

UNIVERSIDAD SAN FRANCISCO DE QUITO
USFQ

Colegio de Ciencias e Ingenierías

**Aplicación web para control y administración
hotelera**

Nicolás David Moina Iturralde
Ingeniería en ciencias de la computación

**Trabajo de fin de carrera presentado como requisito para la obtención del
título de Ingeniero en Ciencias de la Computación**

Quito, 9 de diciembre del 2024

UNIVERSIDAD SAN FRANCISCO DE QUITO

USFQ

Colegio de Ciencias e Ingenierías

**HOJA DE CALIFICACIÓN DE TRABAJO DE FIN
DE CARRERA**

**Aplicación web para control y administración
hotelera**

Nicolás David Moina Iturralde

Daniel Fellig Goldvechmiedt, MSc

Quito, 9 de diciembre del 2024

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual conforme a la Ley Orgánica de Educación Superior del Ecuador.

Nombres y apellidos: Nicolás David Moina Iturralde

Código: 00321192

Cédula de identidad: 1753066925

Lugar y fecha: Quito, 9 de diciembre del 2024

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en <http://bit.ly/COPETheses>.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on <http://bit.ly/COPETheses>.

Resumen

El proyecto que se realizara a continuación tiene como objetivo el desarrollo de una aplicación web para la gestión hotelera, optimizando las diferentes tareas administrativas que se realizan día a día, tales como el registro de clientes, reserva de habitaciones y facturación, ya que todo este se realiza de una forma muy rudimentaria, consumiendo mucho tiempo y siendo poco eficiente al momento de llevar un registro. La motivación principal es modernizar la industria hotelera utilizando las tecnologías mas actuales con el objetivo de mejorar el desempeño y reducir los costos. Para el desarrollo de la aplicación se utilizaran diferentes tecnologías, para el lado del Frontend tenemos NextJS, para el Backend AdonisJS, y MySQL como base de datos. Los resultados que se esperan obtener es una plataforma escalable y adaptable a diferentes tamaños de negocio, de igual forma se busca que esta sea lo mas modular posible para integraciones futuras.

Palabras clave: Aplicación web, gestión hotelera, automatización, AdonisJS, NextJS, eficiencia, escalabilidad, reserva de habitaciones, modernización.

Abstract

The project that will be carried out below aims to develop a web application for hotel management, optimizing the different administrative tasks that are carried out day to day, such as customer registration, room reservations and billing, since all this is done in a very rudimentary way, consuming a lot of time and being inefficient when keeping records. The main motivation is to modernize the hotel industry using the most current technologies with the aim of improving performance and reducing costs. Different technologies will be used to develop the application, for the Frontend we have NextJS, for the Backend AdonisJS, and MySQL as a database. The results that are expected to be obtained are a scalable platform and adaptable to different business sizes, in the same way it is sought that this is as modular as possible for future integrations.

Keywords: Web application, hotel management, automation, AdonisJS, NextJS, efficiency, scalability, room reservation, modernization.

Tabla de contenido

1	Introducción	8
1.1	Descripción del problema	8
1.2	Estado del arte	9
1.3	Objetivos	13
1.3.1	Objetivo general	13
1.3.2	Objetivos específicos	13
2	Desarrollo del Tema	14
2.1	Descripción de la aplicación	14
2.2	Actores	15
2.3	Casos de uso	16
2.3.1	Autenticación de usuarios	16
2.3.2	Administración de habitaciones	19
2.3.3	Administración de ventas	21
2.3.4	Administración de facturas	25
2.3.5	Administración de reservas	28
2.3.6	Administración de usuarios	31
2.4	Arquitectura de la aplicación	35
2.4.1	Aplicación Frontend	35
2.4.2	Aplicación Backend	39
2.4.3	Base de datos	43
2.4.4	Despliegue en la nube	45
3	Conclusiones	46

Índice de Figuras

2.1	La figura muestra el diagrama de secuencia del modulo de autenticación de usuarios	18
2.2	La figura muestra el diagrama de secuencia del modulo de habitaciones .	21
2.3	La figura muestra el diagrama de secuencia del modulo de ventas	24
2.4	La figura muestra el diagrama de secuencia del modulo de facturas	27
2.5	La figura muestra el diagrama de secuencia del modulo de reservas	30
2.6	La figura muestra el diagrama de secuencia del modulo de cuentas de usuario	34
2.7	La figura muestra la arquitectura general de la aplicación	35
2.8	La figura muestra el diagrama de la base de datos	44
3.1	Inicio de sesión del sistema.	51
3.2	Página de ventas.	52
3.3	Página de usuarios.	52

Introducción

1.1 Descripción del problema

A lo largo de los últimos años la industria hotelera ha enfrentado diferentes desafíos y se ha visto obligada a irse adaptando, en la actualidad esta industria requiere nuevas soluciones que están relacionadas con la gestión eficiente de las operaciones que se realizan día a día y al surgimiento de nuevos competidores como AirBnB, a lo largo de estos últimos 5 años se ha visto la gran capacidad que tienen estas nuevas empresa de atraer publico de todas las edades, ya que la mayoría de personas en muchas ocasiones prefieren esta opciones debido a sus bajos costos, por esta razón estas grandes empresas tecnológicas han llegado a tener una participación bastante considerable en el mercado haciendo que la industria de hotelera pierda miles de millones de dólares.

Ante esta problemática las grandes cadenas hoteleras están explorando nuevas opciones y volviendo mucho más eficiente su servicio y haciéndolo más atractivo para el público en general, por otro lado, los hoteles de pequeño y mediano tamaño se están quedando atrás con sistemas rudimentarios y que no se acoplan bien a las necesidades actuales del mercado. Al momento de conversar con uno de los dueños de estos hoteles nos dimos cuenta que diferentes tareas como la reserva de habitaciones, facturación, registro de clientes entre muchas otras se las realizan de manera manual lo cual puede llegar a resultar en una pérdida de eficiencia y en aumentar el margen de error al momento de la obtención de datos.

Este problema afecta mayoritariamente a establecimientos pequeños o medianos que por lo general no cuentan con los recursos necesarios para poder implementar soluciones

tecnológicas adecuadas y como consecuencia de esto los negocios no logran adaptarse a las demandas que el mercado requiere como por ejemplo la necesidad de una gestión eficiente en cuanto a la disponibilidad de habitaciones o el procesamiento de facturas.

De igual forma la ausencia de una solución tecnología puede llegar a afectar negativamente en su rentabilidad o en la experiencia del cliente haciendo que estos opten por otras opciones. Bajo este contexto surge una necesidad la cual es implementar una solución tecnología eficiente que sea capaz de manejar las tareas rutinarias del hotel de manera eficiente. Una aplicación web es una opción moderna y de fácil uso, capaz de adaptarse a diferentes tamaños de hoteles y a diferentes dispositivos, por esta razón se decidió centrar este proyecto en esa tecnología.

1.2 Estado del arte

El área de estudio que está involucrada con este proyecto se centra en la gestión hotelera a través de aplicaciones web, donde se abarcan diferentes conceptos como la automatización de procesos, experiencia de usuario e integración de sistemas.

Conceptos principales

- **Gestión hotelera digital:** La gestión digital en el campo hotelero tiene como objetivo el uso de tecnologías y aplicaciones para mejorar y optimizar las operaciones diarias que se realizan como, por ejemplo, reserva de habitaciones, atención al cliente o gestión de inventario, entre muchas otras.
- **Aplicaciones web:** Las aplicaciones web son programas que están alojados en internet y que pueden ser accedidos desde cualquier parte del mundo. En el contexto hotelero estas aplicaciones permiten tanto a empleados como huéspedes interactuar

con los sistemas del hotel desde cualquier dispositivo conectado a internet facilitando diferentes tareas.

- **Optimización de procesos:** La optimización de procesos en la gestión hotelera tiene como objetivo mejorar las operaciones internas del hotel con el fin de reducir costos, aumentar la eficiencia y mejorar la experiencia que tiene el cliente, esto se logra con la integración de sistemas y análisis de datos para la toma de decisiones informadas.

Aplicaciones en el mercado

- **Oracle Hospitality OPERA:** Esta aplicación como su nombre lo dice está desarrollada por la empresa Oracle y es una de las soluciones más completas en términos de gestión hotelera ofreciendo diferentes módulos, como por ejemplo reservas, gestión de ingresos y análisis de datos de igual forma este sistema puede llegar a integrarse con otros sistemas empresariales lo cual lo convierte en una gran herramienta para grandes cadenas hoteleras en el apartado de costos como ya sabemos los servicios de Oracle pueden llegar a ser muy costosos.
- **Hotelogix:** Esta es una plataforma que está integrada en la nube a diferencia de la opción que ofrece Oracle que es on premise y ofrece diferentes funcionalidades ya mencionadas cómo gestión de reservas, control de inventarios, facturación y gestión de clientes tiene una interfaz bastante intuitiva y de igual forma tiene una buena integración con diferentes canales de distribución lo cual es una opción popular entre hoteles de tamaño medio.
- **Cloudbeds:** Es una plataforma de gestión hotelera que ofrece diferentes soluciones en el ámbito hotelero ya que se encarga de diferentes aspectos como lo son opera-

ciones y finanzas, análisis de datos, distribución y marketing, y experiencia del huésped lo cual lo vuelve una solución bastante completa para hoteles de todos los tamaños.

- **RoomKeyPMS:** A diferencia del resto esta no es una plataforma web sino un programa que se instala, pero de igual forma posee funcionalidades parecidas al resto de soluciones ya presentadas, de igual forma se integra con diferentes sistemas externos y tiene su propio procesador de pagos lo cual lo vuelve una solución bastante atractiva.

Tecnologías web

El desarrollo de aplicaciones web a lo largo del tiempo a avanzado significativamente adaptando diferentes soluciones tecnológicas que permiten crear soluciones mucho más escalables, eficientes y accesibles desde cualquier lugar y desde cualquier dispositivo, en el contexto de la aplicación que se va a desarrollar las siguientes tecnologías juegan un papel crucial.

NextJS es un framework de desarrollo enfocando en aplicaciones web que esta basado en ReactJS otro framework que fue creado. NextJS tiene diversas ventajas en el apartado de optimización de rendimiento y experiencia de usuario como por ejemplo este se destaca por su capacidad de renderizado dentro del servidor (SSR) lo cual consiste en cargar la pagina antes de ser enviada al cliente, esto ayuda significativamente a los tiempo de carga iniciales lo cual es algo bueno cuando hablamos de SEO, esto se debe a que los motores de indexación posicionan mejor a paginas que están optimizadas. (Jartarghar et al., 2022; Lazuardy & Anggraini, 2022).

