

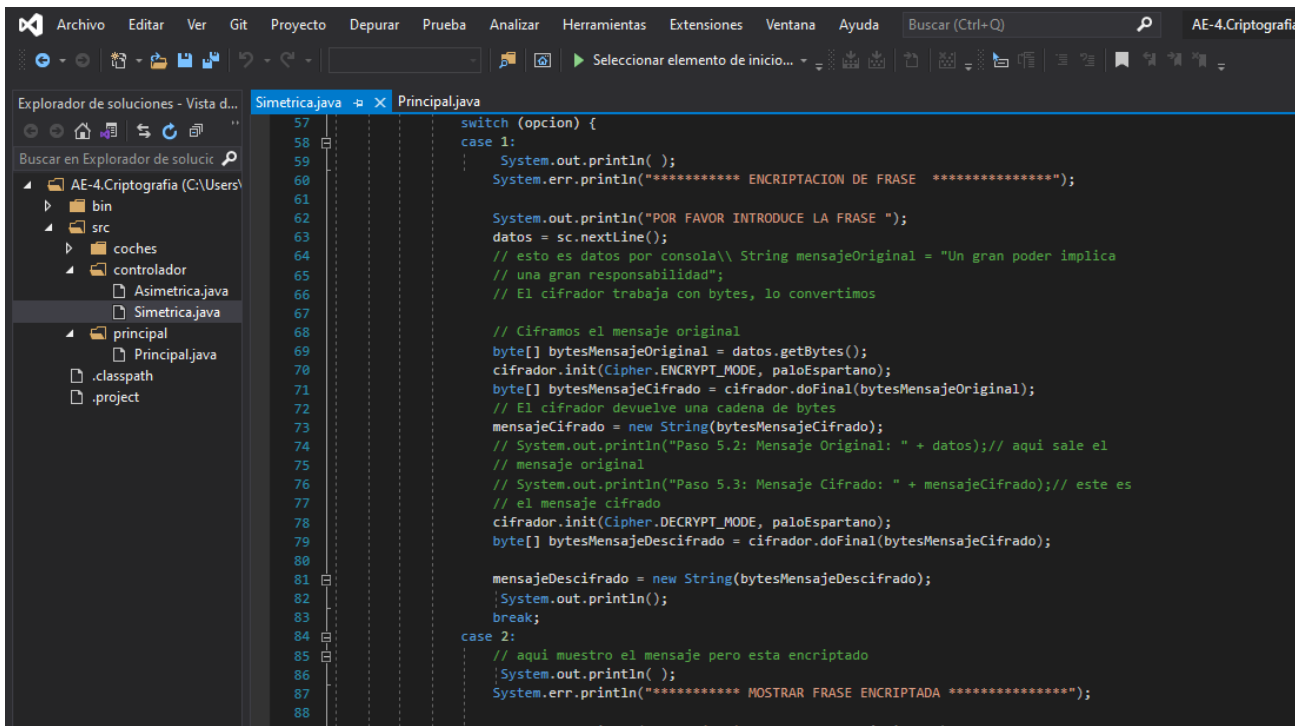


# AE-4. Criptografía

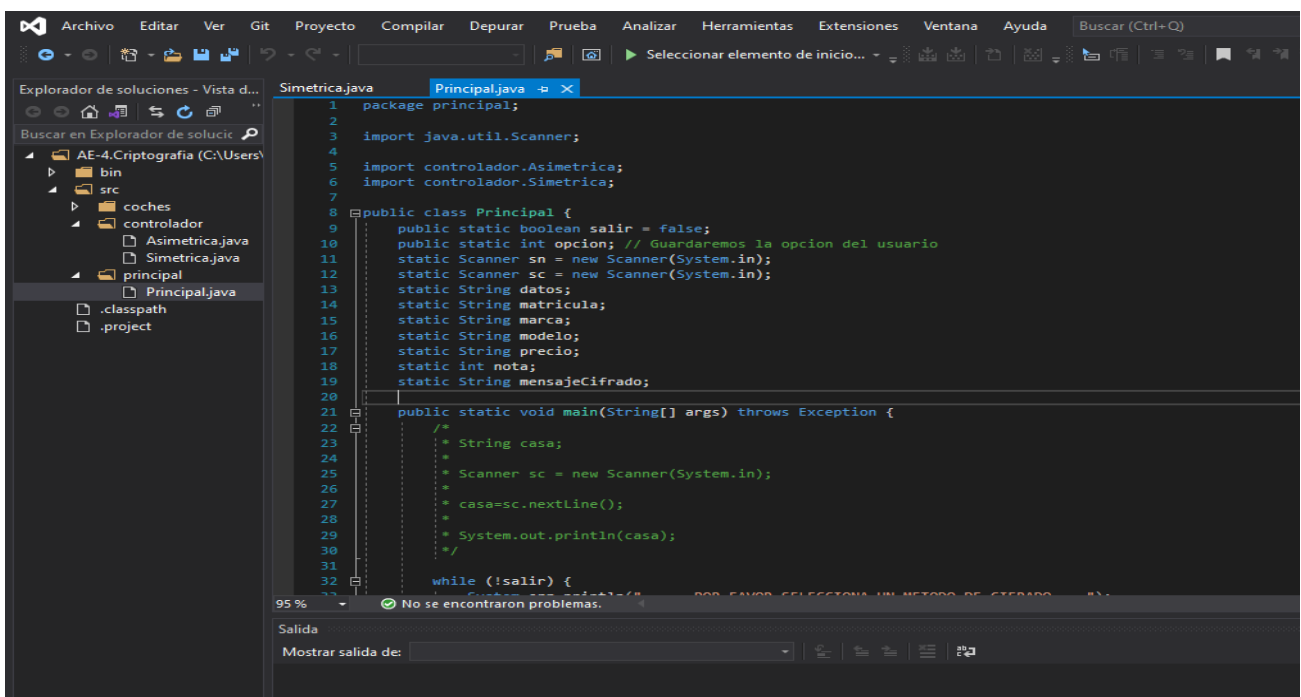


David Lara  
Adrian Caballero  
Miguel Borrás

Para realizar la encriptacion simétrica lo primero ha sido crear una serie de frases predefinidas que se mostraran por consola donde solo se guarda una frase por usuario cada vez mediante el método *cypher* y *AES* que nos servirá para encriptar los mensajes, y mediante la clase *scanner* que nos permitirá leer los datos por consola mejor.

