

**Ciclo: CFGS Desarrollo de Aplicaciones Web**

**Centro: CIFP Pau Casesnoves**

**Proyecto: Web de Hotel “Aura de Mallorca”**

**Alumno: Fernando González Serrano**

**Curso académico: 2023-2024**

## Índice de contenidos

|  |    |
|--|----|
| DATOS PERSONALES, DEL CENTRO Y DE LA EMPRESA.....  | 3  |
| INTRODUCCIÓN.....  | 6  |
| Introducción a la memoria, justificación del proyecto.....   | 6  |
| Objetivos generales, beneficios, motivaciones personales.....  | 7  |
| PLANIFICACIÓN TEMPORAL DEL PROYECTO.....   | 9  |
| Relación prevista de las tareas a realizar en cada fase del proyecto (análisis, diseño, programación, pruebas, producción) y el lugar previsto de realización (empresa, casa o instituto)..... | 9  |
| <i>Cronograma inicial previsto de tareas de cada fase: análisis, diseño, programación, pruebas, producción.</i> .....  | 10 |
| ANÁLISIS DE LA COMPETENCIA.....  | 12 |
| Situación actual: productos relacionados en el mercado y su coste.....   | 12 |
| <i>Ventajas e inconvenientes de los productos del mercado en relación al producto previsto en el proyecto.</i> .....   | 13 |
| RECURSOS DE SOFTWARE PARA EL DESARROLLO.....   | 15 |
| Lenguajes.....   | 15 |
| Frameworks.....  | 15 |
| Librerías.....   | 15 |
| <i>Herramientas para el despliegue.</i> .....  | 16 |
| Bases de datos.....  | 16 |
| FASE DE ANÁLISIS.....  | 17 |
| Requerimientos de la aplicación.....   | 17 |
| Modelo E/R de la base de datos.....  | 19 |
| Diagrama de casos de uso.....  | 20 |
| FASE DE ANÁLISIS.....  | 21 |
| Modelo Relacional de la Base de Datos.....   | 21 |
| Diseño de la interfaz del usuario.....   | 22 |
| Modelo estático del sistema (Diagrama de Clases).....  | 24 |
| Diseño de la infraestructura física.....   | 25 |
| Wireframe.....   | 26 |
| FASE DE PRUEBAS.....   | 30 |
| ANÁLISIS DE RECURSOS Y COSTES.....   | 33 |
| RELACIÓN FINAL DEL CUMPLIMIENTO DE LAS DIFERENTES TAREAS DE CADA FASE DEL PROYECTO.....  | 35 |
| CONCLUSIONES Y OPINIÓN PERSONAL.....   | 40 |
| POSIBLE EVOLUCIÓN Y MEJORA DEL PROYECTO.....   | 42 |
| Anexos.....  | 45 |

## DATOS PERSONALES, DEL CENTRO Y DE LA EMPRESA

El nombre del alumno es Fernando González Serrano, quien realizará el proyecto. Su correo electrónico es nano9gs@hotmail.es. Nació el 27/06/1999 y actualmente reside a la localidad de Palma (Mallorca).

Su nivel de estudios es de CFGS en desarrollo de aplicaciones multiplataforma, curso de especialización en ciberseguridad y hacking ético, bachillerato científico / tecnológico... Además, tiene experiencia en programación con lenguajes orientados a objetos, desarrollo de apps web fullstack, y sobre todo, desarrollo de apps móviles Android con Kotlin, Jetpack Compose con mejores prácticas y últimas tecnologías.

El centro donde el alumno está estudiando el Grado Superior en Desarrollo de Aplicaciones Web es en el CIFP Pau Casesnoves, situado en Inca.

Los nombres de los tutores encargados del alumno autor de este proyecto final del CFGS de Desarrollo de Apps<sup>2</sup> Web son Carles Canals Gozalvez y Margalida Estrany Bonnin. Carles es el tutor principal de FCT y proyecto, mientras que Margalida es la ayudante con potestad para asignar también a alumnos a empresas, ella fue la encargada de todos los alumnos con FCT en Inca, mientras que Carles a todos los de Palma, y como a Fernando le tocó en Inca, fue con Marga a formalizar el contrato de prácticas.

---

<sup>2</sup> Apps: Aplicaciones

Sobre la empresa donde se realizó el proyecto y FCT:

- El nombre de la empresa donde el alumno trabaja: Homerti<sup>3</sup> (Ca S'amtger SL), una sociedad limitada. Su objetivo es ofrecer un servicio completo para la gestión de alquileres vacacionales, centrándose en la gestión de casas. Tiene la sede principal a la localidad de Inca.
- La estructura organizativa de la empresa se divide en diversas marcas que asumen diferentes funciones y roles, las cuales se dividen a la vez en departamentos y empleados, dirigidos por un jefe o cabeza de departamento.
- Dentro de la empresa el alumno ocupa el puesto de Programador Web, con relación al ciclo que se está cursando, realizando tareas de programación web, despliegue, “testing<sup>4</sup>”... La empresa está involucrada en el proyecto en el sentido de que pidió al alumno un gestor de traducciones genérico para utilizarlo ellos, y el alumno Fernando decidió hacerlo en dos proyectos aparte, siendo el principal éste del hotel Aura de Mallorca (el gestor no es entregado, es aparte).

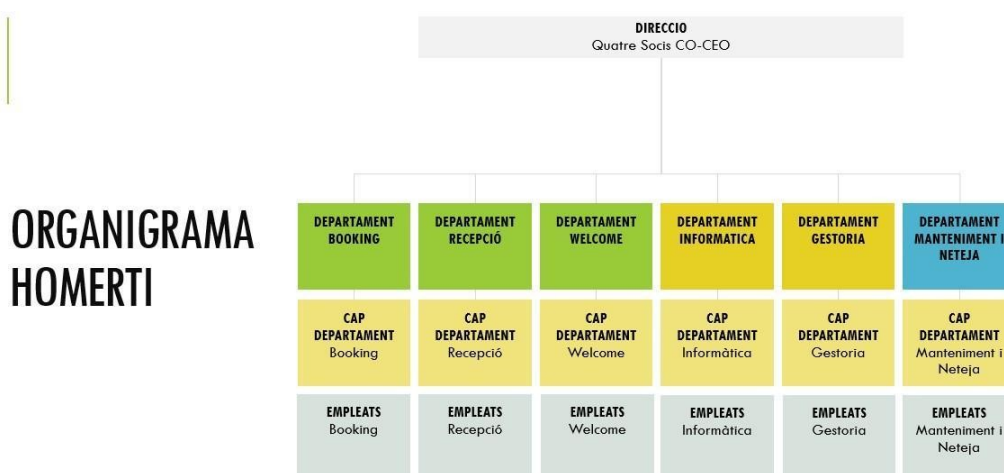


Figura 1: Organigrama de la empresa "Homerti", con más de 100 trabajadores.

<sup>3</sup> Homerti es una empresa muy reconocida en el sector de alquiler turístico fundada por 4 mallorquines.

<sup>4</sup> “testing” se refiere al proceso de probar y confirmar que un sistema o servicio funciona como debe.

Describiendo un poco la jerarquía de la empresa, nos encontramos con:

- Dirección: 4 socios principales con el rol de Co-CEO y Co-Fundador, encargados de gestionar a los jefes de departamento y el funcionamiento general de la empresa.
- Departamento de Booking: cuenta con el jefe de departamento dirigiendo a sus empleados, encargados de gestionar las reservas que entran y sus clientes.
- Departamento de Recepción: cuenta con el jefe de departamento dirigiendo a los empleados de recepción, que se encargan de hacer gestiones con los propietarios y arreglar incidentes que pueda haber.
- Departamento de Marketing y RRPP<sup>5</sup>: Con su jefe y empleados también, se encargan de gestionar e incorporar nuevos propietarios y nuevas casas a la empresa, además de promocionar a la empresa en redes y sitios web.
- Departamento de Informática: con su jefe de departamento y empleados, de igual forma, se encargan de programar, gestionar y llevar el mantenimiento de todos lo relativo al Software de la empresa.
- Departamento de Gestoría: con jefe y empleados de departamento, se encargan de realizar todas las tareas administrativas y de gestionar todas las facturas de la empresa, contabilidad...
- Departamento de Mantenimiento y Limpieza: jefe y empleados de departamento encargados de realizar la limpieza de las casas, los locales de la empresa y, además, arreglar incidencias pertinentes de las casas y de la propia empresa.

---

5 “RRPP”: Relaciones Públicas.

## INTRODUCCIÓN

### *Introducción a la memoria, justificación del proyecto.*

El desarrollo de una web completa para el Hotel Aura de Mallorca es un proyecto ambicioso que requiere una planificación cuidadosa y una ejecución efectiva. En esta sección, se presentará una descripción general del proyecto, sus objetivos y los beneficios que se esperan obtener.

- *Descripción del proyecto*

El Hotel Aura de Mallorca es un **hotel** de cinco estrellas ubicado en la isla de Mallorca, España. El hotel ofrece una amplia gama de servicios y comodidades, que incluyen habitaciones y suites lujosas, restaurantes “gourmet”, piscinas, “spa”, gimnasio y mucho más.

El proyecto consiste en el desarrollo de una web completa para el hotel que cumpla con los siguientes requisitos:

- Un sistema de **reservas** que permita a los clientes reservar habitaciones y servicios del hotel.
- Un sistema de **cuentas** que permita a los clientes gestionar sus reservas y preferencias.
- Un conjunto de características **diferenciadoras** que hagan que la web sea única y atractiva para los clientes.

- *Características diferenciadoras*

Las características diferenciadoras de la web incluyen:

- La **integración** con servicios externos al hotel, como el jardín registrado, enorme y precioso del hotel.
- Un **sistema multiidioma** que permita a los clientes ver la web en su idioma preferido.
- Una característica de **servicio** de consulta **meteorológica** en la fecha que se va a reservar para ver si puede reservar (se mira la fecha de check-in).
- La posibilidad de **registrarse** mediante **código QR** vinculado al email del usuario.

## *Objetivos generales, beneficios, motivaciones personales...*

### **Objetivos del proyecto**

Elegí este proyecto porque me apasiona el desarrollo web y tengo un interés genuino en trabajar en el sector hotelero. Creo firmemente que esta página web completa para el Hotel "Aura de Mallorca" puede convertirse en una herramienta sumamente beneficiosa. Los principales **objetivos** para el proyecto son:

#### **1. Desarrollo integral de la web**

- Crear una plataforma completa que incluya sistema de reservas, cuentas y características diferenciadoras de otras webs hoteleras.
- Incorporar servicios externos vinculados al hotel, como su propio jardín registrado, integrándolos de manera efectiva en la web.

#### **2. Multilingüismo**

- Implementar un sistema multiidioma para llegar a un público más amplio.

#### **3. Beneficios y motivaciones**

- Mejorar la presencia en internet del Hotel "Aura de Mallorca".
- Ofrecer un servicio más completo a los clientes, facilitando la reserva y proporcionando información detallada sobre los servicios y el entorno.

## Objetivos empresariales

### 1. Mejora de Presencia en Internet:

Posicionar el Hotel "Aura de Mallorca" de manera destacada en el ámbito "online".

### 2. Optimización de Servicios:

Ofrecer un sistema de reservas eficiente y fácil de usar para los clientes.

### 3. Diferenciación Competitiva:

Incorporar características únicas, como el jardín registrado, para destacar frente a otros competidores.

### 4. Internacionalización:

Facilitar la experiencia a clientes de diferentes idiomas mediante el sistema multidioma.

## Objetivos personales

### 1. Adquisición de Conocimientos:

Desarrollar habilidades prácticas en diseño, desarrollo e implementación web, explorando lenguajes más allá de los estudiados previamente.

### 2. Solución Práctica y Útil:

Demostrar la capacidad para crear una solución efectiva para una necesidad real en el sector hotelero.

### 3. Resultado Profesional:

Alcanzar un producto final de alta calidad para el beneficio y satisfacción tanto del Hotel "Aura de Mallorca" como de sus clientes.

### 4. Gestión Eficiente de Recursos:

Aprender y aplicar métodos eficientes en la gestión del tiempo y recursos para cumplir con los plazos establecidos.

### 5. Autodidactismo Continuo:

Aprender de forma autodidacta nuevos lenguajes y aplicarlos efectivamente al proyecto, ampliando constantemente las habilidades técnicas.



## PLANIFICACIÓN TEMPORAL DEL PROYECTO

*Relación prevista de las tareas a realizar en cada fase del proyecto (análisis, diseño, programación, pruebas, producción) y el lugar previsto de realización (empresa, casa o instituto)*

Calendario provisional del desarrollo del proyecto es el siguiente (es una primera planificación NO inicial y provisional antes del diagrama de Gantt más específico):

- Análisis y diseño: primeras 2 semanas
  - Estudio de los requisitos del proyecto: Realizado en la empresa Homerti.
  - Definición del diseño de la web: Realizado en la empresa Homerti.
  - Prototipado de la web: Realizado en la empresa Homerti.
- Desarrollo: 4 semanas
  - Desarrollo de la página de inicio: Realizado en la empresa Homerti.
  - Desarrollo de la página de reservas: Realizado en la empresa Homerti.
  - Desarrollo de la página de servicios: Realizado en la empresa Homerti.
  - Desarrollo de la página de contacto: Realizado en la empresa Homerti.
  - Desarrollo de la integración con los servicios externos: Realizado en casa.
- Pruebas: las últimas 2 semanas
  - Pruebas de funcionalidad: Realizado en la empresa Homerti.
  - Pruebas de rendimiento: Realizado en casa.
  - Pruebas de seguridad: Realizado en casa.

