TALLER CORTE 4

Elaborado por:

Javier Alejandro León Mendoza

Docente

Jhan Yuler de la Pava

UNIVERSIDAD EAN

FACULTAD DE INGENIERÍA

BOGOTÁ DC

2025

Contenido

DESCRIPCIÓN DEL CONTEXTO: El Restaurante y su Operación

El restaurante “Las Delicias de mi Tierra” es un negocio que atiende a clientes diariamente, ofreciendo una variedad de platillos y bebidas. Como en muchos restaurantes pequeños o medianos, la administración de pedidos, mesas y generación de recibo de pago se realiza manualmente o con procesos poco automatizados.

Funcionamiento actual del restaurante:

1. **Recepción de clientes**: Los clientes llegan al restaurante y el personal los acomoda en una mesa. Actualmente, este proceso es manual, y en horas pico puede haber confusión o falta de organización en la asignación de mesas.
2. **Toma de pedidos**: Los meseros anotan los pedidos en una libreta o en papel, lo que puede generar errores de interpretación o pérdida de la información.
3. **Envío a la cocina**: Los pedidos se llevan físicamente a la cocina, lo que en momentos de alta demanda puede generar retrasos o desorganización.
4. **Preparación y entrega de platos**: La cocina prepara los platos y los entrega a los meseros para ser llevados a las mesas. Sin un sistema organizado, pueden ocurrir errores como pedidos mezclados o tiempos de espera excesivos.
5. **Generación de recibo de pago**: Cuando el cliente solicita la cuenta, el mesero revisa los pedidos manualmente y calcula el total. Esto puede ser propenso a errores, además de hacer que el proceso sea más lento.

Factores que afectan la eficiencia del restaurante:

- **Altos tiempos de espera**: Cuando el restaurante está lleno, la toma de pedidos y su entrega pueden retrasarse, afectando la satisfacción del cliente.

- **Errores en los pedidos**: Si los meseros cometen errores al anotar un pedido, la cocina puede preparar platillos incorrectos, generando desperdicio de comida y aumentando costos.
- **Problemas en la administración de mesas**: En momentos de alta afluencia, puede haber mesas que se asignan de forma desordenada, lo que genera confusión en el personal y en los clientes.

BENEFICIOS DE IMPLEMENTAR UNA SOLUCIÓN

Desarrollar una aplicación en Java para gestionar el restaurante “Las delicias de mi tierra” traerá múltiples beneficios en términos de eficiencia operativa y satisfacción del cliente. A continuación, se detallan los principales beneficios:

Mayor precisión en la toma de pedidos

- Evita errores humanos: Los meseros podrán ingresar los pedidos directamente en el sistema, evitando confusiones con la letra escrita a mano o errores de comunicación con la cocina. Además, los meseros tendrán la oportunidad de agregar un nuevo producto a un pedido ya realizado, o en caso de algún error en la digitación de un producto, este podrá ser eliminado del pedido en tiempo real.
- Pedidos organizados y priorizados: El sistema enviará automáticamente los pedidos a la cocina con el detalle de cada platillo, optimizando la preparación.
- Tiempo de espera reducido: Al gestionar los pedidos de forma digital, se mejora el flujo de trabajo y los clientes reciben su comida más rápido.

- Mejor generación de recibos: Se realiza el cálculo automático y se muestra el recibo en pantalla mediante SWING

Mejor gestión de mesas y reservaciones

- Asignación eficiente de mesas: Se podrá visualizar en tiempo real qué mesas están ocupadas, y cuáles están disponibles con su respectiva capacidad (número de personas).
- Atención más rápida y organizada: Los meseros sabrán exactamente qué mesas atender y qué pedidos están en proceso.

- Asignación de pedidos eficientes: el menú contará con opciones para agregar o eliminar productos a un pedido en curso, y este será asociado al número de mesa que funcionará como un identificador único

Mejora en la generación de recibos de pago

- Cálculo automático de cuentas: El sistema calculará automáticamente el total de cada pedido, evitando errores humanos.

