

Plan de Acción de Mejora

DAVID ALEJANDRO SOLANO MORENO

2996150

En este documento describere el proceso que estoy llevando a cabo para hacer el plan de acción de mejora correspondiente al proyecto de gestión hotelera asignado por la instructora *Isaura Suarez*. El objetivo es construir la estructura y la base visual del sistema web, Utilizando Flask y BootStrap, junto con el reposito de GitHub para así reforzar y mejorar mis conocimientos en el desarrollo de software.

Primer plan: Lo primero que hice fue crear un repositorio en *GitHub*, y dentro del este genere las carpetas correspondientes al proyecto de las cuales tenia que crear la carpeta **Templates** en la cual van las vistas (*.html*) del proyecto.

Creacion de vistas con BootStrap

Hice las vistas *Index*, *login* y *registro* utilizando HTML con el framework BootStrap guiándome de la guía que nos dejo la instructora y aplicando ya mis conocimientos

Index.html

- Pagina de bienvenido al sistema
- Contiene botones para el inicio de sesión o registrarse
- Pagina sencilla pero bonita y buena para la experiencia de usuario

Login.html

- Formulario de inicio de sesión
- Contiene los campos para ingresar el correo electrónico y contraseña
- Incluye un enlace para iniciar sesión si los datos son validos y otro para registrar si no tiene cuenta

Registro.html

- Formulario para nuevos usuarios
- Contiene campos para nombre completo, correo, contraseña y un botón para confirmar el registro
- Enlace para volver al login

Conexión de las vistas con Flask

Cree el archivo *app.py*, el cual contiene el código para que las vistas se conecten entre si. Por el momento no he utilizado base de datos ya que el objetivo es crear las vistas y establecer el flujo de navegación entre las mismas.

Repositorio de GitHub

[Solano08/AccionDeMejora](https://github.com/Solano08/AccionDeMejora)

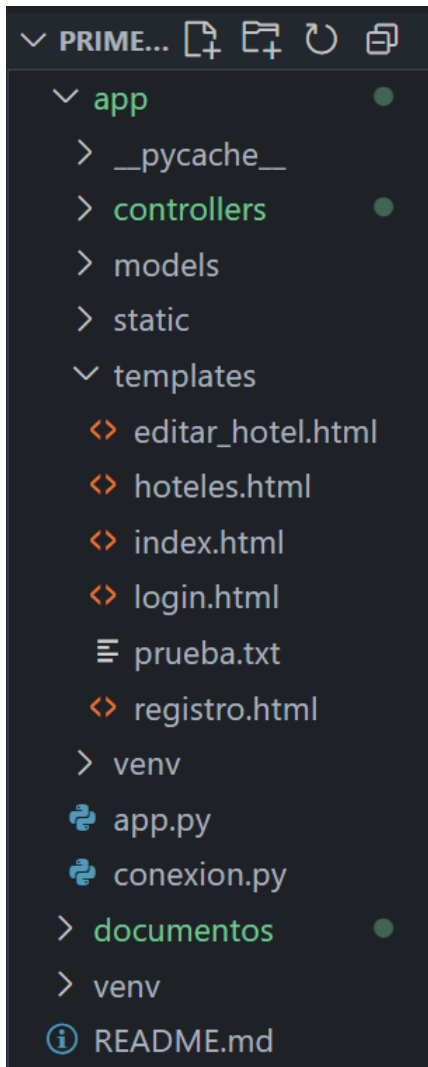
Segundo Plan: Estructura Del Proyecto

AccionDeMejora/

```
|
|
|└─ static/
|  └─ css/
|     └─ estilos.css
|
|└─ templates/
|   └─ index.html
|   └─ login.html
|   └─ registro.html
|
```

