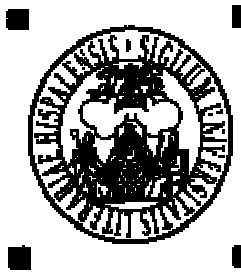

UNIVERSIDAD
de SEVILLA

Proyecto Fin de Carrera

Estudio e implementación de una Autoridad de Certificación

(planos)

Autor: Gabriel Babiano Huete
Tutor: Teresa Ariza Gómez

Índice de contenido

Planos

Mastin v1.0 y XMastin v1.0

Mastin.java

```
package es.us.esi.bab.mastin;


import com.sun.net.ssl.internal.ssl.JSA_RSAPrivateKey;     //Clave privada almacenada en el KeyStore (incluir el paquete jsse.jar en el
CLASSPATH)
import com.sun.net.ssl.internal.ssl.JSA_RSAPublicKey;      //Clave pública almacenada en el KeyStore (incluir el paquete jsse.jar en el
CLASSPATH)

import es.us.esi.bab.mastin.MastinException;   //La clase que me permite lanzar mis excepciones

import java.io.ByteArrayOutputStream;
import java.io.CharArrayWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.PrintStream;

import java.lang.reflect.Array;

import java.math.BigInteger;

import java.security.Key;
import java.security.KeyFactory;
import java.security.KeyStore;
```

```
import java.security.spec.RSAPublicKeySpec;
import java.security.spec.RSAPrivateCrtKeySpec;

import java.util.GregorianCalendar;            //Para la toma de tiempos

import javax.crypto.Cipher; //Cifrador del JCE de Sun Microsystems


/**
 * Clase que contiene tanto la aplicación Mastin v1.0 como los métodos que permiten las funciones de la aplicación a otras aplicaciones
 *
 * @author Gabriel Babiano Huete (<a hfer="mailto:gabrielbabiano@yahoo.es">gabrielbabiano@yahoo.es</a>)
 * @version 1.0
 * @since Mastin v1.0
 */
public class Mastin{


            //Constantes que contienen el nombre de la aplicación, la versión y el "acerca de"

            /**
             * Nombre de la aplicación:
             * @since Mastin v1.0
             */
            public static final String NOMBRE="Mastin";

            /**
             * Versión de la aplicación
             * @since Mastin v1.0
             */
            public static final String VERSION="1.0";

            /**
             * "Acerca de" la aplicación
             * @since Mastin v1.0
             */
            public static final String ABOUT=NOMBRE+" v"+VERSION+"\nAplicación creada por Gabriel Babiano Huete
(gabrielbabiano@yahoo.es)";        //Acerca de la aplicación
```

```
            //Constantes que contienen los nombres de los algoritmos tanto en la version corta (ALGORITMO_MASTIN), como en la version
algoritmo/modo/relleno (ALGORITMO_CIPHER)

            /**
             * Algoritmo que utiliza la aplicación (en versión corta)
             * @since Mastin v1.0
             */
            public static final String ALGORITMO_MASTIN="RSA";

            /**
             * Algoritmo que utiliza el Cipher de la aplicación (en versión larga)
             * @since Mastin v1.0
             */
            public static final String ALGORITMO_CIPHER="RSA/ECB/OAEPPADDING";      //Es el ALGORITMO_MASTIN pero en el formato
algoritmo/modo/relleno


            //Constantes que contienen valores por defecto

            /**
             * Alias que se utiliza por defecto
             * @since Mastin v1.0
             */
            public static final String ALIAS_DEFECTO="mykey";

            /**
             * Tipo de KeyStore que se utiliza por defecto
             * @since Mastin v1.0
             */
            public static final String TIPO_KEYSTORE_DEFECTO="JKS";


            //Variables que se utilizan para la toma de tiempos. Están definidas aquí para que sean accesibles a todos los métodos de la
clase

            /**
             * Tiempo del inicio de la encriptación / desencriptación
             * @since Mastin v1.0
```

```
             */
            private static GregorianCalendar tiempoInicio;
            /**
             * Tiempo del finalización de la encriptación / desencriptación
             * @since Mastin v1.0
             */
            private static GregorianCalendar tiempoFin;


            /**
             * <br><h2>Mastin v1.0</h2>
             * <br>Aplicación creada por Gabriel Babiano Huete (<a hfer="mailto:gabrielbabiano@yahoo.es">gabrielbabiano@yahoo.es</a>)
             * <br>
             * <br><b><i>DESCRIPCIÓN:</i></b>
             * <br>Mastin v1.0 es una aplicación de línea de comandos que encripta o desencripta ficheros a partir de la clave extraida de
un KeyStore creado por la herramienta <b><i>keytool</b></i> de Java
             * <br>
             * <br>El algoritmo que se usa es el algoritmo asimétrico RSA inventado por Rivest, Adleman y Shamir y que viene recogido en
el PKCS#1.
             * <br>El modo de funcionamiento será ECB (Electronic Code Book).
             * <br>El relleno será OAEP (Optimal Asymmetric Encryption Padding) definido en el PKCS#1 v2.
             * <br>
             * <br>La clave, ya sea pública o privada, se extraerá de la entrada en un KeyStore definida por un alias
             * <br>
             * <br>Se podrá especificar la ruta del KeyStore (<i>-KeyStorePath keyStorePath</i>), su tipo (<i>-KeyStoreType
keyStoreType</i>), y su clave (<i>-KeyStorePassword keyStorePassword</i>).
             * <br>
             * <br>Si se desea utilizar un alias distinto a "mykey", se deberá especificar (<i>-Alias alias</i>) así como su clave (<i>-
KeyPassword keyPassword</i>) en caso a que sea distinta a la del KeyStore (<i>-KeyStoreType keyStoreType</i>)
             * <br>
             * <br><b><i>SOFTWARE NECESARIO:</i></b>
             * <br>        <ul><li>Java 2 SDK 1.4.0 (aunque podría bastar con el JRE de incluso versiones posteriores)
             * <br>        <li>Bouncy Castle Provider 1.15</ul>
             * <br>
             * <br><b><i>USO:</i></b><i>    Mastin [opciones] -e/-d -in inFile -out outFile</i>
             * <br>                <ul><li><i>-e</i>:     encripta
             * <br>                <li><i>-d</i>:     desencripta
             * <br>                <li><i>-in inFile</i>:     inFile es la ruta del fichero de entrada
             * <br>                <li><i>-out outFile</i>:   outFile es la ruta del fichero de salida</ul>
```

```
             * <br>
             * <br>        Lista de opciones:
             * <br>                <ul><li><i>-about</i>:         muestra información acerca de la aplicación
             * <br>                <li><i>-Alias alias</i>:   alias del la entrada en la KeyStore de donde se extraerá la clave privada.
             * <br>                                Por defecto se usará "mykey"
             * <br>                <li><i>-KeyPassword keyPassword</i>:       especifica la contraseña de la clave del alias.
             * <br>                                Por defecto se usará la misma clave que para el KeyStore ("-KeyStorePassword
KeyStorePassword")
             * <br>                <li><i>-KeyStorePassword KeyStorePassword</i>:     especifica la contraseña de entrada al KeyStore
             * <br>                <li><i>-KeyStorePath KeyStorePath</i>:     especifica la ruta del fichero KeyStore
             * <br>                <li><i>-KeyStoreType KeyStoreType</i>:     especifica el tipo de KeyStore.
             * <br>                                Por defecto se usará "JKS"
             * <br>                <li><i>-v</i>:             muestra información del desarrollo de la aplicación(equivalente a -verbose)
             * <br>                <li><i>-verbose</i>:       muestra información del desarrollo de la aplicación
             * <br>                <li><i>-version</i>:       muestra información de la versión de la aplicación</ul>
             *
             * @since Mastin v1.0
             */
            public static void main(String args[]){

                        try{
                                    //Variables del KeyStore y el alias
                                    String pathKeyStore=new String();   //Ruta del KeyStore
                                    String tipoKeyStore=new String();   //Tipo del KeyStore
                                    CharArrayWriter passwordKeyStore=new CharArrayWriter();     //Clave del KeyStore
                                    String alias=new String();          //Alias de la entrada en le KeyStore
                                    CharArrayWriter keyPassword=new CharArrayWriter();          //Clave del alias


                                    //Variables relacionadas con las rutas de los ficheros de entrada y salida
                                    String pathFileIn=new String();                 //Ruta del fichero de entrada al programa. Puede ser el
fichero a encriptar en caso de que usemos la opción -e o a desencriptar en el caso en que usemos la opción -d
                                    String pathFileOut=new String();    //Ruta del fichero que será la salida del programa

                                    int opMode=0;           //Modo de operación del Cipher. Puede contener los valores
javax.crypto.Cipher.ENCRYPT_MODE en el caso de que se encripte o javax.crypto.Cipher.DECRYPT_MODE para el caso en que se desencripte

                                    boolean verbose=false;  //Variable boolean que me dice si debo imprimir los mensajes generados por la
salida standard o no
```

```
                                    //Si no tiene argumentos, mostramos las instrucciones para su uso
                                    if(Array.getLength(args)==0){
                                                System.out.println("\n\n"+ABOUT+"\n");
                                                System.out.println("USO:\tMastin [opciones] -e/-d -in inFile -out outFile");
                                                System.out.println("\t\t-e:\tencripta");
                                                System.out.println("\t\t-d:\tdesencripta");
                                                System.out.println("\t\t-in inFile:\tinFile es la ruta del fichero de entrada");
                                                System.out.println("\t\t-out outFile:\toutFile es la ruta del fichero de salida");
                                                System.out.println("\n\tLista de opciones:");
                                                System.out.println("\t\t-about:\t\tmuestra información acerca de la aplicación");
                                                System.out.println("\t\t-Alias alias:\talias del la entrada en la KeyStore de donde se
extraerá la clave privada.\n\t\t\t\tPor defecto se usará \"mykey\"");
                                                System.out.println("\t\t-KeyPassword keyPassword:\tespecifica la contraseña de la clave
del alias.\n\t\t\t\tPor defecto se usará la misma clave que para el KeyStore (\"-KeyStorePassword KeyStorePassword\")");
                                                System.out.println("\t\t-KeyStorePassword KeyStorePassword:\tespecifica la contraseña de
entrada al KeyStore");
                                                System.out.println("\t\t-KeyStorePath KeyStorePath:\tespecifica la ruta del fichero
KeyStore");
                                                System.out.println("\t\t-KeyStoreType KeyStoreType:\tespecifica el tipo de
KeyStore.\n\t\t\t\tPor defecto se usará \"JKS\"");
                                                System.out.println("\t\t-v:\t\tmuestra información del desarrollo de la
aplicación(equivalente a -verbose)");
                                                System.out.println("\t\t-verbose:\tmuestra información del desarrollo de la aplicación");
                                                System.out.println("\t\t-version:\tmuestra información de la versión de la aplicación");
                                                System.out.println("\n\n");

                                                //Saldremos del programa
                                                System.exit(0);
                                    }

                                    //Recorremos todos los argumentos, mostrando lo que nos piden y vamos asignando valores a las
variables
                                    for(int i=0; i<Array.getLength(args); i++){
                                                if(args[i].compareToIgnoreCase("-version")==0){
                                                            System.out.println(NOMBRE+" v"+VERSION);

                                                }else if(args[i].compareToIgnoreCase("-about")==0){
```

```
                System.out.println("\n\n"+ABOUT+"\n");

            }else if(args[i].compareToIgnoreCase("-e")==0){
                opMode=javax.crypto.Cipher.ENCRYPT_MODE;

            }else if(args[i].compareToIgnoreCase("-d")==0){
                opMode=javax.crypto.Cipher.DECRYPT_MODE;

            }else if(args[i].compareToIgnoreCase("-in")==0){
                pathFileIn=new String(args[i+1]);

            }else if(args[i].compareToIgnoreCase("-out")==0){
                pathFileOut=new String(args[i+1]);

            }else if(args[i].compareToIgnoreCase("-v")==0){
                verbose=true;

            }else if(args[i].compareToIgnoreCase("-verbose")==0){
                verbose=true;

            }else if(args[i].compareToIgnoreCase("-KeyStorePath")==0){
                pathKeyStore=new String(args[i+1]);

            }else if(args[i].compareToIgnoreCase("-KeyStoreType")==0){
                tipoKeyStore=new String(args[i+1]);

            }else if(args[i].compareToIgnoreCase("-KeyStorePassword")==0){
                passwordKeyStore.write(args[i+1],0,args[i+1].length());

            }else if(args[i].compareToIgnoreCase("-Alias")==0){
                alias=args[i+1];

            }else if(args[i].compareToIgnoreCase("-KeyPassWord keyPassword")==0){
                keyPassword.write(args[i+1],0,args[i+1].length());

            }


        }
```

```
                                //En caso de que no se nos hallan especificado algunos valores, ponemos los valores por defecto
                                if(pathKeyStore.length()==0){
                                        pathKeyStore+=System.getProperty("user.home")+File.separatorChar+".keystore";
                                }
                                if(tipoKeyStore.length()==0){
                                        tipoKeyStore+=TIPO_KEYSTORE_DEFECTO;
                                }
                                if(keyPassword.size()==0 && passwordKeyStore.size()!=0){
                                        keyPassword.write(passwordKeyStore.toString());
                                }
                                if(alias.length()==0){
                                        alias=Mastin.ALIAS_DEFECTO;
                                }

                                //Llamamos a la función es.us.esi.bab.mastin.Mastin.engineFile con el Key extraido de la KeyStore
mediante es.us.esi.bab.mastin.getKeyFromKeyStore
                                es.us.esi.bab.mastin.Mastin.workWithFiles(opMode, pathFileIn, pathFileOut,
es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore.toCharArray(), tipoKeyStore, keyPassword.toCharArray(),
alias, verbose, System.out), verbose, System.out);
                                if(verbose){
                                        System.out.println(" ");           //Por cuestiones puramente estéticas
                                }

                        }catch(Exception e){
                                System.out.println(e);
                        }
                }



                /**
                 * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Puede mostrar mensajes o no dependiendo del valor de
verbose
                 *
                 * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
                 * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
                 * @param tipoKeyStore Tipo del KeyStore del cual se va a extraer la Key
```

```
             * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param alias Alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param verbose Dependiendo de su valor:<ul><li><b><i>true</i></b>: se imprimirán mensajes por
<i>out</i><li><b><i>false</i></b>: no se imprimirán mensajes por <i>out</i></ul>
             * @param out Por donde se imprimirían los mensajes
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, String tipoKeyStore, char[] keyPassword,
String alias, boolean verbose, PrintStream out) throws Exception{

                        //Tomamos el tiempo en que comienza el trabajo
                        tiempoInicio=new GregorianCalendar();

                        //Abrimos el KeyStore
                        KeyStore ks = KeyStore.getInstance(tipoKeyStore);
                        FileInputStream fks=new FileInputStream(pathKeyStore);
                        ks.load(fks,passwordKeyStore);

                        if(verbose){
                                    out.println("\n\n"+ABOUT+"\n");
                                    out.println("\nComenzado en:\t"+tiempoInicio.getTime());
                                    out.println("\nKeyStore de tipo \""+tipoKeyStore+"\" localizado en \""+pathKeyStore+"\"");
                        }

                        //Comprobamos que el KeyStore contiene el alias especificado
                        if(!ks.containsAlias(alias)){
                                    //En caso de que el Key Store no contenga dicho alias, lanzaremos una MastinException
                                    throw new MastinException("El KeyStore "+pathKeyStore+" no contiene el alias "+alias);
                        }else if(verbose){
                                    out.print("\nEncontrado el alias "+alias);
                        }

                        //Obtenemos una instancia de KeyFactory de tipo RSA
                        KeyFactory kf=KeyFactory.getInstance("RSA");
```

```
                        //Vemos si el alias se corresponde con una Key o con un Certificate
                        if(!ks.isCertificateEntry(alias)){

                                    //Si el alias corresponde a una Key, encriptaremos con la clave privada
                                    //Obtenemos la clave privada de tipo com.sun.net.ssl.internal.ssl.JSA_RSAPrivateKey que luego
transformaremos con el KeyFactory a una de tipo PrivateKey
                                    com.sun.net.ssl.internal.ssl.JSA_RSAPrivateKey llave=(com.sun.net.ssl.internal.ssl.JSA_RSAPrivateKey)
ks.getKey(alias,keyPassword);

                                    if(verbose){
                                                out.print(" de tipo Key.");
                                                out.println("\n\nClave privada:");
                                                out.println("\tAlgoritmo:\t"+llave.getAlgorithm());
                                                out.println("\tFormato:\t"+llave.getFormat()+"\n");
                                                out.println("\tLongitud del módulo (n):\t"+llave.getModulus().bitLength()+" bytes");
                                    }

                                    //Si el algoritmo de la clave privada no coincide con el que utiliza Mastin, lanzaremos una
MastinException
                                    if(!es.us.esi.bab.mastin.Mastin.ALGORITMO_MASTIN.equals(llave.getAlgorithm())){
                                                throw new MastinException("El algoritmo "+llave.getAlgorithm()+" de la clave del alias
"+alias+" no es "+es.us.esi.bab.mastin.Mastin.ALGORITMO_MASTIN);
                                    }else{
                                                //Realizamos la conversión de tipo com.sun.net.ssl.internal.ssl.JSA_RSAPrivateKey a
RSAPrivateCrtKeySpec y luego a PrivateKey con ayuda del KeyFactory y será lo que devolvamos
                                                //Generamos un RSAPrivateCrtKeySpec debido a que si generásemos un RSAPrivateKeySpec, no
funcionaría
                                                RSAPrivateCrtKeySpec miRSAPrivateCrtKeySpec=new RSAPrivateCrtKeySpec(llave.getModulus(),
llave.getPublicExponent(), llave.getPrivateExponent(), llave.getPrimeP(), llave.getPrimeQ(), llave.getPrimeExponentP(),
llave.getPrimeExponentQ(), llave.getCrtCoefficient());
                                                return kf.generatePrivate(miRSAPrivateCrtKeySpec);
                                    }


                        }

                        //En el caso en que el alias no corresponda a una Key, corresponderá a un Certificate y por lo tanto,
encriptaremos con una clave pública
```

```
                        //Obtenemos la clave pública de tipo com.sun.net.ssl.internal.ssl.JSA_RSAPublicKey que posteriormente
transformaremos con el KeyFactory a una de tipo PublicKey
                        com.sun.net.ssl.internal.ssl.JSA_RSAPublicKey llave=(com.sun.net.ssl.internal.ssl.JSA_RSAPublicKey)
ks.getCertificate(alias).getPublicKey();

                        if(verbose){
                                    out.print(" de tipo Certificado:");
                                    out.println(""+ks.getCertificate(alias));
                                    out.println("\nClave Pública:");
                                    out.println("\tAlgoritmo:\t"+llave.getAlgorithm());
                                    out.println("\tFormato:\t"+llave.getFormat());
                                    out.println("\tLongitud del módulo (n):\t"+llave.getModulus().bitLength()+" bytes");
                        }

                        //Si el algoritmo de la clave pública no coincide con el que utiliza Mastin, lanzaremos una MastinException
                        if(!es.us.esi.bab.mastin.Mastin.ALGORITMO_MASTIN.equals(llave.getAlgorithm())){
                                    throw new MastinException("El algoritmo "+llave.getAlgorithm()+" de la clave del alias "+alias+" no es
"+es.us.esi.bab.mastin.Mastin.ALGORITMO_MASTIN);
                        }

                        //Realizamos la conversión de tipo com.sun.net.ssl.internal.ssl.JSA_RSAPublicKey a RSAPublicKeySpec y luego a
PublicKey con ayuda del KeyFactory y será lo que devolvamos
                        RSAPublicKeySpec miRSAPublicKeySpec=new RSAPublicKeySpec(llave.getModulus(), llave.getPublicExponent());
                        return kf.generatePublic(miRSAPublicKeySpec);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Puede mostrar mensajes o no dependiendo del valor de
verbose.<br>Toma como tipo de KeyStore predeterminado {@link es.us.esi.bab.mastin.Mastin#TIPO_KEYSTORE_DEFECTO TIPO_KEYSTORE_DEFECTO}
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param alias Alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param verbose Dependiendo de su valor:<ul><li><b><i>true</i></b>: se imprimirán mensajes por
<i>out</i><li><b><i>false</i></b>: no se imprimirán mensajes por <i>out</i></ul>
```

```
             * @param out Por donde se imprimirían los mensajes
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, char[] keyPassword, String alias, boolean
verbose, PrintStream out) throws Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, TIPO_KEYSTORE_DEFECTO,
keyPassword, alias, verbose, out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Puede mostrar mensajes o no dependiendo del valor de
verbose.<br>Toma como tipo de KeyStore predeterminado {@link es.us.esi.bab.mastin.Mastin#TIPO_KEYSTORE_DEFECTO
TIPO_KEYSTORE_DEFECTO}.<br>Toma como alias predeterminado {@link es.us.esi.bab.mastin.Mastin#ALIAS_DEFECTO ALIAS_DEFECTO}
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param verbose Dependiendo de su valor:<ul><li><b><i>true</i></b>: se imprimirán mensajes por
<i>out</i><li><b><i>false</i></b>: no se imprimirán mensajes por <i>out</i></ul>
             * @param out Por donde se imprimirían los mensajes
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, char[] keyPassword, boolean verbose,
PrintStream out) throws Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, TIPO_KEYSTORE_DEFECTO,
keyPassword, ALIAS_DEFECTO, verbose, out);
            }
```

```
	/**
	 * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Puede mostrar mensajes o no dependiendo del valor de verbose.<br>Toma como tipo de KeyStore predeterminado {@link es.us.esi.bab.mastin.Mastin#TIPO_KEYSTORE_DEFECTO TIPO_KEYSTORE_DEFECTO}.<br>Toma como contraseña para el alias (<i>keyPassword</i>), la misma que para el KeyStore (<i>passwordKeyStore</i>)
	 *
	 * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
	 * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
	 * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
	 * @param alias Alias de la entrada del KeyStore de la que se va a extraer la Key
	 * @param verbose Dependiendo de su valor:<ul><li><b><i>true</i></b>: se imprimirán mensajes por <i>out</i><li><b><i>false</i></b>: no se imprimirán mensajes por <i>out</i></ul>
	 * @param out Por donde se imprimirían los mensajes
	 * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
	 * @throws Exception Excepción producida en la ejecución del código
	 * @since Mastin v1.0
	 */
	public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, String alias, boolean verbose, PrintStream out) throws Exception{
		return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, TIPO_KEYSTORE_DEFECTO, passwordKeyStore, alias, verbose, out);
	}



	/**
	 * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Muestra todos los mensajes por <i>out</i>.<br>Toma como tipo de KeyStore predeterminado {@link es.us.esi.bab.mastin.Mastin#TIPO_KEYSTORE_DEFECTO TIPO_KEYSTORE_DEFECTO}.<br>Toma como contraseña para el alias (<i>keyPassword</i>), la misma que para el KeyStore (<i>passwordKeyStore</i>).<br>Toma como alias predeterminado {@link es.us.esi.bab.mastin.Mastin#ALIAS_DEFECTO ALIAS_DEFECTO}
	 *
	 * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
	 * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
```

```
             * @param out Por donde se imprimirían los mensajes
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, PrintStream out) throws Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, TIPO_KEYSTORE_DEFECTO,
passwordKeyStore, ALIAS_DEFECTO, true, out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>No muestra mensajes
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param tipoKeyStore Tipo del KeyStore del cual se va a extraer la Key
             * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param alias Alias de la entrada del KeyStore de la que se va a extraer la Key
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, String tipoKeyStore, char[] keyPassword,
String alias) throws Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, tipoKeyStore, keyPassword,
alias, false, System.out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>No muestra mensajes.<br>Toma como tipo de KeyStore
```

```
predeterminado {@link es.us.esi.bab.mastin.Mastin#TIPO_KEYSTORE_DEFECTO TIPO_KEYSTORE_DEFECTO}
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param alias Alias de la entrada del KeyStore de la que se va a extraer la Key
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, char[] keyPassword, String alias) throws
Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, TIPO_KEYSTORE_DEFECTO,
keyPassword, alias, false, System.out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Muestra todos los mensajes por <i>out</i>.<br>Toma como
tipo de KeyStore predeterminado {@link es.us.esi.bab.mastin.Mastin#TIPO_KEYSTORE_DEFECTO TIPO_KEYSTORE_DEFECTO}
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param alias Alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param out Por donde se imprimirían los mensajes
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, char[] keyPassword, String alias,
PrintStream out) throws Exception{
```

```
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, TIPO_KEYSTORE_DEFECTO,
keyPassword, alias, true, out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Puede mostrar mensajes o no dependiendo del valor de
verbose.<br>Toma como alias predeterminado {@link es.us.esi.bab.mastin.Mastin#ALIAS_DEFECTO ALIAS_DEFECTO}
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param tipoKeyStore Tipo del KeyStore del cual se va a extraer la Key
             * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param verbose Dependiendo de su valor:<ul><li><b><i>true</i></b>: se imprimirán mensajes por
<i>out</i><li><b><i>false</i></b>: no se imprimirán mensajes por <i>out</i></ul>
             * @param out Por donde se imprimirían los mensajes
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, String tipoKeyStore, char[] keyPassword,
boolean verbose, PrintStream out) throws Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, tipoKeyStore, keyPassword,
ALIAS_DEFECTO, verbose, out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Toma como alias predeterminado {@link
es.us.esi.bab.mastin.Mastin#ALIAS_DEFECTO ALIAS_DEFECTO}.<br>No muestra mensajes
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
```

```
             * @param tipoKeyStore Tipo del KeyStore del cual se va a extraer la Key
             * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, String tipoKeyStore, char[] keyPassword)
throws Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, tipoKeyStore, keyPassword,
ALIAS_DEFECTO, false, System.out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Muestra todos los mensajes por <i>out</i>.<br>Toma como
alias predeterminado {@link es.us.esi.bab.mastin.Mastin#ALIAS_DEFECTO ALIAS_DEFECTO}
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param tipoKeyStore Tipo del KeyStore del cual se va a extraer la Key
             * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param out Por donde se imprimirían los mensajes
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, String tipoKeyStore, char[] keyPassword,
PrintStream out) throws Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, tipoKeyStore, keyPassword,
ALIAS_DEFECTO, true, out);
            }
```

```
	/**
	 * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Muestra todos los mensajes por <i>out</i>.<br>Toma como
alias predeterminado {@link es.us.esi.bab.mastin.Mastin#ALIAS_DEFECTO ALIAS_DEFECTO}.<br>Toma como contraseña para el alias
(<i>keyPassword</i>), la misma que para el KeyStore (<i>passwordKeyStore</i>)
	 *
	 * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
	 * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
	 * @param tipoKeyStore Tipo del KeyStore del cual se va a extraer la Key
	 * @param out Por donde se imprimirían los mensajes
	 * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
	 * @throws Exception Excepción producida en la ejecución del código
	 * @since Mastin v1.0
	 */
	public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, String tipoKeyStore, PrintStream out)
throws Exception{
		return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, tipoKeyStore,
passwordKeyStore, ALIAS_DEFECTO, true, out);
	}



	/**
	 * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Puede mostrar mensajes o no dependiendo del valor de
verbose.<br>Toma como contraseña para el alias (<i>keyPassword</i>), la misma que para el KeyStore (<i>passwordKeyStore</i>).<br>Toma como
alias predeterminado {@link es.us.esi.bab.mastin.Mastin#ALIAS_DEFECTO ALIAS_DEFECTO}.<br>Toma como tipo de KeyStore predeterminado {@link
es.us.esi.bab.mastin.Mastin#TIPO_KEYSTORE_DEFECTO TIPO_KEYSTORE_DEFECTO}
	 *
	 * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
	 * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
	 * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
	 * @param verbose Dependiendo de su valor:<ul><li><b><i>true</i></b>: se imprimirán mensajes por
<i>out</i><li><b><i>false</i></b>: no se imprimirán mensajes por <i>out</i></ul>
	 * @param out Por donde se imprimirían los mensajes
	 * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
```

```
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, boolean verbose, PrintStream out) throws
Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, TIPO_KEYSTORE_DEFECTO,
passwordKeyStore, ALIAS_DEFECTO, verbose, out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>No muestra mensajes.<br>Toma como contraseña para el alias
(<i>keyPassword</i>), la misma que para el KeyStore (<i>passwordKeyStore</i>).<br>Toma como alias predeterminado {@link
es.us.esi.bab.mastin.Mastin#ALIAS_DEFECTO ALIAS_DEFECTO}
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param tipoKeyStore Tipo del KeyStore del cual se va a extraer la Key
             * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, String tipoKeyStore) throws Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, tipoKeyStore,
passwordKeyStore, ALIAS_DEFECTO, false, System.out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>No muestra mensajes.<br>Toma como contraseña para el alias
(<i>keyPassword</i>), la misma que para el KeyStore (<i>passwordKeyStore</i>).<br>Toma como alias predeterminado {@link
es.us.esi.bab.mastin.Mastin#ALIAS_DEFECTO ALIAS_DEFECTO}.<br>Toma como tipo de KeyStore predeterminado {@link
```

```
es.us.esi.bab.mastin.Mastin#TIPO_KEYSTORE_DEFECTO TIPO_KEYSTORE_DEFECTO}
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore) throws Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, TIPO_KEYSTORE_DEFECTO,
passwordKeyStore, ALIAS_DEFECTO, false, System.out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>No muestra mensajes.<br>Toma como alias predeterminado
{@link es.us.esi.bab.mastin.Mastin#ALIAS_DEFECTO ALIAS_DEFECTO}.<br>Toma como tipo de KeyStore predeterminado {@link
es.us.esi.bab.mastin.Mastin#TIPO_KEYSTORE_DEFECTO TIPO_KEYSTORE_DEFECTO}
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, char[] keyPassword) throws Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, TIPO_KEYSTORE_DEFECTO,
keyPassword, ALIAS_DEFECTO, false, System.out);
            }
```

```
            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Muestra todos los mensajes por <i>out</i>.<br>Toma como
alias predeterminado {@link es.us.esi.bab.mastin.Mastin#ALIAS_DEFECTO ALIAS_DEFECTO}.<br>Toma como tipo de KeyStore predeterminado {@link
es.us.esi.bab.mastin.Mastin#TIPO_KEYSTORE_DEFECTO TIPO_KEYSTORE_DEFECTO}
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param out Por donde se imprimirían los mensajes
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, char[] keyPassword, PrintStream out) throws
Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, TIPO_KEYSTORE_DEFECTO,
keyPassword, ALIAS_DEFECTO, true, out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Puede mostrar mensajes o no dependiendo del valor de
verbose.<br>Toma como contraseña para el alias (<i>keyPassword</i>), la misma que para el KeyStore (<i>passwordKeyStore</i>)
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param tipoKeyStore Tipo del KeyStore del cual se va a extraer la Key
             * @param alias Alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param verbose Dependiendo de su valor:<ul><li><b><i>true</i></b>: se imprimirán mensajes por
<i>out</i><li><b><i>false</i></b>: no se imprimirán mensajes por <i>out</i></ul>
             * @param out Por donde se imprimirían los mensajes
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
```

```
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, String tipoKeyStore, String alias, boolean
verbose, PrintStream out) throws Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, tipoKeyStore,
passwordKeyStore, alias, verbose, out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Toma como contraseña para el alias (<i>keyPassword</i>), la
misma que para el KeyStore (<i>passwordKeyStore</i>).<br>Muestra todos los mensajes por <i>out</i>
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param tipoKeyStore Tipo del KeyStore del cual se va a extraer la Key
             * @param alias Alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param out Por donde se imprimirían los mensajes
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, String tipoKeyStore, String alias,
PrintStream out) throws Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, tipoKeyStore,
passwordKeyStore, alias, true, out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Toma como contraseña para el alias (<i>keyPassword</i>), la
misma que para el KeyStore (<i>passwordKeyStore</i>).<br>No muestra mensajes
```

```
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param tipoKeyStore Tipo del KeyStore del cual se va a extraer la Key
             * @param alias Alias de la entrada del KeyStore de la que se va a extraer la Key
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, String tipoKeyStore, String alias) throws
Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, tipoKeyStore,
passwordKeyStore, ALIAS_DEFECTO, false, System.out);
            }



            /**
             * Obtiene una clave, pública en caso de que el alias corresponda a una entrada de tipo Certificate o privada en caso de que
el alias corresponda a una entrada de tipo Key en el KeyStore especificado.<br>Muestra todos los mensajes por <i>out</i>
             *
             * @param pathKeyStore Ruta del KeyStore del cual se va a extraer la Key
             * @param passwordKeyStore Contraseña del KeyStore del cual se va a extraer la Key
             * @param tipoKeyStore Tipo del KeyStore del cual se va a extraer la Key
             * @param keyPassword Contraseña del alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param alias Alias de la entrada del KeyStore de la que se va a extraer la Key
             * @param out Por donde se imprimirían los mensajes
             * @return Dependiendo del tipo de entrada referida por el alias en el KeyStore, devolverá un Key de
tipo:<ul><li><b><i>PrivateKey</i></b>: si es un alias corresponde a una entrada de tipo Key en el KeyStore<li><b><i>PublicKey</i></b>: si
el alias corresponde a una entrada de tipo Certificate en el KeyStore</ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static Key getKeyFromKeyStore(String pathKeyStore, char[] passwordKeyStore, String tipoKeyStore, char[] keyPassword,
String alias, PrintStream out) throws Exception{
                        return es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(pathKeyStore, passwordKeyStore, tipoKeyStore, keyPassword,
alias, true, out);
```

```
            }




            /**
             *Encripta o desencripta según opMode con el algoritmo RSA, en modo ECB y con relleno OAEP, el fichero de entrada en el
fichero de salida con la clave especificada pudiendo mostrar o no mensajes según verbose por la salida especificada por out
             *
             * @param opMode Modo de operación del Cipher:<ul><li><b><i>javax.crypto.Cipher.ENCRYPT_MODE</i></b>:
encripta<li><b><i>javax.crypto.Cipher.DECRYPT_MODE</i></b>: desencripta</ul>
             * @param pathFileIn Ruta del fichero de entrada
             * @param pathFileOut Ruta del fichero de salida
             * @param Key Clave RSA que se va a utilizar. Puede ser:<ul><li><b><i>PublicKey</i></b><li><b><i>PrivateKey</i></b></ul>
             * @param verbose Dependiendo de su valor:<ul><li><b><i>true</i></b>: se imprimirán mensajes por
<i>out</i><li><b><i>false</i></b>: no se imprimirán mensajes por <i>out</i></ul>
             * @param out Por donde se imprimirían los mensajes
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static void workWithFiles(int opMode, String pathFileIn, String pathFileOut, java.security.Key key, boolean verbose,
PrintStream out) throws Exception{

                        //Comprobamos opMode
                        if(opMode!=javax.crypto.Cipher.ENCRYPT_MODE && opMode!=javax.crypto.Cipher.DECRYPT_MODE){
                                    throw new MastinException("El modo de operación no es válido");
                        }

                        //Abrimos los ficheros origen y destino
                        File fileIn=new File(pathFileIn);
                        File fileOut=new File(pathFileOut);

                        //Comprobamos que podemos trabajar con ellos
                        if(!fileIn.exists()){
                                    throw new MastinException("El archivo \""+pathFileIn+"\" no existe");
                        }
                        if(!fileIn.canRead()){
```

```
                                    throw new MastinException("No se ha podido leer del archivo \""+pathFileIn+"\"");
                        }
                        if(fileIn.length()==0){
                                    throw new MastinException("El archivo \""+pathFileIn+"\" tiene longitud 0");
                        }
                        if(!fileOut.createNewFile()){
                                    if(verbose){
                                                out.println("\nEl archivo \""+pathFileOut+"\" se sobreescribirá");
                                    }
                        }else{
                                    if(verbose){
                                                out.println("\nSe creará el archivo \""+pathFileOut+"\"");
                                    }
                        }
                        if(!fileOut.canWrite()){
                                    throw new MastinException("No se ha puede escribir en el archivo \""+pathFileOut+"\"");
                        }

                        //Leemos los bytes del fileIn en dataIn
                        FileInputStream canalEntrada=new FileInputStream(fileIn);
                        byte dataIn[]=new byte[(int)fileIn.length()];
                        int numBytes=canalEntrada.read(dataIn);

                        if(verbose){
                                    out.println("\nLeidos "+numBytes+" bytes del archivo \""+pathFileIn+"\"");
                        }

                        //Instalamos dinámicamente el Provider: "org.bouncycastle.jce.provider.BouncyCastleProvider"
                        if(verbose){
                                    out.print("\nAñadimos dinámicamente el Provider:
\"org.bouncycastle.jce.provider.BouncyCastleProvider\"");
                        }
                        int inst=java.security.Security.addProvider(new org.bouncycastle.jce.provider.BouncyCastleProvider());

                        if(verbose){
                                    if(inst!=-1){
                                                out.print("\tOK");
                                    }else{
                                                out.print("\tYA INSTALADO ANTERIORMENTE");
```

```
                                        }
                                }

                                //Obtenemos una instancia del Cipher
                                javax.crypto.Cipher cifrador=javax.crypto.Cipher.getInstance(Mastin.ALGORITMO_CIPHER,"BC");

                                //Lo inicializamos con el modo de operación y la clave
                                cifrador.init(opMode, key);

                                //Creamos el ByteArrayOutputStream de la salida
                                ByteArrayOutputStream baos=new ByteArrayOutputStream();

                                if(verbose){
                                        if(opMode==javax.crypto.Cipher.ENCRYPT_MODE){
                                                out.println("\n\nCifrador:");
                                        }else{
                                                out.println("\n\nDescifrador:");
                                        }
                                        out.println("\tAlgoritmo:\t\t\t"+cifrador.getAlgorithm());
                                        out.println("\tProveedor:\t\t\t"+cifrador.getProvider());
                                        out.println("\tTamaño del bloque de entrada:\t"+cifrador.getBlockSize()+" bytes");
                                        out.println("\tTamaño del bloque de salida:\t"+cifrador.getOutputSize(Array.getLength(dataIn))+"
bytes");
                                }

                                //Realizamos la operación con el Cipher
                                if(Array.getLength(dataIn)<=cifrador.getBlockSize()){
                                        //Si la longitud del array de bytes de entrada es menor o igual al tamaño del bloque de entrada al
Cipher, lo encriptamos / desencriptamos en un solo paso
                                        byte[] byteTemp=cifrador.doFinal(dataIn);
                                        baos.write(byteTemp, 0, Array.getLength(byteTemp));
                                }else{
                                        //En caso de que la longitud del array de bytes de entrada es más grande que el tamaño del bloque de
entrada al Cipher, descompondremos dicho array en otros del tamaño del bloque y los encriptaremos / desencriptamos en varios pasos
                                        int inOffset=0;
                                        byte[] byteTemp=new byte[cifrador.getOutputSize(cifrador.getBlockSize())];         //El tamaño de
dicho array es el del mayor bloque que nos vamos a encontrar a la salida
```

```
                                    for(inOffset=0; Array.getLength(dataIn)-inOffset>cifrador.getBlockSize();
inOffset+=cifrador.getBlockSize()){
                                                byteTemp=cifrador.doFinal(dataIn, inOffset, cifrador.getBlockSize());
                                                baos.write(byteTemp, 0, Array.getLength(byteTemp));
                                    }
                                    byteTemp=cifrador.doFinal(dataIn, inOffset, Array.getLength(dataIn)-inOffset);
                                    baos.write(byteTemp, 0, Array.getLength(byteTemp));
                        }

                        FileOutputStream canalSalida=new FileOutputStream(fileOut);
                        canalSalida.write(baos.toByteArray());
                        canalSalida.close();

                        if(verbose){
                                    out.println("\nTamaño del fichero de salida:\t"+baos.size()+" bytes");
                                    //Tomamos el tiempo de finalización
                                    tiempoFin=new GregorianCalendar();
                                    out.println("\nFinalizado en:\t"+tiempoFin.getTime());
                                    long tiempoTranscurrido=(tiempoFin.getTimeInMillis()-tiempoInicio.getTimeInMillis())/1000;
                                    out.println("Transcurridos:\t"+tiempoTranscurrido+" segundos");
                                    if(tiempoTranscurrido==0){
                                                tiempoTranscurrido++;
                                    }
                                    out.println("Rendimiento de entrada:\t"+numBytes/tiempoTranscurrido+" bytes/s");
                                    out.println("Rendimiento de salida:\t"+baos.size()/tiempoTranscurrido+" bytes/s");
                        }
            }



            /**
             *Encripta o desencripta según opMode con el algoritmo RSA, en modo ECB y con relleno OAEP, el fichero de entrada en el
fichero de salida con la clave especificada mostrando todos los mensajes por la salida especificada por out
             *
             * @param opMode Modo de operación del Cipher:<ul><li><b><i>javax.crypto.Cipher.ENCRYPT_MODE</i></b>:
encripta<li><b><i>javax.crypto.Cipher.DECRYPT_MODE</i></b>: desencripta</ul>
             * @param pathFileIn Ruta del fichero de entrada
             * @param pathFileOut Ruta del fichero de salida
             * @param Key Clave RSA que se va a utilizar. Puede ser:<ul><li><b><i>PublicKey</i></b><li><b><i>PrivateKey</i></b></ul>
```

```
             * @param out Por donde se imprimirían los mensajes
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static void workWithFiles(int opMode, String pathFileIn, String pathFileOut, java.security.Key key, PrintStream out)
throws Exception{
                        es.us.esi.bab.mastin.Mastin.workWithFiles(opMode, pathFileIn, pathFileOut, key, true, out);
            }



            /**
             *Encripta o desencripta según opMode con el algoritmo RSA, en modo ECB y con relleno OAEP, el fichero de entrada en el
fichero de salida con la clave especificada no mostrando mensajes
             *
             * @param opMode Modo de operación del Cipher:<ul><li><b><i>javax.crypto.Cipher.ENCRYPT_MODE</i></b>:
encripta<li><b><i>javax.crypto.Cipher.DECRYPT_MODE</i></b>: desencripta</ul>
             * @param pathFileIn Ruta del fichero de entrada
             * @param pathFileOut Ruta del fichero de salida
             * @param Key Clave RSA que se va a utilizar. Puede ser:<ul><li><b><i>PublicKey</i></b><li><b><i>PrivateKey</i></b></ul>
             * @throws Exception Excepción producida en la ejecución del código
             * @since Mastin v1.0
             */
            public static void workWithFiles(int opMode, String pathFileIn, String pathFileOut, java.security.Key key) throws Exception{
                        es.us.esi.bab.mastin.Mastin.workWithFiles(opMode, pathFileIn, pathFileOut, key, false, System.out);
            }

}
```