El calendario es solo una estimación inicial y puede variar a medida que avance el proyecto. Es importante realizar un seguimiento del progreso del proyecto para asegurarse de que se cumple el calendario previsto.

## Cronograma inicial previsto de tareas de cada fase: análisis, diseño, programación, pruebas, producción.

Dividiremos el desarrollo del proyecto en varias fases y lo haremos mediante un cronograma inicial.

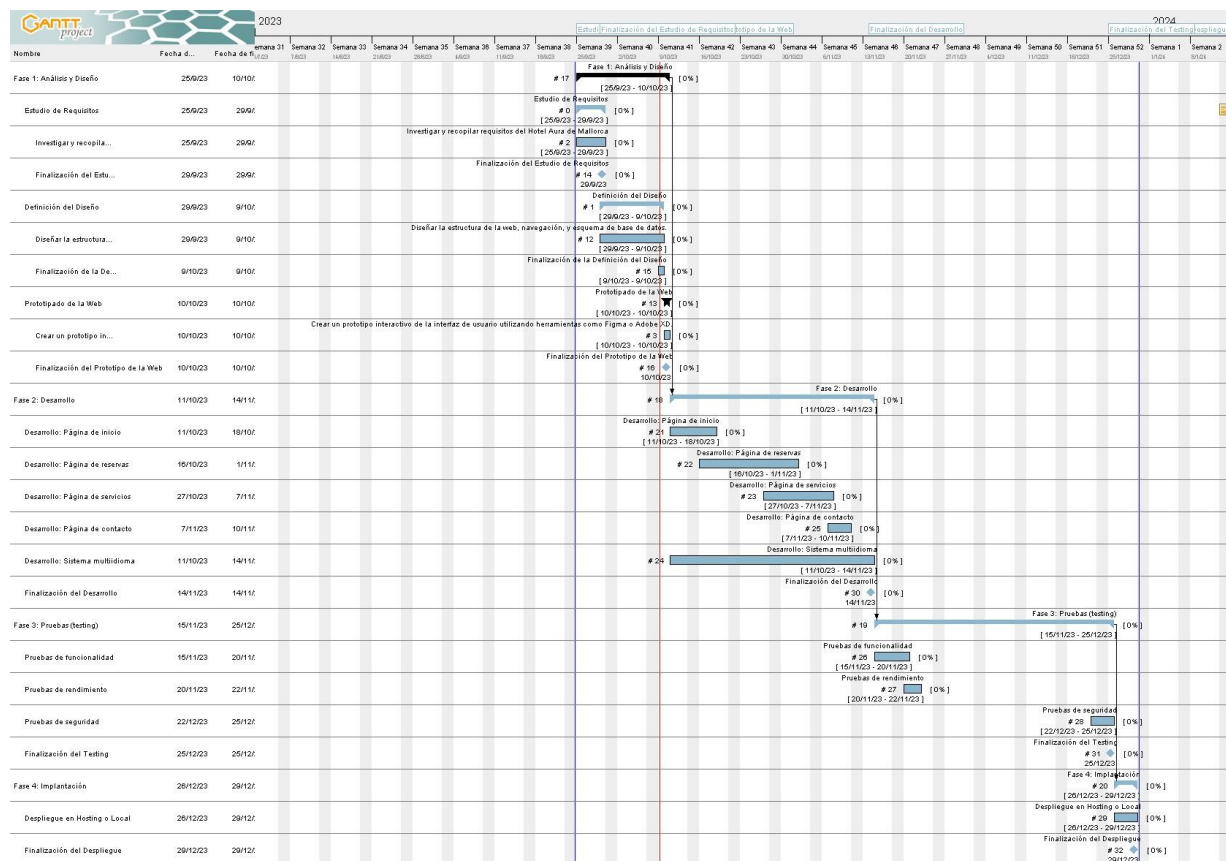


Figura 2: Diagrama de Gantt inicial para el desarrollo del proyecto.

El diagrama de Gantt para el proyecto "Web de Hotel 'Aura de Mallorca'" se divide en varias fases claramente definidas, y cada fase se representa con barras horizontales en el eje temporal. El diagrama muestra las relaciones entre estas fases y proporciona una visión general de la duración total del proyecto.

Las fases de dicho diagrama, como se pueden ver, son:

- Fase de Análisis y Diseño (2 semanas): Esta fase comprende el estudio de los requisitos del proyecto, la definición del diseño de la web y el prototipado de la misma. Estas tareas se llevan a cabo en la empresa Homerti.
- Fase de Desarrollo (4 semanas): En esta etapa, se lleva a cabo el desarrollo de las diferentes secciones de la web, que incluyen la página de inicio, la sección de reservas, la página de servicios y la página de contacto. Estas tareas se realizan en la empresa Homerti, excepto el desarrollo de la integración con los servicios externos, que se completa en casa.
- Fase de Pruebas (2 semanas): En esta fase, se realizan pruebas exhaustivas, que incluyen pruebas de funcionalidad en la empresa Homerti, pruebas de rendimiento y pruebas de seguridad en casa.

El diagrama de Gantt muestra visualmente cómo estas fases se superponen y se relacionan entre sí, lo que ayuda a comprender cuándo se llevan a cabo las tareas y cuánto tiempo se dedica a cada una de ellas. El tiempo total estimado para completar todas las fases es de unas 8 semanas, o alrededor 2 meses y medio.

## ANÁLISIS DE LA COMPETENCIA

### *Situación actual: productos relacionados en el mercado y su coste.*

En el mercado de sitios web de hoteles y servicios de reserva, existen diversas soluciones y competidores que ofrecen servicios similares. Estos productos varían en términos de funcionalidad, alcance y precio. A continuación, se analizarán algunos de los productos y competidores relevantes en el mercado:

1. **Competidores directos de otros hoteles:** Muchos hoteles tienen sus propios sitios web que ofrecen información sobre habitaciones, instalaciones y opciones de reserva. La mayoría de estos sitios están diseñados de manera estática y pueden carecer de la flexibilidad y la integración de servicios externos que ofrece el proyecto de "Aura de Mallorca". El coste de mantener estos sitios puede variar, pero generalmente implica costos iniciales significativos y costos continuos de desarrollo y mantenimiento.
2. **Plataformas de terceros de reservas:** Existen plataformas de terceros como Booking.com, Expedia, y Airbnb que permiten a los hoteles listar sus propiedades y gestionar reservas. Estas plataformas suelen cobrar comisiones significativas a los hoteles por cada reserva, lo que puede resultar en costes considerables a largo plazo. Además, los hoteles pueden tener una presencia limitada en estas plataformas, ya que compiten con numerosos otros establecimientos.
3. **Sitios web de agencias de viajes y comparadores de precios:** Hay sitios web que actúan como agencias de viajes en línea y comparadores de precios. Estos sitios permiten a los usuarios comparar múltiples opciones de alojamiento y servicios. A menudo, ganan dinero a través de comisiones por reserva. Los costes para los hoteles pueden variar según las comisiones acordadas con estas plataformas.

*Ventajas e inconvenientes de los productos del mercado en relación al producto previsto en el proyecto.*

Ventajas en comparación con los productos y servicios existentes en el mercado:

1. **Personalización y Experiencia Única:** A diferencia de los sitios web estáticos de otros hoteles, el proyecto "Aura de Mallorca" ofrece una experiencia altamente personalizada para los visitantes. Los clientes pueden elegir entre planes básicos y VIP, lo que añade un elemento de exclusividad a la reserva.
2. **Integración de Servicios Externos:** La capacidad de ofrecer servicios externos vinculados al hotel, como su propio jardín registrado, es una característica distintiva. Esto agrega un valor adicional y una experiencia única para los clientes que buscan más que solo alojamiento.
3. **Sistema Multi-idioma:** La disponibilidad de múltiples idiomas hace que el sitio sea más accesible para una audiencia internacional.
4. **Gestión de Usuarios Simplificada:** La posibilidad de gestionar cuentas de usuario a través de un modal es conveniente para los clientes y facilita la interacción con el sitio.
5. **Integración del Clima en la Reserva de Servicios:** La función de considerar el clima al reservar es una característica innovadora que tiene el potencial de influir positivamente en la toma de decisiones de los clientes.

### Inconvenientes del Producto del Proyecto en Comparación con la Competencia

Aunque el proyecto "Aura de Mallorca" presenta muchas ventajas, también hay algunos inconvenientes a considerar:

1. **Coste Inicial de Desarrollo:** El desarrollo de un sitio web tan completo y personalizado puede implicar costes iniciales significativos en términos de desarrollo y diseño.
2. **Gestión y Mantenimiento Continuo:** El mantenimiento continuo del sitio web y la gestión de la API de pronóstico del clima requerirán recursos y costes adicionales a lo largo del tiempo.
3. **Competencia con Plataformas de Terceros:** Aunque el proyecto ofrece una experiencia única, todavía compite con plataformas de terceros que pueden tener una audiencia más amplia.

En resumen, el proyecto "Aura de Mallorca" se destaca por su enfoque en la personalización, la integración de servicios externos y la consideración del clima en las reservas. Sin embargo, debe abordar los costes iniciales y de mantenimiento continua, así como la competencia con otros actores del mercado. El éxito dependerá de una estrategia efectiva de marketing y de mantener la satisfacción del cliente a lo largo del tiempo.

## RECURSOS DE SOFTWARE PARA EL DESARROLLO

### Lenguajes

1. **TypeScript**: Se utiliza en el frontend, proporcionando ventajas de tipado estático y facilidad de mantenimiento en el código React.
2. **JavaScript (Node.js)**: El backend se desarrolla en Node.js, que es un entorno de tiempo de ejecución de JavaScript.

### Frameworks

1. **React**: En el frontend, React es el marco de trabajo principal utilizado para construir la interfaz de usuario del sitio web, usando la herramienta de compilación Vite.
2. **Express.js**: En el backend, Express.js es un marco de aplicación web que facilita el enrutamiento, la gestión de solicitudes y otras operaciones relacionadas con el servidor, usando Vite de igual forma.

### Librerías

1. **React-Bootstrap**: Se utiliza para mejorar la interfaz de usuario de React, proporcionando componentes de diseño responsivo y estilos predefinidos.
2. **i18next**: Esta librería se utiliza para implementar la funcionalidad de múltiples idiomas en el proyecto, permitiendo que el sitio se presente en diferentes lenguajes.
3. **Stripe**: Stripe se utiliza tanto en el frontend como en el backend para la integración de pagos, lo que permite a los clientes realizar transacciones seguras en el sitio web.
4. **Bcrypt**: Se utiliza para el hash y la gestión segura de contraseñas en el backend, garantizando la seguridad de las cuentas de usuario.
5. **CORS (Cross-Origin Resource Sharing)**: Se utiliza para gestionar las políticas de seguridad de acceso entre diferentes dominios y habilitar las solicitudes desde el frontend al backend.
6. **Jsonwebtoken**: Esta librería se utiliza para autenticar y autorizar a los usuarios, generando tokens de acceso para la autenticación.
7. **OpenWeatherMap API**: Se integra en el backend para obtener información en tiempo real sobre el clima y proporcionar funcionalidades basadas en el pronóstico del tiempo.
8. **Nodemailer**: Librería para poder enviar correos desde el backend.
9. **Multer**: Librería para procesar las imágenes y almacenarlas en el backend.
10. **Qrcode.react**: Librería en el frontend para generar y procesar los códigos QR.

### *Herramientas para el despliegue*

1. **Vite**: Utilizado para el desarrollo del frontend, con un servidor local para pruebas y desarrollo rápido. Además, se utiliza para compilar y empaquetar el código para la producción.
2. **Debian (Sistema Operativo)**: Debian se utiliza como sistema operativo en la máquina virtual (VM) para el despliegue del proyecto. Proporciona un entorno estable y seguro para alojar la aplicación. Usado en la versión 12.
3. **VirtualBox**: Se emplea para la gestión de la máquina virtual que aloja el servidor Debian.
4. **Apache**: Se utiliza como servidor web en el servidor Debian para servir el frontend y manejar las solicitudes HTTP entrantes.

### *Bases de datos*

**MariaDB**: Es el sistema de gestión de bases de datos relacional (RDBMS) utilizado para almacenar y gestionar los datos del sitio web y la información de los usuarios (usado MySQL con XAMPP para development).



## FASE DE ANÁLISIS

### *Requerimientos de la aplicación*

#### **Requerimientos Funcionales:**

##### **1. Gestión de Usuarios:**

Los usuarios deben poder registrarse en el sitio web.

Los usuarios registrados deben poder iniciar sesión de manera segura.

Los usuarios deben poder editar sus perfiles, incluyendo la carga de una foto de perfil.

Los usuarios deben poder cerrar sesión y eliminar sus cuentas.

##### **2. Reservas:**

Los clientes deben poder buscar habitaciones disponibles en el hotel por fechas.

Los clientes deben poder realizar reservas de habitaciones y servicios en el hotel.

Se deben ofrecer diferentes opciones de planes, como el plan básico y VIP, con servicios adicionales.

