

Ahora considere la declaración de la llave “authors”:

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

Considere las siguientes preguntas:

- a) ¿Qué estructura de datos se usa para este índice?
 - b) ¿Cuántos elementos se espera almacenar inicialmente?
 - c) ¿Cuál es el factor de carga máximo?
 - d) ¿Con cuántos elementos serán necesarios agregar para hacer re-hash de la tabla?
- A) Se usa un Map de tipo ‘CHAINING’-
- B) Se esperan almacenar 800 elementos en el MAP
- C) Es factor de carga máximo es 4.
- D) Como el factor de carga es 4, eso significa que cuando se carguen mas de 200 elementos se hará re-hash a la tabla.

```
def addBook(catalog, book):
    """
    Esta funcion adiciona un libro a la lista de libros,
    adicionalmente lo guarda en un Map usando como llave su Id.
    Adicionalmente se guarda en el indice de autores, una referencia
    al libro.
    Finalmente crea una entrada en el Map de años, para indicar que este
    libro fue publicado 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)
```

considere la instrucción “mp.put(...)” como se ve individualmente en la siguiente sección de código.

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

Responda las siguientes preguntas:

- a) ¿Qué hace la instrucción “mp.put(...)”?
 - b) ¿Qué papel cumple “book['goodreads_book_id']” en esa instrucción?
 - c) ¿Qué papel cumple el tercer parámetro “book” en esa instrucción?
- a) Agregar una pareja llave, valor a la tabla de hash.
- b) Es la llave que se va a guardar en la tabla de hash
- c) Es el valor que tendrá la llave a guardar en la tabla de hash.

Considere ahora el siguiente fragmento de código:

```
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
```

Teniendo en cuenta este fragmento responda las siguientes preguntas:

- a) ¿Qué hace la instrucción “mp.get(...)”?
 - b) ¿Qué papel cumple “year” en esa instrucción?
 - c) ¿Qué hace la instrucción “me.getValue(...)”?
-
- a) Retorna la pareja llave, valor, donde la llave sea igual al parámetro “year”.
 - b) Saber si “year” encontró, o tiene algo diferente a “None” la llave en la tabla de hash.
 - c) Retorna los libros publicados en el año que entró por parámetro.