MastinException.java

```
package es.us.esi.bab.mastin;

/**
 * Clase que me permite lanzar las excepciones provocadas en la aplicación Mastin
 * @author Gabriel Babiano Huete
 * @version 1.0
```

```
 * @since Mastin v1.0
 */
public class MastinException extends Exception{

            /**
             * Constructor de la clase MastinException
             * @param mesaje mensaje que se desea que muestre la excepción cuando se muestre
             * @since Mastin v1.0
             */
            public MastinException(String mensaje){
                        super(mensaje);
            }
}
```

XMastin.java

```
package es.us.esi.bab.xmastin;


import java.awt.*;
import java.awt.event.*;
import java.io.File;
import java.io.FileReader;
import java.io.PrintStream;
import java.io.FileOutputStream;
import javax.swing.*;

import java.util.Date;


/**
 * Clase que utiliza la API de Mastin v1.0 para encriptar / desencriptar archivos con el algoritmo RSA, en el modo ECB y con relleno OAEP.
 *
 * @author Gabriel Babiano Huete (<a hfer="mailto:gabrielbabiano@yahoo.es">gabrielbabiano@yahoo.es</a>)
 * @version 1.0
 * @since XMastin v1.0
 */
```

```
public class XMastin implements ActionListener{

            //Constantes que contienen el nombre de la aplicación, la versión y el "acerca de"
            /**
             * Nombre de la aplicación
             *
             * @since XMastin v1.0
             */
            public static final String NOMBRE="XMastin";    //Nombre de la aplicación

            /**
             * Version de la aplicación
             *
             * @since XMastin v1.0
             */
            public static final String VERSION="1.0";       //Versión de la aplicación

            /**
             * "Acerca de" la aplicación
             *
             * @since XMastin v1.0
             */
            public static final String ABOUT=NOMBRE+" v"+VERSION+"\nAplicación creada por Gabriel Babiano Huete
(gabrielbabiano@yahoo.es)";         //Acerca de la aplicación


            private int opMode=0;   //Modo de operación

            //Los objetos se definen aquí para que sean accesibles
            private static JButton botonGo, botonFileIn, botonFileOut, botonRutaKS, botonJava, botonJSSE, botonLineaDeComandos,
botonBouncyCastle, botonMastin, botonAbout, botonCerrarLog, botonCerrarFrame, botonCerrarLineaDeComandos;
            private static JTextField txtFileIn, txtFileOut, txtRutaKS, txtJava, txtJSSE, txtBouncyCastle, txtAlias, txtMastin;
            private static JPasswordField passwordKS, passwordAlias;
            private static JRadioButton jrbEnc, jrbDesenc;
            private static JComboBox comboTiposKS;
            private static JCheckBox boxAbout, boxVersion, boxVerbose;
            private static JFrame frame, frameLog, frameLineaDeComandos;
```

```
           public static void main(String[] args) {
                       try {
                       //EL "Java Look and Feel" que funciona en todas partes
                                   UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());


                                   //Icono de los botones de seleccionar fichero
                                   ImageIcon  openIcon;
                                   if(File.separatorChar=='/'){       //Si estamos en Linux
                                               openIcon=new ImageIcon("/usr/share/Mastin_v1.0/images/open.gif");
                                   }else{ //Si estamos en win32
                                               openIcon=new ImageIcon("c:"+File.separatorChar+"Archivos de
Programa"+File.separatorChar+"Mastin_v1.0"+File.separatorChar+"images"+File.separatorChar+"open.gif");
                                   }

                                   //GribBag para tener el formulario ordenado
                                   GridBagLayout gridBag=new GridBagLayout();
                                   GridBagConstraints constraints=new GridBagConstraints();

                                   //Panel de las acciones
                                   jrbEnc=new JRadioButton("Encriptar",true);      //Seleccionamos encriptar por defecto
                                   jrbDesenc=new JRadioButton("Desencriptar");
                                   ButtonGroup bg = new ButtonGroup();
                                   bg.add(jrbEnc);
                                   bg.add(jrbDesenc);
                                   JPanel panelAcciones=new JPanel();
                                   panelAcciones.setBorder(BorderFactory.createCompoundBorder(BorderFactory.createTitledBorder("Acciones"
), BorderFactory.createEmptyBorder(5,5,5,5)));
                                   panelAcciones.add(jrbEnc);
                                   panelAcciones.add(jrbDesenc);

                                   //Panel de los ficheros
                                   JLabel labelFileIn=new JLabel("Ruta del fichero de entrada:");
                                   txtFileIn=new JTextField(30);
                                   botonFileIn=new JButton(openIcon);
                                   botonFileIn.setToolTipText("Pulse aquí para seleccionar el archivo");
                                   botonFileIn.addActionListener(new XMastin());
                                   JLabel labelFileOut=new JLabel("Ruta del fichero de salida:");
                                   txtFileOut=new JTextField(30);
```

```
                                botonFileOut=new JButton(openIcon);
                                botonFileOut.setToolTipText("Pulse aquí para seleccionar el archivo");
                                botonFileOut.addActionListener(new XMastin());
                                JPanel panelFicheros=new JPanel();
                                panelFicheros.setBorder(BorderFactory.createCompoundBorder(BorderFactory.createTitledBorder("Ficheros"
), BorderFactory.createEmptyBorder(5,5,5,5)));
                                panelFicheros.setLayout(gridBag);
                                constraints.gridx=0;
                                constraints.gridy=0;
                                gridBag.setConstraints(labelFileIn,constraints);
                                panelFicheros.add(labelFileIn);
                                constraints.gridx=1;
                                constraints.gridy=0;
                                gridBag.setConstraints(txtFileIn,constraints);
                                panelFicheros.add(txtFileIn);
                                constraints.gridx=2;
                                constraints.gridy=0;
                                gridBag.setConstraints(botonFileIn,constraints);
                                panelFicheros.add(botonFileIn);
                                constraints.gridx=0;
                                constraints.gridy=1;
                                gridBag.setConstraints(labelFileOut,constraints);
                                panelFicheros.add(labelFileOut);
                                constraints.gridx=1;
                                constraints.gridy=1;
                                gridBag.setConstraints(txtFileOut,constraints);
                                panelFicheros.add(txtFileOut);
                                constraints.gridx=2;
                                constraints.gridy=1;
                                gridBag.setConstraints(botonFileOut,constraints);
                                panelFicheros.add(botonFileOut);

                                //Panel del KeyStore
                                JLabel labelRutaKS=new JLabel("Ruta del KeyStore:");
                                txtRutaKS=new JTextField(30);
                                txtRutaKS.setText(System.getProperty("user.home")+File.separatorChar+".keystore");
                                botonRutaKS=new JButton(openIcon);
                                botonRutaKS.setToolTipText("Pulse aquí para seleccionar el KeyStore a utilizar");
                                botonRutaKS.addActionListener(new XMastin());
```

```
                                JLabel labelTipoKS=new JLabel("Tipo de KeyStore:");
                                String[] tiposKS={"JKS"};
                                comboTiposKS=new JComboBox(tiposKS);
                                comboTiposKS.setSelectedIndex(0);
                                JLabel labelPasswordKS=new JLabel("Password del KeyStore:");
                                passwordKS=new JPasswordField(10);
                                JPanel panelKS=new JPanel();
                                panelKS.setBorder(BorderFactory.createCompoundBorder(BorderFactory.createTitledBorder("KeyStore"),
BorderFactory.createEmptyBorder(5,5,5,5)));
                                panelKS.setLayout(gridBag);
                                constraints.gridx=0;
                                constraints.gridy=0;
                                gridBag.setConstraints(labelRutaKS,constraints);
                                panelKS.add(labelRutaKS);
                                constraints.gridx=1;
                                constraints.gridy=0;
                                gridBag.setConstraints(txtRutaKS,constraints);
                                panelKS.add(txtRutaKS);
                                constraints.gridx=2;
                                constraints.gridy=0;
                                gridBag.setConstraints(botonRutaKS,constraints);
                                panelKS.add(botonRutaKS);
                                constraints.gridx=0;
                                constraints.gridy=1;
                                gridBag.setConstraints(labelTipoKS,constraints);
                                panelKS.add(labelTipoKS);
                                constraints.gridx=1;
                                constraints.gridy=1;
                                gridBag.setConstraints(comboTiposKS,constraints);
                                panelKS.add(comboTiposKS);
                                constraints.gridx=0;
                                constraints.gridy=2;
                                gridBag.setConstraints(labelPasswordKS,constraints);
                                panelKS.add(labelPasswordKS);
                                constraints.gridx=1;
                                constraints.gridy=2;
                                gridBag.setConstraints(passwordKS,constraints);
                                panelKS.add(passwordKS);
```

```
                                    //Panel del alias
                                    JLabel labelAlias=new JLabel("Alias:");
                                    txtAlias=new JTextField(10);
                                    txtAlias.setText("mykey");
                                    JLabel labelPasswordAlias=new JLabel("Password del alias:");
                                    passwordAlias=new JPasswordField(10);
                                    JPanel panelAlias=new JPanel();
                                    panelAlias.setBorder(BorderFactory.createCompoundBorder(BorderFactory.createTitledBorder("Alias"),
BorderFactory.createEmptyBorder(5,5,5,5)));
                                    panelAlias.setLayout(gridBag);
                                    constraints.gridx=0;
                                    constraints.gridy=0;
                                    gridBag.setConstraints(labelAlias,constraints);
                                    panelAlias.add(labelAlias);
                                    constraints.gridx=1;
                                    constraints.gridy=0;
                                    gridBag.setConstraints(txtAlias,constraints);
                                    panelAlias.add(txtAlias);
                                    constraints.gridx=0;
                                    constraints.gridy=1;
                                    gridBag.setConstraints(labelPasswordAlias,constraints);
                                    panelAlias.add(labelPasswordAlias);
                                    constraints.gridx=1;
                                    constraints.gridy=1;
                                    gridBag.setConstraints(passwordAlias,constraints);
                                    panelAlias.add(passwordAlias);

                                    //Panel general, que incluye el de las acciones, ficheros, KeyStore y alias
                                    JPanel panelGeneral=new JPanel();
                                    panelGeneral.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
                                    panelGeneral.setLayout(gridBag);
                                    constraints.gridx=0;
                                    constraints.gridy=0;
                                    gridBag.setConstraints(panelAcciones,constraints);
                                    panelGeneral.add(panelAcciones);
                                    constraints.gridx=0;
                                    constraints.gridy=1;
                                    gridBag.setConstraints(panelFicheros,constraints);
                                    panelGeneral.add(panelFicheros);
```

```
                                    constraints.gridx=0;
                                    constraints.gridy=2;
                                    gridBag.setConstraints(panelKS,constraints);
                                    panelGeneral.add(panelKS);
                                    constraints.gridx=0;
                                    constraints.gridy=3;
                                    gridBag.setConstraints(panelAlias,constraints);
                                    panelGeneral.add(panelAlias);

                                    //Panel de las opciones
                                    boxVerbose=new JCheckBox("Verbose",true);
                                    boxVersion=new JCheckBox("Version",true);
                                    boxAbout=new JCheckBox("Acerca de...",true);
                                    JPanel panelOpciones=new JPanel();
                                    panelOpciones.setBorder(BorderFactory.createCompoundBorder(BorderFactory.createTitledBorder("Opciones"
), BorderFactory.createEmptyBorder(5,5,5,5)));
                                    panelOpciones.setLayout(gridBag);
                                    constraints.gridx=0;
                                    constraints.gridy=0;
                                    gridBag.setConstraints(boxVerbose,constraints);
                                    panelOpciones.add(boxVerbose);
                                    constraints.gridx=0;
                                    constraints.gridy=1;
                                    gridBag.setConstraints(boxVersion,constraints);
                                    panelOpciones.add(boxVersion);
                                    constraints.gridx=0;
                                    constraints.gridy=2;
                                    gridBag.setConstraints(boxAbout,constraints);
                                    panelOpciones.add(boxAbout);

                                    //Panel de Java2
                                    JLabel labelJava=new JLabel("Ruta de \"java\":");
                                    txtJava=new JTextField(30);
                                    if(File.separatorChar=='/'){        //Si estamos en Linux
                                                txtJava.setText("/usr/java/j2sdk1.4.0_02/jre/bin/java");
                                    }else{      //Si estamos en Win32
                                                txtJava.setText("c:\\j2sdk1.4.0_02\\jre\\bin\\java.exe");
                                    }
                                    botonJava=new JButton(openIcon);
```

```
                                botonJava.setToolTipText("Pulse aquí para seleccionar la ruta de \"java\"");
                                botonJava.addActionListener(new XMastin());
                                JLabel labelJSSE=new JLabel("Ruta de \"jsse.jar\":");
                                txtJSSE=new JTextField(30);
                                if(File.separatorChar=='/'){        //Si estamos en Linux
                                          txtJSSE.setText("/usr/java/j2sdk1.4.0_02/jre/lib/jsse.jar");
                                }else{      //Si estamos en Win32
                                          txtJSSE.setText("c:\\j2sdk1.4.0_02\\jre\\lib\\jsse.jar");
                                }
                                botonJSSE=new JButton(openIcon);
                                botonJSSE.setToolTipText("Pulse aquí para seleccionar la ruta de \"jsse.jar\"");
                                botonJSSE.addActionListener(new XMastin());
                                JPanel panelJava2=new JPanel();
                                panelJava2.setBorder(BorderFactory.createCompoundBorder(BorderFactory.createTitledBorder("Java 2 SDK
1.4.0"), BorderFactory.createEmptyBorder(5,5,5,5)));
                                panelJava2.setLayout(gridBag);
                                constraints.gridx=0;
                                constraints.gridy=0;
                                gridBag.setConstraints(labelJava,constraints);
                                panelJava2.add(labelJava);
                                constraints.gridx=1;
                                constraints.gridy=0;
                                gridBag.setConstraints(txtJava,constraints);
                                panelJava2.add(txtJava);
                                constraints.gridx=2;
                                constraints.gridy=0;
                                gridBag.setConstraints(botonJava,constraints);
                                panelJava2.add(botonJava);
                                constraints.gridx=0;
                                constraints.gridy=1;
                                gridBag.setConstraints(labelJSSE,constraints);
                                panelJava2.add(labelJSSE);
                                constraints.gridx=1;
                                constraints.gridy=1;
                                gridBag.setConstraints(txtJSSE,constraints);
                                panelJava2.add(txtJSSE);
                                constraints.gridx=2;
                                constraints.gridy=1;
                                gridBag.setConstraints(botonJSSE,constraints);
```

```
                                panelJava2.add(botonJSSE);

                                //Panel de BouncyCastle
                                JLabel labelBouncyCastle=new JLabel("Ruta del Provider de Bouncy Castle 1.15");
                                txtBouncyCastle=new JTextField(30);
                                if(File.separatorChar=='/'){        //Si estamos en Linux
                                        txtBouncyCastle.setText("/usr/share/crypto-115/jars/bcprov-jdk14-115.jar");
                                }else{      //Si estamos en Win32
                                        txtBouncyCastle.setText("\"c:\\archivos de programa\\crypto-115\\jars\\bcprov-jdk14-
115.jar\"");
                                }
                                botonBouncyCastle=new JButton(openIcon);
                                botonBouncyCastle.setToolTipText("Pulse aquí para seleccionar la ruta del Provider de Bouncy Castle");
                                botonBouncyCastle.addActionListener(new XMastin());
                                JPanel panelBouncyCastle=new JPanel();
                                panelBouncyCastle.setBorder(BorderFactory.createCompoundBorder(BorderFactory.createTitledBorder("Bounc
y Castle Provider 1.15"), BorderFactory.createEmptyBorder(5,5,5,5)));
                                panelBouncyCastle.setLayout(gridBag);
                                constraints.gridx=0;
                                constraints.gridy=0;
                                gridBag.setConstraints(labelBouncyCastle,constraints);
                                panelBouncyCastle.add(labelBouncyCastle);
                                constraints.gridx=1;
                                constraints.gridy=0;
                                gridBag.setConstraints(txtBouncyCastle,constraints);
                                panelBouncyCastle.add(txtBouncyCastle);
                                constraints.gridx=2;
                                constraints.gridy=0;
                                gridBag.setConstraints(botonBouncyCastle,constraints);
                                panelBouncyCastle.add(botonBouncyCastle);

                                //Panel de Mastin
                                JLabel labelMastin=new JLabel("Ruta del paquete Mastin");
                                txtMastin=new JTextField(30);
                                if(File.separatorChar=='/'){        //Si estamos en Linux
                                        txtMastin.setText("/usr/share/Mastin_v1.0/jars/");
                                }else{      //Si estamos en Win32
                                        txtMastin.setText("\"c:\\archivos de programa\\Mastin_v1.0\\jars\\\"");
                                }
```

```
                                botonMastin=new JButton(openIcon);
                                botonMastin.setToolTipText("Pulse aquí para seleccionar la ruta del paquete Mastin");
                                botonMastin.addActionListener(new XMastin());
                                JPanel panelMastin=new JPanel();
                                panelMastin.setBorder(BorderFactory.createCompoundBorder(BorderFactory.createTitledBorder("Mastin"),
BorderFactory.createEmptyBorder(5,5,5,5)));
                                panelMastin.setLayout(gridBag);
                                constraints.gridx=0;
                                constraints.gridy=0;
                                gridBag.setConstraints(labelMastin,constraints);
                                panelMastin.add(labelMastin);
                                constraints.gridx=1;
                                constraints.gridy=0;
                                gridBag.setConstraints(txtMastin,constraints);
                                panelMastin.add(txtMastin);
                                constraints.gridx=2;
                                constraints.gridy=0;
                                gridBag.setConstraints(botonMastin,constraints);
                                panelMastin.add(botonMastin);



                                //Panel de la línea de comandos, que incluye el de las opciones, el de Java2, el de BouncyCastle y el
de Mastin
                                JPanel panelLineaDeComandos=new JPanel();
                                panelLineaDeComandos.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
                                panelLineaDeComandos.setLayout(gridBag);
                                constraints.gridx=0;
                                constraints.gridy=0;
                                gridBag.setConstraints(panelOpciones,constraints);
                                panelLineaDeComandos.add(panelOpciones);
                                constraints.gridx=0;
                                constraints.gridy=1;
                                gridBag.setConstraints(panelJava2,constraints);
                                panelLineaDeComandos.add(panelJava2);
                                constraints.gridx=0;
                                constraints.gridy=2;
                                gridBag.setConstraints(panelBouncyCastle,constraints);
                                panelLineaDeComandos.add(panelBouncyCastle);
```

```
                                constraints.gridx=0;
                                constraints.gridy=3;
                                gridBag.setConstraints(panelMastin,constraints);
                                panelLineaDeComandos.add(panelMastin);

                                //PanelTabb que incluye el general y el de la línea de comandos
                                JTabbedPane panelTabbed=new JTabbedPane();
                                panelTabbed.addTab("General", panelGeneral);
                                panelTabbed.addTab("Linea de comandos",panelLineaDeComandos);

                                //Panel de abajo
                                botonGo=new JButton("Go!");
                                botonGo.setToolTipText("Pulse aquí para realizar la acción selecionada");
                                botonGo.addActionListener(new XMastin());
                                botonCerrarFrame=new JButton("Cerrar");
                                botonCerrarFrame.setToolTipText("Pulse aquí para cerrar");
                                botonCerrarFrame.addActionListener(new XMastin());
                                botonLineaDeComandos=new JButton("Línea de comandos");
                                botonLineaDeComandos.setToolTipText("Pulse aquí para ver la línea de comandos resultante");
                                botonLineaDeComandos.addActionListener(new XMastin());
                                botonAbout=new JButton("Acerca de...");
                                botonAbout.setToolTipText("Pulse aquí para ver el \"Acerca de...\"");
                                botonAbout.addActionListener(new XMastin());
                                JPanel panelAbajo=new JPanel();
                                panelAbajo.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
                                panelAbajo.setLayout(new BoxLayout(panelAbajo,BoxLayout.X_AXIS));
                                panelAbajo.add(Box.createHorizontalGlue());
                                panelAbajo.add(botonGo);
                                panelAbajo.add(Box.createHorizontalGlue());
                                panelAbajo.add(botonLineaDeComandos);
                                panelAbajo.add(botonAbout);
                                panelAbajo.add(botonCerrarFrame);

                                //El JFrame que incluye el panelTabbed y el de abajo
                                frame = new JFrame("XMastin v1.0");
                                frame.getContentPane().setLayout(gridBag);
                                constraints.fill=GridBagConstraints.HORIZONTAL;
                                constraints.gridx=0;
                                constraints.gridy=0;
```

```
                        gridBag.setConstraints(panelTabbed,constraints);
                        frame.getContentPane().add(panelTabbed);
                        constraints.gridx=0;
                        constraints.gridy=1;
                        gridBag.setConstraints(panelAbajo,constraints);
                        frame.getContentPane().add(panelAbajo);
                        frame.setResizable(false);
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.pack();
                frame.setVisible(true);

                }catch(Exception e){
                        JOptionPane.showMessageDialog(frame,e.toString(),"Excepción",JOptionPane.ERROR_MESSAGE);

                }

        }



        /**
         * Método para obtener la línea de comandos a partir de los datos del formulario de XMastin
         *
         * @return la línea de comandos a partir de los datos del formulario XMastin
         * @since XMastin v1.0
         */
        public String obtenerLineaDeComandos(){

                String cadena="";

                //Localización de "java"
                cadena+=txtJava.getText();

                //CLASSPATH
                if(File.separatorChar=='/'){       //Si estamos en Linux
                        cadena+=" -cp "+txtJSSE.getText()+":"+txtBouncyCastle.getText()+":"+txtMastin.getText();
                }else{      //Si estamos en Win32
                        cadena+=" -cp "+txtJSSE.getText()+";"+txtBouncyCastle.getText()+";"+txtMastin.getText();
```

```
                }

                //Clase es.us.esi.bab.mastin.Mastin
                cadena+=" es.us.esi.bab.mastin.Mastin";

                //Verbose
                if(boxVerbose.isSelected()){
                            cadena+=" -v";
                }

                //"Acerca de"
                if(boxAbout.isSelected()){
                            cadena+=" -about";
                }

                //Version
                if(boxVersion.isSelected()){
                            cadena+=" -version";
                }

                //Encriptar / desencriptar
                if(jrbEnc.isSelected()){
                            //Si está seleccionado enriptar
                            cadena+=" -e ";
                }else{
                            //Si está seleccionado desencriptar
                            cadena+=" -d ";
                }

                //Fichero de entrada
                if(txtFileIn.getText().length()>0){
                            cadena+=" -in "+txtFileIn.getText();
                }

                //Fichero de salida
                if(txtFileOut.getText().length()>0){
                            cadena+=" -out "+txtFileOut.getText();
                }
```

```
                //Ruta del KeyStore
                if(txtRutaKS.getText().length()>0){
                            cadena+=" -KeyStorePath "+txtRutaKS.getText();
                }

                //Tipo de KeyStore
                cadena+=" -KeyStoreType "+(String) comboTiposKS.getSelectedItem();

                //Contraseña del KeyStore
                if((new String(passwordKS.getPassword())).length()>0){
                            cadena+=" -KeyStorePassword "+new String(passwordKS.getPassword());
                }

                //Alias de la entrada al KeyStore
                if(txtAlias.getText().length()>0){
                            cadena+=" -Alias "+txtAlias.getText();
                }

                //Contraseña de la entrada al KeyStore
                if((new String(passwordAlias.getPassword())).length()>0){
                            cadena+=" -KeyPassword "+new String(passwordAlias.getPassword());
                }

                return cadena;
        }



        /**
         * Método de la interfaz ActionListener para dar utilidad a los botones
         *
         * @param evt evento que ha ocurrido
         * @since XMastin v1.0
         */
        public void actionPerformed(ActionEvent evt){

                //Ponemos el cursor de espera (reloj)
                frame.setCursor(new Cursor(Cursor.WAIT_CURSOR));
                try{
```

```
                                        //Obtenemos el boton que se ha pulsado
                                        Object source=evt.getSource();

                                        //El JFileChooser para seleccionar archivos
                                        JFileChooser fc=new JFileChooser();
                                        fc.setDialogTitle("Seleccionar archivo...");

                                        if(source==botonGo){

                                                    if(jrbEnc.isSelected()){
                                                                //Si está seleccionado enriptar
                                                                opMode=javax.crypto.Cipher.ENCRYPT_MODE;
                                                    }else{
                                                                //Si está seleccionado desencriptar
                                                                opMode=javax.crypto.Cipher.DECRYPT_MODE;
                                                    }

                                                    //Archivo de log llamado MastinLog.txt
                                                    File fileLog;
                                                    if (File.separatorChar=='/'){       //Si estamos en Linux
                                                                fileLog=new File("/usr/share/Mastin_v1.0/MastinLog.txt");
                                                    }else{      //Si estamos en Win32
                                                                fileLog=new File("c:"+File.separatorChar+"archivos de
programa"+File.separatorChar+"Mastin_v1.0"+File.separatorChar+"MastinLog.txt");
                                                    }
                                                    PrintStream ps=new PrintStream(new FileOutputStream(fileLog));
                                                    if(new String(passwordAlias.getPassword()).length()>0){
                                                                es.us.esi.bab.mastin.Mastin.workWithFiles(opMode, txtFileIn.getText(),
txtFileOut.getText(), es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(txtRutaKS.getText(), passwordKS.getPassword(), (String)
comboTiposKS.getSelectedItem(), passwordAlias.getPassword(), txtAlias.getText(), boxVerbose.isSelected(), ps), boxVerbose.isSelected(),
ps);
                                                    }else{      //Si no se rellena la contraseña del alias
                                                                es.us.esi.bab.mastin.Mastin.workWithFiles(opMode, txtFileIn.getText(),
txtFileOut.getText(), es.us.esi.bab.mastin.Mastin.getKeyFromKeyStore(txtRutaKS.getText(), passwordKS.getPassword(), (String)
comboTiposKS.getSelectedItem(), txtAlias.getText(), boxVerbose.isSelected(), ps), boxVerbose.isSelected(), ps);
                                                    }
                                                    frameLog=new JFrame("XMastin v1.0: "+fileLog.getAbsolutePath());
                                                    JTextArea txtLog=new JTextArea(40,20);
                                                    txtLog.setEditable(false);
```

```
                    //Comprobamos que podemos trabajar con el fichero de log
                    if(fileLog.exists() && fileLog.canRead() && fileLog.length()>0){
                              //Copiamos el contenido del fichero de log en el canalEntrada
                              FileReader canalEntrada=new FileReader(fileLog);
                              int intLeido=canalEntrada.read();
                              //Y del canalEntrada al txtLog
                              while(intLeido!=-1){
                                        txtLog.setText(txtLog.getText()+(char)intLeido);
                                        intLeido=canalEntrada.read();
                              }
                              canalEntrada.close();
                    }

                    GridBagLayout gridBag=new GridBagLayout();
                    GridBagConstraints constraints=new GridBagConstraints();
                    frameLog.getContentPane().setLayout(gridBag);

                    JScrollPane scrollPane = new JScrollPane(txtLog);
                    scrollPane.setPreferredSize(new Dimension(500,300));
                    botonCerrarLog=new JButton("Cerrar");
                    botonCerrarLog.setToolTipText("Pulse aquí para cerrar");
                    botonCerrarLog.addActionListener(new XMastin());
                    constraints.fill=GridBagConstraints.HORIZONTAL;
                    constraints.gridx=0;
                    constraints.gridy=0;
                    gridBag.setConstraints(scrollPane,constraints);
                    frameLog.getContentPane().add(scrollPane);
                    constraints.fill=GridBagConstraints.NONE;
                    constraints.gridx=0;
                    constraints.gridy=1;
                    gridBag.setConstraints(botonCerrarLog,constraints);
                    frameLog.getContentPane().add(botonCerrarLog);
                    frameLog.setResizable(false);
                    frameLog.pack();
          frameLog.setVisible(true);


          }else if(source==botonFileIn){
```

```
                                        if(fc.showDialog(frame,"Seleccionar")==JFileChooser.APPROVE_OPTION){
                                                txtFileIn.setText(fc.getSelectedFile().getPath());
                                        }

                                }else if(source==botonFileOut){
                                        if(fc.showDialog(frame,"Seleccionar")==JFileChooser.APPROVE_OPTION){
                                                txtFileOut.setText(fc.getSelectedFile().getPath());
                                        }

                                }else if(source==botonRutaKS){
                                        if(fc.showDialog(frame,"Seleccionar")==JFileChooser.APPROVE_OPTION){
                                                txtRutaKS.setText(fc.getSelectedFile().getPath());
                                        }

                                }else if(source==botonJava){
                                        if(fc.showDialog(frame,"Seleccionar")==JFileChooser.APPROVE_OPTION){
                                                txtJava.setText(fc.getSelectedFile().getPath());
                                        }

                                }else if(source==botonLineaDeComandos){
                                        GridBagLayout gridBag=new GridBagLayout();
                                        GridBagConstraints constraints=new GridBagConstraints();
                                        frameLineaDeComandos=new JFrame("Línea de comandos");
                                        frameLineaDeComandos.getContentPane().setLayout(gridBag);
                                        JTextArea txtLineaDeComandos=new JTextArea(40,10);
                                        txtLineaDeComandos.setText(obtenerLineaDeComandos());
                                        txtLineaDeComandos.setEditable(false);
                                        JScrollPane scrollPane = new JScrollPane(txtLineaDeComandos);
                                        scrollPane.setPreferredSize(new Dimension(500,50));
                                        JLabel labelLineaDeComandos=new JLabel("Seleccione y pulse Ctrl^C para copiar y Ctrl^V
para pegar");
                                        botonCerrarLineaDeComandos=new JButton("Cerrar");
                                        botonCerrarLineaDeComandos.setToolTipText("Pulse aquí para cerrar");
                                        botonCerrarLineaDeComandos.addActionListener(new XMastin());
                                        constraints.fill=GridBagConstraints.BOTH;
                                        constraints.gridx=0;
                                        constraints.gridy=0;
                                        constraints.gridwidth=2;
                                        gridBag.setConstraints(scrollPane,constraints);
```

```
                frameLineaDeComandos.getContentPane().add(scrollPane);
                constraints.fill=GridBagConstraints.NONE;
                constraints.gridwidth=1;
                constraints.gridx=0;
                constraints.gridy=1;
                gridBag.setConstraints(labelLineaDeComandos,constraints);
                frameLineaDeComandos.getContentPane().add(labelLineaDeComandos);
                constraints.gridx=1;
                constraints.gridy=1;
                gridBag.setConstraints(botonCerrarLineaDeComandos,constraints);
                frameLineaDeComandos.getContentPane().add(botonCerrarLineaDeComandos);

                frameLineaDeComandos.setResizable(false);

                frameLineaDeComandos.pack();
        frameLineaDeComandos.setVisible(true);

        }else if(source==botonBouncyCastle){
                if(fc.showDialog(frame,"Seleccionar")==JFileChooser.APPROVE_OPTION){
                        txtBouncyCastle.setText(fc.getSelectedFile().getPath());
                }
        }else if(source==botonJSSE){
                if(fc.showDialog(frame,"Seleccionar")==JFileChooser.APPROVE_OPTION){
                        txtJSSE.setText(fc.getSelectedFile().getPath());
                }
        }else if(source==botonMastin){
                if(fc.showDialog(frame,"Seleccionar")==JFileChooser.APPROVE_OPTION){
                        txtMastin.setText(fc.getSelectedFile().getPath());
                }
        }else if(source==botonAbout){
                JOptionPane.showMessageDialog(frame,ABOUT,"Acerca de...",JOptionPane.INFORMATION_MESSAGE);

        }else if(source==botonCerrarLog){
                frameLog.dispose();
        }else if(source==botonCerrarFrame){
                System.exit(0);
        }else if(source==botonCerrarLineaDeComandos){
                frameLineaDeComandos.dispose();
        }
```

```
                        }catch(Exception e){
                                    JOptionPane.showMessageDialog(frame,e.toString(),"Excepción",JOptionPane.ERROR_MESSAGE);

                        }finally{   //El contenido de finally se ejecuta tanto si se ha capturado una Exception (catch) como si no
                                    //Volvemos a colocar el cursor por defecto
                                    frame.setCursor(new Cursor(Cursor.DEFAULT_CURSOR));
                        }
            }
}
```

