



Trabajo Práctico 2

Pilas y colas

Estructuras de datos
Comisión A (Turno Mañana)
Comisión B (Turno Noche)
Primer cuatrimestre de 2021

Docentes	Sergio Gonzalez Ariel Clocchiatti
Email	sergio.gonzalez@unahur.edu.ar aclocchi@gmail.com
Fecha de entrega	30 / 05 / 2020

Índice

1. Introducción	2
1.1. Presentación del problema	2
2. Objetivos	3
2.1. Estanterías	3
2.2. Escritorio de atención	3
3. Datos de prueba	4
4. Entrega	5

1. Introducción

En este trabajo práctico vamos a integrar los temas de la primer parte de la materia, centrándonos en las estructuras de datos Pila (*Stack*) y Cola (*Queue*). Van a tener que utilizar los conocimientos que fueron adquiriendo hasta ahora, para resolver un problema e implementar esta solución en el lenguaje de programación *Python*.

No se pide modelar este ejercicio usando objetos. No se compliquen buscando herencias y clases abstractas. No está mal si necesitan poner un “*if*” en algunos métodos donde normalmente tratarían de usar herencia y polimorfismo.

1.1. Presentación del problema

En una biblioteca deciden informatizar el manejo de la salida y entrada de libros mediante el sistema de préstamo. Para esto contratan a una estudiante de la UNAHUR para que modele el problema y diseñe e implemente una solución que le permita automatizar el proceso.

Durante este trabajo van a implementar tipos de datos y operaciones para trabajar con los préstamos de libros a los usuarios. Para resolver el problema de organización de la biblioteca, vamos a modelar los **libros**, las **estanterías** y finalmente los **escritorios de atención** de la biblioteca. Cada escritorio de atención tiene una depósito con varias estanterías con libros.

Para el modelado de los libros, se deben tener en cuenta estas condiciones:

- Cada libro se identifica con un código (3 letras y 5 números).
- Los libros pueden ser de 3 géneros: *Teatro*, *Poesía* o *Narración*.
- Cada libro tiene dos tipos: *Nacional* o *Internacional*. Según la nacionalidad del autor / autora.



Figura 1: El bibliotecario más joven, tomando un descanso mientras ordena los libros que devolvieron los usuarios en la semana.

2. Objetivos

2.1. Estanterías

Definir un TDA que represente a una estantería en la que se guardan los libros en la biblioteca. Cada estantería se modela con 2 pilas de libros, una de libros de autores *Nacionales* y otra de libros de autores *Internacionales*. Estas pilas se van modificando a medida que se prestan y se devuelven libros. Cada estantería se identifica con un número (entre 0 y 999). Las estanterías tienen una cantidad crítica de libros para cada pila (por defecto 50 libros).

El TDA *Estanteria* debe incluir las siguientes operaciones:

- **guardarLibro(libro):** Agrega el libro que recibe por parámetro a la pila que corresponde a su **tipo**. Si se excede la cantidad **crítica** en alguna pila, se almacena el libro pero se debe informar por pantalla la situación.
- **primerLibroDisponible():** Retorna el primer libro disponible en la estantería. Si hay libros **Nacionales**, devuelve el primer libro a extraer de la pila correspondiente, caso contrario, el primer libro a extraer de la pila de **Internacionales**. No desapila el libro, solo lo muestra.
- **libroParaRecomendar(generoDeLibro):** Recibe por parámetro un género literario y desapila y retorna el primer libro de ese género que se puede prestar. Los libros de autores **Nacionales** se recomiendan primero, si en la pila de Nacionales no hay ningún libro de ese género, se recomiendan los de autores **Internacionales**.
- **buscarLibro(codigoLibro):** Recibe un código de libro (**codigoLibro**) y si en alguna de las pilas (cualquiera de las dos) está ese libro, lo retorna. **El libro no debe ser eliminado de la pila.**
- **prestarLibro(codigoLibro):** Recibe un código de libro (**codigoLibro**) y si el libro se encuentra en alguna de las pilas de la estantea **lo elimina de ella** y lo retorna.
- **librosPorTipo():** Cuenta la cantidad de libros de cada tipo (*Nacional* e *Internacional* por separado) y **retorna los dos valores**.
- **cantidadTotalLibros():** Retorna la cantidad **total** de libros en la estantería, sumando los de las dos pilas.
- **esCritica():** Retorna *True* si alguna de las dos pilas tiene una cantidad de libros mayor que la cantidad **crítica** o *False* en caso contrario.
- **librosPorGenero(generoLibro):** Retorna la cantidad **total** de libros (**de cualquier tipo**) que son del género que se recibe por parámetro (*Teatro*, *Poesía* o *Narración*).

