

```
<!--Universidad Católica de Temuco-->
```

Proyecto Programación II{

```
<Por="Paulo Villalobos & Denys Rodríguez"/>
```

```
<Fecha="15/10/2025"/>
```

```
<Docente="Guido Mellado"/>
```

}



Contenidos

- 01 Introducción
- 02 Estructura
- 03 Problemática
- 04 Solución
- 05 Diagrama de Clase
- 06 Flujo General
- 07 Carga de Datos
- 08 Control de Stock
- 09 Carta Restaurante
- 10 Pedido y Facturación
- 11 Botones
- 12 Decisiones Técnicas
- 13 Validación de Errores
- 14 Logros Claves
- 15 Conclusiones

Introducción {

En este proyecto desarrollamos y optimizamos una aplicación de escritorio usando de base Python y Custom Tkinter como interfaz. que gestione el flujo completo de un restaurante desde el stock de ingredientes hasta la generación de pedidos y boletas garantizando un código limpio, funcional y mantenible

- Estructuración.
- Revisión de errores.
- Corrección.
- Asignación del equipo.
- Formateo.
- Limpieza.
- Exportación.
- Depuración.

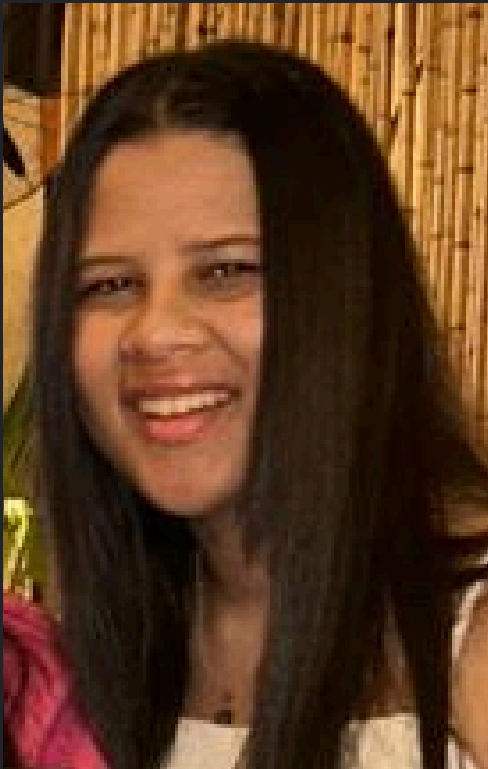
}

Equipo {



PAULO
VILLALOBOS

FUNCIONAMIENTO
STOCK Y PEDIDO



DENYS
RODRÍGUEZ

MANEJO DE ARCHIVOS



Problematica {

El problema surge de la necesidad de un restaurante de contar con una aplicación que facilite la *gestión completa de su funcionamiento interno*, desde el manejo del stock hasta la generación de boletas. La idea es crear una herramienta que permita *cargar ingredientes, ya sea manualmente o desde un archivo, para mantener actualizado el inventario. A partir de este stock, la aplicación debe **verificar qué menús pueden prepararse* y mostrarlos al usuario, evitando ofrecer platos cuyos ingredientes no estén disponibles.