instalarMastin.sh

```
# Copiamos recursivamente
cp --remove-destination -R Mastin_v1.0 /usr/share/Mastin_v1.0
cp --remove-destination Mastin_v1.0/bin/XMastin.sh /bin/
mkdir $HOME/Desktop/"XMastin v1.0"
cp --remove-destination "XMastin v1.0" $HOME/Desktop/"XMastin v1.0/"
cp --remove-destination ayuda $HOME/Desktop/"XMastin v1.0/"
cp --remove-destination "desinstalar Mastin v1.0 y XMastin v1.0" $HOME/Desktop/
cp -R --remove-destination crypto-115 /usr/share/crypto-115
```

instalarMastin.bat

```
@echo off
xcopy /e Mastin_v1.0 "c:\archivos de programa\Mastin_v1.0\"
copy "XMastin v1.0.lnk" "%allusersprofile%"\escritorio\
md "%allusersprofile%\Men£ Inicio\Programas\XMastin v1.0"
copy "XMastin v1.0.lnk" "%allusersprofile%\Men£ Inicio\Programas\XMastin v1.0"
copy "desinstalar Mastin v1.0 y XMastin v1.0.lnk" "%allusersprofile%\Men£ Inicio\Programas\XMastin v1.0"
copy ayuda.lnk "%allusersprofile%\Men£ Inicio\Programas\XMastin v1.0"
copy "XMastin v1.0.lnk" %windir%\escritorio\
md "%windir%\menu inicio\programas\XMastin v1.0"
```

```
copy "XMastin v1.0.bat" "%windir%\men£ inicio\programas\XMastin v1.0\"
copy "desinstalar Mastin v1.0 y XMastin v1.0.lnk" "%windir%\men£ inicio\programas\XMastin v1.0\"
copy ayuda.lnk "%windir%\men£ inicio\programas\XMastin v1.0\"
xcopy /e crypto-115 "c:\archivos de programa\crypto-115\"
copy .\Mastin_v1.0\bin\XMastin.bat "%windir%\"
```

desinstalarMastin.sh

```
rm -R /usr/share/Mastin_v1.0
rm -R /usr/share/crypto-115
rm /bin/XMastin.sh
rm -R $HOME/Desktop/"XMastin v1.0/"
rm $HOME/Desktop/"desinstalar Mastin v1.0 y XMastin v1.0"
```

desinstalarMastin.bat

```
@echo off
rd /s /q "c:\Archivos de programa\crypto-115"
del /q "%allusersprofile%\escritorio\XMastin v1.0.lnk"
rd /s /q "%allusersprofile%\Men£ Inicio\Programas\XMastin v1.0"
del /q "%windir%\escritorio\XMastin v1.0.lnk"
rd /s /q "%windir%\men£ inicio\programas\XMastin v1.0"
rd /s /q "c:\Archivos de programa\Mastin_v1.0"
```

index.html

```
<html>
  <head>
    <title></title>
    <meta content="">
    <style></style>
  </head>
```

```
<body>
      <center><h1>Manual de usuario<br>Mastin v1.0 y XMastin v1.0</h1></center>
         <hr>
         <h2><a name="indice">Índice</a></h2>
         <ul>
                     <li><a href="#indice">Índice</a>
                     <li><a href="#descripcion">Descripción</a>
                     <ul>
                                 <li><a href="#descMastin">Mastin v1.0</a>
                                 <li><a href="#descXMastin">XMastin v1.0</a>
                     </ul>
                     <li><a href="#software">Software necesario</a>
                     <li><a href="#instalacion">Instalación</a>
                     <ul>
                                 <li><a href="#instWindows">Usuarios de Windows</a>
                                 <li><a href="#instLinux">Usuarios de Linux</a>
                     </ul>
                     <li><a href="#keytool">Keytool</a>
                     <ul>
                                 <li><a href="#synopsis">Synopsis</a>
                                 <li><a href="#descripcion">Descripción</a>
                                 <li><a href="#opciones">Opciones</a>
                     </ul>
                     <li><a href="#uso">Uso</a>
                     <ul>
                                 <li><a href="#usoMastin">Mastin v1.0</a>
                                 <li><a href="#usoXMastin">XMastin v1.0</a>
                     </ul>
                     <li><a href="#desinstalacion">Desinstalación</a>
                     <ul>
                                 <li><a href="#desinstWindows">Usuarios de Windows</a>
                                 <li><a href="#desinstLinux">Usuarios de Linux</a>
                     </ul>
         </ul>

         <hr>
         <h2><a name="descripcion">Descripción</a></h2>
         <a name="descMastin"><h3>Mastin v1.0</h3></a>
```

```
            Aplicación de línea de comandos que encripta o desencripta ficheros a partir de la clave extraida de un KeyStore creado por la
herramienta <b><i>keytool</b></i> de Java
            <br>El algoritmo que se usa es el algoritmo asimétrico RSA inventado por Rivest, Adleman y Shamir y que viene recogido en el
PKCS#1.
            <br>El modo de funcionamiento será ECB (Electronic Code Book) y el relleno será OAEP (Optimal Asymmetric Encryption Padding)
definido en el PKCS#1 v2.
            <br>La clave, ya sea pública o privada, se extraerá de la entrada en un KeyStore definida por un alias
            <br>Se podrá especificar la ruta del KeyStore (<i>-KeyStorePath keyStorePath</i>), su tipo (<i>-KeyStoreType
keyStoreType</i>), y su clave (<i>-KeyStorePassword keyStorePassword</i>).
            <br>Si se desea utilizar un alias distinto a "mykey", se deberá especificar (<i>-Alias alias</i>) así como su clave (<i>-
KeyPassword keyPassword</i>) en caso a que sea distinta a la del KeyStore (<i>-KeyStoreType keyStoreType</i>)
            <br>Al estar escrito en Java, se puede ejecutar en cualquier Sistema Operativo, siendo testados tanto en Windows (XP) como en
Linux (SuSE Linux 8.0).
            <a name="descXMastin"><h3>XMastin v1.0</h3></a>
            Aplicación para un entorno gráfico que utiliza el API suministrado por Mastin para realizar las mismas funciones que éste.
            <br>Al estar escrito en Java, se puede ejecutar en cualquier Sistema Operativo, siendo testados tanto en Windows (XP) como en
Linux (SuSE Linux 8.0).
            <br>
            <br>Tanto Mastin como XMastin fueron creadas por Gabriel Babiano Huete (<a
href="mailto:gabrielbabiano@yahoo.es">gabrielbabiano@yahoo.es</a>) con motivo de su Proyecto Fin de Carrera de Ingeniería de
Telecomunicación en la <a href="http://www.esi.us.es">Escuela Superior de Ingenieros</a> de la <a href="http://www.us.es">Universidad de
Sevilla</a> en 2002.

            <hr>
            <h2><a name="software">Software necesario</a></h2>
            Tanto para la ejecución de Mastin como XMastin se necesitan tener el siguiente software correctamente instalado:
            <ul>
                        <li><a href="http://www.java.sun.com/j2se/1.4.1/download.html">Java 2 v1.4.0 JRE (Java Runtime Enviroment) o SDK
(Standard Development Kit) o posterior</a>
                        <li><a href="http://java.sun.com/products/jce/index-14.html">jce_policy-1_4_0.zip</a>
                        <li><a href="http://www.bouncycastle.org/latest_releases.html">JCE Provider 1.15 de Bouncy Castle</a> (se instala
automáticamente en la instalación de Mastin)
            </ul>
            No debemos olvidar que dado que XMastin se basa en el API que suministra Mastin, requerirá además que este último se encuentre
correctamente instalado

            <hr>
            <h2><a name="instalacion">Instalación</a></h2>
            <h3><a name="instWindows">Usuarios de Windows</a></h3>
```

```
            Para instalar Mastin v1.0 y XMastin v1.0 deberá de tener instalado previamente Java 2 v1.4.0 pudiendo elegir entre solamente
el JRE (Java Runtime Enviroment) o el SDK (Standard Development Kit) completo. Deberemos posteriormente instalar jce_policy-1_4_0.zip.
            <br>Posteriormente deberá ejecutar "instalarMastin.bat". Dicho batch copiará automáticamente:
            <ul>
                        <li>el directorio "Mastin_v1.0" recursivamente en "c:\archivos de programa\Mastin_v1.0\"
                        <li>el direcotrio "crypto-115" recursivamente en "c:\archivos de programa\crypto-115\"
                        <li>el batch "XMastin.bat" en %windir%
                        <li>el acceso directo "XMastin v1.0.lnk" en el escritorio
                        <li>el acceso directo "XMastin v1.0.lnk" en el directorio XMastin v1.0 creado en "Menú de inicio|Programas" ("Menú
de inicio|Programas|XMastin v1.0")
                        <li>el acceso directo "ayuda.lnk" en "Menú de inicio|Programas|XMastin v1.0"
                        <li>el acceso directo "desinstalar Mastin v1.0 y XMastin v1.0"  en "Menú de inicio|Programas"
            </ul>
            En esta instalación también se está instalando el proveedor de JCE suministrado por <a
href="http://www.bouncycastle.org">Bouncy Castle</a>
            <h3><a name="instLinux">Usuarios de Linux</a></h3>
            Para instalar Mastin v1.0 y XMastin v1.0 deberá de tener instalado previamente Java 2 v1.4.0 pudiendo elegir entre solamente
el JRE (Java Runtime Enviroment) o el SDK (Standard Development Kit) completo. Deperemos posteriormente instalar jce_policy-1_4_0.zip
            <br>Posteriormente deberá ejecutar "sh ./instalarMastin.sh". Dicho script copiará automáticamente:
            <ul>
                        <li>el directorio "Mastin_v1.0" recursivamente en "/usr/share/Mastin_v1.0"
                        <li>el directorio "crypto-115" recursivamente en "/usr/share/crypto-115"
                        <li>el script Mastin_v1.0/bin/XMastin.sh en /bin/
                        <li>el acceso directo "XMastin v1.0" en el escritorio "$HOME/Desktop/". Dicho acceso directo apunta al script
"/bin/XMastin.sh"
                        <li>el acceso directo "desinstalar Mastin v1.0 y XMastin v1.0" en el escritorio "$HOME/Desktop/". Dicho acceso
directo apunta al script "/usr/share/Mastin_v1.0/bin/desinstalarMastin.sh"
            </ul>
            En esta instalación también se está instalando el proveedor de JCE suministrado por <a
href="http://www.bouncycastle.org">Bouncy Castle</a>

            <hr>
            <h2><a name="keytool">Keytool</a></h2>
                        <h3><a name="synopsis">Sysnopsis</a></h3>
                                    <table BGCOLOR="#e6e6ff">
                                                <td><code>
                                                            <b>keytool [comandos]</b>
                                                </td></code>
                                    </table>
```

```
                        <h3><a name="descripcion">Descripción</a></h3>
                                    <p>keytool es la herramienta de gestión de certificados y claves de Java. Gestiona una base de datos
(keystore) de claves privadas y certificados X.509 con claves públicas autenticadas por entidades de confianza. El keystore se protege por
una contraseña.</p>
                                    <p><h4>Entradas</h4>
                                                <ul>Existen dos tipos de entradas en un keystore:
                                                            <li><b><i>keyEntry</i></b>:         almacena entre otras cosas la clave privada
por lo que se almacena de forma protegida con otra contraseña para prevenir accesos no autorizados.
                                                            <li><b><i>trustedCertEntry</i></b>: contiene una clave pública certificada. Se
denomina ?trusted? (?de confianza?) porque el dueño del keystore confía que la clave pública del certificado pertenece realmente al
?subject? (?sujeto?) dueño del certificado. El expendedor del certificado se responsabiliza de esto firmando el certificado.
                                                </ul>
                                    </p>
                                    <p><h4>Alias</h4>
                                                A todas las entradas del keystore se accede mediante un único alias. Los alias no
distinguen mayúsculas de minúsculas.
                                                <br>El alias se especifica cuando se añade una entrada al keystore usando -genkey para
generar un par de claves pública/privada o cuando se añade un certificado mediante el comando -import a la lista de certificados de
confianza. Posteriores llamadas a keytool usarán el mismo alias para referirse a dicha entrada.
                                    </p>

                                    <p><h4>Localización del KeyStore</h4>
                                                Cada comando keytool tiene la opción -keystore para especificar el nombre y localización
del archivo persistente que contiene el keystore. Este archivo se almacena por defecto en un fichero llamado .keystore situado en el
directorio del usuario determinado por la propiedad del sistema user.home.
                                    <p>
                                    <p><h4>Creación del KeyStore</h4>
                                                Se crea un KeyStore cuando se usa cualquiera de los comandos -genkey, -import o
-identitydb para añadir datos a un keystore que no existe aún.
                                                <br>Más específicamente, si no se especifica la opción -keystore y no existe aún un
keystore, se creará uno llamado .keystore en el directorio determinado por user.home.
                                    </p>
                                    <p><h4>Implementación del KeyStore</h4>
                                                La clase KeyStore del paquete java.security proporciona un interfaz bien definido para
acceder y modificar la información contenida en un keystore. Se peden disponer de distintas implementaciones, cada una de un tipo
determinado.
                                                <br>Existe una implementación propiedad de Sun Microsystems denominada JKS (Java
KeyStore). Protege cada clave privada con una contraseña individual y también protege la integridad del keystore entero con otra
contraseña (que puede ser igual a la anterior).
```

```
                                        <br>Por defecto se utiliza el tipo de keystore especificado explícitamente en la propiedad
keystore.type en el archivo de propiedades de seguridad (normalmente es JKS). Dicho archivo se llama java.security y reside en
java.home\lib\security donde java.home es el directorio del JRE.
                                </p>
                                <p><h4>Algoritmos soportados y tamaños de claves</h4>
                                        keytool permite a los usuarios especificar el algoritmo de creación de pares de claves o
de firma de cualquier proveedor de servicios criptográficos registrado. Es decir, las opciones -keyalg y -sigalg deben ser soportadas por
la implementación del proveedor.
                                </p>
                                <p><h4>Generando pares de claves</h4>
                                        Para generar un par de claves pública/privada, usaremos el comando -genkey de la forma:
                                        <table BGCOLOR="#e6e6ff">
                                                <td><code>
                                                        gab:~ # <b>keytool -genkey -keyalg RSA -keysize 2048</b>
                                                        <br>Enter keystore password:  <b>miContrasena</b>
                                                        <br>What is your first and last name?
                                                        <br>  [Unknown]:  <b>Gabriel Babiano Huete</b>
                                                        <br>What is the name of your organizational unit?
                                                        <br>  [Unknown]:  <b>Departamento de Telematica</b>
                                                        <br>What is the name of your organization?
                                                        <br>  [Unknown]:  <b>Universidad de Sevilla</b>
                                                        <br>What is the name of your City or Locality?
                                                        <br>  [Unknown]:  <b>Sevilla</b>
                                                        <br>What is the name of your State or Province?
                                                        <br>  [Unknown]:  <b>Spain</b>
                                                        <br>What is the two-letter country code for this unit?
                                                        <br>  [Unknown]:  <b>ES</b>
                                                        <br>Is <CN=Gabriel Babiano Huete, OU=Departamento de Telematica,
O=Universidad de Sevilla, L=Sevilla, ST=Spain, C=ES> correct?
                                                        <br>  [no]:  <b>y</b>
                                                        <br>
                                                        <br>Enter key password for <mykey>
                                                        <br>(RETURN if same as keystore password):
                                                </td></code>
                                        </table>
                                        <br>Crea una pareja de claves pública/privada para el algoritmo RSA de tamaño 2048 bits en
el alias por defecto mykey en el keystore por defecto .keystore situado el el directorio casa del usuario.
                                </p>
                                <p><h4>Generando una CSR</h4>
```

```
                                                Para crear una Petición de Firmado de Certificado (CSR) usando el formato PKCS#10 se usará
el comando -certreq. Con el siguiente ejemplo, generamos una CSR de la clave pública contenida en la entrada referenciada por el alias
mykey y lo guardamos en el fichero csr.pem:
                                                <table BGCOLOR="#e6e6ff">
                                                            <td><code>
                                                                        <b>keytool -certreq -alias mykey -file csr.pem</b>
                                                            </td></code>
                                                </table>
                                                Posteriormente, este CSR se podrá remitir a una CA, y ésta, nos remitirá dicha petición ya
firmada en un certificado una vez que haya comprobado la identidad del solicitante.

                                    </p>
                                    <p><h4>Importando certificados</h4></p>
                                                Para importar certificados desde un archivo usaremos el comando -import de la siguiente
manera:
                                                <table BGCOLOR="#e6e6ff">
                                                            <td><code>
                                                                        <b>keytool -import -alias cert -file certfile.cer</b>
                                                            </td></code>
                                                </table>
                                                que importa el certificado del archivo certfile.cer y lo guarda el en keystore en la
entrada identificado por el alias cert.
                                                <ul>Se puede importar un certificado por dos posibles motivos:
                                                            <li>para añadirlo a la lista de certificados de confianza
                                                            <li>para importar un certificado respuesta de una CA com resultado de enviarle
una CSR (obtenida mediante -certreq) a dicha CA.
                                                </ul>
                                                <ul>Dependiendo del tipo de entrada que sea la apuntada por el alias:
                                                            <li>Si el alias apunta a una keyEntry, entonces keytool asume que se está
importando un certificado respuesta. keytool comprueba si la clave pública del certificado coincide con la guardada bajo el alias y se
sale si son diferentes.
                                                            <li>Si el alias no apunta a un keyEntry, entonces keytool asume que se está
añadiendo una entrada de certificado de confianza. En este caso, el alias no debería existir en el keystore. Si no fuera así, keytool
mostraría un error y no importaría el certificado. Si el alias no existiese en el keystore, keytool crearía una entrada de certificado
confiado con el alias especificado y lo asociaría con el certificado importado.
                                                </ul>
                                                keytool puede importar certificados X.509 v1, v2 y v3 y cadenas de certificados en formato
PKCS#7 consistentes en certificados de ese tipo. Los datos para ser importados deben ser provistos ya sea en formato codificado binario o
en formato codificado imprimible (también conocido como  Base64) definido en el standard RFC 1421. En el último caso, la codificación debe
estar limitada por una cadena que empiece por -----BEGIN al principio y -----END al final.
```

```
                                    </p>
                                    <p><h4>Exportando certificados</h4>
                                                Para exportar un certificado a un archivo se usa el comando -export de la forma:
                                                <table BGCOLOR="#e6e6ff">
                                                            <td><code>
                                                                        <b>keytool -export -alias cert -file certfile.cer</b>
                                                            </td></code>
                                                </table>
                                                Este ejemplo exporta el certificado cert al archivo certfile.cer. Esto es, si el alias
cert corresponde a un keyEntry, se exporta el certificado que autentica la clave pública de jane. Si por el contrario, cert es el alias de
un trustedCertEntry, entonces se exporta el certificado confiado.
                                    <p>
                                    <p><h4>Mostrando certificados</h4>
                                                Para mostrar el contenido de un keystore, se usa el comando -list, de la forma:
                                                <table BGCOLOR="#e6e6ff">
                                                            <td><code>
                                                                        <b>keytool -list</b>
                                                            </td></code>
                                                </table>
                                                o de la forma
                                                <table BGCOLOR="#e6e6ff">
                                                            <td><code>
                                                                        <b>keytool -list -alias mykey</b>
                                                            </td></code>
                                                </table>
                                                para mostrar el contenido de la entrada del keystore referida por mykey
                                                <br>Para mostrar el contenido de un certificado guardado en un archivo, se usa el comando
-printcert de la forma:
                                                <table BGCOLOR="#e6e6ff">
                                                            <td><code>
                                                                        <b>keytool -printcert -file certfile.cer</b>
                                                            </td></code>
                                                </table>
                                                que muestra la información contenida en el archivo certfile.cer. Para esto último no se
necesita un keystore.
                                    </p>
                                    <p><h4>Borrando entradas</h4>
                                                Para borrar entradas, ya sean keyEntry o trustedCertEntry, lo podemos hacer con el comando
-delete. Con el siguiente ejemplo borramos la entrada asociada al alias mykey:
```

```
                                        <table BGCOLOR="#e6e6ff">
                                                <td><code>
                                                        <b>keytool -delete -alias mykey</b>
                                                </td></code>
                                        </table>
                                </p>

                        <h3><a name="opciones">Opciones</a></h3>
                                <p>Opciones que aparecen para la mayoría de los comandos:
                                </p>
                                <TABLE WIDTH=643 BORDER=1 BORDERCOLOR="#000000" CELLPADDING=4 CELLSPACING=0 STYLE="page-break-after:
auto">
                                        <COL WIDTH=192>
                                        <COL WIDTH=433>
                                        <THEAD>
                                                <TR VALIGN=TOP>
                                                        <TH WIDTH=192 BGCOLOR="#9999ff">
                                                                <P CLASS="western">Opcion</P>
                                                        </TH>
                                                        <TH WIDTH=433 BGCOLOR="#9999ff">
                                                                <P CLASS="western">Descripci&oacute;n</P>
                                                        </TH>
                                                </TR>
                                        </THEAD>
                                        <TBODY>
                                                <TR VALIGN=TOP>
                                                        <TD WIDTH=192 BGCOLOR="#ccccff">
                                                                <P CLASS="courier-tabla-western">-v</P>
                                                        </TD>
                                                        <TD WIDTH=433>
                                                                <P>&ldquo;Verbose&rdquo;. Se mostrar&aacute;
informaci&oacute;n

                                                                detallada</P>
                                                        </TD>
                                                </TR>
                                                <TR VALIGN=TOP>
                                                <TD WIDTH=192 BGCOLOR="#ccccff">
                                                                <P CLASS="courier-tabla-western">-help</P>
                                                        </TD>
```

```
                                                            <TD WIDTH=433>
                                                                        <P>Se mostrar&aacute; la ayuda</P>
                                                            </TD>
                                                </TR>
                                                <TR VALIGN=TOP>
                                                            <TD WIDTH=192 BGCOLOR="#ccccff">
                                                                        <P CLASS="courier-tabla-western">-storetype
<I>storetype</I></P>
                                                            </TD>
                                                            <TD WIDTH=433>
                                                                        <P>Especifica el tipo de keystore</P>
                                                            </TD>
                                                </TR>
                                                <TR VALIGN=TOP>
                                                            <TD WIDTH=192 BGCOLOR="#ccccff">
                                                                        <P CLASS="courier-tabla-western">-keystore
<I>keystore</I></P>
                                                            </TD>
                                                            <TD WIDTH=433>
                                                                        <P>Localizaci&oacute;n del keystore</P>
                                                            </TD>
                                                </TR>
                                                <TR VALIGN=TOP>
                                                            <TD WIDTH=192 BGCOLOR="#ccccff">
                                                                        <P CLASS="courier-tabla-western">-storepass
<I>storepass</I></P>
                                                            </TD>
                                                            <TD WIDTH=433>
                                                                        <P>Contrase&ntilde;a utilizada para proteger la
integridad del
                                                                        keystore.
                                                                        </P>
                                                                        <P>storepass debe tener al menos 6 d&iacute;gitos.</P>
                                                                        <P>Si no se especifica en la l&iacute;nea de comandos,
se
                                                                        preguntar&aacute; por ella.</P>
                                                            </TD>
                                                </TR>
                                                <TR VALIGN=TOP>
```

```
                                                            <TD WIDTH=192 BGCOLOR="#ccccff">
                                                                        <P CLASS="courier-tabla-western">-provider
<I>provider</I></P>
                                                            </TD>
                                                            <TD WIDTH=433>
                                                                        <P>Nombre de la clase del proveedor del servicio
criptogr&aacute;fico</P>
                                                            </TD>
                                                      </TR>
                                                </TBODY>
                                    </TABLE>
                                    <h4>Opciones por defecto</h4>
                                                <table BGCOLOR="#e6e6ff">
                                                      <td><code>
                                                            <b>-alias mykey
                                                            <br>-keyalg DSA
                                                            <br>-keysize 1024
                                                            <br>-validity 90
                                                            <br>-keystore
el_archivo_de_nombre_.keystore_en_el_directorio_casa_del_usuario
                                                            <br>-file stdin_si_se_lee_o_stdout_si_se_escribe
                                                            </b>
                                                      </td></code>
                                                </table>
                                                <ul>El algoritmo de firma (opción -sigalg) se deriva del algoritmo de la clave privada:
                                                      <li>si es DSA, entonces la firma es SHA1
                                                      <li>si es RSA, entonces la firma es MD5
                                                </ul>
                                                Para más información acerca de keytool, se puede consultar la documentación suministrada
por Java.

            <hr>
            <h2><a name="uso">Uso</a></h2>

            <h3><a name="usoMastin">Mastin v1.0</a></h3>
            Una vez realizada la <a href="#instalacion">instalación</a>, podemos proceder a ejecutarlo por línea de comandos:
            <br>
            <br><b><i>    Mastin [opciones] -e/-d -in inFile -out outFile</b></i>
            <br>                  <ul><li><i>-e</i>:     encripta
            <br>                  <li><i>-d</i>:     desencripta
```

```
           <br>                <li><i>-in inFile</i>:     inFile es la ruta del fichero de entrada
           <br>                <li><i>-out outFile</i>:   outFile es la ruta del fichero de salida</ul>
           <br>        <b>Lista de opciones:</b>
           <br>                <ul><li><i>-about</i>:         muestra información acerca de la aplicación
           <br>                <li><i>-Alias alias</i>:   alias del la entrada en la KeyStore de donde se extraerá la clave privada. Por
defecto se usará "mykey"
           <br>                <li><i>-KeyPassword keyPassword</i>:       especifica la contraseña de la clave del alias. Por defecto se
usará la misma clave que para el KeyStore (<i>-KeyStorePassword KeyStorePassword</i>)
           <br>                <li><i>-KeyStorePassword KeyStorePassword</i>:     especifica la contraseña de entrada al KeyStore
           <br>                <li><i>-KeyStorePath KeyStorePath</i>:     especifica la ruta del fichero KeyStore
           <br>                <li><i>-KeyStoreType KeyStoreType</i>:     especifica el tipo de KeyStore. Por defecto se usará "JKS"
           <br>                <li><i>-v</i>:             muestra información del desarrollo de la aplicación (equivalente a -verbose)
           <br>                <li><i>-verbose</i>:       muestra información del desarrollo de la aplicación
           <br>                <li><i>-version</i>:       muestra información de la versión de la aplicación</ul>

           <br><h4>Ejemplos:</h4>
           <br><h5>Para encriptar:</h5>
           <table BGCOLOR="#e6e6ff">
                       <td><code>
                                   <b>java -cp /usr/java/crypto-115/jars/bcprov-jdk14-
115.jar:/usr/java/j2sdk1.4.0_02/jre/lib/jsse.jar:./jars/ es.us.esi.bab.mastin.Mastin -e -in kk.cer -out kk.crypt  -keystorepassword
miclave -alias cert -v</b>
                       </td></code>
           </table>
           <br><h5>Para desencriptar:</h5>
           <table BGCOLOR="#e6e6ff">
                       <td><code>
                                   <b>java -cp /usr/java/crypto-115/jars/bcprov-jdk14-
115.jar:/usr/java/j2sdk1.4.0_02/jre/lib/jsse.jar:./jars/ es.us.esi.bab.mastin.Mastin -d -in kk.crypt -out kk2.cer -alias mykey
-keystorepassword miclave -v</b>
                       </td></code>
           </table>

           <br>
           <br>
           <br>Se recomienda que en caso de disponer de entorno gráfico, se ejecute XMastin ya que facilita el paso de parámetros e
incluso dispone de un generador de líneas de comando equivalente

           <h3><a name="usoXMastin">XMastin v1.0</a></h3>
           Una vez realizada la <a href="#instalacion">instalación</a>, podemos proceder a ejecutarlo:
```

```
            <ul>
                        <li>en Windows podemos proceder haciendo doble click sobre los accesos directos a "XMastin v1.0" situados en el
escritorio o bien en "Menú de inicio|Programas|XMastin v1.0" o bien ejecutando "XMastin" en la línea de comandos
                        <li>en Linux podemos entrarnos en la carpeta "XMastin v1.0" situada en el escritorio y una vez abierta, hacer
doble click en el acceso directo a "XMastin v1.0" o bien en ejecutando "XMastin" en el shell
            </ul>
            <br>Una vez abierto, el programa, nos encontraremos con:
            <center><img src="images/1.jpeg" align="rigth"></center>
            <br>Nos encontramos con dos pestañas: "General" y "Línea de comandos". En la pestaña "General", seleccionamos la acción que
queremos realizar: encriptar o desencriptar. En este ejemplo vamos a encriptarlo, pero para desencriptarlo, se procedería de forma similar
            <center><img src="images/2.jpeg"></center>
            <br>Para especificar el fichero de entrada, tenemos dos opciones: escribirlo o seleccionarlo. Si deseamos seleccionarlo,
pulsaremos el botón de la derecha:
            <center><img src="images/3.jpeg"></center>
            <br>Tras lo cual saldrá un diálogo similar a éste:
            <center><img src="images/4.jpeg"></center>
            <br>Seleccionaremos el archivo que deseemos y pulsaremos "Seleccionar" para seleccionarlo. Podemos pulsar cancelar en caso
contrario:
            <center><img src="images/5.jpeg"></center>
            <br>Obraremos de la misma manera con el fichero de salida. Si el archivo ya existiese, se sobreescribirá
            <center><img src="images/6.jpeg"></center>
            <br>Si el KeyStore que se muestra es el adecuado (Mastín pone el KeyStore creado por defecto por <i>keytool</i>), introducimos
La contraseña del mismo
            <center><img src="images/7.jpeg"></center>
            <br>Cambiamos al alias que contiene la contraseña en el KeyStore con la que deseamos trabajar (Mastín pone por defecto el del
alias que crea por defecto <i>keytool</i>). Si el alias se refiere a una entrada de tipo Key en el KeyStore, encriptaremos cun la clave
privada. Si por el contrario, el alias se refiere a una entrada de Certificado en el Keytool, encriptaremos lógicamente con la clave
pública
            <center><img src="images/8.jpeg"></center>
            <br>Escribimos la contraseña de dicho alias en caso de que sea distinta a la del KeyStore
            <center><img src="images/9.jpeg"></center>
            <br>Pulsamos el botón de "Go!" situado en la parte inferior cuando ya hallamos  introducidos los datos para que la aplicación
se ponga a trabajar
            <center><img src="images/10.jpeg"></center>
            <br>Si los datos son correctos y hemos introducidos todos los necesarios, al cabo de algunos segundos (que dependen del tamaño
del archivo de entrada y de si la clave es pública o privada) aparecerá una ventana que muestra el contenido del archivo "MastinLog.txt"
situado en el mismo directorio de la aplicación; en el que podemos ver el resultado de nuestra acción.
            <center><img src="images/11.jpeg"></center>
            <br>Para cerrarlo, pulsaremos el botón "Cerrar" situado en la parte inferior.
            <center><img src="images/12.jpeg"></center>
```

```
            <br>En la pestaña de "Línea de comandos" podemos cambiar las opciones para posteriormente obtener la línea de comandos
equivalente a los datos que le hemos introducido en este formulario.
            <center><img src="images/13.jpeg"></center>
            <br>El el recuadro de opciones, podemos cambiar el verbose (si deseamos o no, que Mastín nos muestre los comentarios por sa
salida standard)
            <center><img src="images/14.jpeg"></center>
            <br>Si deseamos o no, que nos muestre la versión
            <center><img src="images/15.jpeg"></center>
            <br>O el "Acerca de" la aplicación Mastín
            <center><img src="images/16.jpeg"></center>
            <br>También podremos cambiar las rutas que se muestran por defecto, tanto del ejecutable <i>java</i> como de los distintos
paquetes necesarios para que se ejecute correctamante Mastín
            <center><img src="images/17.jpeg"></center>
            <br>Por último, pulsamos el botón "Línea de comandos" situado en la parte inferior.
            <center><img src="images/19.jpeg"></center>
            <br>Nos aparecerá una ventana cuyo contenido será la línea de comandos equivalente al formulario que hemos rellenado en caso
de que quisiésemos ejecutar Mastin en línea de comandos
            <center><img src="images/20.jpeg"></center>
            <br>Podemos seleccionar el texto, y copiarlo al portapaeles pulsando Control+C.
            <center><img src="images/21.jpeg"></center>
            <br>Podremos cerrar esta ventana pulsando el botón "Cerrar" situado en la parte inferior
            <center><img src="images/22.jpeg"></center>
            <br>Podemos ver el "Acerca de" de XMastin pulsando el botón "Acerca de" situado en la parte inferior de la ventana principal
            <center><img src="images/23.jpeg"></center>
            <br>Cuyo resultado es el siguiente
            <center><img src="images/24.jpeg"></center>
            <br>Pulsamos aceptar para cerrar dicha ventana
            <center><img src="images/25.jpeg"></center>
            <br>Si deseamos salir del programa XMastin, podemos pulsar en cualquier momento (excepto en el momento en que está trabajando)
el botón "Cerrar" situado en la parte inferior derecha de la ventana principal.
            <center><img src="images/26.jpeg"></center>

            <hr>
            <h2><a name="desinstalacion">Desinstalación</a></h2>
            <h3><a name="desinstWindows">Usuarios de Windows</h3>
            La desinstalación es incluso más sencilla que la instalación. Tan sólo hay que hacer click en "desinstalar Mastin v1.0 y
XMastin v1.0" dentro de "Menú de inicio|Programas|XMastin v1.0"
            <h3><a name="desinstLinux">Usuarios de Linux</h3>
            La desinstalación es incluso más sencilla que la instalación. Tan sólo hay que hacer click en "desinstalar Mastin v1.0 y
XMastin v1.0" dentro del escritorio
```

```
  </body>
</html>
```

