

Sistema de Gestión de Restaurante

Evaluación 2 – Programación II
Ignacio Loncón, Lukas Trillat
Universidad Católica de Temuco
Facultad de Ingeniería
15 de octubre de 2025

Índice

| | |
|---|----------|
| 1. Introducción | 2 |
| 2. Descripción de la solución | 3 |
| 2.1. Pestaña 1: Carga de ingredientes | 3 |
| 2.2. Pestaña 2: Stock | 3 |
| 2.3. Pestaña 3: Carta del restaurante | 3 |
| 2.4. Pestaña 4: Pedido | 3 |
| 2.5. Pestaña 5: Boleta | 3 |
| 3. Diagrama de clases | 4 |
| 4. Descripción de las clases principales | 5 |
| 5. Flujo del programa | 5 |
| 6. Resultados y conclusiones | 6 |

1. Introducción

El presente informe describe el desarrollo de una aplicación para la gestión de un restaurante, realizada como parte de la Evaluación 2 del ramo de Programación II. El objetivo principal del proyecto es diseñar e implementar una solución informática completa que facilite la administración de los recursos, pedidos y operaciones básicas del restaurante de forma eficiente y automatizada.

El sistema desarrollado busca optimizar el flujo de trabajo mediante una interfaz visual intuitiva y una estructura lógica robusta. A través del uso de Programación Orientada a Objetos (POO), se garantiza un diseño modular, escalable y fácil de mantener, donde cada componente cumple un rol claramente definido dentro del sistema.

Las principales funcionalidades implementadas en la aplicación son las siguientes:

- **Carga de ingredientes desde un archivo CSV:** permite inicializar el sistema con los datos base del inventario de forma rápida y automatizada.
- **Gestión del stock de ingredientes:** actualiza la cantidad disponible de cada insumo, evitando inconsistencias en los pedidos.
- **Generación de menús predefinidos:** crea opciones de platos según la disponibilidad actual de ingredientes.
- **Gestión de pedidos y emisión de boletas:** permite registrar pedidos, calcular totales y generar boletas detalladas en formato PDF.
- **Interfaz gráfica amigable:** ofrece una experiencia de usuario clara, ordenada y funcional.

En conjunto, esta aplicación representa una solución práctica que combina la teoría de la POO con el desarrollo de software aplicado a un contexto real, promoviendo buenas prácticas de diseño, fortaleciendo la comprensión de conceptos fundamentales en el desarrollo orientado a objetos en C#.

2. Descripción de la solución

La aplicación cuenta con una interfaz gráfica estructurada en cinco pestañas principales, cada una orientada a un módulo del sistema.

2.1. Pestaña 1: Carga de ingredientes

Permite cargar ingredientes desde un archivo CSV. Cuenta con un botón que incorpora los ingredientes al inventario.

2.2. Pestaña 2: Stock

Permite gestionar el inventario del restaurante. Los usuarios pueden añadir, modificar o eliminar ingredientes, y generar los menús predefinidos según la disponibilidad de ingredientes.

2.3. Pestaña 3: Carta del restaurante

Genera un archivo PDF con los menús disponibles en tiempo real. El PDF ofrece información como el nombre y precio de cada menú.

2.4. Pestaña 4: Pedido

Permite crear un pedido seleccionando los menús disponibles. El usuario puede definir cantidades, visualizar el total y generar la boleta en PDF. Este módulo actualiza automáticamente el stock tras cada pedido.

2.5. Pestaña 5: Boleta

Esta sección permite visualizar las boletas generadas para cada pedido. El usuario puede consultar información como los menús solicitados, las cantidades, los precios unitarios y el total del pedido.

3. Diagrama de clases

En la Figura se muestra el diagrama de clases de la aplicación.

