

Proyecto RestaurantTickets

Carlos E. Soto Pérez, Diego A. Sanabria Ascanio, Juan D. Jaimes García y Kevin A.

Mendoza Muñoz

Ingeniería en Software, Facultad de Ingeniería, Tecnológico de Antioquia Institución

Universitaria

PREPRO25014020001: Introducción al Desarrollo de Software

Docente: Christian Jaimes

12 de mayo, 2025

Introducción

Esta aplicación facilita la gestión de pedidos de comida mexicana de forma sencilla y rápida. Pensada para un pequeño negocio familiar, permite tomar pedidos, aplicar descuentos automáticos, llevar el control de la fila de preparación y generar estadísticas diarias del negocio.

Objetivos

- Permitir al usuario registrar pedidos.
- Calcular automáticamente subtotales, descuentos y tiempos estimados de preparación.
- Mantener en memoria la cola de pedidos y mostrar cuántos minutos faltan para atender cada uno.
- Generar estadísticas diarias de ventas y recaudación sin salir de la consola.
- Ofrecer comandos simples para reiniciar datos o cancelar pedidos en cualquier momento.

Alcance funcional

Este proyecto de consola en Python gestiona pedidos de combos de un restaurante de comida mexicana con las siguientes características:

Configuración de ambiente y zona horaria

Se usa la librería zoneinfo para establecer la zona horaria (por defecto America/Bogota) al registrar marcas de tiempo.

Definición de productos (combos)

Los combos disponibles se declaran de forma inmutable en el diccionario COMBOS, con clave de identificador entero y valores:

- nombre (string)
- precio (float)
- tiempo de preparación (int, en minutos)

Se incluyen:

1. { "nombre": "Combo de Tacos al Pastor", "precio": 42000.0, "tiempo": 15 }
2. { "nombre": "Combo de Tacos de Birria", "precio": 42000.0, "tiempo": 20 }
3. { "nombre": "Combo de Quesadillas", "precio": 35000.0, "tiempo": 10 }

Descuentos

Aquí se almacena la configuración de descuento en DESCUENTO, con los campos:

- activo (bool)
- tope_minimo (float): umbral de subtotal para aplicar descuento
- porcentaje (float): porcentaje de descuento sobre el subtotal

Queda definido como:

DESCUENTO = {

"activo": True,

"tope_minimo": 100000.0,

```
"porcentaje": 0.10  
}
```

Registro y gestión de pedidos

Al registrar un pedido:

- Se solicita al usuario seleccionar combos y cantidades, validando rangos numéricos.
- Se permite cancelar en cualquier paso con la tecla c y finalizar selección con f.
- Se calcula el subtotal, el descuento (si aplica), el total y el tiempo_estimado (suma de tiempos de cada combo × cantidad).
- Se registra la marca de tiempo con la hora actual de Colombia.
- El pedido se almacena en el diccionario pedidos y su ID se añade a la lista fila.
- Se actualizan las estadísticas diarias por combo y los totales.
- Se imprime un ticket detallado en consola.

Cola de preparación (fila)

Esta lista almacena los IDs de pedidos en orden de entrada.

Al mostrar la fila, se itera sumando el tiempo_estimado de cada pedido previo para calcular el tiempo restante acumulado de cada uno.

Estadísticas diarias

La estructura global estadísticas tiene campos:

- fecha (formato ISO de la fecha actual)

- total_pedidos (int)
- recaudado (float): suma de totales de pedidos
- ventas_por_combo (dict): conteo de unidades vendidas por cada combo

Se muestran estas estadísticas formateadas en consola.

Reinicio de datos

Existe una funcionalidad que reinicia en memoria todos los datos manejados durante la ejecución del programa.

Interfaz de usuario

El programa presenta un menú interactivo con las siguientes opciones:

1. Generar pedido
2. Ver fila de pedidos
3. Ver estadísticas diarias
4. Reiniciar datos
5. Salir

Manejo de errores

Se capturan excepciones y errores inesperados de forma limpia.

Requisitos

Requisitos Funcionales

1. **Registrar pedido:** el usuario elige uno o varios combos y especifica la cantidad de cada uno.
2. **Aplicar descuento:** si el subtotal alcanza el monto mínimo configurable, se aplica automáticamente el porcentaje de descuento.
3. **Mostrar fila de preparación:** se visualiza la lista de pedidos pendientes con el tiempo acumulado que falta para atenderlos.
4. **Ver estadísticas diarias:** se consulta el número total de pedidos, el total recaudado y las unidades vendidas por combo.
5. **Reiniciar datos:** se restablecen todos los pedidos, la fila y las estadísticas.

Requisitos No Funcionales

- **Portabilidad:** corre en cualquier sistema con Python 3.8+ sin dependencias externas (solo librerías estándar).
- **Usabilidad:** interacción mediante menús de texto claros.
- **Rendimiento:** todas las operaciones responden en menos de un segundo en una máquina moderna.
- **Legibilidad:** el código está organizado en funciones cortas, con nombres descriptivos y manejo de errores para entradas inválidas.
- **Mantenibilidad:** estructura sencilla en un solo archivo, facilitando futuras ampliaciones o refactorizaciones.

Arquitectura y Diseño

El proyecto se concentra en un único script `main.py`, que agrupa:

- **Variables globales:** diccionarios COMBOS, DESCUENTO, almacenamiento en memoria (pedidos, fila, estadísticas) y contador de IDs.
- **Funciones de utilidad:**
 - `generar_id()`: incrementa y devuelve un identificador único por pedido.
 - `calcular_descuento(subtotal)`: devuelve el valor a descontar según el umbral.
 - `leer_entero(prompt, minimo, maximo)`: valida entradas numéricas y comandos especiales (f/c).
- **Flujos principales:**
 - `registrar_pedido()`: lógica de selección de combos, cálculo de totales, actualización de estadísticas y despliegue de tique.
 - `ver_fila()`: itera sobre la lista de IDs pendientes, sumando tiempos para mostrar el tiempo restante de cada pedido.
 - `ver_estadisticas()`: presenta la fecha, total de pedidos, recaudado y detalle de unidades vendidas.
 - `reiniciar_datos()`: resetea todas las estructuras para comenzar de cero.
- **Punto de entrada:**
 - La función `main()` ejecuta el bucle de menú que enlaza cada opción a las funciones anteriores y captura excepciones para manejar interrupciones o errores inesperados.