Autoridad de Certificación (CA) v1.0

index.html

```
<html>
  <head>
    <title>Autoridad de Certificación: Página principal</title>
    <meta content="">
    <style></style>
  </head>
  <body>
        <table>
                <th colspan=2 bgcolor="#9591f4">
                        <center><font size=+8><b><img src="logo_us.gif" height=100 width=150 align="middle">Autoridad de Certificación
(CA)<img src="esi.gif" height=75 width=75 align="middle"></font></b></center>
                </th>
                <tr>
                        <td bgcolor=#FFF6D5>Bienvenido a esta Autoridad de Certificación.
                                <ul>En esta Autoridad de Certificación podrá:
                                        <li><a href="/CA/cert_1.html">Obtener su Certificado</a> a partir de una Petición de Firma de
Certificado (CSR). Podrá obtenerun certificado por dirección de correo electrónico (email). Esta autoridad admitirá sólo un certificado
válido por dirección de correo electrónico, certificando tan sólo que dicho certificado pertenece a una determinada dirección de correo
electrónico. <b>Gratis!</b>
                                        <li><a href="/CA/cert_1.html">Revocar el Certificado</a> firmado por esta Autoridad de
Certificación. Sólo podrá revocar un Certificado si es considerado como válido actualmente. <b>Gratis!</b>
                                        <li><a href="/CA/descargar.html">Buscar</a> el certificado de otras personas a partir de la
dirección de correo electrónico (email). <b>Gratis!</b>
                                        <li><a href="/CA/descargar.html">Descargar el certificado de la CA</a>. <b>Gratis!</b>
                                                <li><a href="/CA/descargar.html">Descargar todos los certificados válidos</a>
firmados por esta CA junto con la CRL en formato PKCS#7. <b>Gratis!</b>
                                                <li><a href="/CA/descargar.html">Obtener la Lista de Certificados Revocados
(CRL)</a> por esta CA. <b>Gratis!</b>

                                </ul>
                        <td bgcolor=#E6E6FF width="40%">
                                <h3>
```

```
                                        <h2>Página Principal</h2><h3>
                                                <br><br><a href="/CA/cert_1.html">Gestionar su certificado (firma o revocación)</a>
                                                <br><br><a href="/CA/descargar.html">Descargar certificados</a>
                                        </h3>

                </tr>
                <tr>
                        <td  colspan=2>
                        <hr>
                        <font size=-1>Creado por <a href="mailto:GabrielBabiano@yahoo.es">Gabriel Babiano Huete</a> en 2002 para su
Proyecto Fin de Carrera titulado: <i>"Estudio e implementación de una Autoridad de Certificación"</i>.
                                <br><a href="http://trajano.us.es">Área de Ingeniería Telemática</a> perteneciente al <a
href="http://www.esi.us.es/ISA"> Departamento de Ingeniería de Sistemas y Automática</a> en la <a href="http://www.esi.us.es">Escuela
Superior de Ingenieros</a> de la <a href="http://www.us.es"> Universidad de Sevilla</a>.
                        </font>
                </tr>

        </table>

  </body>
</html>
cert_1.html
<html>
  <head>
    <title>Autoridad de Certificación: Gestión de su certificado</title>
    <meta content="">
    <style></style>
  </head>
  <body>
        <table>
                <th colspan=2 bgcolor="#9591f4">
                        <center><font size=+8><b><img src="/CA/logo_us.gif" height=100 width=150 align="middle">Autoridad de Certificación
(CA)<img src="/CA/esi.gif" height=75 width=75 align="middle"></font></b></center>
                </th>
                <tr>
                        <td bgcolor=#FFF6D5>
                                En esta página podrá gestionar su certificado:
                                <ul>
                                        <li><b>Firmar una Petición de Firma de Certificado (CSR)</b> en caso de que a su dirección de
correo electrónico no le corresponda ningún certificado válido firmado por esta Autoridad de Cartificación (CA).
```

```
                                        <li><b>Revocar un Certificado</b> considerado como válido actualmente por esta Autoridad de
Certificación (CA).
                                </ul>
                                <br><form action="/examples/servlet/ObtenerEmail" method="POST">Para ello, si es tan amable,
introduzca:<br><b>su email: </b><input name="email"><input type="submit" value="Enviar"></form>
                                <br>En breves momentos enviaremos una clave a dicha dirección que necesitará posteriormente.
                                <br><br><b>NOTA IMPORTANTE:</b> en caso de que se repita la misma dirección de email de alguna ya
introducida, se sobreescribirá la antigua clave por lo que <b>la clave válida será la última recibida</b>.

                        <td bgcolor=#E6E6FF width="40%">
                                <h3>
                                        <a href="/CA/index.html">Página Principal</a>
                                                <h2><br><br>Gestionar su certificado (firma o revocación)</h2><h3>
                                                <br><br><a href="/CA/descargar.html">Descargar certificados</a>
                                    </h3>

                </tr>
                <tr>
                        <td  colspan=2>
                        <hr>
                        <font size=-1>Creado por <a href="mailto:GabrielBabiano@yahoo.es">Gabriel Babiano Huete</a> en 2002 para su
Proyecto Fin de Carrera titulado: <i>"Estudio e implementación de una Autoridad de Certificación"</i>.
                                    <br><a href="http://trajano.us.es">Área de Ingeniería Telemática</a> perteneciente al <a
href="http://www.esi.us.es/ISA"> Departamento de Ingeniería de Sistemas y Automática</a> en la <a href="http://www.esi.us.es">Escuela
Superior de Ingenieros</a> de la <a href="http://www.us.es"> Universidad de Sevilla</a>.
                        </font>
                </tr>

        </table>

  </body>
</html>
```

descargar.html

```
<html>
  <head>
```

```
    <title>Autoridad de Certificación: Obtener Certificado de esta CA</title>
    <meta content="">
    <style></style>
  </head>
  <body>
        <table>
                <th colspan=2 bgcolor="#9591f4">
                        <center><font size=+8><b><img src="/CA/logo_us.gif" height=100 width=150 align="middle">Autoridad de Certificación
(CA)<img src="/CA/esi.gif" height=75 width=75 align="middle"></font></b></center>
                </th>
                <tr>
                        <td bgcolor=#FFF6D5>
                                <form action="/examples/servlet/Descargar" method="POST">
                                        <fieldset>
                                                <legend align="left"><b>Sujeto:</b></legend>
                                                <br><input type="radio" name="sujeto" value="buscar" checked><label
for="buscar">Perteneciente al email:<input name="email"></LABEL>
                                                <br><input type="radio" name="sujeto" value="CA" ><label for="CA">Sólo el certificado de
la Autoridad Certificadora (CA)</LABEL>
                                                <br><input type="radio" name="sujeto" value="CRL"><label for="CRL">Sólo la Lista de
Certificados Revocados (CRL)</LABEL>
                                                <br><input type="radio" name="sujeto" value="all"><label for="all">Todos los certificados
válidos expedidos por esta CA y la CRL (sólo de tipo PKCS#7)</LABEL>
                                        </fieldset>
                                        <fieldset>
                                                <legend align="left"><b>Tipo de Certificado:</b></legend>
                                                <br><input type="radio" name="tipo" value="X509" checked><label for="X509">X.509</LABEL>
                                                <br><input type="radio" name="tipo" value="PKCS7"><label for="PKCS7">PKCS#7</LABEL>
                                        </fieldset>
                                        <fieldset>
                                                <legend align="left"><b>Formato del Certificado:</b></legend>
                                                <br><input type="radio" name="formato" value="PEM" checked><label for="PEM">PEM</LABEL>
                                                <br><input type="radio" name="formato" value="DER"><label for="DER">DER</LABEL>
                                        </fieldset>

                                                                <input type="radio" checked name="archivo" value="si"><label for="archivo">En
archivo</label>
                                        <input type="radio" name="archivo" value="no"><label for="archivo">Directamente al
navegador</label>
```

```
                                                    <center><input type="submit" value="Descargar"></center>
                                            </form>
                                            <br><br>En caso de que se descargue un archivo, se recomienda que al guardarlo, se
utilicen las extensiones:
                                <ul>
                                    <li><b>.cer</b>, <b>.crt</b> o <b>.cert</b> para certificados X.509
                                                    <li><b>.p7b</b> para PKCS#7
                                                    <li><b>.crl</b> para CRL X.509
                                            </ul>
                        <td bgcolor=#E6E6FF width="40%">
                                <h3>
                                        <a href="/CA/index.html">Página Principal</a>
                                        <br><br><a href="/CA/cert_1.html">Gestionar su certificado (firma o revocación)</a>
                                                <br><br><h2>Descargar certificados</h2><h3>
                                    </h3>

                </tr>
                <tr>
                        <td  colspan=2>
                        <hr>
                        <font size=-1>Creado por <a href="mailto:GabrielBabiano@yahoo.es">Gabriel Babiano Huete</a> en 2002 para su
Proyecto Fin de Carrera titulado: <i>"Estudio e implementación de una Autoridad de Certificación"</i>.
                                <br><a href="http://trajano.us.es">Área de Ingeniería Telemática</a> perteneciente al <a
href="http://www.esi.us.es/ISA"> Departamento de Ingeniería de Sistemas y Automática</a> en la <a href="http://www.esi.us.es">Escuela
Superior de Ingenieros</a> de la <a href="http://www.us.es"> Universidad de Sevilla</a>.
                        </font>
                </tr>

        </table>

  </body>
</html>
```

CAException.java

```
import java.lang.*;
```

```
/**
 * Excepción de la Autoridad de Certificación (CA)
 *
 * @author Gabriel Babiano Huete (<a hfer="mailto:gabrielbabiano@yahoo.es">gabrielbabiano@yahoo.es</a>)
 * @version 1.0
 */

public class CAException extends Exception{

            /**
             * Constructor de CAException
             *
             * @param s Mensaje de la excepción
             * @since CAException v1.0
             */
            public CAException(String s){
                        super(s);
            }

}
```

Descargar.java

```
//javac -classpath /opt/jakarta/common/lib/servlet.jar:/opt/jakarta/webapps/examples/WEB-INF/classes/ BuscarCertificado.java


import java.io.*;
import java.util.*;
import java.security.*;

//Incluir en el CLASSPATH /opt/jakarta/common/lib/servlet.jar
import javax.servlet.*;
import javax.servlet.http.*;

//Incluir en el CLASSPATH /opt/jakarta/webapps/examples/WEB-INF/classes/
```

```
/**
 * HttpServlet que permite descargar un certificado con un tipo MIME adecuado
 *
 * @author Gabriel Babiano Huete (<a hfer="mailto:gabrielbabiano@yahoo.es">gabrielbabiano@yahoo.es</a>)
 * @version 1.0
 */

public class Descargar extends HttpServlet {


           /**
            * Sobreescribe el método doPost(HttpServletRequest req, HttpServletResponse res) de la clase HttpServlet.
            * Llamado por el servidor para permitir al servlet manejar ena petición POST. De esta manera permite al cliente mandar datos
de longitud ilimitada al servidor web de una vez
            *
            * @param req Objeto HttpServletRequest que contiene la petición que el cliente ha hecho al servidor
            * @param resp Objeto HttpServletResponse que contiene la respuesta que el servlet envía al cliente
            * @throws IOException Cuando la petición para el POST no se puede manejar
            * @throws ServletException Si se detecta un error de entrada o salida cuando el servlet maneja la petición
            * @since Descargar v1.0
            */

           protected void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException{

                      try{

                                 //Obtenemos los datos a partir de la petición
                                 String sujeto=(req.getParameter("sujeto")).trim();         //Sujeto del certificado
                                 String email=(req.getParameter("email")).trim();           //dirección de correo electrónico del
sujeto (en caso en que el sujeto sea "buscar")
                                 String tipo=(req.getParameter("tipo")).trim();  //Tipo de certificado. Puede tomar los valores "X509" o
"PKCS7"
                                 String formato=(req.getParameter("formato")).trim();       //Formato de codificación del certificado.
Puede tomar los valores: "PEM" o "DER"
                                 String archivo=(req.getParameter("archivo")).trim();       //Indica si se desea descargar en modo
archivo ("si") o directamoente al navegador ("no")

                                 if(sujeto.equals("buscar")){
                                            //Si hemos seleccionado buscar el certificado que pertenece al email
```

```
                                        if(email.length()==0){
                                                //Si la cadena email tiene longitud 0, le volvemos a pedir los datos
                                                PrintWriter out = res.getWriter();
                                                out.write("<html>"
                                                                +"<head>"
                                                                        +"<title>Autoridad de Certificación:
Certificado no encontrado</title>"
                                                                        +"<meta content=\"\">"
                                                                        +"<style></style>"
                                                                +"</head>"
                                                                +"<body>"
                                                                +"<table>"
                                                                        +"<th colspan=2 bgcolor=\"#9591f4\">"
                                                                        +"<center><font size=+3><b><img
src=\"/CA/logo_us.gif\" height=100 width=100 align=\"middle\">Autoridad de Certificación (CA)<img src=\"/CA/esi.gif\" height=100 width=100
align=\"middle\"></font></b></center>"
                                                                        +"</th>"
                                                                +"<tr>"
                                                                        +"<td bgcolor=#FFF6D5>Ha introducido un
email de longitud 0.<br><a href=\"/CA/descargar.html\">Pulse aquí para repetir el formulario</a>"
                                                                        +"<td bgcolor=#E6E6FF width=\"40%\">"
                                                                                +"<h3>"
                                                                                        +"<a
href=\"/CA/index.html\">Página Principal</a>"

            +"<br><br><a href=\"/CA/cert_1.html\">Gestionar su certificado (firma o revocación)</a>"

            +"<br><br><h2>Buscar certificado</h2><h3>"

            +"<br><br><a href=\"/CA/CRL.pem\">Lista de Certificados Revocados (CRL)</a>"
                                                                                +"</h3>"
                                                                                +"</tr>"
                                                                        +"<tr>"
                                                                        +"<td  colspan=2>"
                                                                                +"<hr>"
                                                                                +"<font size=-1>Creado por <a
href=\"mailto:GabrielBabiano@yahoo.es\">Gabriel Babiano Huete</a> en 2002 para su Proyecto Fin de Carrera titulado: <i>\"Estudio e
implementación de una Autoridad de Certificación\"</i>."
                                                                                +"<br><a
href=\"http://trajano.us.es\">Área de Ingeniería Telemática</a> perteneciente al <a href=\"http://www.esi.us.es/ISA\"> Departamento de
```

```
Ingeniería de Sistemas y Automática</a> en la <a href=\"http://www.esi.us.es\">Escuela Superior de Ingenieros</a> de la <a
href=\"http://www.us.es\"> Universidad de Sevilla</a>."
                                                                                                +"</font>"
                                                                                        +"</tr>"
                                                                                        +"</table>"
                                                                                +"</body>"
                                                                        +"</html>");
                                                                out.close();

                                                        }else{
                                                                //Si la cadena email no tiene longitud 0

                                                                //Obtenemos el Hashtable email/serial de ficheroMiIndex
                                                                Hashtable htEmailSerial=UtilCA.fileToHashtable(UtilCA.RUTA_FICHERO_MI_INDEX);
                                                                //Comprobamos que dicho email no tiene ya un certificado de esta CA
                                                                if(htEmailSerial.containsKey(email)){
                                                                        //Si al email le corresponde un certificado válido en esta CA
                                                                        //Enviamos el certificado en tipo, formato y modo solicitado
                                                                        PrintWriter out = res.getWriter();
                                                                        //Establecemos los tipos MIME de la respuesta
                                                                        if(archivo.equals("si")){
                                                                                //Si desea que se le envíe en modo archivo
                                                                                res.setContentType("application/octect-stream");
                                                                        }else{
                                                                                //Si desea que se le envíe directamente al navegador
                                                                                if(tipo.equals("X509")){
                                                                                        res.setContentType("application/x-x509-ca-
cert");

                                                                                }else if(tipo.equals("PKCS7")){
                                                                                        res.setContentType("application/x-pkcs7-
certificates");

                                                                                }
                                                                        }

                                                                        //Mandamos el archivo pedido
                                                                        if(tipo.equals("X509")){
                                                                                out.print(UtilCA.fileToString(UtilCA.RUTA_DIR_CA_FILES+
htEmailSerial.get(email)+formato+".cer"));
                                                                        }else if(tipo.equals("PKCS7")){
                                                                                out.print(UtilCA.fileToString(UtilCA.RUTA_DIR_CA_FILES+
```

```
htEmailSerial.get(email)+formato+".p7b"));
                                                                    }
                                                                    out.close();

                                                            }else{
                                                                    //Si email no está registrada o tiene el certificado revocado
                                                                    PrintWriter out = res.getWriter();
                                                                    out.write("<html>"
                                                                                    +"<head>"
                                                                                            +"<title>Autoridad de
Certificación: Certificado de "+email+" no encontrado</title>"
                                                                                            +"<meta content=\"\">"
                                                                                            +"<style></style>"
                                                                                    +"</head>"
                                                                                    +"<body>"
                                                                                    +"<table>"
                                                                                            +"<th colspan=2
bgcolor=\"#9591f4\">"
                                                                                            +"<center><font size=+3><b><img
src=\"/CA/logo_us.gif\" height=100 width=150 align=\"middle\">Autoridad de Certificación (CA)<img src=\"/CA/esi.gif\" height=75 width=75
align=\"middle\"></font></b></center>"
                                                                                            +"</th>"
                                                                                            +"<tr>"
                                                                                            +"<td bgcolor=#FFF6D5>La
dirección de correo electrónico (email): "+email+" no está registrada en esta Autoridad Certificadora (CA) o su certificado ha sido
revocado."

            +"<ul>Si así lo desea:"

                        +"<li>Puede <a href=\"/CA/cert_1.html\">registrarse</a>"

                        +"<li>O puede volver a <a href=\"/CA/descargar.html\">buscar</a>"

            +"</ul>"                                                                                        +"<td
bgcolor=#E6E6FF width=\"40%\">"
                                                                                                            +"<h3>"
                                                                                                                    +"<a
href=\"/CA/index.html\">Página Principal</a>"
```

```
            +"<br><br><a href=\"/CA/cert_1.html\">Gestionar su certificado (firma o revocación)</a>"

            +"<br><br><a href=\"/CA/descargar.html\">Descargar certificados</a>"
                                                                                                    +"</h3>"
                                                                                                    +"</tr>"
                                                                                        +"<tr>"
                                                                                        +"<td  colspan=2>"
                                                                                                            +"<hr>"
                                                                                                            +"<font
size=-1>Creado por <a href=\"mailto:GabrielBabiano@yahoo.es\">Gabriel Babiano Huete</a> en 2002 para su Proyecto Fin de Carrera titulado:
<i>\"Estudio e implementación de una Autoridad de Certificación\"</i>."
                                                                                                    +"<br><a
href=\"http://trajano.us.es\">Área de Ingeniería Telemática</a> perteneciente al <a href=\"http://www.esi.us.es/ISA\"> Departamento de
Ingeniería de Sistemas y Automática</a> en la <a href=\"http://www.esi.us.es\">Escuela Superior de Ingenieros</a> de la <a
href=\"http://www.us.es\"> Universidad de Sevilla</a>."
                                                                                                            +"</fon
t>"
                                                                                        +"</tr>"
                                                                                        +"</table>"
                                                                                +"</body>"
                                                                        +"</html>");
                                                                out.close();

                                                        }
                                                }

                                        }else if(sujeto.equals("CA")){
                                                PrintWriter out = res.getWriter();
                                                //Establecemos los tipos MIME de la respuesta
                                                if(archivo.equals("si")){
                                                        //Si desea que se le envíe en modo archivo
                                                        res.setContentType("application/octect-stream");
                                                }else{
                                                        //Si desea que se le envíe directamente al navegador
                                                        if(tipo.equals("X509")){
                                                                res.setContentType("application/x-x509-ca-cert");
                                                        }else if(tipo.equals("PKCS7")){
                                                                res.setContentType("application/x-pkcs7-certificates");
                                                        }
                                                }
```

```
                        //Mandamos el archivo pedido
                        if(tipo.equals("X509")){
                                    out.print(UtilCA.fileToString(UtilCA.RUTA_DIR_CA_FILES+"CA"+formato+".cer"));
                        }else if(tipo.equals("PKCS7")){
                                    out.print(UtilCA.fileToString(UtilCA.RUTA_DIR_CA_FILES+"CA"+formato+".p7b"));
                        }
                        out.close();

            }else if(sujeto.equals("CRL")){
                        PrintWriter out = res.getWriter();
                        //Establecemos los tipos MIME de la respuesta
                        if(archivo.equals("si")){
                                    //Si desea que se le envíe en modo archivo
                                    res.setContentType("application/octect-stream");
                        }else{
                                    //Si desea que se le envíe directamente al navegador
                                    if(tipo.equals("X509")){
                                                res.setContentType("application/x-x509-crl");
                                    }else if(tipo.equals("PKCS7")){
                                                res.setContentType("application/x-pkcs7-crl");
                                    }
                        }

                        //Mandamos el archivo pedido
                        if(tipo.equals("X509")){
                                    out.print(UtilCA.fileToString(UtilCA.RUTA_DIR_CA_FILES+"CRL"+formato+".crl"));
                        }else if(tipo.equals("PKCS7")){
                                    out.print(UtilCA.fileToString(UtilCA.RUTA_DIR_CA_FILES+"CRL"+formato+".p7b"));
                        }
                        out.close();

            }else if(sujeto.equals("all")){
                        PrintWriter out = res.getWriter();
                        //Establecemos los tipos MIME de la respuesta
                        if(archivo.equals("si")){
                                    //Si desea que se le envíe en modo archivo
                                    res.setContentType("application/octect-stream");
                        }else{
                                    //Si desea que se le envíe directamente al navegador
```

```
                                                            //Sólo está disponible en tipo PKCS#7
                                                            res.setContentType("application/x-pkcs7-certificates");
                                                }

                                                //Mandamos el archivo pedido
                                                //Sólo está disponible en tipo PKCS#7
                                                out.print(UtilCA.fileToString(UtilCA.RUTA_DIR_CA_FILES+"all"+formato+".p7b"));
                                                out.close();

                                    }

                        }catch(Exception e){
                                    try{
                                                UtilCA.stringToFile(UtilCA.fileToString(UtilCA.RUTA_FICHERO_LOG)+"\n"+new
Date()+"\t"+e.toString(),UtilCA.RUTA_FICHERO_LOG);
                                    }catch(Exception e2){
                                                //Si vuelve a ocurrir un error, se despreciará
                                    }
                        }
            }



            /**
             * Devuelve el nombre del servlet
             *
             * @return el nombre del servlet
             * @since Descargar v1.0
             */

            public java.lang.String getServletName(){
                        return "Descargar v1.0";
            }



            /**
             * Devuelve el nombre del servlet y el autor
             *
             * @return el nombre del servlet y el autor
```

```
             * @since Descargar v1.0
             */

            public String getServletInfo(){
                        return "Servlet Descargar v1.0 por Gabriel Babiano Huete (gabrielbabiano@yahoo.es)";
            }
}
```