Por otra parte ReactJS se concibió con el fin de crear aplicaciones web en poco tiempo utilizando el paradigma de componentes reutilizables y personalizables, esto hace que el

código sea mucho mas fácil de leer y desarrollar (Lazuardy & Anggraini, 2022). La integración que tiene ReactJS con el DOM virtual (Documento Object Model) optimiza toda la actualización de la interfaz, ya que el paradigma que utiliza internamente este framework hace que únicamente se renderizen los elementos que han cambiado, mejorando así el rendimiento de la aplicación ya que no requiere hacer una recarga de la pagina esto es muy beneficioso tanto para el cliente como para el servidor (Jartarghar et al., 2022). De igual forma NextJS ayuda a facilitarnos la división del código, lo que permite que únicamente se carguen las partes necesarias de la aplicación, mejorando los tiempos de cargar ya que es un híbrido por lo que el código que nosotros deseemos podemos hacer que se renderice tanto en el cliente como en el servidor y de igual forma ayudando a la eficiencia del servidor al momento de enviar los componentes (Lazuardy & Anggraini, 2022). Todos estos beneficios son ideales para aplicaciones web grandes y que requieren un alto rendimiento.

AdonisJS es un framework que se basa en NodeJS y nos proporciona una estructura robusta, igual que varios competidores en el mercado como lo es Laravel que uno de los frameworks mas famosos para el desarrollo web en php. Una de las características mas importantes que tiene AdonisJS es la arquitectura MVC (Modelo vista controlador) que nos permite tener una gran modularidad al momento del desarrollo, separando diferentes responsabilidades facilitándonos la escalabilidad del código (Neumann et al., 2018). De igual forma este framework nos permite dividir el código en componentes que se pueden reutilizar, lo cual es una gran ventaja si deseamos mejorar nuestras aplicaciones en un futuro, haciendo que el desarrollo sea más sencillo de leer (Gamez-Diaz et al., 2017).

Otro de los beneficios que AdonisJS nos puede ofrecer es su propio sistema de ORM (Object Relational Mapping) que nos simplifica las interacciones que se hacen con la

base de datos, evitándonos escribir consultas SQL complejas y minimizando los errores que pueden existir cuando estamos manipulando datos internos. Esto se alinea con una tendencia general en cuanto al desarrollo backend, la cual es abstraer la complejidad y permitir a los desarrolladores enfocar en la lógica del negocio (Neumann et al., 2018).

1.3 Objetivos

1.3.1 Objetivo general

El objetivo general de este proyecto es desarrollar una aplicación web para la gestión de las operaciones hoteleras del día a día como, por ejemplo, registro de clientes, reserva de habitaciones y facturación, enfocándose en la facilidad de uso tanto para el personal interno el hotel como para los huéspedes.

1.3.2 Objetivos específicos

- Desarrollar un sistema de reservas que permita tanto al personal de hotel como a los clientes gestionar la reserva de habitaciones de forma rápida y eficiente.
- Implementación de un módulo para consultas de habitaciones donde el personal del hotel pueda acceder a la información actualizada sobre la disponibilidad y el estado de las habitaciones, optimizando el proceso de asignación y mantenimiento.
- Incorporar un sistema de gestión de clientes que permita al hotel llevar un registro detallado de los huéspedes para mejorar la personalización del servicio.
- Asegurar la escalabilidad y flexibilidad del sistema, desarrollando funciones que permitan futuras integraciones o adaptaciones de forma rápida.

Desarrollo del Tema

2.1 Descripción de la aplicación

Para esta aplicación la arquitectura consiste de las siguientes secciones y módulos:

- **Administración de habitaciones:** Dentro de este modulo el usuario de la aplicación dependiendo de sus roles administrativos podrá ver, crear, actualizar y eliminar habitaciones de igual forma podrá ver sus diferentes estados los cuales pueden ser Ocupado o Libre.
- **Ventas:** En este modulo el usuario dependiendo de sus roles administrativos podrá crear podrá ver, crear, actualizar y eliminar ventas, tener en cuenta que para poder realizar una venta primero se debe registrar a un cliente y de igual forma se debe seleccionar una habitación como mínimo.
- **Facturas:** En este modulo el usuario dependiendo de sus roles administrativos podrá ver, crear, actualizar y eliminar ventas, hay que tener en cuenta que para poder crear una factura se necesita una venta a la cual asociarla, de igual forma las facturas una vez creadas no se pueden eliminar únicamente se las puede anular.
- **Autenticación:** La aplicación tiene un sistema de registro e inicio de sesión robusto para evitar el ingreso al sistema de partes no autorizadas, para poder registrar a un usuario existe un usuario administrador que lo hace, para poder registrarlo primero se necesita un correo electrónico de la persona que va a ser registrada, una vez registrado el usuario el sistema enviara un correo con las credenciales que se

registraron en este caso usuario y contraseña, una vez con las credenciales el usuario nuevo debe cambiar la contraseña que se le genere y ya podrá hacer uso del sistema.

- **Reservas:** Este modulo se encarga de la administración de reservas, en este caso el usuario puede ver y eliminar reservas y por otra parte el cliente puede crear una reserva, hay que tener en cuenta que para poder realizar una reserva no se necesita tener una cuenta dentro del sistema ya que esto tiene que ser accedido por cualquier persona.
- **Usuarios:** Este modulo puede ser accedido únicamente por cuentas que tiene permisos de administración y este se encarga de agregar o quitar permisos a las cuentas que están creadas y de igual forma como ya se menciono anteriormente crear cuentas, hay que tener en cuenta que existen dos roles dentro de la aplicación, administrador y empleado, de igual forma existe una característica mas que es la de estado, la cual puede tener dos valores, activo o inactivo y esto nos ayuda a tener cuentas activas o inactivas, esto se realizo con el objetivo de que en caso de no querer permitir el acceso al sistema únicamente se desactive la cuenta y no se la elimine ya que el proceso de creación es un poco largo, con esto el administrador se ahorra tiempo.

2.2 Actores

- **Usuario:** Este actor tiene acceso al sistema del hotel, en este caso el acceso que este tenga dependerá del rol que se le haya asignado al momento de crearle la cuenta, tener en cuenta que se pueden agregar o quitar roles según lo vaya necesitando.
- **Usuario administrador:** El actor tiene acceso completo al sistema del hotel

para poder administrarlo como el desee, de igual forma este puede crear y eliminar usuarios, por otra parte puede modificar los roles para que los usuarios puedan tener mas libertades o restricciones dentro del sistema.

- **Cliente:** Este actor únicamente puede crear reservas en el sistema.

2.3 Casos de uso

2.3.1 Autenticación de usuarios

Inicio de sesión

Este caso de uso nos muestra el proceso de iniciar sesión dentro del sistema de administración, los actores que participan dentro de este caso son los usuarios y usuarios administradores por lo tanto su flujo de operación es el siguiente.

1. El usuario ingresa al portal web en la sección de iniciar sesión, e ingresa su usuario y contraseña en el formulario que se le presenta.
2. La aplicación verifica si las credenciales ingresadas son correctas.
3. En caso de que estas credenciales sean incorrectas se lanza un error indicando al usuario que las credenciales son incorrectas, por otra parte si estas son correctas el sistema hace una redirección hacia el dashboard de la aplicación

Solicitud de restablecimiento de contraseña

De igual forma este caso de uso tiene como objetivo si el usuario olvida su contraseña este pueda solicitar un cambio dentro de la aplicación, este caso de uso tiene el siguiente flujo.

1. Dentro del formulario de inicio de sesión el usuario da click en el enlace que dice olvidaste tu contraseña.
2. Se redirige al usuario a otro formulario donde se le pide que ingrese el correo electrónico registrado en la aplicación.
3. La aplicación verifica si el correo esta registrado dentro del sistema y en caso de no encontrar ningún correo asociado se le notifica al usuario que el correo fue enviado, por otra parte si el sistema encuentra el correo la aplicación genera un token de acceso y lo guarda en la base de datos, después se le envía el correo con el token al usuario.

Restablecimiento de contraseña

Este caso de uso nos indica el proceso que realiza el usuario para poder hacer el restablecimiento de su contraseña

1. El usuario revisa su correo electrónico y abre el correo de solicitud de restablecimiento de contraseña, y hace click en el enlace.
2. Una vez ingresado al enlace el sistema comprueba el token y verifica que este no haya sido alterado o que no haya expirado, el token tiene una duración de 1 hora.
3. En caso de que el token no haya sido alterado o no haya expirado se le muestra al usuario un formulario en donde puede cambiar su contraseña se le pide que vuelva a ingresar la contraseña por razones de seguridad y la aplicación verifica que estas coincidan en caso de que no, se muestra un mensaje de error y en caso de que si coincidan se hace el cambio de contraseña.

