

TRABAJO FINAL DE GRADO

**POR: JOSE DEL PINO
CASTILLO**

CONTENIDOS

1

Introducción

2

Análisis de requisitos

3

Diseño

4

Codificación

5

Estructura del manual de instalación y configuración

6

Conclusiones

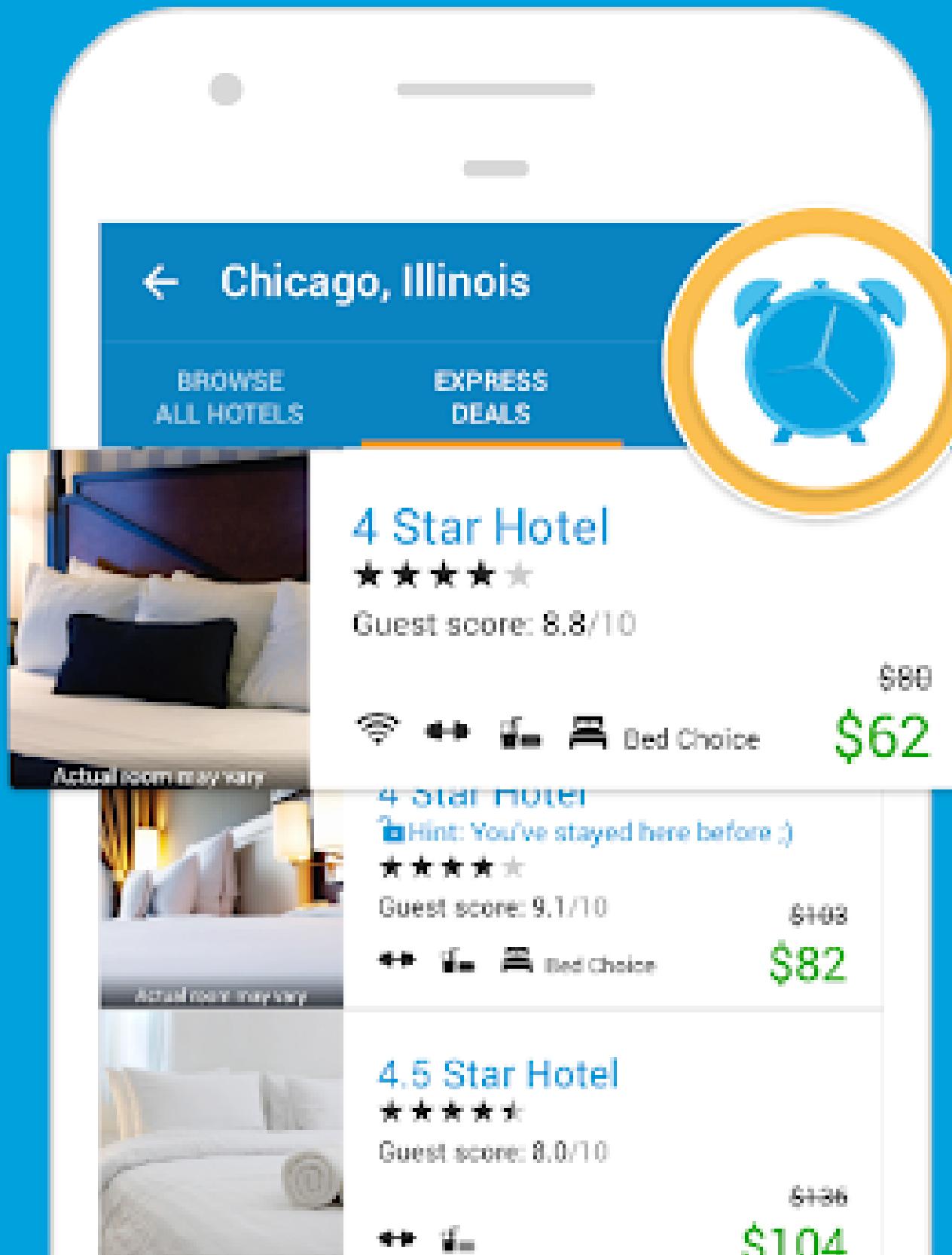
7

Trabajo futuro

8

Bibliografía y webgrafía.

RESERVA VIAJES DE ÚLTIMO MINUTO



INTRODUCCIÓN

El proyecto consiste en el desarrollo de una plataforma de reserva de hoteles, que permitirá a los usuarios buscar, comparar y reservar habitaciones de hotel de manera fácil y conveniente. La plataforma estará diseñada para ofrecer una experiencia intuitiva y eficiente tanto para los clientes como para los administradores de hoteles.