##### **3. Servicios y Experiencias:**

Los clientes deben poder explorar y obtener información detallada sobre los servicios internos y externos del hotel.

Los clientes deben poder reservar servicios, eligiendo de manera independiente y opcional los que se quieran al hacer la reserva junto al hospedaje, y el sistema debe considerar el pronóstico del tiempo al permitir o no la reserva de ciertos servicios.

##### **4. Interfaz Multilingüe:**

El sitio web debe ofrecer contenido en múltiples idiomas para atraer a una audiencia internacional.

Los usuarios deben poder seleccionar su idioma preferido.

##### **5. Procesamiento de Pagos:**

El sitio debe permitir el procesamiento de pagos seguros a través de Stripe, y se debe considerar la posibilidad de integrar PayPal en el futuro.

Los usuarios deben recibir confirmación de reserva exitosa o fallida (si es el caso).

##### **6. Comunicación con el Hotel:**

Debe haber una página de contacto que permita a los clientes ponerse en contacto con el hotel para consultas y solicitudes especiales.

## Requerimientos No Funcionales:

### 1. Seguridad:

La aplicación debe garantizar la seguridad de los datos del usuario, incluyendo contraseñas y datos de pago.

Debe protegerse contra ataques comunes, como inyección de SQL y ataques de fuerza bruta.

### 2. Rendimiento:

El sitio web debe ser rápido y responsivo, con tiempos de carga mínimos.

Debe ser escalable para manejar un crecimiento de tráfico.

### 3. Disponibilidad:

La aplicación debe estar disponible las 24 horas del día, los 7 días de la semana, para los usuarios que deseen hacer reservas en cualquier momento.

### 4. Usabilidad:

La interfaz de usuario debe ser intuitiva y fácil de usar, independientemente del idioma seleccionado.

Debe ser accesible desde una variedad de dispositivos, incluyendo computadoras de escritorio y dispositivos móviles.

### 5. Mantenimiento y Escalabilidad:

La aplicación debe ser fácil de mantener y actualizar, con un sistema de versiones claro. Debe ser escalable para agregar nuevas funcionalidades y adaptarse a las necesidades cambiantes del hotel.

## Modelo E/R de la base de datos

Este es el modelo Entidad – Relación de la base de datos del Hotel Aura de Mallorca.

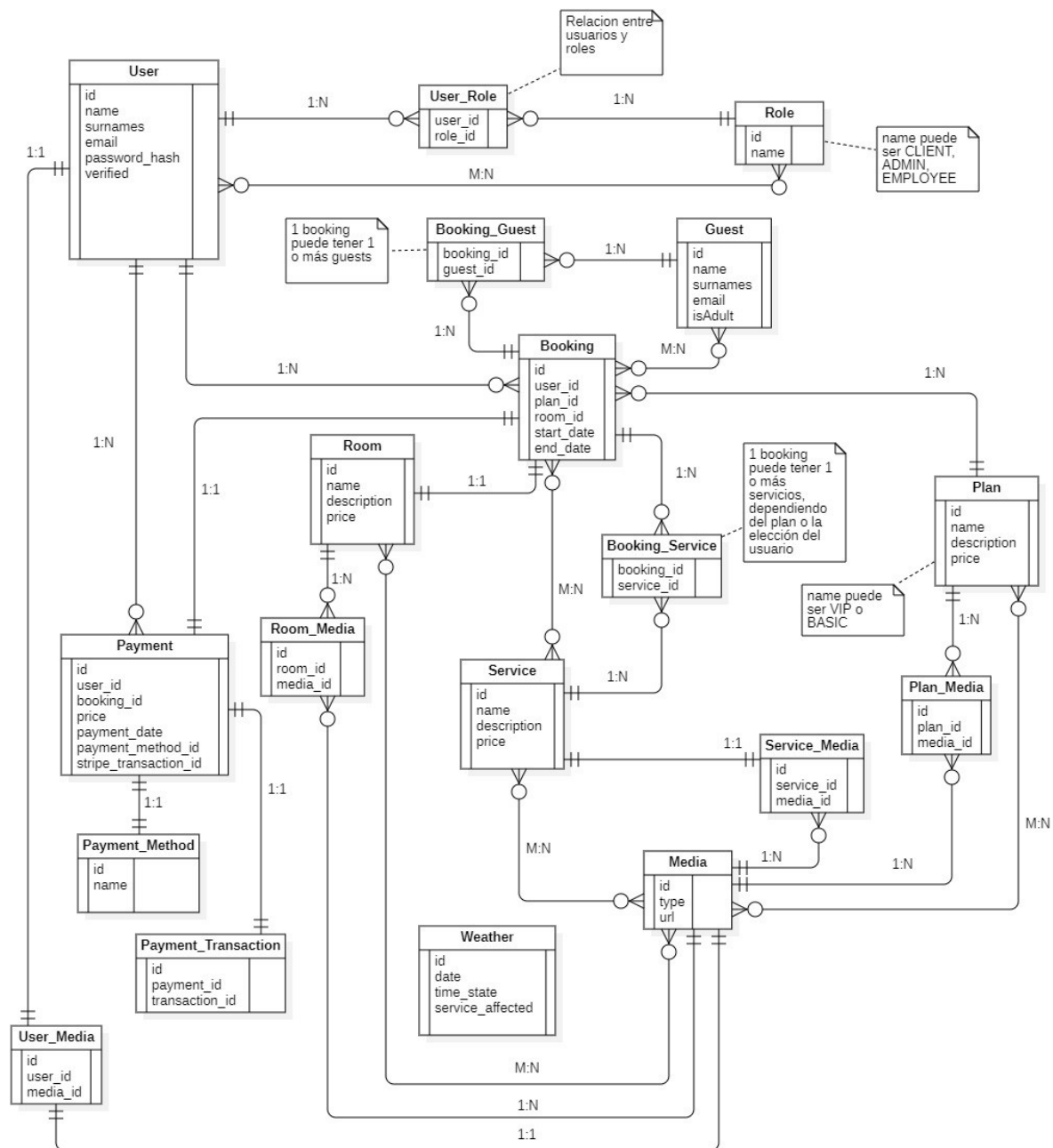


Figura 3: Modelo / Diagrama Entidad - Relación de la base de datos de la app

El modelo Entidad-Relación (E/R) para la base de datos del Hotel "Aura de Mallorca" presenta entidades principales y relaciones clave para representar la estructura de la información sin proporcionar ejemplos específicos de entidades.

## Diagrama de casos de uso

Este es el diagrama de casos de uso de la app.

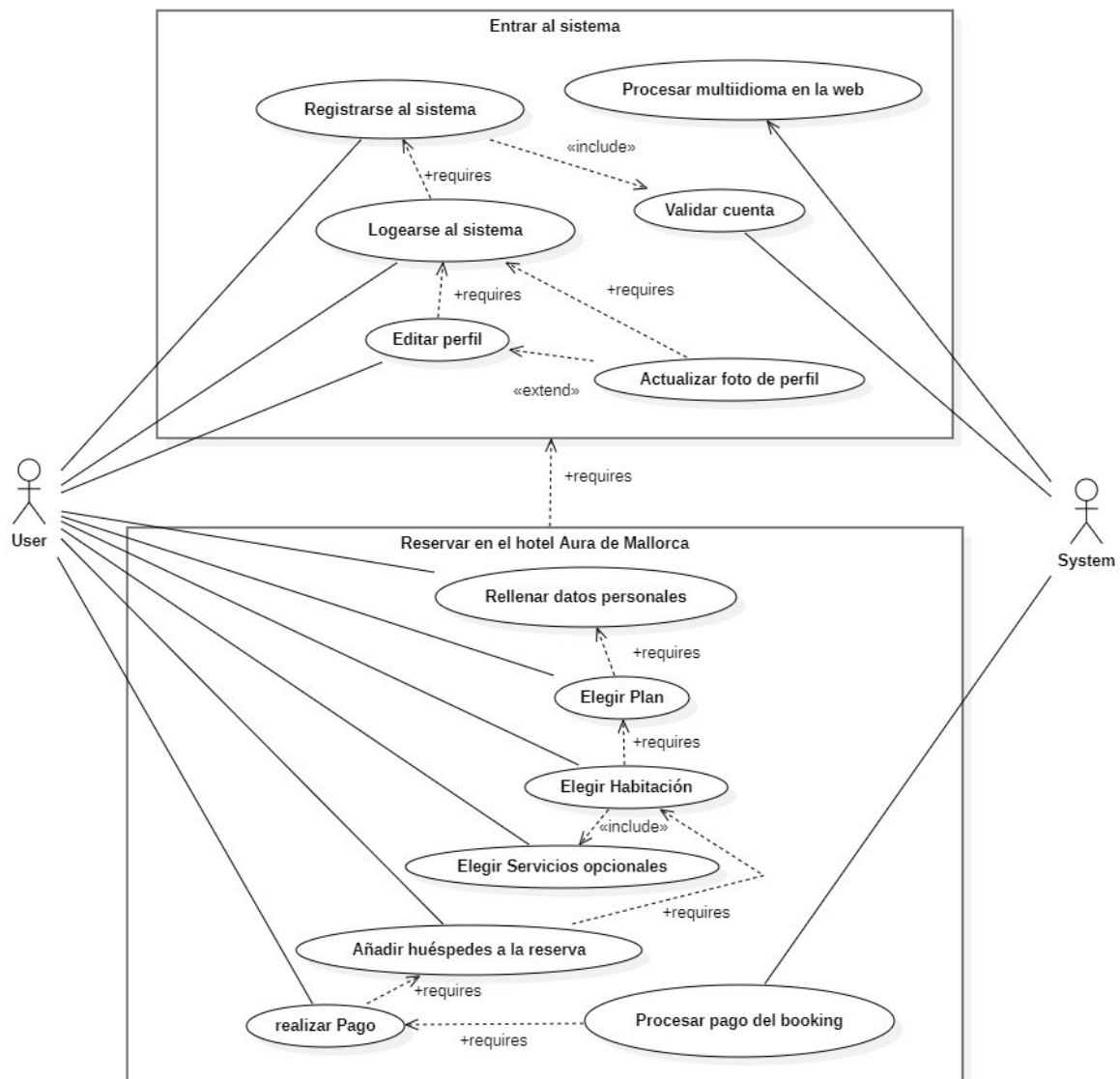


Figura 4: Diagrama de casos de uso de la app

El diagrama de casos de uso para el sistema del Hotel "Aura de Mallorca" representa las interacciones entre los actores y el sistema, abarcando funciones como "Gestión de Reservas" y "Registro de Usuarios".

El actor User (usuario) puede entrar al sistema o reservar en el hotel, tiene muchas funciones, mientras que el actor System (sistema) tiene funciones de verificar y procesar las acciones del usuario.

## Modelo Relacional de la Base de Datos

The diagram illustrates a database schema for a travel application, featuring 15 tables and their relationships. Each table is represented by a blue box containing its name, columns, data types, and constraints. Relationships are shown as dashed lines connecting attributes across different tables.

**Tables and their attributes:**

- app\_user**: id INT(11), user\_name VARCHAR(255), user\_surnames VARCHAR(255), user\_email VARCHAR(255), user\_password VARCHAR(255), user\_verified TINYINT(1), verification\_token VARCHAR(255), verification\_token\_expiry TIMESTAMPTZ, created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- user\_role**: id INT(11), user\_id INT(11), role\_id INT(11), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- role**: id INT(11), name ENUM(...), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- booking\_guest**: id INT(11), booking\_id INT(11), guest\_id INT(11), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- guest**: id INT(11), guest\_name VARCHAR(255), guest\_surnames VARCHAR(255), guest\_email VARCHAR(255), idkdu CHAR(1), idsystem\_user CHAR(1), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- booking**: id INT(11), user\_id INT(11), plan\_id INT(11), room\_id INT(11), booking\_start\_date DATE, booking\_end\_date DATE, created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- room**: id INT(11), room\_name VARCHAR(255), room\_description TEXT, room\_price DECIMAL(10,2), room\_availability\_start DATE, room\_availability\_end DATE, created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- payment**: id INT(11), user\_id INT(11), booking\_id INT(11), payment\_amount DECIMAL(10,2), payment\_date DATE, payment\_method\_id INT(11), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- booking\_service**: id INT(11), booking\_id INT(11), service\_id INT(11), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- plan**: id INT(11), plan\_name VARCHAR(255), plan\_description TEXT, plan\_price DECIMAL(10,2), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- room\_media**: id INT(11), room\_id INT(11), media\_id INT(11), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- service**: id INT(11), serv\_name VARCHAR(255), serv\_description TEXT, serv\_price DECIMAL(10,2), serv\_availability\_start DATE, serv\_availability\_end DATE, created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- service\_media**: id INT(11), service\_id INT(11), media\_id INT(11), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- plan\_media**: id INT(11), plan\_id INT(11), media\_id INT(11), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- payment\_method**: id INT(11), payment\_method\_name VARCHAR(255), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- payment\_transaction**: id INT(11), payment\_id INT(11), transaction\_id VARCHAR(255), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- user\_media**: id INT(11), user\_id INT(11), media\_id INT(11), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- weather**: id INT(11), weather\_date DATE, weather\_state VARCHAR(255), created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.
- media**: id INT(11), type ENUM(...), uri TEXT, created\_at TIMESTAMPTZ, updated\_at TIMESTAMPTZ.

