

## **MANUAL TÉCNICO**

**Proyecto:** Sistema de Gestión de Pedidos

**Desarrollado por:** Francisco Simón Sanmiguel Royo

**Lenguaje:** Python 3.10 con Tkinter

**IDE utilizado:** Thonny

## **1. Introducción**

Este manual técnico describe la estructura, funcionamiento, y componentes principales del sistema de gestión de pedidos desarrollado en Python. El sistema permite registrar, visualizar y modificar el estado de pedidos en un entorno local

## MANUAL TÉCNICO

**Proyecto:** Sistema de Gestión de Pedidos

**Desarrollado por:** Francisco Simón Sanmiguel Royo

**Lenguaje:** Python 3.10 con Tkinter

**IDE utilizado:** Thonny

## 2. Estructura del Proyecto

Proyecto-de-software GestionPedidos.py <a href="#">Prueba.py</a> Manual Tecnico.pdf README.md	# Código fuente principal # Script opcional de prueba # Este documento # Descripción general del proyecto
---	--

## **MANUAL TÉCNICO**

**Proyecto:** Sistema de Gestión de Pedidos

**Desarrollado por:** Francisco Simón Sanmiguel Royo

**Lenguaje:** Python 3.10 con Tkinter

**IDE utilizado:** Thonny

### **3. Dependencias**

- Python 3.10+
- Tkinter (viene integrado en Python)
- Thonny IDE (para ejecución y pruebas)
- PyInstaller (opcional para empaquetado)

### **4. Componentes Principales**

## MANUAL TÉCNICO

**Proyecto:** Sistema de Gestión de Pedidos

**Desarrollado por:** Francisco Simón Sanmiguel Royo

**Lenguaje:** Python 3.10 con Tkinter

**IDE utilizado:** Thonny

### 4.1 Clase Pedido

Modelo de datos. Cada objeto representa un pedido con los atributos:

- producto: nombre del producto
- cantidad: unidades solicitadas
- estado: estado actual del pedido (Pendiente, En preparación, Entregado, Cancelado)

Métodos:

- cambiar\_estado(nuevo\_estado)

### 4.2 Clase SistemaPedidos

Contiene la lógica de negocio:

- Lista interna de pedidos
- Métodos:
  - agregar\_pedido(producto, cantidad)
  - obtener\_pedidos()
  - resumen\_por\_estado()

### 4.3 Clase App

Encapsula toda la interfaz gráfica con Tkinter:

- Entradas: producto y cantidad
- Tabla para mostrar pedidos
- Combo para cambiar estado
- Botones para agregar, cambiar estado y mostrar resumen

## 5. Ejecución

## **MANUAL TÉCNICO**

**Proyecto:** Sistema de Gestión de Pedidos

**Desarrollado por:** Francisco Simón Sanmiguel Royo

**Lenguaje:** Python 3.10 con Tkinter

**IDE utilizado:** Thonny

Desde Thonny:

1. Abrir el archivo GestionPedidos.py
2. Ejecutar el script
3. Usar la interfaz para probar las funcionalidades

## MANUAL TÉCNICO

**Proyecto:** Sistema de Gestión de Pedidos

**Desarrollado por:** Francisco Simón Sanmiguel Royo

**Lenguaje:** Python 3.10 con Tkinter

**IDE utilizado:** Thonny

## 6. Pruebas Básicas

El sistema fue verificado mediante ejecución directa en consola de Thonny. Además, se incluyó un script opcional Prueba .py con 5 casos de prueba:

- Crear pedido
- Cambiar estado válido e inválido
- Agregar pedidos al sistema
- Verificar resumen por estado

## MANUAL TÉCNICO

**Proyecto:** Sistema de Gestión de Pedidos

**Desarrollado por:** Francisco Simón Sanmiguel Royo

**Lenguaje:** Python 3.10 con Tkinter

**IDE utilizado:** Thonny

## 7. Compilación a Ejecutable (opcional)

```
import PyInstaller.__main__

PyInstaller.__main__.run([
    '--onefile',
    '--noconsole',
    'GestionPedidos.py'
])
```

El ejecutable aparecerá en **dist/GestionPedidos.exe** (Windows).

## **MANUAL TÉCNICO**

**Proyecto:** Sistema de Gestión de Pedidos

**Desarrollado por:** Francisco Simón Sanmiguel Royo

**Lenguaje:** Python 3.10 con Tkinter

**IDE utilizado:** Thonny

## **8. Recomendaciones Finales**

- No cerrar Thonny si estás haciendo pruebas en vivo.
- Asegúrate de tener todos los archivos en la misma carpeta.

**Fin del manual técnico.**