

# OBSERVACIONES DEL LA PRACTICA

Adriana Sofia Rozo Cepeda - Cod 202211498

Diego Fernando Galván Cruz - Cod 202213709

Nicolás Bedoya Figueroa - Cod 202212100

## a. ¿Qué estructura de datos se usa para este índice?

La estructura implementada es una tabla de hash. En los dos casos es separate chaining y la otra es probing.

## b. ¿Cuántos elementos se espera almacenar inicialmente?

Se espera que guarde 800 elementos.

## c. ¿Cuál es el tamaño de las tablas de hash para 'years' y 'authors'?

Para years va a ser de 40 y para authors va a ser de 800.

## d. ¿Cuál es el factor de carga máximo?

Para authors el factor de carga es de 4.0 y para years es de 0.5.

## e. ) ¿Qué hace la instrucción "mp.put(...)?"

La función mp.put toma como parametros 3 valores, el mapa, la llave y el valor de la llave. Entonces, a dicho mapa le va agregar una nueva llave con su respectivo valor. En el caso del ejemplo que nos muestran mp.put toma el mapa catalog["bookIds"] y le agraga una nueva llave que serán los ids de los libros book['goodreads\_book\_id'] y adicionalmente a estos indices o ids guarda la referencia al libro con book.

## f. ¿Qué papel cumple "book['goodreads\_book\_id']" en esa instrucción?

Es la llave que contiene la llave:valor, en la tabla de hash y contiene el Id del libro del archivo.

## g. ¿Qué papel cumple el tercer parámetro "book" en esa instrucción?

Este es un valor asignado a la pareja llave:valor en la tabla de hash y en este encontramos la informacion completa del libro seleccionado.

## h. ¿Qué hace la instrucción "mp.get(...)?"

La función `mp.get(...)` retorna la pareja llave, valor, cuya llave sea igual a `key`. El map donde se guarda la pareja sería en este caso: `catalog['years']`, y `year` sería la llave asociada a la pareja.

**i. ¿Qué papel cumple “year” en esa instrucción?**

El papel de `Year` es que es la llave asociada a la pareja de llave:valor.

**j. ¿Qué hace la instrucción titulada “me.getValue(...)”?**

La función `me.getValue(...)` retorna el valor de una pareja de un Map. Tomando como parámetro la pareja llave valor, en este caso sería: `(year)['books']`

**k) ¿Cuál es la llave y el valor del nuevo índice implementado?, describa el tipo de dato y/o las estructuras de datos involucradas.**

La nueva llave sería `title1` y el valor es el `title` y se implementó un mapa 'PROBING'

```
271
272 def addBookTitle(catalog, title):
273     # TODO Lab 6, agregar el libro al map de titulos
274     """
275     Completar la descripcion de addBookTitle
276     """
277     titles = catalog['titles']
278     existtitle = mp.contains(titles, title)
279     if existtitle:
280         entry = mp.get(titles, title)
281         title1 = me.getValue(entry)
282     else:
283         mp.put(catalog['titles'], title1, title)
284
```

```
1
2
3 # TODO Lab 6, agregar el ADT map con newMap()
4 catalog['titles'] = mp.newMap(150,
5                               maptype='PROBING',
6                               loadfactor=0.5,
7                               comparefunction=compareTitles)
8
9 return catalog
10
```

**l) ¿Qué pasa cuando existan dos o más libros con el mismo título (ej.: diferente edición)?**

Reemplaza el libro anterior del mapa por el nuevo libro ingresado en la entrada propuesta.

**m) Con el índice implementado ¿Cómo solucionaría que dos o más libros tengan el mismo título?**

Según lo entendido a partir de la pregunta anterior, la mejor forma de lograr esta implementación es otorgando una nueva llave al ingresado por parámetro, para que no se actualicen los datos ya propuestos en la tabla, y así, se pueda hacer una reorganización y encontrarlo como un nuevo valor.