



### **Avance del proyecto**

**Nombre Alumno:** Abish Abril Santana García

**Nombre Alumno:** Juan Esteban Campos Cruz

**Nombre Alumno:** Juan Carlos Malanco Rdez.

**Materia:** Estructura de datos

**Nombre Profesor:**

Rojas Perea, Adalberto Emmanuel



## **Descripción**

Los aprendedores desarrollarán un programa en Java que simule un sistema de gestión de tareas para una pequeña empresa. Este sistema utilizará diferentes estructuras de datos (pilas, colas y listas) para manejar tareas asignadas a distintos departamentos. El objetivo es practicar la implementación de estas estructuras y entender sus aplicaciones prácticas.

## **Objetivo**

Implementar y manipular diferentes estructuras de datos en Java, así como entender las aplicaciones prácticas de pilas, colas y listas en el manejo de información para desarrollar habilidades de programación y resolución de problemas en un contexto empresarial simulado.

## Instrucciones

1. **Revisar conceptos básicos:** antes de empezar a programar, revisa la información sobre pilas, colas, listas y una introducción a las estructuras de datos para refrescar tus conocimientos.
2. **Entorno de desarrollo:** asegúrate de tener Java y Apache Netbeans instalados en tu computadora. Además, configura tu entorno de desarrollo para iniciar el proyecto.
3. **Diseño del programa:**
  - a. **Estructura del programa:** divide tu programa en tres partes principales, cada una correspondiente a una estructura de datos (pila, cola, lista).
  - b. **Pilas:** utiliza pilas para manejar tareas urgentes que deben realizarse lo antes posible. implementa operaciones básicas como *push*, *pop* y *peek*.

### 1. Pila - Tareas Urgentes

Explicación: Se usa una pila para manejar tareas urgentes. La última tarea agregada es la primera en atenderse.

```
private static Stack<TareaUrgente> tareasUrgentes = new Stack<>();

// Operaciones:
tareasUrgentes.push(new TareaUrgente(descripcion)); // Agregar
TareaUrgente resuelta = tareasUrgentes.pop();        // Resolver
TareaUrgente siguiente = tareasUrgentes.peek();      // Ver siguiente
```

- c. **Colas:** emplea colas para administrar tareas en una secuencia FIFO (primero en entrar, primero en salir), ideal para tareas programadas regularmente. implementa operaciones básicas como **enqueue**, **dequeue** y **front**.

## 2. Cola - Pedidos

Explicación: Se usa una cola para gestionar pedidos. Los pedidos se atienden en orden de llegada.

```
// Operaciones:
colaDePedidos.add(nuevoPedido);           // Agregar
Pedido preparado = colaDePedidos.poll();  // Preparar
Pedido siguiente = colaDePedidos.peek();  // Ver siguiente
```

- d. **listas:** usa listas para tareas que pueden requerir acceso aleatorio, como la consulta de tareas pendientes por departamento. implementa operaciones básicas como **insert**, **delete** y **find**.

## 3. Lista - Menú de Productos

Explicación: Se usa una lista para el menú, permitiendo acceso aleatorio para buscar, agregar o eliminar productos.

```
// Operaciones:
menu.add(new Producto(...));             // Agregar producto
menu.remove(producto);                   // Eliminar producto
buscarProducto(id);                      // Buscar por ID
```

## 4. Funcionalidad adicional:

- a. Implementa una interfaz de usuario sencilla en la consola para interactuar con el sistema.

**Explicación:** El sistema cuenta con un menú interactivo por consola que permite entrar a todas las opciones usando entradas numéricas.

```
switch (opcion) {           // Ejecuta la opción seleccionada
    case 1: gestionarTareasUrgentes(); break;
    case 2: gestionarPedidos(); break;
    case 3: gestionarMenu(); break;
    case 4: verTodasLasTareasPendientes(); break;
    case 0: System.out.println("Saliendo..."); break;
    default: System.out.println("Opción no válida.");
}
```

- b. Permite al usuario agregar, ver y eliminar tareas en cada estructura de datos.