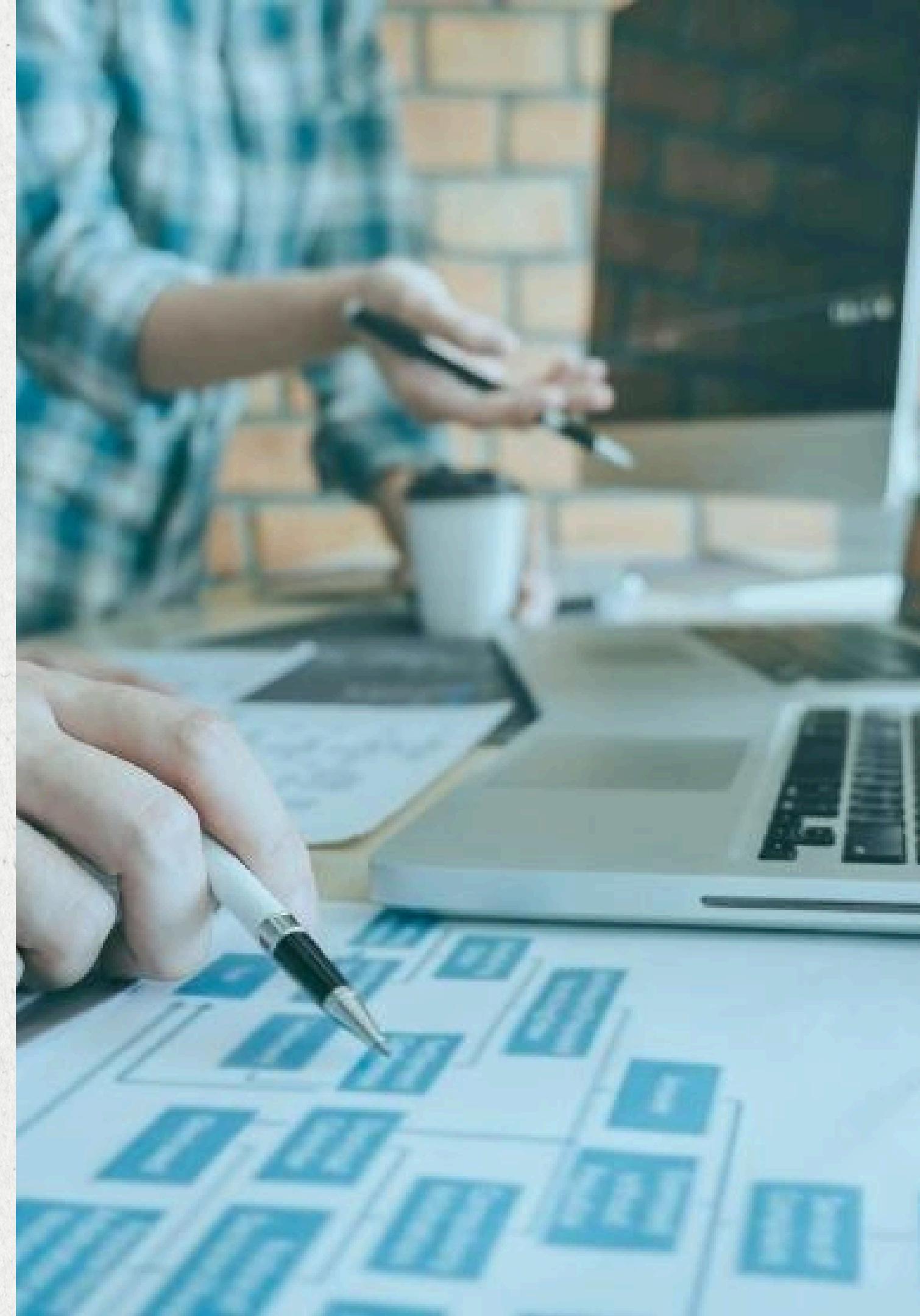
PRINCIPALES OBJETIVOS

- **Facilitar la reserva de habitaciones:** La plataforma permitirá a los usuarios buscar disponibilidad de habitaciones en diferentes hoteles, visualizar detalles de las habitaciones y realizar reservas de manera rápida y sencilla.
- **Optimizar la gestión de hoteles:** Para los administradores de hoteles, la plataforma ofrecerá herramientas para gestionar la disponibilidad de habitaciones, precios, promociones y reservas de manera eficiente.
- **Mejorar la experiencia del cliente:** Se priorizará la experiencia del usuario en la plataforma, asegurando un diseño atractivo, navegación intuitiva y procesos de reserva transparentes. Además, se ofrecerá soporte al cliente para resolver consultas y problemas de manera rápida y efectiva.

