

1. CONCEPTOS FUNDAMENTALES

- **Keystore:** Es el **Almacén de Claves**. Un fichero físico (ej. DAM) que funciona como un maletín donde guardas tus llaves digitales y certificados. Es tu identidad como programador.
- **Storepass:** Es la **contraseña del maletín**. Necesaria para abrir y ver el contenido del Keystore.
- **Keypass:** Es la **contraseña de la llave**. Clave específica para usar una llave concreta (ej. eli) dentro del maletín.
- **Certificado (.cer):** Es la "fotocopia" de tu llave pública que permites que otros importen para confiar en tu firma.

2. ESTRUCTURA: PAQUETE POR DEFECTO

(Ejecutar dentro de la carpeta src)

1. **Compilar:** `javac *.java`
2. **Empaquetar:** `jar cvf app.jar *.class`
3. **Generar Llaves:** `keytool -genkeypair -alias eli -keystore DAM -storepass distancia -keypass 2011alex -keyalg RSA//por defecto dsa`
4. **Firmar:** `jarsigner -keystore DAM -storepass distancia -keypass 2011alex app.jar eli` *(Nota: jarsigner se escribe todo junto)*
5. **Verificar:** `jarsigner -verify app.jar`

3. ESTRUCTURA: CON PAQUETES

(Ejemplo para package com.datos y package com.interfaz)

1. **Compilar:** `javac com/datos/*.java com/interfaz/*.java`
2. **Empaquetar:** `jar cvf app.jar com/datos/*.class com/interfaz/*.class`
3. **Firmar y Verificar:** *(Los comandos de firma y verificación son iguales al paquete por defecto, ya que actúan sobre el .jar final).*

4. GESTIÓN DE CERTIFICADOS (IMPORT/EXPORT)

(Comandos clave de tu última consola, recuerda que van sin espacios)

- **Exportar Certificado:** Extrae tu clave pública a un archivo. `keytool -exportcert -alias eli -keystore DAM -file eli.cer -rfc -storepass distancia`

- **Importar Certificado:** Agrega un certificado a un nuevo almacén para confiar en él.
keytool -importcert -alias eli -file eli.cer -keystore DAM2 -storepass distancia

Si en el examen te piden restringir el acceso a **c:/datos**, el proceso completo es este:

Paso A: Crear el archivo seguridad.policy

Java

```
// Contenido del archivo seguridad.policy donde esta app.jar
grant {
    // Permiso de lectura para la carpeta específica
    permission java.io.FilePermission "C:/datos/-", "read";

    // Permisos mínimos para que Java funcione (Logger y archivos temporales)
    permission java.io.FilePermission
"C:/Users/User/Documents/NetBeansProjects/6/src/-", "read,write";
};
```

Paso B: Ejecutar el comando activando ese archivo

Bash

```
java -Djava.security.manager -Djava.security.policy=seguridad.policy
-jar app.jar
```

Resumen Directo para el Examen

- **El código grant:** Va en un archivo de texto externo con extensión **.policy**.
- **El parámetro -Djava.security.manager:** Es el interruptor que **enciende** la seguridad.
- **El parámetro -Djava.security.policy=nombre.policy:** Es el que le dice a Java **qué reglas** leer.

¿Te gustaría que te pre