



# UNA HUR

UNIVERSIDAD NACIONAL  
DE HURLINGHAM

## ESTRUCTURA DE DATOS

TP N°2:  
Biblioteca

Tema:  
PILAS Y COLAS

Alumnos:

Miranda Tomas - DNI 41.575.063

Araus Luciano - DNI 42.030.424

Rey Pablo - DNI 31.727.015

Profesores:

Gonzalez Sergio, Aclocchi Ariel

## 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 un depósito con varias estanterías con libros.

## 2. TDAs:

### 2.1. TDA Libro:

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.



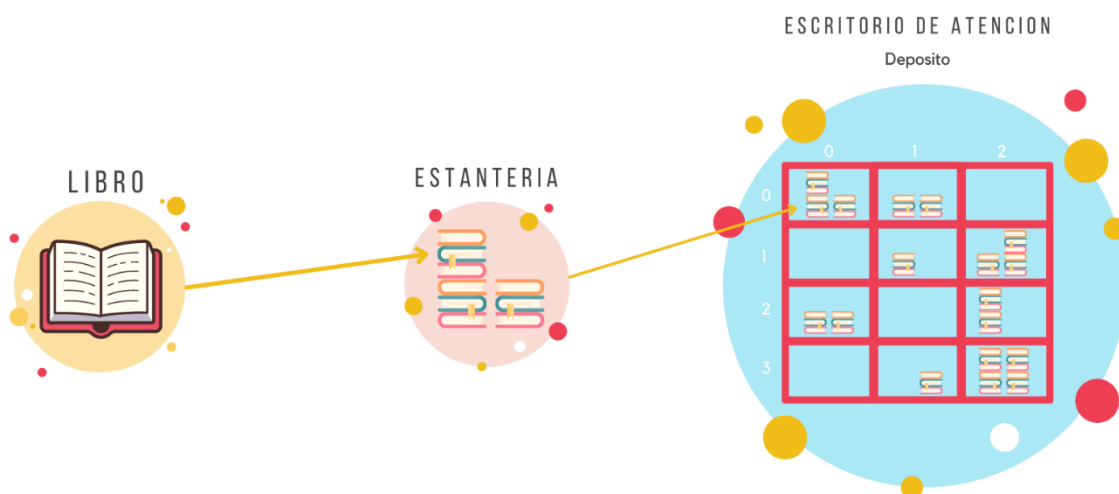
## 2.2. TDA Estantería:

El TDA Estantería representa a una estantería en la que se guardan libros dentro de una 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 tiene un **numero de identificación** (entre 0 y 999). Las estanterías tienen una cantidad **crítica** de libros para cada pila (por defecto 50 libros).



## 2.3. TDA EscritorioAtencion:

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. Puede haber lugares del depósito donde no hay una estantería, cuando un lugar está **vacío**, se representa como **None** en la estructura.



### 3. Funciones

#### 3.1. Funciones TDA Libro:

El TDA **Libro** solo posee las funciones necesarias para su constructor y para sus respectivos validadores.

#### 3.2. Funciones TDA Estantería:

El TDA **Estantería** incluye las siguientes operaciones:

1. **guardarLibro(libro)**: Se definen **2 bloques** según la nacionalidad del libro (**Nacionales o Internacionales**). En primera instancia, se compara la nacionalidad del libro pasado por parámetro con las 2 nacionalidades existentes. A su vez, se chequea el tamaño de la pila correspondiente según su nacionalidad y, si es inferior a la cantidad **critica** establecida (50), lo apila. En caso contrario, lo apila y a su vez informa que la cantidad critica fue superada.
2. **primerLibroDisponible()**: Esta función se encarga de retornar el primer libro disponible de la estantería, **partiendo de** los libros **nacionales**. Si las 2 pilas se encuentran vacías, se lanza una excepción informándolo.
3. **libroParaRecomendar(generoDeLibro)**: Esta función se encarga de devolver el primer libro que encuentre con el **género** que se pase **por parámetro**. **Primero** busca en la pila de autores **nacionales** y **luego** en la pila de autores **internacionales**.

