

OBSERVACIONES DEL LA PRACTICA

Estudiante 1 María Alejandra Moreno Bustillo Cod. 202021603

Estudiante 2 Juliana Delgadillo Cheyne Cod.202020986

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

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

R// La estructura de datos que se está utilizando para este índice es 'Chaining', también conocida como separate chain. Esto se puede ver en la línea de código 90.

- 2) ¿Cuántos elementos se espera almacenar inicialmente?

R// Inicialmente, se esperan guardar 800 elementos, lo cual se ve reflejado en la línea de código 89.

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

R//El factor de carga máximo de este map es de 4.0 y este es un factor de carga bastante alto. La línea de código en la que este se puede evidenciar es en la 91.

```
def addBook(catalog, book):
    """
    Esta función adiciona un libro a la lista de libros,
    adicionalmente lo guarda en un Map usando como llave su Id.
    Adicionalmente se guarda en el índice de autores, una referencia
    al libro.
    Finalmente crea una entrada en el Map de años, para indicar que este
    libro fue publicaco en ese año.
    """
    lt.addLast(catalog['books'], book)
    mp.put(catalog['bookIds'], book['goodreads_book_id'], book)
    authors = book['authors'].split(",") # Se obtienen los autores
    for author in authors:
        addBookAuthor(catalog, author.strip(), book)
    addBookYear(catalog, book)
```

- 4) ¿Qué hace la instrucción "**mp.put(...)**"?

R// Es una función que recibe los siguientes tres parámetros: map (el map donde se quiere almacenar la pareja llave-valor), luego Key (llave asociada a la pareja que se busca agregar) y Value (que es el valor asociado a esa pareja) y retorna el mapa con esa pareja llave-valor. El objetivo de esta función es ingresar una pareja llave-valor a la tabla de hash, en el caso de que esa llave ya exista, se reemplaza el valor antiguo por el ingresado.

- 5) ¿Qué papel cumple "**book['goodreads_book_id']**" en esa instrucción?

R//El papel que cumple "book['goodreads_book_id']" es de ser la llave asociada a la pareja que se busca adicionar a la tabla de hash.

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

R//El papel que cumple "book" es de ser el valor asociado a la pareja llave-valor que se busca adicionar a la tabla de hash.

```
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
```

- 7) ¿Qué hace la instrucción “**mp.get(...)**”?
R// La instrucción mp.get() se encarga de retornar una pareja llave-valor si y solo si la llave es igual a la que se le pasó como parámetro en la segunda posición.
- 8) ¿Qué papel cumple “**year**” en esa instrucción?
R// Year en esa instrucción es la llave que se pretende buscar en el map catalog['years'].
- 9) ¿Qué hace la instrucción “**me.getValue(...)**”?
R// La instrucción mp.getValue() le entra como parámetro una pareja llave, valor y su objetivo es retornar el valor de dicha pareja.