



Trabajo Práctico: Pilas y colas

Estructuras de datos
Comision A (Turno Noche)
Comision B (Turno Mañana)
Primer cuatrimestre de 2020

Docentes	Sergio Gonzalez Ariel Clocchiatti
Email	sergio.gonzalez@unahur.edu.ar aclocchi@gmail.com
Fecha de entrega	05 / 06 / 2020

Índice

1. Introducción	2
1.1. Presentación del problema	2
2. Objetivos	3
2.1. Juzgados	3
2.2. Tribunales	3
3. Informe	4

1. Introducción

En este trabajo práctico vamos a trabajar con la integración de 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*.

1.1. Presentación del problema

En los tribunales de justicia, por fin deciden informatizar el manejo de los expedientes, desde el momento de la entrada hasta que termina el proceso. Para eso contratan a un/una estudiante de la UNAHUR para que modele el problema y diseñe e implemente una solución que le permita automatizar el proceso.

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

- Cada expediente tiene un número que lo identifica.
- Los expedientes pueden ser de 5 tipos, según el fuero de la causa: *Civil*, *Penal*, *Laboral*, *Familia* o *Comercial*.
- Los expedientes pueden ser de 2 tipos, según su prioridad: *Normal* o *Urgente*.
- Cada expediente tiene dos estados: *Investigación* o *Juicio*.

Durante este trabajo van a implementar tipos de datos y operaciones para trabajar con ellos. Para resolver el problema de organización de los tribunales, vamos a modelar los expedientes, los juzgados y finalmente el edificio de tribunales con su mesa de entradas.



Figura 1: Juzgado que gano el premio al "Más ordenado del año 2019".

2. Objetivos

2.1. Juzgados

Definir un TDA que represente a un juzgado que maneja dos colas de expedientes, una de expedientes *Urgentes* y otra de expedientes "Normales". Cada juzgado se identifica con el nombre del Juez que lo preside. Los juzgados tienen una cantidad crítica de expedientes de cada tipo (por defecto 50 expedientes).

El TDA **Juzgado** debe incluir las siguientes operaciones:

- **recibirExpediente(exp)**: Agrega el expediente que recibe por parámetros a la cola que corresponde a su **prioridad**. Si se excede la cantidad **crítica**, se debe informar por pantalla.
- **primerExpedienteATratar()**: Retorna el primer expediente a tratar. Si hay expedientes **urgentes**, devuelve el expediente a extraer de la cola correspondiente, caso contrario, el expediente a extraer de los **normales**. No desencola el expediente, solo lo muestra.
- **tratarExpediente()**: Desencola el primer expediente a tratar de la cola en la que está y lo retorna. **Si hay expedientes urgentes, es urgente, sino normal**.
- **cantidadTotalExpedientes()**: Cantidad **total** de expedientes, sumando los de las dos colas.
- **expedientesPorTipo()**: Cantidad de expedientes de cada tipo (*Normales* y *Urgentes* por separado) y **retorna ambos valores**.
- **esCritico()**: Retorna *True* si alguna de las dos colas tiene una cantidad mayor que la cantidad **crítica** o *False* en caso contrario.
- **enJuicio()**: Retorna la cantidad total de expedientes (*Normales* o *Urgentes*) que están en estado de *Juicio*.
- **buscarExpediente(nroExp)**: Recibe un número de expediente y si está en alguna de las colas (cualquiera de las dos) del juzgado lo retorna. **El expediente no debe ser eliminado del juzgado**.
- **eliminarExpediente(nroExp)**: Recibe un número de expediente y si está en alguna de las colas del juzgado, **lo elimina de ella**.
- **cambiaDeEstado(nroExp)**: Cambia la prioridad del expediente de número *nroExp* (de *Normal* a *Urgente* o viceversa), en consecuencia, **debe cambiarlo de cola**. La operación debe **verificar** previamente que el expediente esté en el juzgado.

2.2. Tribunales

En los edificios de los tribunales, los juzgados se organizan en una cantidad fija por piso. En cada edificio, tenemos M pisos con N oficinas cada uno. Cada oficina corresponde a un único juzgado, identificado con el nombre del juez que lo tiene a cargo. Modelar con un TDA el Edificio de Tribunales, teniendo en cuenta que puede haber oficinas vacías, cuando una oficina está vacía, se representa como *None* en la estructura. Implementar las siguientes operaciones para el TDA del **EdificioTribunales**:

- **establecerJuzgado(piso, oficina, juzgado)**: Pone el **juzgado** en la oficina en la que se encuentra.
- **cantidadDeJuzgadosCriticos(piso)**: Operación **recursiva** que retorna la cantidad de juzgados en situación **crítica** en el piso que recibe por parámetro.

- **juzgadoMenosRecargado():** Retorna la ubicación (piso y oficina) del juzgado que tiene el **menor número** de expedientes *Urgentes* pendientes en todo el edificio.
- **buscaJuzgado(juez):** Recibe el nombre del juez y retorna la **ubicación** de la oficina en el edificio (piso y oficina).
- **mesaDeEntradas(pilaDeExpedientes, juez):** Operación que actúa como la **Mesa de entradas** del edificio de tribunales. Recibe como entrada una pila de expedientes y el juzgado que se encuentra de turno (Nombre del Juez). La función debe enviar los expedientes al juzgado que se encuentre de turno. Si el juzgado de turno llega a la **cantidad crítica** de expedientes, los expedientes **restantes** se deben enviar al **juzgado menos recargado** de todo el edificio.
- **moverExpediente(nroExp, juezOrigen, juezDestino):** Toma el expediente de número *nroExp* del juzgado del **juezOrigen** y lo envía al juzgado del **juezDestino**.

3. Informe

La entrega del trabajo práctico debe ser con un informe escrito, incluyendo:

- Descripción de cada una de las **estructuras de datos** diseñadas e implementadas. Incluir una descripción escrita de los algoritmos. Pueden incluir diagrama de flujo.
- Descripción de la implementación en *Python*. Explicar **claramente** que hace cada función y procedimiento implementados.
- Código completo de la implementación.
- 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.