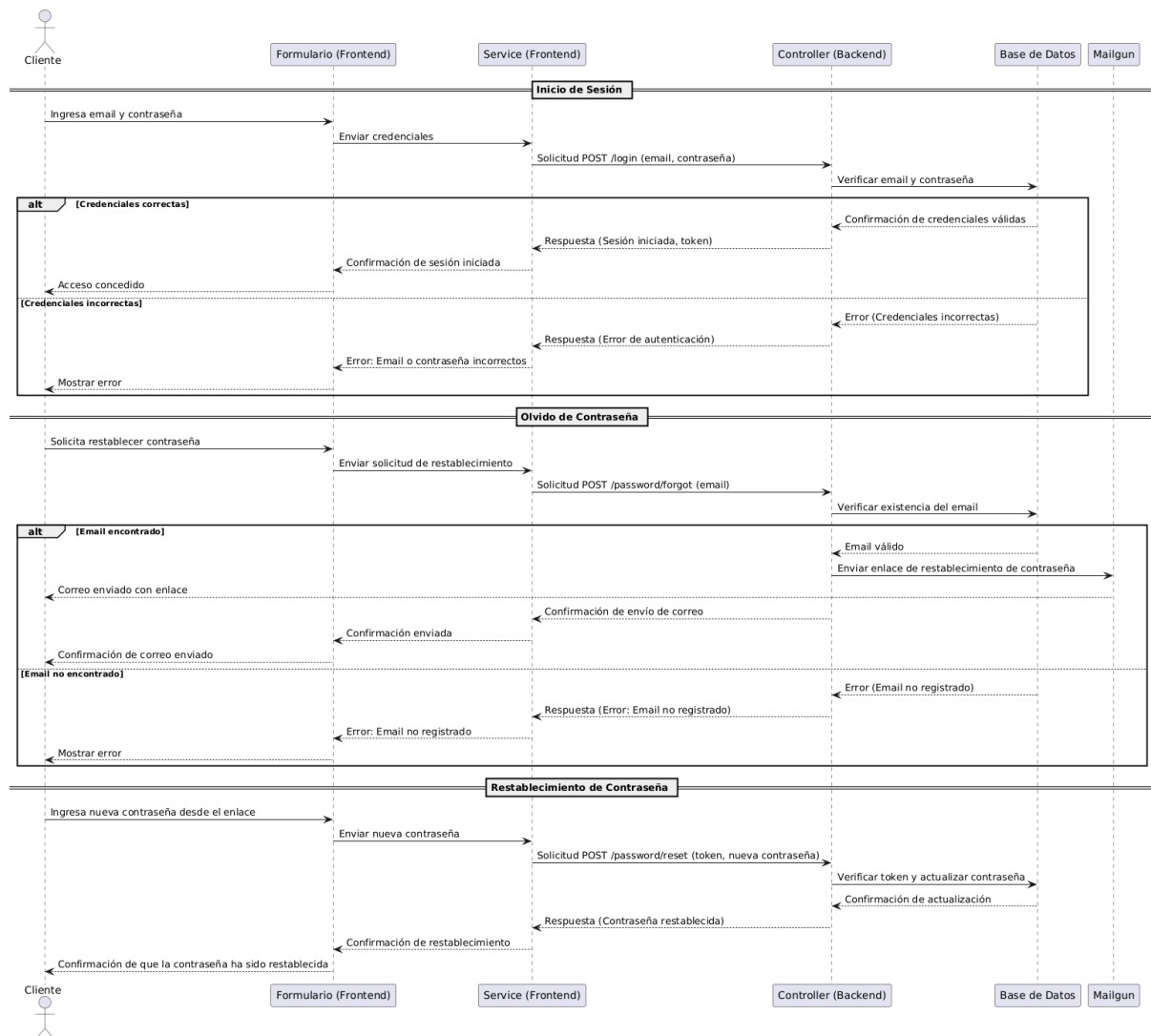


Figure 2.1: La figura muestra el diagrama de secuencia del modulo de autenticación de usuarios

2.3.2 Administración de habitaciones

Creación de habitación

Para este caso de uso se nos muestra el proceso de crear una habitación dentro de la aplicación, los actores que participan son los usuarios con roles de administrador y además usuarios con roles regulares y el flujo es el siguiente.

1. El usuario accede al sistema y se dirige al apartado de habitaciones.
2. Una vez dentro de esa sección el usuario hace click en crear nueva habitación, una vez hecho click se abre un formulario que permite el ingreso de los datos.
3. El usuario ingresa los datos que se le solicitan en este caso son, numero de habitación, descripción, numero de camas, y el listado de precios para diferentes personas, tener en cuenta que el listado de precios puede variar dependiendo del numero de personas ingresen a la habitación.
4. Una vez ingresados los datos el sistema verifica que estos sean correctos, en caso de que estos sean incorrectos el sistema lanza un error mostrando los errores, por otra parte si no existen errores el sistema guarda los datos dentro de la base de datos

Actualización de habitación

Este caso de uso nos muestra el proceso que se realiza para modificar los datos de una habitación ya creada dentro del sistema, los actores que están involucrados son los usuarios con permisos de administrador y los usuarios regulares.

1. El usuario ingresa a la aplicación en el apartado de habitaciones, y selecciona la habitación que desea modificar.

2. La aplicación despliega el formulario con los datos actuales de la habitación para que el usuario los edite.
3. El usuario ingresa los datos que desea modificar y envía el formulario.
4. El sistema verifica que estos sean correctos, en caso de que estos sean incorrectos se lanza un error y se muestra cuales campos tienen errores, por otra parte si los datos son correctos se guardan en la base de datos.
5. Finalmente se notifica al usuario que la habitación fue modificada exitosamente.

Visualización de habitaciones

Este caso de uso detalla el proceso en el que cualquier usuario puede consultar la información de las habitaciones disponibles. El flujo es el siguiente:

1. El usuario accede al portal web y selecciona la sección de habitaciones disponibles.
2. La aplicación muestra una lista de habitaciones con detalles como el número, tipo de habitación, número de camas, estado y una breve descripción.
3. El usuario puede filtrar o buscar habitaciones específicas según sus necesidades.

Eliminación de habitación

Este caso de uso nos muestra el proceso para eliminar una habitación del sistema, los actores que están involucrados son los usuarios con permisos de administración y los usuarios regulares. El flujo es el siguiente:

1. El administrador accede a la sección de habitaciones y selecciona la que desea eliminar.
2. La aplicación solicita una confirmación del administrador para proceder con la eliminación.

3. Si el administrador confirma la operación, la aplicación elimina la habitación de la base de datos y lo notifica al administrador. En caso de que no se confirme, la operación es cancelada.

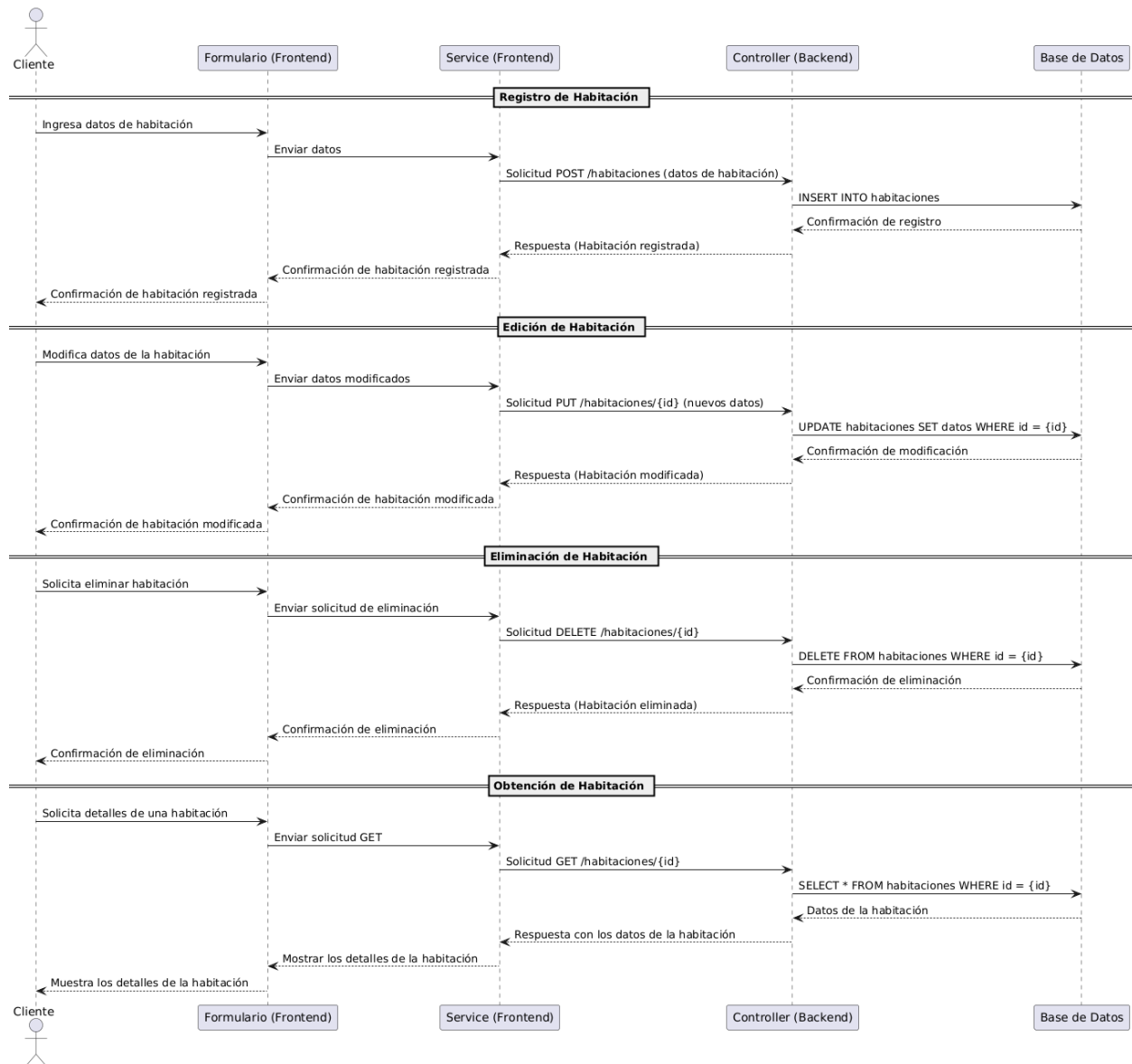


Figure 2.2: La figura muestra el diagrama de secuencia del módulo de habitaciones

2.3.3 Administración de ventas

Registro de ventas

Este caso de uso nos muestra el proceso que se tiene que seguir para poder registrar

una venta dentro del sistema. Los actores que participan son administradores y usuarios regulares, el flujo de registro es el siguiente.

1. El usuario de la aplicación accede a la aplicación web y selecciona el apartado de nueva venta dentro de el menú.
2. Al ingresar se le mostrara un formulario que deberá llenar, este se divide en dos partes, el primer apartado es el de registro de personas, dentro de este formulario se tiene un botón llamado validar cédula, en este se ingresa la cédula del cliente en caso de que si exista el cliente la información de este se auto completara, por otra parte si no existe el cliente se mostrara un mensaje de que el cliente no existe y se pedirá que se ingresen los datos para poder continuar con la venta.
3. Una vez ingresados a los clientes se puede comenzar con la segunda parte de la venta que es la selección de fechas y de habitaciones, una vez seleccionados, se mostraran los precios y el formulario se envía.
4. La aplicación valida que los datos ingresados sean correctos y en caso de no sello se informara al usuario que la información es incorrecta, en caso de ser correcta la información se envía a la base de datos.
5. Finalmente el sistema le confirma al usuario que la venta ha sido registrada con éxito.