ANÁLISIS DE REQUISITOS

El desarrollo y funcionamiento del proyecto de reserva de hoteles requiere el cumplimiento de una serie de requisitos. A continuación, se detallan los requisitos necesarios:

- 1. Registro de usuarios:** La plataforma debe permitir a los usuarios registrarse creando una cuenta con información básica como nombre, correo electrónico y contraseña.
- 2. Inicio de sesión:** Los usuarios registrados deben poder iniciar sesión en la plataforma utilizando su correo electrónico y contraseña.
- 3. Búsqueda de hoteles:** La plataforma debe ofrecer a los usuarios la posibilidad de buscar hoteles por ubicación.
- 4. Visualización de detalles del hotel:** Una vez realizada la búsqueda, los usuarios deben poder ver detalles de los hoteles disponibles.
- 5. Reserva de habitaciones:** Los usuarios deben poder seleccionar las fechas de entrada y salida.
- 6. Gestión de reservas:** Los usuarios registrados deben poder acceder a su historial de reservas.
- 7. Diseño responsivo:** La plataforma debe ser compatible con dispositivos móviles y tabletas.



DISEÑO

I. DISEÑO DE UI/UX

2. DATOS

3. ALMACENAMIENTO

4. DIAGRAMA UML

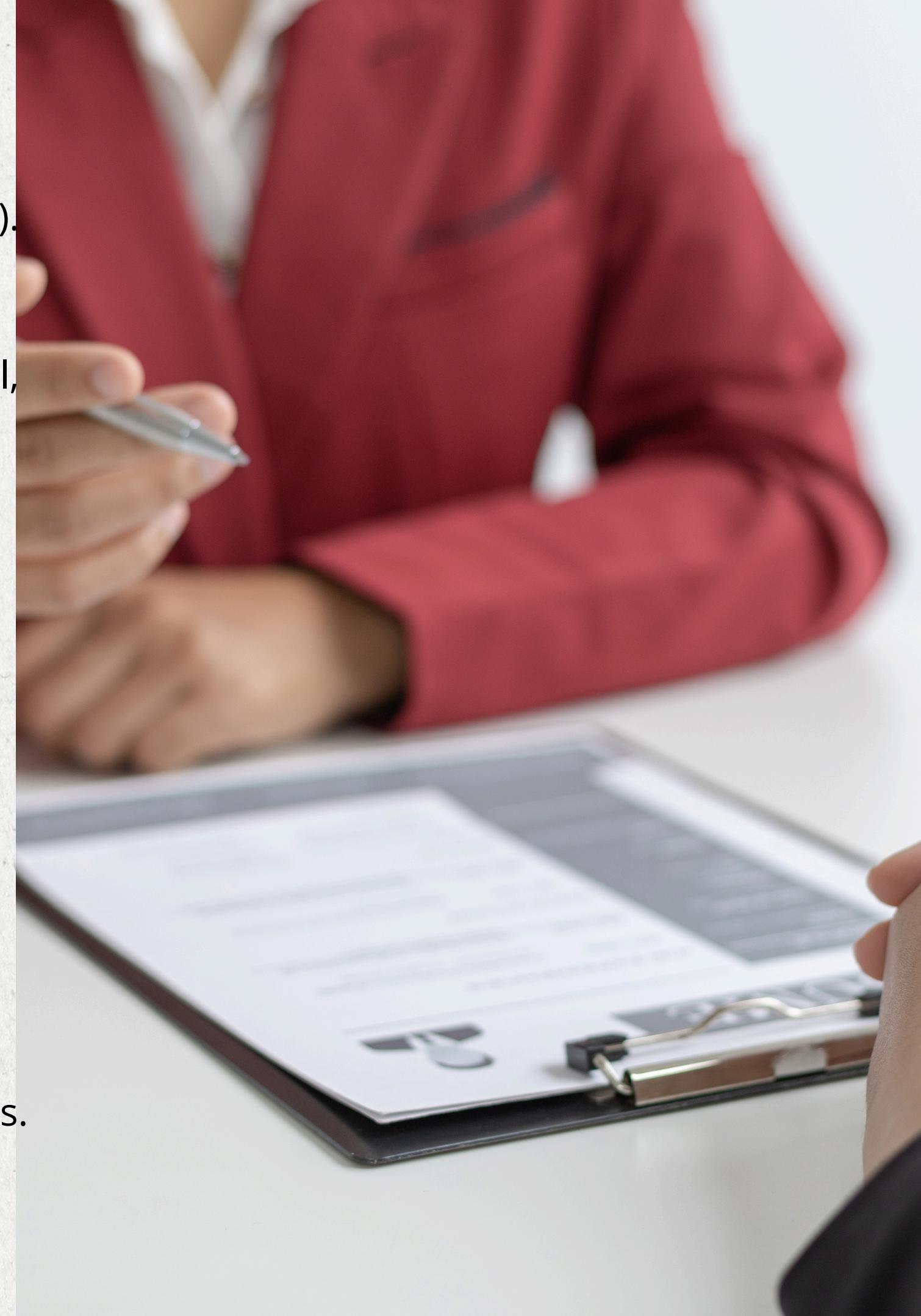
El diseño del sistema de reservas de hoteles se centra en ofrecer una interfaz de usuario (UI) y una experiencia de usuario (UX) intuitiva y atractiva, optimizando la interacción con la plataforma.



