

Nombre: [titomartir/Pilas-y-Colas: Ejercicios de clases](#)

Caramelos Salvavidas (Simulación con TDA Pila)

1. Descripción del TDA:

Este problema simula el comportamiento de un niño que selecciona caramelos amarillos de un paquete utilizando una **pila**. El niño saca los caramelos uno por uno, come solo los amarillos y devuelve los caramelos de otros colores al paquete, manteniendo el orden original. Para esto, se utiliza el TDA Pila, que sigue el principio **LIFO (Last In, First Out)**.

2. Invariante de TDA:

- La pila representa el paquete de caramelos.
- Cada caramelo es un elemento de la pila, donde el color amarillo es el que el niño prefiere.
- El orden de los caramelos se mantiene al devolver los no deseados a la pila.

3. Operaciones:

Nombre de operación: sacarCaramelo

- **Descripción:** Saca un caramelo de la pila (paquete).
- **Descripción operacional:**
 - Se extrae el caramelo en la cima de la pila.
 - Si el caramelo es amarillo, el niño lo come.
 - Si no es amarillo, se guarda en una pila temporal para devolverlo luego.
- **Precondición:** La pila no debe estar vacía.
- **Postcondición:** El caramelo se elimina de la pila original y se procesa según su color.

Nombre de operación: devolverCaramelos

- **Descripción:** Devuelve los caramelos no deseados a la pila (paquete) en el orden original.
- **Descripción operacional:**
 - Los caramelos guardados en la pila temporal se devuelven a la pila original.
- **Precondición:** La pila temporal no debe estar vacía.
- **Postcondición:** Los caramelos no deseados se restauran en la pila original en el orden correcto.

Nombre de operación: mostrarPaquete

- **Descripción:** Muestra el estado actual del paquete de caramelos.
- **Descripción operacional:**
 - Se imprime el contenido de la pila.
- **Precondición:** Ninguna.
- **Postcondición:** Ninguna.

Código con Precondiciones y Postcondiciones

java

Copy

```
import java.util.Stack;
```

```
import java.util.Scanner;
```

```
public class CaramelosSalvavidas {
```

```
    public static void main(String[] args) {
```

```
        Stack<String> paquete = new Stack<>();
```

```
        Stack<String> temporal = new Stack<>();
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        // Llenar el paquete con caramelos de diferentes colores
```

```
        paquete.push("Rojo");
```

```
        paquete.push("Amarillo");
```

```
        paquete.push("Verde");
```

```
        paquete.push("Amarillo");
```

```
        paquete.push("Azul");
```

```
        paquete.push("Amarillo");
```

```
        System.out.println("Paquete inicial: " + paquete);
```

```
// Proceso de selección de caramelos
while (!paquete.isEmpty()) {
    // Precondición: La pila no debe estar vacía.
    String caramelo = paquete.pop(); // Sacar un caramelo
    if (caramelo.equals("Amarillo")) {
        System.out.println("¡Caramelo amarillo encontrado! El niño se lo come.");
    } else {
        System.out.println("Caramelo " + caramelo + " guardado para devolver.");
        temporal.push(caramelo); // Guardar en la pila temporal
    }
    // Postcondición: El caramelo se elimina de la pila original y se procesa según su color.
}

// Devolver los caramelos no deseados al paquete
while (!temporal.isEmpty()) {
    // Precondición: La pila temporal no debe estar vacía.
    paquete.push(temporal.pop()); // Devolver a la pila original
    // Postcondición: Los caramelos no deseados se restauran en la pila original en el orden
    // correcto.
}

System.out.println("Paquete final después de devolver los caramelos: " + paquete);
}
}
```

Explicación del código

1. Estructura de datos:

- Se utilizan dos pilas:

- paquete: Representa el paquete de caramelos.
- temporal: Almacena los caramelos no deseados para devolverlos al paquete.

2. Proceso de selección:

- El niño saca caramelos uno por uno de la pila paquete.
- Si el caramelo es amarillo, lo come.
- Si no es amarillo, lo guarda en la pila temporal.

3. Devolución de caramelos:

- Los caramelos no deseados se devuelven a la pila paquete desde la pila temporal, manteniendo el orden original.

4. Salida:

- Se muestra el estado inicial y final del paquete de caramelos.

Ejemplo de ejecución

Entrada:

El paquete inicial contiene: [Rojo, Amarillo, Verde, Amarillo, Azul, Amarillo].

Salida:

Copy

Paquete inicial: [Rojo, Amarillo, Verde, Amarillo, Azul, Amarillo]

¡Caramelo amarillo encontrado! El niño se lo come.

Caramelo Azul guardado para devolver.

¡Caramelo amarillo encontrado! El niño se lo come.

Caramelo Verde guardado para devolver.

¡Caramelo amarillo encontrado! El niño se lo come.

Caramelo Rojo guardado para devolver.

Paquete final después de devolver los caramelos: [Rojo, Verde, Azul]

Resumen

Este programa simula el proceso de selección de caramelos amarillos utilizando el TDA Pila. Los caramelos no deseados se devuelven al paquete en el orden original, respetando las

precondiciones y postcondiciones de cada operación. Es una aplicación práctica del TDA Pila en un contexto cotidiano.