*Funciones Auxiliares:*

- 3.1. **recomendarLibroNacional(generoDeLibro)**: Recomienda un libro de autor nacional del género pasado por parámetro. Clona la pila de autores nacionales y la guarda en una variable. Mientras no esté vacía, y **si el género del primer libro de la pila es igual al género pasado por parámetro, guarda el libro** en una variable que luego retornara. El libro a su vez es removido de la pila auxiliar.
- 3.2. **recomendarLibroInternacional(generoDeLibro)**: Recomienda un libro de autor internacional del género pasado por parámetro. Clona la pila de autores internacionales y la guarda en una variable. Mientras no esté vacía, y **si el género del primer libro de la pila es igual al género pasado por parámetro, guarda el libro** en una variable que luego retornara. El libro a su vez es removido de la pila auxiliar.
4. **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 se elimina de la pila.**

*Funciones Auxiliares:*

- 4.1. **buscarLibroNacional(codigoLibro)**: Esta función se encarga de buscar un libro de autor nacional, y lo devuelve. Para llevarlo a cabo, se clona la pila de autores nacionales y se guarda como una pila auxiliar. Mientras la pila auxiliar no esté vacía y **el código del libro que se vaya a**

**desapilar sea el mismo al código buscado, se guarda el libro que se desapilo** (para luego ser retornado) y se termina de vaciar la lista (si tuviera libros). En el caso de que no la pila este vacía o no se encuentre el código de libro buscado, se devuelve None.

- 4.2. **buscarLibroInternacional(codigoLibro)**: Esta función se encarga de buscar un libro de autor internacional, y lo devuelve. Para llevarlo a cabo, se clona la pila de autores internacionales y se guarda como una pila auxiliar. Mientras la pila auxiliar no esté vacía y **el código del libro que se vaya a desapilar sea el mismo al código buscado, se guarda el libro que se desapilo** (para luego ser retornado) y se termina de vaciar la lista (si tuviera libros). En el caso de que no la pila este vacía o no se encuentre el código de libro buscado, se devuelve None.
5. **prestarLibro(codigoLibro)**: Recibe un código de libro (**codigoLibro**) y se busca que coincida con el código del libro que se encuentre en alguna de las pilas de la estantería. Luego si se encuentra **lo elimina de ella** y lo retorna. En caso contrario, lanza una excepción informándolo.
- Funciones Auxiliares:*
- 5.1. **prestarLibroNacional(codigoLibro)**: Recibe un código de libro por parámetro y lo busca en la pila de autores nacionales. Para poder llevarlo a cabo, se crea un **clon de la pila y se invierte**, a su vez, la **pila de autores original es vaciada** para luego ser llenada (con los libros de la pila invertida) **filtrando los libros** cuyo código no coincida con el buscado. **Si el código coincide, se guarda el libro en una variable y se retira** de la pila. La pila clon auxiliar es desapilada y apilada en la pila de autores original hasta vaciarse por completo. Finalmente, si el libro código del libro se encontró, se devuelve el libro desde la variable que lo hospedaba.
- 5.2. **prestarLibroInternacional(codigoLibro)**: Recibe un código de libro por parámetro y lo busca en la pila de autores Internacionales. Para poder llevarlo a cabo, se crea un **clon de la pila y se invierte**, a su vez, la **pila de autores original es vaciada** para luego ser llenada (con los libros de la pila invertida) **filtrando los libros** cuyo código no coincida con el buscado. **Si el código coincide, se guarda el libro en una variable y se retira** de la pila. La pila clon auxiliar es desapilada y apilada en la pila de autores original hasta vaciarse por completo. Finalmente, si el libro código del libro se encontró, se devuelve el libro desde la variable que lo hospedaba.
6. **librosPorTipo()**: Devuelve el tamaño de las pilas de **autores nacionales e internacionales** por separado.
7. **cantidadTotalLibros()**: Retorna la cantidad **total** de libros en la estantería, sumando los de las dos pilas.
8. **esCritica()**: Retorna True si alguna de las dos pilas tiene una cantidad de libros mayor que la cantidad **crítica** o False en caso contrario.
9. **librosPorGenero(generoLibro)**: Retorna la cantidad **total** de libros (**de cualquier nacionalidad**) que son del género que se recibe por parámetro (Teatro, Poesía o Narración).

