



FACULTAD DE INGENIERÍA Y ARQUITECTURA

INGENIERÍA EN DESARROLLO DE SOFTWARE

MANEJO DE ESTRUCTURA DE DATOS

Lab2

Docente:
Ing. Sergio Alberto Bernal

19 de octubre de 2024

Índice

1. Descripción del problema	3
2. Requisitos del sistema	3
3. Funcionalidad del sistema	4
3.1. Ejemplo de flujo del programa	4
3.1.1. Registrar una nueva orden	4
3.1.2. Ver órdenes pendientes	4
3.1.3. Despachar una orden	4
3.1.4. Ver órdenes despachadas	5
3.1.5. Salir del programa	5
4. Aspectos a evaluar	5

1. Descripción del problema

La pupusería ".^{El} Comalito " ha solicitado tu ayuda para desarrollar un sistema que les permita llevar un control más eficiente de las órdenes de sus clientes. Cada cliente puede ordenar múltiples tipos de pupusas en una misma orden, y el sistema debe gestionar estas órdenes, despacharlas y llevar un registro de las mismas.

El sistema es una aplicación de consola que debe mostrar un menú de opciones para que el usuario pueda realizar las siguientes operaciones:

1. **Registrar una nueva orden:** El sistema permitirá al usuario registrar una nueva orden indicando el nombre del cliente y los diferentes tipos de pupusas que desea pedir (por ejemplo, *revuelta*, *queso*, *frijol*).
2. **Ver órdenes pendientes:** El sistema mostrará las órdenes que aún no han sido despachadas, listando el nombre del cliente y los detalles de la orden.
3. **Despachar una orden:** El sistema permitirá al usuario despachar la primera orden registrada y la removerá de la cola de órdenes pendientes.
4. **Ver órdenes despachadas:** El sistema mostrará las órdenes que ya han sido servidas y despachadas, listándolas en el orden en el que fueron despachadas.
5. **Salir:** Salir del programa.

2. Requisitos del sistema

Para la implementación de este sistema, deberás hacer uso de las siguientes estructuras de datos:

- **Lista (List):** Para almacenar los diferentes tipos de pupusas en cada orden.
- **Cola (Queue):** Para gestionar las órdenes pendientes, de manera que la primera orden registrada sea la primera en ser despachada.
- **Pila (Stack):** Para almacenar las órdenes que ya han sido despachadas, permitiendo verlas en el orden inverso al que fueron despachadas.

Cada orden de pupusas debe contener una lista de elementos, donde cada elemento incluye:

- **Tipo de pupusa:** El tipo específico de pupusa (e.g., *revuelta*, *queso*, *chicharrón*).
- **Cantidad:** El número de pupusas de ese tipo que se ordenaron.

3. Funcionalidad del sistema

El sistema debe mostrar un menú interactivo con las siguientes opciones:

```
Bienvenido a la Pupuseria "El Comalito"
Seleccione una opcion:
1. Registrar una nueva orden
2. Ver ordenes pendientes
3. Despachar orden
4. Ver ordenes despachadas
5. Salir
```

3.1. Ejemplo de flujo del programa

3.1.1. Registrar una nueva orden

Cuando el usuario elija la opción **1**, el sistema debe permitirle ingresar los detalles de la orden, incluyendo el nombre del cliente y los diferentes tipos de pupusas que desea ordenar:

```
Ingrese el nombre del cliente: Juan Perez
Cuantos tipos de pupusas desea ordenar?: 2

Ingrese el tipo de pupusa #1: Revuelta
Ingrese la cantidad de pupusas revueltas: 3

Ingrese el tipo de pupusa #2: Queso
Ingrese la cantidad de pupusas de queso: 2

Orden registrada para Juan Perez: 3 pupusas de Revuelta, 2 pupusas de ↵
Queso.
```

3.1.2. Ver órdenes pendientes

Cuando el usuario elija la opción **2**, el sistema debe mostrar la lista de órdenes pendientes. Estas órdenes deben aparecer en el orden en el que fueron registradas:

```
Ordenes pendientes:
1. Juan Perez: 3 pupusas de Revuelta, 2 pupusas de Queso
2. Maria Lopez: 2 pupusas de Chicharron, 1 pupusa de Frijol
```

3.1.3. Despachar una orden

Al seleccionar la opción **3**, el sistema despachará la primera orden en la cola y la removerá de la lista de órdenes pendientes:

```
Despachando la orden de: Juan Perez: 3 pupusas de Revuelta, 2 pupusas ↵
de Queso.
```

3.1.4. Ver órdenes despachadas

La opción 4 mostrará todas las órdenes que ya han sido despachadas:

Órdenes despachadas:

1. Juan Perez: 3 pupusas de Revuelta, 2 pupusas de Queso

3.1.5. Salir del programa

Al seleccionar la opción 5, el sistema finalizará, cerrando el menú de opciones y terminando la ejecución.

4. Aspectos a evaluar

Para la solución de este ejercicio, se evaluarán los siguientes aspectos:

- **Uso correcto de las estructuras de datos:** Se debe verificar que el alumno utilice adecuadamente las estructuras de **List**, **Queue**, y **Stack**.
- **Manejo de errores:** El sistema debe validar las entradas del usuario y manejar posibles errores como índices fuera de rango o entradas no válidas.
- **Principios SOLID y buenas prácticas:** El código debe seguir los principios de diseño SOLID, garantizando modularidad, responsabilidad única y extensibilidad.
- **Correcto funcionamiento del sistema:** Se evaluará que el sistema cumpla con todas las funcionalidades descritas y que los resultados mostrados sean correctos.