# Pràctica U5P1. Resum de missatges

Un “message digest” o resum de missatges (funció hash) és una marca digital d’un bloc de dades. Existeixen un gran nombre d’algorismes disenyats per processar estos message digest, els més coneguts SHA-1,SHA-256 i MD5.

Java té el paquet **security** que ens permet utilitzar la **classe MessageDigest**. Esta classe permet a les aplicacions implementar algorismes de resum de missatges. Disposa d’un constructor protegit, per crear l’objecte MessageDigest utilitzarem el mètode *getInstance (String algorithm)*

Alguns mètodes de la classe són:

| MÉTODOS | MISIÓN |
|---|---|
| **public static MessageDigest getInstance(String algoritmo)**<br><br>**public static MessageDigest getInstance(String algoritmo, String proveedor)** | Devuelve un objeto *MessageDigest* que implementa el algoritmo de resumen especificado<br><br>En el primer caso, los proveedores de seguridad se buscan según el orden establecido en el fichero **java.security**. En el segundo caso se busca el proveedor dado. Nombres válidos para el proveedor de seguridad predeterminado de *Sun* son SHA, SHA-1 y MD5<br><br>Puede lanzar la excepción *NoSuchAlgorithmException* si no hay proveedor que implemente el algoritmo dado. Si el nombre de proveedor no se encuentra se produce *NoSuchProviderException* |
| **void update(byte input)** | Realiza el resumen del byte especificado |
| **void update(byte[] input)** | Realiza el resumen del array de bytes especificado |
| **byte[] digest()** | Completa el cálculo del valor hash, devuelve el resumen obtenido |
| **byte [] digest (byte [] entrada)** | Realiza una actualización final sobre el resumen utilizando el array de bytes indicado en el argumento, y luego completa el cálculo de resumen |
| **void reset()** | Reinicializa el objeto resumen para un nuevo uso |

| | |
|---|---|
| **int getDigestLength()** | Devuelve la longitud del resumen en bytes, o 0 si la operación no está soportada por el proveedor |
| **String getAlgorithm()** | Devuelve un String que identifica el algoritmo |
| **Provider getProvider()** | Devuelve el proveedor del objeto |
| **static boolean isEqual(byte[] digesta, byte[] digestb)** | Comprueba si dos mensajes resumen son iguales. Devuelve true si son iguales y false en caso contrario |

Veiem un exemple en el que es crea un resum d'un text pla, s'utilitza el mètode MessageDigest.getInstance ("SHA") i s'obté instància amb l'algorisme SHA:

```
1  package u5Exemple1_SHA;
2  import java.security.MessageDigest;
3  import java.security.Provider;
4
5  public class U5Exemple1_SHA {
6      public static void main(String[] args) throws Exception {
7          String text = "Sóc el contingut d'un text";
8          System.out.println("Text origen per a hash " + text);
9
10         // CREE OBJECTE MessageDigest
11         MessageDigest md = MessageDigest.getInstance("SHA");
12
13           // ES POT CREAR EL RESUM AMB CLAU UTILITZANT digest(bytes[])
14         String clave="ClauXifratge";
15         byte dataBytes[]=text.getBytes();
16         md.update(dataBytes);
17         byte resum_amb_clau[]=md.digest(clave.getBytes());
18
19         // S'INTRODUEIX EL TEXT EN BYTES A RESUMIR
20         md.update(text.getBytes());
21
22         // ES CALCULA EL RESUM
23         byte resum[] = md.digest();
24
25         System.out.println("Nonmbre de bytes " + md.getDigestLength());
26         System.out.println("Algorisme " + md.getAlgorithm());
27         System.out.println("Missatge resum " + resum_amb_clau + new String(resum));
28
29         // CONVERTEISC L'ARRAY DE BYTES A HEXADECIMAL
30         StringBuffer hexString = new StringBuffer();
31         for (int i = 0; i < resum.length; i++) {
32             String hex = Integer.toHexString(0xff & resum[i]);
33             if (hex.length() == 1)
34                 hexString.append('0');
35             hexString.append(hex);
36         }
37
38         System.out.println("Missatge en hexadecimal : " + hexString.toString());
39         Provider proveedor = md.getProvider();
40         System.out.println("Proveidor " + proveedor.toString());
41     }
42 }
```