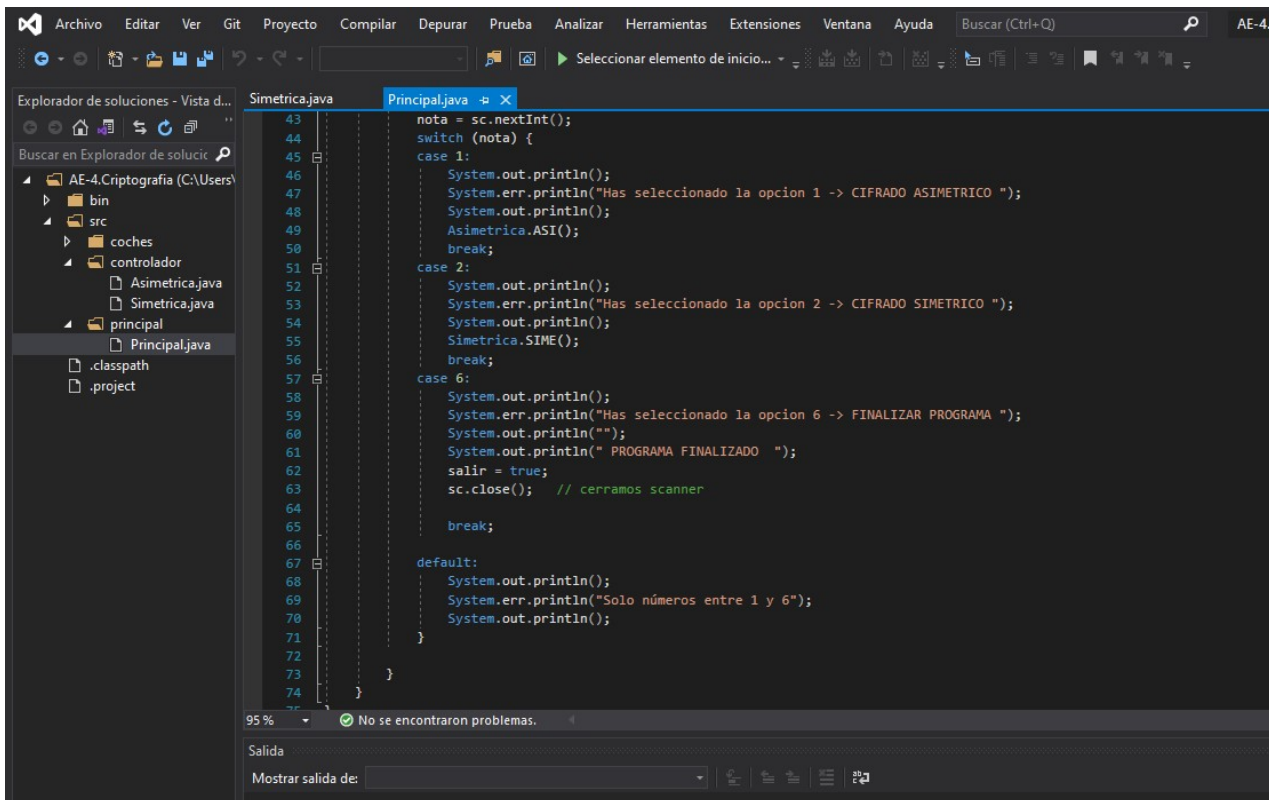


```
57 switch (opcion) {
58     case 1:
59         System.out.println( );
60         System.err.println("***** ENCRIPCACION DE FRASE *****");
61
62         System.out.println("POR FAVOR INTRODUCIR LA FRASE ");
63         datos = sc.nextLine();
64         // esto es datos por consola\\ String mensajeOriginal = "Un gran poder implica
65         // una gran responsabilidad";
66         // El cifrador trabaja con bytes, lo convertimos
67
68         // Ciframos el mensaje original
69         byte[] bytesMensajeOriginal = datos.getBytes();
70         cifrador.init(Cipher.ENCRYPT_MODE, paloEspartano);
71         byte[] bytesMensajeCifrado = cifrador.doFinal(bytesMensajeOriginal);
72         // El cifrador devuelve una cadena de bytes
73         mensajeCifrado = new String(bytesMensajeCifrado);
74         // System.out.println("Paso 5.2: Mensaje Original: " + datos);// aqui sale el
75         // mensaje original
76         // System.out.println("Paso 5.3: Mensaje Cifrado: " + mensajeCifrado);// este es
77         // el mensaje cifrado
78         cifrador.init(Cipher.DECRYPT_MODE, paloEspartano);
79         byte[] bytesMensajeDescifrado = cifrador.doFinal(bytesMensajeCifrado);
80
81         mensajeDescifrado = new String(bytesMensajeDescifrado);
82         System.out.println();
83         break;
84
85     case 2:
86         // aqui muestro el mensaje pero esta encriptado
87         System.out.println( );
88         System.err.println("***** MOSTRAR FRASE ENCRIPCADA *****");
89         System.out.println("Mensaje Cifrado: " + mensajeCifrado);
```