Tu viaje de ensueño, a un solo clic

I. DISEÑO DE UI/UX

- **Página de inicio:**
 - Barra de búsqueda para encontrar hoteles por ubicación.
 - Banner principal con buscador de hoteles (ubicación, fechas, etc...).
 - Lugares más buscados.
- **Página de búsqueda:**
 - Resultados de búsqueda en tarjetas con imagen, nombre del hotel, ubicación, precio por noche y puntuación.
 - Botón para ver más detalles de cada hotel.
- **Página de detalles del hotel:**
 - Imágenes del hotel en un carrusel.
 - Descripción detallada del hotel.
 - Disponibilidad de habitaciones con precios.
 - Ubicación del hotel.
 - Botón para reservar.
- **Página de reserva:**
 - Formulario de reserva con fechas de estadía, número de habitaciones, precio total calculado.
 - Botón para confirmar reserva.
- **Página de perfil de usuario:**
 - Historial de reservas con opciones para ver detalles de las reservas.



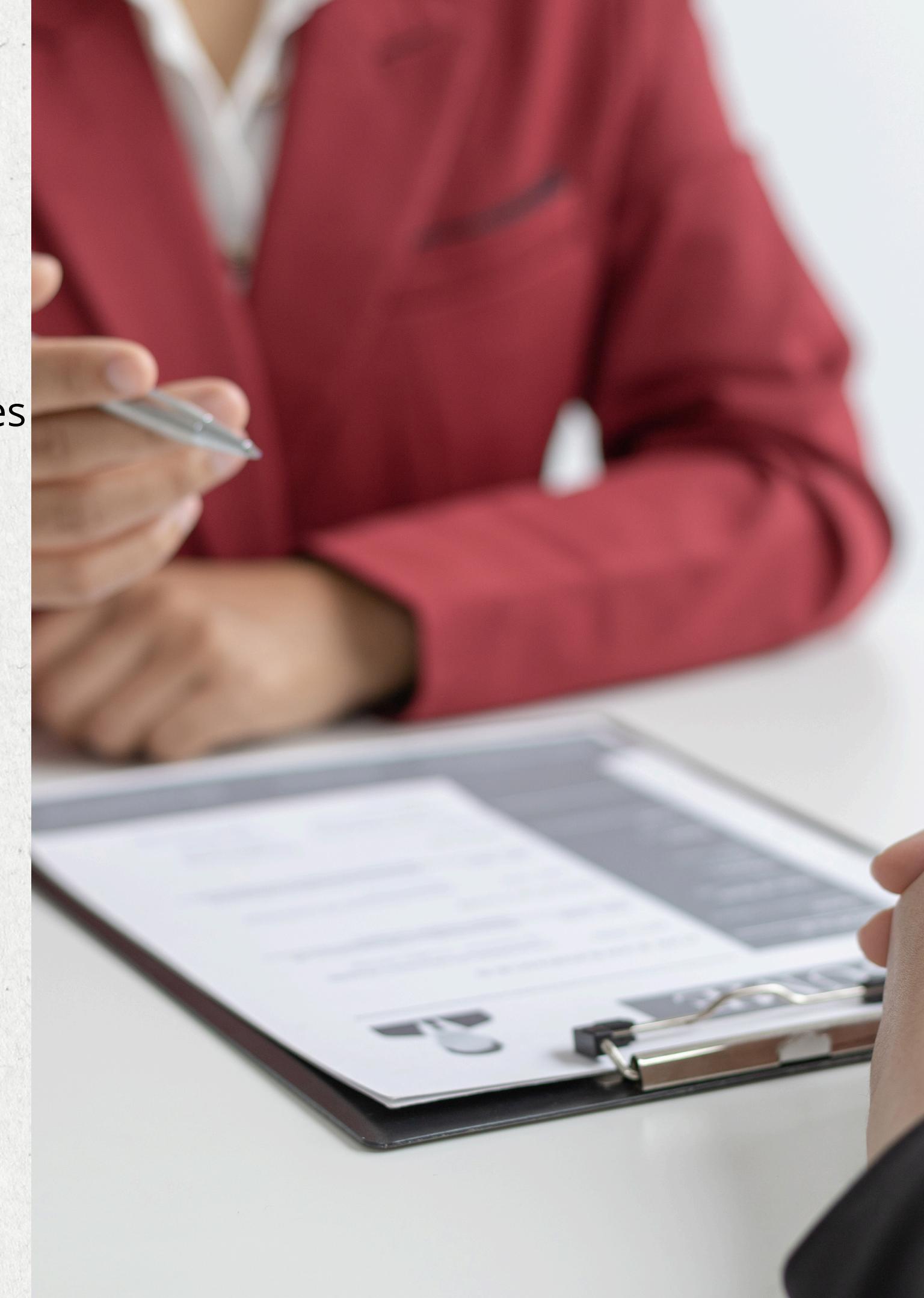
2. DATOS

- **2.1 Modelos de datos:**

- **Cliente:**
 - ID, Nombre, Email, Contraseña, Edad, Nacionalidad, Idioma.
- **Hotel:**