Experiencia mejorada para los clientes

- Servicio más rápido y organizado: Al reducir errores y tiempos de espera, los clientes tendrán una mejor experiencia en el restaurante.
- Posibilidad de pedidos en línea: En futuras versiones, se podría agregar la opción de pedidos para llevar o a domicilio, ampliando las oportunidades de venta.
- Mayor fidelización: Un servicio eficiente y confiable hará que los clientes regresen y recomienden el restaurante a otros.

Ventaja competitiva frente a otros restaurantes

- Diferenciación en el mercado: Un restaurante con tecnología avanzada y procesos optimizados se destaca frente a la competencia.

- Mejor reputación: Un servicio eficiente genera buenas reseñas en redes sociales y plataformas como Google Maps o TripAdvisor.
- Crecimiento del negocio: Un sistema bien implementado permite manejar más clientes sin comprometer la calidad del servicio.

PROPUESTA DE SOLUCIÓN

El objetivo con esta solución, es dotar de tabletas digitales a los meseros para que puedan tomar y gestionar los pedidos y las mesas del restaurante, en el caso de la toma de pedidos, la cocina contará con una pantalla en la que se deben visualizar los pedidos realizados con su respectiva mesa. Además, los meseros cuentan con opciones interactivas para gestionar las mesas de acuerdo a su ocupación, capacidad y respectivos pedidos, facilitando la gestión y optimizando los procesos del restaurante en cuanto a gestión de pedidos y de mesas. El sistema solo generará el recibo de pago para que el cliente pueda pagar en caja (Ya que la facturación hace parte de un módulo distinto al abordado en este documento)

Por lo anterior, se propone desarrollar una aplicación en Java que optimice los procesos clave del negocio. La solución incluirá funcionalidades específicas para la toma de pedidos, administración de mesas y generación de pedidos de pago.

1. Desarrollo de una Aplicación en Java

Se diseñará una aplicación que permitirá digitalizar y automatizar las operaciones del restaurante. La aplicación contará con una interfaz intuitiva para facilitar su uso por parte del personal.

2. Descripción del Funcionamiento de la Aplicación

La aplicación estará compuesta por varios módulos, cada uno de los cuales abordará una necesidad específica del restaurante.

- **Gestión de Pedidos:**
    - Los meseros podrán ingresar pedidos en una interfaz gráfica.
    - Los pedidos se enviarán automáticamente a la cocina.
    - Se permitirá la modificación o cancelación de pedidos (solo cuando el pedido tenga el estado “EN PREPARACION”).
    - Se podrá consultar los pedidos activos según la mesa.

- **Administración de Mesas y Reservaciones:**
    - Se visualizarán en tiempo real las mesas disponibles y ocupadas.
    - Las mesas serán visualizadas de acuerdo a su capacidad, por lo que los meseros tendrán la posibilidad de distribuir de manera correcta a los usuarios.

- **Generación de recibos de pago:**
    - Se generarán automáticamente los recibos de pago de cada pedido.

3. Funcionalidades Principales y Forma de Interacción

- **Usuarios de la Aplicación:**
    - **Meseros:** Ingresarán pedidos y verán el estado de cada mesa, también podrán visualizar sus asignaciones de mesa y el estado actual del pedido.

Cocina: Recibirá órdenes y actualizará el estado de preparación.

- **Interfaz de Usuario:**

- Se desarrollará una aplicación con una interfaz gráfica sencilla e intuitiva.

- **Tecnologías Utilizadas:**

Backend: Java para la gestión de datos y lógica de negocio.

