

Universidad Interamericana  
Arecibo, Puerto Rico

# **Sistemas de Reservas de Hotel**

Javier X. Ramos Hernández  
R00581390  
COMP3400 SOFTWARE ENGINEERING  
Prof. Dastas

## Contents

<b>Resumen Ejecutivo .....</b>	<b>3</b>
<b>Objetivos SMART .....</b>	<b>3</b>
<b>Miembros del Equipo y Roles .....</b>	<b>3</b>
<b>Backlog de Producto .....</b>	<b>4</b>
<b>Estructura de Desglose del Trabajo (WBS) .....</b>	<b>4</b>
<b>Diseños del Proyecto .....</b>	<b>5</b>
<b>Tabla casos de prueba.....</b>	<b>7</b>
<b>Documentación por Lección.....</b>	<b>7</b>
<b>Lección 1: Introducción al Proyecto y Configuración del Entorno .....</b>	<b>7</b>
<b>Lección 2: Diseño de la Arquitectura y Comunicación entre Capas.....</b>	<b>8</b>
<b>Lección 3: Desarrollo de la Capa de Datos .....</b>	<b>9</b>
<b>Lección 4: Desarrollo de la Lógica de Negocio con Flask .....</b>	<b>10</b>
<b>Lección 5: Desarrollo de la Capa de Presentación .....</b>	<b>11</b>
<b>Lecciones 6 a 10 .....</b>	<b>11</b>
<b>Capturas de pantalla de las funcionalidades de la aplicación.....</b>	<b>12</b>
<b>Retrospectiva del Trabajo .....</b>	<b>15</b>
<b>Conclusión.....</b>	<b>16</b>

## Resumen Ejecutivo

El Sistema de Reservas de Hotel es una aplicación web diseñada para facilitar el registro, autenticación y gestión de reservas de habitaciones. Este proyecto fue desarrollado siguiendo una arquitectura en capas que asegura escalabilidad y mantenimiento eficiente. Las funcionalidades principales incluyen:

- Registro de clientes.
  - Gestión de habitaciones y reservas.
  - Interfaz administrativa y de cliente.
- 

## Objetivos SMART

- **Específico:** Desarrollar una aplicación funcional que permita gestionar reservas de hotel en un entorno realista.
- **Medible:** Implementar al menos 10 funcionalidades clave y realizar pruebas con 20 usuarios.
- **Alcanzable:** Completar el desarrollo en 1Resumen Ejecutivo

El Sistema de Reservas de Hotel es una aplicación web diseñada para facilitar el registro, autenticación y gestión de reservas de habitaciones. Este proyecto fue desarrollado siguiendo una arquitectura en capas que asegura escalabilidad y mantenimiento eficiente. Las funcionalidades principales incluyen:

- Registro de clientes.
  - Gestión de habitaciones y reservas.
  - Interfaz administrativa y de cliente.
- 

## Miembros del Equipo y Roles

1. **Estudiante 1:** Responsable de la Capa de Datos
    - Configuración de MariaDB y creación de tablas.
    - Población de datos y configuración de acceso.
    - Responsable de la Lógica de Negocio
    - Desarrollo de la API RESTful con Flask.
    - Implementación de autenticación y operaciones CRUD.
    - Responsable de la Capa de Presentación
    - Diseño de interfaces con HTML, Bootstrap y jQuery.
    - Integración con la API y configuración de NGINX.
-

## **Backlog de Producto**

1. Configuración de VMs.
2. Instalación de herramientas necesarias.
3. Diseño de base de datos (ER y scripts).
4. Implementación de API RESTful.
5. Creación de interfaz de usuario.
6. Integración y pruebas funcionales.
7. Configuración de acceso remoto con SERVEO.
8. Creación del repositorio y documentación.
9. Revisión y pruebas finales.
10. Presentación del proyecto.

## **Estructura de Desglose del Trabajo (WBS)**

### **1. Planeación:**

- Definición de objetivos y alcance.
- Asignación de roles.

### **2. Configuración Inicial:**

- Creación de VMs.
- Instalación de herramientas.

### **3. Desarrollo:**

- Capa de Datos.
- Capa de Lógica de Negocio.
- Capa de Presentación.

### **4. Integración y Pruebas:**

- Comunicación entre capas.
- Pruebas unitarias y funcionales.