Actualización de ventas

Este caso de uso explica el proceso para modificar los detalles de una venta existente. El flujo es el siguiente:

1. El usuario dentro de la aplicación se va al apartado de ventas.

2. La aplicación muestra una tabla con todas la ventas realizadas, y el usuario selecciona la venta que desea modificar, una vez seleccionada la venta se le redirige al usuario a una pagina donde puede editar la venta
3. Una vez redirigido el usuario, modifica los datos que desea y envía el formulario.
4. La aplicación valida que los datos ingresados sean correctos y en caso de no sello se informara al usuario que la información es incorrecta, en caso de ser correcta la información se envía a la base de datos.
5. Finalmente el sistema le confirma al usuario que la venta ha sido actualizada con éxito.

Visualización de ventas

Este caso de uso detalla el proceso para consultar las ventas realizadas. Tanto administradores como usuarios regulares pueden acceder a esta información. El flujo es el siguiente:

1. El usuario accede a la sección de ventas dentro del portal.
2. La aplicación muestra una lista de ventas realizadas, incluyendo detalles como el cliente, las fechas de estadía, las habitaciones reservadas y el total pagado.
3. El usuario puede buscar o filtrar las ventas por diferentes criterios, como fecha o cliente.

Cancelación de ventas

Este caso de uso describe el proceso para cancelar una venta registrada. Solo los administradores tienen permisos para realizar esta operación. El flujo es el siguiente:

1. El administrador selecciona una venta previamente registrada desde la lista de ventas.
2. La aplicación solicita una confirmación para proceder con la cancelación.
3. Si el administrador confirma la cancelación, la venta es marcada como cancelada en la base de datos y se notifica al administrador.

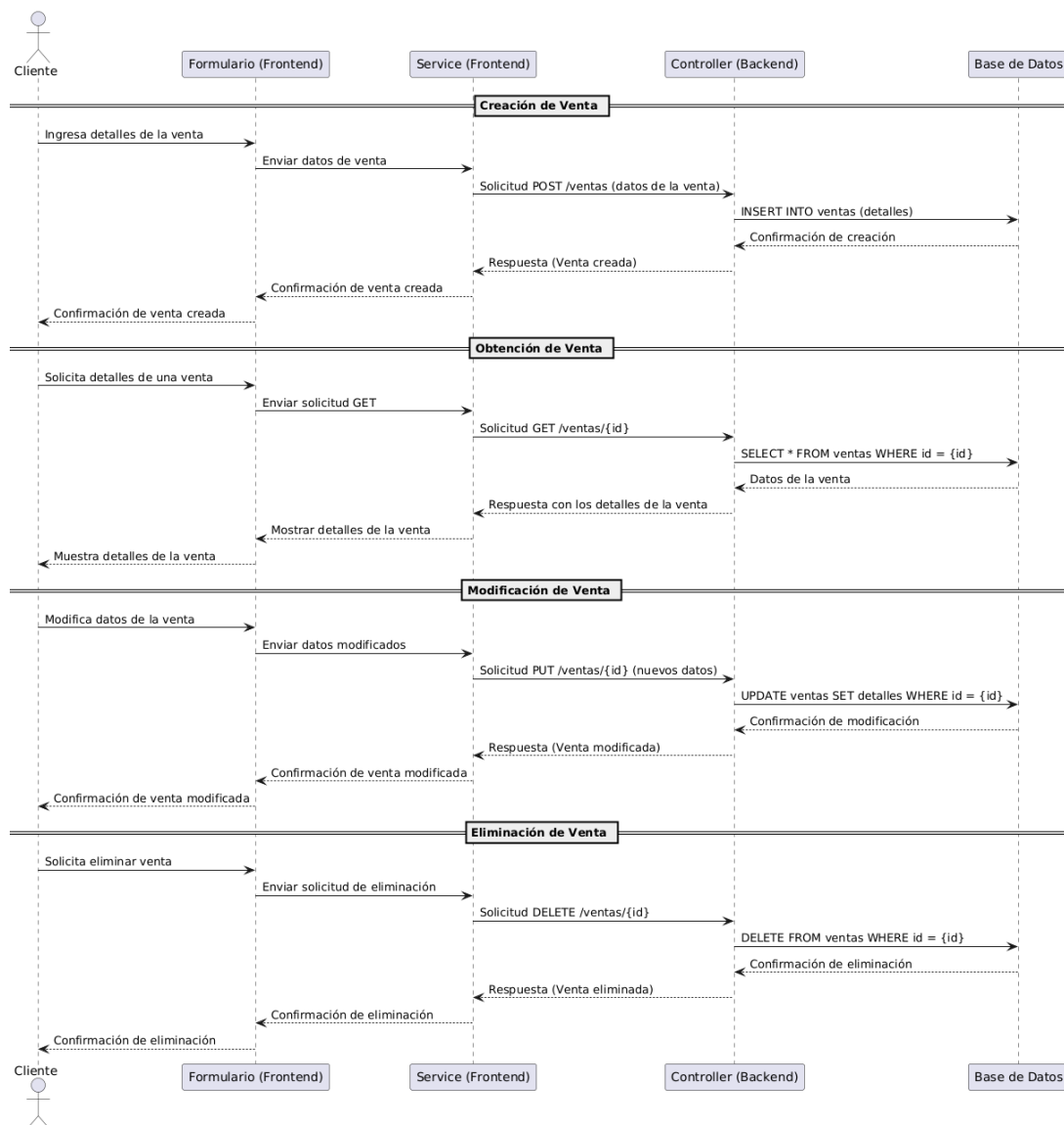


Figure 2.3: La figura muestra el diagrama de secuencia del modulo de ventas

2.3.4 Administración de facturas

Generación de facturas

Este caso de uso describe el proceso de generación de una factura tras la confirmación de una venta. Los actores involucrados son administradores y usuarios regulares.

1. El usuario accede al sistema en la sección de ventas y selecciona una venta previamente registrada para generar una factura.
2. La aplicación verifica los datos de la venta, como el cliente, las habitaciones reservadas y las fechas de estadía, y calcula el total a facturar.
3. El sistema genera automáticamente una factura con los detalles de la venta, que incluye el número de la factura, los datos del cliente, las fechas de estadía, las habitaciones reservadas, el total de la venta y cualquier impuesto aplicable.
4. La factura generada se guarda en la base de datos y se le muestra al usuario un mensaje de confirmación.
5. Opcionalmente, se le envía una copia de la factura al correo electrónico del cliente.

Actualización de facturas

Este caso de uso detalla el proceso de modificar una factura previamente generada. Los actores que participan son administradores y usuarios regulares.

1. El usuario accede a la sección de facturas y selecciona la factura que desea modificar.
2. El sistema muestra los detalles actuales de la factura, permitiendo al usuario editar la información.
3. El usuario edita los campos requeridos, como los datos del cliente o el total, y envía el formulario.

4. La aplicación valida los datos ingresados y, en caso de ser correctos, actualiza la factura en la base de datos.
5. Finalmente, el sistema notifica al usuario que la factura ha sido actualizada con éxito.

Visualización de facturas

Este caso de uso muestra el proceso de visualización de facturas generadas. Tanto administradores como usuarios regulares pueden consultar esta información.

1. El usuario accede a la sección de facturas dentro del portal.
2. La aplicación muestra una lista de facturas generadas, con detalles como el número de la factura, el cliente, las fechas de estadía, el total y el estado de pago.
3. El usuario puede buscar o filtrar las facturas según criterios como fecha, cliente o estado de pago.

Eliminación de facturas

Este caso de uso describe el proceso para eliminar una factura del sistema. Solo los administradores pueden realizar esta operación.

1. El administrador accede a la sección de facturas y selecciona la factura que desea eliminar.
2. La aplicación solicita una confirmación para proceder con la eliminación.
3. Si el administrador confirma, la aplicación elimina la factura de la base de datos y lo notifica al administrador.
4. En caso de que no se confirme, la operación es cancelada.

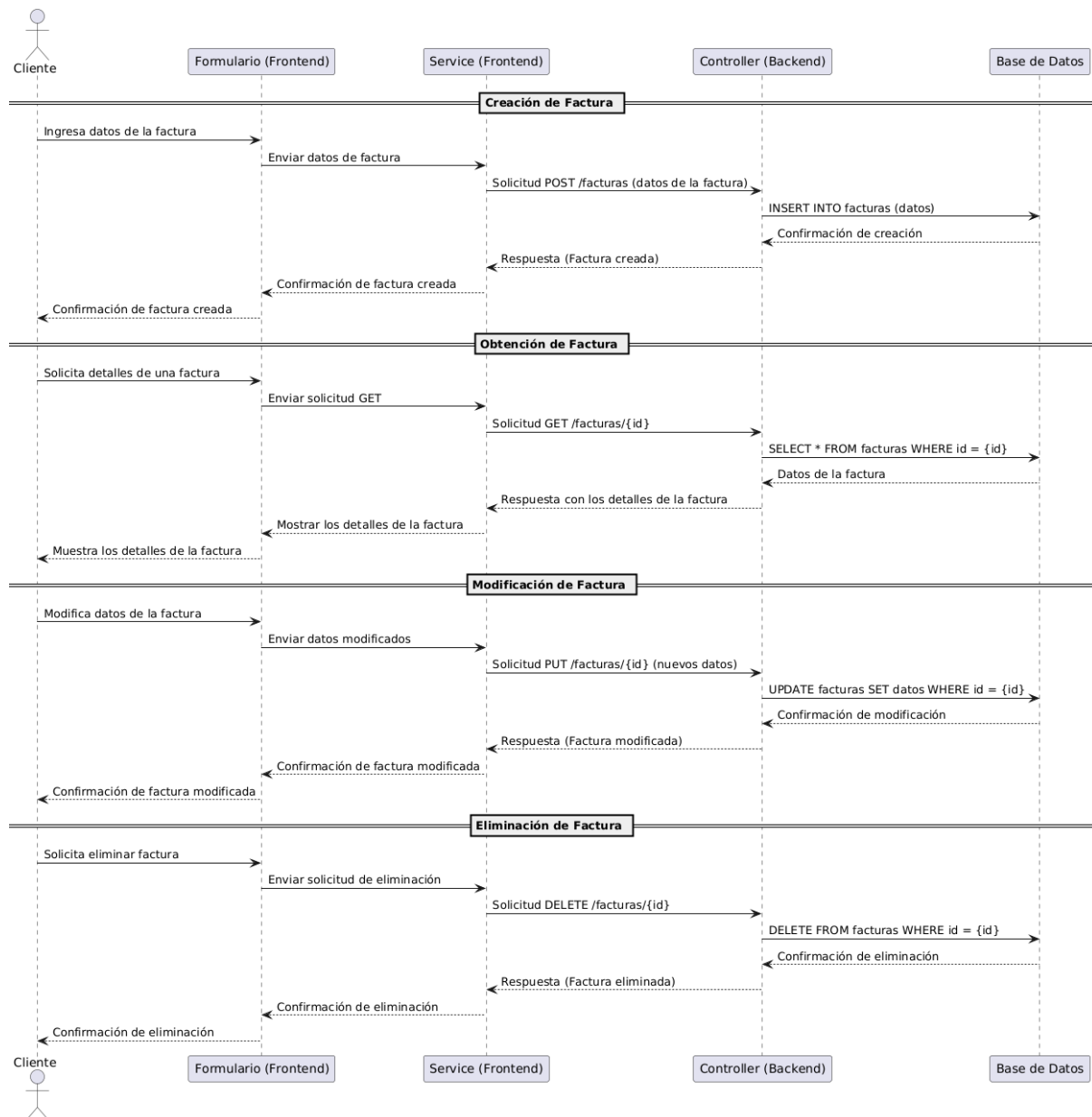


Figure 2.4: La figura muestra el diagrama de secuencia del modulo de facturas

2.3.5 Administración de reservas

Creación de reservas

Este caso de uso describe el proceso para registrar una nueva reserva dentro del sistema de administración hotelera. Los actores que participan son administradores y usuarios regulares.