**Explicación:** El sistema permite realizar las tres operaciones básicas (agregar, ver y eliminar) en cada una de las tres estructuras de datos principales. Cada estructura tiene su propio menú específico con estas funcionalidades implementadas.

```
System.out.println(x:"\n--- Gestión de Menú ---");
System.out.println(x:"1. Ver menú (Ver Lista)");
System.out.println(x:"2. Agregar producto (Insertar)");
System.out.println(x:"3. Eliminar producto (Borrar)");
System.out.println(x:"4. Buscar producto por ID");
System.out.println(x:"0. Volver al menú principal");
```

- c. Incluye una opción para ver todas las tareas pendientes, ordenadas por urgencia y departamento.

**Explicación:** La opción del menú principal muestra un resumen completo de todas las tareas pendientes, organizadas por tipo (urgentes y pedidos).

```
// Ver todas las tareas pendientes
private static void verTodasLasTareasPendientes() {
    System.out.println(x:"\n--- Resumen de Tareas Pendientes ---");
    System.out.println(x:"\n** TAREAS URGENTES (PILA) **");
    if (tareasUrgentes.isEmpty()) {
        System.out.println(x:"No hay tareas urgentes.");
    } else {
        // Recorre la pila sin modificarla
        Stack<TareaUrgente> tempStack = (Stack<TareaUrgente>) tareasUrgentes.clone();
        while (!tempStack.isEmpty()) {
            System.out.println("- " + tempStack.pop());
        }
    }

    System.out.println(x:"\n** PEDIDOS EN COLA **");
    if (colaDePedidos.isEmpty()) {
        System.out.println(x:"No hay pedidos en la cola.");
    } else {
        // Recorre la cola sin modificarla
        for (Pedido p : colaDePedidos) {
            System.out.println("- Pedido #" + p.getIdPedido() + ", Total: $" + p.getTotal());
        }
    }
}
```

5. **Pruebas:** realiza varias pruebas para asegurarte de que todas las operaciones funcionan correctamente en cada estructura de datos.

Menú principal

```
--- Sistema de Gestión de Pizzería ---
1. Gestionar Tareas Urgentes (Pila)
2. Gestionar Pedidos (Cola)
3. Gestionar Menú de Productos (Lista)
4. Ver todas las tareas pendientes
0. Salir
Selecciona una opción: █
```

Gestionar Menú de Productos

```
--- Sistema de Gestión de Pizzería ---
1. Gestionar Tareas Urgentes (Pila)
2. Gestionar Pedidos (Cola)
3. Gestionar Menú de Productos (Lista)
4. Ver todas las tareas pendientes
0. Salir
Selecciona una opción: 3

--- Gestión de Menú ---
1. Ver menú (Ver Lista)
2. Agregar producto (Insertar)
3. Eliminar producto (Borrar)
4. Buscar producto por ID
0. Volver al menú principal
Selecciona una opción: 1

--- Menú Completo ---
ID: 1, Nombre: Pizza Pepperoni, Precio: $150.00
ID: 2, Nombre: Pizza Hawaiana, Precio: $140.00
ID: 3, Nombre: Pizza Vegetariana, Precio: $130.00
ID: 4, Nombre: Refresco de Cola, Precio: $30.00
```

## Gestionar Tareas Urgentes

```
--- Sistema de Gestión de Pizzería ---
1. Gestionar Tareas Urgentes (Pila)
2. Gestionar Pedidos (Cola)
3. Gestionar Menú de Productos (Lista)
4. Ver todas las tareas pendientes
0. Salir
Selecciona una opción: 1

--- Gestión de Tareas Urgentes ---
1. Agregar tarea (Push)
2. Resolver tarea (Pop)
3. Ver tarea más urgente (Peek)
0. Volver al menú principal
Selecciona una opción: 1
Describe la tarea urgente: Pizza Peperoni
Tarea agregada a la pila.

--- Gestión de Tareas Urgentes ---
1. Agregar tarea (Push)
2. Resolver tarea (Pop)
3. Ver tarea más urgente (Peek)
0. Volver al menú principal
Selecciona una opción: 3
Tarea más urgente: Tarea Urgente #1: Pizza Peperoni
```

## Ver todas las tareas pendientes

```
--- Sistema de Gestión de Pizzería ---
1. Gestionar Tareas Urgentes (Pila)
2. Gestionar Pedidos (Cola)
3. Gestionar Menú de Productos (Lista)
4. Ver todas las tareas pendientes
0. Salir
Selecciona una opción: 4

--- Resumen de Tareas Pendientes ---

** TAREAS URGENTES (PILA) **
- Tarea Urgente #1: Pizza Peperoni

** PEDIDOS EN COLA **
No hay pedidos en la cola.
```