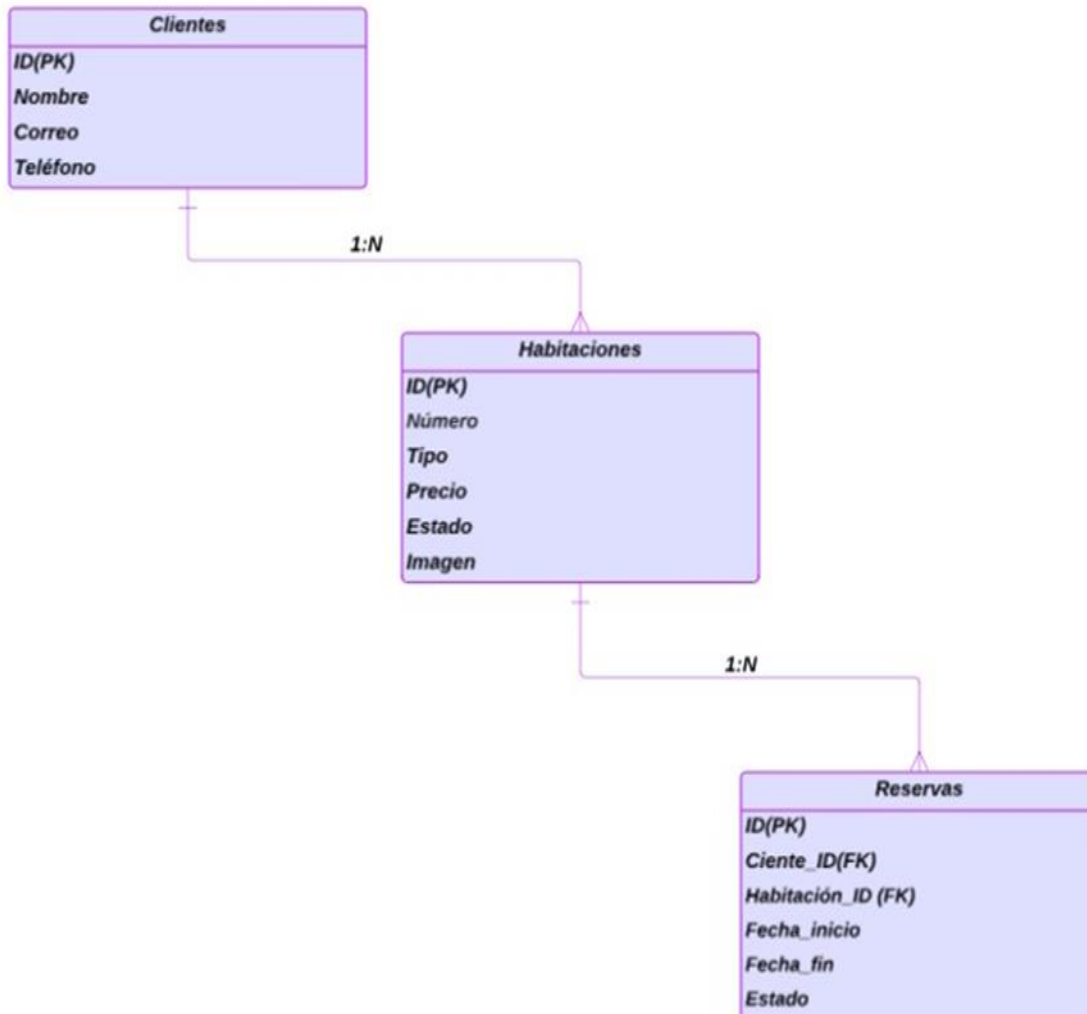
### **5. Documentación y Entrega:**

- Creación de documentación.
- Presentación final.

