**RESUMEN LABORATORIO NO. 5: TABLAS DE HASH**

# Objetivos

Incluir en el diseño de soluciones el uso de Maps o Tablas de símbolos y utilizar tablas de hash para su implementación

1. Familiarizarse con las estrategias de manejo de colisiones en tablas de hash
2. Comparar los tiempos de respuesta de las tablas de hash cuando su factor de carga cambia

# Desarrollo

En los grupos previamente definidos sigan los siguientes pasos para el laboratorio de hoy.

## Cargar el proyecto del Reto No. 2

Asegúrense que todos los miembros del equipo tienen acceso al esqueleto del reto 2 que comenzaron a trabajar en el laboratorio anterior. Verifiquen que todos tienen la misma versión.

## Implementar requerimiento 1 del Reto 2

Deben trabajar en la implementación del requerimiento 1 del reto 2. Para esta implementación, deben seguir utilizando el esquema de desarrollo Modelo-Vista-Controlador.

La implementación de la solución la deben hacer mediante el uso del TAD Map. Revisen cuál es el número de elementos esperados a guardar en el Map, esto dependerá del número de elementos que se encuentran en los archivos con los datos del proyecto. Para este laboratorio pruebe con archivos con la mayor cantidad de información disponible.

### Utilizar Separate Chaining

Utilicen inicialmente una tabla de hash basada en *Separate Chaining* como mecanismo de manejo de colisiones.

### Utilizar Linear Probing

Ahora, cambien la implementación de la tabla de hash por una que utilice *Linear Probing.*

**Pregunta 1:** Notan alguna diferencia en tiempo de carga y de consulta entre las dos implementaciones? ¿Si es así cuál es más rápida?

El linear probing es más rápido en la carga y consulta de archivos. Esto se debe a que al cargar los archivos las direcciones en memoria están mas próximas que si se utilizara el chain probing,

### Cambiar los factores de carga

Ahora prueben nuevamente con una tabla de hash basada en ***Separate Chaining*** pero cambiando el factor de carga de la tabla. En una primera prueba utilicen un factor de carga de 2. En una segunda prueba utilicen un factor de carga de 0.4, finalmente en una tercera prueba utilicen un factor de carga de 10, probando siempre con el mismo archivo de pruebas (mismo número de elementos a guardar en la tabla).

**Pregunta 2**. Nota alguna diferencia en los tiempos de carga y/o de respuesta cuando el factor de carga cambia utilizando *Separate Chaining?* ¿Describa las diferencias encontradas?

A medida que se aumenta el factor de carga el tiempo de respuesta es mayor. En este caso se esta haciendo un compromiso entre la cantidad de memoria utilizada y la velocidad de lectura. Una capacidad de carga menor implicaría tiempos menores, pero requiere más capacidad en memoria pues se esta aumentado el tamaño del hashmap en intervalos mas pequeños.

**Pregunta 3.** Nota alguna diferencia en los tiempos de carga y/o de respuesta cuando utiliza *linear probing* (factor de carga 0.5) y cuando el factor de carga es 10 en *separate chaining?* Describa las diferencias.

En este caso el linear probing tiene tiempos de consulta más rápidos que el separate chaining. Esto se debe a los factores descritos en las dos preguntas anteriores: con un factor de carga igual el linear probing seria mas rápido. Si a esto se le suma que el chain probing va a tener un factor de carga considerablemente mayor su tiempo de carga y ejecución va a ser más lento.

## Compartir el producto de la practica con los evaluadores

El resultado de este laboratorio es la implementación del requerimiento 1 del reto 2. Para entregar exitosamente sus resultados de este laboratorio, por favor recuerde las siguientes indicaciones:

1. Invitar al profesor y monitores del laboratorio asignados.
2. Incluir en el **README** del repositorio los datos completos de los integrantes del grupo (nombre completo, correo Uniandes y código de estudiante).
3. Incluir en la carpeta *Docs* un documento en formato PDF que indique lo siguiente:
   1. Datos completos de los integrantes del grupo (nombre completo, correo Uniandes y código de estudiante).
   2. Las tres preguntas del laboratorio con sus respectivas respuestas, en un documento PDF llamado respuestas-lab5.pdf, marcado con el nombre de los integrantes del grupo. Dentro del documento la respuesta a las preguntas:

**Pregunta 1 (Paso 2.2.2):** Notan alguna diferencia en el tiempo de carga (creación de la tabla de hash e inserción de datos) y/o el tiempo de consulta entre las dos implementaciones? ¿Si es así cuál es más rápida?

La implementación de la tabla de hash es más rápida si se utiliza un arreglo tipo arraylist. Por el momento no fue posible notar una diferencia en los tiempos de carga entre las dos implementaciones

**Pregunta 2 (Paso 2.2.3)**. Nota alguna diferencia en el tiempo de carga (creación de la tabla de hash e inserción de datos) y/o el tiempo de respuesta cuando el factor de carga cambia utilizando *Separate Chaining?* ¿Describa las diferencias encontradas?

A medida que se aumenta el factor de carga el tiempo de respuesta es mayor. En este caso se esta haciendo un compromiso entre la cantidad de memoria utilizada y la velocidad de lectura. Una capacidad de carga menor implicaría tiempos menores, pero requiere más capacidad en memoria pues se esta aumentado el tamaño del hashmap en intervalos mas pequeños.

**Pregunta 3 (Paso 2.2.3).** Nota alguna diferencia en el tiempo de carga (creación de la tabla de hash e inserción de datos) y/o en el tiempo de respuesta cuando utiliza *linear probing* (factor de carga 0.5) y cuando el factor de carga es 10 en *separate chaining?* Describa las diferencias.

El linear probing tiene tiempos de consulta más rápidos que el separate chaining. Esto se debe a los factores descritos en las dos preguntas anteriores: con un factor de carga igual el linear probing seria mas rápido. Si a esto se le suma que el chain probing va a tener un factor de carga considerablemente mayor su tiempo de carga y ejecución va a ser más lento.

1. Generar una versión llamada “Entrega Final – laboratorio 5” *(git commit -m “Entrega Final – laboratorio 5”)* en el depósito de GitHub antes de la media noche (11:59 pm) del miércoles 16 de septiembre.
2. Entregar el enlace (URL) del repositorio por Sicuaplus antes de la media noche (11:59 pm) del miércoles 16 de septiembre.

Recuerden que cualquier documento solicitado durante las actividades debe incluirse en el repositorio GIT y que solo se calificará hasta el último **COMMIT** realizado dentro de las fechas límites