Aclaración: Pueden usar los TDAs Pila (*Stack*) y Cola (*Queue*) que programaron en las guías de ejercicios correspondientes.

2.2. Escritorio de atención

Definir un TDA que represente a un escritorio donde les usuaries piden y devuelven los libros. El escritorio tiene un depósito para los libros, en este depósito hay estanterías que se distribuyen en forma de filas y columnas, es decir, en el depósito tenemos M filas con N estanterías cada una. Modelar con un TDA el Escritorio de atención, teniendo en cuenta que puede haber lugares del depósito donde no hay una estantería, cuando un lugar esta vacío, se representa como *None* en la estructura. Implementar las siguientes operaciones para el TDA del *EscritorioDeAtención*:

- **establecerEstanteria(nroFila, nroColumna, estanteria):** Pone la **estantería** en la ubicación correspondiente.



Figura 2: Fila de estanterías, cada depósito tiene M filas de estas con N estanterías en cada una.

- **cantidadDeEstanteriasCriticas(nroFila)**: Operación **recursiva** que retorna la cantidad de estanterías en situación **crítica** en la fila que recibe por parámetro.
- **estanteríaMenosRecargada()**: Retorna la ubicación (número de fila y número de columna) de la estantería que tiene el **menor número** de libros de tipo *Nacional* en todo el depósito.
- **buscaEstanteria(nroEstanteria)**: Recibe el número de estantería y retorna la **ubicación** de la estantería en el depósito (número de fila y número de columna).
- **guardarLibros(colaDelibros)**: Operación que recibe como entrada una cola de libros y los debe guardar de a uno en la **estantería menos recargada** del depósito. Notar que la estantería menos recargada del depósito puede ir variando a medida que se van mandando los libros de la cola.
- **sacarLibros(colaDelibros)**: Operación que recibe como entrada una cola con códigos de libros y los debe buscar y sacar de las estanterías del depósito. **La operación debe retornar una pila con todos los libros que se sacaron.** Notar que algunos libros pueden no estar en el depósito.
- **moverLibro(codigoLibro, nroEstanteriaOrigen, nroEstanteriaDestino)**: Saca el libro con código **codigoLibro** de la estantería **nroEstanteriaOrigen** y lo pasa a la estantería de **nroEstanteriaDestino**.

3. Datos de prueba

Previo a la entrega del trabajo práctico, deben controlar que su implementación funcione con un lote de datos de prueba que nosotros les vamos a entregar. El lunes 24 de mayo, subiremos al *webcampus* el lote de datos de prueba, junto con un programa en *Python* para ejecutarlo. Por favor, respetar los nombres de las operaciones y de los TDAs (*Libro*, *Estanteria* y *Escritorio-DeAtencion*).

4. Entrega

La entrega del trabajo práctico debe ser:

- Un informe escrito (doc, pdf. etc), incluyendo:
 - Descripción de cada una de las **estructuras de datos** diseñadas e implementadas. Incluir una descripción escrita de los algoritmos. Pueden incluir diagramas de flujo.
 - Descripción de la implementación en *Python*. Explicar **claramente** que hace cada función y procedimiento implementados.
- Código completo y comentado de la implementación.
- Opcional: Video explicando cómo funciona el algoritmo.

Luego de la entrega, les vamos a hacer preguntas a las/los integrantes del grupo sobre el trabajo, así que les aconsejamos no copiar código que encuentren en internet.