



# UNIVERSIDAD TECMILENIO

## Actividad 2

**Nombre Alumno:** Abish Abril Santana García

**Materia:** Estructura de datos

**Nombre Profesor:**

Rojas Perea, Adalberto Emmanuel

Descripción



**HALCONES**



**HALCONES**

En esta actividad, implementarás estructuras de datos de pilas y colas utilizando listas ligadas en Java. Estas estructuras simularán un sistema operativo básico que gestionará tareas y comandos.

Desarrollo de un módulo simple para un sistema operativo que realice lo siguiente:

- **Pilas:** Que gestione un historial de comandos ejecutados, permitiendo revisar o deshacer el último comando.
- **Colas:** Que administre los programas en ejecución, asegurando que se procesen en el orden de llegada.

### Objetivo

- Hacer uso de listas ligadas para implementar pilas y colas.
- Aplicar estas estructuras en un contexto de simulación de gestión de tareas.
- Fortalecer la habilidad de programación en Java.

## Instrucciones

1. Revisa y ajusta el código de listas ligadas entregado en la Actividad 1, asegurándote de que la implementación sea adecuada para ser utilizada en las estructuras de pilas y colas.
2. Implementación de pilas y colas:
  - a. Implementa las clases Pila y Cola utilizando las listas ligadas.

**Explicación:** Ambas clases usan nodos de lista ligada. La Pila solo necesita un puntero al tope, mientras que la Cola necesita dos punteros: al frente y al final.

### Clase Pila con lista ligada:

```
class Pila {  
    private Nodo tope;  
  
    public Pila() {  
        this.tope = null;  
    }  
}
```

### Clase Cola con lista ligada:

```
class Cola {  
    private Nodo frente;  
    private Nodo fin;  
  
    public Cola() {  
        this.frente = null;  
        this.fin = null;  
    }  
}
```

- b. Codifica métodos para operaciones básicas como agregar (Push/Enqueue), remover elementos (Pop/Dequeue) y verificar el tope de la Pila o de la Cola (Peek).

**Explicación:** La Pila usa LIFO (último en entrar, primero en salir) con push/pop. La Cola usa FIFO (primero en entrar, primero en salir) con encolar/desencolar. Ambos tienen métodos para verificar el elemento superior sin removerlo (peek/verProximo).

#### Métodos de la Pila:

```
public boolean estaVacia() {
    return tope == null;
}

public void push(String comando) {
    Nodo nuevo = new Nodo(comando);
    nuevo.siguiente = tope;
    tope = nuevo;
    System.out.println("Comando '" + comando + "' ejecutado");
    ejecutar(comando);
}

public String pop() {
    if (estaVacia()) {
        System.out.println("No hay comandos para deshacer");
        return null;
    }
    String comando = tope.dato;
    tope = tope.siguiente;
    System.out.println("Deshaciendo: " + comando);
    return comando;
}

public String peek() {
    if (estaVacia()) {
        System.out.println("No hay comandos recientes");
        return null;
    }
    return tope.dato;
}
```

## Métodos de la Cola:

```
public boolean estaVacia() {
    return frente == null;
}

public void encolar(String proceso) {
    Nodo nuevo = new Nodo(proceso);
    if (estaVacia()) {
        frente = nuevo;
    } else {
        fin.siguiente = nuevo;
    }
    fin = nuevo;
    System.out.println("Proceso '" + proceso + "' en cola");
}

public String desencolar() {
    if (estaVacia()) {
        System.out.println("No hay procesos");
        return null;
    }
    String proceso = frente.dato;
    frente = frente.siguiente;
    if (frente == null) {
        fin = null;
    }
    System.out.println("Ejecutando: " + proceso);
    return proceso;
}

public String verProximo() {
    if (estaVacia()) {
        System.out.println("No hay procesos");
        return null;
    }
    return frente.dato;
}
```

### 3. Integración y pruebas:

- a. Integra las estructuras de datos en un módulo de gestión de sistema operativo simulado.

**Explicación:** El código integra ambas estructuras en un menú interactivo. La pila gestiona comandos del sistema y la cola gestiona procesos. Cada opción del menú llama a los métodos correspondientes de las estructuras.

```
public class Actividad2 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        Pila pila = new Pila();  
        Cola cola = new Cola();  
  
        int opcion;  
  
        System.out.println("Sistema Operativo ");  
  
        do {  
            System.out.println("\n1. Ejecutar comando");  
            System.out.println("2. Deshacer comando");  
            System.out.println("3. Ver último comando");  
            System.out.println("4. Ver historial");  
            System.out.println("5. Agregar proceso");  
            System.out.println("6. Ejecutar proceso");  
            System.out.println("7. Ver próximo proceso");  
            System.out.println("8. Ver cola de procesos");  
            System.out.println("9. Salir");  
            System.out.print("Opción: ");
```

- b. Realiza pruebas para verificar la funcionalidad y detectar errores en el código.

**Explicación:** Cada método incluye verificación de errores para casos donde las estructuras están vacías. El sistema muestra mensajes apropiados cuando se intenta realizar operaciones inválidas, lo que permite detectar y manejar errores durante las pruebas.

```
// En método pop() de Pila
if (estaVacia()) {
    System.out.println("No hay comandos para deshacer");
    return null;
}

// En método desencolar() de Cola
if (estaVacia()) {
    System.out.println("No hay procesos");
    return null;
}

// En método peek() de Pila
if (estaVacia()) {
    System.out.println("No hay comandos recientes");
    return null;
}

// En método verProximo() de Cola
if (estaVacia()) {
    System.out.println("No hay procesos");
    return null;
}
```

## Resultados

1. Ejecutar comando
2. Deshacer comando
3. Ver último comando
4. Ver historial
5. Agregar proceso
6. Ejecutar proceso
7. Ver próximo proceso
8. Ver cola de procesos
9. Salir

Opción: 1

Comandos: ls, cd documentos, cd imagenes, cd musica, pwd, help

Comando: cd imagenes

Comando 'cd imagenes' ejecutado

Directorio cambiado a Imágenes

C:\Users\SANTANA\Pictures

1. Ejecutar comando
2. Deshacer comando
3. Ver último comando
4. Ver historial
5. Agregar proceso
6. Ejecutar proceso
7. Ver próximo proceso
8. Ver cola de procesos
9. Salir

Opción: 1

Comandos: ls, cd documentos, cd imagenes, cd musica, pwd, help

Comando: cd musica

Comando 'cd musica' ejecutado

Directorio cambiado a Música

C:\Users\SANTANA\Music\Rolitas chidas

1. Ejecutar comando
2. Deshacer comando
3. Ver último comando
4. Ver historial
5. Agregar proceso
6. Ejecutar proceso
7. Ver próximo proceso
8. Ver cola de procesos
9. Salir

Opción: 4

--- Historial ---

- pwd
- cd musica
- cd imagenes
- cd documentos