L'execució ens oferirà el següent resultat:

```
Text origen per a hash Sóc el contingut d'un text
Nonmbre de bytes 20
Algorisme SHA
Missatge resum éÅ,Ã¤¦à=)Mœ@ï6ü°,^®é
Missatge en hexadecimal : e9c582c3a4a6e03d294d9c40ef36fcb0825eaee9
Proveidor SUN version 1.8
```

Es pot crear un resum xifrat **amb clau** utilitzant el mètode digest (bytes[]) on es proporciona la clau en un array en bytes.
S'implementa amb el següent codi:

```
13          // ES POT CREAR EL RESUM AMB CLAU UTILITZANT digest(bytes[])
14        String clave="ClauXifratge";
15        byte dataBytes[]=text.getBytes();
16        md.update(dataBytes);
17        byte resum_amb_clau[]=md.digest(clave.getBytes());
```

Veiem un exemple en el que generem hash d'un fitxer:

```java
package u5Exemple2_SHA;
import java.io.FileInputStream;
import java.security.MessageDigest;
import java.security.Provider;

public class U5Exemple2_SHA {
    public static void main(String[] args) throws Exception {

        // CREE OBJECTE MessageDigest
        MessageDigest md = MessageDigest.getInstance("SHA");
        FileInputStream fis = new FileInputStream("fitxer.txt");

        byte [] dataBytes = new byte[1024];
        int nread =0;
        System.out.println("Contingut del fitxer: ");
        while ((nread=fis.read(dataBytes))!=-1 ) {
            System.out.println(new String(dataBytes));
            md.update(dataBytes, 0, nread);
        }

        byte[] mdbytes=md.digest();
        System.out.println("Nombre de bytes: "+md.getDigestLength());
        System.out.println("Algorisme: "+md.getAlgorithm());
        System.out.println("Missatge resum: "+new String(mdbytes));

        // CONVERTEISC EL MISSATGE RESUM DE ARRAY DE BYTES A HEXADECIMAL
        StringBuffer hexString = new StringBuffer();
        for (int i = 0; i < mdbytes.length; i++)
            hexString.append(Integer.toHexString(0xff & mdbytes[i]));

        System.out.println("Fitxer en hexadecimal:  " + hexString.toString());
        Provider proveedor = md.getProvider();
        System.out.println("Proveidor " + proveedor.toString());

        fis.close();
    }
}
```

L'execució ens oferirà el següent resultat:

```
Contingut del fitxer:
MISSATGE DINS FITXER


Nombre de bytes: 20
Algorisme: SHA
Missatge resum: •▫¼¥-èÔ▫¢▫qÌµdQtpëß
Fitxer en hexadecimal:  951a1ebca5ade8d47a21571ccb564517470ebdf
Proveidor SUN version 1.8
```

## Entregable Pràctica U5P1_1

Realitza un programa que genere un resum utilitzant l'algorisme MD5. El programa demanarà el text a l'usuari per teclat. Comprova el nombre de bytes generats.

## Entregable Pràctica U5P1_2

Suposem que volem guardar un missatge (objecte String) en un fitxer, però volem estar segurs de que, a l'hora de llegir el fitxer, el contingut no ha estat manipulat.
Per això, a més a més de guardar el missatge en un fitxer («data.dat»), generarem un resum (amb SHA) i el guardarem en un altre fitxer («hash.dat»).
Per a la implementació es requereixen dues classes:
**Genera_Resum** → Guarda el missatge al fitxer data.dat, genera el resum i el guarda al fitxer hash.dat.
**Llig_Verifica** → Llig el missatge del fitxer data.dat, obté el resum i el compara amb el resum del fitxer hash.dat. Si són iguals, les dades són vàlides.

Es proporciones les dues classes parcialment desenvolupades i es demana completar-les.

- Classe Genera_Resum.