## Diseños del Proyecto

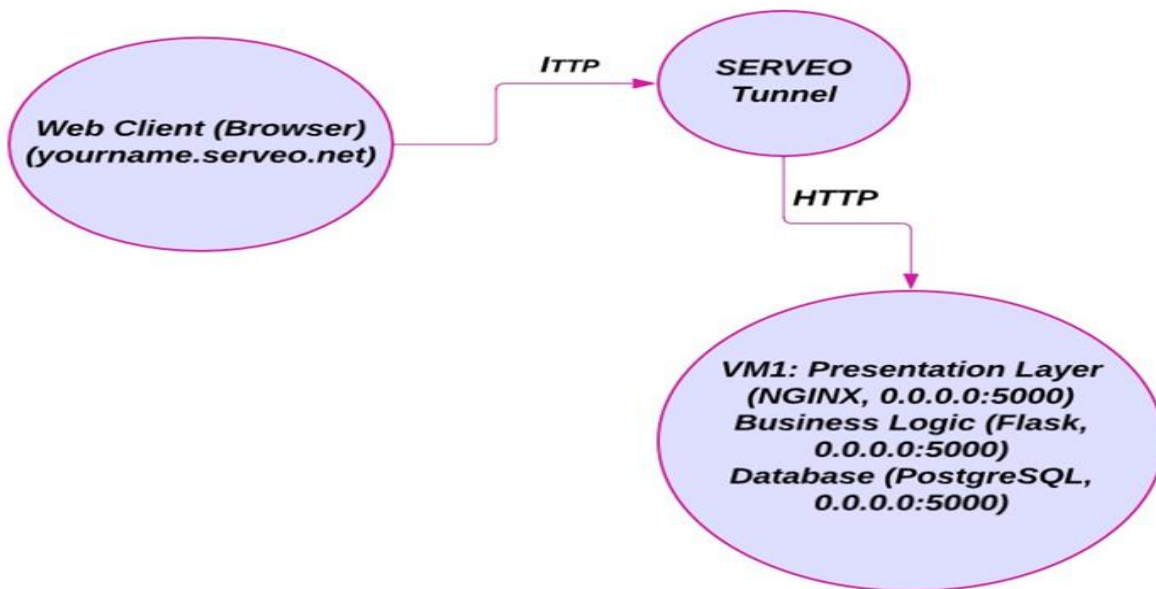
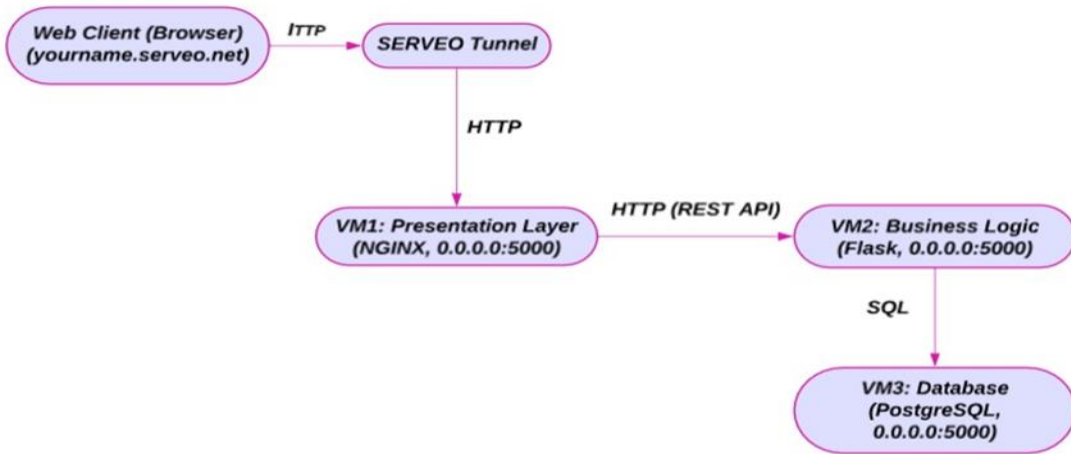
### Diagrama ER

- **Descripción:** Este diagrama detalla la relación entre las tablas principales: clientes, habitaciones y reservas.



## Diagrama de Arquitectura

- **Descripción:** Muestra la disposición de las capas y las tecnologías utilizadas.



## Tabla casos de prueba

ID	Funcionalidad	Entrada	Procedimiento	Resultado Esperado	Resultado Real
TC01	Registro de Usuario	Nombre, email, contraseña	Completar el formulario de registro y enviar	Usuario registrado y redirigido a la página de inicio de sesión	Usuario registrado correctamente y redirigido.
TC02	Inicio de Sesión	Email y contraseña válidos	Ingresar credenciales y hacer clic en "Iniciar Sesión"	Usuario autenticado y redirigido al panel de usuario	Inicio de sesión exitoso y redirección al panel.
TC03	Reserva de Habitación	Seleccionar habitación y fechas	Navegar a la habitación, seleccionar fechas y reservar	Reserva creada y confirmación mostrada al usuario	Reserva confirmada con mensaje de éxito.
TC04	Ver Habitaciones Disponibles	N/A	Acceder a la página de inicio	Lista de habitaciones disponibles mostrada	Habitaciones disponibles cargadas correctamente.

## Documentación por Lección

### Lección 1: Introducción al Proyecto y Configuración del Entorno

#### Objetivos:

- Comprender los requisitos y objetivos del proyecto.
- Configurar máquinas virtuales para cada capa del proyecto.
- Instalar herramientas necesarias como NGINX, Flask y MariaDB.
- Establecer la estructura básica del proyecto y el repositorio Git.

#### Actividades Realizadas:

##### 1. Creación de Máquinas Virtuales:

- Configuración de tres VMs para las capas de presentación, lógica de negocio y datos.
- Especificaciones de las VMs:
  - Sistema Operativo: Ubuntu 20.04 LTS
  - Recursos mínimos: 1 CPU, 1GB RAM, 10GB HDD.

##### 2. Instalación de NGINX en VM1:

```
sudo apt update
```

```
sudo apt install nginx
sudo systemctl start nginx
```

### 3. Configuración de Flask en VM2:

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "Capa de Lógica de Negocio - Flask funcionando"

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

### 4. Instalación de MariaDB en VM3:

```
sudo apt update
sudo apt install mariadb-server
sudo mysql_secure_installation
```

#### Resultado:

- Todas las VMs están configuradas y listas para el desarrollo.