1. El usuario accede al sistema en la sección de reservas y selecciona la opción de crear una nueva reserva.
2. Se le presenta un formulario donde debe ingresar los detalles de la reserva, como nombre, apellido, fechas de inicio y fin, número de personas, y selección de habitaciones disponibles.
3. El usuario selecciona las habitaciones a reservar y la aplicación muestra el precio total basado en el número de personas y el número de días.
4. El usuario confirma la reserva, y la aplicación valida que las habitaciones seleccionadas estén disponibles en las fechas ingresadas.
5. En caso de que los datos sean correctos, la reserva se guarda en la base de datos y el sistema redirige al usuario a la confirmación de la reserva.
6. Se genera una confirmación que puede enviarse al correo del cliente.

Actualización de reservas

Este caso de uso describe el proceso para modificar los detalles de una reserva existente.

1. El usuario accede al sistema en la sección de reservas y selecciona la reserva que desea modificar.

2. La aplicación muestra el formulario con los datos actuales de la reserva para que el usuario los edite.
3. El usuario modifica los campos necesarios, como las fechas de estadía o el número de personas, y envía el formulario.
4. La aplicación valida que las habitaciones estén disponibles en las nuevas fechas ingresadas y que los demás datos sean correctos.
5. Si los datos son correctos, la reserva se actualiza en la base de datos y se notifica al usuario que la reserva ha sido modificada exitosamente.

Visualización de reservas

Este caso de uso detalla el proceso para consultar las reservas realizadas.

1. El usuario accede a la sección de reservas dentro del sistema.
2. La aplicación muestra una lista de reservas, con detalles como el cliente, las fechas de estadía, las habitaciones reservadas, el estado de la reserva (pendiente, confirmada, cancelada) y el total.
3. El usuario puede buscar o filtrar las reservas por criterios como fecha, cliente o estado de la reserva.

Cancelación de reservas

Este caso de uso describe el proceso para cancelar una reserva.

1. El usuario accede a la sección de reservas y selecciona la reserva que desea cancelar.
2. La aplicación solicita una confirmación del usuario para proceder con la cancelación.

3. Si el usuario confirma, la aplicación marca la reserva como cancelada en la base de datos y notifica al usuario.
4. El sistema también puede enviar una notificación de cancelación al correo electrónico del cliente.

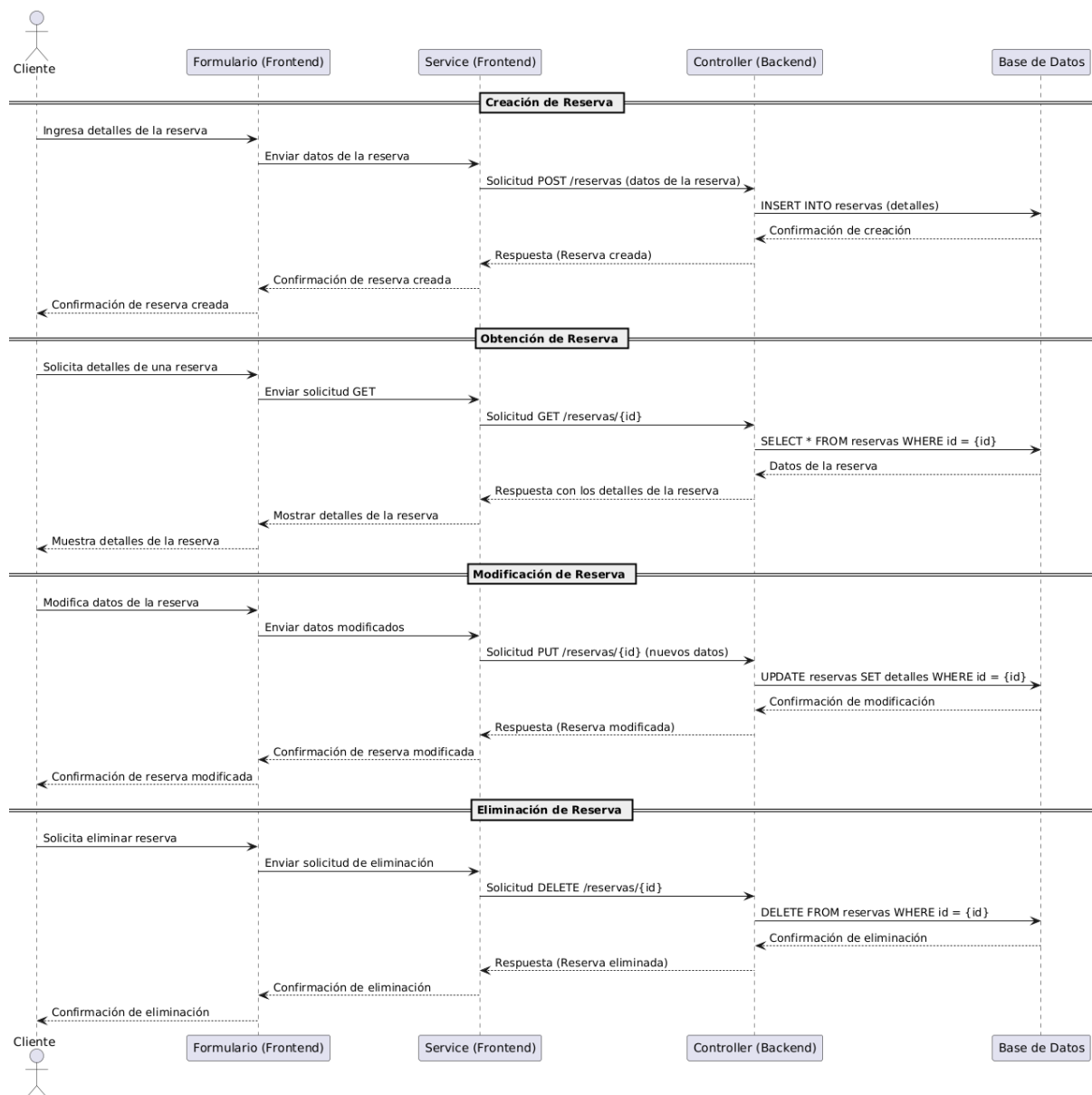


Figure 2.5: La figura muestra el diagrama de secuencia del modulo de reservas

2.3.6 Administración de usuarios

Creación de cuentas de usuario

Este caso de uso describe el proceso que un administrador sigue para crear una nueva cuenta de usuario dentro del sistema.

1. El administrador accede al sistema y navega a la sección de Administración de usuarios.
2. El administrador selecciona la opción Crear nuevo usuario.
3. La aplicación presenta un formulario donde se solicitan los datos del nuevo usuario, como: nombre, apellido, nombre de usuario, correo electrónico, rol (administrador o empleado) y estado (Activo o Inactivo).
4. El administrador llena el formulario con los datos del nuevo usuario y selecciona el estado de la cuenta (activo o inactivo).
5. La aplicación verifica que todos los campos estén correctamente completados y que no haya errores.
6. Si la información es válida, el sistema guarda los datos del usuario en la base de datos y la cuenta es creada con éxito, posteriormente se envía un correo al usuario que fue creado con las credenciales para que pueda ingresar al sistema, se le crea una contraseña temporal y luego podrá restablecer la contraseña.
7. Finalmente, el sistema notifica al administrador que el usuario ha sido creado exitosamente.

Actualización de cuentas de usuario

Este caso de uso detalla el proceso para actualizar la información o los permisos de una cuenta de usuario existente.

1. El administrador accede a la sección de Administración de usuarios y selecciona la cuenta de usuario que desea modificar.
2. La aplicación despliega un formulario con la información actual del usuario, permitiendo al administrador editar campos como el rol (administrador o empleado) y el estado (activo o inactivo).
3. El administrador realiza los cambios necesarios en el formulario y envía los datos.
4. La aplicación valida los datos modificados y, en caso de que sean correctos, actualiza la información en la base de datos.
5. El sistema notifica al administrador que la cuenta ha sido actualizada correctamente.

Desactivación de cuentas de usuario

Este caso de uso describe el proceso que sigue un administrador para desactivar una cuenta de usuario en lugar de eliminarla.

1. El administrador accede a la sección de Administración de usuarios y selecciona la cuenta que desea desactivar.
2. La aplicación muestra una confirmación solicitando la acción de desactivar la cuenta.
3. El administrador confirma la acción y la aplicación cambia el estado de la cuenta a inactivo.
4. El sistema guarda el nuevo estado en la base de datos y notifica al administrador que la cuenta ha sido desactivada exitosamente.

Reactivación de cuentas de usuario

Este caso de uso describe cómo un administrador puede reactivar una cuenta previamente desactivada.

1. El administrador accede a la sección de Administración de usuarios y selecciona la cuenta que está en estado inactivo.
2. La aplicación muestra una opción para reactivar la cuenta.
3. El administrador confirma la reactivación y el sistema cambia el estado de la cuenta a activo.
4. La aplicación guarda el nuevo estado en la base de datos y notifica al administrador que la cuenta ha sido reactivada.

Visualización de cuentas de usuario

Este caso de uso muestra el proceso que sigue un administrador para visualizar todas las cuentas de usuario registradas en el sistema.