**Relationships:**

- app\_user** to **user\_role**: One-to-many relationship on **user\_id**.
- user\_role** to **role**: One-to-many relationship on **role\_id**.
- booking\_guest** to **guest**: One-to-many relationship on **guest\_id**.
- booking** to **room**: One-to-many relationship on **room\_id**.
- booking** to **plan**: One-to-many relationship on **plan\_id**.
- booking** to **booking\_service**: One-to-many relationship on **booking\_id**.
- payment** to **room**: One-to-many relationship on **room\_id**.
- payment** to **payment\_method**: One-to-many relationship on **payment\_method\_id**.
- payment** to **payment\_transaction**: One-to-many relationship on **payment\_id**.
- room** to **room\_media**: One-to-many relationship on **room\_id**.
- service** to **service\_media**: One-to-many relationship on **service\_id**.
- plan** to **plan\_media**: One-to-many relationship on **plan\_id**.
- user** to **user\_media**: One-to-many relationship on **user\_id**.
- media** to **room\_media**: Many-to-many relationship.
- media** to **service\_media**: Many-to-many relationship.
- media** to **plan\_media**: Many-to-many relationship.
- weather** to **media**: One-to-many relationship on **weather\_date**.

*Figura 5: Modelo Relacional de la Base de Datos del Hotel Aura de Mallorca*

## *Diseño de la interfaz del usuario*

### Header (Encabezado) - Parte Superior Fija:

- Logo del Hotel (Izquierda):
  - Ubicación: En la esquina superior izquierda.
  - Justificación: Colocar el logo en la esquina izquierda es una convención de diseño que facilita a los usuarios identificar rápidamente la marca o el sitio web. Es la primera vista del usuario y refuerza la identidad del hotel.
- Menú de Navegación (Centro):
  - Ubicación: En el centro del header.
  - Elementos del Menú: "Home", "Servicios", "Contacto". (Adicionalmente, "Reservas" y "Admin" si es relevante).
  - Justificación: Colocar el menú en el centro proporciona un equilibrio visual. Los elementos del menú están organizados de manera lógica y fácilmente accesibles. Utilizar texto claro y conciso mejora la usabilidad.
- Icono de Usuario y Selector de Idiomas (Derecha):
  - Ubicación: En la esquina superior derecha.
  - Elementos: Icono de usuario y selector de idiomas.
  - Justificación:
    - Icono de Usuario: Fácil acceso para el usuario a funciones de gestión de cuenta.
    - Selector de Idiomas: Ubicado junto al usuario para reflejar la asociación del idioma con la cuenta del usuario. Dropdown para mejorar la accesibilidad y permitir elección rápida.

### Cuerpo de la Página:

- Imagen de Portada con Efecto Parallax (si es la pantalla "Home", sino su contenido) correspondiente:
  - Ubicación: Cubriendo toda la pantalla.
  - Justificación: Si es la home → Una imagen de portada a pantalla completa crea un impacto visual. El efecto parallax al hacer scroll agrega dinamismo y atractivo visual. Si es las otras páginas → Presentar directo el contenido.
- Botón de Reserva Fijo (Abajo a la Izquierda):
  - Ubicación: En la esquina inferior izquierda, con un margen.
  - Justificación: Colocar el botón de reserva en un lugar fijo y fácilmente accesible permite a los usuarios iniciar el proceso de reserva en cualquier punto de la página. El margen evita clics accidentales.

### Footer (Pie de Página):

- Columnas en Horizontal:
  - Izquierda: Nombre del Hotel y Descripción:
    - Justificación: Proporciona información básica del hotel y refuerza la identidad.
  - Derecha: Enlaces Legales (Políticas, Cookies<sup>6</sup>, Aviso Legal, etc.):
    - Justificación: Agrupar enlaces legales en una columna permite a los usuarios acceder fácilmente a información importante.

### Modales Pop-up (User Modal y Booking Modal):

- Ubicación: Centro de la pantalla con fondo difuminado.
- Justificación: Colocar los modales en el centro atrae la atención del usuario. El fondo difuminado destaca el modal y evita distracciones. La X en la esquina facilita cerrar el modal.

### Menú Colapsado en Dispositivos Móviles:

- Icono de Menú (Despliega el Nav):
  - Ubicación: En la esquina superior izquierda.
  - Justificación: Al colapsar el menú, el icono de menú proporciona una forma estándar de mostrar opciones de navegación en dispositivos móviles.

### Razonamiento de Usabilidad y Accesibilidad:

- Visibilidad y Consistencia:
  - Colocar elementos clave como el logo y el menú en posiciones estándar mejora la usabilidad y la consistencia a través de las páginas.
- Accesibilidad Móvil:
  - El menú colapsado facilita la navegación en dispositivos móviles. El botón de reserva fijo y el acceso rápido a funciones de usuario mejoran la accesibilidad en pantallas pequeñas.
- Fácil Acceso a Funciones Cruciales:
  - El icono de usuario y el selector de idiomas están ubicados estratégicamente para un acceso fácil, enfocándose en funciones cruciales y personalización.
- Legibilidad y Claridad:
  - Texto claro y conciso en el menú y enlaces legales mejora la legibilidad y facilita la comprensión.

---

<sup>6</sup> Cookie: un pequeño archivo de texto almacenado en el navegador del usuario, que contiene información sobre la interacción con un sitio web y se utiliza para personalizar la experiencia del usuario.

### Modelo estático del sistema (Diagrama de Clases)

Este es el diagrama de clases de la lógica de dominio de la aplicación web.

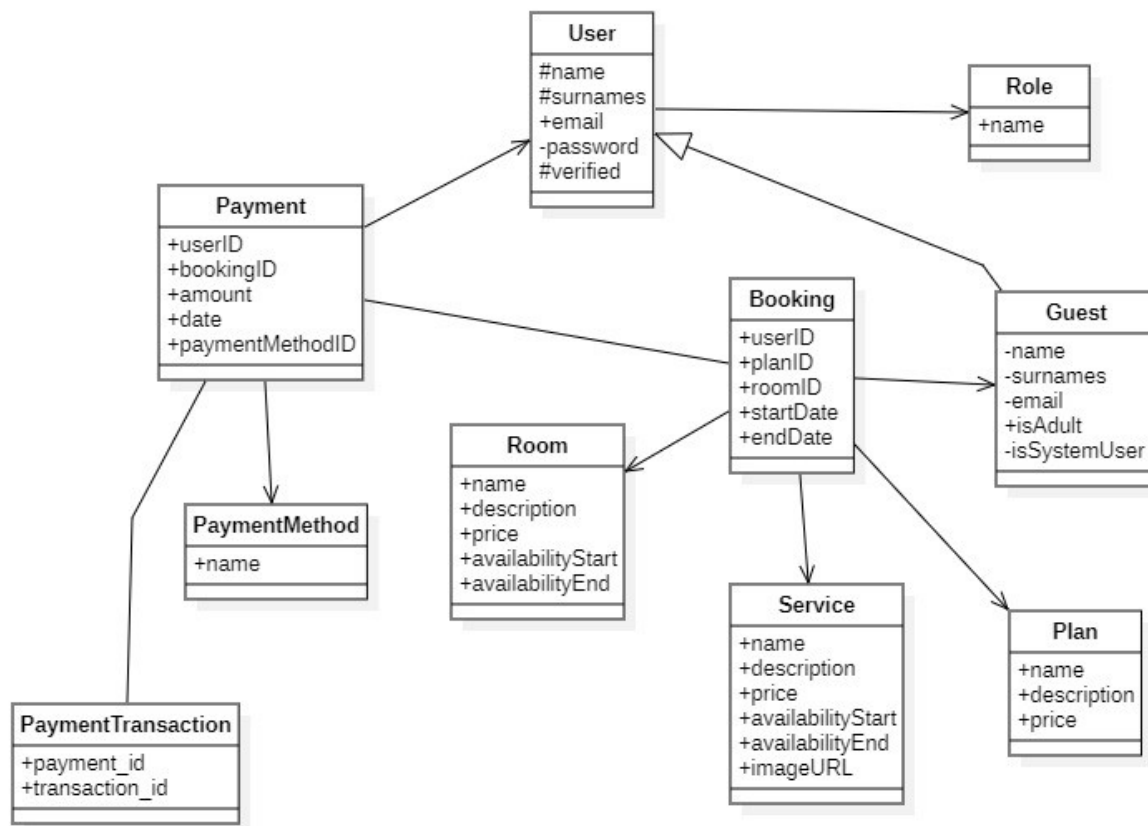


Figura 6: Diagrama de Clases (modelo estático del sistema) del Hotel Aura de Mallorca.

Un diagrama de clases representa la estructura y las relaciones de las clases en un sistema, con el objetivo de visualizar la organización conceptual de las entidades y su interacción dentro del dominio de la aplicación.



## *Diseño de la infraestructura física*

### Servidores y Máquinas Virtuales:

#### 1. Servidor Local:

- Máquina virtual Debian 12 en VirtualBox para desarrollo y presentación.

### Servicios y Software:

#### 1. Servidor Web:

Apache en el puerto 80 y 443 para HTTPS (sirviendo la parte frontend de la aplicación en modo de producción).

#### 2. Gestión de Procesos:

PM2 (con NPM de NodeJS) para gestionar y mantener activo el proceso Node.js del backend en el puerto 3000.

#### 3. Base de Datos:

MariaDB incluido en Debian 12 para la gestión de la base de datos.

#### 4. Servidor de Aplicaciones:

Node.js ejecutando la parte backend de la aplicación.

#### 5. SSL/TLS:

Certificado SSL/TLS para cifrar la comunicación (autofirmado) aplicado al servidor Apache.

### Seguridad:

#### 1. Firewall:

- Configuración de un firewall en Debian para restringir el tráfico (ufw).

### Escalabilidad y Redundancia:

#### 1. Almacenamiento Local:

- Uso del almacenamiento local de la máquina virtual Debian.

## Wireframe

Este es el “wireframe” del sitio web del Hotel.

