



# Programación de servicios y procesos

Act5.3. Confidencialidad e  
identidad

Francisco José García Cutillas | 2FPGS\_DAM



## Índice

Ejercicio 1 .....	3
-------------------	---

## Ejercicio 1

### Confidencialidad e identidad

Programa una aplicación Java que, dado un fichero, lo cifre primeramente con la clave privada del emisor y, posteriormente, con la clave pública del destinatario.

Programa también la aplicación que descifre el fichero.

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   */
4   package com.mycompany.psp_act5_3;
5
6   import java.io.FileInputStream;
7   import java.io.FileOutputStream;
8   import java.security.KeyPair;
9   import java.security.KeyPairGenerator;
10  import java.security.NoSuchAlgorithmException;
11  import java.security.PrivateKey;
12  import java.security.PublicKey;
13  import javax.crypto.Cipher;
14  import javax.crypto.CipherInputStream;
15  import javax.crypto.CipherOutputStream;
16
17  /**
18   *
19   * @author Fran
20   */
21  public class PSP_Act5_3 {
22
23      public static void main(String[] args) {
24
25          try {
26
27              //Generamos las claves privadas y públicas del emisor y el receptor
28              KeyPair clavesEmisor = generarClaves();
29              KeyPair clavesReceptor = generarClaves();
30
31              //Ciframos el fichero con la clave privada del emisor
32              cifrarPrivadaEmisor(rutaFichero: "ficheroCifrar.txt", clavePrivadaEmisor: clavesEmisor.getPrivate());
33
34              //Ciframos el fichero con la clave pública del receptor
35              cifrarPublicaReceptor(rutaFichero: "ficheroCifrar.txt", clavePublicaReceptor: clavesReceptor.getPublic());
36
37              //Desencriptamos el fichero con la clave privada del receptor
38              descifrar(ficheroCifrado: "ficheroCifradoPublicaReceptor.rsa", nombreFicheroSinCifrar: "ficheroReceptorDescifrado.txt",
39                      key: clavesReceptor.getPrivate());
40
41          } catch (Exception ex) {
42
43              System.out.println("Error: " + ex.getMessage());
44
45          }
46
47      }
48
49      //Método para generar el par de claves pública y privada
50      public static KeyPair generarClaves() throws NoSuchAlgorithmException {
51
52          KeyPairGenerator generador = KeyPairGenerator.getInstance("RSA");
53          generador.initialize(2048);
54          KeyPair claves = generador.generateKeyPair();
55
56          return claves;
57
58      }
59  }

```

```

60 //Método para cifrar un fichero con la clave privada del emisor. Recibe la ruta del fichero y la clave privada del emisor
61 public static void cifrarPrivadaEmisor(String rutaFichero, PrivateKey clavePrivadaEmisor) throws Exception {
62
63     //Cifrado con clave privada del emisor.
64     Cipher cipher = Cipher.getInstance("RSA");
65     cipher.init(ops: Cipher.ENCRYPT_MODE, key: clavePrivadaEmisor);
66
67     FileInputStream fis = new FileInputStream(rutaFichero);
68     FileOutputStream fos = new FileOutputStream("ficheroCifradoPrivadaEmisor.rsa");
69     CipherOutputStream cos = new CipherOutputStream(fos, cipher);
70
71     //Leemos el fichero y lo ciframos
72     byte[] buffer = new byte[1024];
73
74     int bytesLeídos = fis.read(buffer);
75
76     while (bytesLeídos != -1) {
77
78         cos.write(buffer, 0, bytesLeídos);
79         bytesLeídos = fis.read(buffer);
80
81     }
82
83     cos.close();
84     fos.close();
85     fis.close();
86
87     System.out.println("Fichero cifrado con clave privada del emisor.");
88
89 }
90
91 //Método para cifrar un fichero con la clave pública del receptor. Recibe la ruta del fichero y la clave pública del receptor
92 public static void cifrarPublicaReceptor(String rutaFichero, PublicKey clavePublicaReceptor) throws Exception {
93
94     //Cifrado con clave pública del receptor.
95     Cipher cipher = Cipher.getInstance("RSA");
96     cipher.init(ops: Cipher.ENCRYPT_MODE, key: clavePublicaReceptor);
97
98     FileInputStream fis = new FileInputStream(rutaFichero);
99     FileOutputStream fos = new FileOutputStream("ficheroCifradoPublicaReceptor.rsa");
100     CipherOutputStream cos = new CipherOutputStream(fos, cipher);
101
102     //Leemos el fichero y lo ciframos
103     byte[] buffer = new byte[1024];
104
105     int bytesLeídos = fis.read(buffer);
106
107     while (bytesLeídos != -1) {
108
109         cos.write(buffer, 0, bytesLeídos);
110         bytesLeídos = fis.read(buffer);
111
112     }
113
114     cos.close();
115     fos.close();
116     fis.close();
117
118     System.out.println("Fichero cifrado con clave pública del receptor.");
119
120 }

```

```

121 //Método para descifrar un fichero. Recibe la ruta del fichero cifrado, la del nuevo sin cifrar y la clave privada del receptor
122 private static void descifrar(String ficheroCifrado, String nombreFicheroSinCifrar, PrivateKey key) throws Exception {
123
124     //Descifrado con clave privada del receptor
125     Cipher cipher = Cipher.getInstance("RSA");
126     cipher.init(ops: Cipher.DECRYPT_MODE, key);
127
128     FileInputStream fis = new FileInputStream(ficheroCifrado);
129     FileOutputStream fos = new FileOutputStream(nombreFicheroSinCifrar);
130     CipherInputStream cis = new CipherInputStream(fis, cipher);
131
132     //Leemos el fichero y lo desciframos
133     byte[] buffer = new byte[1024];
134
135     int bytesLeídos = cis.read(buffer);
136
137     while (bytesLeídos != -1) {
138
139         fos.write(buffer, 0, bytesLeídos);
140         bytesLeídos = cis.read(buffer);
141
142     }
143
144     cis.close();
145     fos.close();
146     fis.close();
147
148     System.out.println("Fichero descifrado.");
149
150 }
151
152 }
153
154

```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ PSP_Act5_3 ---
Fichero cifrado con clave privada del emisor.
Fichero cifrado con clave pública del receptor.
Fichero descifrado.
-----
BUILD SUCCESS
-----
Total time: 0.595 s
Finished at: 2024-02-09T21:03:38+01:00
-----
```

Fichero original

ficheroCifrar.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Este es un mensaje cifrado para su envío a un destinatario.

Fichero cifrado con clave privada del emisor

ficheroCifradoPrivadaEmisor.rsa: Bloc de notas

Archivo Edición Formato Ver Ayuda

...

Fichero cifrado con clave pública del receptor

ficheroCifradoPublicaReceptor.rsa: Bloc de notas

Archivo Edición Formato Ver Ayuda

...

Fichero descifrado con clave privada del receptor

ficheroReceptorDescifrado.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Este es un mensaje cifrado para su envío a un destinatario.