## Lección 2: Diseño de la Arquitectura y Comunicación entre Capas

#### Objetivos:

- Definir la arquitectura en capas.
- Diseñar un diagrama de arquitectura.
- Configurar la comunicación entre capas usando NGINX y SERVEO.

#### Actividades Realizadas:

##### Configuración de NGINX como Proxy Inverso:

```
server {
    listen 80;
    server_name localhost;

    location / {
        root /var/www/html;
        index index.html;
    }

    location /api/ {
        proxy_pass http://192.168.1.20:5000/;
    }
}
```



```

        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

```

### Túnel SSH con SERVEO:

```
ssh -R 80:localhost:80 serveo.net
```

### Resultado:

- Las capas están comunicadas y accesibles remotamente.

## Lección 3: Desarrollo de la Capa de Datos

### Objetivos:

- Crear base de datos en MariaDB.
- Definir scripts SQL para tablas y datos de prueba.

### Actividades Realizadas:

#### Script SQL para Crear Tablas:

```

CREATE TABLE clientes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    email VARCHAR(100) UNIQUE
);

CREATE TABLE habitaciones (
    id INT AUTO_INCREMENT PRIMARY KEY,
    tipo VARCHAR(50),
    precio DECIMAL(10,2)
);

CREATE TABLE reservas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    cliente_id INT,
    habitacion_id INT,
    fecha_inicio DATE,
    fecha_fin DATE,
    FOREIGN KEY (cliente_id) REFERENCES clientes(id),
    FOREIGN KEY (habitacion_id) REFERENCES habitaciones(id)
);

```

#### Población de Datos:

```

INSERT INTO clientes (nombre, email) VALUES ('Juan Pérez',
'juan@example.com');
INSERT INTO habitaciones (tipo, precio) VALUES ('Sencilla', 50.00);

```

```
INSERT INTO reservas (cliente_id, habitacion_id, fecha_inicio,
fecha_fin) VALUES (1, 1, '2023-01-01', '2023-01-05');
```

### Resultado:

- Base de datos configurada y poblada con datos de prueba.

## Lección 4: Desarrollo de la Lógica de Negocio con Flask

### Objetivos:

- Configurar Flask y definir rutas RESTful.
- Implementar autenticación y autorización.

### Actividades Realizadas:

#### Estructura de Proyecto Flask:

```
hotel_app/
├── app.py
├── models/
│   └── routes/
```

#### Ruta RESTful para Clientes:

```
@app.route('/clientes', methods=['GET'])
def get_clientes():
    cursor = db.connection.cursor()
    cursor.execute("SELECT * FROM clientes")
    return jsonify(cursor.fetchall())
```

#### Autenticación con Flask-Login:

```
from flask_login import LoginManager, UserMixin

login_manager = LoginManager()
login_manager.init_app(app)

class User(UserMixin):
    pass
```

### Resultado:

- API funcional con rutas RESTful y autenticación.

## Lección 5: Desarrollo de la Capa de Presentación

### Objetivos:

- Diseñar interfaz de usuario con HTML, Bootstrap y jQuery.
- Consumir API RESTful.

### Actividades Realizadas:

#### Página de Inicio:

```
<h1>Bienvenido al Hotel</h1>
<div id="habitaciones"></div>
<script>
    fetch('/api/habitaciones')
        .then(response => response.json())
        .then(data => console.log(data));
</script>
```