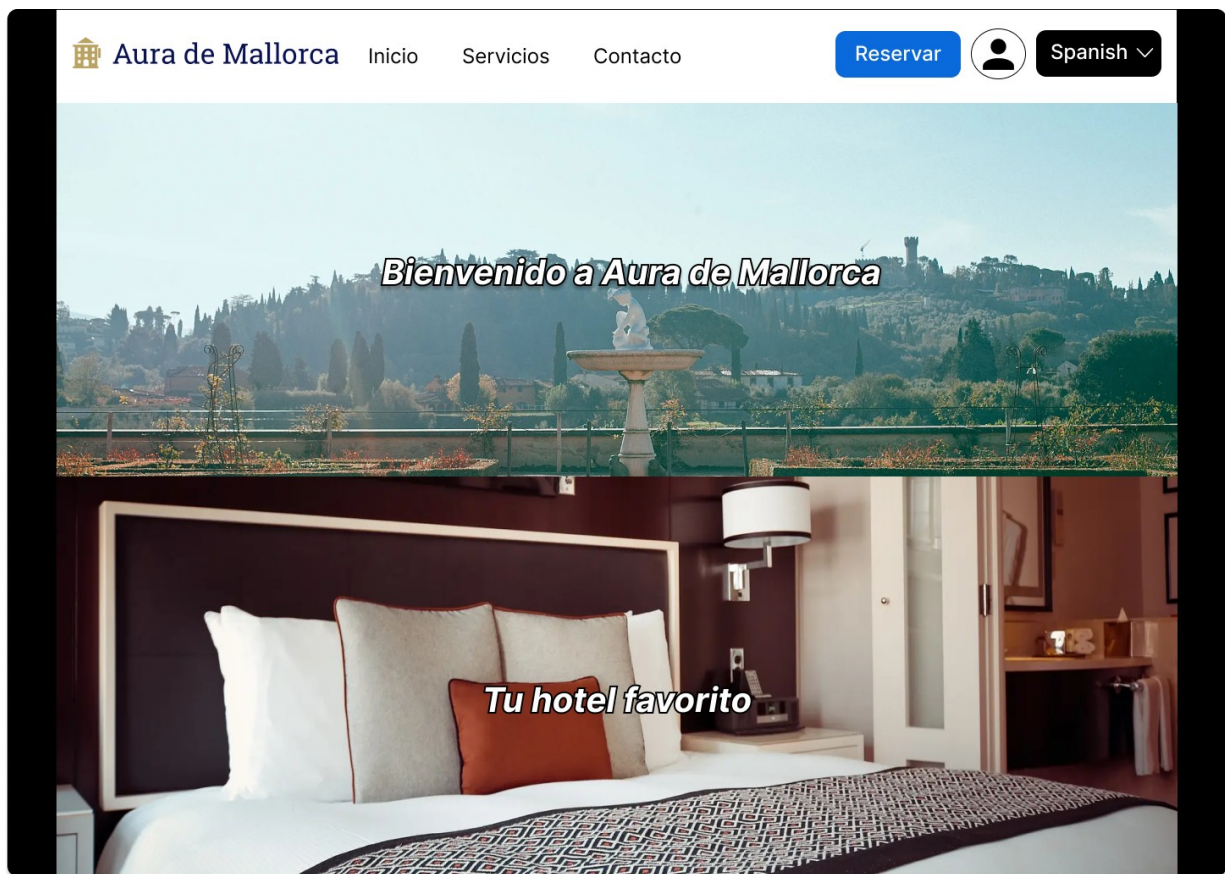
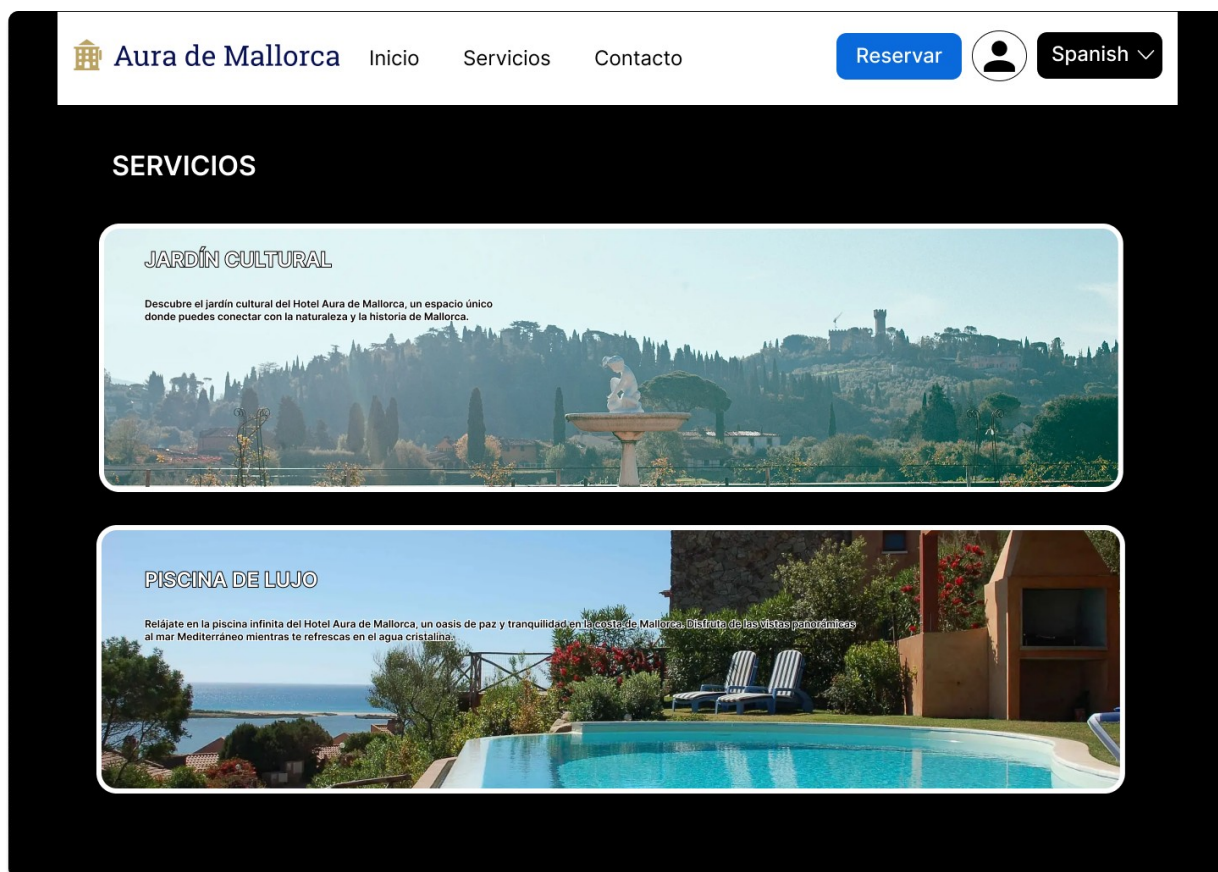


Figura 7: Wireframe: Página de Inicio (Home)

El wireframe es un esquema visual básico que representa la estructura y disposición de los elementos en una interfaz de usuario, utilizado en el diseño de sitios web y aplicaciones. Se centra en la disposición de los elementos, sin incluir detalles visuales o de estilo, para planificar la arquitectura y la navegación de la interfaz.



*Figura 8: Wireframe: Página de Servicios*

Los wireframes son herramientas esenciales en el diseño de interfaces, proporcionando una representación visual simplificada de la disposición de elementos. Sirven para planificar la estructura y funcionalidad de una interfaz, facilitando la comunicación entre diseñadores, desarrolladores y clientes, y permiten identificar y abordar problemas de usabilidad antes de la implementación.

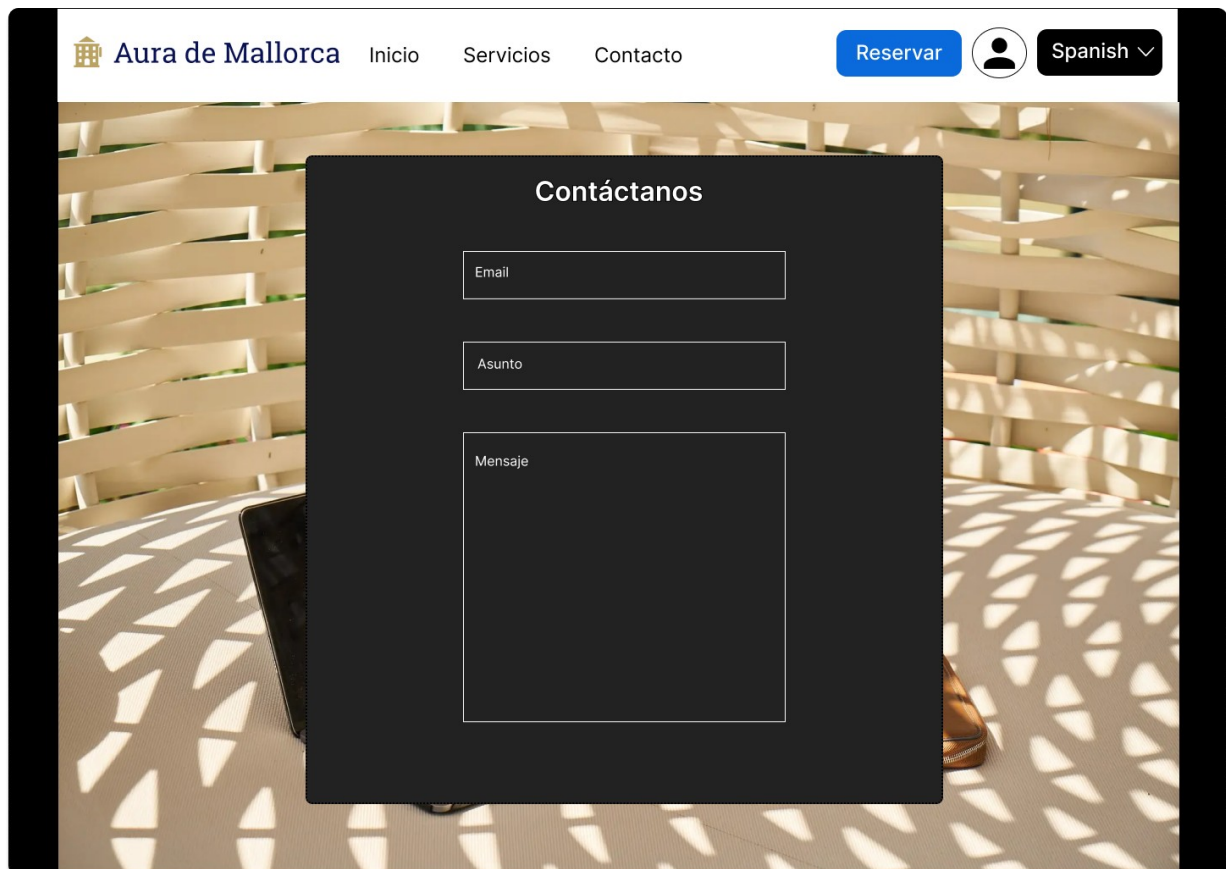


Figura 9: Wireframe: Página de Contacto

También son útiles para conceptualizar y planificar el diseño de una interfaz, estableciendo la disposición de elementos clave. Facilitan la colaboración entre equipos al visualizar la estructura básica antes de la implementación, mejorando la eficiencia y reduciendo posibles errores en la fase de desarrollo.



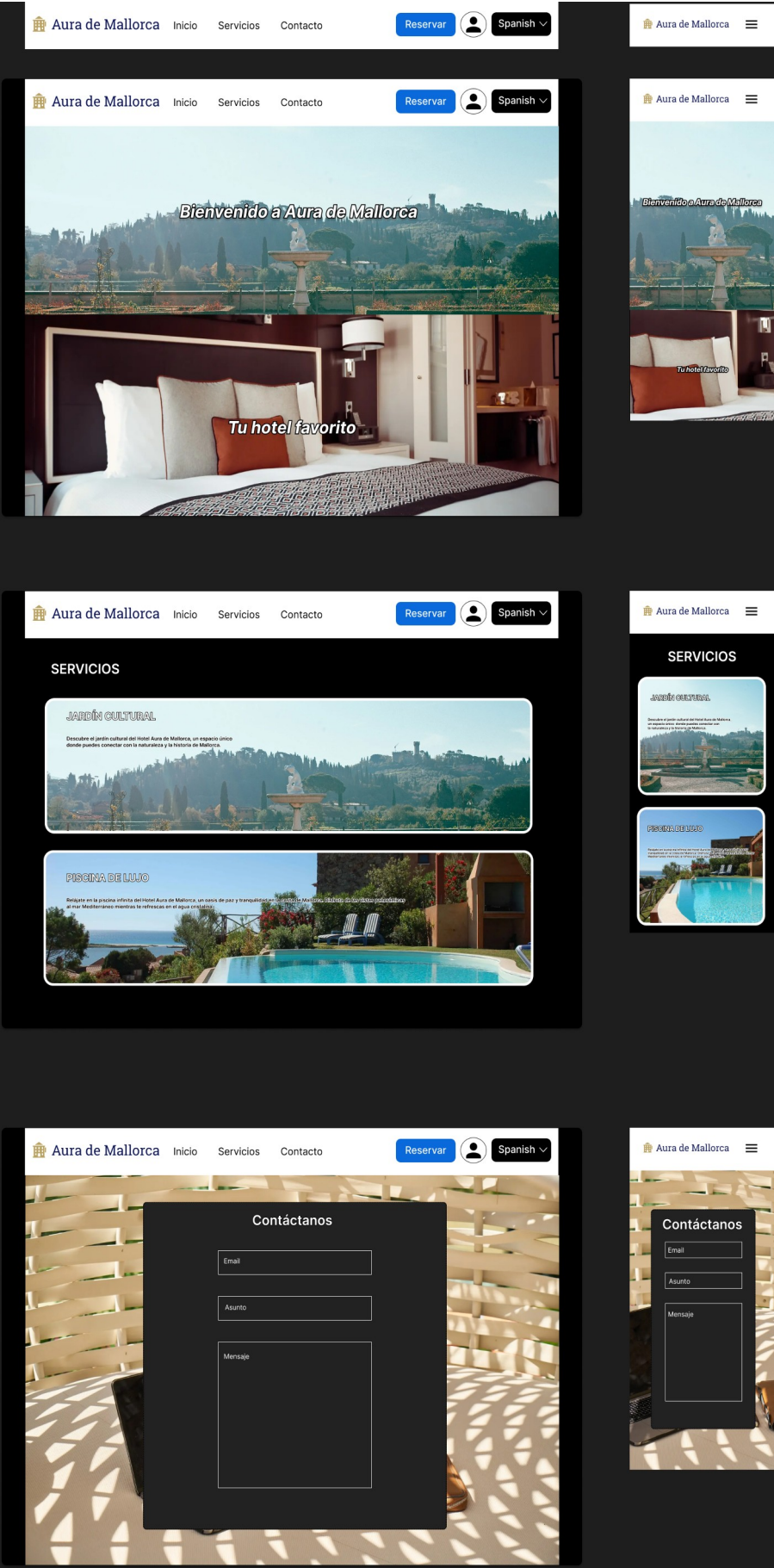


Figura 10: Wireframe: Página Web completa

## FASE DE PRUEBAS

### 1. Página de Home:

- Objetivo: Verificar la presentación correcta del hotel Aura de Mallorca y sus servicios en la página de inicio.
- Pasos:
  1. Cargar la página de inicio.
  2. Verificar la correcta visualización de la información del hotel y servicios.
- Criterios de Éxito: La página de inicio debe cargar sin errores, mostrando correctamente la información del hotel.

### 2. Modal de Booking:

- Objetivo: Evaluar la funcionalidad del proceso de reserva de habitaciones e instalaciones, así como la integración con la API meteorológica.
- Pasos:
  1. Abrir el modal de reservas.
  2. Elegir el plan (basic o VIP).
  3. Seleccionar fechas, habitaciones, servicios (opcional) y huéspedes.
  4. Verificar la consulta a la API meteorológica al darle a “Pagar”.
  5. Completar la reserva.
- Criterios de Éxito: El proceso de reserva debe completarse sin errores, considerando la información meteorológica para permitir o denegar la reserva (y el pago si la reserva tuvo algún error al realizarse).

### 3. Modal de User Management:

- Objetivo: Validar las funciones relacionadas con la gestión de usuarios.
- Pasos:
  1. Abrir el modal de administración de usuario.
  2. Registrarse como nuevo usuario.
  3. Iniciar sesión con el nuevo usuario.
  4. Editar la cuenta y subir una foto de perfil.
  5. Cerrar sesión y eliminar la cuenta.
- Criterios de Éxito: Todas las funciones de gestión de usuarios deben ejecutarse sin errores, con cambios reflejados correctamente en la base de datos.