└─ app.py

└─ README.md



Controlador

El archivo app.py contiene la lógica básica de las rutas y la navegación por nuestro aplicativo, También configuramos la aplicación Flask y definilas las rutas a las diferentes vistas.

```

app > app.py > login
39 # Ruta login
40 @app.route('/login', methods=['GET', 'POST'])
41 def login():
42     if request.method == 'POST':
43         correo = request.form['correo']
44         password = request.form['password']
45         cursor.execute("SELECT * FROM usuarios WHERE correo=%s", (correo,))
46         user = cursor.fetchone()
47         if user and check_password_hash(user[3], password):
48             return redirect(url_for('dashboard'))
49     return render_template('login.html')
50
51 # Ruta registro
52 @app.route('/registro', methods=['GET', 'POST'])
53 def registro():

```

Cada ruta define una vista HTML que se encuentra en la carpeta templates

Esta separación por rutas/funciones nos facilita la expansión del sistema en futuras actualización como por ejemplo agregando panel de usuarios, etc.

También se encargó de gestión la encriptación de contraseña, conexión a BD, etc

Vistas del Proyecto

Esta carpeta contiene las vistas HTML que el usuario utiliza, todas fueron diseñadas con Bootstrap y un diseño único para el hotel

Index.html

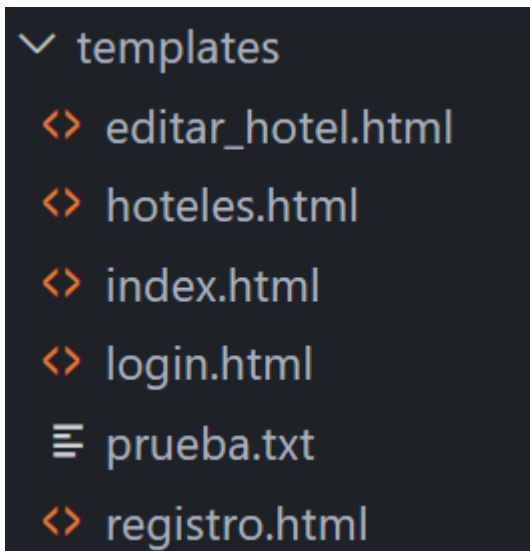
- Página principal de bienvenida.
- Enlaces a Login y Registro.
- Se usaron clases como container, text-center, entre otras

Login.html

- Formulario de autenticación de usuarios.
- Campos: Correo y Contraseña.
- Enlaces hacia la página de registro si el usuario no tiene cuenta.
- Componentes input, label y botones

Registro.html

- Formulario para nuevos usuarios.
- Campos: Nombre completo, correo, contraseña.
- Enlace de regreso al login.
- Botones personalizados y form-control, form-label



Conexión.py

```
conexion.py > ...
import pymysql

def obtener_conexion():
    return pymysql.connect(
        host='localhost',
        port=3307, # Asegúrate de que este puerto sea el correcto
        user='root', # o el usuario que uses
        password='SENA', # tu contraseña de MySQL
        db='hoteles', # nombre de tu base de datos
        charset='utf8mb4',
        cursorclass=pymysql.cursors.DictCursor
    )
```

