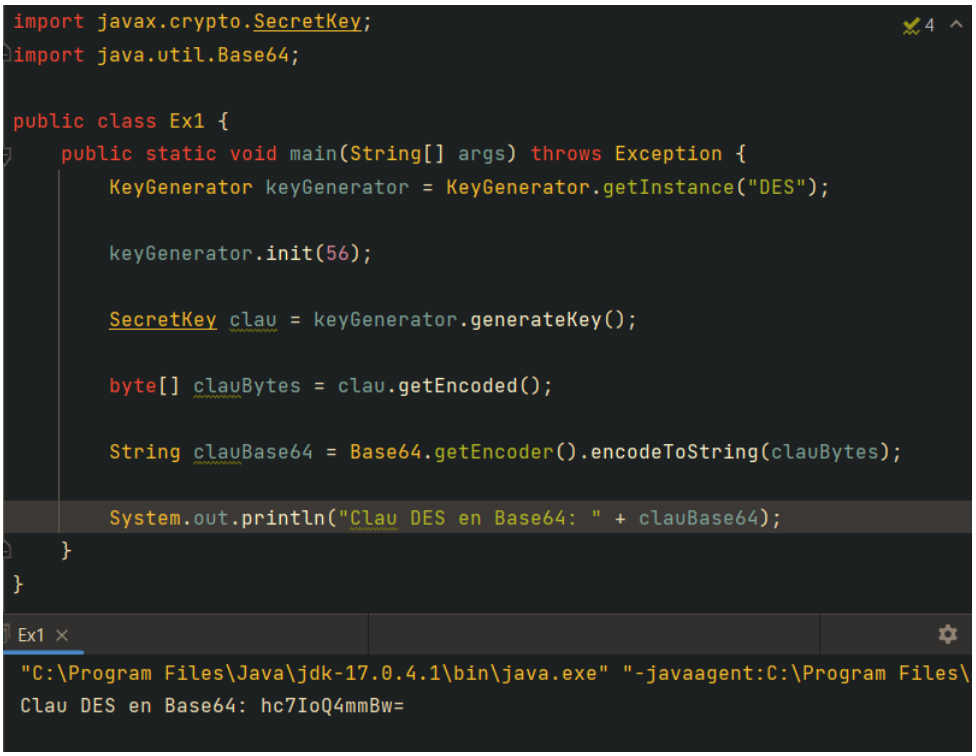
UF1.NF1.Criptografia

1.- Crea una clau simètrica DES i mostra el valor de la clau en Base64.



```
import javax.crypto.SecretKey;
import java.util.Base64;

public class Ex1 {
    public static void main(String[] args) throws Exception {
        KeyGenerator keyGenerator = KeyGenerator.getInstance("DES");

        keyGenerator.init(56);

        SecretKey clau = keyGenerator.generateKey();

        byte[] clauBytes = clau.getEncoded();

        String clauBase64 = Base64.getEncoder().encodeToString(clauBytes);

        System.out.println("Clau DES en Base64: " + clauBase64);
    }
}
```

```
Ex1
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-javaagent:C:\Program Files\
Clau DES en Base64: hc7IoQ4mmBw=
```

2.- Crea un hash amb l'algorisme SHA-512 del teu nom i i cognoms.

```java
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class Ex2 {
    public static void main(String[] args) {
        String texto = "aram mateos andres";

        try {
            MessageDigest instancia = MessageDigest.getInstance("SHA-512");

            byte[] bytes = texto.getBytes(StandardCharsets.UTF_8);

            byte[] hash = instancia.digest(bytes);

            StringBuilder hexString = new StringBuilder();
            for (byte b : hash) {
                String hex = Integer.toHexString(0xff & b);
                if (hex.length() == 1) hexString.append('0');
                hexString.append(hex);
            }

            String hashSHA512 = hexString.toString();

            System.out.println("Hash SHA-512 de \"" + texto + "\": " + hashSHA512);

        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
    }
}
```

```
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.2.1\lib\idea_rt.jar=62909:C:\Program Files\JetBrains\IntelliJ IDEA 202
Hash SHA-512 de "aram mateos andres": c020747b49dad8f60b6be5c49e1ecd87c6ac4a438bc406a98c00f6b253d0667c3451f9a22a022bf85ec8084c661c178e3d610fd6f5b335e3997c7eb87670b1b9

Process finished with exit code 0
```

3.- Crea una clau simètrica AES a partir d'un text codificat (hash) amb SHA-256. Mostra el valor de la clau en Hexadecimal.

```java
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;

public class Ex3 {
    public static void main(String[] args) {
        String texto = "Hola caracola";

        try {
            MessageDigest sha256Digest = MessageDigest.getInstance("SHA-256");
            byte[] hash = sha256Digest.digest(texto.getBytes());
            byte[] claveAESBytes = Arrays.copyOf(hash, 16);

            SecretKey claveAES = new SecretKeySpec(claveAESBytes, "AES");
            byte[] claveHexBytes = claveAES.getEncoded();
            StringBuilder claveHexadecimal = new StringBuilder();
            for (byte b : claveHexBytes) {
                String hex = String.format("%02x", b);
                claveHexadecimal.append(hex);
            }

            System.out.println("Clave AES en hexadecimal: " + claveHexadecimal);

        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
    }
}
```

```
Ex3 ×
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-javaagen
Clave AES en hexadecimal: 9437086a7eef592f3f2373afd9158a7d
```