1. El administrador accede a la sección de Administración de usuarios.
2. La aplicación muestra una lista de todas las cuentas, incluyendo detalles como el nombre, correo electrónico, rol (administrador o empleado) y estado (activo o inactivo).
3. El administrador puede buscar o filtrar usuarios según su rol o estado.

Eliminación de cuentas de usuario Este caso de uso detalla el proceso de eliminar una cuenta de usuario del sistema.

1. El administrador selecciona la cuenta de usuario que desea eliminar dentro de la sección Administración de usuarios.

2. La aplicación solicita una confirmación para proceder con la eliminación.
3. Si el administrador confirma la operación, la cuenta se elimina de la base de datos.
4. El sistema notifica al administrador que la cuenta ha sido eliminada exitosamente.



Figure 2.6: La figura muestra el diagrama de secuencia del modulo de cuentas de usuario

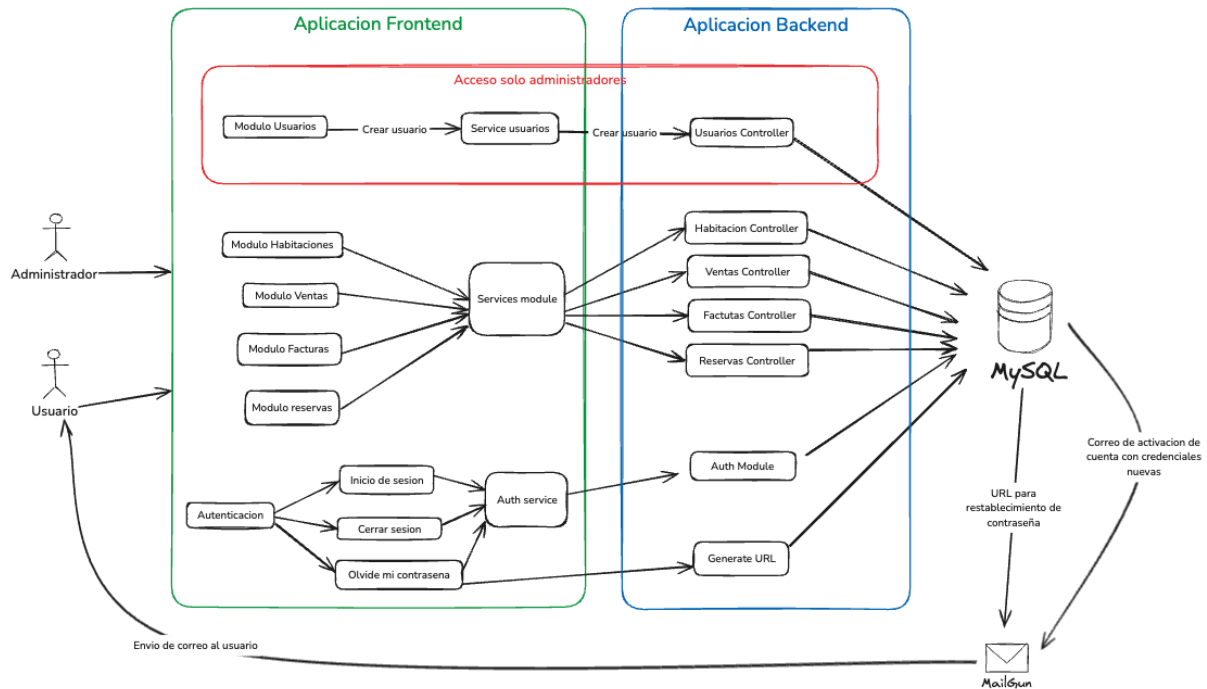


Figure 2.7: La figura muestra la arquitectura general de la aplicación

2.4 Arquitectura de la aplicación

2.4.1 Aplicación Frontend

Tecnologías y herramientas utilizadas

La tecnologías que se usan en una aplicación web son de extrema importancia por esta razón se tiene que ofrecer un diseño interactivo y una experiencia de usuario optimizada. El diseño de la aplicación podría llegar a impactar directamente en su éxito especialmente en aplicaciones que son de alta concurrencia (Flavián, Gurrea & Orús, 2008), como lo podrían ser sistemas para la gestión hotelera donde el usuario esta constantemente interactuando con los componentes de la misma, por lo cual se requiere que todas estas iteraciones sean lo mas intuitivas y eficientes posibles.

Se decidió escoger NextJS como framework principal ya que este nos aporta el renderizado por parte del servidor, ofreciéndonos un enfoque híbrido entre renderizado de ciertos

componentes en el cliente y otros en el servidor, esto tiene diferentes ventajas frente a otros competidores, ya que los tiempos de carga en comparación a otras paginas web sera mucho menores mejorando el posicionamiento que esta tiene en los buscadores, de igual forma en casos donde la conexión no es muy buena es una muy buena opción. Según Iskandar et al. (2020), el SSR es muy ventajoso en aplicaciones que utilizan una gran cantidad de datos y ademas requieren actualizaciones constantes.

Para la interfaz de usuario se decidió usar TailwindCSS debido a que esta tecnología tiene un enfoque utility first, esto permite que los desarrolladores pueden construir estilos de manera rápida y consistente. Comparado con otras herramientas de estilizado Tailwind-CSS permite diseño mas ágil gracias a sus clases predefinidas, evitando al desarrollador escribir código de CSS extenso en un archivo separado.

Se decidió integrar un control de versiones como Git dentro del proyecto ya que es sumamente importante tener un historial claro sobre los cambios que ser van realizando a lo largo del tiempo, todo el código se subió periódicamente a un repositorio de Github donde esta disponible todo el código para que cualquier persona lo pueda visualizar.

Estado y manejo de datos

Una de las cosas más importantes al momento en el que desarrollamos una aplicación es la gestión del estado y los datos, dentro del marco de la gestión hotelera es crucial garantizar que todas las operaciones dentro de nuestro sistema sean eficientes y óptimos para la experiencia de usuario. Las tecnologías de la información son importantes cuando hablamos de análisis predictivo, una ventaja de esto es mejorar la toma de decisiones dentro del negocio. Por ejemplo el uso de sistemas basados en la nube han demostrado ser sumamente efectivos para consolidar datos y garantizar una alta disponibilidad facilitando un acceso dinámico y confiable a la información (González et al., 2019). Bajo este contexto

es muy importante capturar los datos de manera eficiente de modo que permita a los administradores tomar decisiones en tiempo real (Yong-ye, 2009).

En cuanto a la arquitectura y la sincronización de datos se usa APIs RESTful Ya que nos permite gestionar los datos de una forma eficiente teniendo una comunicación directa entre el frontend y el backend (Bilgihan et al., 2011). De igual forma el manejo del estado de la aplicación en tiempo real se lo realiza haciendo uso de tecnologías como Context API de React, que facilita la gestión del estado entre componentes de la aplicación mejorando tanto la legibilidad en el código como en el rendimiento del sistema. Gracias a esto el sistema es capaz de soportar actualizaciones de datos en el backend y que éstas se reflejen inmediatamente en el frontend garantizando una alta disponibilidad a la información (Klein et al., 2020).

Seguridad y autenticación

La seguridad dentro de las aplicaciones es de suma importancia y más cuando hablamos de aplicaciones de gestión hotelera, esto nos ayuda a mantener la integridad de los datos y a prevenir accesos no autorizados. Dentro de esta aplicación se implementaron diferentes estrategias para que la seguridad no se vea comprometida, una de estas es el control de acceso basado en roles lo que nos permite asignar diferentes roles a usuarios dependiendo del papel que estos tengan dentro del establecimiento como por ejemplo administrador o empleado (Zhao et al., 2007). Al implementar este enfoque reducimos en gran medida la complejidad de la gestión de los permisos dándoles únicamente acceso a lo que se requiere y no a toda la aplicación. Por otra parte también se implementó la posibilidad de habilitar o deshabilitar cuentas desde un panel administrativo, esto nos proporciona un nivel adicional de seguridad y rapidez al momento de otorgar accesos (Liu & Li, 2023). Por otra parte para que se pueda garantizar la autenticidad de los usuarios y de igual

forma agregar un nivel de protección adicional a las sesiones de nuestra aplicación se optó por el uso de JSON web tokens, esta es una práctica bastante utilizada en arquitecturas web modernas. Los JWT nos permiten autenticar a usuarios de manera segura y confiable, la forma en la que éstos funcionan es que los tokens son firmados digitalmente dentro de la aplicación entregándole al usuario un token que es válido durante un cierto período de tiempo (Almeida & Canedo, 2022). Esta implementación junto con otros protocolos estándares permiten una comunicación segura entre los diferentes servicios de la aplicación frontend y backed evitando accesos indebidos. De igual forma al momento en el que implementamos estas tecnologías como por ejemplo la autenticación continua ofrece una protección adicional frente a ataques de suplantación de identidad (Junquera-Sánchez et al., 2021).

Estilos y Diseño UI/UX

El diseño de UX/UI dentro de una aplicación es sumamente importante por esta razón se desarrolló una aplicación responsivo desarrollada con TailwindCSS, esto nos garantiza una experiencia fluida y adaptable a diferentes entornos cómo los pueden ser escritorio móvil o tablet. Según algunas investigaciones un diseño donde se prioriza la responsividad donde la aplicación es adaptable a diferentes tamaños de pantalla mejora mucho más la interacción que tiene el usuario con esta aplicación aumentando la accesibilidad y la apertura hacía mucha más gente (Dospinescu et al., 2017). TailwindCSS tiene un enfoque de clases utilitarias que facilita mucho más la lectura y escritura del código haciéndolo mucho más eficiente y menos complejo al momento en el que se desarrolla.

Para el desarrollo de esta aplicación se implementaron ciertos principios de diseño que se centra en el usuario estos nos garantizan interfaces intuitivas que mejoran mucho más la satisfacción que tiene el usuario además de la retención (Firmansyah et al., 2023). La

consistencia visual y la simplicidad de los componentes son sumamente importantes en aplicaciones responsivas, esto nos ayuda a que las funcionalidades internas que tiene la aplicación sean mucho más accesibles independientemente del dispositivo en el cual se esté usando. Por otra parte el enfoque de mobile first de TailwindCSS Nos ayuda a priorizar el desarrollo en dispositivos móviles optimizando la usabilidad en estos dispositivos y cumpliendo con las expectativas de los usuarios en la actualidad (Wijaya et al., 2021).

