

OBSERVACIONES DEL LA PRACTICA

Camilo Ortiz 201821615
Kevin Fernando Gomez Cod 202015120

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

Según la definición del índice “authors” que se muestra a continuación:

```
"""
Este indice crea un map cuya llave es el autor del libro
"""
catalog['authors'] = mp.newMap(800,
                              matype='CHAINING',
                              loadfactor=4.0,
                              comparefunction=compareAuthorsByName)
```

Se observa que la estructura de datos utilizada para sostener internamente la Tabla de Hash es encadenamiento separado, lo cual se observa al especificar `matype='CHAINING'`.

2) ¿Cuántos elementos se espera almacenar inicialmente?

Según la definición del índice “authors”, se indicó en el primer argumento de la función `newMap` (numelements) el tamaño inicial de la Tabla de Hash como 800 elementos que en este caso son 800 autores.

```
"""
Este indice crea un map cuya llave es el autor del libro
"""
catalog['authors'] = mp.newMap(800,
                              matype='CHAINING',
                              loadfactor=4.0,
                              comparefunction=compareAuthorsByName)
```

3) ¿Cuál es el factor de carga máximo?

Según la definición del índice “authors”, el argumento `loadfactor` se especificó como 4.0, es decir, que la Tabla de Hash va a manejar un factor de carga máximo de 4.0.

```
"""
Este indice crea un map cuya llave es el autor del libro
"""
catalog['authors'] = mp.newMap(800,
                              matype='CHAINING',
                              loadfactor=4.0,
                              comparefunction=compareAuthorsByName)
```

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

```
mp.put(catalog['bookIds'], book['goodreads_book_id'], book)
```

La instrucción “mp.put(...)” agrega el libro a la Tabla de Hash del índice “bookIds” usando el id del mismo libro como la llave y todo el libro como el valor. Entonces la pareja llave, valor sería id del libro, libro. En el caso de que el id del libro ya exista como llave en la Tabla de Hash, se reemplaza el libro ya que no es una colisión en sí; esto no debería ocurrir porque el id del libro (‘goodreads_book_id’) es único.

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

El papel que cumple “book[‘goodreads_book_id’]” en la instrucción mp.put(...) es obtener el dato específico del libro que se quiere usar como la llave en la Tabla de Hash del índice “bookIds”.

```
mp.put(catalog['bookIds'], book['goodreads_book_id'], book)
```

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

La parte de “book” al final de la instrucción mp.put(...) tiene el objetivo de pasar como argumento el valor que se quiere relacionar a la llave obtenida en “book[‘goodreads_book_id’]” en la Tabla de Hash del índice “bookIds”.

```
mp.put(catalog['bookIds'], book['goodreads_book_id'], book)
```

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

```
def getBooksByYear(catalog, year):  
    """  
    Retorna los libros publicados en un año  
    """  
    year = mp.get(catalog['years'], year)  
    if year:  
        return me.getValue(year)['books']  
    return None
```

En la función getBooksByYear la instrucción “mp.get(...)” retorna la pareja llave, valor cuya llave, que debería ser única, corresponde con el segundo argumento dado a la instrucción, es decir, con “year” (el año en el que se quiere conocer los libros publicados).

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

El segundo parámetro “year” en la instrucción “mp.get(...)” tiene el papel de proporcionar la llave de la cual queremos obtener la pareja llave, valor almacenada en la Tabla de Hash del índice “years”.

```
def getBooksByYear(catalog, year):
    """
    Retorna los libros publicados en un año
    """
    year = mp.get(catalog['years'], year)
    if year:
        return me.getValue(year)['books']
    return None
```

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

La instrucción “**me.getValue(...)**” retorna el valor de la pareja llave, valor que se consigue de la función “**mp.get(...)**”. En el caso del fragmento se retornan los “books” del valor de la pareja “year” pasada a la función “**getValue(...)**”, es decir, los libros que fueron publicados ese año. Lo anterior solo sucede si en la Tabla de Hash del índice “years” se encuentra la llave.

```
def getBooksByYear(catalog, year):
    """
    Retorna los libros publicados en un año
    """
    year = mp.get(catalog['years'], year)
    if year:
        return me.getValue(year)['books']
    return None
```