

Observaciones laboratorio 3

PASO 3: ESTUDIAR EL EJEMPLO EN VS CODE

- ¿Cuáles son los mecanismos de interacción (I/O: Input/Output) que tiene el view.py con el usuario?

El mecanismo usado es la función “printMenu()”, es donde se generan los outputs como opciones, con la ayuda de un “while”, se generan los outputs que salen como respuesta de los inputs recibidos por la función.

```
37
38 def printMenu():
39     print("Bienvenido")
40     print("1- Cargar información en el catálogo")
41     print("2- Consultar los Top x libros por promedio")
42     print("3- Consultar los libros de un autor")
43     print("4- Libros por género")
44     print("0- Salir")
45
87 while True:
88     printMenu()
89     inputs = input('Seleccione una opción para continuar\n')
90     if int(inputs[0]) == 1:
91         print("Cargando información de los archivos ....")
92         catalog = initCatalog()
93         loadData(catalog)
94         print('Libros cargados: ' + str(len(catalog['books'])))
95         print('Autores cargados: ' + str(len(catalog['authors'])))
96         print('Géneros cargados: ' + str(len(catalog['tags'])))
97         print('Asociación de Géneros a Libros cargados: ' +
98               str(len(catalog['book_tags'])))
99
100     elif int(inputs[0]) == 2:
101         number = input("Buscando los TOP ?: ")
102         books = controller.getBestBooks(catalog, int(number))
103         printBestBooks(books)
104
105     elif int(inputs[0]) == 3:
106         authorname = input("Nombre del autor a buscar: ")
107         author = controller.getBooksByAuthor(catalog, authorname)
108         printAuthorData(author)
109
110     elif int(inputs[0]) == 4:
111         label = input("Etiqueta a buscar: ")
112         book_count = controller.countBooksByTag(catalog, label)
113         print('Se encontraron: ', book_count, ' Libros')
114
115     else:
116         sys.exit(0)
```

Además del menú se tiene las funciones que definen la búsqueda de información y outputs, tal como el siguiente fragmento de código:

```
def printAuthorData(author):
    if author:
        print('Autor encontrado: ' + author['name'])
        print('Promedio: ' + str(author['average_rating']))
        print('Total de libros: ' + str(len(author['books'])))
        for book in author['books']:
            print('Titulo: ' + book['title'] + ' ISBN: ' + book['isbn'])
    else:
        print('No se encontro el autor')
```

- ¿Cómo se almacenan los datos de GoodReads en el model.py?
- ¿Cuáles son las funciones que comunican el view.py y el model.py?

En estas capturas se evidencia como las dos funciones presentes en view.py llaman a controller, siendo estas mismas funciones en controller (imagen 2) las que llaman a model.py, dando así una comunicación entre model y view mediante estas dos funciones.

```
47 def initCatalog():
48     """
49     Inicializa el catalogo de libros
50     """
51     return controller.initCatalog()
52
53
54 def loadData(catalog):
55     """
56     Carga los libros en la estructura de datos
57     """
58     controller.loadData(catalog)
```

Imagen 1 (funciones en view)

```
35 def initCatalog():
36     """
37     Llama la funcion de inicializacion del catalogo del modelo
38     """
39     catalog = model.newCatalog()
40     return catalog
41
42
43 # Funciones para la carga de datos
44
45
46 def loadData(catalog):
47     """
48     Carga los datos de los archivos y cargar los datos en la
49     estructura de datos
50     """
51     loadBooks(catalog)
52     loadTags(catalog)
53     loadBooksTags(catalog)
54     sortBooks(catalog)
55
```

Imagen 2(funciones en controller)

PASO 4: ESTUDIAR EL USO DE LISTAS

- ¿Cómo se crea una lista?

Lo primero que se hace es definir las condiciones que debe cumplir la lista, estas condiciones se dan como parámetro a la función “newList”, acá se dice que tipo de lista será, si tendrá alguna función de comparación, algún identificador, el tipo de archivo y el delimitador que usa el archivo.

```
def newList(datastructure='SINGLE_LINKED',
            cmpfunction=None,
            key=None,
            filename=None,
            delimiter=","):
    """Crea una lista vacia
```

Luego se envían estos parámetros a la estructura de dato correspondiente creando y retornando la lista solicitada.

```
    """
    try:
        lst = lt.newList(datastructure, cmpfunction, key, filename, delimiter)
        return lst
    except Exception as exp:
        error.reraise(exp, 'TADList->newList: ') Find related code in ISIS1
```

- ¿Qué hace el parámetro cmpfunction=None en la función newList()?

Este parámetro indica que no se usara ninguna función de comparación al momento de crear la función en la estructura de dato.

- ¿Qué hace la función addLast()?

Esta función tiene como parámetro la lista y el elemento a agregar a esta lista, este elemento se agrega en la última posición de la lista, luego se cambia el apuntador para que indique que el elemento agregado es el último de la lista.

- ¿Qué hace la función getElement()?

Esta función tiene como parámetros la lista y la posición de elemento dentro de esta lista, lo que hace la función es buscar esa posición y retornar el elemento que se encuentre allí.

- ¿Qué hace la función subList()?

Esta función tiene como parámetros la lista, una posición a partir de la cual se obtendrán los elementos y la cantidad de elementos que se debe obtener. Luego envía la orden a la estructura de dato para que cree la lista con las condiciones dadas, luego de esto retorna la lista.

PASO 5: CAMBIOS EN EL USO DE TAD LISTA (ADC list)

- ¿Observó algún cambio en el comportamiento del programa al cambiar la implementación del parámetro "ARRAY_LIST" a "SINGLE_LINKED"?

Tratando de contabilizar los tiempos el cambio no es mucho, ya que los tiempos de espera son muy altos.