REFLEXIÓN SOBRE PRINCIPIOS SOLID

Los principios SOLID contribuyen a mejorar la legibilidad, el mantenimiento y el testeo del código planteado, por tal razón, se tienen en cuenta los siguientes principios:

S - Responsabilidad Única: Cada clase en el proyecto tiene una única responsabilidad: Pedido gestiona productos y estado, Mesa su disponibilidad, Cocina los pedidos activos y las operaciones relacionadas con la gestión desde cocina, y GestorRestaurante la interacción. También se utilizan clases específicas para la implementación de interfaces (PedidoImpl, CocinaImpl)

O - Abierto/Cerrado: La clase abstracta Producto permite añadir nuevos tipos sin modificar el código existente.

L - Sustitución de Liskov: Las subclases Plato y Bebida respetan el contrato de Producto (como el método descripcion()), por lo que pueden sustituirse sin alterar el comportamiento del sistema

I - Segregación de Interfaces: Las interfaces Cocina y Pedido están bien definidas con métodos específicos y no obligan a las clases que las implementan a depender de métodos que no usan. Esto cumple con el principio de segregación de interfaces

D - Inversión de Dependencias: El sistema depende de interfaces (PedidoLogica, CocinaLogica) y no de implementaciones concretas, lo cual permite invertir la

dependencia. Además, la lectura/escritura de archivos está delegada a GestorArchivoProducto, separando la lógica de negocio de la de persistencia.

DISEÑO DE PERSISTENCIA EN ARCHIVOS

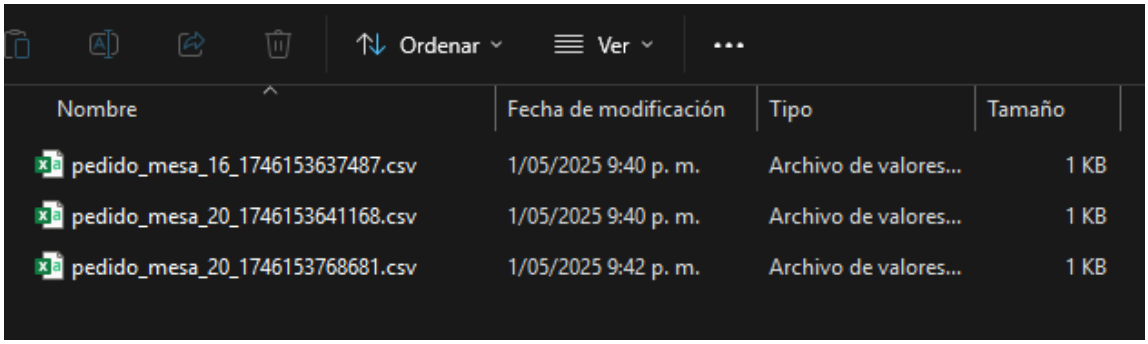
Para trabajar la persistencia, se implementa una lógica que permite leer un archivo CSV para recibir los datos, y se incluye la escritura para dejar trazabilidad de los pedidos creados. A continuación, se presenta la estructura tanto de lectura como de escritura que debe contener cada archivo:

- El menú se carga desde un archivo CSV data/menu.csv con cabecera y estructura: TIPO;NOMBRE;PRECIO;ALCOHOLICA.

