**Plataforma de Gestión de Pedidos de Delivery**

**Introducción**

Esta plataforma está diseñada para la gestión eficiente de pedidos de delivery. Utiliza Python con las bibliotecas tkinter para la interfaz de usuario y sqlite3 para la administración de la base de datos.

**Características principales**

1. **Gestor de Pedidos:** Permite agregar, actualizar, eliminar y visualizar pedidos.
2. **Interfaz gráfica:** Diseñada con tkinter para una experiencia práctica e intuitiva.
3. **Base de datos integrada:** Utiliza SQLite para almacenar y gestionar los datos de los pedidos.
4. **Validaciones:** Implementa validaciones para los campos del formulario.

**Estructura del Código**

**Declaración de colores**

Los colores se definen para mantener una paleta consistente en la interfaz:

* bg\_color: Fondo general de la aplicación.
* title\_bg\_color: Fondo del título principal.
* button\_bg\_color: Fondo de los botones.
* button\_del\_bg\_color: Fondo del botón de eliminación.
* Otros colores son utilizados para campos de entrada y encabezados del Treeview.

**Conexión a la Base de Datos**

La función create\_db establece la conexión con SQLite y crea una base de datos llamada orders.db.

**Creación de la Tabla**

La función orders\_table verifica si existe una tabla llamada orders y la crea si no está presente.

Estructura de la tabla:

* id: Identificador único (clave primaria).
* nombre: Nombre del cliente.
* telefono: Teléfono del cliente.
* direccion: Dirección de entrega.
* total: Monto total del pedido.
* pedido: Descripción del pedido.
* fecha: Fecha del pedido.

**Validaciones**

Se incluyen validaciones para los campos:

* validate\_name: Verifica que el nombre contenga solo letras.
* validate\_phone: Permite hasta 10 dígitos en el teléfono.
* validate\_address: Asegura que la dirección no esté vacía.

**Formulario de Ingreso**

El formulario incluye campos para:

1. Nombre del cliente.
2. Teléfono.
3. Dirección.
4. Monto total.
5. Pedido.
6. Fecha (seleccionable a través de un calendario).

**Tabla de Visualización**

Se utiliza un Treeview para mostrar los pedidos almacenados. Incluye:

* Estilos personalizados para el encabezado y las filas.
* Opciones para seleccionar y manipular los registros directamente desde la tabla.

**Botones y Funciones**

1. **Guardar:** Agrega un nuevo pedido a la base de datos.
2. **Eliminar:** Elimina un pedido seleccionado.
3. **Actualizar:** Modifica un pedido existente.

**Código de los botones**

* add\_order: Valida los datos y guarda un nuevo pedido.
* delete\_order: Elimina el pedido seleccionado.
* update\_order: Actualiza los detalles del pedido seleccionado.

**Instalación y Configuración**

1. Instalar Python (versión 3.7 o superior).
2. Instalar las dependencias:

pip install tkcalendar

1. Ejecutar el código:

python gestion\_pedidos.py

**Uso de la Plataforma**

1. Ingrese los detalles del pedido en el formulario.
2. Seleccione una fecha usando el calendario.
3. Presione **Guardar** para registrar el pedido.
4. Para actualizar o eliminar, seleccione un pedido de la tabla y use los botones correspondientes.

**Mejoras Futuras**

1. Implementar la exportación de datos a formatos como CSV o Excel.
2. Incluir búsqueda avanzada por cliente, por pedido o fecha.
3. Agregar un sistema de inicio de sesión para usuarios con distintas jerarquías.
4. Mejorar el diseño gráfico.
5. Implementar una sección de historial de pedidos realizados.
6. Contador de pedidos recibidos en el día.
7. Posible idea de dashboard donde poder visualizar estadísticas relevantes para el usuario.

**Notas Finales**

La plataforma está diseñada para ser escalable y adaptarse a las necesidades de pequeños negocios que gestionan pedidos de delivery.