

Informe detallado de la aplicación Hilton Hotel - Análisis del flujo y estructura del proyecto

Juan Sebastian Martinez Pinto

Instructora: Isaura María Suárez Novoa

Análisis y Desarrollo de Software (ADSO) 2996150

Servicio Nacional de Aprendizaje (SENA)

2025

Objetivo General

Desarrollar una aplicación web para la gestión de hoteles Hilton, permitiendo a usuarios registrados administrar información de hoteles, categorías, habitaciones y tipos de habitación. La plataforma utiliza Python con Flask, una base de datos MySQL y Bootstrap para el diseño responsivo.

1. Información general del proyecto

- **Nombre del proyecto:** Sistema de Gestión Hotelera - Hilton Hotels
- **Tecnologías empleadas:** Python 3.13, Flask, MySQL (Workbench), HTML + Jinja2, Bootstrap 5, Flask-MySQLdb, Flask-Login, Bcrypt
- **Patrón de diseño:** Modelo Vista Controlador (MVC)
- **Entorno de desarrollo:** Visual Studio Code, sistema operativo Windows

2. Estructura de carpetas y archivos principales

```

HILTON-HOTEL/
|
├─ app/
|   ├── __init__.py
|   ├── extensions.py
|   |
|   └─ controllers/
|       ├── auth_controller.py
|       ├── categoria_controller.py
|       ├── habitacion_controller.py
|       ├── hotel_controller.py
|       └─ tipo_habitacion_controller.py
|   |
|   └─ models/
|       ├── categoria_model.py
|       ├── habitacion_model.py
|       ├── hotel_model.py
|       ├── pais_model.py
|       ├── tipo_habitacion_model.py
|       └─ usuario_model.py
|   |
|   └─ templates/
|       ├── base.html
|       |
|       ├── categoria/
|           ├── crear.html
|           ├── editar_categoria.html
|           └─ listar.html
|       |
|       ├── habitacion/
|           ├── crear.html
|           ├── editar_habitacion.html
|           └─ listar.html
|       |
|       ├── hotel/
|           ├── crear.html
|           ├── editar_hotel.html
|           └─ listar.html
|       |
|       ├── tipohabitacion/
|           ├── crear.html
|           ├── editar_tipo_habitacion.html
|           └─ listar.html
|       |
|       └─ usuario/
|           ├── dashboard.html
|           ├── login.html
|           └─ registro.html
|
├─ .env
├─ .gitignore
├─ config.py
├─ requirements.txt
└─ run.py

```

3. Flujo de navegación de la aplicación

- El usuario inicia en la página de inicio de sesión (/login) o en la página de registro (/registro), si es nuevo.
- Una vez autenticado correctamente, es redirigido al panel principal de la aplicación (/dashboard).
- Desde el **Dashboard**, el usuario tiene acceso a las siguientes gestiones:
- **Gestión de Hoteles**
Permite crear, listar, editar y eliminar hoteles registrados.
- **Gestión de Categorías de Hotel**
Permite administrar las categorías disponibles (económico, estándar, lujo, etc.), junto con su descripción y servicios típicos.
- **Gestión de Tipos de Habitación**
Permite registrar, modificar o eliminar los tipos de habitación como “Individual”, “Suite”, “Familiar”, entre otros.
- **Gestión de Habitaciones**
Asocia habitaciones a hoteles, países y tipos, con su respectivo código identificador.
- Cada módulo incluye un CRUD independiente con formularios de creación y edición, así como opciones para eliminar o visualizar registros existentes.
- El sistema oculta todas las rutas de gestión hasta que el usuario ha iniciado sesión.

4. Rutas implementadas

- / – Redirección automática al login
- /login – Ingreso de usuarios al sistema
- /registro – Registro de nuevos usuarios
- /logout – Cierre de sesión del usuario
- /dashboard – Panel principal (visible solo si el usuario ha iniciado sesión)
- Gestión de Hoteles
- /hoteles – Listar hoteles registrados
- /hoteles/crear – Crear un nuevo hotel
- /hoteles/editar/<id> – Editar un hotel existente
- /hoteles/eliminar/<id> – Eliminar un hotel
- Gestión de Categorías de Hotel
- /categorias – Listar categorías existentes
- /categorias/crear – Crear una nueva categoría
- /categorias/editar/<id> – Editar una categoría existente
- /categorias/eliminar/<id> – Eliminar una categoría
- Gestión de Tipos de Habitación
- /tipos_habitacion – Listar tipos de habitación registrados
- /tipos_habitacion/crear – Crear un nuevo tipo de habitación
- /tipos_habitacion/editar/<id> – Editar un tipo de habitación

- /tipos_habitacion/eliminar/<id> – Eliminar un tipo de habitación
- Gestión de Habitaciones
- /habitaciones – Listar habitaciones del sistema
- /habitaciones/crear – Crear una nueva habitación
- /habitaciones/editar/<id> – Editar una habitación existente
- /habitaciones/eliminar/<id> – Eliminar una habitación

5. Base de datos usada

La base de datos usada fue diseñada en MySQL Workbench y contiene las siguientes tablas principales:

- usuarios
- pais
- hotel
- categoria
- tipohabitacion
- habitacion

6. Consideraciones de seguridad

- Se usa Bcrypt para el hash de contraseñas
- Flask-Login protege rutas con @login_required
- El sistema solo permite acceso al dashboard si está autenticado

7. Mejoras visuales e intuitivas implementadas

- Navbar con Bootstrap que solo muestra enlaces si hay sesión activa
- Mensajes de feedback (registro, login fallido, logout)
- Templates reutilizan base.html

8. Archivos necesarios para despliegue

- .env con credenciales
- Requeriments.txt con:

Flask
 Flask-MySQLdb
 Flask-Login
 Flask-Bcrypt
 python-dotenv