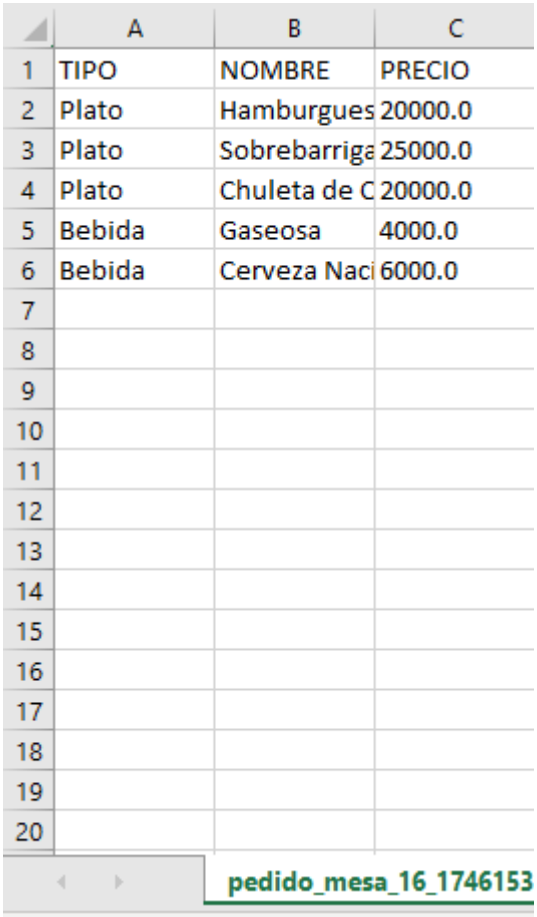
	A	B	C	D
1	TIPO	NOMBRE	PRECIO	ALCOHOLICA
2	Plato	Hamburgues	20000	
3	Plato	Pizza Person	8000	
4	Plato	MenÃº Corri	15000	
5	Plato	Sobrebarriga	25000	
6	Plato	Chuleta de C	20000	
7	Bebida	Gaseosa	4000	false
8	Bebida	Cerveza Naci	6000	true
9	Bebida	Cerveza Imp	10000	true

- Los pedidos se guardan en data/pedidos/pedido_mesa_X.csv con los productos solicitados.
- Las carpetas se crean automáticamente si no existen.
- Los recibos generados no sobrescriben archivos anteriores. Cada recibo se guarda con un nombre único que incluye la marca de tiempo del momento en que se genera

EJEMPLO DE CREACIÓN DE ARCHIVO:



Estos archivos generados contienen toda la información de los pedidos realizados, y se generan cuando el cliente desea conocer su recibo de pago. Como se observa en la siguiente imagen, el archivo de escritura se construye con la misma estructura del archivo de lectura (exceptuando el campo “ALCOHOLICA”), pero el nombre del CSV se genera con base en el número de la mesa que realizó el pedido y el tiempo en milisegundos de la generación del recibo para tener una trazabilidad más precisa:



	A	B	C
1	TIPO	NOMBRE	PRECIO
2	Plato	Hamburgues	20000.0
3	Plato	Sobrebarriga	25000.0
4	Plato	Chuleta de C	20000.0
5	Bebida	Gaseosa	4000.0
6	Bebida	Cerveza Naci	6000.0
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

pedido_mesa_16_1746153

También es importante destacar que:

- Meseros se cargan desde archivo meseros.txt
- Se conserva trazabilidad con marcas de tiempo
- Pedidos se guardan en objetos asociados a Mesa, y se serializan para preservar estado tras reinicio.

OBSERVACIONES EN LA IMPLEMENTACIÓN FINAL

- **Menú y meseros** ya no están codificados estáticamente. Se cargan desde archivos externos (menu.csv, meseros.txt). Esto permite a los administradores modificar la oferta sin alterar el código.
- **Persistencia de estado**: Al cerrar la aplicación, se guarda el estado completo de las mesas y meseros, incluyendo los pedidos en curso. Esto permite reiniciar sin pérdida de datos.
- **Recibo en pantalla**: Se genera un recibo visual en un JTextArea con formato tipo ticket, mostrando los productos y el total, al mismo tiempo que se guarda el CSV correspondiente.
- **Limpieza de vistas**: Al cambiar entre paneles (inicio, cocina, recibo, consulta), las vistas se limpian automáticamente para evitar residuos de datos anteriores.
- **Uso de clases anónimas y lambdas**: Se manejan eventos mediante lambdas (equivalentes modernos a clases anónimas), mejorando la legibilidad y simplicidad del código.
- **Diseño MVC** completo: Las vistas (PanelX), controladores (ControladorX) y modelos (Mesa, Pedido, etc.) están claramente separados, y se comunican por interfaces.

Con estas mejoras, la aplicación no solo resuelve el problema propuesto, sino que lo hace aplicando principios de arquitectura de software, orientación a objetos y buenas prácticas de usabilidad.