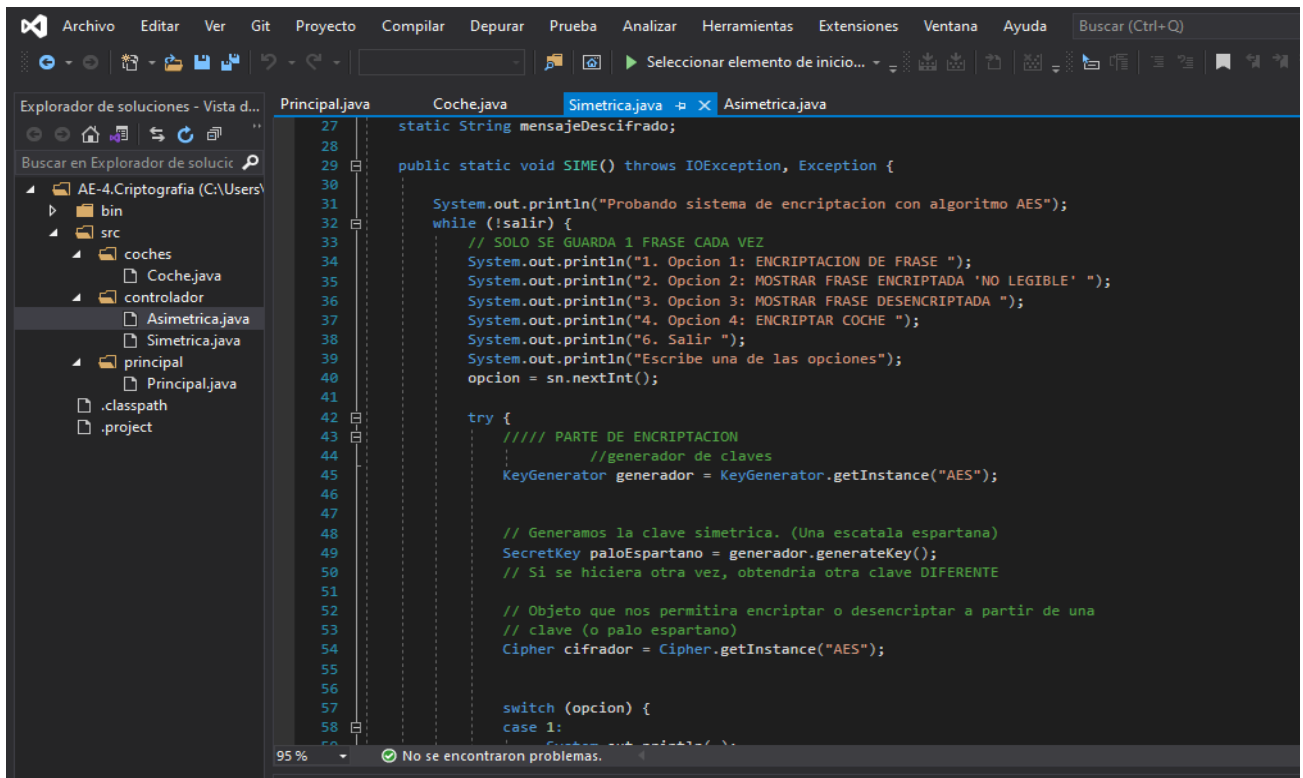
```
1 package principal;
2
3 import java.util.Scanner;
4
5 import controlador.Asimetrica;
6 import controlador.Simetrica;
7
8 public class Principal {
9     public static boolean salir = false;
10     public static int opcion; // Guardaremos la opcion del usuario
11     static Scanner sn = new Scanner(System.in);
12     static Scanner sc = new Scanner(System.in);
13     static String datos;
14     static String matricula;
15     static String marca;
16     static String modelo;
17     static String precio;
18     static int nota;
19     static String mensajeCifrado;
20
21     public static void main(String[] args) throws Exception {
22         /*
23          * String casa;
24          * Scanner sc = new Scanner(System.in);
25          * casa=sc.nextLine();
26          * System.out.println(casa);
27          */
28         while (!salir) {
29             System.out.println("POR FAVOR SELECCIONA UN METODO DE CIPHER: ");
```

También guardaremos la elección que haya hecho el usuario mediante la variable opción, y una vez finalizado cerraremos *scanner*.