ObtenerCSR.java

```
//javac -classpath /opt/jakarta/common/lib/servlet.jar:/opt/jakarta/webapps/examples/WEB-INF/classes/ ObtenerCSR.java


import java.io.*;
import java.util.*;
import java.security.*;

//Incluir en el CLASSPATH /opt/jakarta/common/lib/servlet.jar
import javax.servlet.*;
import javax.servlet.http.*;

//Incluir en el CLASSPATH /opt/jakarta/webapps/examples/WEB-INF/classes/

/**
 * HttpServlet que permite firmar una Petición de Firma de Certificado (CSR) en formato PKCS#10
 *
 * @author Gabriel Babiano Huete (<a hfer="mailto:gabrielbabiano@yahoo.es">gabrielbabiano@yahoo.es</a>)
 * @version 1.0
 */

public class ObtenerCSR extends HttpServlet {


            /**
             * Sobreescribe el método doPost(HttpServletRequest req, HttpServletResponse res) de la clase HttpServlet.
             * Llamado por el servidor para permitir al servlet manejar ena petición POST. De esta manera permite al cliente mandar datos
de longitud ilimitada al servidor web de una vez
```

```
             *
             * @param req Objeto HttpServletRequest que contiene la petición que el cliente ha hecho al servidor
             * @param resp Objeto HttpServletResponse que contiene la respuesta que el servlet envía al cliente
             * @throws IOException Cuando la petición para el POST no se puede manejar
             * @throws ServletException Si se detecta un error de entrada o salida cuando el servlet maneja la petición
             * @since ObtenerCSR v1.0
             */

            protected void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException{

                        try{
                                    PrintWriter out = res.getWriter();

                                    //Obtenemos los datos a partir de la petición
                                    String email=(req.getParameter("email")).trim();          //dirección de correo electrónico del
solicitante
                                    String clave=(req.getParameter("clave")).trim();          //clave correspondiente a la dirección de
correo electrónico del solicitante
                                    String CSR=(req.getParameter("CSR")).trim();              //Petición de Firma de Certificado (CSR) en
formato PKCS#10
                                    String formato=(req.getParameter("formato")).trim();      //Formato de codificación del certificado.
Puede tomar los valores: "PEM" o "DER"
                                    String archivo=(req.getParameter("archivo")).trim();      //Indica si se desea descargar en modo
archivo ("si") o directamoente al navegador ("no")
                                    String tipo=(req.getParameter("tipo")).trim();            //Tipo de certificado. Puede tomar los
valores "X509" o "PKCS7"

                                    //Obtenemos el Hashtable email/serial de miIndex.txt
                                    Hashtable htEmailSerial=UtilCA.fileToHashtable(UtilCA.RUTA_FICHERO_MI_INDEX);
                                    //Comprobamos que dicho email no tiene ya un certificado de esta CA
                                    if(htEmailSerial.containsKey(email)){
                                                //Si al email ya le correspondía un certificado válido en esta CA
                                                res.setContentType("text/html");
                                                out.write("<html>"
                                                                        +"<head>"
                                                                                    +"<title>Autoridad de Certificación: Revocar
Certificado</title>"
                                                                                    +"<meta content=\"\">"
                                                                                    +"<style></style>"
                                                                        +"</head>"
```

```
                                                        +"<body>"
                                                        +"<table>"
                                                                    +"<th colspan=2 bgcolor=\"#9591f4\">"
                                                                                +"<center><font size=+3><b><img
src=\"/CA/logo_us.gif\" height=100 width=150 align=\"middle\">Autoridad de Certificación (CA)<img src=\"/CA/esi.gif\" height=75 width=75
align=\"middle\"></font></b></center>"
                                                                    +"</th>"
                                                                    +"<tr>"
                                                                                +"<td bgcolor=#FFF6D5>"
                                                                                +"<br>Si desea pedir una nueva
certificación en esta Autoridad de Certificación (CA), es necesario que antes revoque su antiguo certificado."
                                                                                                                +"<form
action=\"/examples/servlet/Revocar\" method=\"POST\">Si desea revocar dicho certificado, mire el último email con tema <i>\"Mensaje desde
la CA\"</i> en la dirección de <b>email</b>: <input name=\"email\" value=\""+email+"\"> y remítanos la <b>clave</b> que se indica en dicho
mensaje:<input name=\"clave\"><input type=\"submit\" value=\"Enviar\"></form>"
                                                                                +"<td bgcolor=#E6E6FF width=\"40%\">"
                                                                                +"<h3>"
                                                                                            +"<a
href=\"/CA/index.html\">Página Principal</a>"

           +"<br><br><h2>Gestionar su certificado (firma o revocación)</h2><h3>"
                                                                                                                +"<br><br><a
href=\"/CA/descargar.html\">Descargar certificados</a>"
                                                                                            +"</h3>"
                                                                                            +"</tr>"
                                                                    +"<tr>"
                                                                                +"<td  colspan=2>"
                                                                                            +"<hr>"
                                                                                            +"<font size=-1>Creado por <a
href=\"mailto:GabrielBabiano@yahoo.es\">Gabriel Babiano Huete</a> en 2002 para su Proyecto Fin de Carrera titulado: <i>\"Estudio e
implementación de una Autoridad de Certificación\"</i>."
                                                                                +"<br><a href=\"http://trajano.us.es\">Área
de Ingeniería Telemática</a> perteneciente al <a href=\"http://www.esi.us.es/ISA\"> Departamento de Ingeniería de Sistemas y
Automática</a> en la <a href=\"http://www.esi.us.es\">Escuela Superior de Ingenieros</a> de la <a href=\"http://www.us.es\"> Universidad
de Sevilla</a>."
                                                                                            +"</font>"
                                                                    +"</tr>"
                                                                    +"</table>"
                                                        +"</body>"
                                                +"</html>");
```

```
                                }else{
                                        //Si al email no le corresponde (por ahora) un certificado válido en esta CA

                                        //Creamos un Hashtable email/clave
                                        Hashtable htEmailClave = UtilCA.fileToHashtable(UtilCA.RUTA_FICHERO_EMAILS);

                                        //Comprobamos que la pareja email/clave se encuentra en la HashTable
                                        if(htEmailClave.containsKey(email)==true){

                                                //Obtenemos la clave verdadera que corresponde a ese email
                                                String claveOK=(String) htEmailClave.get(email);
                                                //Comparamos la clave correcta con la clave que nos llega en la petición
                                                if(claveOK.equals(clave)==true){

                                                        //Copiamos el CSR a un archivo
                                                        UtilCA.stringToFile(CSR,UtilCA.RUTA_FICHERO_CSR);

                                                        //A continuación le firmamos su CSR
                                                        Runtime r=Runtime.getRuntime();
                                                        r.exec(UtilCA.SCRIPT_CERTIFICA+" "+UtilCA.RUTA_FICHERO_CSR+"
"+UtilCA.RUTA_FICHERO_LAST_CERT);

                                                        //Esperamos a que se cree el fichero que contiene el certificado
                                                        File ficheroCertificado=new File(UtilCA.RUTA_FICHERO_LAST_CERT);
                                                        do{         //Esperamos mientras que el fichero no exista
                                                        }while(ficheroCertificado.exists()==false);

                                                        //Obtenemos el número de serie del certificado (en serial.old),
para a partir de él obtener el nombre del certificado dentro del directorio /usr/share/ssl/demoCA/newcerts
                                                        int cuenta=0;           //Para "controlar" si se ha firmado el CSR
o no

                                                        do{         //Esperamos mientras que el fichero no exista. Se hace
de esta manera porque el método "wait" no funciona bien

                                                                cuenta++;
                                                        }while(new File(UtilCA.RUTA_FICHERO_SERIAL_OLD).exists()==false &&
cuenta<500000);

                                                        if(cuenta==500000){
                                                                //Si se cumple la cuenta, es que se ha producido un
error y el CSR no se ha podido firmar
```

```
                                                                    out.write("<html>"
                                                                                +"<head>"
                                                                                            +"<title>Autoridad
de Certificación: CSR incorrecta</title>"
                                                                                            +"<meta
content=\"\">"
                                                                                            +"<style></style>"
                                                                                +"</head>"
                                                                                +"<body>"
                                                                                +"<table>"
                                                                                            +"<th colspan=2
bgcolor=\"#9591f4\">"
            +"<center><font size=+3><b><img src=\"/CA/logo_us.gif\" height=100 width=150 align=\"middle\">Autoridad de Certificación
(CA)<img src=\"/CA/esi.gif\" height=75 width=75 align=\"middle\"></font></b></center>"
                                                                                            +"</th>"
                                                                                            +"<tr>"
                                                                                                        +"<td
bgcolor=#FFF6D5>Ha introducido una <b>Petición de Firma de Certificado (CSR) incorrecta.</b>"
                                                                                                        +"<br><
a href=\"/CA/cert_1.html\">Pulse aquí para repetir.</a>"
                                                                                                        +"<td
bgcolor=#E6E6FF width=\"40%\">"
                                                                                                        +"<h3>"
            +"<a href=\"/CA/index.html\">Página Principal</a>"
                        +"<br><br><a href=\"/CA/cert_1.html\">Gestionar su certificado (firma o revocación)"
                        +"<br><br><a href=\"/CA/descargar.html\">Descargar certificados</a>"
            +"</h3>"
            +"</tr>"
                                                                                            +"<tr>"
                                                                                                        +"<td
colspan=2>"
            +"<hr>"
```

```
           +"<font size=-1>Creado por <a href=\"mailto:GabrielBabiano@yahoo.es\">Gabriel Babiano Huete</a> en 2002 para su Proyecto Fin
de Carrera titulado: <i>\"Estudio e implementación de una Autoridad de Certificación\"</i>."
                                                                                                                                +"<br><
a href=\"http://trajano.us.es\">Área de Ingeniería Telemática</a> perteneciente al <a href=\"http://www.esi.us.es/ISA\"> Departamento de
Ingeniería de Sistemas y Automática</a> en la <a href=\"http://www.esi.us.es\">Escuela Superior de Ingenieros</a> de la <a
href=\"http://www.us.es\"> Universidad de Sevilla</a>."

           +"</font>"
                                                                                                                                    +"</tr>"
                                                                                                                                    +"</table>"
                                                                                                                  +"</body>"
                                                                                                        +"</html>");
                                                                                                throw new CAException("CSR incorrecta");

                                                                                        }

                                                                                        //Copiamos serial.old
                                                                                        String
cadenaFicheroSerialOld=UtilCA.fileToString(UtilCA.RUTA_FICHERO_SERIAL_OLD).trim();
                                                                                        //Se elimina serial.old
                                                                                        r.exec(UtilCA.SCRIPT_BORRA_FICHERO+"
"+UtilCA.RUTA_FICHERO_SERIAL_OLD);

                                                                                        //Copiamos el certificado en el directorio files y creamos el
certificado en formato DER
                                                                                        r.exec(UtilCA.SCRIPT_CREA_DER+"
"+UtilCA.RUTA_DIR_CA+"newcerts/"+cadenaFicheroSerialOld+".pem "+UtilCA.RUTA_DIR_CA_FILES+cadenaFicheroSerialOld+"PEM.cer
"+UtilCA.RUTA_DIR_CA_FILES+cadenaFicheroSerialOld+"DER.cer "+UtilCA.RUTA_DIR_CA_FILES+cadenaFicheroSerialOld+"PEM.p7b
"+UtilCA.RUTA_DIR_CA_FILES+cadenaFicheroSerialOld+"DER.p7b");

                                                                                        //Actualizamos miIndex.txt
                                                                                        htEmailSerial.put(email,cadenaFicheroSerialOld);
                                                                                        UtilCA.hashtableToFile(htEmailSerial,UtilCA.RUTA_FICHERO_MI_INDEX);

                                                                                        //Actualizaremos la CRL y el PKCS#7
                                                                                        String[] serialsValidos=(String[])
htEmailSerial.values().toArray(new String[0]);
                                                                                        //Creación de un fichero llamado allCACert.pem que contenga todos
los certificados X.509 en formato PEM válidos incluido el certificado X.509 en formato PEM de la CA
                                                                                        String cadenaAllValidCerts=new String();
```

```
                                                        cadenaAllValidCerts+=UtilCA.fileToString(UtilCA.RUTA_DIR_CA+"cacert
.pem");
                                                        for(int i=1;i<=htEmailSerial.size();i++){
                                                                    cadenaAllValidCerts+=UtilCA.fileToString(UtilCA.RUTA_DI
R_CA+"newcerts/"+serialsValidos[i-1]+".pem");
                                                        }
                                                        UtilCA.stringToFile(cadenaAllValidCerts,UtilCA.RUTA_FICHERO_ALL_VAL
ID_CERTS);
                                                        r.exec(UtilCA.SCRIPT_ACTUALIZA_CA+"
"+UtilCA.RUTA_FICHERO_ALL_VALID_CERTS);


                                                        //Establecemos los tipos MIME de la respuesta
                                                        if(archivo.equals("si")){
                                                                    //Si desea que se le envíe en modo archivo
                                                                    res.setContentType("application/octect-stream");
                                                        }else{
                                                                    //Si desea que se le envíe directamente al navegador
                                                                    if(tipo.equals("X509")){
                                                                                res.setContentType("application/x-x509-
user-cert");
                                                                    }else if(tipo.equals("PKCS7")){
                                                                                res.setContentType("application/x-pkcs7-
certificates");
                                                                    }
                                                        }

                                                        //Mandamos el archivo pedido
                                                        if(tipo.equals("X509")){
                                                                    out.print(UtilCA.fileToString(UtilCA.RUTA_DIR_CA_FILES+
cadenaFicheroSerialOld+formato+".cer"));
                                                        }else if(tipo.equals("PKCS7")){
                                                                    out.print(UtilCA.fileToString(UtilCA.RUTA_DIR_CA_FILES+
cadenaFicheroSerialOld+formato+".p7b"));
                                                        }

                                                        //Se elimina lastCertificate
                                                        r.exec(UtilCA.SCRIPT_BORRA_FICHERO+"
"+UtilCA.RUTA_FICHERO_LAST_CERT);

                                                }else{
```

```
                                                        //Si la clave que nos llega no es correcta
                                                        res.setContentType("text/html");
                                                        out.write("<html>"
                                                                        +"<head>"
                                                                                    +"<title>Autoridad de
Certificación: clave incorrecta</title>"
                                                                                    +"<meta content=\"\">"
                                                                                    +"<style></style>"
                                                                        +"</head>"
                                                                        +"<body>"
                                                                        +"<table>"
                                                                                    +"<th colspan=2
bgcolor=\"#9591f4\">"
                                                                                                +"<center><font
size=+3><b><img src=\"/CA/logo_us.gif\" height=100 width=150 align=\"middle\">Autoridad de Certificación (CA)<img src=\"/CA/esi.gif\"
height=75 width=75 align=\"middle\"></font></b></center>"
                                                                                    +"</th>"
                                                                                    +"<tr>"
                                                                                                +"<td
bgcolor=#FFF6D5>"
                                                                                                +"<b>Clave
incorrecta.</b>"

            +"<br>Pulse <a href=\"/CA/cert_1.html\">aquí</a> si desea que le volvamos a remitir una nueva clave a la dirección de email"
                                                                                                +"<td
bgcolor=#E6E6FF width=\"40%\">"
                                                                                                +"<h3>"
                                                                                                            +"<a
href=\"/CA/index.html\">Página Principal</a>"

            +"<br><br><a href=\"/CA/cert_1.html\">Gestionar su certificado (firma o revocación)</a>"

            +"<br><br><a href=\"/CA/descargar.html\">Descargar certificados</a>"

            +"</h3>"
                                                                                                +"</tr>"
                                                                                    +"<tr>"
                                                                                                +"<td  colspan=2>"
                                                                                                            +"<hr>"
                                                                                                            +"<font
```

```
size=-1>Creado por <a href=\"mailto:GabrielBabiano@yahoo.es\">Gabriel Babiano Huete</a> en 2002 para su Proyecto Fin de Carrera titulado:
<i>\"Estudio e implementación de una Autoridad de Certificación\"</i>."
                                                                                                                                +"<br><a
href=\"http://trajano.us.es\">Área de Ingeniería Telemática</a> perteneciente al <a href=\"http://www.esi.us.es/ISA\"> Departamento de
Ingeniería de Sistemas y Automática</a> en la <a href=\"http://www.esi.us.es\">Escuela Superior de Ingenieros</a> de la <a
href=\"http://www.us.es\"> Universidad de Sevilla</a>."
                                                                                                                                                +"</fon
t>"
                                                                                                                    +"</tr>"
                                                                                                                    +"</table>"
                                                                                                          +"</body>"
                                                                                                +"</html>");

                                                                          }
                                                                }else{
                                                                          //Si al email no le corresponde ninguna clave
                                                                          res.setContentType("text/html");
                                                                          out.write("<html>"
                                                                                                +"<head>"
                                                                                                          +"<title>Autoridad de Certificación:
"+email+" no registrado</title>"
                                                                                                          +"<meta content=\"\">"
                                                                                                          +"<style></style>"
                                                                                                +"</head>"
                                                                                                +"<body>"
                                                                                                +"<table>"
                                                                                                          +"<th colspan=2 bgcolor=\"#9591f4\">"
                                                                                                                    +"<center><font size=+3><b><img
src=\"/CA/logo_us.gif\" height=100 width=150 align=\"middle\">Autoridad de Certificación (CA)<img src=\"/CA/esi.gif\" height=75 width=75
align=\"middle\"></font></b></center>"
                                                                                                          +"</th>"
                                                                                                          +"<tr>"
                                                                                                                    +"<td bgcolor=#FFF6D5>"
                                                                                                                    +"La dirección de correo
electrónico (email): "+email+" no está registrada en esta Autoridad Certificadora (CA)."
                                                                                                                                                +"<br>
Si así lo desea, se puede <a href=\"/CA/cert_1.html\">registrar</a>"
                                                                                                                    +"<td bgcolor=#E6E6FF
width=\"40%\">"
                                                                                                                    +"<h3>"
```

```
                                                                                                                                                +"<a
href=\"/CA/index.html\">Página Principal</a>"

            +"<br><br><a href=\"/CA/cert_1.html\">Gestionar su certificado (firma o revocación)"

            +"<br><br><a href=\"/CA/descargar.html\">Descargar certificados</a>"

                                                                                                                                                +"</h3>"
                                                                                                                                                +"</tr>"
                                                                                                                                  +"<tr>"
                                                                                                                                              +"<td  colspan=2>"
                                                                                                                                                +"<hr>"
                                                                                                                                                +"<font size=-
1>Creado por <a href=\"mailto:GabrielBabiano@yahoo.es\">Gabriel Babiano Huete</a> en 2002 para su Proyecto Fin de Carrera titulado:
<i>\"Estudio e implementación de una Autoridad de Certificación\"</i>."
                                                                                                                                              +"<br><a
href=\"http://trajano.us.es\">Área de Ingeniería Telemática</a> perteneciente al <a href=\"http://www.esi.us.es/ISA\"> Departamento de
Ingeniería de Sistemas y Automática</a> en la <a href=\"http://www.esi.us.es\">Escuela Superior de Ingenieros</a> de la <a
href=\"http://www.us.es\"> Universidad de Sevilla</a>."
                                                                                                                                                +"</font>"
                                                                                                                                  +"</tr>"
                                                                                                                                  +"</table>"
                                                                                                                        +"</body>"
                                                                                                              +"</html>");
                                                }
                                    }

                                    out.close();
                        }catch(Exception e){
                                    try{
                                                UtilCA.stringToFile(UtilCA.fileToString(UtilCA.RUTA_FICHERO_LOG)+"\n"+new
Date()+"\t"+e.toString(),UtilCA.RUTA_FICHERO_LOG);
                                    }catch(Exception e2){
                                                //Si vuelve a ocurrir un error, se despreciará
                                    }
                        }
            }
```

```
            /**
             * Devuelve el nombre del servlet
             *
             * @return el nombre del servlet
             * @since ObtenerCSR v1.0
             */

            public java.lang.String getServletName(){
                        return "ObtenerCSR v1.0";
            }


            /**
             * Devuelve el nombre del servlet y el autor
             *
             * @return el nombre del servlet y el autor
             * @since ObtenerCSR v1.0
             */

             public String getServletInfo(){
                        return "Servlet ObtenerCSR por Gabriel Babiano Huete";
            }
}
```

ObtenerEmail.java

```
//javac -classpath /opt/jakarta/common/lib/servlet.jar:/opt/jakarta/webapps/examples/WEB-INF/classes/ ObtenerEmail.java


import java.io.*;
import java.util.*;
import java.security.*;

//Incluir en el CLASSPATH /opt/jakarta/common/lib/servlet.jar
import javax.servlet.*;
import javax.servlet.http.*;
```

```
//Incluir en el CLASSPATH /opt/jakarta/webapps/examples/WEB-INF/classes/


/**
 * HttpServlet que permite obtener el email del cliente
 *
 * @author Gabriel Babiano Huete (<a hfer="mailto:gabrielbabiano@yahoo.es">gabrielbabiano@yahoo.es</a>)
 * @version 1.0
 */

public class ObtenerEmail extends HttpServlet {


           /**
            * Sobreescribe el método doPost(HttpServletRequest req, HttpServletResponse res) de la clase HttpServlet.
            * Llamado por el servidor para permitir al servlet manejar ena petición POST. De esta manera permite al cliente mandar datos
de longitud ilimitada al servidor web de una vez
            *
            * @param req Objeto HttpServletRequest que contiene la petición que el cliente ha hecho al servidor
            * @param resp Objeto HttpServletResponse que contiene la respuesta que el servlet envía al cliente
            * @throws IOException Cuando la petición para el POST no se puede manejar
            * @throws ServletException Si se detecta un error de entrada o salida cuando el servlet maneja la petición
            * @since ObtenerEmail v1.0
            */

           protected void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException{

                      try{
                                 PrintWriter out = res.getWriter();

                                 //Obtenemos la dirección de correo electrónico a partir de la petición
                                 String email=(req.getParameter("email")).trim();                      //dirección de correo
electrónico del solicitante
                                 //NOTA: en caso de que se repita la misma dirección de email de alguna ya introducida, se
sobreescribirá la antigua clave.

                                 if(email.length()==0){
                                 out.write("<html>"
                                                                                +"<head>"
                                                                                           +"<title>Autoridad de Certificación: email
```

```
"+email+" incorrecto</title>"
                                                                    +"<meta content=\"\">"
                                                                    +"<style></style>"
                                                                +"</head>"
                                                                +"<body>"
                                                                +"<table>"
                                                                    +"<th colspan=2 bgcolor=\"#9591f4\">"
                                                                        +"<center><font size=+3><b><img
src=\"/CA/logo_us.gif\" height=100 width=150 align=\"middle\">Autoridad de Certificación (CA)<img src=\"/CA/esi.gif\" height=75 width=75
align=\"middle\"></font></b></center>"
                                                                    +"</th>"
                                                                    +"<tr>"
                                                                        +"<td bgcolor=#FFF6D5>"
                                                                        +"Ha introducido una dirección
de correo electrónica incorrecta, <a href=\"/CA/cert_1.html\">pulse aquí si desea volver a introducirla</a>"
                                                                        +"<td bgcolor=#E6E6FF
width=\"40%\">"
                                                                        +"<h3>"
                                                                            +"<a
href=\"/CA/index.html\">Página Principal</a>"

            +"<br><br><a href=\"/CA/cert_1.html\">Gestionar su certificado (firma o revocación)"

            +"<br><br><a href=\"/CA/descargar.html\">Descargar certificados</a>"
                                                                            +"</h3>"
                                                                            +"</tr>"
                                                                    +"<tr>"
                                                                        +"<td  colspan=2>"
                                                                            +"<hr>"
                                                                            +"<font size=-
1>Creado por <a href=\"mailto:GabrielBabiano@yahoo.es\">Gabriel Babiano Huete</a> en 2002 para su Proyecto Fin de Carrera titulado:
<i>\"Estudio e implementación de una Autoridad de Certificación\"</i>."
                                                                        +"<br><a
href=\"http://trajano.us.es\">Área de Ingeniería Telemática</a> perteneciente al <a href=\"http://www.esi.us.es/ISA\"> Departamento de
Ingeniería de Sistemas y Automática</a> en la <a href=\"http://www.esi.us.es\">Escuela Superior de Ingenieros</a> de la <a
href=\"http://www.us.es\"> Universidad de Sevilla</a>."
                                                                            +"</font>"
                                                                    +"</tr>"
                                                                    +"</table>"
                                                                +"</body>"
```

```
                                                        +"</html>");
                                }else{
                                        //Creamos un Hashtable email/clave
                                        Hashtable ht = UtilCA.fileToHashtable(UtilCA.RUTA_FICHERO_EMAILS);

                                        //Tanto si se ha creado miFichero como si no, procederemos a añadir la nueva entrada
                                        //Para ello crearemos un número aleatorio seguro (SecureRandom) de tipo SHA1PRNG...
                                        SecureRandom random = SecureRandom.getInstance("SHA1PRNG");
                                        //...a partir de la fecha en milisegundos desde las 00:00:00 GMT del 1 de enero de 1970...
                                        Date fecha=new Date();
                                        //...como semilla.
                                        random.setSeed(fecha.getTime());
                                        //Introducimos en la HashTable el email y dicho número aleatorio seguro
                                        String clave=new String(Long.toString(random.nextLong()));
                                        ht.put(email, clave);

                                        //Pasamos la Hashtable al fichero
                                        UtilCA.hashtableToFile(ht,UtilCA.RUTA_FICHERO_EMAILS);

                                        //Creamos ahora el fichero con el email y la clave que mandaremos a la dirección de correo
electrónico especificado en email

                                        File suFichero= new File(UtilCA.RUTA_FICHERO_IMPORTANTE);   //Lo que haya en suFichero es
destruido cuando escribimos encima.

                                        FileOutputStream suCanalSalida=new FileOutputStream(suFichero); //lo que escribamos al
canalSalida, pasará a suFichero

                                        String aSuFichero=new String();                 //Cadena que contendrá lo que
posteriormente escribiremos en suFichero


                                        aSuFichero=aSuFichero+"\n\temail: "+email+"\n\tclave: "+clave;

                                        byte suB[]=aSuFichero.getBytes();
                                        suCanalSalida.write(suB);
                                        suCanalSalida.close();

                                        //Mandamos suFichero a la dirección de email especificada en la cadena email
                                        Runtime r=Runtime.getRuntime();
                                        r.exec(UtilCA.SCRIPT_MANDA_EMAIL+" "+email+" "+UtilCA.RUTA_FICHERO_IMPORTANTE);

                                        //mandamos la respuesta
                                        res.setContentType("text/html");
```

```
                                        Hashtable htMiIndex=UtilCA.fileToHashtable(UtilCA.RUTA_FICHERO_MI_INDEX);

                                        if(htMiIndex.containsKey(email)){
                                                    //Si al email le corresponde un certificado válido en esta CA
                                                    out.write("<html>"
                                                                +"<head>"
                                                                            +"<title>Autoridad de Certificación: email
"+email+" incorrecto</title>"
                                                                            +"<meta content=\"\">"
                                                                            +"<style></style>"
                                                                +"</head>"
                                                                +"<body>"
                                                                +"<table>"
                                                                            +"<th colspan=2 bgcolor=\"#9591f4\">"
                                                                                        +"<center><font size=+3><b><img
src=\"/CA/logo_us.gif\" height=100 width=150 align=\"middle\">Autoridad de Certificación (CA)<img src=\"/CA/esi.gif\" height=75 width=75
align=\"middle\"></font></b></center>"
                                                                            +"</th>"
                                                                            +"<tr>"
                                                                                        +"<td bgcolor=#FFF6D5>"
                                                                                        +"<br>Antes de pedir una nueva
certificación en esta Autoridad de Certificación (CA), es necesario que revoque su antiguo certificado."
                                                                                                    +"<ul>"

                                                    +"<li><form action=\"/examples/servlet/Revocar\" method=\"POST\">Si desea revocar dicho certificado, mire el último email con
tema <i>\"Mensaje desde la CA\"</i> en la dirección de correo electrónico:<input name=\"email\" value=\""+email+"\" readonly> y remítanos
la clave que se indica en dicho mensaje:<input name=\"clave\"><input type=\"submit\" value=\"Enviar\"></form>"

                                                    +"<li>O bien, <a href=\"/CA/index.html\">pulse aquí si desea ir a la página principal</a>"
                                                                                                    +"</ul>
"
                                                                                        +"<td bgcolor=#E6E6FF
width=\"40%\">"
                                                                                        +"<h3>"
                                                                                                    +"<a
href=\"/CA/index.html\">Página Principal</a>"

                                                    +"<br><br><a href=\"/CA/cert_1.html\">Gestionar su certificado (firma o revocación)"

                                                    +"<br><br><a href=\"/CA/descargar.html\">Descargar certificados</a>"
```

```
                                                                                                   +"</h3>"
                                                                                                   +"</tr>"
                                                                             +"<tr>"
                                                                                         +"<td  colspan=2>"
                                                                                                   +"<hr>"
                                                                                                   +"<font size=-
1>Creado por <a href=\"mailto:GabrielBabiano@yahoo.es\">Gabriel Babiano Huete</a> en 2002 para su Proyecto Fin de Carrera titulado:
<i>\"Estudio e implementación de una Autoridad de Certificación\"</i>."
                                                                                         +"<br><a
href=\"http://trajano.us.es\">Área de Ingeniería Telemática</a> perteneciente al <a href=\"http://www.esi.us.es/ISA\"> Departamento de
Ingeniería de Sistemas y Automática</a> en la <a href=\"http://www.esi.us.es\">Escuela Superior de Ingenieros</a> de la <a
href=\"http://www.us.es\"> Universidad de Sevilla</a>."
                                                                                                   +"</font>"
                                                                             +"</tr>"
                                                                             +"</table>"
                                                                   +"</body>"
                                                         +"</html>");

                                        }else{
                                                  //Si al email no le corresponde un certificado válido en esta CA
                                                  out.write("<html>"
                                                                   +"<head>"
                                                                             +"<title>Autoridad de Certificación: email
"+email+" incorrecto</title>"
                                                                             +"<meta content=\"\">"
                                                                             +"<style></style>"
                                                                   +"</head>"
                                                                   +"<body>"
                                                                   +"<table>"
                                                                             +"<th colspan=2 bgcolor=\"#9591f4\">"
                                                                                         +"<center><font size=+3><b><img
src=\"/CA/logo_us.gif\" height=100 width=150 align=\"middle\">Autoridad de Certificación (CA)<img src=\"/CA/esi.gif\" height=75 width=75
align=\"middle\"></font></b></center>"
                                                                             +"</th>"
                                                                             +"<tr>"
                                                                                         +"<td bgcolor=#FFF6D5>"
                                                                                         +"<br>Por favor, consulte (en
otra ventana) en su dirección de correo electrónico ("+email+") el último mensaje recibido de los titulados <i>\"Mensaje desde la CA\"</i>
y posteriormente introduzca los siguientes datos:"
                                                                                                             +"<form
```

```
action=\"/examples/servlet/ObtenerCSR\" method=\"POST\">"

            +"<b>email</b>: <input name=\"email\" value=\""+email+"\" readonly>"

            +"<br><b>clave</b>: <input name=\"clave\">"

            +"<br><b>Petición de Firma de Certificado (CSR) en formato PKCS#10</b>: <textarea name=\"CSR\" rows=20 cols=100></textarea>"

            +"<fieldset>"
                                                                                                                    +"<lege
nd align=\"left\"><b>Tipo de Certificado:</b></legend>"
                                                                                                                    +"<br><
input type=\"radio\" name=\"tipo\" value=\"X509\" checked><label for=\"X509\">X.509</LABEL>"
                                                                                                                    +"<br><
input type=\"radio\" name=\"tipo\" value=\"PKCS7\"><label for=\"PKCS7\">PKCS#7</LABEL>"

            +"</fieldset>"
                                                                                                                    +"<fiel
dset>"
                                                                                                                    +"<lege
nd align=\"left\"><b>Formato del Certificado:</b></legend>"
                                                                                                                    +"<br><
input type=\"radio\" name=\"formato\" value=\"PEM\" checked><label for=\"PEM\">PEM</LABEL>"
                                                                                                                    +"<br><
input type=\"radio\" name=\"formato\" value=\"DER\"><label for=\"DER\">DER</LABEL>"
                                                                                                +"</fieldset>"
                                                                                                +"<input
type=\"radio\" checked name=\"archivo\" value=\"si\"><label for=\"archivo\">En archivo</label>"
                                                                                                +"<input
type=\"radio\" name=\"archivo\" value=\"no\"><label for=\"archivo\">Directamente al navegador</label>"

            +"<br><center><input type=\"submit\" value=\"Enviar\"><input type=\"reset\" value=\"Borrar todo\"></center>"
                                                                                                                    +"</for
m>"
                                                                                                                    +"<br><
br>En caso de que se descargue un archivo, se recomienda que al guardarlo, se utilicen las extensiones:"
                                                                                                +"<ul>"
                                                                                                +"<li><b>.cer</b>,
<b>.crt</b> o <b>.cert</b> para certificados X.509"
```

```
            +"<li><b>.p7b</b> para PKCS#7"

            +"<li><b>.crl</b> para CRL X.509"
                                                                                                                                                    +"</ul>
"
                                                                                                                            +"<td bgcolor=#E6E6FF
width=\"40%\">"
                                                                                                                            +"<h3>"
                                                                                                                                        +"<a
href=\"/CA/index.html\">Página Principal</a>"

            +"<br><br><a href=\"/CA/cert_1.html\">Gestionar su certificado (firma o revocación)"

            +"<br><br><a href=\"/CA/descargar.html\">Descargar certificados</a>"
                                                                                                                                        +"</h3>"
                                                                                                                                        +"</tr>"
                                                                                                                +"<tr>"
                                                                                                                            +"<td  colspan=2>"
                                                                                                                                        +"<hr>"
                                                                                                                                        +"<font size=-
1>Creado por <a href=\"mailto:GabrielBabiano@yahoo.es\">Gabriel Babiano Huete</a> en 2002 para su Proyecto Fin de Carrera titulado:
<i>\"Estudio e implementación de una Autoridad de Certificación\"</i>."
                                                                                                                            +"<br><a
href=\"http://trajano.us.es\">Área de Ingeniería Telemática</a> perteneciente al <a href=\"http://www.esi.us.es/ISA\"> Departamento de
Ingeniería de Sistemas y Automática</a> en la <a href=\"http://www.esi.us.es\">Escuela Superior de Ingenieros</a> de la <a
href=\"http://www.us.es\"> Universidad de Sevilla</a>."
                                                                                                                                        +"</font>"
                                                                                                                +"</tr>"
                                                                                                                +"</table>"
                                                                                                    +"</body>"
                                                                                        +"</html>");

                                                }
                                    }
                                    out.close();


                        }catch(Exception e){
                                    try{
                                                UtilCA.stringToFile(UtilCA.fileToString(UtilCA.RUTA_FICHERO_LOG)+"\n"+new
```

```
Date()+"\t"+e.toString(),UtilCA.RUTA_FICHERO_LOG);
                                    }catch(Exception e2){
                                                //Si vuelve a ocurrir un error, se despreciará
                                    }
                        }
            }

            /**
             * Devuelve el nombre del servlet
             *
             * @return el nombre del servlet
             * @since ObtenerEmail v1.0
             */

            public java.lang.String getServletName(){
                        return "ObtenerEmail v1.0";
            }



            /**
             * Devuelve el nombre del servlet y el autor
             *
             * @return el nombre del servlet y el autor
             * @since ObtenerEmail v1.0
             */

            public String getServletInfo(){
                        return "Servlet ObtenerEmail v1.0 por Gabriel Babiano Huete (gabrielbabiano@yahoo.es)";
            }
}
```

