

Tarea 1 del RA5.

Resumen (hash) y validación de entrada con expresiones regulares.

Enunciado parte 1. MessageDigest

El objetivo de esta actividad es trabajar con las clases que nos permiten hacer resúmenes (*hashes*) en Java, además, de filtrar las entradas con expresiones regulares.

Deberás crear una aplicación que presente un menú con las siguientes opciones:

1. Listar los algoritmos de hash soportados.
Estudiar la clase *Security*** y la clase ***Provider***.
2. De String a hash
Deberás pedir un texto por teclado el algoritmo de hash. A continuación, obtendrá el hash del texto y lo mostrará por pantalla en formato hexadecimal.
3. De fichero a *hash*.
Esta opción pedirá por teclado el *path* de un fichero y el algoritmo de *hash*. A continuación, obtendrá el *hash* de un fichero y lo mostrará por pantalla en formato hexadecimal.
4. Salir.

Deberás crear estas dos clases.

Clase App:

Esta clase será la encargada de mostrar el menú con las opciones e interactuará con el usuario pidiendo los valores por teclado y mostrando los *hashes* obtenidos.

Clase HashThing:

Esta clase será la encargada de trabajar con la clase MessageDigest para hacer los resúmenes. Al menos tendrá los siguientes métodos:

- public String getAlgorithms()
 - Devuelve un String con todos los algoritmos permitidos de hash.
- public String getStringHash(String algorithm, String plaintext)
 - Tiene como argumentos de entrada el nombre del algoritmo de *hash* y el String para hacer el resumen.
 - Devuelve el resumen del texto plano como String en formato hexadecimal.

- Public getFilehash(String algorithm, File f)
 - Tiene como argumentos de entrada el nombre del algoritmo de *hash* y un fichero.
 - Devuelve el resumen del fichero como String en formato hexadecimal.

Enunciado parte 2. Expresiones regulares

Deberás sustituir la opción de String a hash por estas dos opciones:

1. De nombre de usuario a *hash*.
 Esta opción pedirá el nombre de usuario por teclado mientras no cumpla el patrón.
 A continuación, pedirá por teclado el nombre del algoritmo de *hash* a utilizar hasta que teclee uno válido.
 Cuando el nombre del usuario coincida con el patrón y el algoritmo de *hash* exista, calculará el *hash* del nombre de del usuario y lo mostrará por pantalla en formato hexadecimal.

El patrón a cumplir por el nombre del usuario es el siguiente:

- Debe tener una longitud de 4 a 8 letras (minúsculas) y números,
- Debe empezar por una letra (minúscula) y
- A continuación, le siguen letras (minúsculas) y números.

2. De código de producto a *hash*.
 Deberá tener un comportamiento similar a la anterior, pero con un código de producto que debe tener el siguiente formato:

ZZZ-CC-CODIG-ADVE

Donde:

- ZZZ: Código de zona: los 3 primeros caracteres en formato binario indica la zona en la que se suministra el producto. Por ejemplo 000, 001, 010, etc.
- CC: Categoría del producto: los 2 caracteres siguientes deben ser 2 letras en mayúsculas. Las categorías son las siguientes: MT, TO, NO, CO, IR, SE y NE.
- CODIG: Código producto: 5 dígitos numéricos.
- ADVE: Advertencia: 4 caracteres siendo el primero una letra en mayúscula (S, H o P) y los siguientes 3 dígitos. Por ejemplo: S000, H290, H314, P234, P260, etc.

En la clase **App** tendrás que dos métodos para comprobar si se cumple con el patrón o no:

- private static boolean isValidUserName(String input)
 - Devuelve true si la cadena pasada como parámetro cumple con el patrón de nombre de usuario.
- private static boolean isValidCode(String input)
 - Devuelve true si la cadena pasada como parámetro cumple con el código de producto.

***Es importante tratar todas las excepciones y errores que se puedan producir de tal manera que la aplicación responda ante ellos.

Ejemplos de ejecución de las distintas opciones.

```
Elige la operación:
1) Lista de algoritmos de hash.
2) De nombre de usuario a Hash.
3) De código de producto a Hash.
4) De fichero a Hash.
5) Salir.

1
SHA-1
MD2
MD5
SHA-512/256
SHA3-512
SHA-256
SHA-384
SHA-512/224
SHA-512
SHA3-256
SHA-224
SHA3-384
SHA3-224

Elige la operación:
1) Lista de algoritmos de hash.
2) De nombre de usuario a Hash.
3) De código de producto a Hash.
4) De fichero a Hash.
5) Salir.

2
Por favor, introduce el nombre de usuario para convertir a hash: 1pedro
La cadena no cumple con los requisitos. Inténtalo de nuevo.
Por favor, introduce el nombre de usuario para convertir a hash: pedro
La cadena cumple con los requisitos.
Introduce el algoritmo: MD4
No existe el algoritmo de resumen indicado. Inténtalo de nuevo.
Introduce el algoritmo: MD5
c6cc8094c2dc07b700ffcc36d64e2138

3
Por favor, introduce el código del producto para convertir a hash: ADFA
La cadena no cumple con los requisitos. Inténtalo de nuevo.
Por favor, introduce el código del producto para convertir a hash: 101-MT-12345-S000
La cadena cumple los requisitos.
Introduce el algoritmo: SHA-256
49c1d44c94877bcafe7a39dac749741755127e768530876727a5ec707d2697ff

Elige la operación:
1) Lista de algoritmos de hash.
2) De nombre de usuario a Hash.
3) De código de producto a Hash.
4) De fichero a Hash.
5) Salir.
```

4
Por favor, introduce el path o el fichero para hacer hash: leon.jpg
Introduce el algoritmo: MD5
d2b0182756e9caa0f3815b43323014bf

Elige la operación:
1) Lista de algoritmos de hash.
2) De nombre de usuario a Hash.
3) De código de producto a Hash.
4) De fichero a Hash.
5) Salir.