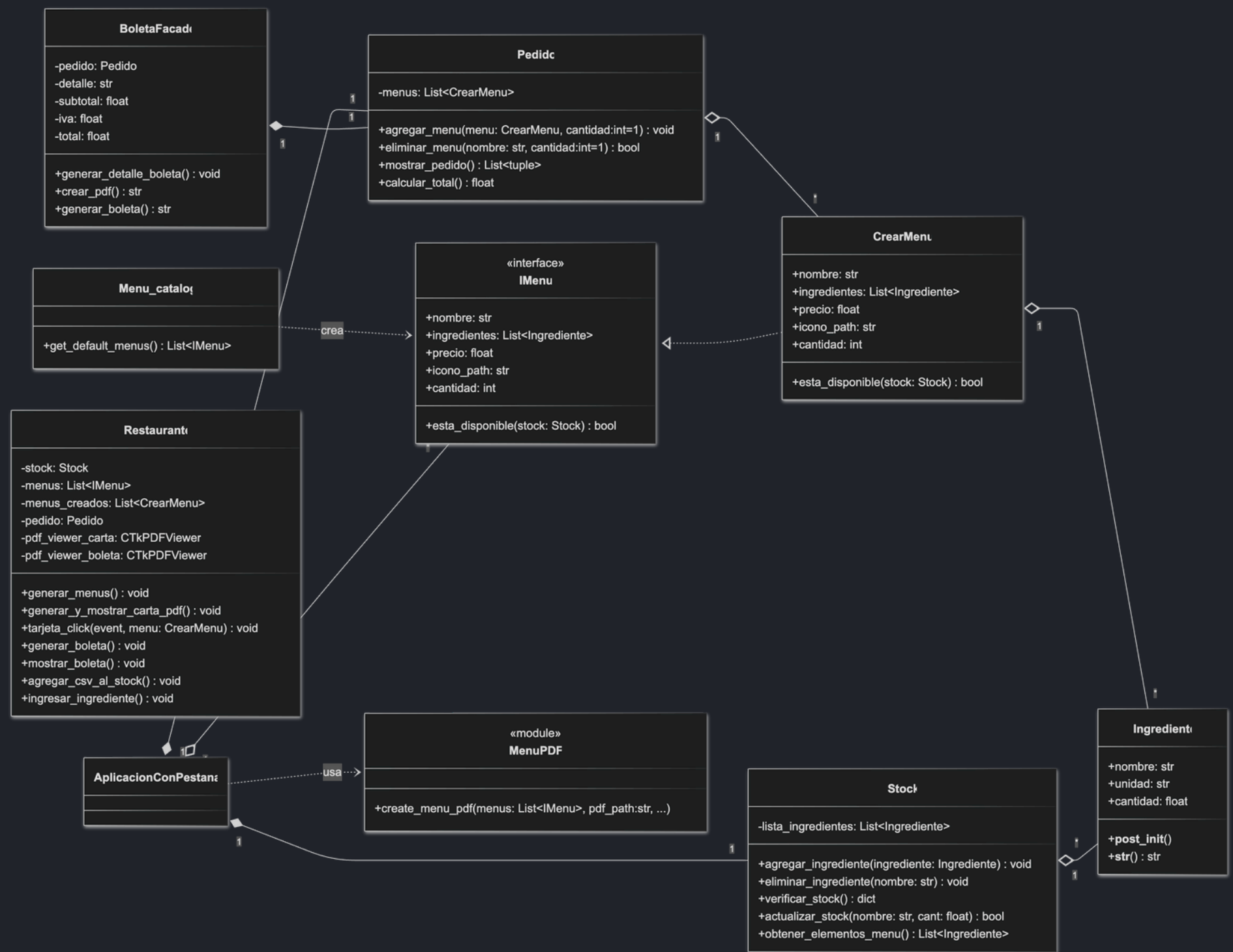
}

Solución {

Crear una aplicación de escritorio que permita cargar archivos CSV o Excel con el Stock recibido, y también agregar Stock manualmente, validando nombre, unidad y cantidad en ambos casos. Con el Stock disponible se crean los Menús, desde los cuales se puede generar la Carta en PDF. En la sección de Pedidos, se pueden agregar menús al pedido usando el Stock disponible, eliminar menús seleccionados y actualizar las cantidades automáticamente. Finalmente, se puede generar una Boleta, que confirma el pedido, descuenta el Stock usado y muestra los menús comprados, el subtotal, el IVA y el total.

}

Diagrama de Clases {



}

Explicación {

Primero, el usuario interactúa con la interfaz principal `AplicacionConPestanas`, que gestiona las funciones del sistema a través de la clase `Restaurant`. Desde ahí se pueden cargar ingredientes manualmente o desde un archivo CSV, actualizando el `Stock`, que mantiene la lista de `Ingrediente` y permite agregarlos, eliminarlos o ajustar sus cantidades.

Después, la aplicación genera los menús usando el catálogo base de `Menu_catalog`. Cada `CrearMenu` revisa su disponibilidad con `esta_disponible`, comparando los ingredientes necesarios con el `Stock`. Solo los menús con todos los ingredientes disponibles se incluyen en la carta, que se genera en PDF mediante `MenuPDF` y se muestra con `CTkPDFViewer`.

Cuando un cliente hace un pedido, `Pedido` guarda los menús seleccionados, controla cantidades y calcula subtotales y totales. Este consume `Stock`, que puede ser devuelto con `EliminarPedido`. Luego, `BoletaFacade` usa esta información para generar la boleta en PDF con detalle, subtotal, IVA y total, quitando de forma definitiva el `Stock` usado. En todo momento, el `Stock` es el centro del sistema: cualquier cambio en los ingredientes afecta directamente los menús disponibles, la carta y los pedidos, manteniendo todo actualizado y coherente.

}

Estructura

Ingrediente

Es el elemento más básico. Define el nombre, la unidad (kg/unid) y la cantidad. Asegura que la cantidad sea siempre un número decimal.

IMenu / CrearMenu

Define cómo debe ser cualquier menú (nombre, precio, etc.) y contiene la regla de venta: solo se ofrece si hay suficiente stock de todos sus ingredientes.