#### Interactividad con jQuery:

```
$(document).ready(function() {
    $('#tabla').DataTable();
});
```

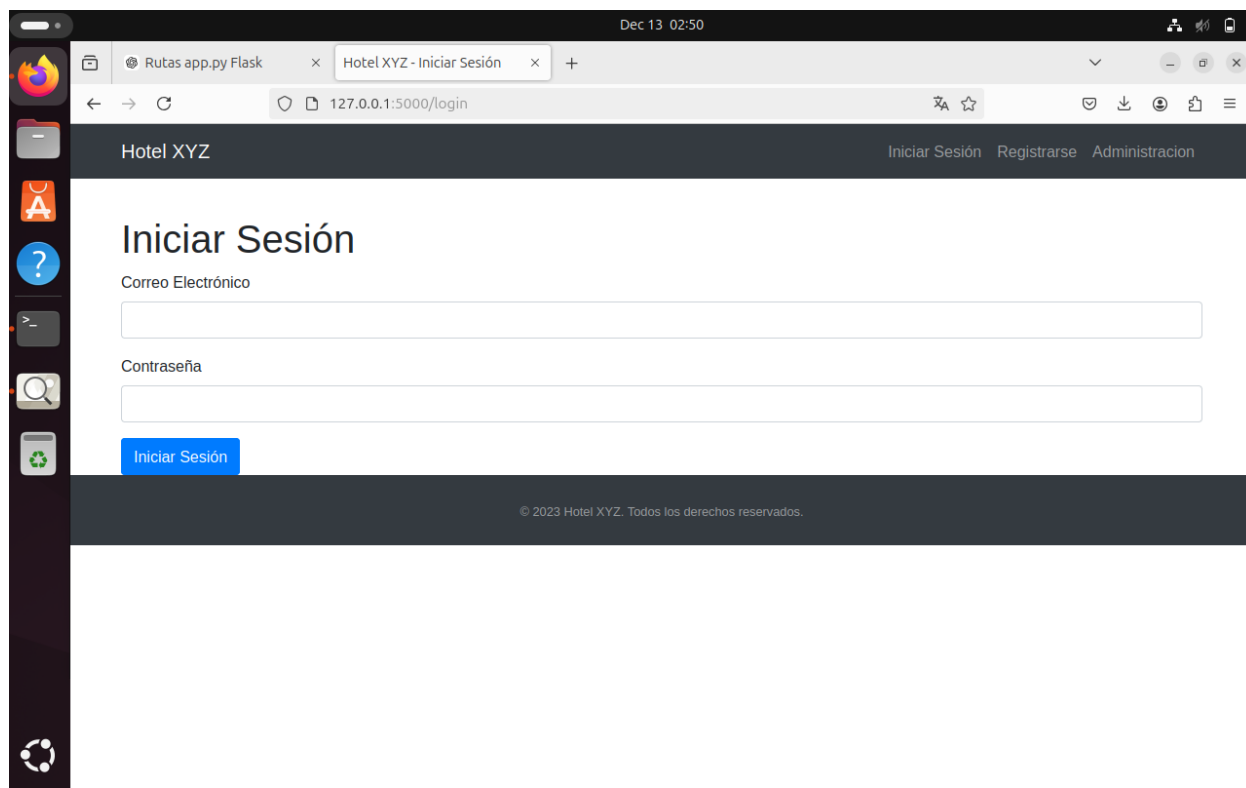
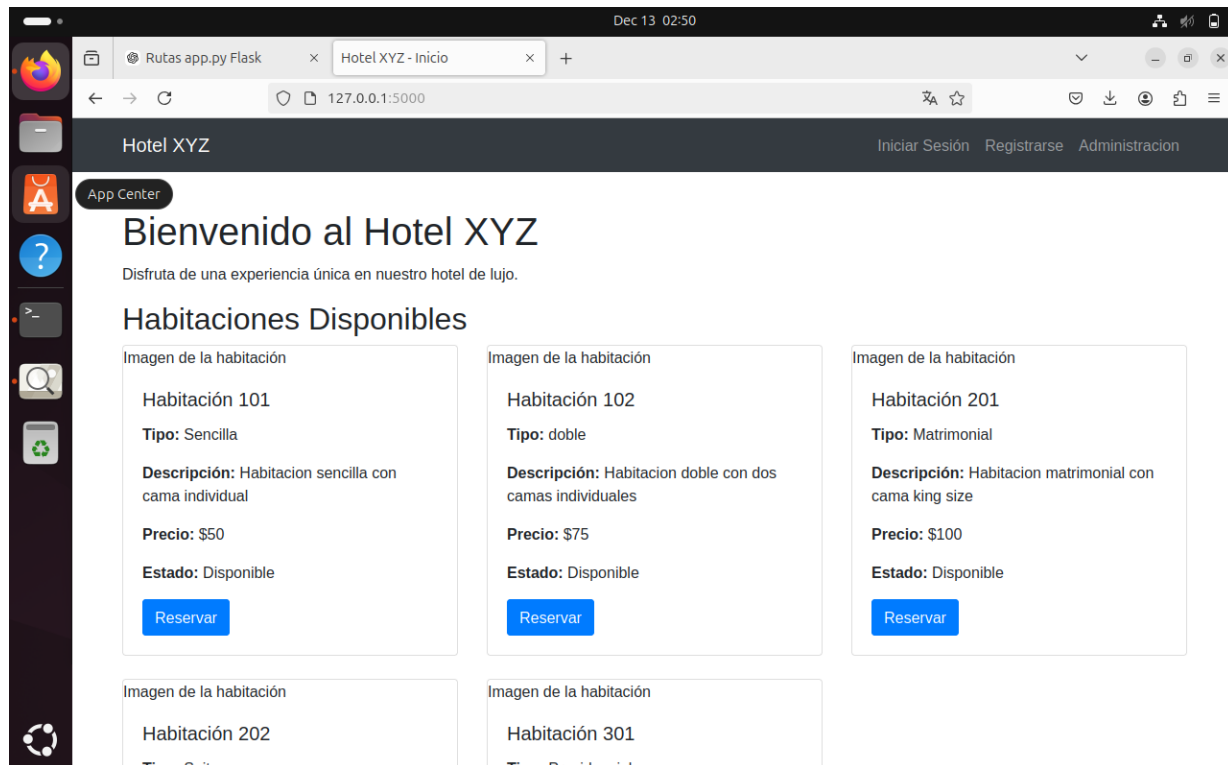
#### Resultado:

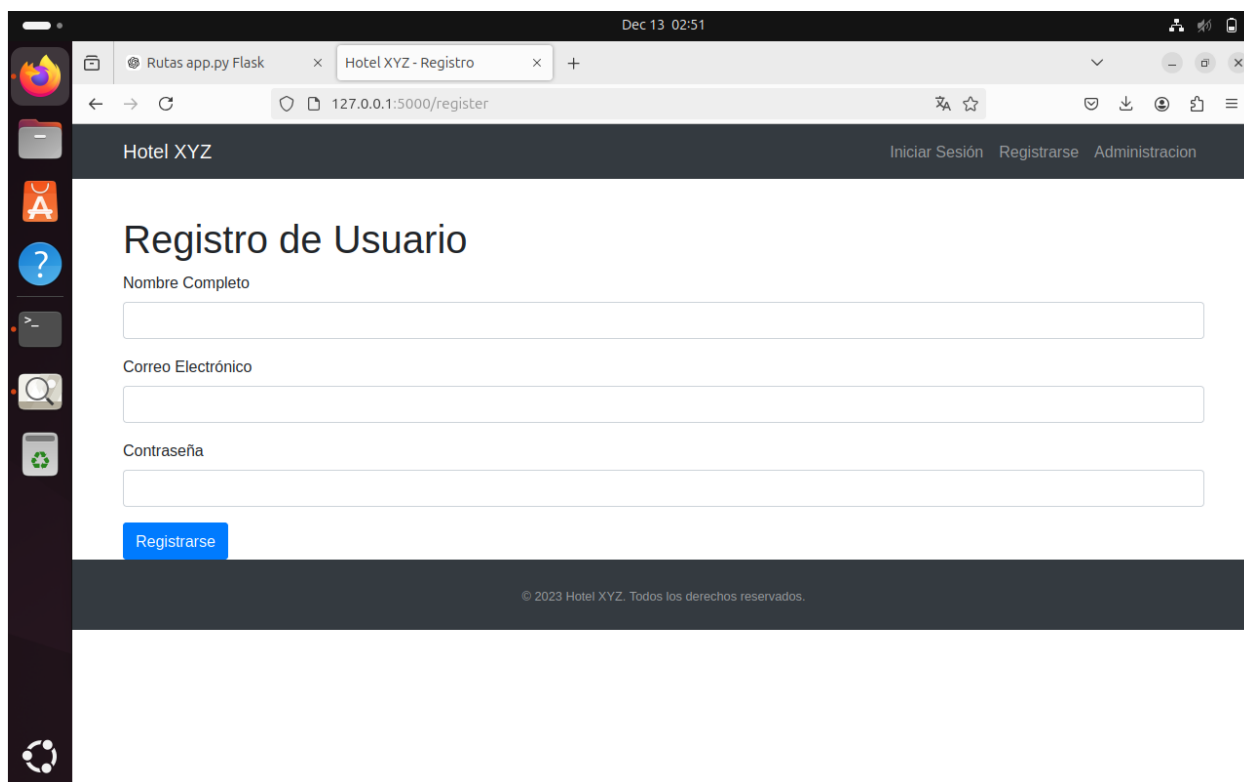
- Interfaz de usuario completa y funcional.

## Lecciones 6 a 10

- Estas lecciones cubren integración, pruebas, documentación y entrega final. Cada paso asegura la cohesión y calidad del proyecto.

## Capturas de pantalla de las funcionalidades de la aplicación





The screenshot shows a web browser window with the title "Hotel XYZ - Registro". The address bar shows the URL "127.0.0.1:5000/register". The page has a dark header with the "Hotel XYZ" logo and navigation links: "Iniciar Sesión", "Registrarse", and "Administración". The main content area is titled "Registro de Usuario" and contains three input fields: "Nombre Completo", "Correo Electrónico", and "Contraseña". A blue "Registrarse" button is positioned below the fields. The footer displays the copyright notice "© 2023 Hotel XYZ. Todos los derechos reservados."

