

# UNIDAD TEMÁTICA 3 – Diccionarios, Mapas, Hashing e implementaciones JAVA

## Trabajo de Aplicación 4

### EJERCICIO 1 Implementar el TDA THash en Java

Dadas las siguientes operaciones:

- **public int buscar(int unaClave)** -devuelve la cantidad de comparaciones realizadas-
- **public int insertar(int unaClave)** -devuelve la cantidad de comparaciones realizadas-
- **public int funcionHashing(int unaClave)** -devuelve la posición generada por la función-

Se solicita:

- 1) Analizar en pseudo-código las operaciones solicitadas.
- 2) Desarrollar una función de hash lo más eficiente posible (pseudo-código + implementación).
- 3) Implementar las operaciones de inserción y búsqueda. Es necesario tener en cuenta que se **debe retornar la cantidad de comparaciones realizadas**.
- 4) Preparar el código para insertar y buscar en forma masiva, desde los archivos “**claves\_buscar.txt**” y “**claves\_insertar.txt**”

**Notas:** El método de resolución de colisiones a usar es el direccionamiento abierto lineal con exploración cuadrática:  $h(i) = h(0) + i^2$ , circular. El método principal es el encargado de crear la tabla de hashing, luego de leer las claves del archivo de entrada, estableciendo un tamaño de la tabla igual a **cantClaves/0.9**

#### Dinámica de trabajo para el ejercicio:

Se ha de trabajar en dos sub-equipos, para luego integrar el código y hacer las pruebas correspondientes.

Sub-equipo “A”	Sub-equipo “B”
<b>PASO 1:</b>	
revisa el seudocódigo de la inserción y la búsqueda.	revisa el seudocódigo de una función de hashing lo más eficiente posible..
<b>PASO 2:</b> Los subequipos intercambian los seudocódigos generados y pasan a implementar.	
implementa el método con la función de hashing y lo prueba con un conjunto pequeño de datos.	<ul style="list-style-type: none"><li>• Diseña el programa principal encargado de recorrer el archivo con claves a insertar, insertarlas, recorrer el archivo con claves a buscar, buscarlas, y escribir por consola la cantidad de comparaciones realizadas en promedio -para las búsquedas exitosas e infructuosa</li><li>• Implementa los métodos de inserción y busqueda.</li></ul>
<b>PASO 3:</b> Integrar todo el código y <b>efectuar las pruebas</b> con los archivos provistos, “ <b>claves_buscar.txt</b> ” y “ <b>claves_insertar.txt</b> ”, indicando <b>cantidad de comparaciones</b> tanto al <b>insertar</b> como al <b>buscar</b> (y en este último caso también <b>si la búsqueda tiene o no éxito</b> )	

## EJERCICIO 2

Utilizando como base el programa desarrollado en el Ejercicio 1, se desea hacer una evaluación de la evolución del rendimiento de la tabla hash en las operaciones de inserción y búsqueda, a medida que se va incrementando el factor de carga.

Para ello, se deberá realizar una planilla o tabla que contenga 4 columnas:

1. Factor de carga (empezar en 70%, incrementando en 5% hasta el 90%, y luego incrementos del 1% hasta llegar al 99%)
2. Promedio de comparaciones realizadas para insertar una clave.
3. Promedio de comparaciones realizadas en las búsquedas exitosas
4. Promedio de comparaciones realizadas en las búsquedas sin éxito

<b>Factor de carga %</b>	<b>Prom. Comparaciones Inserción</b>	<b>Prom. Comp. Búsqueda exitosa</b>	<b>Prom. Comp. Búsqueda exitosa</b>
70			
75			
80			
85			
90			
91			
92			
93			
94			
95			
96			
97			
98			
99			

Se sugiere utilizar una planilla electrónica para registrar los resultados.

Graficar los resultados.