#### 4. Página de Servicios:

- Objetivo: Evaluar la presentación de información sobre servicios internos y externos.
- Pasos:
  1. Cargar la página de servicios.
  2. Verificar la información sobre servicios.
- Criterios de Éxito: La información de servicios debe presentarse correctamente.

#### 5. Página de Contacto:

- Objetivo: Comprobar la capacidad de los clientes para ponerse en contacto con el hotel.
- Pasos:
  1. Cargar la página de contacto.
  2. Utilizar el formulario de contacto para enviar un mensaje.
- Criterios de Éxito: El formulario de contacto debe funcionar correctamente, enviando mensajes al hotel.

En conclusión: las pruebas han demostrado que todas de las funcionalidades se desempeñan según lo previsto. Los problemas encontrados se han abordado y corregido. La integración con la API<sup>7</sup> meteorológica ha demostrado ser efectiva para garantizar la toma de decisiones condicionales en el proceso de reserva.

---

<sup>7</sup> API: Application Programming Interface es un conjunto de reglas y herramientas que permite la interacción entre diferentes software, facilitando la comunicación y el intercambio de datos entre aplicaciones.

Hablando en **usabilidad**, se llevaron a cabo sesiones de simulación con usuarios para evaluar la usabilidad de la web. Se recopilaban comentarios y observaciones, y se realizaron ajustes para mejorar la experiencia del usuario. Las principales conclusiones incluyen:

La navegación es intuitiva, y los usuarios encuentran fácilmente la información que buscan.

El proceso de reserva es claro y sencillo, con la información meteorológica proporcionando una capa adicional de utilidad.

La gestión de usuarios es eficiente, y las funciones de edición y eliminación de cuentas son comprensibles.

Estas observaciones se utilizarán para implementar mejoras adicionales y optimizar aún más la usabilidad antes del lanzamiento oficial de la web.



## ANÁLISIS DE RECURSOS Y COSTES

### Recursos Utilizados

#### 1. Hardware:

- Portátil:
  - Procesador: Intel Core i7
  - RAM: 8 GB
  - Almacenamiento: 128 GB SSD
  - Tarjeta Gráfica: Intel HD Graphics

#### 2. Software:

- Sistema Operativo:
  - Windows 11
- Entorno de Desarrollo:
  - Visual Studio Code (VSCode) con extensiones de React, HTML, CSS, JS, y NodeJS instalado en el ordenador.
- Servidor Local:
  - XAMPP con PHPMyAdmin
- Navegadores:
  - Chrome, Firefox, Opera

### Costes

Los costes asociados a este proyecto son mínimos ya que se basan en tecnologías de software libre y herramientas gratuitas. Se utilizan recursos disponibles en el portátil del desarrollador, y no hay gastos directos asociados con la infraestructura de desarrollo.

#### 1. Hardware:

- El portátil utilizado para el desarrollo ya está en posesión del desarrollador y no genera costos adicionales para el proyecto.

#### 2. Software:

- Todas las herramientas de desarrollo, como Visual Studio Code, XAMPP, y los navegadores Chrome, Firefox, y Opera, son de código abierto o gratuitas para uso personal.
- El sistema operativo Windows 11 puede ser parte de las licencias adquiridas previamente y no se considera un costo directo para el proyecto.

### 3. Infraestructura de Despliegue:

Se utiliza una máquina virtual Debian en VirtualBox para el despliegue con Apache, MariaDB para Linux instalados y con certificado SSL autofirmado aplicado al servidor Apache con los virtualhosts configurados para la web y htaccess. VirtualBox es una herramienta de virtualización de código abierto y no implica costos adicionales.

#### Resumen

En general, el proyecto se desarrolla utilizando recursos disponibles y herramientas gratuitas. Los costos asociados son mínimos, limitándose al hardware existente y al uso de software libre. Esto hace que el proyecto sea económicamente eficiente y accesible para desarrolladores individuales o pequeños equipos. La elección de estas tecnologías también facilita la escalabilidad<sup>8</sup> y futuras modificaciones sin incurrir en costos significativos.

---

<sup>8</sup> Escalabilidad: capacidad para manejar un crecimiento sostenible en usuarios y carga de trabajo sin comprometer el rendimiento.

## RELACIÓN FINAL DEL CUMPLIMIENTO DE LAS DIFERENTES TAREAS DE CADA FASE DEL PROYECTO

Basándonos en el diagrama inicial de Gantt realizado a principio del proyecto con la planificación que inicial que se hizo para cada fase de desarrollo, ahora compraremos lo que se ha terminado cumpliendo con ese diagrama (se ha ido registrando todas las tareas estos 3 meses en Trello en un tablero Kanban con “To-Do”, “In Progress” y “Done” como columnas).

Primero recordemos la planificación inicial:

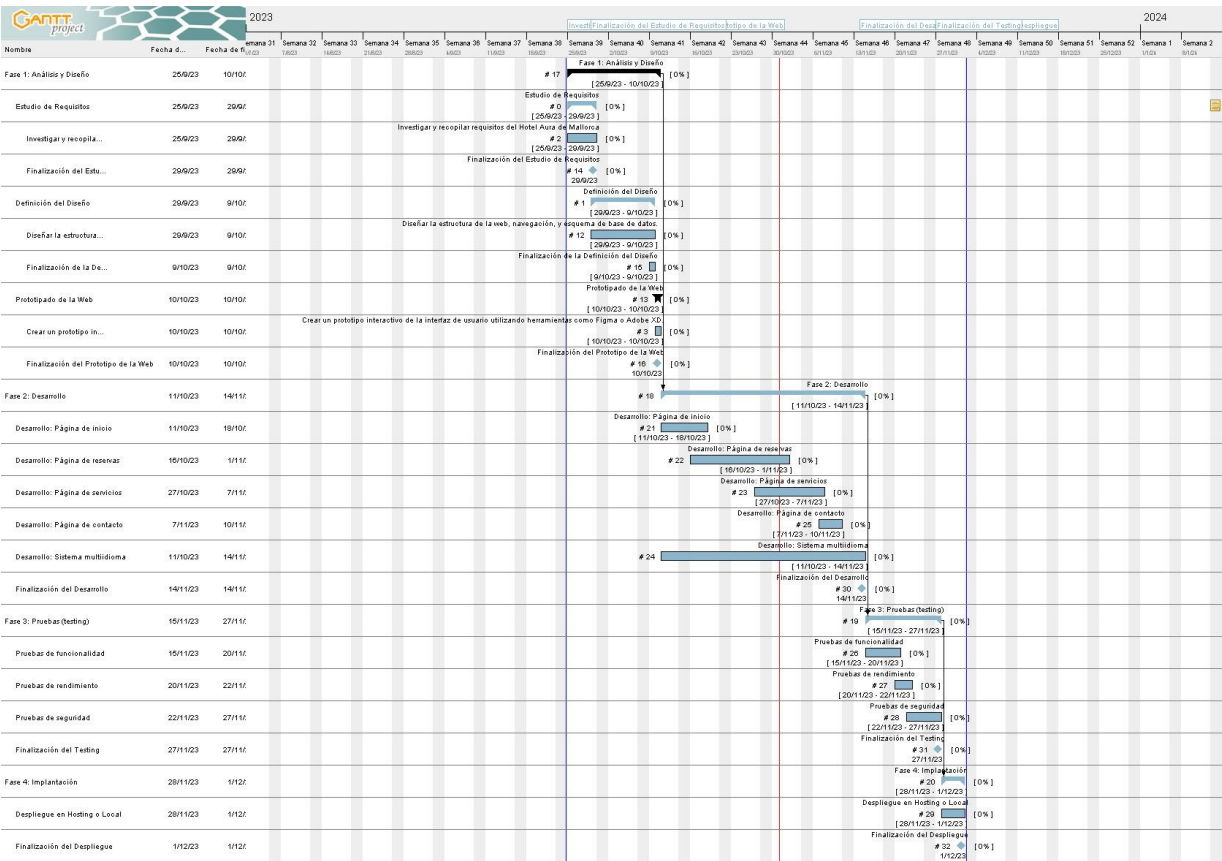


Figura 11: Diagrama de Gantt: Planficación inicial (duplicado del punto ya visto anteriormente)

Y esto ha sido el tablero Kanban en Trello al finalizar el proyecto:

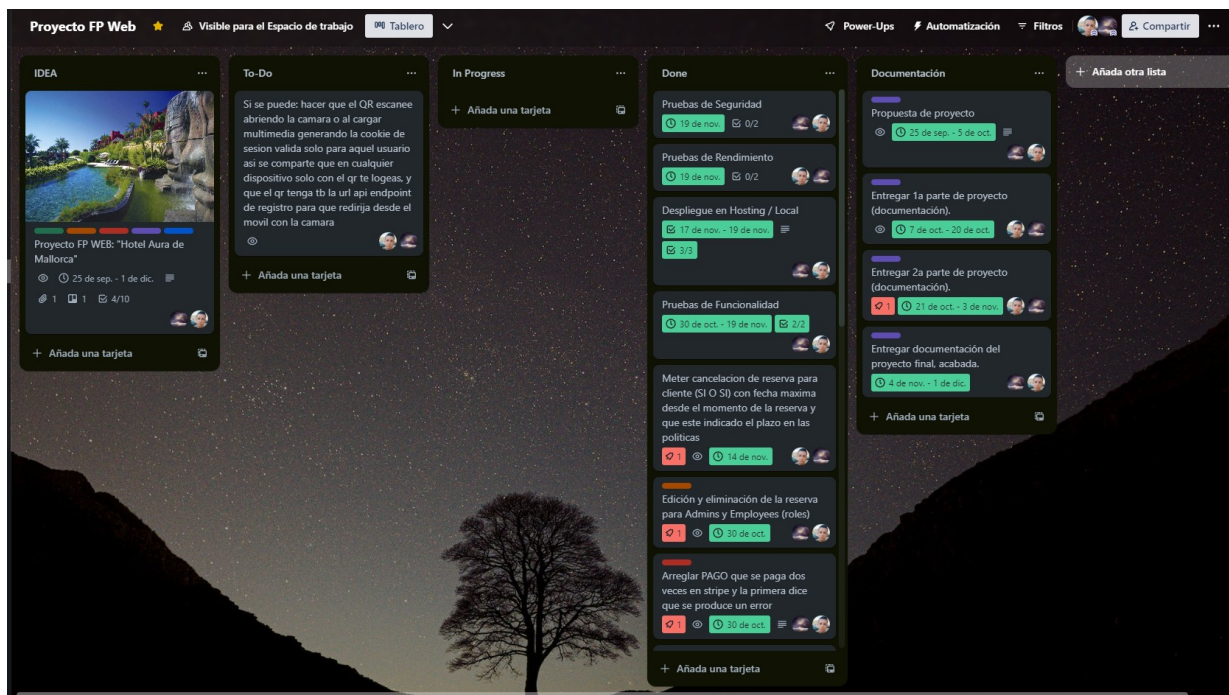


Figura 12: Tablero Kanban (con Trello) del resultado final de tareas al terminar el proyecto

También cabe recalcar que se hizo una fase de SPRINT SCRUM durante una semana para agilizar el desarrollo:

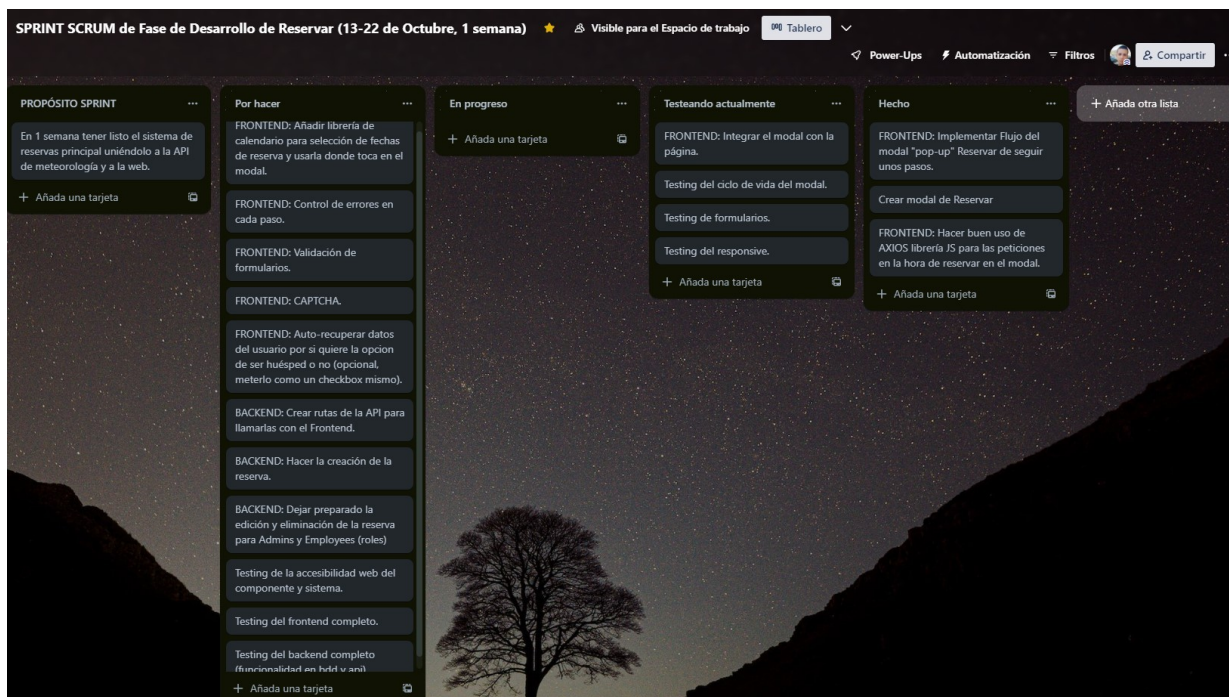


Figura 13: Sprint SCRUM (al iniciarlo)



