

Sistema de Gestión de Clientes y Menús en un Proyecto de Restauración:

Introducción

Este proyecto consiste en el desarrollo de una aplicación para gestionar ingredientes, menús, clientes y pedidos en un sistema de restauración. Utilizando la arquitectura ORM con SQLAlchemy y XAMPP con phpMyAdmin como gestor de base de datos, el sistema permite realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los ingredientes y los menús, gestionar los datos de los clientes y registrar los pedidos realizados. La aplicación también permite asociar ingredientes con menús a través de recetas, facilitando la creación de menús personalizados. La interfaz se ha diseñado utilizando la librería custom tkinter para aplicaciones de escritorio en Python.

Requisitos

Tecnologías Utilizadas:

- **Python 3.x**
- **SQLAlchemy** como ORM para gestionar la base de datos MySQL.
- **MySQL** a través de XAMPP y phpMyAdmin para la gestión de la base de datos.
- **Tkinter y CustomTkinter** para la interfaz gráfica de usuario.
- **XAMPP** como entorno de desarrollo local

Instalación del Proyecto: Descarga e instala XAMPP desde <https://www.apachefriends.org/es/index.html>. Asegúrate de tener MySQL en funcionamiento dentro de XAMPP. Configura la base de datos en phpMyAdmin, creando la base de datos orm_clientes. Descarga o clona el proyecto desde el repositorio. Configuración de la Base de Datos: Abre database.py y asegúrate de que la URL de conexión a la base de datos esté configurada correctamente: python Copiar código DATABASE_URL = "mysql+mysqlconnector://root:@localhost/orm_clientes" Si es necesario, ajusta las credenciales de MySQL según tu configuración. Ejecución del Proyecto: Abre la terminal y navega hasta el directorio del proyecto. Ejecuta el siguiente comando para inicializar las tablas de la base de datos: bash Copiar código python main.py Esto creará las tablas necesarias en la base de datos según la estructura definida en los modelos..

Estructura del Proyecto

El proyecto está estructurado en componentes reutilizables que representan las diferentes páginas y secciones del sitio web. A continuación, se describen algunos componentes principales:

- HeaderLog: Cabecera reutilizable para la navegación.
- FooterPM: Pie de página estándar.

- Páginas de servicios: Cada servicio está organizado como un componente individual.
- Registro: Formulario de registro de usuarios.

Funcionalidades Principales

Uso de la Aplicación: Panel de Ingredientes: Permite gestionar los ingredientes disponibles, crear, leer, actualizar y eliminar ingredientes. Panel de Menús: Crea menús a partir de los ingredientes disponibles. Los menús pueden tener varios ingredientes, con cantidades específicas. Panel de Clientes: Permite gestionar los clientes del sistema. Los clientes pueden ser registrados, actualizados y eliminados. Panel de Pedidos: Permite registrar pedidos de los clientes, asociando los menús seleccionados con un cliente. Manejo de Errores: Si un ingrediente o cliente ya existe, se mostrará un mensaje de advertencia. Si ocurre un error en la base de datos, se revertirá la transacción y se mostrará un mensaje de error..

Modelo de Base de Datos:

El sistema utiliza **SQLAlchemy** como ORM (Object-Relational Mapping) para interactuar con una base de datos **MySQL** a través de **XAMPP**. El modelo de datos está compuesto por varias tablas que gestionan la información relacionada con ingredientes, menús, recetas, clientes y pedidos.

Tablas del Sistema:

1. **Ingrediente:**
 - **nombre** (PK)
 - **tipo**
 - **cantidad**
 - **unidad_medida**
2. **Receta:**
 - **id** (PK)
 - **nombre**
 - **descripcion**
3. **RecetaIngrediente:**
 - **receta_id** (FK)
 - **ingrediente_id** (FK)
 - **cantidad**
4. **Cliente:**
 - **nombre** (PK)
 - **correo** (único)
5. **Pedido:**
 - **id** (PK)
 - **cliente_nombre** (FK)
 - **receta_nombre** (FK)

- **total**
- 6. **PedidoMenu:**
 - **pedido_id** (FK)
 - **receta_id** (FK)

Relaciones Entre Tablas:

- **Cliente y Pedido:** Un cliente puede realizar muchos pedidos, pero un pedido pertenece a un solo cliente.
- **Receta y RecetaIngrediente:** Una receta puede tener múltiples ingredientes asociados a ella, con una cantidad específica por cada ingrediente.
- **Pedido y Receta:** Un pedido puede contener varios menús (recetas) y una receta puede ser parte de varios pedidos.