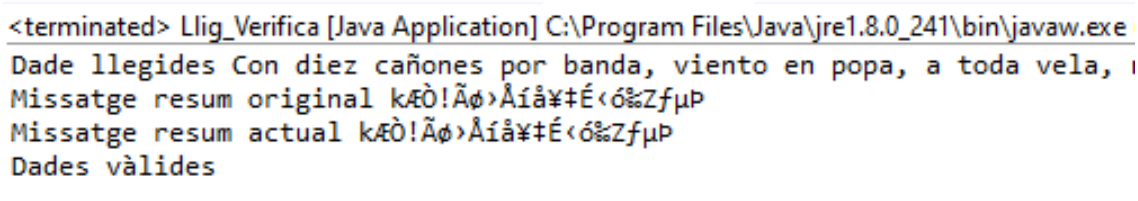
```java
import java.io.FileNotFoundException;

public class Genera_Resum {
    public static void main(String args[]) {
        try {
            // CREE OBJECTE MessageDigest
            MessageDigest md = MessageDigest.getInstance("SHA");

            // MISSATGE
            String missatge = "Con diez cañones por banda," + " viento en popa, a toda vela,"
                    + " no corta el mar, sino vuela" + " un velero bergantín." + " Bajel pirata que llaman,"
                    + " por su bravura, el Temido," + " en todo mar conocido" + " del uno al otro confín.";

            // S'INTRODUEIX EL TEXT EN BYTES A RESUMIR
            // ...
            // ES CALCULA EL RESUM
            // ...
            System.out.println("Missatge resum " + new String(resum));

            // ESCRIC MISSATGE COM A OBJECTE AL FITXER data.dat
            FileOutputStream fosDatos = new FileOutputStream("data.dat");
            ObjectOutputStream oosDatos = new ObjectOutputStream(fosDatos);
            // ...

            // DESPRÉS ESCRIC EL RESUM COM A OBJECTE AL FITXER hash.dat
            FileOutputStream fosHash = new FileOutputStream("hash.dat");
            ObjectOutputStream oosHash = new ObjectOutputStream(fosHash);
            // ...

            oosDatos.close(); // TANQUE FLUX
            fosDatos.close(); // TANQUE FITXER
            oosHash.close(); // TANQUE FLUX
            fosHash.close(); // TANQUE FITXER

        } catch (IOException | NoSuchAlgorithmException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

- Classe Llig_Verifica

```java
import java.io.FileInputStream;

public class Llig_Verifica {
    public static void main(String args[]) {
        try {
            // PRIMER LLIG OBJECTE MISSATGE DE TIPUS string DEL FITXER data.dat
            InputStream fisDades = new FileInputStream("data.dat");
            ObjectInputStream oisDades = new ObjectInputStream(fisDades);
            Object o = oisDades.readObject();
            String dades = (String) o;
            System.out.println("Dade llegides " + dades);

            // DESPRÉS LLIG OBJECTE resum DE TIPUS string DEL FITXER hash.dat
            InputStream fisHash = new FileInputStream("hash.dat");
            ObjectInputStream oisHash = new ObjectInputStream(fisHash);
            o = oisHash.readObject();
            byte resum_original[] = (byte[]) o;
            System.out.println("Missatge resum original " + new String(resum_original));

            // GENERE RESUM DEL MISSATGE
            // ...
            // ...
            // ...
            System.out.println("Missatge resum actual " + new String(resum_actual));

            // COMPARE ELS DOS RESUMS
            if (MessageDigest.isEqual(resum_actual, resum_original))
                System.out.println("Dades vàlides");
            else
                System.out.println("Dades no vàlides ");

            oisDades.close(); // TANQUE FLUX
            fisDades.close(); // TANQUE FITXER
            oisHash.close(); // TANQUE FLUX
            fisHash.close(); // TANQUE FITXER

        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

Per a comprovar-ho:

- Dades vàlides: executem primer Genera_Resum i després Llig_Verifica




- Dades no vàlides:
  - Executem Genera_Resum
  - Canviem el nom del fitxer hash.dat per hash10.dat (explorador d'arxius)
  - Canviem el missatge de la classe Genera_Resum (per exemple, canviar diez por ONCE).
  - Tornem a executar Genera_Resum.
  - Eliminem l'arxiu hash.dat i renombrar l'arxiu hash10.dat per hash.dat
  - Executem Llig_Verifica