

**TRABAJO PRACTICO N° 5**

**PILAS Y COLAS**

**Nota1:** para cada clase hacer un programa que pruebe cada uno de los métodos que provee.

**Nota2:** documentar cada clase y utilizar Javadoc para generar la misma.

1. Crear la clase **DoublyLinkedStack** que implementa la interfaz **Stack**. Utilizar la clase **DoublyLinkedList** para su implementación. Crear tres pilas, una implementada con un array, otro con una lista enlazada y la última con una lista doblemente enlazada. Medir los tiempos para las mismas operaciones en las distintas implementaciones.

2. Utilizar una pila para cargar una nómina de empleados (ejercicio 2.10). Calcular el sueldo para cada uno de ellos.

3. Implementar el método *clone()* para la clase **ArrayStack**

4. Implementar el método *clone()* para la clase **LinkedStack**

5. Realizar una aplicación que controle el balanceo de paréntesis, corchetes y llaves. La aplicación debe leer el nombre del archivo desde la línea de comando. Si hay un desbalanceo debe indicar la línea donde se encuentra el error.

6. Realizar una aplicación que controle los delimitadores del lenguaje de marcas HTML. Este lenguaje permite atributos opcionales que son parte del tag de apertura. En general el formato utilizado para expresar un atributo es `<name attribute1="value1" attribute2="value2">`, por ejemplo `<table border="3" cellpadding="5">`. La aplicación debe leer el nombre del archivo desde la línea de comando e indicar el tag, la línea y la columna si encuentra un error.

7. Utilizar una cola para cargar empleados y facturas (ejercicio 2.11). Calcular los importes a pagar.

8. Crear la clase **ArrayCircularQueue** que implementa la interfaz **CircularQueue**.

9. Crear la clase **ArrayCircularDeque** que implementa la interfaz **Deque**.

10. Crear la clase **DoublyLinkedDeque** que implementa la interfaz **Deque**. Utilizar la clase **DoublyLinkedList** para su implementación.

11. Realizar una aplicación que simule las colas de un supermercado. Los clientes, una vez que realizaron sus compras, se ubican en la caja donde hay menos personas a ser atendida. El tiempo de atención de cada cliente es proporcional a la cantidad de productos más un tiempo adicional por el pago de

**TRABAJO PRACTICO N° 5**

**PILAS Y COLAS**

los mismos. Después de un determinado tiempo no ingresan más clientes al supermercado y la simulación termina cuando se atiende al último cliente en la caja. La aplicación debe ser configurable para permitir probar con distintos escenarios la simulación. Por ejemplo: número de cajas, cantidad de clientes, cantidad máxima de productos que compra un cliente, tiempos de atención en la caja, etc. Al finalizar la simulación debe realizar un reporte indicando la cantidad de clientes atendido por cada caja, el tiempo que estuvo sin atender a ningún cliente, el tiempo promedio de atención de un cliente, el tiempo máximo que esperó un cliente, cantidad de productos comprados en promedio, etc.