Hotel XYZ

Iniciar Sesión Registrarse Administración

## Registro de Usuario

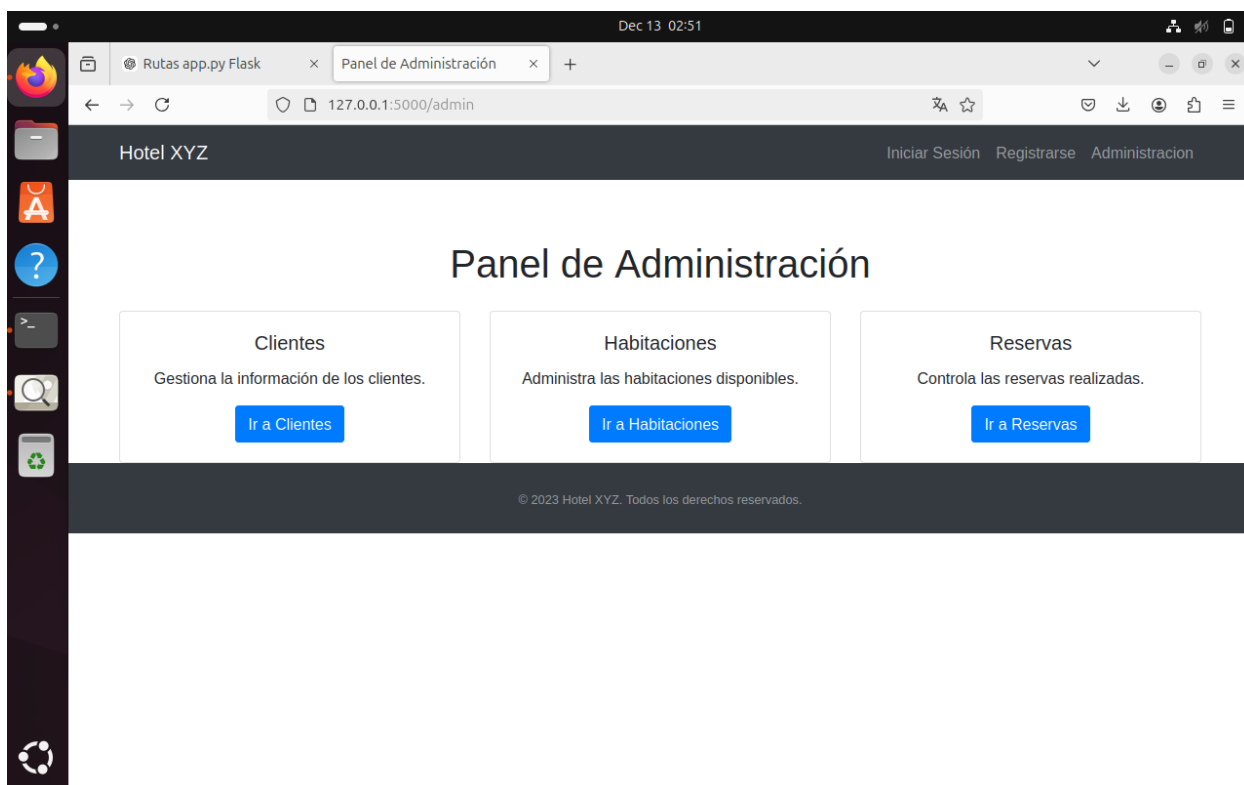
Nombre Completo

Correo Electrónico

Contraseña

Registrarse

© 2023 Hotel XYZ. Todos los derechos reservados.



The screenshot shows a web browser window with the title "Panel de Administración". The address bar shows the URL "127.0.0.1:5000/admin". The page has a dark header with the "Hotel XYZ" logo and navigation links: "Iniciar Sesión", "Registrarse", and "Administración". The main content area is titled "Panel de Administración" and features three cards: "Clientes" (with the description "Gestiona la información de los clientes." and a button "Ir a Clientes"), "Habitaciones" (with the description "Administra las habitaciones disponibles." and a button "Ir a Habitaciones"), and "Reservas" (with the description "Controla las reservas realizadas." and a button "Ir a Reservas"). The footer displays the copyright notice "© 2023 Hotel XYZ. Todos los derechos reservados."

Hotel XYZ

Iniciar Sesión Registrarse Administración

## Panel de Administración

**Clientes**

Gestiona la información de los clientes.

Ir a Clientes

**Habitaciones**

Administra las habitaciones disponibles.

Ir a Habitaciones

**Reservas**

Controla las reservas realizadas.

Ir a Reservas

© 2023 Hotel XYZ. Todos los derechos reservados.

Dec 13 02:52

Rutas app.py Flask x Administración de Clientes x +

127.0.0.1:5000/admin/clientes

Hotel XYZ Iniciar Sesión Registrarse Administración

## Administración de Clientes

Mostrar 10 registros Buscar:

ID	Nombre	Dirección	Teléfono	Email	Fecha de Registro
1	Juan Perez	Calle Falsa 123	555-1234	juan.perez@example.com	2024-12-07 03:00:57
2	Maria Garcia	Avenida Siempre Viva 456	555-5678	maria.garcia@example.com	2024-12-07 03:00:57
3	Carlos Lopez	Boulevard de los Suenos 789	555-9012	carlos.lopez@example.com	2024-12-07 03:00:57

Mostrando 1 a 3 de 3 registros Anterior 1 Siguiente

© 2023 Hotel XYZ. Todos los derechos reservados.