Pedido

Es la cesta de compra del cliente. Gestiona la lista de menús solicitados y, si se cancela un menú, devuelve automáticamente los ingredientes al Stock.

Stock

Es el registro oficial del inventario. Almacena todos los ingredientes disponibles. Impide mezclar unidades para un mismo ingrediente (ej. no permite "Manzana" en kg y unid a la vez).

AplicacionConPestanas

La Interfaz Gráfica (GUI) con cinco secciones (Carga, Stock, Carta, Pedido, Boleta) construida con CustomTkinter.

BoletaFacade

Automatiza la factura. Calcula subtotales, aplica el IVA (19%) y genera el documento final (boleta.pdf).

}

Flujo General {

Primero se cargan ingredientes (CSV/Excel o manual), el Stock se actualiza y queda listo para validar disponibilidad. Luego, la app genera menús válidos según stock; esos menús se publican en la Carta (PDF) y se visualizan con CTKPDFViewer. Cuando un cliente pide, Pedido registra menús y cantidades, actualiza totales y Stock, al finalizar, BoletaFacade genera la boleta en PDF con subtotal, IVA y total. Todo reacciona al Stock: si cambia, cambian los menús disponibles y la carta.

}

Carga de Datos {

01. Botón "Cargar CSV o Excel"

Función:

Abre un diálogo para seleccionar el archivo fuente (.csv o .xlsx).

Lógica:

El sistema usa Pandas para leer el archivo de forma eficiente.

Se intenta decodificar (UTF-8 → latin-1 si falla) y se limpian los encabezados de las columnas (ej. se eliminan espacios y se pasa a minúsculas).

02. Treeview (Vista Previa)

Función:

Muestra la data del archivo antes de la importación.

Lógica:

Se revisa que existan las columnas críticas: nombre, unidad y cantidad. Si falta alguna, se alerta inmediatamente.

03. Botón "Agregar al Stock"

Función:

Ejecuta la importación final.

Lógica:

1. Transformación: Cada fila válida se convierte en un objeto Ingrediente.

2. Sincronización: Se llama a `Stock.agregar_ingrediente()`, que suma cantidades si el ingrediente ya existe (mismo nombre + unidad) o lanza una advertencia si la unidad es inconsistente.

Carga de Datos {

04. Mensajería Visual

Función:

Notificaciones de éxito o error (CTkMessageBox).

Lógica:

Informa sobre errores críticos (ej. "Columnas Faltantes", "Unidad Inválida") o el éxito de la carga.
Exportar a Hojas de cálculo

}

Control de Stock {

01. Formulario (Ingreso Manual)

Función:

Permite dar alta, baja o modificar ingredientes específicos.

Lógica:

Se verifica que el Nombre sea solo letras, la Cantidad sea numérica y positiva, y la Unidad sea kg o unid. 2.
Sincronización: Llama a `Stock.agregar_ingredientes()`, que suma si el ítem ya existe con la misma unidad o bloquea si la unidad es diferente.

02. Treeview (Inventario)

Función:

Refleja fielmente el estado interno del Stock.

Lógica:

La tabla se repuebla (`actualizar_treeview()`) en tiempo real con cada acción de ingreso o eliminación, garantizando que lo que se ve es el estado real.

03. Botón "Generar Menú"

Función:

Determina qué se puede vender en este momento.

Lógica:

Recorre la lista de menús y ejecuta `menu.esta_disponible(stock)`. Solo los menús que cumplen con los requisitos de cantidad y unidad son marcados como viables para la venta.

}

Carta Restaurante {

Qué hace?

Esta pestaña transforma el estado actual del inventario en un documento de venta dinámico y profesional (carta.pdf). Su objetivo es garantizar que la carta solo muestre menús disponibles, proporcionando una fuente de verdad al cliente final.

01. Botón "Generar Carta (PDF)"

Función:

Inicia el proceso de filtrado de stock y creación del documento.

Lógica:

1. Filtro de Disponibilidad: El sistema recorre todos los menús y usa la función `m.esta_disponible(self.stock)` para filtrar solo los viables.

2. Generación con FPDF: Llama a la librería `menu_pdf.create_menu_pdf()`, que recibe los menús filtrados y crea el PDF.

02. Visor de PDF

Función:

Muestra el archivo `carta.pdf` dentro de la aplicación (Visor CTKPDFViewer).