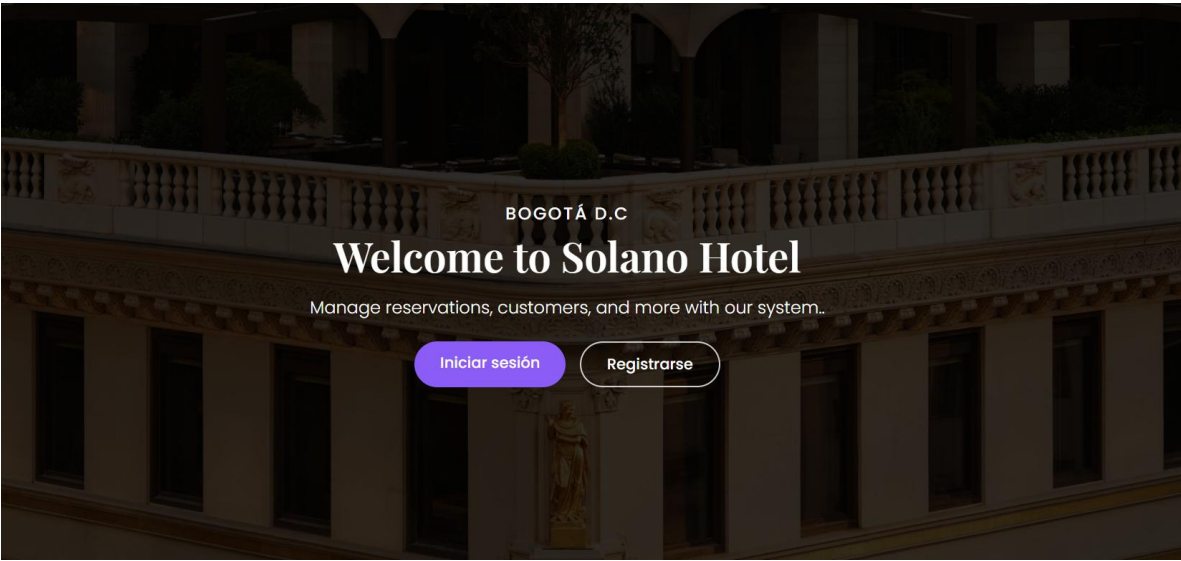
Este archivo nos sirve como puente entre Flask y la base de datos **Hoteles**. Lo que hacemos es poner los datos de nuestro SQL correctamente

Encriptacion de contraseñas

Para la encriptación de contraseña maneje la librería `wekzeug.security`, Esta la trabaje en vistas de `registro.html` y `login.html`

4	der2	der2@gmail.com	srypt:32768:8:1\$Wpp6EPNtCGnkxRjy\$271387...	repcionista
5	ds	dsa@gmail.com	srypt:32768:8:1\$zjuOFvHndkXGCNkZ\$170d79...	repcionista

```
1  -- Crear la base de datos
2  CREATE DATABASE IF NOT EXISTS hoteles;
3  USE hoteles;
4
5  -- Tabla: categorías de hotel
6  CREATE TABLE categorias (
7      id INT AUTO_INCREMENT PRIMARY KEY,
8      nombre VARCHAR(50) NOT NULL
9  );
10
11 -- Tabla: hoteles
12 CREATE TABLE hoteles (
13     id INT AUTO_INCREMENT PRIMARY KEY,
14     nombre VARCHAR(100) NOT NULL,
15     direccion VARCHAR(150),
16     telefono VARCHAR(20),
17     anio_apertura INT,
18     categoria_id INT,
19     FOREIGN KEY (categoria_id) REFERENCES categorias(id)
20 );
21
22 -- Tabla: tipos de habitación
23 CREATE TABLE tipos_habitacion (
24     id INT AUTO_INCREMENT PRIMARY KEY,
25     nombre VARCHAR(50) NOT NULL,
26     descripcion TEXT
27 );
28
29 -- Tabla: habitaciones
30 CREATE TABLE habitaciones (
31     id INT AUTO_INCREMENT PRIMARY KEY,
32     codigo VARCHAR(10) NOT NULL,
33     numero INT,
34     piso INT,
35     estado ENUM('disponible', 'ocupada') DEFAULT 'disponible',
36     hotel_id INT,
37     tipo_id INT,
38     FOREIGN KEY (hotel_id) REFERENCES hoteles(id),
39     FOREIGN KEY (tipo_id) REFERENCES tipos_habitacion(id)
40 );
41
42 -- Tabla: tipos de cliente
43 CREATE TABLE tipos_cliente (
44     id INT AUTO_INCREMENT PRIMARY KEY,
45     nombre VARCHAR(50) NOT NULL -- nacional, extranjero, agencia, etc.
46 );
47
48 -- Tabla: clientes
49 CREATE TABLE clientes (
50     id INT AUTO_INCREMENT PRIMARY KEY,
51     nombre VARCHAR(100) NOT NULL,
```



BOGOTÁ D.C

Welcome to Solano Hotel

Manage reservations, customers, and more with our system..

