



## ProyectoP2

Axel Gonzales, Juan Carrera Y Cristian Muñoz

October 15, 2025

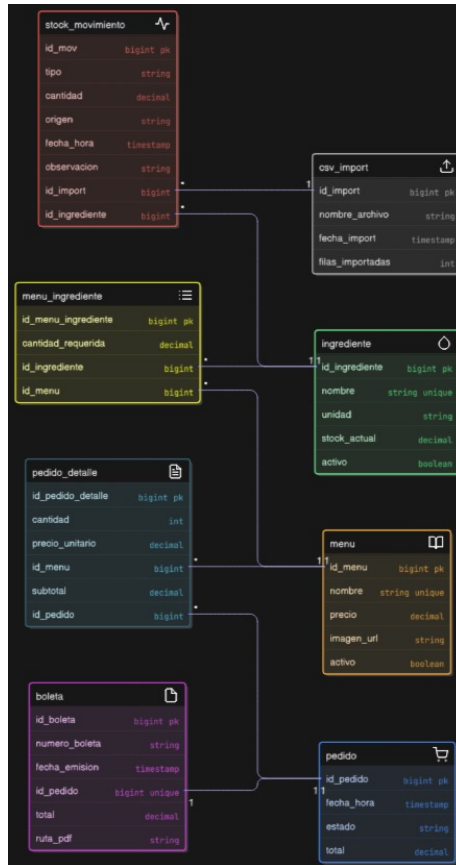
### 1 Introducción

En el siguiente informe se enseñara como realizamos una aplicación de escritorio completa con interfaz gráfica utilizando la librería CustomTkinter.

#### 1.1 Problemática

En muchos negocios gastronómicos pequeños, la administración de ingredientes y pedidos se realiza de forma manual, lo que genera errores frecuentes en el control del stock, retrasos en los pedidos y dificultades para mantener registros actualizados. Esta falta de organización provoca pérdidas económicas y disminuye la eficiencia del servicio. Ante esta situación, surge la necesidad de desarrollar una aplicación que automatice la gestión de ingredientes y pedidos, permitiendo un control más preciso, rápido y accesible de la información.

## 1.2 Diagrama de clases



## 2 Propósito general del código

El código crea una aplicación con una interfaz gráfica lo que nos permite esta aplicación es:

Importar ingredientes desde un archivo csv

Cargar y gestionar ingredientes en un stok

Crear y mostrar menús disponibles

Realizar pedidos seleccionando menús, controlando el stock.

Generar y visualizar PDF con la carta y la boleta.

En pocas palabras es una aplicación pensada para gestionar un restaurante.

### 3 Estructura General

El código está organizado por una clase principal:

```
class AplicacionConPestanas .
```

Esta clase controla toda la lógica de la aplicación: interfaz, carga de datos, validaciones y acciones. Dentro de la clase se definen muchos métodos que separan las funciones por secciones.

### 4 Interfaz Gráfica

en el método `__init__` es donde se genera la ventana principal

```
self.title("Gestión de ingredientes y pedidos")  
self.geometry("870x700")
```

y las pestañas se crean con:

```
self.tabview = ctk.CTkTabview(self, command=self.on_tab_change)
```

Esto permite una interfaz con varias secciones independientes:

Carga de ingredientes (CSV)

Stock

Carta del restaurante

Pedido

Boleta

Cada pestaña se configura con un método propio (`configurar_pestana1`, `configurar_pestana2`, etc.).

### 5 Cargar archivo CSV

En la pestaña “Carga de ingredientes”, se puede importar un archivo CSV con columnas nombre, unidad y cantidad.

```
def cargar_csv
```

Usa `filedialog.askopenfilename` para abrir un cuadro de diálogo y seleccionar el archivo. Luego lo lee con Pandas:

```
df = pd.read_csv(file_path)
```

y muestra visualmente con un Treeview (una tabla). Después, al presionar “Agregar al Stock”, los ingredientes del CSV se agregan al objeto Stock.

una de las buenas prácticas que podemos observar es que se validan los nombres de columnas antes de procesar el archivo y se manejan errores con `try/except`

## 6 Gestion Stock

La pestaña stock cuenta con un formulario para ingresar ingredientes manualmente los campos a reynar serian:

Nombre

Unidad(Kg o unidad)

cantidad

con el botón "ingresar ingredientes" se llama a la función ingresar-ingredientes() que valida los datos

Que el nombre no esta vacío y contenga solo letras (validar-nombre)

Que la cantidad sea numérica y positiva (validar-cantidad)

luego crea el ingrediente ingresado y lo agrega al stock

También puede eliminar ingredientes seleccionados

## 7 Creacion y validacion de la carta

En la pestaña "Carta restaurante" se genera un PDF con los menús disponibles:

```
def generar_y_mostrar_carta_pdf
```

Para hacer esto se utiliza una función externa llamada create-menu-PDF() que crea el archivo carta.pdf y luego lo muestra con **CTkPDFViewer**

## 8 Gestión de pedidos

En la pestaña pedido es donde se muestra cada menú disponible, cada menú tiene una imagen, un nombre y se le puede dar click para agregar el pedido

```
def tarjeta_click
```

Esto verifica si hay suficiente stock de ingredientes antes de permitir el pedido, si alcanza descuenta los ingredientes del stock y agrega el menú al pedido.(self.pedido.agregar-menu(menu))

El pedido se muestra en una tabla y se actualiza el total de pedido, tambien permite eliminar pedidos y generar el PDF de la boleta.

## 9 Generacion de boleta

Para generar una voleta se utiliza la clase Boletafacade, que es la que se encarga de crear el PDF con el menu de pedidos. Luego se puede visualiza dentro del

programa usando el mismo **CTkPDFViewer**.

## 10 Manejo de pestañas

Con el método **on-tab-changes** detecta cuándo el usuario cambia de pestaña y actualiza la información correspondiente.

```
if selectedtab == "Stock" :  
    self.actualizartreeview()
```

Esto mantiene sincronizado la interfaz con los datos del sistema.

## 11 Diseno y Estilo

para la apariencia se utiliza el siguiente codigo:

```
ctk.setappearance_mode("Dark")  
ctk.setdefault_color_theme("blue")
```

## 12 Flujo del programa

Con todo lo anterior visto en el informe podemos decir que el Flujo de nuestro programa es el siguiente:

**El usuario abre el programa.**

**Carga ingredientes manualmente o desde CSV.**

**Consulta el stock disponible.**

**Ve los menús disponibles (Carta).**

**Hace un pedido (se descuenta del stock).**

**Genera y visualiza la boleta en PDF.**

## 13 Conclusión

Como se puede apreciar nuestra aplicación funciona correctamente para ayudar en la administración de un negocio, además que el código de la aplicación se encuentra bien estructurado y es simple y fácil de entender permitiendo que cualquier usuario pueda utilizarlo sin mucha dificultad