búsqueda para localizar todos los certificados de la colección cuyo período de validez ya haya expirado. Mostrar en la consola el número de certificados encontrados que han expirado.

2) Buscar los certificados válidos de una entidad usando su nombre distintivo. Utilizar el parámetro **Nombre** de tipo string al que se le pasa la cadena del campo Subject de un certificado. Aprovecha que ya se han mostrado estas cadenas en la fase previa de la práctica.

Para encontrar **todos los certificados de una entidad**, en el primer parámetro de Find() usar X509FindType.FindBySubjectName y en el segundo parámetro utilizar un string que contenga una palabra distintiva de la entidad, como por ejemplo "Microsoft".

Para encontrar **un certificado concreto de una entidad**, en el primer parámetro de Find() usar X509FindType.FindBySubjectDistinguishedName y en el segundo parámetro utilizar un string que contenga los caracteres exactos del campo Subject del certificado, como por ejemplo:

"CN=Microsoft Root Certificate Authority 2011, O=Microsoft Corporation, L=Redmond, S=Washington, C=US"

Si se añade un blanco extra tras una de las comas o al final, el método Find() NO encuentra el certificado. Pero si se cambian caracteres de mayúsculas a minúsculas o viceversa, el método Find() SI encuentra el certificado. Comprueba esto.

El método debe controlar los errores que pueden ocurrir al buscar un certificado. Por ejemplo…

Inicializar la variable **CertBuscado** con el valor null.

SI el número de certificados de la colección **CertsEncontrados** es igual a 1

Asignar a CertBuscado el único elemento de la colección, que es CertEncontrados[0].

SI el número de certificados de la colección **CertsEncontrados** es igual a 0

Escribir en la consola que no se han encontrado certificados

Se puede salir del programa usando Environment.Exit(1) o dejar a Main() el control de errores.

SI el número de certificados de la colección **CertsEncontrados** es mayor que 1

Escribir en la consola que se ha encontrado más de un certificado

Se puede salir del programa usando Environment.Exit(1) o dejar a Main() el control de errores.

En el método Main() Hay que declarar un string con el nombre del sujeto del certificado que se desea extraer del almacén de certificados. Por ejemplo:

string **NombreSujetoCer** = "CN=zpUSU.as";

Cada prueba con un certificado requiere una sentencia como la anterior. No borres las sentencias que no se usen, sino que debes comentarlas con // para poder repetir pruebas con diferentes certificados rápidamente.

Y después se llama al método de extracción del certificado.

X509Certificate2 **Certificado** = ExtraeCertificado(**NombreSujetoCer**);

Si se opta por realizar el control de errores en Main() se deberá comprobar si **Certificado** == null.

3. Acceso al almacén de certificados del usuario: añadir y eliminar certificados

Realiza un nuevo programa (**nueva solución de Visual Studio**) que permita acceder al almacén de certificados en el que Windows almacena los certificados del usuario.

1) Vacía manualmente el almacén de certificados personales con la herramienta certmgr.

2) Crea una clase con el método VerCerts(X509Store AL) o créalo como método estático junto con el método Main(). Debe mostrar en la consola el nombre del almacén, y debajo, el nombre del sujeto de cada uno de los certificados que contiene el almacén, cada nombre en una línea de la consola. Si el almacén no contiene certificados debe indicarlo.

3) Crea un objeto **Almacen** de la clase **X509Store** para trabajar con el almacén denominado **My** y ubicado en **CurrentUser**.

4) Abre **Almacen** con el método **Open()** para lectura y escritura. Requiere también que solo se abra un almacén ya existente.

5) Crea tres objetos, **Cert1**, **Cert2** y **Cert3** de la clase **X509Certificate2** y pasa en los constructores los ficheros zpACas.cer, zpSERas.cer y zpUSUas.cer. Coloca los ficheros .cer en el mismo directorio en el que está el ejecutable.

6) Añade el certificado **Cert1** a **Almacen** usando el método **Add()**. Muestra el contenido de Almacen en consola llamando al método VerCerts().

7) Crea un objeto **ColeCert**, una colección de certificados vacía, a partir de la clase **X509Certificate2Collection**. Usa el método **Add()** para añadir los certificados **Cert2** y **Cert3**.

8) Añade un rango de certificados a **Almacen** usando el método **AddRange()**. El rango se define mediante la colección ColeCert. Muestra el contenido de Almacen en consola llamando al método VerCerts().

9) Elimina el certificado **Cert1** de **Almacen** usando el método **Remove()**. Muestra el contenido de Almacen en consola llamando al método VerCerts().

10) Elimina un rango de certificados de **Almacen** usando el método **RemoveRange()**. El rango se define mediante la colección ColeCert. Muestra el contenido de Almacen en consola llamando al método VerCerts().

11) Cierra **Almacen** usando el método **Close()**.