Lógica:

Feedback Inmediato: Una vez generado, el PDF se carga al instante, permitiendo al usuario revisar la carta sin abrir programas externos.
Exportar a Hojas de cálculo

}

Pedidos y Facturación {

Selección del Menú (Clic en Tarjeta)

Este módulo representa el cierre del flujo: gestiona la selección final del cliente, sincroniza el inventario en la venta y genera el documento fiscal final, asegurando la consistencia entre el stock físico y las transacciones.

Resultado y Mecanismo:

1. Verificación:
`menu.esta_disponible(stock)`
revisa el inventario.

2. Actualización: Si hay stock, se descuenta la cantidad de ingredientes del Stock y se añade el menú al objeto Pedido.

Regla:

La tabla Treeview y el Total se actualizan dinámicamente. "Sin Stock → Alerta"

Eliminar Menú

Resultado y Mecanismo:

El sistema devuelve al Stock la cantidad exacta de ingredientes que ese menú había consumido.

Regla:

Esta acción garantiza que el inventario siempre refleje la realidad si se cancela o modifica parte de la orden.

Pedidos y Facturación {

El patrón BoletaFacade es el orquestador que centraliza los cálculos y la documentación, sin depender de la interfaz gráfica (GUI).

Flujo de Facturación

Botón "Generar Boleta": Se verifica que el Pedido no esté vacío.

Cálculo (BoletaFacade): La Fachada ejecuta:

- Subtotal (Suma de ítems).
- IVA (19%) del subtotal.
- Total (Subtotal + IVA).

Generación de PDF:

facade.crear_pdf() genera el archivo boleta.pdf, con encabezado, tabla de ítems y totales detallados, listo para impresión.

Cierre: El Pedido actual se limpia, preparando el sistema para la siguiente transacción.

Diseño Técnico

- **BoletaFacade:** Centraliza el cálculo de impuestos y el formato del documento. Esto facilita futuras modificaciones del IVA o la inclusión de descuentos sin tocar la lógica de la interfaz.
- **Visor PDF (CTkPDFViewer):** Permite visualizar la boleta inmediatamente en la pestaña "Boleta", sin necesidad de software externo.

}

Botones {

01. Gestión de Stock

Logra:

Permite carga masiva de CSV o EXCEL o manual. Valida nombres, unidades y cantidades, evitando errores o duplicados antes de agregarlos

Valor agregado:

Inventario confiable y totalmente consistente, con actualización en tiempo real

02. Generación de Carta

Logra:

Crea la carta.pdf dinámica. Solo muestra los menús que tienen stock suficiente para ser separados

Valor agregado:

El documento de venta siempre refleja la capacidad operativa real.

03. Pedidos y Boletas

Logra:

Descuenta ingredientes reales al vender y los restituye al cancelar. Calcula subtotal, IVA (19%) y total con precisión.

Valor agregado:

Transacción Precisa. Genera boleta.pdf automáticamente, lista para impresión, eliminando cálculos manuales.

}

Decisiones Técnicas {

Herramienta	Razón de elección	Beneficios Clave
CustomTkinter	Moderniza la interfaz clásica de Tkinter. Ligero e ideal para prototipos funcionales sin licencias complejas (a diferencia de Qt).	Maneja fallos de codificación (latin-1 fallback). Alerta si faltan columnas, pero no colapsa.
Pandas	Simplifica la lectura y validación masiva de archivos CSV/Excel. Ofrece robustez y limpieza de datos superiores al manejo nativo.	Importación de datos rápida y segura.
FPDF	Librería minimalista que da control total sobre el diseño del PDF (layout, tipografía). Liviana y sin dependencias gráficas.	PDFs limpios, estéticos y controlados para la Carta y la Boleta.

}

Validación de Errores {

Punto de Control

Validación

Manejo de Error

Carga de Archivo

Columnas obligatorias (nombre, unidad, cantidad) y extensión correcta (.csv, .xlsx).

Maneja fallos de codificación (latin-1 fallback). Alerta si faltan columnas, pero no colapsa.

Gestión de Stock

Nombre: Uso de regex (solo letras) para evitar símbolos. Cantidad: Debe ser numérica y positiva. Unidad: Debe ser 'kg' o 'unid'.

Bloquea entradas con formato incorrecto. Si hay duplicidad con unidad distinta, lanza una alerta específica: "La unidad de [Nombre] es 'unid', no 'kg'." Exportar a Hojas de cálculo

}

Logros Claves {



AUTOMATIZACIÓN

Cierre total del ciclo: Desde la Carga de datos hasta la Boleta PDF. Cada cambio en el Stock afecta inmediatamente la Carta y el Pedido.



ROBUSTEZ Y CONTROL

Validaciones exhaustivas en cada paso. El sistema no colapsa ante errores de formato o falta de stock, garantizando la integridad del inventario.