4.- Modifica l'exercici 2 per aplicar un salt abans de fer la codificació. Busca com generar el salt utilitzant la classe SecureRandom. Per a què serveix? És diferent al hash resultant?

```java
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.util.Arrays;

public class Ex4 {
    1 usage
    private static byte[] concatenateArrays(byte[] a, byte[] b) {
        byte[] result = Arrays.copyOf(a, a.length + b.length);
        System.arraycopy(b, 0, result, a.length, b.length);
        return result;
    }

    public static void main(String[] args) {
        String text = "aram mateos andres";

        try {
            SecureRandom saltSegur = new SecureRandom();
            byte[] salt = new byte[16];
            saltSegur.nextBytes(salt);

            byte[] bytes = concatenateArrays(salt, text.getBytes(StandardCharsets.UTF_8));

            MessageDigest instancia = MessageDigest.getInstance("SHA-512");
            byte[] hash = instancia.digest(bytes);

            StringBuilder hexString = new StringBuilder(2 * hash.length);
            for (byte b : hash) {
                String hex = String.format("%02x", b);
                hexString.append(hex);
            }

            String hashSHA512 = hexString.toString();

            System.out.println("Hash SHA-512 de \"" + text + "\" amb el salt: " + hashSHA512);

        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
    }
}
```

```
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Intell
Hash SHA-512 de "aram mateos andres" amb el salt: 4180acc7fcf10cf650128a3940b6613cb805d39679a7b
```

5.- Crea una base de dades que es digui UF1Cripto<nom><cognoms> (pots utilitzar el SGBD que vulguis), que contingui, per cada usuari, el hash de la seva contrasenya i el salt aplicat.

No he sigut capaç de connectar la base de dades (Hem dona error a l'hora de connectar el ide amb la bdd)

6.- Crea un mètode que, a partir d'un usuari i contrasenya, guardi a la base de dades de l'exercici anterior, el salt i el seu hash. (Per fer el salt, genera un enter aleatori i passa'l a Hexadecimal).

7.- Crea un mètode que, a partir d'un usuari i contrasenya, verifiqui aplicant el hash i el salt corresponent, si coincideix amb el de la base de dades.

8.- Crea un mètode que retorni un parell de claus asimètriques.

```java
public class Ex8 {
    1 usage
    public static KeyPair generarParClavesAsimetricas() {
        try {
            KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance("RSA");

            keyPairGenerator.initialize(2048);

            KeyPair keyPair = keyPairGenerator.generateKeyPair();

            return keyPair;
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
            return null;
        }
    }

    public static void main(String[] args) {
        KeyPair parClaves = generarParClavesAsimetricas();

        if (parClaves != null) {
            System.out.println("Claus generades");
            System.out.println("Clau publica: " + parClaves.getPublic());
            System.out.println("Clau privada: " + parClaves.getPrivate());
        } else {
            System.out.println("Error al generar les claus");
        }
    }
}
```

```
Ex8 ×
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\In
Claus generades
Clau publica: Sun RSA public key, 2048 bits
  params: null
  modulus: 219622100080607283149867244940915781322564064261388451827551116473852820274065347
  public exponent: 65537
Clau privada: SunRsaSign RSA private CRT key, 2048 bits
  params: null
  modulus: 219622100080607283149867244940915781322564064261388451827551116473852820274065347
  private exponent: 655226988087046081417791370784023652945577897442787693113537880106658417

Process finished with exit code 0
```

9.- Crea un mètode encripti una contrasenya a partir d'una clau privada.

10.- Modifica l'exercici anterior perquè encripti el hash de la contrasenya.

11.- Crea un mètode que desencripti la informació encriptada de l'exercici anterior a partir d'una clau pública.

12.- Utilitzant el keytool, crea una clau simètrica i mostra-la pel cmd.

```
C:\Users\aramm>keytool -genkeypair -alias aram -keyalg RSA -keysize 2048 -keystore mykeystore.jks -storepass 123456
What is your first and last name?
  [Unknown]:  aram mateos
What is the name of your organizational unit?
  [Unknown]:  iesvidreres
What is the name of your organization?
  [Unknown]:  ies
What is the name of your City or Locality?
  [Unknown]:  vidreres
What is the name of your State or Province?
  [Unknown]:  girona
What is the two-letter country code for this unit?
  [Unknown]:  ES
Is CN=aram mateos, OU=iesvidreres, O=ies, L=vidreres, ST=girona, C=ES correct?
  [no]:  s
```

```
C:\Users\aramm>keytool -list -v -keystore mykeystore.jks -storepass 123456 -alias aram
Alias name: aram
Creation date: 2 oct 2023
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=aram mateos, OU=iesvidreres, O=ies, L=vidreres, ST=girona, C=ES
Issuer: CN=aram mateos, OU=iesvidreres, O=ies, L=vidreres, ST=girona, C=ES
Serial number: 7e4637d8f37f2896
Valid from: Mon Oct 02 21:50:27 CEST 2023 until: Sun Dec 31 20:50:27 CET 2023
Certificate fingerprints:
         SHA1: 96:8C:6B:50:25:3D:54:1B:61:8A:A3:2F:61:76:02:A4:B4:71:8D:A5
         SHA256: C4:D9:21:A8:B2:41:77:DC:26:A6:74:73:54:3F:BF:78:7A:12:B6:CB:F1:69:1D:D6:F0:52:41:03:75:6B:33:38
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: AF CE 3A 76 45 77 45 0A   19 34 5D 6D DC C7 BD E7  ..:vEwE..4]m....
0010: 70 6B 77 81                                        pkw.
]
]
```

13.- Crea un mètode que mostri la clau creada a l'exercici anterior

14.- Crea un mètode que crei una clau simètrica al keytool i la mostri per pantalla.