Funcionalidades Principales:

1. **Gestión de Ingredientes:**
 - **Agregar Ingredientes:** Se pueden agregar nuevos ingredientes a la base de datos, asegurando que no haya duplicados.
 - **Actualizar Ingredientes:** Se pueden modificar los detalles de un ingrediente, como su cantidad o unidad de medida.
 - **Eliminar Ingredientes:** Se pueden eliminar ingredientes de la base de datos.
2. **Gestión de Menús:**
 - **Crear Menús:** Los menús se crean a partir de una combinación de ingredientes seleccionados, con la cantidad de cada ingrediente específica.
 - **Visualizar Menús:** Los menús creados se pueden visualizar con los ingredientes que los componen y sus cantidades.
 - **Actualizar Menús:** Se puede modificar la composición de un menú, agregando o eliminando ingredientes.
 - **Eliminar Menús:** Se puede eliminar un menú, eliminando su relación con los ingredientes.
3. **Gestión de Clientes:**
 - **Agregar Clientes:** Se pueden agregar nuevos clientes al sistema, asegurando que no haya correos duplicados.
 - **Actualizar Clientes:** Se pueden modificar los detalles de los clientes, como su nombre y correo electrónico.
 - **Eliminar Clientes:** Los clientes pueden ser eliminados del sistema.
4. **Gestión de Pedidos:**
 - **Registrar Pedidos:** Los clientes pueden realizar pedidos, seleccionando los menús que desean y obteniendo un total calculado.

- **Visualizar Pedidos:** Los pedidos realizados por los clientes se pueden visualizar, mostrando los detalles de los menús pedidos y el total a pagar.
 - **Eliminar Pedidos:** Se pueden eliminar pedidos si es necesario.
5. **Generación de Reportes:**
- El sistema permite generar reportes detallados sobre los ingredientes más utilizados, los menús más solicitados y las ventas totales.

Consideraciones Técnicas:

1. **Base de Datos:** La base de datos utilizada es **MySQL**, configurada a través de **XAMPP** y gestionada con **phpMyAdmin**. La estructura de las tablas se ha definido de acuerdo a las necesidades de gestión de ingredientes, menús, clientes y pedidos.
2. **Framework y Librerías:**
 - **SQLAlchemy:** Se utiliza para gestionar la base de datos y facilitar las interacciones con las tablas a través del ORM.
 - **Tkinter y CustomTkinter:** Estas librerías se utilizan para crear la interfaz gráfica del sistema, permitiendo una interacción fácil y fluida para los usuarios.
 - **XAMPP:** Se utiliza para crear un servidor local que aloje la base de datos y facilite su gestión mediante phpMyAdmin.
3. **Arquitectura:** El proyecto está estructurado en módulos independientes que permiten gestionar cada aspecto del sistema:
 - **Base de Datos:** Contiene los modelos y la lógica para interactuar con la base de datos.
 - **Interfaz de Usuario:** Utiliza Tkinter y CustomTkinter para mostrar los paneles y formularios.
 - **Lógica de Negocio:** Contiene el código que gestiona las operaciones CRUD sobre los ingredientes, menús, clientes y pedidos.
4. **Seguridad:** El sistema utiliza claves primarias y foráneas en la base de datos para garantizar la integridad de los datos y evitar problemas como la duplicación de registros. Además, el sistema verifica la unicidad de los correos electrónicos de los clientes para evitar registros duplicados.
5. **Escalabilidad:** El sistema ha sido diseñado de forma modular, lo que facilita agregar nuevas funcionalidades o realizar mejoras en el futuro. La estructura de la base de datos y la lógica de negocio permite agregar más funcionalidades sin alterar el núcleo del sistema.