Iniciar sesión

Registrarse

BOGOTÁ D.C

Registro de Usuario

Nombre completo

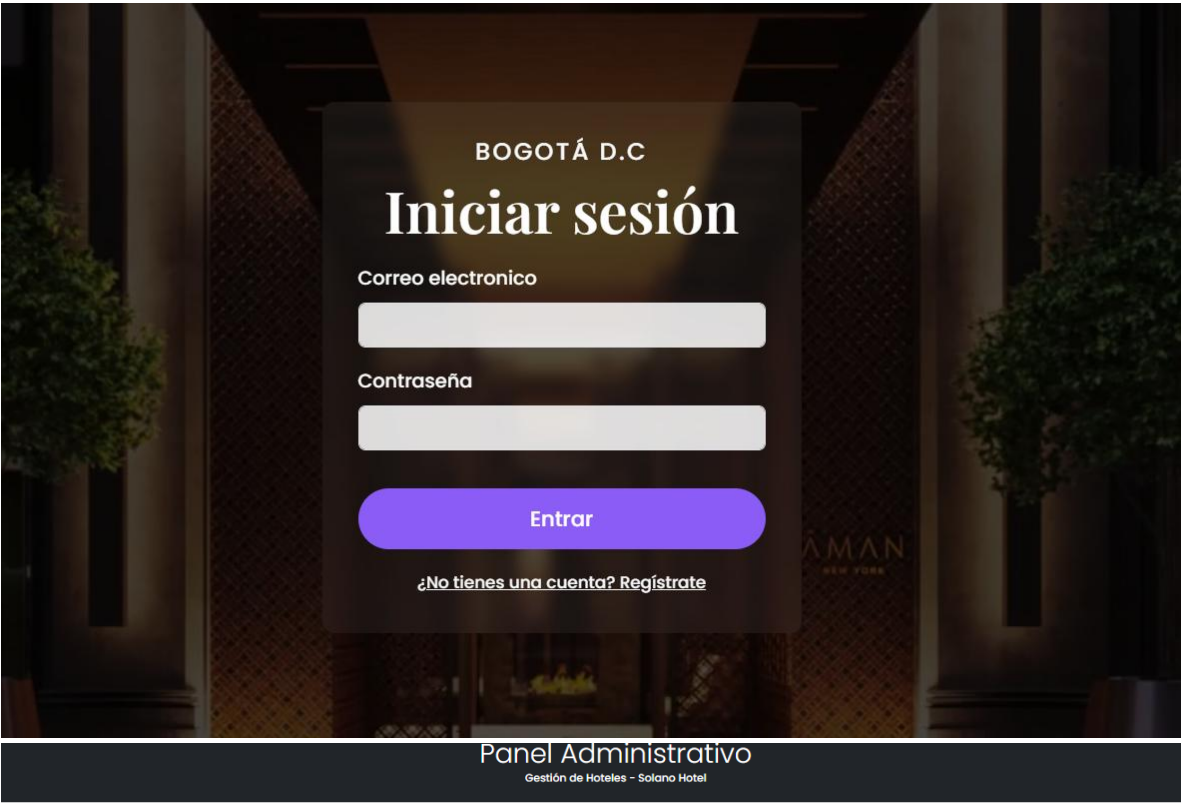
Correo electrónico

Usuario

Contraseña

Registrarse

[¿Ya tienes una cuenta? Inicia sesión](#)



GESTIÓN DE HOTELES

Nombre del Hotel

Categoría

Lujo

Dirección

Teléfono

Año de Fundación

Guardar Hotel

Cancelar

Lista de Hoteles Registrados

ID	Nombre	Categoría	Dirección	Teléfono	Año	Acciones
No hay hoteles registrados.						