2.4.2 Aplicación Backend

Elección de Tecnologías y Frameworks

Para el desarrollo de esta aplicación se utilizaron diferentes tecnologías que tienen una gran capacidad para poder abordar todas las necesidades que tiene el proyecto. El lenguaje de programación que se utilizó para el desarrollo de la aplicación fue Typescript ya que este lenguaje nos da un tipado estático y tiene una integración directa con Javascript, Typescript mejora en gran medida la robustez del código en proyectos grandes y de igual forma hace que el mantenimiento sea mucho más fácil, por otra parte facilita la detección de errores al momento de desarrollar mejorando así la experiencia que tiene el programador (Dospinescu et al., 2017). De igual forma este tiene una gran compatibilidad con frameworks modernos.

Por otra parte el framework de backend que se esta usando dentro del proyecto es AdonisJS, y se seleccionó gracias a la capacidad que éste tiene para poder estructurar aplicaciones con patrones de diseño como MVC. AdonisJS está construido sobre Node.js y ofrece diferentes herramientas integradas como ORM o validadores, lo cual es una gran ventaja ya que reduce ampliamente el tiempo de desarrollo y nos ofrece un alto grado de seguri-

dad para poder implementarlo cómo una API RESTful (Molina Ríos & Pedreira-Souto, 2019). También su integración con Typescript permite que el código esté organizado.

Por último para la gestión de bases de datos se optó por MySQL ya que tiene un rendimiento bastante eficiente en aplicaciones transaccionales. De igual forma MySQL es muy famoso por su capacidad de manejar grandes volúmenes de datos con una gran eficiencia cuando se realizan consultas complejas, lo cual es sumamente esencial para la aplicación que se desarrollo (Truskowski et al., 2020). Además tiene un gran soporte y una gran comunidad capaz de resolver todos los problemas que puedan llegar a surgir al momento del desarrollo lo cual lo convierte en una opción bastante buena para proyectos que se van a lanzar a producción.

Diseño de la Arquitectura del Backend

La arquitectura que se utilizó en el backend se basa en un enfoque monolítico altamente modular, esta fue una elección estratégica ya que combina 2 aspectos bastante importantes los cuales son un despliegue único y además una estructura modular.(Molina Ríos & Pedreira-Souto, 2019). Esta arquitectura nos permite organizar en módulos muy bien definidos donde cada uno tiene una responsabilidad diferente, lo cual nos facilita la creación de nuevos módulos dentro de la aplicación sin impactar las funcionalidades del resto. Con este enfoque nos aseguramos que que existan muchos menos errores al momento de modificar o agregar nuevos componentes; esto es una gran ventaja en sistemas que requieren un escalabilidad progresiva a lo largo del tiempo.

El uso de esta arquitectura modular monolítica de igual forma nos ayuda a maximizar la reutilización del código y nos ayuda a tener muy bien definidas funciones individuales

dentro de nuestra aplicación. Cada módulo interactúa en interfaces separadas lo cual nos permite aislar de forma efectiva la lógica del negocio.

Autenticación y Autorización

La autenticación que se ha implementado en el sistema utiliza internamente JSON web tokens esta es una solución moderna y segura que nos da la capacidad de poder manejar diferentes sesiones dentro de la aplicación sin almacenar ningún dato sensible dentro del servidor, JWT nos permite autenticar a diferentes usuarios mediante la generación de tokens firmados digitalmente por el servidor de igual forma esto incluye diferentes datos necesarios para poder identificar al usuario y los permisos que éste tiene dentro de la aplicación. Al implementar este método no sólo nos aseguramos que las sesiones sean seguras si no también nos ayuda al rendimiento del sistema ya que elimina la necesidad de realizar consultas continuas al servidor para verificar las credenciales.

Por otra parte en cuanto a la autorización se implementó un control basado en roles, esto nos ayuda a gestionar los diferentes permisos que los usuarios tienen dentro de la aplicación de una manera mucho más granular. Al implementar este enfoque podemos definir roles específicos dentro de la aplicación como administrador, empleado u otros, asignando permisos delimitados a cada uno de estos. Esto hace que cada usuario acceda únicamente a los recursos y funcionalidades dentro de la aplicación para los que fue contratado. De igual forma la modularidad que existe dentro del sistema nos ayuda a gestionar estos permisos permitiendo agregar o modificar diferentes roles sin tener que afectar al resto de la aplicación esto nos ayuda a garantizar la protección de datos y el cumplimiento de los principios de mínima exposición y privilegios (Dospinescu et al., 2017).

Gestión de las APIs

Dentro de nuestra aplicación el diseño de la API se basa en los principios REST para que de esta forma se pueda garantizar una estructura clara y escalable. Las rutas de todo nuestro backend están organizadas en función de las necesidades que requiere la aplicación siguiendo las convenciones estándares como el uso de métodos HTTP para que de esta forma se puedan realizar diferentes operaciones sobre los recursos de nuestra aplicación. Cada uno de los endpoints se asocia a un módulo funcional dentro del sistema, lo cual permite una navegación e integración fluida con los diferentes módulos de la aplicación. Este enfoque nos ayuda a que cada parte de la API sea bastante fácil de mantener y ampliar, alineándose con el diseño monolítico modular de la arquitectura que se propuso. Por otra parte en términos de seguridad se implementaron diferentes medidas para poder proteger a nuestra API contra ataques comunes. Cada entrada de nuestro endpoint posee una validación asegurando que los datos que se envían desde el lado de los clientes cumplan con los formatos esperados y que no incluyen contenido malicioso que podría llegar a afectar a nuestros sistemas. Además se están utilizando JWT para poder autenticar y autorizar las diferentes solicitudes que ingresan a nuestra aplicación garantizando que sólo los usuarios válidos puedan acceder a estos recursos. De igual forma para prevenir ataques de falsificación de solicitudes entre diferentes sitios incluimos tokens específicos en las solicitudes sensibles aplicando las políticas de CORS.

Integraciones de Terceros

Dentro de nuestro sistema se integraron diferentes servicios externos como API de correo electrónico, para que se puedan gestionar y enviar correos dentro de nuestra aplicación de manera eficiente y segura. Esta integración nos permiten simplificar la gestión y además es una forma confiable de que los usuarios reciban todas las notificaciones de manera oportuna. Además el diseño modular de nuestra aplicación nos permite integrar eso de

manera sencilla sin afectar al resto de módulos.

Para manejar errores en las integraciones con servicios externos, se implementan estrategias de manejo de excepciones y validaciones exhaustivas. Cada solicitud a un servicio externo está encapsulada en bloques de prueba que capturan y registran errores para análisis posterior. Se definen mecanismos de reintentos automáticos con límites establecidos para garantizar la entrega en caso de fallas temporales. Adicionalmente, se emplean alertas para notificar a los administradores sobre problemas críticos, lo que minimiza el tiempo de respuesta ante fallos (Ehsan et al., 2022). Este enfoque asegura la continuidad operativa y protege la experiencia del usuario frente a interrupciones en los servicios externos.

2.4.3 Base de datos

MySQL es uno de las bases de datos relacionales más utilizadas a nivel mundial debido a su excelente rendimiento simples y robustez, este sistema de base de datos destaca ampliamente con diversas aplicaciones y tiene la capacidad de manejar altos volúmenes de datos transaccionales con una gran rapidez. MySQL generalmente se utiliza en sistemas que requieren operaciones constantes como aplicaciones de correo electrónico o sistemas de gestión empresarial. Según Grover y Johari (2015), MySQL es altamente efectivo en diferentes tareas que están relacionadas con operaciones de inserción eliminación o consultas con un gran nivel de complejidad lo cual lo convierte en una opción bastante sólida para aplicaciones en donde la integridad de los datos es bastante crítico.

Por otra parte otra de las ventajas más importantes de MySQL es su alta compatibilidad con diferentes entornos virtualizados como lo son los contenedores de Docker ya que estos facilitan la escalabilidad del despliegue en diferentes infraestructuras sobre todo en

ambientes Cloud como lo pueden ser AWS o Azure. Estudios como el de Truskowski et al. (2020) han demostrado que esta base de datos tiene un buen desempeño en entornos virtuales usados que requieren un alto rendimiento en consultas SQL estándar estas características al igual que su amplia comunidad de soporte hacen que esta opción sea una solución sumamente confiable y eficiente en sistemas como el que se desarrolló