Y aquí podemos ver cómo se cumplió exitosamente todo lo propuesto en el SPRINT en el plazo asignado:

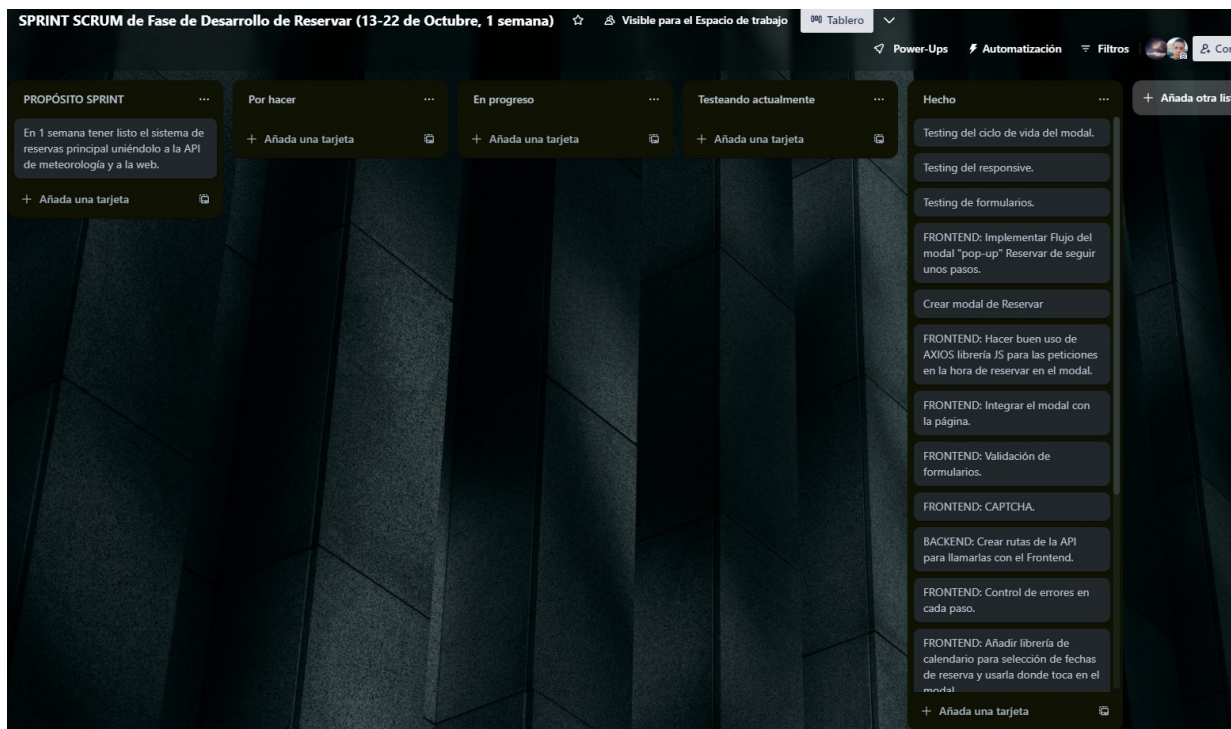


Figura 14: Sprint SCRUM (al finalizarlo)

Recordado esto, procedemos a hacer la relación final de cumplimiento de tareas:

#### Fase 1 - Análisis y Diseño (25/9/2023 al 10/10/2023)

- Actividades Planificadas:
  - Definición de Requisitos.
  - Diseño de la Arquitectura.
  - Planificación de Recursos y Tiempos.
- Cumplimiento:
  - Todas las actividades se completaron según lo planeado.
  - Se generó un diagrama de Gantt para la planificación temporal.

### Fase 2 - Desarrollo (11/10/2023 al 14/11/2023)

- Actividades Planificadas:
  - Implementación de la Página de Home.
  - Desarrollo del Modal de Booking.
  - Creación del Modal de User Management.
  - Desarrollo de la Página de Servicios.
  - Implementación de la Página de Contacto.
- Cumplimiento:
  - Se terminó la implementación de todas las funcionalidades.
  - El desarrollo se extendió en una semana y media más de lo previsto en el Gantt.
  - Trello Kanban utilizado para el seguimiento y gestión de las tareas.

### Fase 3 - Pruebas (15/11/2023 al 27/11/2023)

- Actividades Planificadas:
  - Plan de Pruebas de los Módulos Principales.
  - Simulación de Pruebas con Usuarios.
  - Conclusiones y Ajustes.
- Cumplimiento:
  - Se realizaron pruebas exhaustivas, siguiendo el plan establecido.
  - Las simulaciones de pruebas con usuarios proporcionaron valiosos comentarios.
  - Ajustes realizados para mejorar la usabilidad y corregir problemas identificados.

#### Fase 4 - Implantación (28/11/2023 al 1/12/2023)

- Actividades Planificadas:
  - Despliegue en la Máquina Virtual Debian.
  - Configuración de Apache y PM2.
  - Configuración de SSL y Dominio.
- Cumplimiento:
  - La implantación se realizó según lo programado.
  - Se configuraron correctamente Apache y PM2.
  - El sitio web se aseguró mediante SSL y se asoció con el dominio aurademallorca.com.

#### Resumen General

- La Fase 1 y Fase 3 se completaron según el cronograma previsto.
- La Fase 2 (Desarrollo) se extendió, terminando una semana y media después de lo planeado en el Gantt, pero se mantuvo bajo control utilizando un enfoque ágil y el tablero Kanban.
- La Fase 4 se llevó a cabo según lo planificado, finalizando el proyecto en la fecha programada.

A pesar de la pequeña demora en la fase de desarrollo, el proyecto se completó dentro del plazo general de tres meses. Las adaptaciones ágiles y la gestión eficiente de tareas contribuyeron al éxito general del proyecto.

## CONCLUSIONES Y OPINIÓN PERSONAL

### CONCLUSIONES

Este proyecto de desarrollo web para el Hotel "Aura de Mallorca" ha sido una experiencia enriquecedora y desafiante. Algunas conclusiones clave pueden resumirse de la siguiente manera:

#### 1. Planificación y Gestión del Tiempo:

- La planificación inicial del proyecto mediante el diagrama de Gantt proporcionó una estructura valiosa, pero la realidad demostró que las estimaciones de tiempo para la fase de desarrollo fueron optimistas. Aprender de esta experiencia es crucial para mejorar futuras planificaciones.

#### 2. Desarrollo Ágil y Adaptativo:

- La flexibilidad y adaptabilidad demostradas durante la fase de desarrollo fueron esenciales. La transición a un enfoque ágil, utilizando Trello Kanban, permitió una gestión efectiva de tareas y la corrección de desviaciones en el tiempo.

#### 3. Pruebas Rigurosas y Simulación de Usuarios:

- La fase de pruebas fue crucial para garantizar la robustez y funcionalidad del sistema. Las simulaciones con usuarios aportaron perspectivas valiosas sobre la usabilidad, permitiendo ajustes que mejoraron significativamente la experiencia del usuario.

#### 4. Infraestructura y Despliegue:

- La elección de utilizar una máquina virtual Debian y la configuración de Apache y PM2 para el despliegue fueron decisiones acertadas. Garantizar la seguridad mediante SSL y asociar el dominio contribuyó a un entorno de producción robusto.



## OPINIÓ PERSONAL

Este proyecto ha sido un desafío emocionante y educativo. Algunas reflexiones personales incluyen:

### 1. Aprendizaje Continuo:

- El proyecto proporcionó una oportunidad para aprender nuevas tecnologías y mejorar habilidades existentes. La utilización de tecnologías como React, XAMPP, y la implementación de funcionalidades complejas como la integración con la API meteorológica, ha sido valiosa para mi crecimiento como desarrollador web.

### 2. Gestión de Errores y Resolución de Problemas:

- Enfrentar desafíos inesperados durante el desarrollo y las pruebas me permitió mejorar mis habilidades para diagnosticar problemas y encontrar soluciones efectivas. La resiliencia es clave en el desarrollo de software.

### 3. Colaboración y Comunicación:

- Trabajar en este proyecto destacó la importancia de una comunicación efectiva y colaboración, incluso en un proyecto individual. El uso de herramientas como Trello facilitó la organización y el seguimiento de tareas.

### 4. Satisfacción del Cliente (en este caso, el Usuario):

- La satisfacción del usuario final es la medida definitiva de éxito. La simulación de pruebas con usuarios y la implementación de ajustes basados en sus comentarios reales son aspectos críticos para garantizar que la web cumpla con las expectativas y necesidades.

En resumen, este proyecto ha sido una experiencia valiosa que ha fortalecido mis habilidades técnicas y mi comprensión de la gestión de proyectos web. Las lecciones aprendidas y las habilidades adquiridas serán fundamentales para proyectos futuros y mi desarrollo profesional en la industria.

## POSIBLE EVOLUCIÓN Y MEJORA DEL PROYECTO

La evolución y mejora continua de un proyecto son aspectos cruciales para mantener la relevancia y eficacia a lo largo del tiempo. Aquí se presentan algunas sugerencias para la posible evolución y mejora del proyecto del Hotel "Aura de Mallorca":

### 1. Ampliación de Funcionalidades:

- Explore oportunidades para agregar nuevas funcionalidades que puedan mejorar la experiencia del usuario. Por ejemplo, podría considerar la integración de un sistema de comentarios y reseñas para las habitaciones y servicios.

### 2. Integración de Más Métodos de Pago:

- Actualmente, el proyecto utiliza Stripe como método de pago. Considere la posibilidad de integrar más opciones, como PayPal, para brindar a los usuarios una gama más amplia de opciones de pago.

### 3. Mejoras en la Interfaz de Usuario (UI) y Experiencia de Usuario (UX):

- Realice evaluaciones periódicas de la interfaz de usuario para identificar áreas de mejora en la navegación y presentación de información. Ajuste el diseño según las tendencias actuales y los comentarios de los usuarios.

### 4. Implementación de un Sistema de Reseñas y Retroalimentación:

- Introduzca un sistema que permita a los clientes dejar reseñas y comentarios sobre su experiencia en el hotel. Esto no solo proporcionará valiosa retroalimentación sino que también ayudará a atraer a nuevos clientes.

### 5. Desarrollo de una Aplicación Móvil:

- Considere la posibilidad de desarrollar una aplicación móvil para facilitar a los usuarios el acceso a la información y servicios del hotel desde dispositivos móviles. Esto puede mejorar la accesibilidad y la conveniencia.

### 6. Optimización para Motores de Búsqueda (SEO):

- Implemente estrategias de SEO para mejorar la visibilidad del sitio web en los motores de búsqueda. Esto puede incluir la optimización de palabras clave, metaetiquetas y la creación de contenido relevante.

### 7. Mejoras en la Seguridad:

- Realice auditorías de seguridad periódicas para identificar posibles vulnerabilidades. Mantenga los componentes del sistema y las bibliotecas actualizados para proteger contra posibles amenazas.

8. **Implementación de un Sistema de Recomendaciones:**

- Desarrolle un sistema que ofrezca recomendaciones personalizadas a los usuarios basadas en sus preferencias y comportamientos anteriores. Esto puede aumentar la satisfacción del cliente y fomentar la retención.

9. **Ampliación de la Cobertura de Idiomas:**

- Si el hotel atrae a una audiencia internacional, considere la posibilidad de expandir el sistema multi-idioma para cubrir más opciones lingüísticas y llegar a un público más amplio.

10. **Monitoreo y Análisis Continuo:**

- Implemente herramientas de monitoreo para realizar un seguimiento del rendimiento del sitio y del comportamiento del usuario. Utilice análisis para comprender mejor cómo los usuarios interactúan con la plataforma y tome decisiones informadas para futuras mejoras.

Estas sugerencias son puntos de partida, y la dirección específica de la evolución del proyecto dependerá de las necesidades cambiantes del hotel y de los comentarios de los usuarios. La mejora continua es clave para mantener la relevancia y la eficacia a largo plazo.