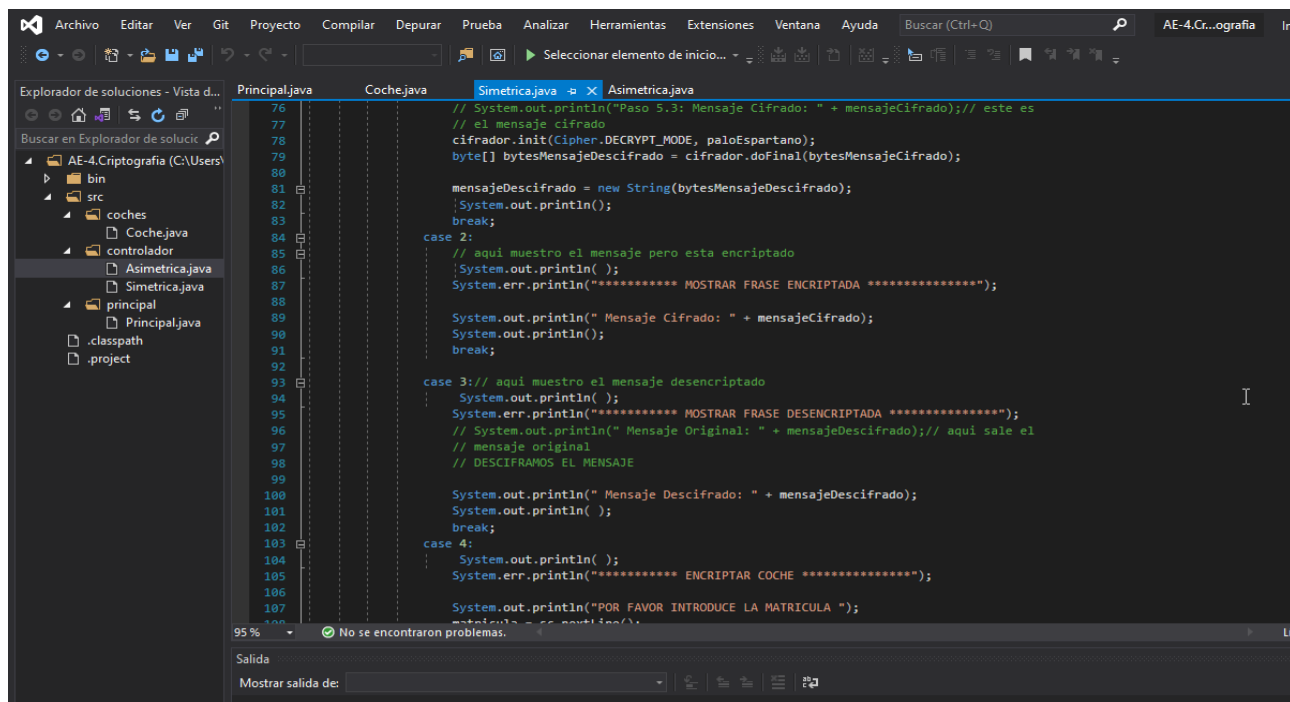
```
43 nota = sc.nextInt();
44 switch (nota) {
45     case 1:
46         System.out.println();
47         System.err.println("Has seleccionado la opcion 1 -> CIFRADO ASIMETRICO ");
48         System.out.println();
49         Asimetrica.ASI();
50         break;
51     case 2:
52         System.out.println();
53         System.err.println("Has seleccionado la opcion 2 -> CIFRADO SIMETRICO ");
54         System.out.println();
55         Simetrica.SIME();
56         break;
57     case 6:
58         System.out.println();
59         System.err.println("Has seleccionado la opcion 6 -> FINALIZAR PROGRAMA ");
60         System.out.println("");
61         System.out.println(" PROGRAMA FINALIZADO ");
62         salir = true;
63         sc.close(); // cerramos scanner
64
65     break;
66
67     default:
68         System.out.println();
69         System.err.println("Solo números entre 1 y 6");
70         System.out.println();
71 }
72 }
73 }
74 }
```

Después, tras crear el coche y darle sus atributos, mediante *while* le daremos al usuarios las opciones para claves que hemos generado las con *keygenerator*



```
27 static String mensajeDescifrado;
28
29 public static void SIME() throws IOException, Exception {
30
31     System.out.println("Probando sistema de encriptacion con algoritmo AES");
32     while (!salir) {
33         // SOLO SE GUARDA 1 FRASE CADA VEZ
34         System.out.println("1. Opcion 1: ENCRYPTACION DE FRASE ");
35         System.out.println("2. Opcion 2: MOSTRAR FRASE ENCRYPTADA 'NO LEGIBLE' ");
36         System.out.println("3. Opcion 3: MOSTRAR FRASE DESENCRIPTADA ");
37         System.out.println("4. Opcion 4: ENCRYPTAR COCHE ");
38         System.out.println("6. Salir ");
39         System.out.println("Escribe una de las opciones");
40         opcion = sn.nextInt();
41
42         try {
43             // PARTE DE ENCRYPTACION
44             // generador de claves
45             KeyGenerator generador = KeyGenerator.getInstance("AES");
46
47             // Generamos la clave simetrica. (Una escatola espartana)
48             SecretKey paloEspartano = generador.generateKey();
49             // Si se hiciera otra vez, obtendria otra clave DIFERENTE
50
51             // Objeto que nos permitira encriptar o desencriptar a partir de una
52             // clave (o palo espartano)
53             Cipher cifrador = Cipher.getInstance("AES");
54
55             switch (opcion) {
56                 case 1:
57                     // ...
58                 case 2:
59                     // ...
60                 case 3:
61                     // ...
62                 case 4:
63                     // ...
64                 case 6:
65                     // ...
66             }
67         } catch (Exception e) {
68             e.printStackTrace();
69         }
70     }
71 }
```