 - ID, Nombre, Ubicación, Precio por noche, N° Habitaciones Totales, Calificación, Imágenes, URL Imágenes.
- **Reserva:**
 - ID, Fecha de inicio, Fecha de fin, Precio total, Número de habitaciones, Cliente (ID de cliente, referenciado), Hotel (ID de hotel, referenciado).

- **2.2 Relación entre datos:**

- Un Cliente puede tener múltiples Reservas.
- Un Hotel puede tener múltiples Reservas.
- Una Reserva está asociada a un Cliente y a un Hotel.



3. ALMACENAMIENTO

- **3.1 Tablas de la base de datos:**

- **Ciudadanos:**

- id (PK, AUTO_INCREMENT), nombre (VARCHAR), email (VARCHAR, UNIQUE), contraseña (VARCHAR), edad (INTEGER), nacionalidad (VARCHAR), idioma (VARCHAR).

- **Hoteles:**

- id (PK, AUTO_INCREMENT), nombre (VARCHAR), ubicacion (VARCHAR), precio_noche (INTEGER), nº habitaciones totales (INTEGER), calificacion (INTEGER), imagenes (TEXT), url_imagen (TEXT).

- **Reservas:**

- id (PK, AUTO_INCREMENT), fecha_inicio (DATE), fecha_fin (DATE), precio_total (INTEGER), num_habitaciones (INTEGER), cliente_id (FK, REFERENCES Ciudadanos(id)), hotel_id (FK, REFERENCES Hoteles(id)).