Revocar.java

```
//javac -classpath /opt/jakarta/common/lib/servlet.jar:/opt/jakarta/webapps/examples/WEB-INF/classes/ ObtenerEmail.java
```

```
import java.io.*;
import java.util.*;
import java.security.*;

//Incluir en el CLASSPATH /opt/jakarta/common/lib/servlet.jar
import javax.servlet.*;
import javax.servlet.http.*;

//Incluir en el CLASSPATH /opt/jakarta/webapps/examples/WEB-INF/classes/

public class Revocar extends HttpServlet {


           protected void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException{
                      try{
                                 PrintWriter out = res.getWriter();
                                 res.setContentType("text/html");

                                 //Obtenemos la dirección de correo electrónico a partir de la petición
                                 String email=(req.getParameter("email")).trim();                      //dirección de correo
electrónico del solicitante
                                 String clave=(req.getParameter("clave")).trim();                      //clave correspondiente a la
dirección de correo electrónico del solicitante

                                 //Creamos un Hashtable email/clave
                                 Hashtable htEmailClave = UtilCA.fileToHashtable(UtilCA.RUTA_FICHERO_EMAILS);


                                 if(htEmailClave.containsKey(email)){
                                            //Si la dirección de email está registrada

                                            if(((String) htEmailClave.get(email)).equals(clave)){
                                                       //Si la clave de email coincide

                                                       //Creamos una HashTable (htMiIndex) de parejas email/serial
                                                       Hashtable htEmailSerial =
UtilCA.fileToHashtable(UtilCA.RUTA_FICHERO_MI_INDEX);

                                                       if(htEmailSerial.containsKey(email)){
                                                                  //Si htMiIndex contiene a email, es equivalente a decir que email
```

```
tiene un certificado válido asociado

                                                                        //Revocamos el certificado
                                                                        Runtime r=Runtime.getRuntime();
                                                                        r.exec(UtilCA.SCRIPT_REVOCA+"
"+UtilCA.RUTA_DIR_CA+"newcerts/"+htEmailSerial.get(email)+".pem");

                                                                        //Eliminamos el email de miIndex y actualizamos el archivo donde se
almacena
                                                                        htEmailSerial.remove(email);
                                                                        UtilCA.hashtableToFile(htEmailSerial,UtilCA.RUTA_FICHERO_MI_INDEX);

                                                                        //Actualizaremos la CRL y el PKCS#7
                                                                        String[] serialsValidos=(String[])
htEmailSerial.values().toArray(new String[0]);
                                                                        //Creación de un fichero llamado allValidCerts.pem que contenga
todos los certificados X.509 en formato PEM válidos incluido el certificado X.509 en formato PEM de la CA
                                                                        String cadenaAllValidCerts=new String();
                                                                        cadenaAllValidCerts+=UtilCA.fileToString(UtilCA.RUTA_DIR_CA+"cacert
.pem");
                                                                        for(int i=1;i<=htEmailSerial.size();i++){
                                                                                    cadenaAllValidCerts+=UtilCA.fileToString(UtilCA.RUTA_DI
R_CA+"newcerts/"+serialsValidos[i-1]+".pem");
                                                                        }
                                                                        UtilCA.stringToFile(cadenaAllValidCerts,UtilCA.RUTA_FICHERO_ALL_VAL
ID_CERTS);
                                                                        r.exec(UtilCA.SCRIPT_ACTUALIZA_CA+"
"+UtilCA.RUTA_FICHERO_ALL_VALID_CERTS);

                                                                        out.write("<html>"
                                                                                                +"<head>"
                                                                                                            +"<title>Autoridad de
Certificación: Certificado perteneciente a <b>"+email+"</b> revocado</title>"
                                                                                                            +"<meta content=\"\">"
                                                                                                            +"<style></style>"
                                                                                                +"</head>"
                                                                                                +"<body>"
                                                                                                +"<table>"
                                                                                                            +"<th colspan=2
bgcolor=\"#9591f4\">"
```

```
                                                                                                                                +"<center><font
size=+3><b><img src=\"/CA/logo_us.gif\" height=100 width=150 align=\"middle\">Autoridad de Certificación (CA)<img src=\"/CA/esi.gif\"
height=75 width=75 align=\"middle\"></font></b></center>"
                                                                                                                    +"</th>"
                                                                                                                    +"<tr>"
                                                                                                                                +"<td
bgcolor=#FFF6D5>Certificado perteneciente a <b>"+email+"</b> revocado."

            +"<br>Si así lo desea, puede <a href=\"/CA/cert_1.html\">pedir una certificación</a>."
                                                                                                                    +"<td bgcolor=#E6E6FF
width=\"40%\">"
                                                                                                                                +"<h3>"
                                                                                                                                            +"<a
href=\"/CA/index.html\">Página Principal</a>"

            +"<br><br><a href=\"/CA/cert_1.html\">Gestionar su certificado (firma o revocación)</a>"

            +"<br><br><a href=\"/CA/descargar.html\">Descargar certificado</a>"


            +"</h3>"
                                                                                                                                +"</tr>"
                                                                                                                    +"<tr>"
                                                                                                                                +"<td  colspan=2>"
                                                                                                                                            +"<hr>"
                                                                                                                                            +"<font
size=-1>Creado por <a href=\"mailto:GabrielBabiano@yahoo.es\">Gabriel Babiano Huete</a> en 2002 para su Proyecto Fin de Carrera titulado:
<i>\"Estudio e implementación de una Autoridad de Certificación\"</i>."
                                                                                                                                +"<br><a
href=\"http://trajano.us.es\">Área de Ingeniería Telemática</a> perteneciente al <a href=\"http://www.esi.us.es/ISA\"> Departamento de
Ingeniería de Sistemas y Automática</a> en la <a href=\"http://www.esi.us.es\">Escuela Superior de Ingenieros</a> de la <a
href=\"http://www.us.es\"> Universidad de Sevilla</a>."
                                                                                                                                            +"</fon
t>"
                                                                                                                    +"</tr>"
                                                                                                                    +"</table>"
                                                                                                        +"</body>"
                                                                                            +"</html>");

                                                            }else{
```

```
                                                                //Si htMiIndex no contiene a email, es equivalente a decir que
email no tiene un certificado válido asociado
                                                                out.write("<html>"
                                                                                +"<head>"
                                                                                            +"<title>Autoridad de
Certificación: Error en la revocación</title>"
                                                                                            +"<meta content=\"\">"
                                                                                            +"<style></style>"
                                                                                +"</head>"
                                                                                +"<body>"
                                                                                +"<table>"
                                                                                            +"<th colspan=2
bgcolor=\"#9591f4\">"
                                                                                                        +"<center><font
size=+3><b><img src=\"/CA/logo_us.gif\" height=100 width=150 align=\"middle\">Autoridad de Certificación (CA)<img src=\"/CA/esi.gif\"
height=75 width=75 align=\"middle\"></font></b></center>"
                                                                                            +"</th>"
                                                                                            +"<tr>"
                                                                                                        +"<td
bgcolor=#FFF6D5>No se puede revocar el Certificado perteneciente a <b>"+email+"</b> porque no existe o porque ya se encuentra revocado."
                                                                                            +"<td bgcolor=#E6E6FF
width=\"40%\">"
                                                                                                        +"<h3>"
                                                                                                                    +"<a
href=\"/CA/index.html\">Página Principal</a>"

            +"<br><br><a href=\"/CA/cert_1.html\">Gestionar su certificado (firma o revocación)</a>"

            +"<br><br><a href=\"/CA/descargar.html\">Descargar certificados</a>"

            +"</h3>"
                                                                                                        +"</tr>"
                                                                                            +"<tr>"
                                                                                                        +"<td  colspan=2>"
                                                                                                                    +"<hr>"
                                                                                                                    +"<font
size=-1>Creado por <a href=\"mailto:GabrielBabiano@yahoo.es\">Gabriel Babiano Huete</a> en 2002 para su Proyecto Fin de Carrera titulado:
<i>\"Estudio e implementación de una Autoridad de Certificación\"</i>."
                                                                                                        +"<br><a
href=\"http://trajano.us.es\">Área de Ingeniería Telemática</a> perteneciente al <a href=\"http://www.esi.us.es/ISA\"> Departamento de
```

```
Ingeniería de Sistemas y Automática</a> en la <a href=\"http://www.esi.us.es\">Escuela Superior de Ingenieros</a> de la <a
href=\"http://www.us.es\"> Universidad de Sevilla</a>."
                                                                                                                                +"</fon
t>"
                                                                                                                +"</tr>"
                                                                                                                +"</table>"
                                                                                                    +"</body>"
                                                                                          +"</html>");
                                                                                }

                                                                        }else{
                                                                                //Si la clave de email no coincide
                                                                                out.write("<html>"
                                                                                          +"<head>"
                                                                                                    +"<title>Autoridad de Certificación: Clave
no válida</title>"
                                                                                                    +"<meta content=\"\">"
                                                                                                    +"<style></style>"
                                                                                          +"</head>"
                                                                                          +"<body>"
                                                                                          +"<table>"
                                                                                                    +"<th colspan=2 bgcolor=\"#9591f4\">"
                                                                                                              +"<center><font size=+3><b><img
src=\"/CA/logo_us.gif\" height=100 width=150 align=\"middle\">Autoridad de Certificación (CA)<img src=\"/CA/esi.gif\" height=75 width=75
align=\"middle\"></font></b></center>"
                                                                                                    +"</th>"
                                                                                                    +"<tr>"
                                                                                                              +"<td bgcolor=#FFF6D5>Ha
introducido una clave no válida. <a href=\"/CA/cert_1.html\">Pulse aquí para repetir el proceso</a>"
                                                                                                    +"<td bgcolor=#E6E6FF width=\"40%\">"
                                                                                                              +"<h3>"
                                                                                                                        +"<a
href=\"/CA/index.html\">Página Principal</a>"

                        +"<br><br><a href=\"/CA/cert_1.html\">Gestionar su certificado (firma o revocación)</a>"

                        +"<br><br><a href=\"/CA/descargar.html\">Descargar certificados</a>"
                                                                                                                        +"</h3>"
                                                                                                              +"</tr>"
                                                                                                    +"<tr>"
```

```
                                                                                +"<td  colspan=2>"
                                                                                          +"<hr>"
                                                                                          +"<font size=-
1>Creado por <a href=\"mailto:GabrielBabiano@yahoo.es\">Gabriel Babiano Huete</a> en 2002 para su Proyecto Fin de Carrera titulado:
<i>\"Estudio e implementación de una Autoridad de Certificación\"</i>."
                                                                                +"<br><a
href=\"http://trajano.us.es\">Área de Ingeniería Telemática</a> perteneciente al <a href=\"http://www.esi.us.es/ISA\"> Departamento de
Ingeniería de Sistemas y Automática</a> en la <a href=\"http://www.esi.us.es\">Escuela Superior de Ingenieros</a> de la <a
href=\"http://www.us.es\"> Universidad de Sevilla</a>."
                                                                                          +"</font>"
                                                                          +"</tr>"
                                                                          +"</table>"
                                                                +"</body>"
                                                      +"</html>");

                                          }

                                }else{
                                          //Si la dirección de email no está registrada
                                          out.write("<html>"
                                                                +"<head>"
                                                                          +"<title>Autoridad de Certificación: Email
no registrado</title>"
                                                                          +"<meta content=\"\">"
                                                                          +"<style></style>"
                                                                +"</head>"
                                                                +"<body>"
                                                                +"<table>"
                                                                          +"<th colspan=2 bgcolor=\"#9591f4\">"
                                                                                    +"<center><font size=+3><b><img
src=\"/CA/logo_us.gif\" height=100 width=150 align=\"middle\">Autoridad de Certificación (CA)<img src=\"/CA/esi.gif\" height=75 width=75
align=\"middle\"></font></b></center>"
                                                                          +"</th>"
                                                                          +"<tr>"
                                                                                    +"<td bgcolor=#FFF6D5>Ha
introducido un email no válido. <a href=\"/CA/cert_1.html\">Pulse aquí para repetir el proceso</a>"
                                                                          +"<td bgcolor=#E6E6FF width=\"40%\">"
                                                                                    +"<h3>"
                                                                                          +"<a
href=\"/CA/index.html\">Página Principal</a>"
```

```
          +"<br><br><a href=\"/CA/cert_1.html\">Gestionar su certificado (firma o revocación)</a>"

          +"<br><br><a href=\"/CA/descargar.html\">Descargar certificados</a>"
                                                                                                  +"</h3>"
                                                                                        +"</tr>"
                                                                              +"<tr>"
                                                                                        +"<td  colspan=2>"
                                                                                                  +"<hr>"
                                                                                                  +"<font size=-
1>Creado por <a href=\"mailto:GabrielBabiano@yahoo.es\">Gabriel Babiano Huete</a> en 2002 para su Proyecto Fin de Carrera titulado:
<i>\"Estudio e implementación de una Autoridad de Certificación\"</i>."
                                                                                        +"<br><a
href=\"http://trajano.us.es\">Área de Ingeniería Telemática</a> perteneciente al <a href=\"http://www.esi.us.es/ISA\"> Departamento de
Ingeniería de Sistemas y Automática</a> en la <a href=\"http://www.esi.us.es\">Escuela Superior de Ingenieros</a> de la <a
href=\"http://www.us.es\"> Universidad de Sevilla</a>."
                                                                                                  +"</font>"
                                                                              +"</tr>"
                                                                              +"</table>"
                                                                    +"</body>"
                                                          +"</html>");

                                   }

                                   out.close();


                         }catch(Exception e){
                                   try{
                                             UtilCA.stringToFile(UtilCA.fileToString(UtilCA.RUTA_FICHERO_LOG)+"\n"+new
Date()+"\t"+e.toString(),UtilCA.RUTA_FICHERO_LOG);
                                   }catch(Exception e2){
                                             //Si vuelve a ocurrir un error, se despreciará
                                   }
                         }
          }



          /**
```

```
             * Devuelve el nombre del servlet
             *
             * @return el nombre del servlet
             * @since CA v1.0
             */

            public java.lang.String getServletName(){
                        return "Revocar v1.0";
            }



            /**
             * Devuelve el nombre del servlet y el autor
             *
             * @return el nombre del servlet y el autor
             * @since CA v1.0
             */

            public String getServletInfo(){
                        return "Servlet Revocar v1.0 por Gabriel Babiano Huete (gabrielbabiano@yahoo.es)";
            }
}
```

UtilCA.java

```
//javac -classpath /opt/jakarta/common/lib/servlet.jar:/opt/jakarta/webapps/examples/WEB-INF/classes/ UtilCA.java

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.util.Hashtable;
import java.util.StringTokenizer;


/**
```

```
 * Métodos de utilidad para la CA
 *
 * @author Gabriel Babiano Huete (<a hfer="mailto:gabrielbabiano@yahoo.es">gabrielbabiano@yahoo.es</a>)
 * @version 1.0
 */
public class UtilCA{

	//Variables que contienen rutas de archivos y scripts
	/**
	 * Ruta del directorio base de openssl
	 * @since UtilCA v1.0
	 */
	public static String RUTA_DIR=new String("/usr/share/ssl/");

	/**
	 * Ruta del directorio de la CA
	 * @since UtilCA v1.0
	 */
	public static String RUTA_DIR_CA=new String(UtilCA.RUTA_DIR+"demoCA/");

	/**
	 * Ruta del directorio de los archivos que contienen los certificados
	 * @since UtilCA v1.0
	 */
	public static String RUTA_DIR_CA_FILES=new String(UtilCA.RUTA_DIR_CA+"files/");

	/**
	 * Ruta del fichero donde se guardan las parejas email/clave
	 * @since UtilCA v1.0
	 */
	public static String RUTA_FICHERO_EMAILS=new String(UtilCA.RUTA_DIR_CA+"emails.txt");

	/**
	 * Ruta del fichero donde se guardará el CSR del solicitante
	 * @since UtilCA v1.0
	 */
	public static String RUTA_FICHERO_CSR=new String(UtilCA.RUTA_DIR_CA+"CSR.pem");

	/**
	 * Ruta del fichero donde se guardará el Certificado del solicitante
```

```
             * @since UtilCA v1.0
             */
            public static String RUTA_FICHERO_LAST_CERT=new String(UtilCA.RUTA_DIR_CA+"lastCert.cert");

            /**
             * Ruta del fichero índice
             * @since UtilCA v1.0
             */
            public static String RUTA_FICHERO_INDEX=new String(UtilCA.RUTA_DIR_CA+"index.txt");

            /**
             * Ruta del fichero que contiene el número de serie del último certificado
             * @since UtilCA v1.0
             */
            public static String RUTA_FICHERO_SERIAL_OLD=new String(UtilCA.RUTA_DIR_CA+"serial.old");

            /**
             * Ruta del fichero que contiene mi índice
             * @since UtilCA v1.0
             */
            public static String RUTA_FICHERO_MI_INDEX=new String(UtilCA.RUTA_DIR_CA+"miIndex.txt");

            /**
             * Ruta del fichero donde escribiremos la pareja email/clave del usuario actual y que mandaremos a la dirección de email
especificada en el campo email
             * @since UtilCA v1.0
             */
            public static String RUTA_FICHERO_IMPORTANTE=new String(RUTA_DIR_CA+"importante.txt");

            /**
             * Ruta del fichero que contiene todos los certificados X.509 válidos (incluido en de la UtilCA) en formato PEM
             * @since UtilCA v1.0
             */
            public static String RUTA_FICHERO_ALL_VALID_CERTS=new String(RUTA_DIR_CA_FILES+"allValidCerts.pem");

            /**
             * Ruta del fichero donde almacenaremos el "diario de a bordo"
             * @since UtilCA v1.0
             */
            public static String RUTA_FICHERO_LOG=new String(UtilCA.RUTA_DIR_CA+"CA.log");
```

```
	/**
	 * Ruta del script certifica.sh
	 * @since UtilCA v1.0
	 */
	public static String SCRIPT_CERTIFICA=new String(UtilCA.RUTA_DIR+"certifica.sh");

	/**
	 * Ruta del script mandaEmail.sh
	 * @since UtilCA v1.0
	 */
	public static String SCRIPT_MANDA_EMAIL=new String(RUTA_DIR_CA+"mandaEmail.sh");

	/**
	 * Ruta del script revoca.sh
	 * @since UtilCA v1.0
	 */
	public static String SCRIPT_REVOCA=new String(UtilCA.RUTA_DIR+"revoca.sh");

	/**
	 * Ruta del script actualizaCA.sh
	 * @since UtilCA v1.0
	 */
	public static String SCRIPT_ACTUALIZA_CA=new String(UtilCA.RUTA_DIR+"actualizaCA.sh");

	/**
	 * Ruta del script que creaDER.sh
	 * @since UtilCA v1.0
	 */
	public static String SCRIPT_CREA_DER=new String(UtilCA.RUTA_DIR+"creaDER.sh");

	/**
	 * Ruta del script borraFichero.sh
	 * @since UtilCA v1.0
	 */
	public static String SCRIPT_BORRA_FICHERO=new String(UtilCA.RUTA_DIR_CA+"borraFichero.sh");



	/**
```

```
	 * Obtiene una Hashtable a partir de un fichero
	 *
	 * @param nombreFichero ruta del fichero del que deseamos extraer la Hastable
	 * @return la Hashtable
	 * @throws Exception excepción producida en la ejecución del código
	 * @since UtilCA v1.0
	 */

	protected static Hashtable fileToHashtable(String nombreFichero) throws Exception{

			return fileToHashtable(new File(nombreFichero));

	}




	/**
	 * Obtiene una Hashtable a partir de un fichero
	 *
	 * @param fichero fichero del que deseamos extraer la Hastable
	 * @return la Hashtable
	 * @throws Exception excepción producida en la ejecución del código
	 * @since UtilCA v1.0
	 */

	protected static Hashtable fileToHashtable(File fichero) throws Exception{

			//Creamos la Hashtable que devolveremos
			Hashtable ht = new Hashtable();
			//Si el fichero no existe, se devuelve la Hashtable vacía
			if(fichero.createNewFile()==false){
					//Comprobamos si podemos leer desde el fichero
					if(fichero.canRead()==false){
							//En caso contrario lanzaremos una Excepción
							throw new Exception("No se puede leer el fichero: "+fichero);
					}
					FileInputStream canalEntrada=new FileInputStream(fichero);
					int caracteresDisponibles=canalEntrada.available();
```

```
                        char charLeido;
                        String cadenaFichero=new String();  //String que contendrá todo el archivo
                        //Lo leeremos de char en char y lo vamos añadiendo al final
                        for(int i=1; i<=caracteresDisponibles; i++){
                                    charLeido=(char) canalEntrada.read();
                                    cadenaFichero+=charLeido;
                        }

                        //Crearemos un SrtingTokenizer que tenga como separadores entre tokens: '{', '}', '=', ',' y ' '
                        StringTokenizer st=new StringTokenizer(cadenaFichero,"{}=, ");
                        //Pasaremos dichos tokens a la HashTable
                        while(st.countTokens()>=2){
                                    ht.put(st.nextToken(),st.nextToken());
                        }

                        canalEntrada.close();
            }
            return ht;
}



/**
 * Copia una Hashtable a un fichero
 *
 * @param ht Hashtable que deseamos copiar al fichero
 * @param nombreFichero ruta del fichero al que deseamos copiar la Hastable
 * @throws Exception excepción producida en la ejecución del código
 * @since UtilCA v1.0
 */
public static void hashtableToFile(Hashtable ht, String nombreFichero) throws Exception{

            hashtableToFile(ht,new File(nombreFichero));

}


/**
 * Copia una Hashtable a un fichero
 *
```

```
	 * @param ht Hashtable que deseamos copiar al fichero
	 * @param fichero fichero al que deseamos copiar la Hastable
	 * @throws Exception excepción producida en la ejecución del código
	 * @since UtilCA v1.0
	 */
	public static void hashtableToFile(Hashtable ht, File fichero) throws Exception{

		String aFichero=ht.toString();	//Cadena que contendrá lo que se va a pasar a miFichero
		byte b[]=aFichero.getBytes();	//Para ello se necesita pasar a un array de byte's
		FileOutputStream canalSalida=new FileOutputStream(fichero); //lo que escribamos al canalSalida, pasará a miFichero
		canalSalida.write(b);		//Escribimos el array en el canalSalida
		canalSalida.close();		//Cerramos el canalSalida

	}



	/**
	 * Copia todo el contenido de un fichero a una cadena
	 *
	 * @param nombreFichero ruta del fichero del que deseamos copiar su contenido
	 * @return una cadena con todo el contenido del fichero
	 * @throws Exception excepción producida en la ejecución del código
	 * @since UtilCA v1.0
	 */

	public static String fileToString(String nombreFichero) throws Exception{
		return fileToString(new File(nombreFichero));
	}



	/**
	 * Copia todo el contenido de un fichero a una cadena
	 *
	 * @param fichero fichero del que deseamos copiar su contenido
	 * @return una cadena con todo el contenido del fichero
	 * @throws Exception excepción producida en la ejecución del código
	 * @since UtilCA v1.0
	 */
```

```
public static String fileToString(File fichero) throws Exception{
            FileReader fr=new FileReader(fichero);
            char c;                 //Contendrá el char leido de fr
            int ic=fr.read();//Contendrá el valor int del caracter leido de fr (ic=-1 en caso de caracter nulo)
            String cadena=new String();
            while(ic!=-1){          //Continuamos en el bucle mientras no haya nada que leer
                        c=(char)ic;
                        cadena+=c;
                        ic=fr.read();
            }
            return cadena;
}



/**
 * Copia una cadena a un fichero
 *
 * @param cadena cadena que copia al fichero
 * @param nombreFichero ruta del fichero del que deseamos copiar su contenido
 * @throws Exception excepción producida en la ejecución del código
 * @since UtilCA v1.0
 */

public static void stringToFile(String cadena,String nombreFichero) throws Exception{
            stringToFile(cadena, new File(nombreFichero));
}



/**
 * Copia una cadena a un fichero
 *
 * @param cadena cadena que copia al fichero
 * @param fichero fichero del que deseamos copiar su contenido
 * @throws Exception excepción producida en la ejecución del código
 * @since UtilCA v1.0
 */
```

```
	public static void stringToFile(String cadena,File fichero) throws Exception{
		FileOutputStream fos=new FileOutputStream(fichero);
		byte bCadena[]=cadena.getBytes();
		fos.write(bCadena);
		fos.close();
	}
}
```