Tras ello mostraremos el mensaje encriptado y mediante *break* lo mostraremos desencriptado



```
// System.out.println("Paso 5.3: Mensaje Cifrado: " + mensajeCifrado); // este es
// el mensaje cifrado
cifrador.init(Cipher.DECRYPT_MODE, paloEspartano);
byte[] bytesMensajeDescifrado = cifrador.doFinal(bytesMensajeCifrado);

mensajeDescifrado = new String(bytesMensajeDescifrado);
System.out.println();
break;

case 2:
    // aqui muestro el mensaje pero esta encriptado
    System.out.println();
    System.err.println("***** MOSTRAR FRASE ENCRYPTADA *****");

    System.out.println(" Mensaje Cifrado: " + mensajeCifrado);
    System.out.println();
    break;

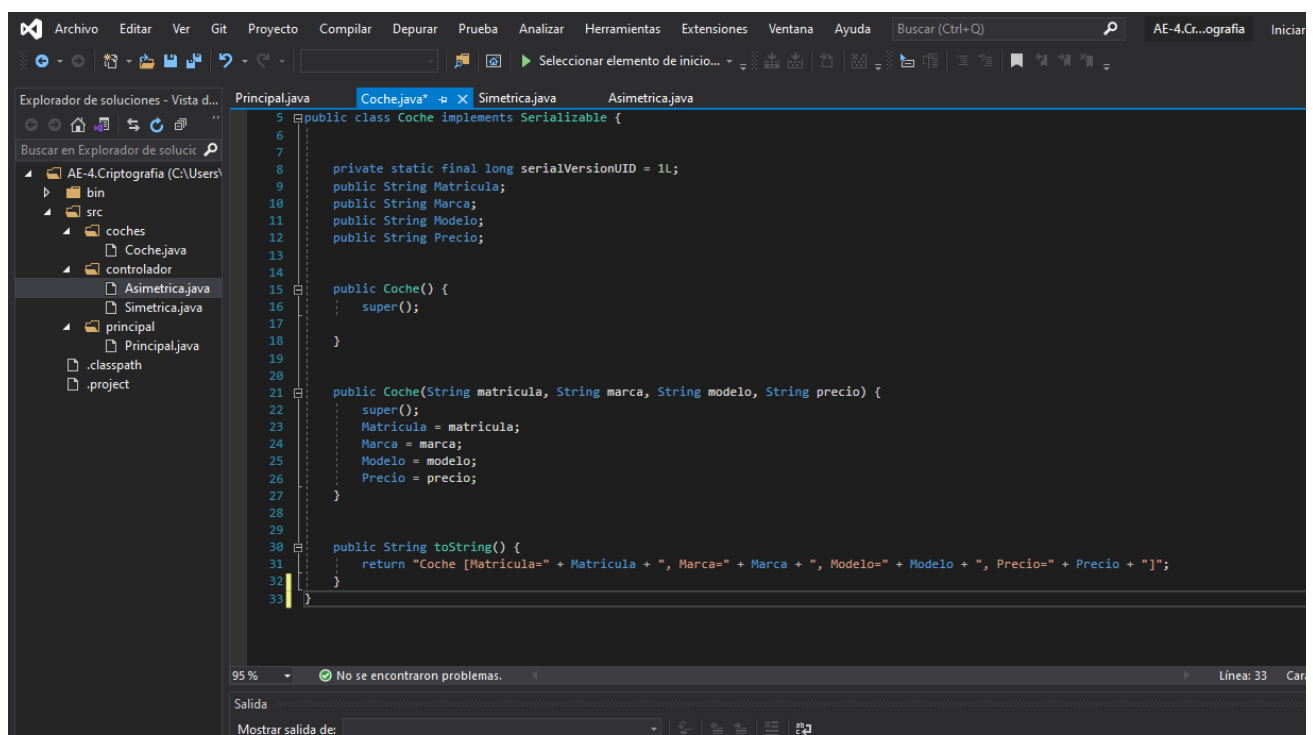
case 3: // aqui muestro el mensaje desencriptado
    System.out.println();
    System.err.println("***** MOSTRAR FRASE DESENCRIPTADA *****");
    // System.out.println(" Mensaje Original: " + mensajeDescifrado); // aqui sale el
    // mensaje original
    // DESCIFRAMOS EL MENSAJE

    System.out.println(" Mensaje Descifrado: " + mensajeDescifrado);
    System.out.println();
    break;

case 4:
    System.out.println();
    System.err.println("***** ENCRYPTAR COCHE *****");

    System.out.println("POR FAVOR INTRODUCE LA MATRICULA ");
    matricula = sc.nextLine();
```