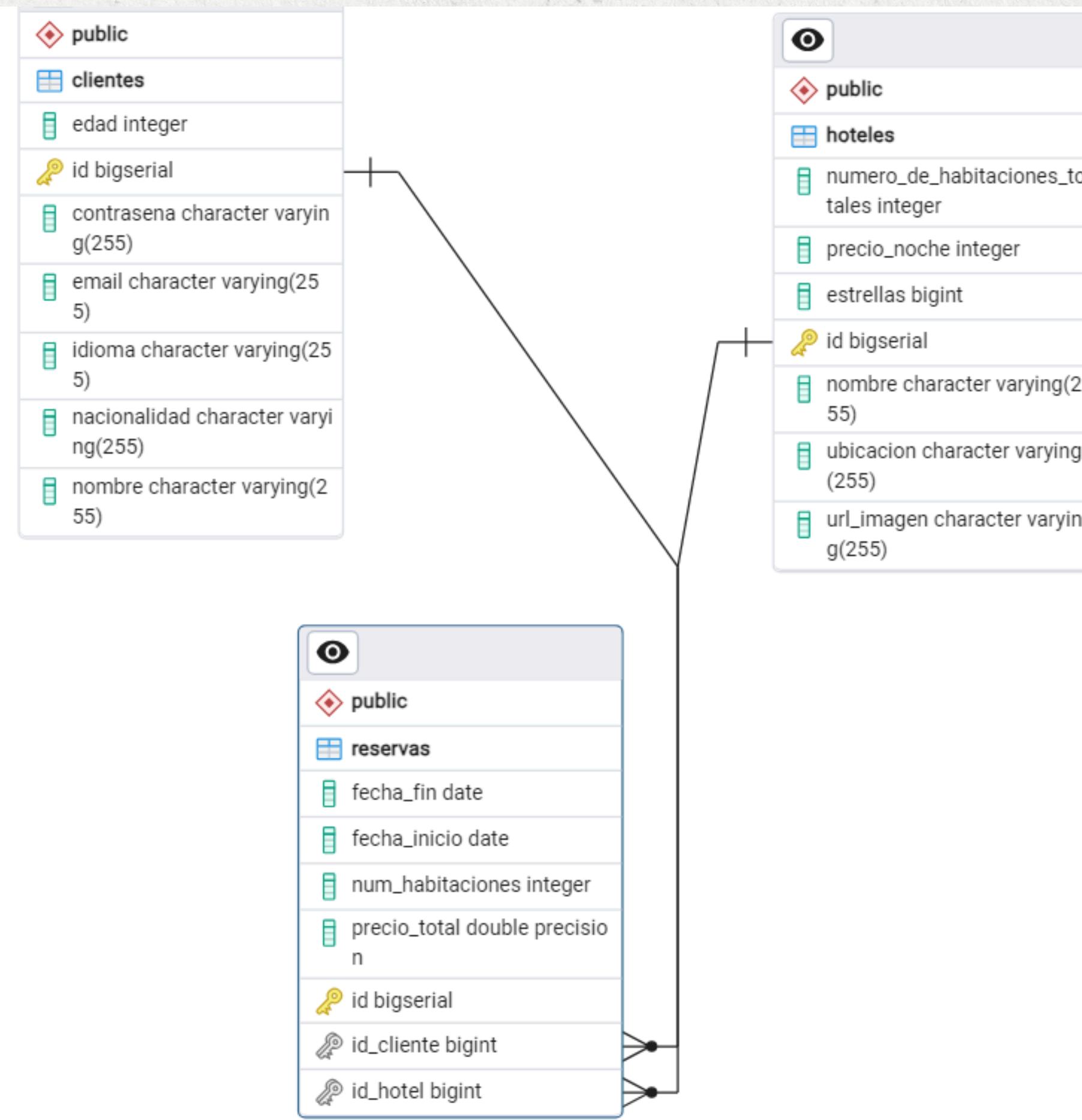
- **3.2 Almacenamiento de imágenes:**

- Las imágenes de los hoteles se almacenan como URLs en la columna url_imagenes de la tabla Hoteles, facilitando su acceso y gestión.



4. DIAGRAMA UML

La siguiente foto es el diagrama UML de la base de datos empleada para la función de esta aplicación:



CODIFICACIÓN

BACKEND

FRONTED

BASE DE DATOS

HERRAMIENTAS

La elección de lenguajes y herramientas es fundamental para el desarrollo efectivo del proyecto. A continuación, se destacan las tecnologías seleccionadas y las razones detrás de su elección:

```
@GetMapping("listar")
public ResponseEntity<List<Cliente>> listarClientes() {
    List<Cliente> clientes = clienteRepository.findAll();
    if (clientes.isEmpty()) {
        return new ResponseEntity<>( HttpStatus.NO_CONTENT );
    }
    return new ResponseEntity<>( clientes, HttpStatus.OK );
}
```

BACKEND

- **Java con Spring Boot:**

- **Robustez y Escalabilidad:** Java es conocido por su robustez y capacidad para manejar aplicaciones grandes y complejas. Spring Boot facilita la creación de aplicaciones independientes, productivas y listas para producción.
- **Seguridad:** Proporciona características integradas de seguridad y es compatible con una amplia gama de herramientas de seguridad.
- **Ecosistema y Soporte:** La comunidad de Spring Boot es amplia, ofreciendo abundante documentación y recursos, y se integra bien con otros componentes del ecosistema Spring.

- **Razón:**

- Escogí Spring Boot para el back a parte de lo mencionado anteriormente, porque quería aprender una nueva tecnología la cual en el mercado laboral se utiliza mucho a día de hoy.



FRONTED

- **Vue.js:**

- **Componentización:** Permite dividir la interfaz de usuario en componentes reutilizables, facilitando la gestión y escalabilidad del código.
- **Simplicidad y Facilidad de Uso:** Conocido por su curva de aprendizaje suave y su sintaxis intuitiva, accesible para desarrolladores principiantes y experimentados.
- **Performance:** Vue.js es ligero y ofrece un rendimiento rápido, crucial para una experiencia de usuario fluida.
- **Popularidad y Comunidad:** Tiene una comunidad creciente y muchas bibliotecas y herramientas de terceros, facilitando encontrar soluciones y apoyo.

- **Razón:**

- Escogí Vue.js porque lo aprendí durante el curso y me resultó bastante fácil a comparación de otros frameworks como Angular, React, etc...
- También es un framework que está en auge hoy en día y cada vez se utiliza más.