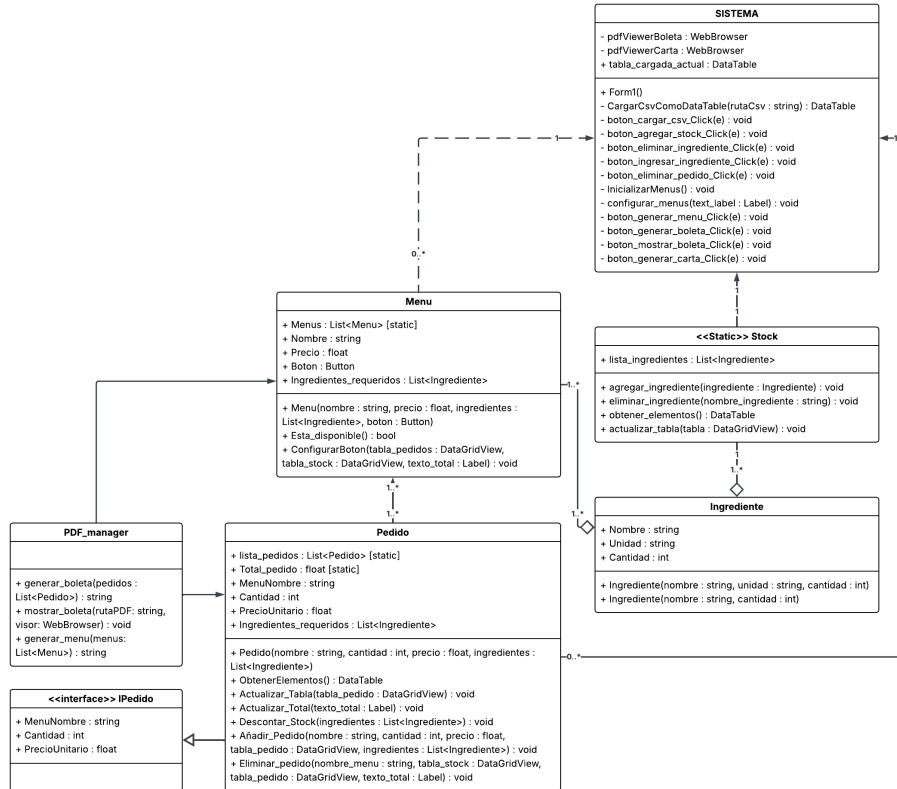


Figura 1: Diagrama UML de clases de la aplicación de restaurante.

4. Descripción de las clases principales

A continuación se presentan las clases con sus respectivas funcionalidades.

| Clase | Descripción |
|-------------|--|
| Ingrediente | Representa un ingrediente del restaurante, incluyendo su nombre, unidad de medida y cantidad disponible. Permite modelar de manera precisa los insumos del inventario. |
| Stock | Administra y controla el inventario de ingredientes, permitiendo agregar, eliminar o consultar los insumos disponibles en el sistema. |
| Menu | Define los menús preestablecidos del restaurante, relacionando cada menú con los ingredientes necesarios y su disponibilidad. Facilita la creación automática de menús según el stock. |
| Pedido | Gestiona los pedidos realizados por los clientes, registra los menús seleccionados y calcula los totales de manera automatizada, integrando la actualización del stock. |
| Sistema | Actúa como coordinador del sistema, integrando la interfaz gráfica con la lógica de las demás clases y asegurando la correcta interacción entre módulos. |
| PDF_Manager | Se encarga de generar los archivos PDF (boletas y cartas) y mostrarlos correctamente dentro de la aplicación. |

5. Flujo del programa

El flujo principal esperado del sistema se resume en los siguientes pasos:

1. Cargar archivo CSV con los ingredientes.
2. Agregar los ingredientes al stock.
3. Generar menús disponibles según el stock.
4. Visualizar la carta en formato PDF.
5. Crear pedidos y generar boletas en PDF.

6. Resultados y conclusiones

El sistema cumple con los requisitos funcionales solicitados, proporcionando una solución completa y eficiente para la gestión del restaurante. Entre sus capacidades, se destacan la carga de ingredientes desde archivos externos, la gestión dinámica del stock y la disponibilidad de menús, la generación correcta de archivos PDF para la carta y las boletas, y la implementación de principios básicos de la Programación Orientada a Objetos (POO), lo que asegura modularidad y facilidad de mantenimiento.

Como posibles mejoras futuras, se propone incorporar persistencia de datos mediante el uso de una base de datos, añadir autenticación de usuarios para mayor seguridad, modernizar la interfaz visual con librerías más actuales y optimizar la estructura del código para favorecer su escalabilidad y mantenimiento a largo plazo.

El código fuente completo del proyecto se encuentra disponible en el siguiente enlace:

<https://github.com/LukasTrillat/Proyecto-Progra-II>

El repositorio incluye todos los archivos de código fuente de la aplicación, escritos en CS, que contienen la implementación completa de la lógica del sistema. Estos archivos permiten ejecutar y probar todas las funcionalidades del proyecto, desde la gestión del inventario y los menús hasta la generación de pedidos y boletas en PDF.