



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

Laboratorio de Computación Salas A y B

Profesor(a): M.C. RENE ADRIAN DAVILA PEREZ

Asignatura: PROGRAMACIÓN ORIENTADA A OBJETOS

Grupo: 01

No de Práctica(s): 03

Integrante(s): 322118311

322094028

322092842

322078673

322067738

*No. de lista o
brigada:* 03

Semestre: 2026-1

Fecha de entrega: 12/09/2025

Observaciones:

CALIFICACIÓN: _____

Índice

1. Introducción	2
2. Marco Teórico	2
3. Desarrollo	3
4. Resultados	4
5. Conclusiones	4
6. Bibliografía	5

1. Introducción

- **Planteamiento del problema.** En esta practica se implementara un programa en java que reciba cadenas desde el método main para posteriormente procesarlas mediante una simulación de una función digestiva generando asi una clave de 32 digitos en formato hexadecimal
- **Motivación.** En la actualidad, el comprender el funcionamiento de la seguridad informatica y la criptografia, toma mas relevancia es por eso que comprender las funciones hash ampliara nuestros conocimientos en estas areas de la computación y su importancia en la integridad de la información.
- **Objetivos.** Para esta practica esperamos poder comprender el funcionamiento asi como la implementación de una función digestiva ademas de ampliar nuestras bases de programacion en lenguaje java, mediante el uso de arrays, colecciones y la manipulación de clases, elementos que nos permitiran implementar este programa.

2. Marco Teórico

Funciones Hash: Las funciones hash o funciones digestivas, son elementos criptográficos que generan una salida única de tamaño fijo para una entrada de tamaño arbitrario. Su objetivo es identificar objetos rápidamente en estructuras como diccionarios o tablas hash y asi lograr la autentificacion de datos. Si dos objetos son iguales deberian dar el mismo hash. [1]

Semillas para generacion de numeros pseudoaleatorios:

Los números pseudoaleatorios se refieren al resultado del proceso de generar una secuencia de números que parece aleatoria, pero que en realidad se genera utilizando un algoritmo determinístico. [2]

Arreglos y colecciones en Java: Un arreglo es una estructura de datos que puede almacenar datos del mismo tipo organizados en una lista ordenada, para acceder a ellos se usan indices, por el contrario el hashmap hace referencia a un lugar estructurado para guardar los datos y acceder a ellos de forma eficiente que guarda los datos en pares Clave-valor, permite buscar datos, borrar e insertar. [3]

Manipulacion de cadenas y conversion de datos: Para dar solucion a la practica es importante conocer los siguientes metodos para la manipulación de cadenas y conversion de datos.

toCharArray: Convierte una cadena de texto en un arreglo de caracteres, en el cual cada elemento del arreglo corresponde a una caracter de la cadena original, lo cual nos permite manipular cada caracter de forma individual.[4]

StringBuilder: Permite crear y manipular cadenas de texto de forma eficiente, StringBuilder a diferencia de la clase String, permite modificar la misma cadena en memoria sin necesidad de generar nuevos objetos Por ejemplo, usando append(),

logramos agregar texto al final de la cadena sin crear un nuevo objeto, ademas existen métodos como insert, delete o reverse.[5]

Integer.toHexString: Convierte un numero de tipo entero en su representacion hexadecimal como cadena String.[6]

Metodo main y argumentos : En Java, los datos que se colocan después del nombre de la clase al ejecutar un programa se reciben en el arreglo args del método main. Cada valor separado por un espacio se guarda como un String dentro de ese arreglo y puedes acceder a ellos con un índice, por ejemplo args[0], args[1]. Así cuando se ejecuta java Ejemplo pedor 763 entonces el programa va a recibir pedro 763 como argumento. [7]

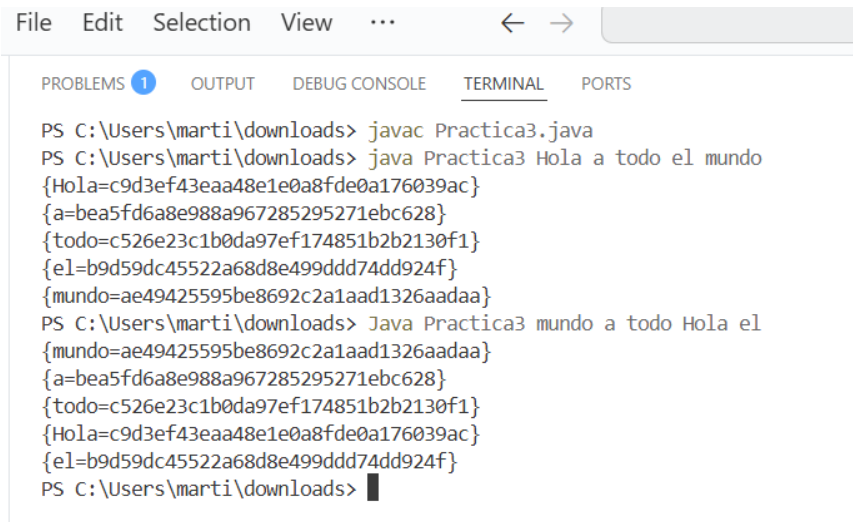
3. Desarrollo

Para poder realizar la práctica, empezamos analizando los conceptos teóricos que se iban a utilizar: funciones hash, la creación de números pseudoaleatorios y el uso de las colecciones (HashMap y ArrayList). Para esto iniciamos importando las clases ArrayList, HashMap Y la clase Random. Crearemos la clase principal a la que llamaremos "Practica3" dentro de la cual tendremos los siguientes metodos:

Método generaHash : El método generaHash recibe una cadena de texto y procede a generar un hash de 32 caracteres hexadecimales. Primero, convertira la cadena en un arreglo de caracteres usando toCharArray() y sumara el valor numérico de cada carácter para obtener una semilla. Esta semilla sera utilizada para inicializar un generador de números pseudoaleatorios (con la clase Random, lo que nos ayudara a que cada palabra produzca un hash reproducible y dependiente de su contenido. Luego, el método genera 32 números aleatorios entre 0 y 15, los convierte a hexadecimal mediante Integer.toHexString() y los agregara a un StringBuilder mediante append(). Finalmente, devolvera la cadena resultante como el hash de la palabra. Simulando así la funcion digestiva

Método main: Lee las palabras que el usuario ingreso, este es el momento en donde las palabras se convierten en un arreglo de caracteres utilizando el metodo toCharArray() e implementamos un ciclo for para poder recorrerlas y transformar las cadenas que tomara desde el metodo main, es decir, las cadenas que ingresamos en terminal mediante el comando "java Practica3 Hola a todo el mundo".

4. Resultados



```
File Edit Selection View ... < >
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\marti\downloads> javac Practica3.java
PS C:\Users\marti\downloads> java Practica3 Hola a todo el mundo
{Hola=c9d3ef43eaa48e1e0a8fde0a176039ac}
{a=bea5fd6a8e988a967285295271ebc628}
{todo=c526e23c1b0da97ef174851b2b2130f1}
{el=b9d59dc45522a68d8e499ddd74dd924f}
{mundo=ae49425595be8692c2a1aad1326aadaa}
PS C:\Users\marti\downloads> Java Practica3 mundo a todo Hola el
{mundo=ae49425595be8692c2a1aad1326aadaa}
{a=bea5fd6a8e988a967285295271ebc628}
{todo=c526e23c1b0da97ef174851b2b2130f1}
{Hola=c9d3ef43eaa48e1e0a8fde0a176039ac}
{el=b9d59dc45522a68d8e499ddd74dd924f}
PS C:\Users\marti\downloads> █
```

Figura 1: Salida

En esta captura de pantalla observamos la salida despues de que tomara las cadenas desde el metodo main, para despues simular la funcion digestiva y generar mediante una semilla, una clave de 32 caracteres alfanumericos para cada una de las cadenas. En la misma captura podemos comprobar que genera una unica clave para cada cadena, lo observamos cambiando el orden de la frase mostrada en la primera ejecucion.

5. Conclusiones

Para poder desarrollar el codigo solicitado para la practica, primero debimos saber y entender que era una funcion digestiva, su funcionamiento y ver un ejemplo para tener una idea, que posteriormente sera una base para hacer el trabajo que se nos he solicitado. Esto principalmente por que una funcion digestiva es el proceso para poder generar un hash.

Recapitulando un Hash transforma un texto ingresado por el usuario a un tipo de cadena alfanumerica, apartir de aqui implementamos un hashmap, que en terminos informales un hashmap es un diccionario en java, apartir de lo mencionado previamente se debe importarla para hacer uso de esta estructura de datos. Por que, si, es una estructura de dato por el manejo de datos que es almacenada (llaves y valores), cabe mencionar que cada llave es unica y no es repetida.

Finalmente podemos concluir que el desarollo de esta practica nos permiti  ampliar nuestro conocimiento en el manejo de arrays, collecciones (En esta caso para utilizando el HashMap) y como manipular las clases.

6. Bibliografía

- [1] Unidad de Apoyo para el Aprendizaje, "Funciones Hash, CUAED/FES Acatlán-UNAM. [En línea]. Disponible: https://repositorio-uapa.cuaed.unam.mx/repositorio/moodle/pluginfile.php/2186/mod_resource/content/1/contenido-uapa/index.html. [Consultado : 12 – sep – 2025].
- [2] Interactive Chaos, "Números pseudoaleatorios, Interactive Chaos, [En línea]. Disponible: <https://interactivechaos.com/es/wiki/numeros-pseudoaleatorios> . [Consultado: 12-sep-2025].
- [3] J. Jesús, "¿Qué es un Hash Map?", Tutoriales Dongee, 13 de abril de 2024. [En línea]. Disponible: <https://www.dongee.com/tutoriales/que-es-un-hash-map/> . [Consultado: 12-sep-2025].
- [4] Oracle, Class String | Java Platform SE 17, Oracle Docs, [En línea]. Disponible: [https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html#toCharArray\(\)](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html#toCharArray()) . [Consultado: 12-sep-2025].
- [5] Oracle, Class StringBuilder | Java Platform SE 17, Oracle Docs, [En línea]. Disponible: <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/StringBuilder.html> . [Consultado: 12-sep-2025].
- [6] Oracle, Class Integer | Java Platform SE 17, Oracle Docs, [En línea]. Disponible: [https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/Integer.html#toHexString\(int\)](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/Integer.html#toHexString(int)) . [Consultado: 12-sep-2025].
- [7] M. Callejón Berenguer, "Cómo funciona el método main() en Java", JavAutodidacta, 7 de junio de 2018. [En línea]. Disponible: <https://javautodidacta.es/metodo-main-en-java/tab-con-10> . [Consultado: 12-sep-2025].