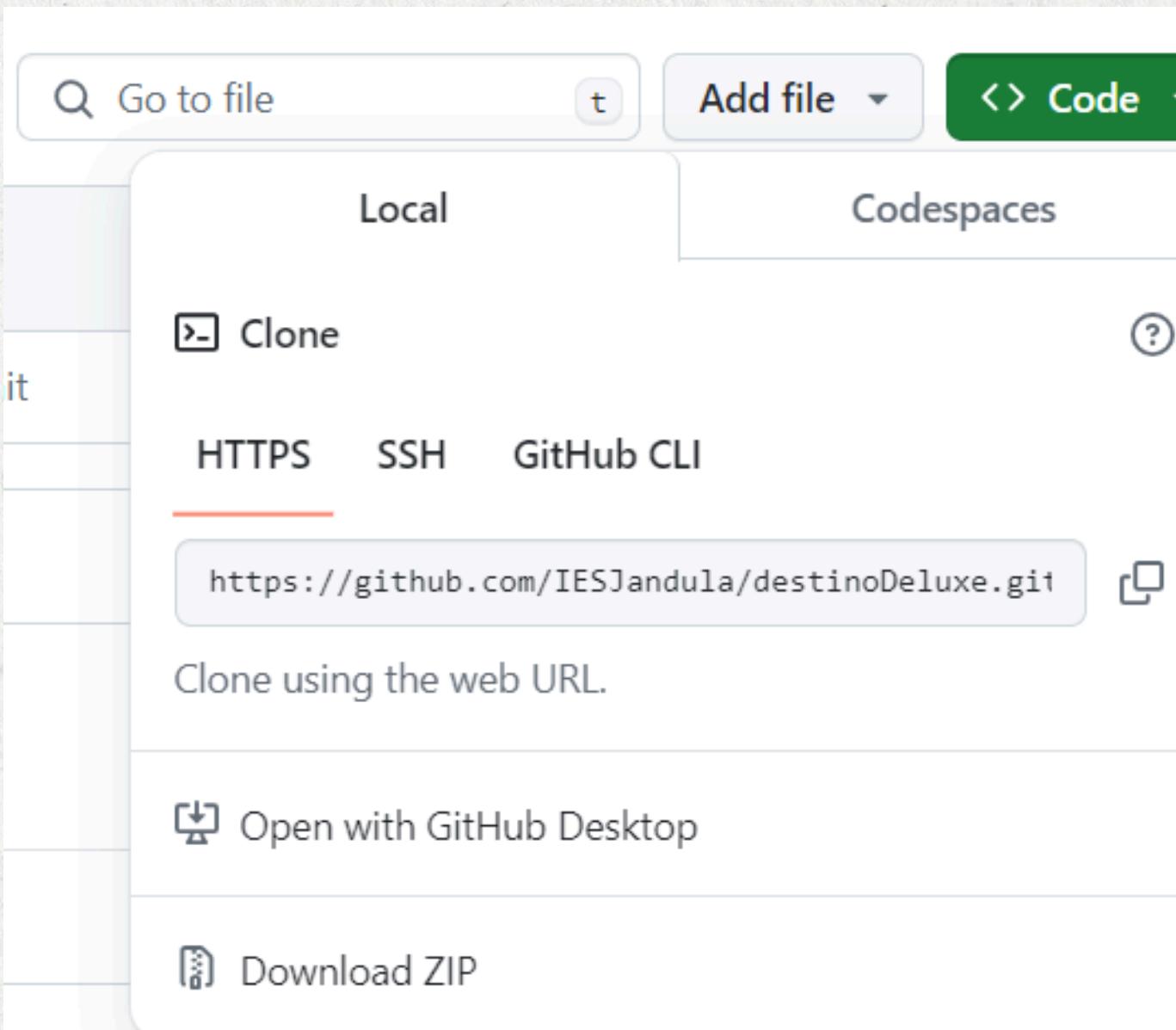
BASE DE DATOS

- **PostgreSQL:**
 - **Relacional y Open Source:** Conocida por su robustez y conformidad con el estándar SQL.
 - **Funciones Avanzadas:** Ofrece características avanzadas como transacciones ACID, replicación y concurrencia avanzada.
 - **Escalabilidad y Rendimiento:** Maneja eficientemente grandes volúmenes de datos y puede escalar según las necesidades del proyecto.
- **Razón:**
 - Elegí PostgreSQL porque mi idea de un principio era desplegar mi base de datos, pero al final no lo hice.
 - A parte también quería aprender otro motor de base de datos que no sea MySQL que ya lo había visto durante el curso.
 - Y hoy en día se utiliza mucho PostgreSQL.