Dec 13 02:52

Rutas app.py Flask x Administración de Habitaciones x +

127.0.0.1:5000/admin/habitaciones

Hotel XYZ Iniciar Sesión Registrarse Administración

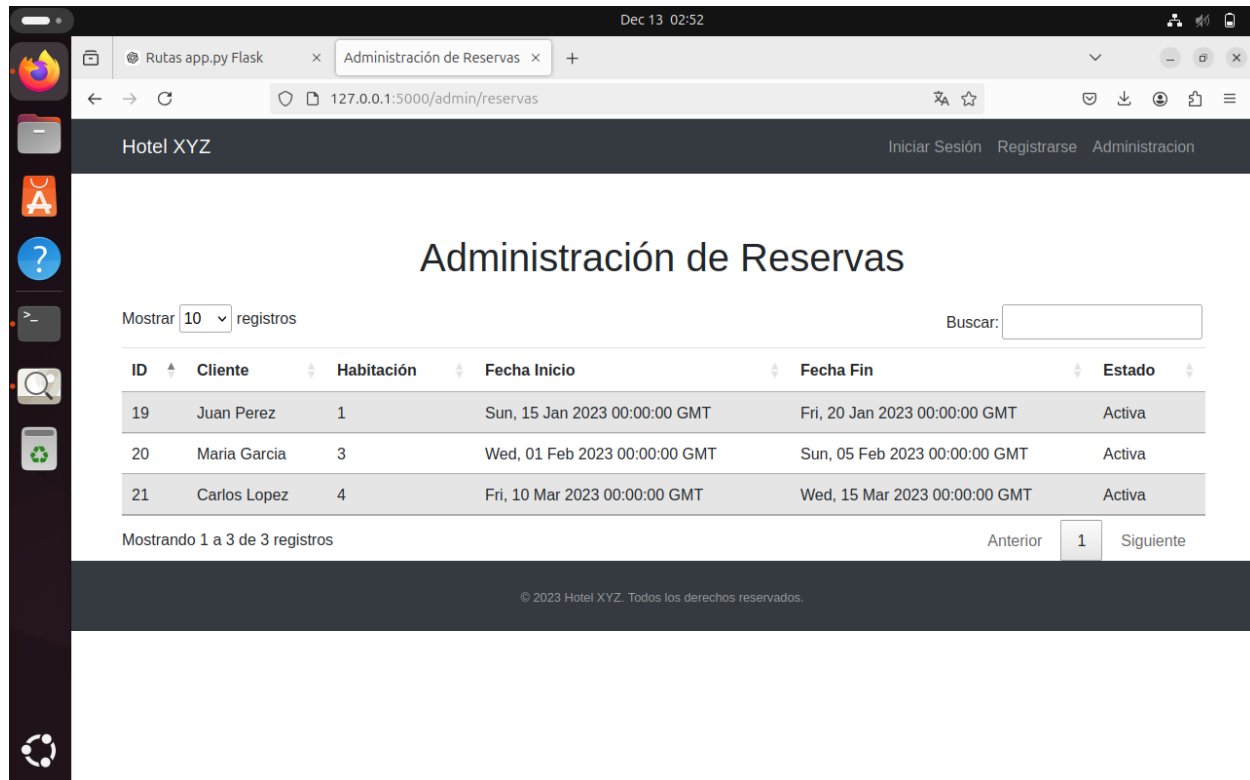
## Administración de Habitaciones

Mostrar 10 registros Buscar:

ID	Número	Tipo	Precio	Estado
1	101	Sencilla	50	Disponible
2	102	doble	75	Disponible
3	201	Matrimonial	100	Disponible
4	202	Suite	150	Disponible
5	301	Presidencial	250	Disponible

Mostrando 1 a 5 de 5 registros Anterior 1 Siguiente

© 2023 Hotel XYZ. Todos los derechos reservados.



## Retrospectiva del Trabajo

### Preguntas Clave

1. **¿Qué salió bien?**
  - o Excelente colaboración entre los miembros del equipo.
  - o Cumplimiento de los objetivos en el tiempo estipulado.
2. **¿Qué podría mejorar?**
  - o Mayor tiempo dedicado a pruebas de usuario.
3. **¿Qué aprendimos?**
  - o La importancia de la planificación y la comunicación.

## **Conclusión**

El desarrollo del Sistema de Reservas de Hotel ha sido un ejercicio integral que permitió aplicar conceptos clave de arquitectura de software, diseño de bases de datos, desarrollo web y habilidades de autogestión. A lo largo del proyecto, se logró crear un sistema funcional que cumple con los objetivos SMART definidos al inicio.

Aunque se cumplieron los objetivos planteados, se identificaron áreas de mejora, como la necesidad de mayor tiempo para pruebas de usuario y optimización del rendimiento de ciertas funcionalidades. Estas áreas representan oportunidades valiosas para futuros proyectos personales o colaborativos.

En general, el proyecto destacó por su capacidad de integrar diferentes tecnologías y por demostrar la importancia de una planificación sólida y una ejecución disciplinada, sentando un precedente positivo para futuros desarrollos de sistemas complejos.