

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

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

La estructura de datos usada para crear el índice es una tabla de símbolos no ordenada.

b) ¿Cuántos elementos se espera almacenar inicialmente?

De acuerdo con el primer parámetro (numelements), para este índice se esperan almacenar inicialmente 800 elementos (en este caso, autores).

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

```
"""
Este índice crea un map cuya llave es el año de publicacion
"""
catalog['years'] = mp.newMap(40,
                             maptype='PROBING',
                             loadfactor=0.5,
                             cmpfunction=compareMapYear)

"""
Este índice crea un map cuya llave es el autor del libro
"""
catalog['authors'] = mp.newMap(800,
                              maptype='CHAINING',
                              loadfactor=4,
                              cmpfunction=compareAuthorsByName)
```

Para 'years' el tamaño de la tabla es de 40 elementos, mientras que para 'authors' es de 800 elementos.

d) ¿Cuál es el factor de carga máximo?

En el ejercicio, el factor de carga máximo es 4. Este factor está en los índices de 'bookIds' y 'authors'.

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

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

Segmento 6. Instrucción mp.put(...) en la función addBook.

"mp.put(...)" es una función que a partir de se encarga de introducir una pareja llave-valor, dadas por parámetro, a la tabla una tabla de hash. Si la llave ya existe en la tabla, se reescribe su valor. En este caso en específico, "mp.put(...)" recibe como llave book['goodreads_book_id'], y su valor correspondiente 'book', y esta pareja llave-valor la almacena en la tabla de hash catalog['bookIds'].

f) ¿Qué papel cumple "book['goodreads_book_id']" en esa instrucción?

"book['goodreads_book_id']" es el nombre de la llave a la que se busca asociar un libro; la llave de este libro es su id.

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

"book" corresponde al valor de la llave, en este caso un libro en particular, que se quiere introducir a la tabla de hash.

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

La instrucción "mp.get(...)" devuelve el valor asociado con una clave dada en la tabla hash.

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

"year" es la clave que se utiliza para buscar el valor correspondiente en la tabla hash.

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

La instrucción titulada "me.getValue(...)" devuelve el valor asociado con una clave dada en la tabla hash.

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

```
def addBookTitle(catalog, title):
    # TODO lab 6, agregar el libro al map de titulos
    """
    Completar la descripción de addBookTitle
    """
    lista_libros_completa = catalog['books']
    dicc_titulos = mp.newMap()

    for libro in lt.iterator(lista_libros_completa):
        titulo = libro["title"]

        if titulo not in dicc_titulos:
            mp.put(dicc_titulos, titulo, libro)

    catalog["Titles"] = dicc_titulos
```

El nuevo índice implementado es 'Titles'. La llave 'Titles' tiene asociado como valor un diccionario, que contiene como llaves los títulos de los libros, y como valor otro diccionario en donde se ha guardado la información respectiva a estos, como el año de publicación, el autor, el isbn y demás. Las estructuras de datos involucradas en este caso son solamente diccionarios, aunque se planteó inicialmente implementar listas también. De esta manera las llaves del diccionario de libros correspondería al título

de un libro, pero su valor sería una lista de diccionarios de la información de aquellos libros con el mismo título (es decir, esa misma llave).

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

Como se plantea la solución, si existiesen dos o más libros con exactamente el mismo título, sólo se retornaría el primero de ellos que fue procesado en la carga de datos.

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

Para dos libros que tengan exactamente el mismo título (título completo), se propone crear un diccionario que tenga como llave el título completo del libro y como valor una lista de diccionarios con la información correspondiente a los diferentes libros. Ahora, si dos libros tienen el mismo título original (que puede ser igual en el caso de una saga de libros por ejemplo), se propone crear otro diccionario (a parte del anterior) cuyas llaves correspondan al título original del libro, y como valor asociado tendrían una lista de diccionarios de la información de cada libro. En ambos casos, la información de libros con títulos iguales quedaría almacenada en una lista.