ESTRUCTURA DEL MANUAL DE INSTALACIÓN Y CONFIGURACIÓN

Lo primero que debemos hacer es ir al repositorio y copiar el enlace del repositorio para despues poder realizar la clonación, un ejemplo de como copiar el enlace:



ESTRUCTURA DEL MANUAL DE INSTALACIÓN Y CONFIGURACIÓN

Una vez tenemos copiado el enlace ejecutamos el siguiente comando:

```
git clone https://github.com/IESJandula/destinoDeluxe.git
```

BACKEND:

- Prerequisitos: Instalación de JDK y Maven.
- Configuración de Spring Boot: Variables de entorno y archivo application.properties o application.yml.
- Base de Datos: Instalación de PostgreSQL, creación de la base de datos y configuración de las credenciales en Spring Boot.
- Ejecución: Instrucciones para abrir el proyecto en IntelliJ y ejecutarlo.

FRONTEDE:

- Prerequisitos: Instalación de Node.js y npm.
- Instalación de Dependencias: Comandos para instalar las dependencias del proyecto.
- Configuración: Variables de entorno y archivos de configuración.
- Ejecutamos los siguientes comandos:
 - npm install
 - npm run dev
- Visualización: La URL local para visualizar el proyecto: <http://localhost:5173>.

CONCLUSIONES

Desarrollo

1. Planificación y Diseño

- Requisitos: Definición clara de necesidades, cubriendo todas las funcionalidades esenciales.
- Diseño UI/UX: Uso de Vue.js para una interfaz moderna, dinámica e intuitiva.

2. Tecnologías Utilizadas

- Backend: Spring Boot para una estructura robusta y escalable.
- Frontend: Vue.js para una interfaz interactiva y de fácil integración.
- Base de Datos: PostgreSQL para un almacenamiento confiable y eficiente.

3. Desafíos y Soluciones

- Integración: Detalle en la configuración de APIs y manejo de datos

Desempeño

1. Usabilidad

- Interfaz: Navegación intuitiva que facilita la gestión de reservas y consulta de disponibilidad.
- Accesibilidad: Consideraciones para asegurar el uso por una amplia gama de usuarios.

2. Escalabilidad

- Estructura Modular: Facilita la adición de nuevas funcionalidades y mejora las existentes sin afectar el rendimiento.
- Base de Datos: PostgreSQL permite la gestión eficiente de un mayor volumen de datos, adaptándose a futuras expansiones.

TRABAJO FUTURO

- **Integración con Servicios de Pago**
 - **Descripción:** Implementar integración con servicios como PayPal y Stripe.
 - **Beneficios:** Facilita el proceso de reserva, ofrece pagos seguros y nuevas oportunidades de ingresos.
- **Soporte Multilingüe**
 - **Descripción:** Añadir soporte para múltiples idiomas.
 - **Beneficios:** Atrae a una audiencia global y mejora la accesibilidad.
- **Sistema de Recomendaciones**
 - **Descripción:** Desarrollar recomendaciones basadas en preferencias y historial.
 - **Beneficios:** Personaliza la experiencia y aumenta la tasa de conversión.
- **Mejoras en Seguridad**
 - **Descripción:** Implementar autenticación de dos factores, encriptación y auditorías de seguridad.
 - **Beneficios:** Protege información sensible y cumple con normativas de seguridad.
- **Ampliación de Funcionalidades de Gestión de Hoteles**
 - **Descripción:** Añadir administración de disponibilidad de habitaciones, tarifas dinámicas, y gestión de hoteles.
 - **Beneficios:** Ofrece herramientas avanzadas para hoteleros y mejora la competitividad de la plataforma.

BIBLIOGRAFIA Y WEBGRAFÍA

- Documentación de Spring
 - Fuente: [Spring Framework Reference Documentation](#)
 - Descripción: Guías y ejemplos detallados sobre las características y módulos del framework Spring.
- Documentación de PostgreSQL
 - Fuente: [PostgreSQL Global Development Group](#)
 - Descripción: Información desde la instalación y configuración hasta características avanzadas y mejores prácticas de PostgreSQL.
- Documentación de Vue.js
 - Fuente: [Vue.js - The Progressive JavaScript Framework](#)
 - Descripción: Guías y ejemplos para el desarrollo de interfaces de usuario interactivas y dinámicas con Vue.js.
- W3Schools
 - Fuente: [W3Schools Online Web Tutorials](#)
 - Descripción: Tutoriales y referencias sobre lenguajes de programación web y desarrollo front-end.
- GitHub
 - Fuente: [GitHub Repositories](#)
 - Descripción: Utilizado para gestionar y versionar el código fuente del proyecto, facilitando la colaboración y el control de versiones.
- ChatGPT de OpenAI
 - Uso: Consultado para obtener explicaciones y resolver dudas específicas durante el desarrollo del proyecto.

PREGUNTAS Y RESPUESTAS

¿Alguna pregunta?



¡GRACIAS!

Jose Del Pino Castillo