admin.html

```
<html>
  <head>
    <title></title>
    <meta content="">
    <style></style>
  </head>
  <body>
	<center><h1>Manual de administrador <br>de la <br>Autoridad de Certificación (CA) v1.0</h1></center>
	<hr>
	<h2><a name="indice">Índice</a></h2>
				<ul>
				<li><a href="#indice">Índice</a>
				<li><a href="#descripcion">Descripción</a>
				<li><a href="#caracteristicas">Características</a>
				<li><a href="#software">Software necesario</a>
				<li><a href="#instalacion">Instalación</a>
				<li><a href="#configuracion">Configuración</a>
				<li><a href="#uso">Uso</a>
					<ul>
						<li><a href="#arranque">Arranque</a>
						<li><a href="#parada">Parada</a>
						<li><a href="#reinicializado">Reinicializado</a>
					</ul>
				<li><a href="#desinstalacion">Desinstalación</a>
		</ul>

		<hr>
```

```
                        <h2><a name="descripcion">Descripción</a></h2>
                                    <p>Conjunto de páginas HTML y servlets escritos en Java que via HTTPS tratan de implementar una
autoridad de certificación. Se acompañan de una serie de shell scripts comentados para la simplificación de las tareas del administrador.
Está basado en utilidades proporcionadas por el Proyecto OpenSSL.
                                    <br>Creado por Gabriel Babiano Huete (<a
href="mailto:gabrielbabiano@yahoo.es">gabrielbabiano@yahoo.es</a>) para su Proyecto Fin de Carrera titulado "Estudio e implementación de
una Autoridad de Certificación".
                                    </p>
                        <h2><a name="caracteristicas">Características</a></h2>
                                    <p>
                                    <ul>Esta autoridad implementa las siguientes <b>funciones</b>:
                                                <li>Certificación inmediata de Peticiones de Firma de Certificado (CSR) según el estándar
PKCS#10. Sólo tendrá validez sólo un certificado por dirección de correo electrónico y por DistiguishedName (DN).
                                                <li>Revocación de certificados firmados por esta autoridad. La Lista de Certificados
Revocados (CRL) se actualizará automáticamente tras cada revocación.
                                                <li>Búsqueda y entrega de certificados válidos de usuarios. Disponible según el estándar
X.509 o PKCS#7. Tanto en codificación PEM como DER. Tanto en archivo como directamente al navegador.
                                                <li>Entrega del certificado raiz de la autoridad. Disponible según el estándar X.509 o
PKCS#7. Tanto en codificación PEM como DER. Tanto en archivo como directamente al navegador.
                                                <li>Entrega de la Lista de Certificados Revocados (CRL) actualizada hasta el momento.
Disponible según el estándar X.509 o PKCS#7. Tanto en codificación PEM como DER. Tanto en archivo como directamente al navegador.
                                                <li>Entrega de la colección de certificados válidos en esta autoridad de certificación
incluido el certificado raiz e la autoridad de certificación junto con la CRL actualizada según el estándar PKCS#7. Tanto en codificación
PEM como DER. Tanto en archivo como directamente al navegador.
                                    </ul>
                                    La certificación de la identidad del usuario está limitada a la de su dirección de correo electrónico
y esta autoridad no valida los datos incluidos en el DistiguishedName (DN) del certificado (de forma similar a los antiguos certificados
de clase 1 de VeriSign). Dicha certificación de la dirección de correo electrónico se lleva a cabo con la entrega de una clave via email a
la dirección de correo electrónico que pretende certificar su CSR.
                                    <ul>La <b>política de las claves</b> es la siguiente:
                                                <li>Para asegurar que quien pretende tanto que la autoridad le firme su CSR como la
revocación de un certificado válido, la autoridad genera en cada ocasión una clave de tipo numérico y la remite instantáneamente a la
dirección de correo electrónico que lo solicita.
                                                <li>Por cada intento de certificación o de revocación (se lleve a cabo o no) se generará y
enviará una clave distinta. La única clave válida será la última recibida en dicha dirección de correo electrónico.
                                    </ul>
                                    Esta aplicación ha sido testada sobre una distribución SuSE Linux 8.0 Proffesional con los directorios
por defecto.
                                    </p>
                        <hr>
```

```
                        <h2><a name="software">Software necesario</a></h2>
                                    <ul>
                                                <li>OpenSSL 0.9.6c-29 (procedente de la distribución SuSE Linux 8.0 Proffesional)
                                                <li>Apache httpd 1.3.23-73 (procedente de la distribución SuSE Linux 8.0 Proffesional)
                                                <li>mod-ssl 2.8.7-41 (procedente de la distribución SuSE Linux 8.0 Proffesional)
                                                <li>jakarta-tomcat 4.0.1-1227 (procedente de la distribución SuSE Linux 8.0 Proffesional)
                                                <li>mutt 1.3.27i-62 (procedente de la distribución SuSE Linux 8.0 Proffesional)
                                                <li>Java 2 v1.4.0 JRE (Java Runtime Enviroment) o SDK (Standard Development Kit)
(procedente de http://java.sun.com)
                                                <li>jce_policy-1_4_0.zip (procedente de http://java.sun.com/products/jce/index-14.html)
                                    </ul>

                        <hr>
                        <h2><a name="instalacion">Instalación</a></h2>
                                    <ol>
                                                <li>Instalar la distribución SuSE Linux 8.0 Proffesional con los mencionados paquetes en
los directorios por defecto.
                                                <li>Instalar J2SDK 1.4.0_02
                                                <li>Instalar "jce_policy-1_4_0.zip"
                                    </ol>

                        <hr>
                        <h2><a name="configuracion">Configuración</a></h2>
                                    <p>Ejecutar en la línea de comandos:</p>
                                    <table BGCOLOR="#e6e6ff">
                                                <td><code>
                                                            gab:~ #<b>instalarCA.sh</b>
                                                </td></code>
                                    </table>
                                    <ul>Dicho script:
                                                <li>Hace copia de seguridad de los archivos que se pudieran sobreescribir
                                                <li>Configura OpenSSL
                                                <li>Configura el servidor jakarta-tomcat
                                                <li>Copia los archivos y directorios necesarios para la CA
                                    </ul>
                                    <p>Copiar al directorio /usr/share/ssl/demoCA/ una serie de archivos que serán de utilidad a la hora
de generar números pseudo-aleatorios (ver página del manual sobre openssl genrsa). Es válido cualquier tipo de archivo, pero se recomienda
archivos comprimidos (ya que así, poseen menos redundancia), y aunque el manual no indica el tamaño máximo o mínimo de estos archivos,
funciona bien con ficheros de entre 2 y 5 Mbytes. En el CD del proyecto no se adjuntan dichos archivos para que sea el propio usuario el
que los elija o cree, para aumentar de esta forma la "aleatoriedad".
```

```
                                    <p>Suponemos que copiamos al directorio /usr/share/ssl/demoCA/ 3 archivos y los renombramos a: file1,
file2 y file3. Para copiar archivos y renombrarlos a la vez, podemos utilizar el comando cp en la línea de comandos (cp archivo_origen
archivo_destino), aunque si la ruta de los archivos es un tanto complicada, sería más recomendable el uso de cualquier utilidad para este
fin: mc (Midnight Comander) en la propia línea de comandos o konqueror (similar al explorador de Microsoft Windows) en X-Windows, por
ejemplo.</p>
                                    <p>Generación de la clave privada RSA (de 2048 bits) de la propia CA (cakey.pem):</p>
                                    <table BGCOLOR="#e6e6ff">
                                              <td><code>
                                              gab:~ # <b>openssl genrsa -rand file1:file2:file3 2048 >
/usr/share/ssl/demoCA/private/cakey.pem</b>
                                              <br>9223693 semi-random bytes loaded
                                              <br>Generating RSA private key, 2048 bit long modulus
                                              <br>.....................................................++
                                              <br>.....................................................................................++
                                              <br>e is 65537 (0x10001)
                                              </td></code>
                                    </table>
                                    <p>NOTA: en cuanto al tamaño de la clave privada tenemos que tener en cuenta que por defecto, openssl
toma 512 bits. Sin la política de JCE sólo se permite criptografía fuerte (claves de hasta 1024 bits). Con la política de JCE de
criptografía ilimitada, keytool llega a admitir una longitud de clave máxima de 2048 bits. Se ha utilizado un tamaño de clave de 2048
bits. En la siguiente tabla se muestra el tiempo de proceso que se requirió para los distintos tamaños de clave privada. Observar que a
partir de un tamaño de clave de aproximadamente 8192 bits, el tiempo que requiere la creación de la clave privada se haría
inadmisible.</p>
                                    <TABLE WIDTH=100% BORDER=1 BORDERCOLOR="#000000" CELLPADDING=4 CELLSPACING=0>
                                              <COL WIDTH=128*>
                                              <COL WIDTH=128*>
                                              <THEAD>
                                                        <TR VALIGN=TOP>
                                                                  <TH WIDTH=50% BGCOLOR="#9999ff">
                                                                            <P CLASS="western">Tama&ntilde;o (bits)</P>
                                                                  </TH>
                                                                  <TH WIDTH=50% BGCOLOR="#9999ff">
                                                                            <P CLASS="western">Tiempo (hh:mm:ss)</P>
                                                                  </TH>
                                                        </TR>
                                              </THEAD>
                                              <TBODY>
                                                        <TR>
                                                                  <TD WIDTH=50% VALIGN=TOP BGCOLOR="#ccccff">
```

```
                                                        <P ALIGN=CENTER>2<SUP>10</SUP>=1024</P>
                                                    </TD>
                                                    <TD WIDTH=50% VALIGN=BOTTOM SDVAL="0,0000115740740740741"
SDNUM="1034;0;HH:MM:SS">
                                                        <P ALIGN=CENTER>00:00:01</P>
                                                    </TD>
                                                </TR>
                                                <TR>
                                                    <TD WIDTH=50% VALIGN=TOP BGCOLOR="#ccccff">
                                                        <P ALIGN=CENTER>2<SUP>11</SUP>=2048</P>
                                                    </TD>
                                                    <TD WIDTH=50% VALIGN=BOTTOM SDVAL="0,0000231481481481481"
SDNUM="1034;0;HH:MM:SS">
                                                        <P ALIGN=CENTER>00:00:02</P>
                                                    </TD>
                                                </TR>
                                                <TR>
                                                    <TD WIDTH=50% VALIGN=TOP BGCOLOR="#ccccff">
                                                        <P ALIGN=CENTER>2<SUP>12</SUP>=4096</P>
                                                    </TD>
                                                    <TD WIDTH=50% VALIGN=BOTTOM SDVAL="0,000057870370370 4"
SDNUM="1034;0;HH:MM:SS">
                                                        <P ALIGN=CENTER>00:00:05</P>
                                                    </TD>
                                                </TR>
                                                <TR>
                                                    <TD WIDTH=50% VALIGN=TOP BGCOLOR="#ccccff">
                                                        <P ALIGN=CENTER>2<SUP>13</SUP>=8192</P>
                                                    </TD>
                                                    <TD WIDTH=50% VALIGN=BOTTOM SDVAL="0,00405092592592593"
SDNUM="1034;0;HH:MM:SS">
                                                        <P ALIGN=CENTER>00:05:50</P>
                                                    </TD>
                                                </TR>
                                                <TR>
                                                    <TD WIDTH=50% VALIGN=TOP BGCOLOR="#ccccff">
                                                        <P ALIGN=CENTER>2<SUP>14</SUP>=16384</P>
                                                    </TD>
                                                    <TD WIDTH=50% VALIGN=BOTTOM SDVAL="0,0140046296296296"
SDNUM="1034;0;HH:MM:SS">
```

```
                                                                    <P ALIGN=CENTER>00:20:10</P>
                                                                </TD>
                                                            </TR>
                                                    </TBODY>
                                            </TABLE>
                                            <p>NOTA: se podría añadir la opción -des3 y encriptar la clave privada con triple DES al dejarla en el
disco duro. Al incluir esta opción, se aumenta en seguridad, pero también es un gran inconveniente ya que cada vez que utilicemos la clave
privada, como por ejemplo firmar las CSR, deberemos de introducir dicha clave, lo cual es muy engorroso y no siempre es posible.</p>

                                            <p>Se desea ser la CA raiz por lo que se debe autofirmar nuestra clave pública. El comando que nos
permite autofirmarnos (con nuestra clave privada) la clave pública y obtener cacert.pem es:</p>
                                            <table BGCOLOR="#e6e6ff">
                                                    <td><code>
                                                    gab:~ # <b>openssl req -new -x509 -key /usr/share/ssl/demoCA/private/cakey.pem -days 365 >
/usr/share/ssl/demoCA/cacert.pem</b>
                                                    <br>You are about to be asked to enter information that will be incorporated
                                                    <br>into your certificate request.
                                                    <br>What you are about to enter is what is called a Distinguished Name or a DN.
                                                    <br>There are quite a few fields but you can leave some blank
                                                    <br>For some fields there will be a default value,
                                                    <br>If you enter '.', the field will be left blank.
                                                    <br>-----
                                                    <br>Country Name (2 letter code) [AU]:<b>ES</b>
                                                    <br>State or Province Name (full name) [Some-State]:<b>Spain</b>
                                                    <br>Locality Name (eg, city) []:<b>Sevilla</b>
                                                    <br>Organization Name (eg, company) [Internet Widgits Pty Ltd]:<b>Universidad de
Sevilla</b>
                                                    <br>Organizational Unit Name (eg, section) []:<b>Departamento de Telematica</b>
                                                    <br>Common Name (eg, YOUR name) []:<b>Autoridad de Certificacion (CA)</b>
                                                    <br>Email Address []:<b>root@192.168.0.3</b>
                                                    </td></code>
                                            </table>
                                            <p>NOTA: en stateProvinceName se ha introducido Spain en lugar de España y que no se ha utilizado las
vocales tildadas en ningún campo. Si hubiésemos introducido España, al contener una ñ, no habría problemas para el keytool, pero sí para
openssl ca ya que detectaría un caracter ilegal en el tipo ASN.1. Se ha intentado corregir esta situación, modificando a cualquiera de las
opciones de la máscara del archivo de configuración situado por defecto en /usr/share/ssl/openssl.cnf:</p>
                                            <table BGCOLOR="#e6e6ff">
                                                    <td><code>
                                                    # This sets a mask for permitted string types. There are several options.
                                                    <br># default: PrintableString, T61String, BMPString.
```

```
                                        <br># pkix   : PrintableString, BMPString.
                                        <br># utf8only: only UTF8Strings.
                                        <br># nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
                                        <br># MASK:XXXX a literal mask value.
                                        <br># WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
                                        <br># so use this option with caution!
                                        <br>string_mask = nombstr
                                        </td></code>
                                </table>
                                <p>Pero sin lograr el resultado esperado.</p>

                                <p>Inicializar la CA ejecutando:</p>
                                <table BGCOLOR="#e6e6ff">
                                        <td><code>
                                                gab:~ # <b>/usr/share/ssl/iniciaCA.sh</b>
                                                <br>rm: cannot remove `/usr/share/ssl/demoCA/importante.txt': no such file or
directory
                                                <br>rm: cannot remove `/usr/share/ssl/demoCA/emails.txt': no such file or
directory
                                                <br>rm: cannot remove `/usr/share/ssl/demoCA/serial*': no such file or
directory
                                                <br>rm: cannot remove `/usr/share/ssl/demoCA/index.txt*': no such file or
directory
                                                <br>rm: cannot remove `/usr/share/ssl/demoCA/miIndex.txt': no such file or
directory
                                                <br>rm: cannot remove `/usr/share/ssl/demoCA/newcerts/*': no such file or
directory
                                                <br>rm: cannot remove `/usr/share/ssl/demoCA/files/*': no such file or
directory
                                                <br>rm: cannot remove `/usr/share/ssl/demoCA/importante.txt: no such file or
directory
                                                <br>Using configuration from /usr/share/ssl/opnessl.cnf
                                        </td></code>
                                </table>

                                <p>Para poder utilizar el protocolo HTTPS, hay que crear un par de claves RSA (en este caso de 2048
bits) en un keystore con el alias tomcat:</p>
                                <table BGCOLOR="#e6e6ff">
                                        <td><code>
                                        gab:~ # <b>/usr/java/j2sdk1.4.0_02/bin/keytool -genkey -alias tomcat -keyalg RSA -keystore
```

```
/opt/jakarta/conf/.keystore -keysize 2048</b>
                                        <br>Escriba la contraseña del almacén de claves:  <b>miclave</b>
                                        <br>¿Cuáles son su nombre y su apellido?
                                        <br>  [Unknown]:  <b>192.168.0.3</b>
                                        <br>¿Cuál es el nombre de su unidad de organización?
                                        <br>  [Unknown]:  <b>Departamento de Telematica</b>
                                        <br>¿Cuál es el nombre de su organización?
                                        <br>  [Unknown]:  <b>Universidad de Sevilla</b>
                                        <br>¿Cuál es el nombre de su ciudad o localidad?
                                        <br>  [Unknown]:  <b>Sevilla</b>
                                        <br>¿Cuál es el nombre de su estado o provincia?
                                        <br>  [Unknown]:  <b>Spain</b>
                                        <br>¿Cuál es el código de país de dos letras de la unidad?
                                        <br>  [Unknown]:  <b>ES</b>
                                        <br>¿Es correcto CN=192.168.0.3, OU=Departamento de Telematica, O=Universidad de Sevilla,
L=Sevilla, ST=Spain, C=ES?
                                        <br>  [no]:  <b>y</b>
                                        <br>
                                        <br>Escriba la contraseña clave para <tomcat>
                                        <br>        (INTRO si es la misma
                                        </td></code>
                                </table>

                                <p>Notar que cuando se pregunta el nombre y apellido, en realidad  se nos pregunta acerca del
CommonName (CN) y ahí hemos de introducir el nombre de la máquina a la que se accede de forma "literal", es decir, que es distinto poner
192.168.0.3, www.us.es o us.es, por ejemplo.</p>
                                <p>Para crear a partir de la CSR X.509 en formato PEM siguiendo el estándar PKCS#10:</p>
                                <table BGCOLOR="#e6e6ff">
                                        <td><code>
                                        gab:~ # <b>/usr/java/j2sdk1.4.0_02/bin/keytool -certreq -alias tomcat -keystore
/opt/jakarta/conf/.keystore -file /opt/jakarta/conf/tomcat.csr</b>
                                        <br>Escriba la contraseña del almacén de claves:  <b>miclave</b>
                                        </td></code>
                                </table>
                                <p>En nuestro caso, y a falta de una CA raiz que nos firme el CSR la firmará nuestra propia CA de
forma manual.</p>
                                <table BGCOLOR="#e6e6ff">
                                        <td><code>
                                                gab: # <b>cd /usr/share/ssl</b>
                                                <br>gab:/usr/share/ssl # <b>openssl ca -in /opt/jakarta/conf/tomcat.csr -out
```

```
/opt/jakarta/conf/tomcat.cer -notext</b>
                                                <br>Using configuration from /usr/share/ssl/openssl.cnf
                                                <br>Check that the request matches the signature
                                                <br>Signature ok
                                                <br>The Subjects Distinguished Name is as follows
                                                <br>countryName           :PRINTABLE:'ES'
                                                <br>stateOrProvinceName   :PRINTABLE:'Spain'
                                                <br>localityName          :PRINTABLE:'Sevilla'
                                                <br>organizationName      :PRINTABLE:'Universidad de Sevilla'
                                                <br>organizationalUnitName:PRINTABLE:'Departamento de Telematica'
                                                <br>commonName            :PRINTABLE:'192.168.0.3'
                                                <br>Certificate is to be certified until Nov 17 08:56:53 2003 GMT (365 days)
                                                <br>Sign the certificate? [y/n]:<b>y</b>
                                                <br>
                                                <br>
                                                <br>1 out of 1 certificate requests certified, commit? [y/n]<b>y</b>
                                                <br>Write out database with 1 new entries
                                                <br>Data Base Updated
                                        </td></code>
                                </table>

                                <p>Instalamos el certificado de la Autoridad de Certificación (CA) en el keystore</p>
                                <table BGCOLOR="#e6e6ff">
                                        <td><code>
                                        gab:/usr/share/ssl # <b>/usr/java/j2sdk1.4.0_02/bin/keytool -import -alias ca  -keystore
/opt/jakarta/conf/.keystore -file /usr/share/ssl/demoCA/cacert.pem</b>
                                        <br>Escriba la contraseña del almacén de claves:  <b>miclave</b>
                                        <br>Propietario: EMAILADDRESS=root@192.168.0.3, CN=Autoridad de Certificacoin (CA),
OU=Departamento de Telematica, O=Universidad de Sevilla, L=Sevilla, ST=Spain, C=ES
                                        <br>Emisor: EMAILADDRESS=root@192.168.0.3, CN=Autoridad de Certificacoin (CA),
OU=Departamento de Telematica, O=Universidad de Sevilla, L=Sevilla, ST=Spain, C=ES
                                        <br>Número de serie: 0
                                        <br>Válido desde: Sun Nov 17 16:31:38 CET 2002 hasta: Mon Nov 17 16:31:38 CET 2003
                                        <br>Huellas digitales del certificado:
                                        <br>MD5:  ED:ED:D8:DB:F4:84:53:58:6E:DC:37:7E:A8:8F:53:6B
                                <br>SHA1: 7E:6A:D5:09:69:64:D2:93:28:4E:1C:51:B9:8F:DF:BB:DE:EF:43:8B
                                        <br>¿Confiar en este certificado? [no]:  <b>y</b>
                                        <br>Se ha añadido el certificado al almacén de claves
                                        </td></code>
                                </table>
```

```
                                    <p>Una firmada la CSR por la CA, obtenemos el certificado (/opt/jakarta/conf/tomcat.cer).</p>
                                    <p>Se importa el certificado a la keystore bajo el mismo alias de tomcat:</p>
                                    <table BGCOLOR="#e6e6ff">
                                                <td><code>
                                                gab:~ # <b>/usr/java/j2sdk1.4.0_02/bin/keytool -import -alias tomcat -file
/opt/jakarta/conf/tomcat.cer -keystore /opt/jakarta/conf/.keystore</b>
                                                <br>Escriba la contraseña del almacén de claves:  <b>miclave</b>
                                                Se ha añadido el certificado al almacén de claves
                                                </td></code>
                                    </table>

                                    <p>Configurar el firewall (si se va a activar) para poder permitir los servicios HTTP y HTTPS hacia el
exterior, en caso que se desee hacerlo.</p>
                                    <p>Comprobar la correcta configuración en el navegador dirigiéndonos por el protocolo HTTPS (puerto
443) al directorio /CA/ de la máquina en que instalado. En este ejemplo nos dirigiríamos hacia:</p>
                                    <table BGCOLOR="#e6e6ff">
                                                <td><code>
                                                gab:~ # <b>mozilla https://192.168.0.3/CA</b>
                                                </td></code>
                                    </table>

                        <hr>
                        <h2><a name="uso">Uso</a></h2>
                                    <h3><a name="arranque">Arranque</a></h3>
                                                <p>Para arrancar el servidor jakarta se debe ejecutar:</p>
                                                <table BGCOLOR="#e6e6ff">
                                                            <td><code>
                                                                        gab:~ # <b>setDefaultJava --devel SunJava2</b>
                                                                        <br>gab:~ # <b>source setJava --devel SunJava2</b>
                                                                        <br>gab:~ # <b>/opt/jakarta/bin/startup.sh</b>
                                                                        <br>Guessing CATALINA_HOME from catalina.sh to /opt/jakarta/bin/..
                                                                        <br>Setting CATALINA_HOME to /opt/jakarta/bin/..
                                                                        <br>Using CLASSPATH:
/opt/jakarta/bin/../bin/bootstrap.jar:/opt/jakarta/bin/../common/lib/activation.jar:/opt/jakarta/bin/../common/lib/jdbc2_0-
stdext.jar:/opt/jakarta/bin/../common/lib/jndi.jar:/opt/jakarta/bin/../common/lib/jta.jar:/opt/jakarta/bin/../common/lib/mail.jar:/opt/jak
arta/bin/../common/lib/naming-common.jar:/opt/jakarta/bin/../common/lib/naming-
resources.jar:/opt/jakarta/bin/../common/lib/servlet.jar:/opt/jakarta/bin/../common/lib/tyrex-
0.9.7.0.jar:/opt/jakarta/bin/../common/lib/xerces.jar:/opt/jakarta/bin/../server/lib/catalina.jar:/opt/jakarta/bin/../server/lib/jakarta-
regexp-1.2.jar:/opt/jakarta/bin/../server/lib/servlets-common.jar:/opt/jakarta/bin/../server/lib/servlets-
```

```
default.jar:/opt/jakarta/bin/../server/lib/servlets-invoker.jar:/opt/jakarta/bin/../server/lib/servlets-
manager.jar:/opt/jakarta/bin/../server/lib/servlets-snoop.jar:/opt/jakarta/bin/../server/lib/servlets-
webdav.jar:/opt/jakarta/bin/../server/lib/tomcat-ajp.jar:/opt/jakarta/bin/../server/lib/tomcat-
util.jar:/opt/jakarta/bin/../server/lib/warp.jar:/opt/jakarta/bin/../lib/jasper-compiler.jar:/opt/jakarta/bin/../lib/jasper-
runtime.jar:/opt/jakarta/bin/../lib/naming-factory.jar:/usr/java/j2sdk1.4.0_02/lib/tools.jar
                                                                <br>Using CATALINA_BASE: /opt/jakarta/bin/..
                                                                <br>Using CATALINA_HOME: /opt/jakarta/bin/..
                                                                <br>Using JAVA_HOME:     /usr/java/j2sdk1.4.0_02
                                                        </td></code>
                                                </table>

                                        <h3><a name="parada">Parada</a></h3>
                                                <p>Para parar el servidor jakarta:</p>
                                                <table BGCOLOR="#e6e6ff">
                                                        <td><code>
                                                                gab:~ # <b>/opt/jakarta/bin/shutdown.sh</b>
                                                                <br>Guessing CATALINA_HOME from catalina.sh to /opt/jakarta/bin/..
                                                                <br>Setting CATALINA_HOME to /opt/jakarta/bin/..
                                                                <br>Using CLASSPATH:
/opt/jakarta/bin/../bin/bootstrap.jar:/opt/jakarta/bin/../common/lib/activation.jar:/opt/jakarta/bin/../common/lib/jdbc2_0-
stdext.jar:/opt/jakarta/bin/../common/lib/jndi.jar:/opt/jakarta/bin/../common/lib/jta.jar:/opt/jakarta/bin/../common/lib/mail.jar:/opt/jak
arta/bin/../common/lib/naming-common.jar:/opt/jakarta/bin/../common/lib/naming-
resources.jar:/opt/jakarta/bin/../common/lib/servlet.jar:/opt/jakarta/bin/../common/lib/tyrex-
0.9.7.0.jar:/opt/jakarta/bin/../common/lib/xerces.jar:/opt/jakarta/bin/../server/lib/catalina.jar:/opt/jakarta/bin/../server/lib/jakarta-
regexp-1.2.jar:/opt/jakarta/bin/../server/lib/servlets-common.jar:/opt/jakarta/bin/../server/lib/servlets-
default.jar:/opt/jakarta/bin/../server/lib/servlets-invoker.jar:/opt/jakarta/bin/../server/lib/servlets-
manager.jar:/opt/jakarta/bin/../server/lib/servlets-snoop.jar:/opt/jakarta/bin/../server/lib/servlets-
webdav.jar:/opt/jakarta/bin/../server/lib/tomcat-ajp.jar:/opt/jakarta/bin/../server/lib/tomcat-
util.jar:/opt/jakarta/bin/../server/lib/warp.jar:/opt/jakarta/bin/../lib/jasper-compiler.jar:/opt/jakarta/bin/../lib/jasper-
runtime.jar:/opt/jakarta/bin/../lib/naming-factory.jar:/usr/java/j2sdk1.4.0_02/lib/tools.jar
                                                                <br>Using CATALINA_BASE: /opt/jakarta/bin/..
                                                                <br>Using CATALINA_HOME: /opt/jakarta/bin/..
                                                                <br>Using JAVA_HOME:     /usr/java/j2sdk1.4.0_02
                                                        </td></code>
                                                </table>

                                        <h3><a name="reinicializado">Reinicializado</a></h3>
                                                <p>Para el reinicializado de la CA podemos ejecutar:</p>
                                                <table BGCOLOR="#e6e6ff">
```

```
                                                <td><code>
                                                        gab:~ # <b>/usr/share/ssl/iniciaCA.sh</b>
                                                </td></code>
                                        </table>
                                        <ul>     Dicho script:
                                                <li>Borra los certificado emitidos, la CRL antigua y los fichero índice.
                                                <li>Mantiene la clave privada y el certificado de la CA.
                                                <li>Vuelve a crear una CRL vacía.
                                                <li>El número de serie vuelve a ser 01.
                                        </ul>
                                        <p>NOTA: esta operación es definitiva y tras el borrado, no se pueden recuperar los datos
tras el reinicio</p>

                <hr>
                <h2><a name="desinstalacion">Desinstalación</a></h2>
                        <p>Para desinstalar la Autoridad de Certificación (CA) se ejecutará:</p>
                        <table BGCOLOR="#e6e6ff">
                                <td><code>
                                        <b>/usr/share/ssl/desinstalarCA.sh</b>
                                </td></code>
                        </table>

        </body>
</html>
```

borraFichero.sh

```
#!/bin/sh
# ------------------------------------------------------------------------
# borraFichero.sh - Script que borra el fichero pasado como parámetro ($1).
#
# $Id: borraFichero.sh,v 1.0
# ------------------------------------------------------------------------

#Borramos el fichero pasado como parámetro ($1)
rm $1
```

mandaEmail.sh

```
#!/bin/sh
# ------------------------------------------------------------------------
# mandaEmail.sh - Script que manda un email con el asunto: "Mensaje desde la CA" a $1 con el fichero $2 adjunto.
#
# $Id: mandaEmail.sh,v 1.0 2002/09/27 13:22:16 $
# ------------------------------------------------------------------------

#Enviamos el email con mutt con el asunto "Mensaje desde la CA" a $1 con el fichero $2 adjunto
/usr/bin/mutt -s "Mensaje desde la CA" $1 < $2
```

actualizaCA.sh

```
#!/bin/sh
# ------------------------------------------------------------------------
# actualizaCA.sh - Script que actualiza los ficheros que contienen la CRL (tanto en formato PEM como DER) y los ficheros PKCS#7 (tanto en
formato PEM como DER). El PKCS#7 contendrá todos los certificados válidos de la CA, el certificado de la CA y la CRL.
#
# $Id: actualizaCA.sh,v 1.0
# ------------------------------------------------------------------------

#Nos situamos en el directorio adecuado
cd /usr/share/ssl

#Generamos la CRL X.509 de la CA en formato PEM
openssl ca -gencrl -out /usr/share/ssl/demoCA/files/CRLPEM.crl

#Pasamos la CRL X.509 del formato PEM al DER
openssl crl -in /usr/share/ssl/demoCA/files/CRLPEM.crl -out /usr/share/ssl/demoCA/files/CRLDER.crl -outform DER

#Creamos la CRL PKCS#7 en formato PEM
openssl crl2pkcs7 -in /usr/share/ssl/demoCA/files/CRLPEM.crl -out /usr/share/ssl/demoCA/files/CRLPEM.p7b

#Pasamos la CRL PKCS#7 del formato PEM al DER
openssl pkcs7 -in /usr/share/ssl/demoCA/files/CRLPEM.p7b -out /usr/share/ssl/demoCA/files/CRLDER.crl -outform DER
```

```
#Creamos el PKCS#7 en formato PEM con el/los certificado/s del archivo pasado como argumento ($1) y la CRL
openssl crl2pkcs7 -in /usr/share/ssl/demoCA/files/CRLPEM.crl -out /usr/share/ssl/demoCA/files/allPEM.p7b -certfile $1

#Pasamos ese PKCS#7 del formato PEM al DER
openssl pkcs7 -in /usr/share/ssl/demoCA/files/allPEM.p7b -out /usr/share/ssl/demoCA/files/allDER.p7b -outform DER
```

certifica.sh

```
#!/bin/sh
# ------------------------------------------------------------------------------
# certifica.sh - Script que certifica la CSR ($1) en el certificado ($2).
#
# $Id: certifica.sh,v 1.0
# ------------------------------------------------------------------------------

#Nos situamos en el directorio adecuado
cd /usr/share/ssl

#Certificamos la CSR ($1) en el certificado ($2) de forma automática y si texto fuera de -----BEGIN CERTIFICATE----- Y -----END
CERTIFICATE-----
/usr/bin/openssl ca -batch -notext -in $1 -out $2
```

creaDER.sh

```
#!/bin/sh
# ------------------------------------------------------------------------------
# creaDER.sh - Script que copia  el certificado en formato PEM ($1) en ($2) y crea a partir de él, certificados en formato DER ($3) y
PKCS#7 en formato PEM ($4) y DER ($5)
#
# $Id: creaDER.sh,v 1.0
# ------------------------------------------------------------------------------

#Copiamos $1 en $2
```

```
cp  $1 $2

#Nos situamos en el directorio adecuado
cd /usr/share/ssl

#Cambiamos el formato del certificado X.509 de PEM ($2) a DER ($3)
openssl x509 -in $2 -out $3 -outform DER

#Pasamos el certificado X.509 ($2) al formato PKCS#7 sin CRL ($4)
openssl crl2pkcs7 -nocrl -certfile $2 -out $4

#Pasamos el certificado PKCS#7 de formato PEM ($4) a DER ($5)
openssl pkcs7 -in $4 -out $5 -outform DER
```

iniciaCA.sh

```
#!/bin/sh
# -----------------------------------------------------------------------------
# resetCA.sh - Script que borra los archivos de la CA antiguos y crea los necesarios para poder arrancar la CA sin errores.
#
# $Id: resetCA.sh,v 1.0 2002/09/29 13:44:16 $
# -----------------------------------------------------------------------------

#Borramos los ficheros antiguos
rm /usr/share/ssl/demoCA/importante.txt
rm /usr/share/ssl/demoCA/emails.txt
rm /usr/share/ssl/demoCA/serial*
rm /usr/share/ssl/demoCA/index.txt*
rm /usr/share/ssl/demoCA/miIndex.txt
rm /usr/share/ssl/demoCA/newcerts/*
rm /usr/share/ssl/demoCA/files/*

#Creamos /usr/share/ssl/demoCA/index.txt como un archivo vacío
touch /usr/share/ssl/demoCA/index.txt

#Inicializamos /usr/share/ssl/demoCA/serial
printf "01" > /usr/share/ssl/demoCA/serial
```

```
#Nos situamos en el directorio adecuado
cd /usr/share/ssl/

#Generamos una CRL X.509 en formato PEM
openssl ca -gencrl -out /usr/share/ssl/demoCA/files/CRLPEM.crl

#Pasamos la CRL X.509 en formato PEM al formato DER
openssl crl -in /usr/share/ssl/demoCA/files/CRLPEM.crl -out /usr/share/ssl/demoCA/files/CRLDER.crl -outform DER

#Pasamos la CRL X.509 en formato PEM a PKCS#7 en formato PEM
openssl crl2pkcs7 -in /usr/share/ssl/demoCA/files/CRLPEM.crl -out /usr/share/ssl/demoCA/files/CRLPEM.p7b

#Pasamos la CRL PKCS#7 en formato PEM al formato DER
openssl pkcs7 -in /usr/share/ssl/demoCA/files/CRLPEM.p7b -out /usr/share/ssl/demoCA/files/CRLDER.p7b -outform DER

#Copiamos el certificado X.509 en formato PEM de la CA
cp /usr/share/ssl/demoCA/cacert.pem /usr/share/ssl/demoCA/files/CAPEM.cer

#Convertimos el certificado X.509 en formato PEM de la CA al formato DER
openssl x509 -in /usr/share/ssl/demoCA/files/CAPEM.cer -out /usr/share/ssl/demoCA/files/CADER.cer -outform DER

#Creamos el certificado PKCS#7 en formato PEM de la CA a partir del certificado X.509 en formato PEM de la CA
openssl crl2pkcs7 -nocrl -certfile /usr/share/ssl/demoCA/files/CAPEM.cer -out /usr/share/ssl/demoCA/files/CAPEM.p7b

#Convertimos el certificado PKCS#7 de la CA del formato PEM al DER
openssl pkcs7 -in /usr/share/ssl/demoCA/files/CAPEM.p7b -out /usr/share/ssl/demoCA/files/CADER.p7b -outform DER

#Creamos el PKCS#7 en formato PEM a partir del certificado X.509 en formato PEM de la CA y el CRL x.509 en formato PEM
openssl crl2pkcs7 -in /usr/share/ssl/demoCA/files/CRLPEM.crl -certfile /usr/share/ssl/demoCA/files/CAPEM.cer -out
/usr/share/ssl/demoCA/files/allPEM.p7b

#Convertimos ese PKCS#7 del formato PEM al formato DER
openssl pkcs7 -in /usr/share/ssl/demoCA/files/allPEM.p7b -out /usr/share/ssl/demoCA/files/allDER.p7b -outform DER
```

makeCA.sh

```
#!/bin/sh
# ----------------------------------------------------------------------------
# makeCA.sh - Script que compila todos los servlets de la CA
#
# $Id: makeCA.sh,v 1.0
# ----------------------------------------------------------------------------

setDefaultJava --devel SunJava2
source setJava --devel SunJava2
alias keytool=/usr/java/j2sdk1.4.0_02/bin/keytool
alias javac=/usr/java/j2sdk1.4.0_02/bin/javac
alias java=/usr/java/j2sdk1.4.0_02/bin/java
alias keytool=/usr/java/j2sdk1.4.0_02/bin/keytool
alias javadoc=/usr/java/j2sdk1.4.0_02/bin/javadoc
alias jdb=/usr/java/j2sdk1.4.0_02/bin/jdb
/usr/java/j2sdk1.4.0_02/bin/javac -classpath /opt/jakarta/common/lib/servlet.jar:/opt/jakarta/webapps/examples/WEB-INF/classes/ -d
/opt/jakarta/webapps/examples/WEB-INF/classes/ /usr/share/ssl/demoCA/servlets/src/CAException.java
/usr/java/j2sdk1.4.0_02/bin/javac -classpath /opt/jakarta/common/lib/servlet.jar:/opt/jakarta/webapps/examples/WEB-INF/classes/ -d
/opt/jakarta/webapps/examples/WEB-INF/classes/ /usr/share/ssl/demoCA/servlets/src/UtilCA.java
/usr/java/j2sdk1.4.0_02/bin/javac -classpath /opt/jakarta/common/lib/servlet.jar:/opt/jakarta/webapps/examples/WEB-INF/classes/ -d
/opt/jakarta/webapps/examples/WEB-INF/classes/ /usr/share/ssl/demoCA/servlets/src/ObtenerCSR.java
/usr/java/j2sdk1.4.0_02/bin/javac -classpath /opt/jakarta/common/lib/servlet.jar:/opt/jakarta/webapps/examples/WEB-INF/classes/ -d
/opt/jakarta/webapps/examples/WEB-INF/classes/ /usr/share/ssl/demoCA/servlets/src/ObtenerEmail.java
/usr/java/j2sdk1.4.0_02/bin/javac -classpath /opt/jakarta/common/lib/servlet.jar:/opt/jakarta/webapps/examples/WEB-INF/classes/ -d
/opt/jakarta/webapps/examples/WEB-INF/classes/ /usr/share/ssl/demoCA/servlets/src/Descargar.java
/usr/java/j2sdk1.4.0_02/bin/javac -classpath /opt/jakarta/common/lib/servlet.jar:/opt/jakarta/webapps/examples/WEB-INF/classes/ -d
/opt/jakarta/webapps/examples/WEB-INF/classes/ /usr/share/ssl/demoCA/servlets/src/Revocar.java
```

revoca.sh

```
#!/bin/sh
# ----------------------------------------------------------------------------
# revoca.sh - Script que revoca el certificado pasado como argumento.
#
```

```
# $Id: revoca.sh,v 1.0
# --------------------------------------------------------------------------

#Nos situamos en el directorio adecuado
cd /usr/share/ssl

#Revocamos el certificado pasado como parámetro
openssl ca -revoke $1
```

instalarCA.sh

```
# Hacemos copia de los archivos de configuración
mv /usr/share/ssl/openssl.cnf /usr/share/ssl/openssl.cnf.old
mv /opt/jakarta/conf/server.xml /opt/jakarta/conf/server.xml.old
mv /opt/jakarta/webapps/ROOT/index.html /opt/jakarta/webapps/ROOT/index.html.old
mv /usr/sbin/setDefaultJava /usr/sbin/setDefaultJava.old
mv /usr/bin/setJava /usr/bin/setJava.old

# Copiamos ls archivos a los directorios adecuados
cp -R opt/ /opt/
cp -R usr/ /usr/
```

desinstalarCA.sh

```
# Deshacemos la copia de los archivos de configuración
mv /usr/share/ssl/openssl.cnf.old /usr/share/ssl/openssl.cnf
mv /opt/jakarta/conf/server.xml.old /opt/jakarta/conf/server.xml

# Borramos los archivos y directorios de la CA
rm -R /usr/share/ssl/demoCA
rm /usr/share/ssl/actualizaCA.sh
rm /usr/share/ssl/certifica.sh
rm /usr/share/ssl/creaDER.sh
rm /usr/share/ssl/iniciaCA.sh
```

```
rm /usr/share/ssl/makeCA.sh
rm /usr/share/ssl/revoca.sh
rm /usr/share/ssl/deinstalarCA.sh
```