## 2.2. Escritorio de atención

El TDA **EscritorioDeAtención** incluye las siguientes operaciones:

- **establecerEstanteria(nroFila, nroColumna, estanteria)**: Esta función se encarga de establecer la estanteria pasada por parámetro en la fila y columna (también pasadas por parámetro) deseadas en el depósito.
- **cantidadDeEstanteriasCriticas(nroFila)**: Esta función comienza preguntando si n es igual al final de la columna y si en esa posición hay una estanteria crítica. En tal caso, el contador suma 1, sino devuelve 0. En caso de que no esté en el final, es decir, fila-columna (ultima posición) se llama recursivamente hasta llegar al final de la fila pasada por parámetro.
- **estanteríaMenosRecargada()**: Devuelve la estanteria menos cargada de libros nacionales del depósito. El algoritmo se inicia con una variable auxiliar con un numero random alto. Comienza a recorrerse el deposito por filas y columnas. Si se topa con una estanteria y si el tamaño de su pila de autores nacionales es menor al numero guardado en la variable auxiliar, guarda la ubicación de cuya estanteria y actualiza la variable auxiliar.
- **buscaEstanteria(nroEstanteria)**: Recibe el número de estantería y retorna la **ubicación** de la estantería en el depósito (número de la y número de columna). Esta función se encarga de buscar una estanteria a través de su nro. de estanteria (pasado por parámetro) y devuelve su fila y columna. Si la estanteria no es encontrada, devuelve una excepción informándolo.
- **guardarLibros(colaDelibros)**: Esta función recibe por parámetro una cola de libros y los guarda uno por uno en la **estantería menos recargada** del depósito. A medida que se desencola la cola de libros se guardan los libros en la estanteria.
- **sacarLibros(colaDelibros)**: Esta función se encarga de eliminar los libros (que se pasan por parámetro en una cola de códigos de libros) del depósito. Se comienza el algoritmo con un recorrido por el depósito. Si se topa con una estanteria, se aplica la función '*buscarLibro()*' pasándole por parámetro el código del primer libro de la cola. Si el libro es encontrado, se retira de la cola y se agrega a una pila junto a los próximos libros que sean encontrados. Finalmente, la pila con los libros encontrados es retornada.
- **moverLibro(codigoLibro, nroEstanteriaOrigen, nroEstanteriaDestino)**: Esta función mueve un libro (pasado por parámetro), de la estanteria de origen a la estanteria destino (ambas pasadas por parámetro). Se comienza buscando la estanteria de origen y el libro a mover. Si el libro existe, se utiliza la función '*prestarLibro()*' para retirarlo y guardarlo en la variable '*libroAMover*'. Posteriormente, se busca la estanteria destino y finalmente se guarda el libro en dicha estanteria. En caso de no encontrarse el libro, se informa mediante una excepción.



Link al trabajo en Colaboratory:

<https://colab.research.google.com/drive/1y4eEsGj7GPcdULvmog4P6wdbFn4pXok>

Link del enunciado:

[https://drive.google.com/file/d/1PDtFuPW4GRg\\_BjL-NIIEggGpqM4eeqkw/view](https://drive.google.com/file/d/1PDtFuPW4GRg_BjL-NIIEggGpqM4eeqkw/view)