OBSERVACIONES DEL LA PRACTICA

Estudiante 1:Juan Felipe García 202014961 jf.garciam1

Estudiante 2: Santiago Rodríguez 202020476 s.rodriguez64

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

El primer mecanismo es el menú, teniendo 5 opciones, cada una de estas activa una función dentro del programa. Dependiendo del input que haga el usuario se va a ejecutar un resultado distinto, las funciones que se tienen son: 1.Cargar información del catalogo, 2.Busca el top de libros dependiendo de los que quiera incluir el usuario, 3. Busca los libros de un autor especifico que el usuario elija, 4. Busca libros con un genero dependiendo del usuario, 0. Salir del programa. Los outputs dependen la opción escogida respondiendo lo que se pregunto con un mensaje y la respuesta.

while True:

    printMenu()

    inputs = input('Seleccione una opción para continuar\n')

    if int(inputs[0]) == 1:

        print("Cargando información de los archivos ....")

        catalog = initCatalog()

        loadData(catalog)

        print('Libros cargados: ' + str(lt.size(catalog['books'])))

        print('Autores cargados: ' + str(lt.size(catalog['authors'])))

        print('Géneros cargados: ' + str(lt.size(catalog['tags'])))

        print('Asociación de Géneros a Libros cargados: ' +

              str(lt.size(catalog['book\_tags'])))

    elif int(inputs[0]) == 2:

        number = input("Buscando los TOP ?: ")

        books = controller.getBestBooks(catalog, int(number))

        printBestBooks(books)

    elif int(inputs[0]) == 3:

        authorname = input("Nombre del autor a buscar: ")

        author = controller.getBooksByAuthor(catalog, authorname)

        printAuthorData(author)

    elif int(inputs[0]) == 4:

        label = input("Etiqueta a buscar: ")

        book\_count = controller.countBooksByTag(catalog, label)

        print('Se encontraron: ', book\_count, ' Libros')

    else:

        sys.exit(0)

sys.exit(0)

1. ¿Cómo se almacenan los datos de **GoodReads** en el **model.py**?

Los datos de GoodReads se almacenan en el model a través de la creación de listas que se guardan en un catálogo. La función de catálogos crea listas vacías (se especifica si es array o single linked) para los libros, autores, tags y booktags. A través de estas listas se permite guardar nueva información de los documentos csv. de GoodReads. También existen funciones para agregar información a las listas del catálogo, crear datos, consultarlos y comparar elementos en el model.py.

def newCatalog():

    """

    Inicializa el catálogo de libros. Crea una lista vacia para guardar

    todos los libros, adicionalmente, crea una lista vacia para los autores,

    una lista vacia para los generos y una lista vacia para la asociación

    generos y libros. Retorna el catalogo inicializado.

 """

    catalog = {'books': None,

               'authors': None,

               'tags': None,

               'book\_tags': None}

    catalog['books'] = lt.newList()

    catalog['authors'] = lt.newList('ARRAY\_LIST',

                                    cmpfunction=compareauthors)

    catalog['tags'] = lt.newList('SINGLE\_LINKED',

                                 cmpfunction=comparetagnames)

   catalog['book\_tags'] = lt.newList('SINGLE\_LINKED')

    return catalog

1. ¿Cuáles son las funciones que comunican el el **view.py** y el **model.py**?
2. ¿Cómo se crea una lista?
3. ¿Qué hace el parámetro **cmpfunction=None** en la función **newList()**?
4. ¿Qué hace la funció **addLast()**?
5. ¿Qué hace la función **getElement()**?
6. ¿Qué hace la función **subList()**?
7. ¿Observó algún cambio en el comportamiento del programa al cambiar la implementación del parámetro **“ARRAY\_LIST”** a **“SINGLE\_LINKED”**?