Para crear los atributos del coche, en la clase coche simplemente creamos una cadena *string* con los atributos *matricula*, *modelo*, *precio* y *marca*.



```
public class Coche implements Serializable {

    private static final long serialVersionUID = 1L;
    public String Matricula;
    public String Marca;
    public String Modelo;
    public String Precio;

    public Coche() {
        super();
    }

    public Coche(String matricula, String marca, String modelo, String precio) {
        super();
        Matricula = matricula;
        Marca = marca;
        Modelo = modelo;
        Precio = precio;
    }

    public String toString() {
        return "Coche [Matricula=" + Matricula + ", Marca=" + Marca + ", Modelo=" + Modelo + ", Precio=" + Precio + "];";
    }
}
```

Para realizar la encriptacion asimétrica primero no importaremos *keypair* ni *keypargenerator* después con la clave cifradora *RSA* lo cifraremos para desenscriptarlo mas adelante con *break*.

```
45
46
47     case 1:
48         System.out.println( );
49         System.err.println("***** ENCRYPTACION DE FRASE *****");
50         System.out.println( );
51         System.out.println("POR FAVOR INTRODUCE LA FRASE ");
52         datos = sc.nextLine();
53         // Obtenemos el par de claves (publica y privada)
54         // Objeto que nos permitira encriptar o desencriptar a partir de una
55         // clave (o palo espartano)
56         KeyPairGenerator generador = KeyPairGenerator.getInstance("RSA");
57         KeyPair claves = generador.generateKeyPair();
58         // Ahora el cifrador lo configuramos para que use la clave simetrica
59         // para encriptar
60
61         Cipher cifrador = Cipher.getInstance("RSA");
62         // esto es datos por consola\\ String mensajeOriginal = "Un gran poder implica
63         // una gran responsabilidad";
64         // El cifrador trabaja con bytes, lo convertimos
65         cifrador.init(Cipher.ENCRYPT_MODE, claves.getPublic());
66         byte[] bytesMensajeOriginal = datos.getBytes();
67         // ciframos el mensaje
68         byte[] bytesMensajeCifrado = cifrador.doFinal(bytesMensajeOriginal);
69         // El cifrador devuelve una cadena de bytes
70
71         mensajeCifrado = new String(bytesMensajeCifrado);
72         cifrador.init(Cipher.DECRYPT_MODE, claves.getPrivate());
73         byte[] bytesMensajeDescifrado = cifrador.doFinal(bytesMensajeCifrado);
74         mensajeDescifrado = new String(bytesMensajeDescifrado);
75         System.out.println( );
76         break;
77     case 2:
78         System.out.println( );
79         // aqui muestra el mensaje que esta encriptado
80         System.out.println(mensajeCifrado);
81         break;
82     default:
83         System.out.println("Solo números entre 1 y 6");
84         break;
85 }
```

```
106
107
108     case 3:
109         Coche coche = new Coche(matricula, marca, modelo, precio);
110         // no dice nada el requerimiento 2 de visualizar los datos
111         // así que meto la visualización de los datos
112         // AQUI CIFRO EL COCHE
113         KeyGenerator generador_clave_coche = KeyGenerator.getInstance("AES");
114         SecretKey paloCoche = generador_clave_coche.generateKey();
115         Cipher cifrador_coche = Cipher.getInstance("AES");
116         cifrador_coche.init(Cipher.ENCRYPT_MODE, paloCoche);
117         SealedObject so = new SealedObject(coche, cifrador_coche);
118         System.out.println( );
119         System.out.println(" COCHE CIFRADO: " + so);
120         System.out.println( );
121         // AQUI DESCIFRO EL COCHE
122         cifrador_coche.init(Cipher.DECRYPT_MODE, paloCoche);
123         Coche coche_descifrado = (Coche) so.getObject(cifrador_coche);
124         System.out.println( );
125         System.out.println("COCHE DESCIFRADO : " + coche_descifrado);
126         System.out.println( );
127         break;
128     case 4:
129         System.out.println( );
130         // aqui muestra el mensaje que esta encriptado
131         System.out.println(coche_descifrado);
132         salir = true;
133         break;
134     case 5:
135         System.out.println( );
136         System.out.println("VOLVIENDO AL MENU PRINCIPAL ");
137         break;
138     default:
139         System.out.println("Solo números entre 1 y 6");
140         break;
141 }
```