

Documentación del Proyecto: Aplicación de Gestión de Restaurante

Este proyecto tiene como objetivo desarrollar una aplicación de escritorio diseñada para la gestión completa de un restaurante. La aplicación abarca la administración de usuarios (tanto administradores como clientes) y el control eficiente de mesas. Mi enfoque ha sido crear una interfaz intuitiva y funcional que satisfaga las necesidades típicas de un restaurante.

Módulo de (inicio_de_sesión):

La clase principal es **LoginGUI**, que genera una ventana de login con dos campos: "Nombre de usuario" y "Contraseña", además de dos botones, uno para iniciar sesión y otro para registrarse. Al iniciar sesión, los datos ingresados por el usuario se comparan con un archivo de texto en el cual están guardados los usuarios registrados. Si el nombre de usuario y la contraseña coinciden con un registro existente, el programa verifica si el rol del usuario es "Admin" o "Cliente". Dependiendo del rol, se abre una nueva ventana para acceder a las funcionalidades de administración o cliente, invocando las clases **MesasGUI** o **MesasClienteGUI** desde otros módulos del proyecto. Si las credenciales no coinciden, se muestra un mensaje de error utilizando **messagebox**.

El proceso de registro abre una nueva ventana mediante el método **abrir_ventana_registro**, donde se solicitan datos como el nombre, apellido, sexo, rol (administrador o cliente) y contraseña. Al completar el registro, los datos se guardan en un archivo de texto y se muestra un mensaje de confirmación. Tanto el proceso de inicio de sesión como el registro están diseñados para interactuar con archivos de texto locales, que almacenan de manera básica los detalles de los usuarios y sus contraseñas. Las contraseñas no están encriptadas, lo cual puede ser un aspecto a mejorar en términos de seguridad. El código finaliza con la creación de la ventana principal (**root**) y el inicio del ciclo de eventos de la aplicación, con el cual se mantiene la ventana activa hasta que el usuario la cierre.

Módulo (ventana_administrador):

En este segundo módulo del código tenemos la clase **MesasGUI** crea una interfaz gráfica que permite administrar las mesas de un restaurante. Para ello, se conecta a una base de datos MySQL donde se almacenan los datos de cada mesa, como el número de mesa, el cliente asignado, la ubicación, el estado y la cantidad de personas. La ventana principal muestra a las mesas disponibles, las cuales pueden ser arrastradas para cambiar su posición en caso de que el cliente solicite cambiarse de sector dentro del restaurante. La interfaz también cuenta con opciones para "añadir", "modificar" y "eliminar" mesas mediante botones ubicados en la parte inferior de la ventana.

Al iniciar el programa carga las mesas, solicitando la información en la base de datos. La opción para "añadir una mesa" abre una ventana emergente en la que el usuario introduce los datos de la nueva mesa, que se guardan en la base de datos y se muestran en la interfaz. La eliminación de mesas seleccionadas se realiza confirmando la acción y actualizando la base de datos y la interfaz.

La opción de "modificar una mesa" también abre una ventana emergente que permite cambiar el número de mesa, el cliente, la ubicación, el estado y la capacidad. Los cambios se reflejan tanto en la base de datos como en la interfaz gráfica, evitando duplicados de números de mesa. Si no se encuentra la mesa en la base de datos o ocurre algún error, se muestra un mensaje de alerta.

Módulo (ventana_cliente):

Este código crea una interfaz gráfica utilizando la biblioteca Tkinter, la cual permite a los clientes de un restaurante reservar mesas. Al igual que en el módulo anterior, en primera instancia se establece una conexión con una base de datos MySQL, donde se almacenan los datos de las mesas, como el número, ubicación, capacidad y estado (libre o reservado). En este contexto, el estado de la mesa es uno de los datos más importantes, ya que brinda al cliente la certeza de si la mesa que desea reservar está disponible o no.

En la interfaz gráfica, si una mesa está ocupada, aparecerá en color rojo; si está libre, aparecerá en verde. Una vez que el cliente solicite la reserva, deberá indicar a nombre de quién estará la misma. Si la conexión con la base de datos falla, el programa se cerrará.

Desafíos Durante el Desarrollo

A lo largo del desarrollo del proyecto, enfrenté varias complicaciones. En un principio, los datos ingresados por ambos tipos de usuarios se almacenaban en un archivo de texto (.txt). Sin embargo, me di cuenta de que podía hacerlo de una mejor manera y decidí que podría implementar una solución más compleja y técnica utilizando una base de datos.

La integración de la base de datos no fue sencilla. Al agregar esta funcionalidad, me encontré con numerosos errores, ya que el programa ya tenía casi toda su lógica definida. Esto resultó en una serie de problemas que impedían la ejecución correcta del programa. Después de varias horas de desarrollo y búsqueda de información, logré establecer la conexión con la base de datos y hacer que el programa funcionara nuevamente.

Sin embargo, surgió un nuevo desafío: los datos se cruzaban en la interfaz gráfica. Por ejemplo, el campo "Nombre de Cliente" mostraba el "Número de la Mesa". Este inconveniente me llevó a revisar nuevamente todo el código y reorganizarlo en función de la nueva estructura de la base de datos.

A pesar de los obstáculos, esta experiencia resultó ser extremadamente enriquecedora. Al final de las pruebas, me sentí muy satisfecho al ver que el programa cumplía con mis expectativas y objetivos iniciales. La capacidad de lograr esto hace que me den ganas de seguir creando y avanzando en distintos proyectos.

Vínculo de repositorio en Git Hub:

<https://github.com/Facu793/Proyecto-restaurante.python/invitations>