
1. TALLER HASH MAP

Un Hash Map es una implementación de la estructura de datos *Mapa* (o *diccionario*) en la cual se aprovecha la estrategia del Hashing combinada con el Direcccionamiento Cerrado (Closed Adressing) para evitar la colisión de datos y tener que realizar un desplazamiento de los mismos a través de la manipulación de un arreglo.

La forma en la que lo realiza es cambiando la estructura del dato que se está almacenando para que pueda guardar referencia de otro de su misma clase. De esta manera la implementación se asemejaría a un arreglo de listas enlazadas, pero sin tener estrictamente una estructura *Lista*, pues todo lo manejarían los objetos almacenados.

1.1 Reto

Usted deberá de crear una clase **User** la cual cuente con los atributos *username*, *email*, *password* y *country*.

Esta clase entonces será almacenada en una estructura de datos HashMap bajo el nombre de **Network** (usando apuntadores a los objetos). Esta clase debe tener las funciones básicas del hashing (constructor, destructor, hash, redimensionamiento y display), *addUser()*, para introducir los datos dados en el archivo *datos_prueba.csv* en forma de un objeto **User** (para esto también hay una función *addUsersFromCSV()*), la función *deleteUser()* para eliminar un dato, la función *find()* para encontrar los datos del usuario, la función *changeData()* para cambiar la cualquiera de los cuatro atributos y finalmente la función *login()* para iniciar sesión en la red.

Para verificar cuál es el mejor tamaño de arreglo para almacenar los datos, ejecútelo con diferentes valores y realice las búsquedas necesarias. También pruebe con una función Hash diferente a la vista en clase y compruebe su eficiencia. Para esto se le entregará un *main()* que podrá cambiar a su antojo.

RECUERDE QUE NO PUEDEN HABER NOMBRES DE USUARIOS REPETIDOS.

1.2 Elaboración con IA

Este ejercicio no ha sido diseñado para que lo realice específicamente por su cuenta. Se espera que usted haga preguntas a la IA de su preferencia y tener un acercamiento controlado a la programación apoyada por herramientas de Inteligencia Artificial. Sin embargo tampoco debe de solicitar todo el código, por lo que deberá de hacer preguntas (o prompts) puntuales para resolver sus dudas. Y deberá de tener un seguimiento en una tabla en formato WORD de:

- El prompt realizado.
- Link a la respuesta o en su defecto la respuesta en sí.
- Mejoría respecto a su código o su diseño.

- Observaciones y posibilidad de mejora de la explicación de la IA.

Esos campos deberán de llenarse cada vez que realice un prompt (mínimo dos), y tendrá peso significativo en la nota.

1.3 Modo de entrega

El código deberá ser entregado en un archivo llamado 'Apellido_Nombre_Network.h', así como el archivo en word llamado 'Apellido_Nombre_Prompts.docx'.

NO ENVIAR EL MAIN. Cualquier archivo que no cumpla con este formato de nombre NO SERÁ CALIFICADO.