## Bibliografía

CIFP Pau Casesnoves (2023-2024). <https://paucasesnovescifp.cat/>

Vite (2023). Herramienta de generación Frontend. <https://vitejs.dev/>

Laravel (2023). Documentación Oficial. <https://laravel.com/docs/10.x>

NodeJS (2023). Entorno de programación usado para Frontend y Backend que utiliza Javascript como lenguaje para su uso. <https://nodejs.org/en/>

React (2023). Página Oficial. <https://react.dev/>

ExpressJS (2023). Página Oficial. <https://expressjs.com/>

NPM.js (2023). Repositorio centralizado para NPM de NodeJS (2023, de aquí se sacan todas las dependencias del proyecto). <https://www.npmjs.com/>

StackOverflow (2023). Sitio web por referencia para resolución de dudas y código en programación en general. <https://stackoverflow.com/>

React Bootstrap (2023). Framework para React con componentes ya hechos para un diseño coherente y cohesionado. <https://react-bootstrap.netlify.app/>

## Anexos

### 1. Manual de usuario

#### 1. *Introducción.*

Este manual de usuario tiene como objetivo ayudarte a utilizar la aplicación "Hotel Aura de Mallorca". Esta aplicación está diseñada para facilitar la reserva de habitaciones y servicios en el Hotel Aura de Mallorca.

#### 2. *Inicio rápido.*

Para empezar, asegúrate de tener Node.js instalado en tu máquina. Puedes descargarlo desde [nodejs.org](https://nodejs.org). Clona el repositorio del proyecto.

Abre una terminal en la carpeta raíz del proyecto y ejecuta los comandos:

*npm install y npm start*

Ahora, abre tu navegador web y accede a <http://localhost:3000> para ver la aplicación en acción.

#### 3. *Uso de la aplicación.*

Una vez en la aplicación, puedes explorar las habitaciones disponibles, seleccionar fechas y servicios adicionales, y realizar reservas.

Para acceder a tu cuenta, ve a la página de inicio de sesión y utiliza tus credenciales. Si no tienes una cuenta, puedes registrarte fácilmente.

#### 4. *Navegación y funcionalidad.*

La aplicación tiene una navegación sencilla. Puedes utilizar el menú de navegación en la parte superior para acceder a diferentes secciones.

#### 5. *Personalización.*

La aplicación no admite personalización en esta versión

#### 6. *Preguntas frecuentes.*

- Pregunta: ¿Cómo puedo modificar o cancelar una reserva?

Respuesta: Para modificar o cancelar una reserva, inicia sesión en tu cuenta y ve a la sección de reservas. Desde allí, puedes administrar tus reservas existentes.

#### 7. *Solución de problemas.*

- Problema: La aplicación no se inicia correctamente.

Solución: Asegúrate de haber ejecutado `npm install` en la carpeta del proyecto y de que Node.js esté instalado. Si sigues teniendo problemas, contacta al soporte técnico.

## 2. Manual de configuración

### 1. *Requisitos del sistema.*

Asegúrate de que tienes Node.js instalado. Puedes descargarlo desde [nodejs.org](https://nodejs.org). También necesitarás XAMPP para configurar el servidor web Apache y MySQL. Puedes descargar XAMPP desde [xampp.org](https://xampp.org). Abre Visual Studio Code para comenzar a trabajar en el proyecto.

### 2. *Configuración inicial.*

Abre XAMPP y asegúrate de que Apache y MySQL estén funcionando. Clona el repositorio del proyecto. Abre el proyecto en Visual Studio Code para comenzar la configuración.

### 3. *Configuración de variables de entorno.*

No se requieren variables de entorno adicionales para este proyecto.

### 4. *Configuración de bases de datos.*

Importa la base de datos MySQL incluida en el proyecto. Puedes hacerlo desde el panel de control de XAMPP. Configura las credenciales de la base de datos en los archivos de configuración del proyecto. Para más info, consulta el [Readme.md](#) general.

### 5. *Configuración completa al detalle (si no existe el **Readme**, aquí los pasos):*

#### ■ Configuración

Debemos cambiar el root password de mysql en XAMPP (si es windows, sino, después de instalar mariadb entraremos con ```sudo mysql o sudo mariadb``` y hacemos el ```CREATE USER 'x'@localhost IDENTIFIED BY 'password';``` y después hacer el ```GRANT ALL PRIVILEGES ON *.* TO 'x'@localhost IDENTIFIED BY 'password';```):

```Panel de control -> shell -> mysqladmin -u root password```

#### ■ ENVIRONMENT VARIABLES:

\* Frontend (.env):

\* FRONT\_URL = dominio de la web CON HTTP ya que se redirigirá automáticamente (ej: `http://aurademallorca.com`). Sin el espacio.

\* TRANSLATIONS\_DATA\_URL = ./

\* API\_URL = dominio de la web CON HTTPS (ej: `https://aurademallorca.com`). Sin el espacio.

\* Backend (.env):

\* API\_URL = dominio de la web CON HTTPS + /api (ej: `https://aurademallorca.com/api`). Sin el espacio.

\* FRONT\_URL = dominio de la web CON HTTP ya que se redirigirá automáticamente (ej: http ://aurademallorca.com). Sin el espacio.

\* CORS\_ORIGIN\_FRONT\_URL = dominio de la web SIN HTTP ya que se sólo es para identificación del dominio del cors, no del protocolo (ej: aurademallorca.com).

\* DB\_URL ='127.0.0.1', referente a localhost, si falla poner este.

NO OLVIDARSE DE PONER EL .htaccess en el root de la carpeta de la app (si es en /var/www/html o dentro de alguna carpeta como /var/www/html) y configurar en /etc/apache2/apache2.conf en el `<Directory>` un AllowOverride all.

PONER EN ADAPTADOR PUENTE LA MÁQUINA VIRTUAL LINUX SI SE USA.

#### ■ Instalación en Linux

##### • Requirements

```
```sudo apt update```
```

```
```sudo apt upgrade```
```

```
```sudo apt install openssh-server```
```

```
```sudo systemctl enable ssh```
```

```
```sudo apt install nodejs npm```
```

```
```sudo apt install mariadb-server mariadb-client -y```
```

```
```sudo systemctl enable mariadb```
```

```
```sudo mysql_secure_installation``` (poner la password a 1234 y darle a que sí a eliminar root externo y dbs de test y todo lo que dice)
```

```
```sudo apt install apache2 -y```
```

```
```sudo systemctl enable apache2```
```

```
```sudo apt -y install php php-common```
```

```
```sudo apt -y install php-cli php-fpm php-json php-pdo php-mysql  
php-zip php-gd php-mbstring php-curl php-xml php-pear php-  
bcmath```
```

```
```sudo apt -y install libapache2-mod-php```
```

```
```sudo a2enmod php8.*```
```

```
```sudo apt install net-tools```
```

```
```sudo apt install openssl -y```
```

```
```sudo a2enmod rewrite / sudo a2enmod headers```
```

```

```sudo a2enmod proxy / sudo a2enmod proxy_http```
```sudo systemctl restart apache2```
```sudo npm -g install pm2```
```sudo pm2 startup```
```sudo apt install ufw```
```sudo systemctl enable ufw```
```sudo ufw allow 80```
```sudo ufw allow 443```
```sudo ufw allow 3000```
```sudo ufw allow http```
```sudo ufw allow https```
```sudo ufw allow mysql```
```sudo ufw allow ssh```

```

IMPORTANTE: EL PROXYPASS DEFINE QUE SI LLAMO A EL DOMINIO DEL VIRTUAL HOST + lo que haya en el proxy pass, me redirige a otra IP o Dominio con un puerto que yo quiera. Ejemplo: ProxyPass /api http ://localhost:3000/api.

Esto me está redirigiendo las peticiones de https ://hotelaurademallorca.com/api (que puedo hacer con axios en el front) en la maquina destino donde esta el frontend servido en producción a redirigir la petición axios al localhost de esa máquina al puerto 3000 + /api endpoint donde escucha mi nodejs express.

- Configuración de la web

- Pre-config:

```

```sudo apt install phpmyadmin```
```sudo mysql < /usr/share/phpmyadmin/sql/create_tables.sql```

```

(si los 2 pasos anteriores fallan):

[Instalación de PHPMYAdmin manual en Debian]  
<https://www.digitalocean.com/community/tutorials/how-to-install-phpmyadmin-from-source-debian-10>)

```

```sudo mariadb```

```



```
```GRANT SELECT, INSERT, UPDATE, DELETE ON phpmyadmin.*
TO 'pma'@'localhost' IDENTIFIED BY 'password';``` (put the same
password that you gonna put in the .env of the app for the connection
and that u put in mariadb secure installation)
```

```
```GRANT ALL PRIVILEGES ON ON *.* TO 'admin'@'localhost'
IDENTIFIED BY 'password';``` (put the same password that you
gonna put in the .env of the app for the connection and that u put in
mariadb secure installation)
```

```
```GRANT ALL PRIVILEGES ON *.* TO 'fer'@'localhost' IDENTIFIED
BY 'password';``` (put the same password that you gonna put in
the .env of the app for the connection and that u put in mariadb
secure installation)
```

```
```GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost'
IDENTIFIED BY 'password';``` (put the same password that you
gonna put in the .env of the app for the connection and that u put in
mariadb secure installation)
```

```
```FLUSH PRIVILEGES;```
```

```
```EXIT;```
```

(if you installed phpmyadmin with the manual way link, you can avoid this, you choose the way you want to install phpmyadmin)

```
```wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz```
```

```
```sudo tar xvf phpMyAdmin-latest-all-languages.tar.gz```
```

```
```sudo mv phpMyAdmin-*-all-languages/
/var/www/html/phpmyadmin```
```

```
```cd /var/www/html/phpmyadmin```
```

```
```sudo cp config.sample.inc.php config.inc.php```
```

```
```sudo mkdir tmp```
```

```
```openssl rand -base64 32``` (copy the value given)
```

```
```sudo nano /var/www/html/phpmyadmin/config.inc.php```
```

Copy the value of pwgen into \$cfg['blowfish\_secret']

Uncomment all the lines of the sections: \*User used to manipulate with storage\* and \*Storage database and tables\* and one last line to this last mentioned section with: ```\$cfg['TempDir'] = '/var/lib/phpmyadmin/tmp';```

```
```sudo chown -R www-data:www-data /var/www/html/phpmyadmin```
```

(avoid until here)

```
``sudo nano /etc/apache2/conf-available/phpmyadmin.conf``: put
the content of phpmyadmin-virtualhost
```

```
``sudo a2enconf phpmyadmin.conf``
```

```
``sudo systemctl restart apache2``
```

- ACTIVAR SSL antes de config de la app (que se configura para https):

Seguir el tutorial de: [https://wiki.debian.org/Self-Signed\\_Certificate](https://wiki.debian.org/Self-Signed_Certificate), hacer cada paso con ``sudo``

- Config de la app:

1. Recibir el código fuente completo con git o con FTP.
2. Borrar los node\_modules de las carpetas frontend y backend por tema permisos con ``sudo rm -r nombrecarpeta/``.
3. Hacer un ``sudo npm install`` para cada carpeta.
4. Setear los .env de cada carpeta (con un sudo nano mismo).
5. Ir a la carpeta de frontend y generar el dist con ``sudo npm run build``.
6. ``sudo mkdir /var/www/html/hotelaurademallorca``
7. ``sudo chown www-data:www-data /var/www/html/hotelaurademallorca``
8. ``sudo chmod 755 /var/www/html/hotelaurademallorca``
9. Entrar al dist/ del frontend con ``cd`` y ejecutar ``sudo cp \* -r /var/www/html/hotelaurademallorca``
10. Ir a la carpeta root del código fuente y moveremos la carpeta backend por cuestiones de claridad con ``sudo mv backend/ /var/www/html/hotelaurademallorca``.
11. Nos movemos a esa carpeta backend con el comando ya visto y vamos a hacer: ``sudo pm2 start index.js`` y ``sudo pm2 save`` en este orden.
12. Creamos los virtualhosts de apache con ``sudo nano /etc/apache2/sites-available/hotelaurademallorca.conf`` (para HTTP) y ``sudo nano /etc/apache2/sites-available/hotelaurademallorca-ssl.conf`` (para HTTPS) y ponemos el contenido que necesitan los dos dentro de los archivos txt en el root del código fuente.

13. Activamos los virtualhosts con ``sudo a2ensite  
hotelaurademallorca.conf`` y ``sudo a2ensite  
hotelaurademallorca-ssl.conf`` y luego debemos hacer un ``sudo  
systemctl reload apache2``.

14. Agregamos a nuestro archivo /etc/hosts una línea nueva con:  
127.0.0.1 hotelaurademallorca.com

15. Hacemos lo mismo que el paso 13 pero en la máquina  
windows pero en vez de poner esa IP, ponemos la IP que tiene la  
VM o en donde esté alojado la app en ese momento (ej:  
192.168.1.102 hotelaurademallorca.com).

### 3. Manual de mantenimiento

#### 1. *Actualizaciones del proyecto.*

Para realizar actualizaciones en el proyecto, sigue los pasos del ciclo de desarrollo, que incluyen la modificación del código fuente y las pruebas.

#### 2. *Respaldo de datos.*

Para realizar copias de seguridad de la BDD, utiliza las herramientas proporcionadas por PHPMyAdmin o realiza copias manuales de los archivos SQL.

#### 3. *Solución de problemas comunes.*

- Problema: Error de conexión a la base de datos.

Solución: Asegúrate de que XAMPP esté en ejecución (o MySQL / MariaDB si está en producción) y que las credenciales de la base de datos sean correctas en los archivos de configuración.

#### 4. *Mantenimiento preventivo.*

Realiza actualizaciones periódicas del proyecto para asegurarte de que esté al día con las últimas tecnologías y prácticas de seguridad.

#### 5. *Soporte y recursos.*

Para obtener soporte adicional, contacta al equipo de soporte directamente.