DISEÑO MODULAR

Estructura en capas (UI / Lógica / Datos) con Patrón Facade (BoletaFacade). Los componentes son independientes y listos para escalar.



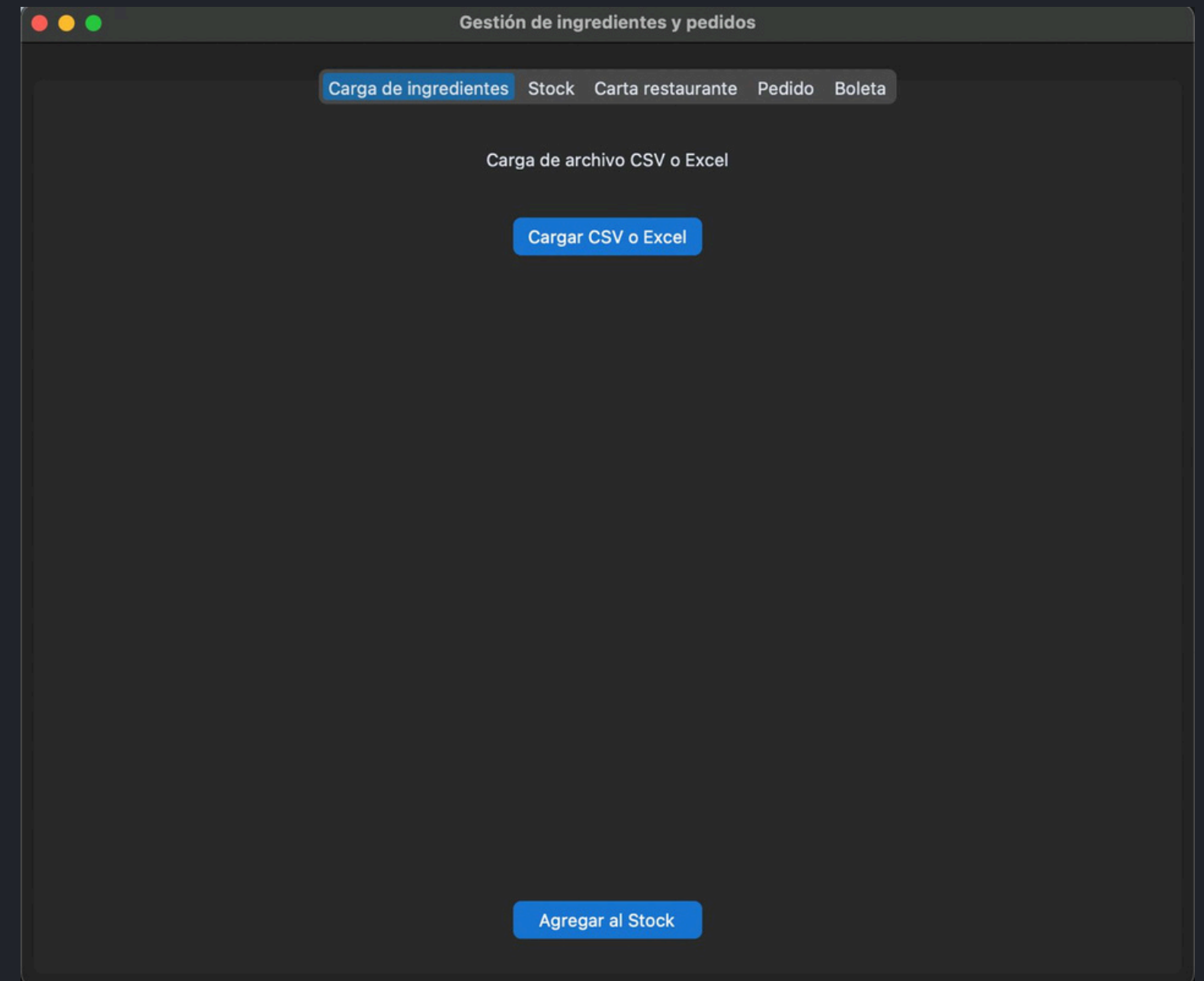
DOCUMENTACIÓN

Carta y Boleta generadas automáticamente en formato PDF profesional, listas para impresión o envío. Exportar a Hojas de cálculo

}

Conclusiones {

El proyecto ha entregado una solución, la cual cumple íntegramente con la problemática entregada, de gestionar el flujo completo de un restaurante, creando una herramienta semi-automatizada, coherente y robusta.



}

```
<!--Universidad Católica de Temuco-->
```

Gracias {

```
<Por="Paulo Villalobos & Denys Rodríguez"/>
```

}