

OBSERVACIONES DEL LA PRACTICA

Est1 = Samuel Josue Freire Tarazona, 202111460, s.freire@uniandes.edu.co

Est2 = Jose David Martinez Oliveros, 202116677, jd.martinezo1@uniandes.edu.co

- 1) ¿Qué estructura de datos se usa para este índice?

La estructura de datos, que se usa en este índice, es una tabla de hash. Esta asocia llaves con valores, para poder hacer su identificación más rápido.

```
mp.newMap()
```

- 2) ¿Cuántos elementos se espera almacenar inicialmente?

Según, la función de creación del mapa, su cantidad esperada o lo que se llama elementos que espera cargar. Se encontró que son maso menos 800 autores en este caso.

```
(800,
```

- 3) ¿Cuál es el factor de carga máximo?

Nuevamente, en la función donde se crea la estructura se puede denotar que sus componentes o parámetros a ingresar se esperan un número. En este caso, se encuentra el factor de carga máximo como 4.0.

```
loadfactor=4.0,
```

- 4) ¿Qué hace la instrucción “mp.put(...)”?

Esta función según la documentación, esta función, en primer lugar recibe tres parámetros. En primer lugar, recibe el mapa donde se van a poner o guardar los valores que se quieren guardar. En segundo lugar, recibe la llave donde se van a anexar los valores o lo que se quiera guardar. En tercer lugar, recibe el valor, el cual se va a guardar en la llave. Y al final, lo que hace es meter o poner en el mapa la llave con el valor relacionado a esta.

```
def put(map, key, value):
```

- 5) ¿Qué papel cumple “book[‘goodreads_book_id’]” en esa instrucción?

Como ya se dijo anteriormente, este valor sirve como llave del mapa. A la cual se le va a indexar o relacionar el valor. Esta llave, sirve como identificador, para poder encontrar el valor más rápido.

```
def put(map, key, value):
```

- 6) ¿Qué papel cumple **“book”** en esa instrucción?

Como ya se dijo anteriormente, este valor sirve como “valor” de la llave del mapa. Este valor se va aguardar o se va relacionar en la llave. Esto permite mayor rapidez a la hora de solicitar este número o dato que se está almacenando.

```
def put(map, key, value):
```

- 7) ¿Qué hace la instrucción **“mp.get(...)”**?

Esta función ayuda a retornar el valor de una llave o que está guardado en una llave, en el mapa especificado. Esta permite buscar rápidamente, en el mapa, el valor que se requiere. Esta función busca dentro del mapa la llave, y devuelve el valor relacionado a esta. Dependiendo de la implementación del mapa, se busca de una manera u otra.

```
def get(map, key):  
    """ Retorna la """
```

- 8) ¿Qué papel cumple **“year”** en esa instrucción?

En este caso “year” es la llave a buscar en el mapa. Como se ve en la implementación la función va retornar el valor que se iguala a “year” en este caso.

```
def get(map, key):  
    """ Retorna la """
```

- 9) ¿Qué hace la instrucción **“me.getValue(...)”**?

En este caso, la función getvalue como su nombre lo indica, retorna una llave a partir de su valor. Es lo opuesto a la función getvalue, ya que busca el valor y retorna la llave.

```
def getValue(entry):  
    """ """
```