/usr/share/ssl/openssl.cnf

```
#
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
#

# This definition stops the following lines choking if HOME isn't
# defined.
HOME                            = .
RANDFILE                = $ENV::HOME/.rnd

# Extra OBJECT IDENTIFIER info:
#oid_file               = $ENV::HOME/.oid
oid_section             = new_oids

# To use this configuration file with the "-extfile" option of the
# "openssl x509" utility, name here the section containing the
# X.509v3 extensions to use:
# extensions                    =
# (Alternatively, use a configuration file that has only
# X.509v3 extensions in its main [= default] section.)

[ new_oids ]

# We can add new OIDs in here for use by 'ca' and 'req'.
# Add a simple OID like this:
# testoid1=1.2.3.4
# Or use config file substitution like this:
# testoid2=${testoid1}.5.6

####################################################################
[ ca ]
```

```
default_ca  = CA_default                        # The default ca section

######################################################################
[ CA_default ]

dir                     = ./demoCA              # Where everything is kept
certs                   = $dir/certs                        # Where the issued certs are kept
crl_dir                 = $dir/crl              # Where the issued crl are kept
database    = $dir/index.txt        # database index file.
new_certs_dir           = $dir/newcerts                     # default place for new certs.

certificate = $dir/cacert.pem       # The CA certificate
serial                  = $dir/serial                       # The current serial number
crl                     = $dir/crl.pem                      # The current CRL
private_key = $dir/private/cakey.pem# The private key
RANDFILE    = $dir/private/.rand    # private random number file

x509_extensions         = usr_cert              # The extentions to add to the cert

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crl_extensions        = crl_ext

default_days            = 365                               # how long to certify for
default_crl_days= 30                            # how long before next CRL
default_md  = md5                               # which md to use.
preserve    = no                                # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy                  = policy_match

# For the CA policy
[ policy_match ]
countryName             = optional
stateOrProvinceName     = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
```

```
emailAddress                    = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName                    = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress                    = optional

####################################################################
[ req ]
default_bits                    = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions         = v3_ca     # The extentions to add to the self signed cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix       : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
```

```
countryName_default             = AU
countryName_min                         = 2
countryName_max                         = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Some-State

localityName                            = Locality Name (eg, city)

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = Internet Widgits Pty Ltd

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     = World Wide Web Pty Ltd

organizationalUnitName          = Organizational Unit Name (eg, section)
#organizationalUnitName_default =

commonName                      = Common Name (eg, YOUR name)
commonName_max                          = 64

emailAddress                            = Email Address
emailAddress_max                = 40

# SET-ex3                       = SET extension number 3

[ req_attributes ]
challengePassword               = A challenge password
challengePassword_min           = 4
challengePassword_max           = 20

unstructuredName                = An optional company name

[ usr_cert ]

# These extensions are added when 'ca' signs a request.

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.
```

```
basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType                                = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment                         = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy

# Copy subject details
# issuerAltName=issuer:copy

#nsCaRevocationUrl                = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
```

```
#nsSslServerName

[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

[ v3_ca ]


# Extensions for a typical CA


# PKIX recommendation.

subjectKeyIdentifier=hash

authorityKeyIdentifier=keyid:always,issuer:always

# This is what PKIX recommends but some broken software chokes on critical
# extensions.
#basicConstraints = critical,CA:true
# So we do this instead.
basicConstraints = CA:true

# Key usage: this is typical for a CA certificate. However since it will
# prevent it being used as an test self-signed certificate it is best
# left out by default.
# keyUsage = cRLSign, keyCertSign

# Some might want this also
# nsCertType = sslCA, emailCA

# Include email address in subject alt name: another PKIX recommendation
# subjectAltName=email:copy
# Copy issuer details
# issuerAltName=issuer:copy
```

```
# DER hex encoding of an extension: beware experts only!
# obj=DER:02:03
# Where 'obj' is a standard or added object
# You can even override a supported extension:
# basicConstraints= critical, DER:30:03:01:01:FF

[ crl_ext ]

# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always
```

/usr/sbin/setDefaultJava

```
#! /bin/bash
# Copyright (c) 2000-2002 SuSE Linux AG, Nuernberg, Germany.
# All rights reserved.
#
# Author: Petr Mladek <pmladek@suse.cz>

PRINT_HELP=no
DEVEL=no

case "$#" in
  "1")
    if [ "$1" = "--help" ] ; then
      PRINT_HELP=yes
    else
      JAVA_VER=$1
    fi
    ;;
  "2")
    if [ "$1" != "--devel" ] ; then
      PRINT_HELP=yes
    else
```

```
      DEVEL=yes
      JAVA_VER=$2
    fi
    ;;
  *)
    PRINT_HELP=yes
    ;;
esac

if [ "$PRINT_HELP" = "yes" ] ; then
  echo "This script sets default java for your computer"
  echo
  echo "Usage: setDefaultJava --help"
  echo "       setDEfaultJava [--devel] java_ver"
  echo
  echo "Command line options:"
  echo "    --help   - print this help"
  echo "    --devel  - use this options if you request the development kit"
  echo "    java_ver - is one from: SunJava1, SunJava2, IBMJava2"
  echo "                                Java2, default"
  echo "             - SunJava1 - java ver. 1.1.x, from Sun Microsystems"
  echo "                 - SunJava2 - java ver. 1.2.x or 1.3.x, from Sun Microsystems"
  echo "                 - IBMJava2 - java ver. 1.2.x or 1.3.x, from IBM"
  echo "                 - Java2    - java ver. 1.2.x or 1.3.x, any vendor"
  echo "                 - default  - any version, any vendor"
  echo
  exit 0
fi

# check if we are started as root
# only one of UID and USER must be set correctly
if test "$UID" != 0 -a "$USER" != root; then
    echo "You must be root to start $0."
    exit 1
fi

# test if there is desired java
JAVALINKTO=

case "$JAVA_VER" in
```

```
          "SunJava1")
            if [ "$DEVEL" = "yes" ] ; then
              if [ -x /usr/lib/SunJava1/bin/javac ] ; then JAVALINKTO=/usr/lib/SunJava1; fi
        else
              if [ -x /usr/lib/SunJava1/bin/jre ] ; then JAVALINKTO=/usr/lib/SunJava1; fi
            fi
            ;;
          "SunJava2")
            if [ "$DEVEL" = "yes" ] ; then
              if [ -x /usr/java/j2sdk1.4.0_02/bin/javac ] ; then JAVALINKTO=/usr/java/j2sdk1.4.0_02; fi
            else
              if [ -x /usr/java/j2sdk1.4.0_02/jre/bin/java ] ; then JAVALINKTO=/usr/java/j2sdk1.4.0_02; fi
            fi
            ;;
      "IBMJava2")
            if [ "$DEVEL" = "yes" ] ; then
              if [ -x /usr/lib/IBMJava2/bin/javac ] ; then JAVALINKTO=/usr/lib/IBMJava2; fi
            else
              if [ -x /usr/lib/IBMJava2/jre/bin/java ] ; then JAVALINKTO=/usr/lib/IBMJava2; fi
            fi
            ;;
      "Java2")
            if [ "$DEVEL" = "yes" ] ; then
              if   [ -x /usr/lib/SunJava2/bin/javac ] ; then JAVALINKTO=/usr/lib/SunJava2
              elif [ -x /usr/lib/IBMJava2/bin/javac ] ; then JAVALINKTO=/usr/lib/IBMJava2
              fi
            else
              if   [ -x /usr/lib/SunJava2/jre/bin/java ] ; then JAVALINKTO=/usr/lib/SunJava2
              elif [ -x /usr/lib/IBMJava2/jre/bin/java ] ; then JAVALINKTO=/usr/lib/IBMJava2
          fi
            fi
            ;;
      "default")
            if [ "$DEVEL" = "yes" ] ; then
              if   [ -x /usr/lib/SunJava1/bin/javac ] ; then JAVALINKTO=/usr/lib/SunJava1
              elif [ -x /usr/lib/SunJava2/bin/javac ] ; then JAVALINKTO=/usr/lib/SunJava2
              elif [ -x /usr/lib/IBMJava2/bin/javac ] ; then JAVALINKTO=/usr/lib/IBMJava2
          fi
            else
              if   [ -x /usr/lib/SunJava1/bin/jre ] ; then JAVALINKTO=/usr/lib/SunJava1
```

```
               elif [ -x /usr/lib/SunJava2/jre/bin/java ] ; then JAVALINKTO=/usr/lib/SunJava2
               elif [ -x /usr/lib/IBMJava2/jre/bin/java ] ; then JAVALINKTO=/usr/lib/IBMJava2
            fi
              fi
          ;;
       *)
            echo "Error: I do not know \"$JAVA_VER\""
            echo "       Please, use setDefaultJava --help"
            exit 1
esac

if [ -z "$JAVALINKTO" ] ; then
  if [ "$DEVEL" = "yes" ] ; then
    echo "Error: The development version of $JAVA_VER is not installed"
  else
    echo "Error: The requested $JAVA_VER is not installed"
  fi
  exit 1
fi

# create requested link
if [ -L /usr/lib/java ] ; then rm /usr/lib/java; fi
ln -sf `linkto $JAVALINKTO` /usr/lib/java
exit 0
```

/usr/bin/setJava

```
#! /bin/bash
# Copyright (c) 2000-2002 SuSE Linux AG, Nuernberg, Germany.
# All rights reserved.
#
# Author: Petr Mladek <pmladek@suse.cz>

# Decription:
# This script sets java for current shell (for next applications)
#
# It sets environment variables: PATH, JAVA_BINDIR, JAVA_HOME,
```

```
#                               JRE_HOME, JDK_HOME, SDK_HOME
#
# When exist more possibilities it prefers:
#           - SunJava1 before SunJava2
#           - SunJava2 before IBMJava2

EXIT_setJava=no
PRINT_HELP_setJava=no
WANT_HELP_setJava=no
NEW_JAVA_setJava=no
DEVEL_setJava=no

case "$#" in
  "1")
    if [ "$1" = "--help" ] ; then
      PRINT_HELP_setJava=yes
      WANT_HELP_setJava=yes
    else
      JAVA_VER_setJava=$1
    fi
    ;;
  "2")
    if [ "$1" != "--devel" ] ; then
      PRINT_HELP_setJava=yes
    else
      DEVEL_setJava=yes
      JAVA_VER_setJava=$2
    fi
    ;;
  *)
    PRINT_HELP_setJava=yes
    ;;
esac

if [ "$PRINT_HELP_setJava" = "yes" ] ; then
  echo "This script sets java for current shell (for next applications)"
  echo
  echo "It sets environment variables: PATH, JAVA_BINDIR, JAVA_HOME,"
  echo "                               JRE_HOME, JDK_HOME, SDK_HOME"
  echo
```

```
  echo "Usage: setJava --help"
  echo "       source setJava [--devel] java_ver"
  echo
  echo "Command line options:"
  echo "    --help   - print this help"
  echo "    --devel  - use this options if you request the development kit"
  echo "    java_ver - is one from: SunJava1, SunJava2, IBMJava2"
  echo "                                  Java2, default"
  echo "             - SunJava1 - java ver. 1.1.x, from Sun Microsystems"
  echo "                 - SunJava2 - java ver. 1.2.x or 1.3.x, from Sun Microsystems"
  echo "                 - IBMJava2 - java ver. 1.2.x or 1.3.x, from IBM"
  echo "                 - Java2    - java ver. 1.2.x or 1.3.x, any vendor"
  echo "                 - default  - any version, any vendor"
  echo
  echo "Note: Do not forget to run this script with \"source\" shell command !!!"
  echo
  EXIT_setJava=yes
fi

if [ "$EXIT_setJava" = "no" ] ; then
  # find current java in $PATH
  JAVA_OLD_PATH_setJava=
  JAVA_FIRST_IN_PATH_setJava=no
  OLD_IFS=$IFS
  IFS=":,$IFS"
  for i in $PATH ; do
    # we find first occurence
    if [ "$JAVA_FIRST_IN_PATH_setJava" = "no" ] ; then
      case "$i" in
        "/usr/lib/java/bin" | \
        "/usr/lib/java/jre/bin" | \
        "/usr/lib/SunJava1/bin" | \
        "/usr/lib/SunJava2/bin" | \
        "/usr/lib/SunJava2/jre/bin" | \
        "/usr/lib/IBMJava2/bin" | \
        "/usr/lib/IBMJava2/jre/bin")
          JAVA_OLD_PATH_setJava=$i
             JAVA_FIRST_IN_PATH_setJava=yes
          ;;
      esac
```

```
    fi
  done
  IFS=$OLD_IFS
  export IFS
fi

if [ "$EXIT_setJava" = "no" ] ; then
  # select existing java for given request
  SELECTED_JAVA_setJava=
  case "$JAVA_VER_setJava" in
    "SunJava1")
      if [ "$DEVEL_setJava" = "yes" ] ; then
        if [ -x /usr/lib/SunJava1/bin/javac ] ; then SELECTED_JAVA_setJava="SunJava1-JDK" ; fi
      else
        if   [ -x /usr/lib/SunJava1/bin/javac ] ; then SELECTED_JAVA_setJava="SunJava1-JDK"
        elif [ -x /usr/lib/SunJava1/bin/jre ] ; then SELECTED_JAVA_setJava="SunJava1-JRE"
        fi
      fi
      ;;
    "SunJava2")
      if [ "$DEVEL_setJava" = "yes" ] ; then
        if [ -x /usr/java/j2sdk1.4.0_02/bin/javac ] ; then SELECTED_JAVA_setJava="SunJava2-SDK" ; fi
      else
        if   [ -x /usr/java/j2sdk1.4.0_02/bin/javac ] ; then SELECTED_JAVA_setJava="SunJava2-SDK"
        elif [ -x /usr/java/j2sdk1.4.0_02/jre/bin/java ] ; then SELECTED_JAVA_setJava="SunJava2-JRE"
        fi
      fi
      ;;
    "IBMJava2")
      if [ "$DEVEL_setJava" = "yes" ] ; then
        if [ -x /usr/lib/IBMJava2/bin/javac ] ; then SELECTED_JAVA_setJava="IBMJava2-SDK" ; fi
      else
        if   [ -x /usr/lib/IBMJava2/bin/javac ] ; then SELECTED_JAVA_setJava="IBMJava2-SDK"
        elif [ -x /usr/lib/IBMJava2/jre/bin/java ] ; then SELECTED_JAVA_setJava="IBMJava2-JRE"
        fi
      fi
      ;;
    "Java2")
      if [ "$DEVEL_setJava" = "yes" ] ; then
        if   [ -x /usr/lib/SunJava2/bin/javac ] ; then SELECTED_JAVA_setJava="SunJava2-SDK"
```

```
       elif [ -x /usr/lib/IBMJava2/bin/javac ] ; then SELECTED_JAVA_setJava="IBMJava2-SDK"
       fi
     else
       if   [ -x /usr/lib/SunJava2/bin/javac ] ; then SELECTED_JAVA_setJava="SunJava2-SDK"
       elif [ -x /usr/lib/SunJava2/jre/bin/java ] ; then SELECTED_JAVA_setJava="SunJava2-JRE"
       elif [ -x /usr/lib/IBMJava2/bin/javac ] ; then SELECTED_JAVA_setJava="IBMJava2-SDK"
       elif [ -x /usr/lib/IBMJava2/jre/bin/java ] ; then SELECTED_JAVA_setJava="IBMJava2-JRE"
       fi
     fi
     ;;
   "default")
     if [ "$DEVEL_setJava" = "yes" ] ; then
       if   [ -x /usr/lib/SunJava1/bin/javac ] ; then SELECTED_JAVA_setJava="SunJava1-JDK"
       elif [ -x /usr/lib/SunJava2/bin/javac ] ; then SELECTED_JAVA_setJava="SunJava2-SDK"
       elif [ -x /usr/lib/IBMJava2/bin/javac ] ; then SELECTED_JAVA_setJava="IBMJava2-SDK"
       fi
     else
       if   [ -x /usr/lib/SunJava1/bin/javac ] ; then SELECTED_JAVA_setJava="SunJava1-JDK"
       elif [ -x /usr/lib/SunJava1/bin/jre ] ; then SELECTED_JAVA_setJava="SunJava1-JRE"
       elif [ -x /usr/lib/SunJava2/bin/javac ] ; then SELECTED_JAVA_setJava="SunJava2-SDK"
       elif [ -x /usr/lib/SunJava2/jre/bin/java ] ; then SELECTED_JAVA_setJava="SunJava2-JRE"
       elif [ -x /usr/lib/IBMJava2/bin/javac ] ; then SELECTED_JAVA_setJava="IBMJava2-SDK"
       elif [ -x /usr/lib/IBMJava2/jre/bin/java ] ; then SELECTED_JAVA_setJava="IBMJava2-JRE"
       fi
     fi
     ;;
   *)
     echo "Error: I do not know $JAVA_VER_setJava"
     echo "       Please, use setJava --help"
     EXIT_setJava=yes
     ;;
  esac
fi

if [ "$EXIT_setJava" = "no" ] && [ -z "$SELECTED_JAVA_setJava" ] ; then
  if [ "$DEVEL_setJava" = "yes" ] ; then
    echo "Error: The development version of $JAVA_VER_setJava is not installed"
  else
    echo "Error: The requested $JAVA_VER_setJava is not installed"
  fi
```

```
  EXIT_setJava=yes
fi

if [ "$EXIT_setJava" = "no" ] ; then
  # set new values for environment variables
  JAVA_NEW_PATH_setJava=
  case "$SELECTED_JAVA_setJava" in
    "SunJava1-JDK")
      JAVA_NEW_PATH_setJava=/usr/lib/SunJava1/bin
      JAVA_BINDIR=/usr/lib/SunJava1/bin
      JAVA_HOME=/usr/lib/SunJava1
      JRE_HOME=/usr/lib/SunJava1
      JDK_HOME=/usr/lib/SunJava1
      unset SDK_HOME
      ;;
    "SunJava1-JRE")
      JAVA_NEW_PATH_setJava=/usr/lib/SunJava1/bin
      JAVA_BINDIR=/usr/lib/SunJava1/bin
      JAVA_HOME=/usr/lib/SunJava1
      JRE_HOME=/usr/lib/SunJava1
      unset JDK_HOME
      unset SDK_HOME
      ;;
    "SunJava2-SDK")
      JAVA_NEW_PATH_setJava=/usr/java/j2sdk1.4.0_02/bin
      JAVA_BINDIR=/usr/java/j2sdk1.4.0_02/bin
      JAVA_HOME=/usr/java/j2sdk1.4.0_02
      JRE_HOME=/usr/java/j2sdk1.4.0_02/jre
      JDK_HOME=/usr/java/j2sdk1.4.0_02
      SDK_HOME=/usr/java/j2sdk1.4.0_02
      ;;
    "SunJava2-JRE")
      JAVA_NEW_PATH_setJava=/usr/java/j2sdk1.4.0_02/jre/bin
      JAVA_BINDIR=/usr/java/j2sdk1.4.0_02/jre/bin
      JAVA_HOME=/usr/java/j2sdk1.4.0_02/jre
      JRE_HOME=/usr/java/j2sdk1.4.0_02/jre
      unset JDK_HOME
      unset SDK_HOME
      ;;
    "IBMJava2-SDK")
```

```
      JAVA_NEW_PATH_setJava=/usr/lib/IBMJava2/bin
      JAVA_BINDIR=/usr/lib/IBMJava2/bin
      JAVA_HOME=/usr/lib/IBMJava2
      JRE_HOME=/usr/lib/IBMJava2/jre
      JDK_HOME=/usr/lib/IBMJava2
      SDK_HOME=/usr/lib/IBMJava2
      ;;
    "IBMJava2-JRE")
      JAVA_NEW_PATH_setJava=/usr/lib/IBMJava2/jre/bin
      JAVA_BINDIR=/usr/lib/IBMJava2/jre/bin
      JAVA_HOME=/usr/lib/IBMJava2/jre
      JRE_HOME=/usr/lib/IBMJava2/jre
      unset JDK_HOME
      unset SDK_HOME
      ;;
  esac
fi

if [ "$EXIT_setJava" = "no" ] ; then
  # set new java in $PATH
  if [ -n "$JAVA_OLD_PATH_setJava" ] ; then
    PATH=`echo $PATH | sed -e "s|$JAVA_OLD_PATH_setJava|$JAVA_NEW_PATH_setJava|"`
  else
    PATH="$PATH:$JAVA_NEW_PATH_setJava"
  fi
  NEW_JAVA_setJava=yes
fi

unset JAVA_FIRST_IN_PATH_setJava
unset JAVA_OLD_PATH_setJava
unset JAVA_NEW_PATH_setJava
unset DEVEL_setJava
unset PRINT_HELP_setJava
unset SELECTED_JAVA_setJava
unset EXIT_setJava

export PATH JAVA_BINDIR JAVA_HOME JRE_HOME JDK_HOME SDK_HOME


test "$NEW_JAVA_setJava" = "yes" -o "$WANT_HELP_setJava" = "yes"
```

/opt/jakarta/conf/server.xml

```
<!-- Example Server Configuration File -->
<!-- Note that component elements are nested corresponding to their
     parent-child relationships with each other -->

<!-- A "Server" is a singleton element that represents the entire JVM,
     which may contain one or more "Service" instances.  The Server
     listens for a shutdown command on the indicated port.

     Note:  A "Server" is not itself a "Container", so you may not
     define subcomponents such as "Valves" or "Loggers" at this level.
 -->

<Server port="8005" shutdown="SHUTDOWN" debug="0">


  <!-- A "Service" is a collection of one or more "Connectors" that share
       a single "Container" (and therefore the web applications visible
       within that Container).  Normally, that Container is an "Engine",
       but this is not required.

       Note:  A "Service" is not itself a "Container", so you may not
       define subcomponents such as "Valves" or "Loggers" at this level.
   -->

  <!-- Define the Tomcat Stand-Alone Service -->
  <Service name="Tomcat-Standalone">

    <!-- A "Connector" represents an endpoint by which requests are received
         and responses are returned.  Each Connector passes requests on to the
         associated "Container" (normally an Engine) for processing.

         By default, a non-SSL HTTP/1.1 Connector is established on port 8080.
         You can also enable an SSL HTTP/1.1 Connector on port 8443 by
         following the instructions below and uncommenting the second Connector
         entry.  SSL support requires the following steps (see the SSL Config
         HOWTO in the Tomcat 4.0 documentation bundle for more detailed
```

```
      instructions):
      * Download and install JSSE 1.0.2 or later, and put the JAR files
        into "$JAVA_HOME/jre/lib/ext".
      * Execute:
          %JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA (Windows)
          $JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA  (Unix)
        with a password value of "changeit" for both the certificate and
        the keystore itself.

      By default, DNS lookups are enabled when a web application calls
      request.getRemoteHost().  This can have an adverse impact on
      performance, so you can disable it by setting the
      "enableLookups" attribute to "false".  When DNS lookups are disabled,
      request.getRemoteHost() will return the String version of the
      IP address of the remote client.
-->

<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
<Connector className="org.apache.catalina.connector.http.HttpConnector"
           port="80" minProcessors="5" maxProcessors="75"
           enableLookups="true" redirectPort="8443"
           acceptCount="10" debug="0" connectionTimeout="60000"/>
<!-- Note : To disable connection timeouts, set connectionTimeout value
 to -1 -->

<!-- Define an SSL HTTP/1.1 Connector on port 8443 -->

<Connector className="org.apache.catalina.connector.http.HttpConnector"
           port="443" minProcessors="5" maxProcessors="75"
           enableLookups="true"
              acceptCount="10" debug="0" scheme="https" secure="true">
  <Factory className="org.apache.catalina.net.SSLServerSocketFactory"
           clientAuth="false" keystoreFile="/opt/jakarta/conf/.keystore" keystorePass="miclave" protocol="TLS" />
</Connector>


<!-- Define an AJP 1.3 Connector on port 8009 -->
<!--
<Connector className="org.apache.ajp.tomcat4.Ajp13Connector"
           port="8009" minProcessors="5" maxProcessors="75"
```

```
            acceptCount="10" debug="0"/>
    -->

    <!-- Define a Proxied HTTP/1.1 Connector on port 8081 -->
    <!-- See proxy documentation for more information about using this. -->
    <!--
    <Connector className="org.apache.catalina.connector.http.HttpConnector"
               port="8081" minProcessors="5" maxProcessors="75"
               enableLookups="true"
               acceptCount="10" debug="0" connectionTimeout="60000"
               proxyPort="80"/>
    -->

    <!-- Define a non-SSL HTTP/1.0 Test Connector on port 8082 -->
    <!--
    <Connector className="org.apache.catalina.connector.http10.HttpConnector"
               port="8082" minProcessors="5" maxProcessors="75"
               enableLookups="true" redirectPort="8443"
               acceptCount="10" debug="0"/>
    -->

    <!-- An Engine represents the entry point (within Catalina) that processes
         every request.  The Engine implementation for Tomcat stand alone
         analyzes the HTTP headers included with the request, and passes them
         on to the appropriate Host (virtual host). -->

    <!-- Define the top level container in our container hierarchy -->
    <Engine name="Standalone" defaultHost="localhost" debug="0">

      <!-- The request dumper valve dumps useful debugging information about
           the request headers and cookies that were received, and the response
           headers and cookies that were sent, for all requests received by
           this instance of Tomcat.  If you care only about requests to a
           particular virtual host, or a particular application, nest this
           element inside the corresponding <Host> or <Context> entry instead.

           For a similar mechanism that is portable to all Servlet 2.3
           containers, check out the "RequestDumperFilter" Filter in the
           example application (the source for this filter may be found in
           "$CATALINA_HOME/webapps/examples/WEB-INF/classes/filters").
```

```
     Request dumping is disabled by default.  Uncomment the following
     element to enable it. -->
<!--
<Valve className="org.apache.catalina.valves.RequestDumperValve"/>
-->

<!-- Global logger unless overridden at lower levels -->
<Logger className="org.apache.catalina.logger.FileLogger"
        prefix="catalina_log." suffix=".txt"
        timestamp="true"/>

<!-- Because this Realm is here, an instance will be shared globally -->

<Realm className="org.apache.catalina.realm.MemoryRealm" />

<!-- Replace the above Realm with one of the following to get a Realm
     stored in a database and accessed via JDBC -->

<!--
<Realm  className="org.apache.catalina.realm.JDBCRealm" debug="99"
       driverName="org.gjt.mm.mysql.Driver"
    connectionURL="jdbc:mysql://localhost/authority?user=test;password=test"
        userTable="users" userNameCol="user_name" userCredCol="user_pass"
    userRoleTable="user_roles" roleNameCol="role_name" />
-->

<!--
<Realm  className="org.apache.catalina.realm.JDBCRealm" debug="99"
       driverName="oracle.jdbc.driver.OracleDriver"
    connectionURL="jdbc:oracle:thin:@ntserver:1521:ORCL?user=scott;password=tiger"
        userTable="users" userNameCol="user_name" userCredCol="user_pass"
    userRoleTable="user_roles" roleNameCol="role_name" />
-->

<!--
<Realm  className="org.apache.catalina.realm.JDBCRealm" debug="99"
       driverName="sun.jdbc.odbc.JdbcOdbcDriver"
    connectionURL="jdbc:odbc:CATALINA"
        userTable="users" userNameCol="user_name" userCredCol="user_pass"
```

```
        userRoleTable="user_roles" roleNameCol="role_name" />
    -->

    <!-- Define the default virtual host -->
    <Host name="localhost" debug="0" appBase="webapps" unpackWARs="true">

      <!-- Normally, users must authenticate themselves to each web app
           individually.  Uncomment the following entry if you would like
           a user to be authenticated the first time they encounter a
           resource protected by a security constraint, and then have that
           user identity maintained across *all* web applications contained
           in this virtual host. -->
      <!--
      <Valve className="org.apache.catalina.authenticator.SingleSignOn"
                debug="0"/>
      -->

      <!-- Access log processes all requests for this virtual host.  By
           default, log files are created in the "logs" directory relative to
           $CATALINA_HOME.  If you wish, you can specify a different
           directory with the "directory" attribute.  Specify either a relative
           (to $CATALINA_HOME) or absolute path to the desired directory.
      -->
      <Valve className="org.apache.catalina.valves.AccessLogValve"
               directory="logs"  prefix="localhost_access_log." suffix=".txt"
               pattern="common"/>

      <!-- Logger shared by all Contexts related to this virtual host.  By
           default (when using FileLogger), log files are created in the "logs"
           directory relative to $CATALINA_HOME.  If you wish, you can specify
           a different directory with the "directory" attribute.  Specify either a
           relative (to $CATALINA_HOME) or absolute path to the desired
           directory.-->
      <Logger className="org.apache.catalina.logger.FileLogger"
               directory="logs"  prefix="localhost_log." suffix=".txt"
                  timestamp="true"/>

      <!-- Define properties for each web application.  This is only needed
           if you want to set non-default properties, or have web application
           document roots in places other than the virtual host's appBase
```

```
      directory.  -->

<!-- Tomcat Root Context -->
<!--
  <Context path="" docBase="ROOT" debug="0"/>
-->

<!-- Tomcat Manager Context -->
<Context path="/manager" docBase="manager"
 debug="0" privileged="true"/>

<!-- Tomcat Examples Context -->
<Context path="/examples" docBase="examples" debug="0"
         reloadable="true">
  <Logger className="org.apache.catalina.logger.FileLogger"
             prefix="localhost_examples_log." suffix=".txt"
      timestamp="true"/>
  <Ejb   name="ejb/EmplRecord" type="Entity"
         home="com.wombat.empl.EmployeeRecordHome"
       remote="com.wombat.empl.EmployeeRecord"/>
  <!-- PersistentManager: Uncomment the section below to test Persistent
                       Sessions.

       saveOnRestart: If true, all active sessions will be saved
         to the Store when Catalina is shutdown, regardless of
         other settings. All Sessions found in the Store will be
         loaded on startup. Sessions past their expiration are
         ignored in both cases.
       maxActiveSessions: If 0 or greater, having too many active
         sessions will result in some being swapped out. minIdleSwap
         limits this. -1 means unlimited sessions are allowed.
         0 means sessions will almost always be swapped out after
         use - this will be noticeably slow for your users.
       minIdleSwap: Sessions must be idle for at least this long
         (in seconds) before they will be swapped out due to
       maxActiveSessions. This avoids thrashing when the site is
         highly active. -1 or 0 means there is no minimum - sessions
         can be swapped out at any time.
       maxIdleSwap: Sessions will be swapped out if idle for this
         long (in seconds). If minIdleSwap is higher, then it will
```

```
        override this. This isn't exact: it is checked periodically.
        -1 means sessions won't be swapped out for this reason,
        although they may be swapped out for maxActiveSessions.
        If set to >= 0, guarantees that all sessions found in the
        Store will be loaded on startup.
      maxIdleBackup: Sessions will be backed up (saved to the Store,
        but left in active memory) if idle for this long (in seconds),
        and all sessions found in the Store will be loaded on startup.
        If set to -1 sessions will not be backed up, 0 means they
        should be backed up shortly after being used.

      To clear sessions from the Store, set maxActiveSessions, maxIdleSwap,
      and minIdleBackup all to -1, saveOnRestart to false, then restart
      Catalina.
-->
                <!--
<Manager className="org.apache.catalina.session.PersistentManager"
    debug="0"
    saveOnRestart="true"
    maxActiveSessions="-1"
    minIdleSwap="-1"
    maxIdleSwap="-1"
    maxIdleBackup="-1">
      <Store className="org.apache.catalina.session.FileStore"/>
</Manager>
                -->
<Environment name="maxExemptions" type="java.lang.Integer"
             value="15"/>
<Parameter name="context.param.name" value="context.param.value"
           override="false"/>
<Resource name="jdbc/EmployeeAppDb" auth="SERVLET"
          type="javax.sql.DataSource"/>
<ResourceParams name="jdbc/EmployeeAppDb">
  <parameter><name>user</name><value>sa</value></parameter>
  <parameter><name>password</name><value></value></parameter>
  <parameter><name>driverClassName</name>
    <value>org.hsql.jdbcDriver</value></parameter>
  <parameter><name>driverName</name>
    <value>jdbc:HypersonicSQL:database</value></parameter>
</ResourceParams>
```

```
        <Resource name="mail/Session" auth="Container"
                  type="javax.mail.Session"/>
        <ResourceParams name="mail/Session">
          <parameter>
            <name>mail.smtp.host</name>
            <value>localhost</value>
          </parameter>
        </ResourceParams>
      </Context>

    </Host>

  </Engine>

</Service>

<!-- The MOD_WEBAPP connector is used to connect Apache 1.3 with Tomcat 4.0
     as its servlet container. Please read the README.txt file coming with
     the WebApp Module distribution on how to build it.
     (Or check out the "jakarta-tomcat-connectors/webapp" CVS repository)

     To configure the Apache side, you must ensure that you have the
     "ServerName" and "Port" directives defined in "httpd.conf".  Then,
     lines like these to the bottom of your "httpd.conf" file:

       LoadModule webapp_module libexec/mod_webapp.so
       WebAppConnection warpConnection warp localhost:8008
       WebAppDeploy examples warpConnection /examples/

     The next time you restart Apache (after restarting Tomcat, if needed)
     the connection will be established, and all applications you make
     visible via "WebAppDeploy" directives can be accessed through Apache.
-->

<!-- Define an Apache-Connector Service -->
<Service name="Tomcat-Apache">

  <Connector className="org.apache.catalina.connector.warp.WarpConnector"
   port="8008" minProcessors="5" maxProcessors="75"
   enableLookups="true"
```

```
    acceptCount="10" debug="0"/>

   <!-- Replace "localhost" with what your Apache "ServerName" is set to -->
   <Engine className="org.apache.catalina.connector.warp.WarpEngine"
    name="Apache" debug="0" appBase="webapps">

     <!-- Global logger unless overridden at lower levels -->
     <Logger className="org.apache.catalina.logger.FileLogger"
             prefix="apache_log." suffix=".txt"
             timestamp="true"/>

     <!-- Because this Realm is here, an instance will be shared globally -->
     <Realm className="org.apache.catalina.realm.MemoryRealm" />

   </Engine>

  </Service>

</Server>
```