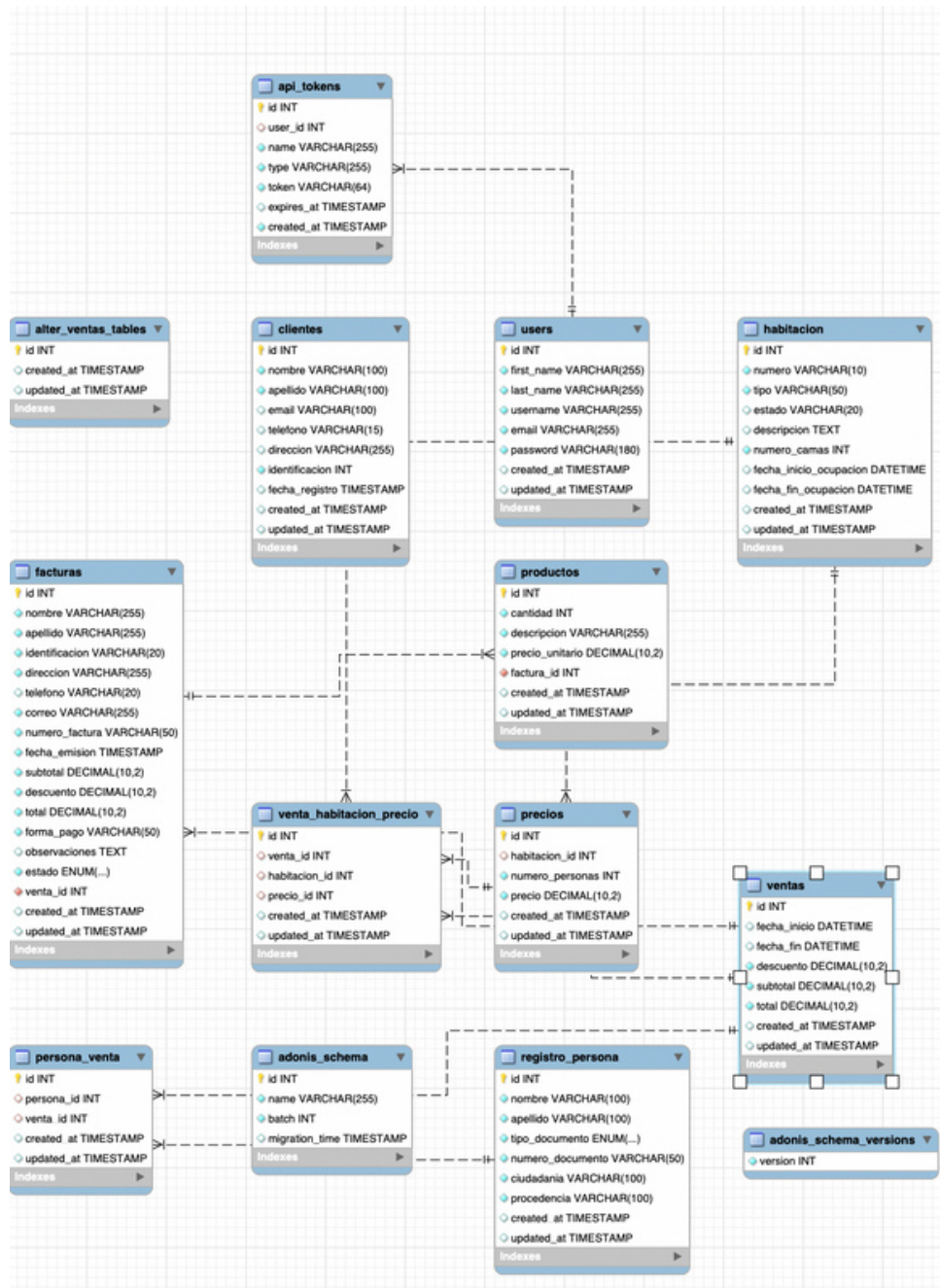


Figure 2.8: La figura muestra el diagrama de la base de datos

2.4.4 Despliegue en la nube

Para que nuestra aplicación sea visible y esta expuesta a internet se utilizaron los servicios de Cloud, para el apartado del frontend como nuestra aplicación está creada en NextJS se decidió utilizar Vercel ya que ofrece el servicio de hosting gratuito para este tipo de aplicaciones. En cuanto al dominio se lo compró en esa misma plataforma ya que los costos eran relativamente bajos frente al resto de competidores, el dominio que se compró es bethalabs.com y lo que se planea hacer es agregar subdominios para poder hacer el hosting de esta aplicación en específico. Por lo tanto la aplicación esta hosteada en hote-lapp.bethalabs.com de esta forma la aplicación se ve mucho más profesional a diferencia de hostearla con un dominio que se asigna automáticamente por la propia plataforma.

Ahora para el apartado backend y de base de datos se decidió AWS la razón es porque anteriormente ya se tienen conocimientos previos sobre el uso de esta plataforma y cómo desplegar aplicaciones en contenedores, Lo que se utilizó para la base de datos fue un contenedor EC2. Dentro de este contenedor se descargó Docker y posteriormente MySQL empezó a correr dentro del mismo, para que este contenedor pueda ser accedido por nuestro backend se abrió el puerto 3306 el cual es el puerto por donde corre MySQL.

En cuanto al backend de igual forma se lo decidió alojar en AWS. Ya que se tiene conocimientos sobre esta plataforma y además es mucho más sencillo conectar la base de datos al backend trabajando en una misma plataforma, de igual forma se creó una instancia de EC2. Para poder alojar la aplicación, además se utilizó la herramienta NGINX para poder aplicar un certificado SSL por donde puede correr la información de forma encriptada.

Conclusiones

La aplicación desarrollada para el control y administración hotelera nos demostró que es una herramienta bastante eficaz al momento de automatizar diferentes procesos internos, como la gestión de reservas, habitaciones, usuarios y facturación. Esto nos ayuda en gran medida a poder minimizar los tiempos de respuestas y además a reducir errores, de igual forma optimiza la experiencia tanto para el cliente como para el personal del hotel. Además el diseño modular implementado en la aplicación nos ayuda que esta tenga una navegación sencilla, asegurándonos que el sistema sea escalable y adaptable a diferentes necesidades que se puedan presentar en el futuro.

La integración de diferentes tecnologías como contenedores de Docker frameworks modernos tanto para frontend como para backend hace que la aplicación sea una solución que esta novedosa en el área de la gestión hotelera. De igual forma todo el enfoque que se le dio a la seguridad de los datos internos de la aplicación y a la autenticación robusta de usuarios, garantiza la fiabilidad de la información haciendo que los usuarios no tengan que preocuparse por brechas de seguridad. Esta aplicación con sus diferentes funcionalidades internas hace que los datos estén disponibles al instante reduciendo los problemas previos que se tenían dentro del establecimiento sobre disponibilidad de habitaciones y tiempos de espera haciendo que la reputación de hotel aumente en gran medida.

Por ultimo en términos académicos y profesionales este proyecto representa una contribución al demostrar como tecnologías emergentes pueden llegarse a aplicar a un sector altamente competitivo como lo es el hotelero. El diseño interno de la aplicación demuestra que es escalable y adaptable a hoteles de distintos tamaños abriendo diferentes posibilidades de implementación mas allá del prototipo inicial que se esta presentando. De igual

manera esta aplicación no solo responde a las demandas tecnológicas actuales si no que también podría llegar a ser un punto de partida para futuras innovaciones en cuanto a la gestión hotelera.

Trabajo futuro

Existen diferentes oportunidades para poder mejorar y expandir la aplicación de control hotelero. Una de la principales mejoras que se podría llegar a realizar es la integración de un modulo para la gestión de inventario para que de esa forma se pueda gestionar eficientemente los recursos del hotel, como por ejemplo productos de limpieza alimentos y bebidas entre muchos otros insumos que son necesarios para las operaciones del día a día.

Por otra parte también seria muy beneficioso establecer llamadas hacia la API del SRI para poder habilitar la facturación electrónica dentro de la aplicación. Esta funcionalidad permitirá a los usuarios de la aplicación emitir facturas para poder cumplir con las regulaciones fiscales que están vigentes dentro del país mejorando la transparencia y la profesionalidad del servicio.

Por ultimo se podría integrar un modulo de pagos en linea que facilite a los clientes de la aplicación para poder realizar pagos anticipados de las habitaciones. Esta implementación se la podría realizar en plataformas como PayPal o cualquier otro servicio de pagos electrónicos, mejorando así la comodidad del cliente y ademas reduciendo la carga administrativa del hotel automatizando el proceso de cobro.

Referencias bibliográficas

- [1] Almeida, M. G. de, & Canedo, E. D. (2022). Authentication and authorization in microservices architecture: A systematic literature review. *Applied Sciences*.
- [2] Bilgihan, A., Okumus, F., Nusair, K., & Kwun, D. (2011). Information technology applications and competitive advantage in hotel companies. *Journal of Hospitality and Tourism Technology*, 2(2), 139-153.
- [3] Caffery, L., Crew-Wegner, K., Reid, W., & Wootton, R. (2004). Automatic message handling for a national counselling service. *Journal of Telemedicine and Telecare*, 10(1), 18–21.
- [4] Dospinescu, O., Strîmbei, C., Strainu, R.-M., & Nistor, A. (2017). Adaptive UIX Layer for University Information SOA-BUS. *International Journal of Computers, Communications & Control*.
- [5] Ehsan, A., Abuhaliqa, M. A., Catal, C., & Mishra, D. (2022). RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions. *Applied Sciences*.
- [6] Firmansyah, B., Jonathan, M., Andreas, J., Philip, S., & Hidayaturrahman. (2023). Application of UI/UX in Tourism Information Service Problems: A Review. *Proceedings of the 8th International Conference on Sustainable Information Engineering and Technology*.
- [7] Flavian, C., Gurrea, R., & Orus, C. (2008). *The Relevance of Web Design for the Website Success: A heuristic analysis*. COLLECTeR, Iberoamerica.

- [8] Gamez-Diaz, A., Fernandez, P., & Ruiz-Cortes, A. (2017). An Analysis of RESTful APIs Offerings in the Industry. In *ICSOC 2017* (pp. 589–604). https://doi.org/10.1007/978-3-319-69035-3_43.
- [9] Guilding, C., Kennedy, D., & McManus, L. (2001). Extending the boundaries of customer accounting: Applications in the hotel industry. *Journal of Hospitality & Tourism Research*, 25(2), 173-194.
- [10] González, R., Gascó, J., & Llopis, J. (2019). ICTs in hotel management: a research review. *International Journal of Contemporary Hospitality Management*.
- [11] Grover, P., & Johari, R. (2015). MVM: MySQL Versus MongoDB. In Proceedings of the 2nd International Conference on Advances in Computing and Information Technology, 899–909.
- [12] Iskandar, T. F., Suseno, J. E., Suprayogi, M. H., & Kusuma, E. (2020). *Comparison between client-side and server-side rendering in the web development*. IOP Conf. Ser.: Mater. Sci. Eng. 801, 012136. Recuperado de <https://iopscience.iop.org/article/10.1088/1757-899X/801/1/012136/pdf>.
- [13] Jartarghar, H. A., Salanke, G. R., Kumar, A. R., Sharvani, G. S., & Dalali, S. (2022). *React apps with server-side rendering: Next.js*. Journal of Telecommunication, Electronic and Computer Engineering, 14(4), 25-29.
- [14] Junquera-Sánchez, J., Cilleruelo, C., de Marcos, L., & Martínez-Herráiz, J. J. (2021). Access control beyond authentication. *Security and Communication Networks*, 2021, 8146553.

- [15] Klein, R., Koch, S., Steinhardt, C., & Strauss, A. (2020). A review of revenue management: Recent generalizations and advances in industry applications. *European Journal of Operational Research*, 284(1), 397-412.
- [16] Lazuardy, M. F. S., & Anggraini, D. (2022). Modern Front-End Web Architectures with React.js and Next.js. *International Research Journal of Advanced Engineering and Science*, 7(1), 132-141.
- [17] Liu, R., & Li, Y. (2023). Implementation and research based on WLAN intelligent access control system architecture. *SPIE Proceedings*, 12941, 129415F-1–129415F-6.
- [18] Molina Ríos, J., & Pedreira-Souto, N. (2019). Approach of Agile Methodologies in the Development of Web-Based Software. *Inf.*, 10(10), 314.
- [19] Mustafa, S. Z. S., & Adruce, S. A. Z. (2017). A Critical Review of Research on Websites Development and Design Related to Diversity Issues. Academia.edu.
- [20] Neumann, A., Laranjeiro, N., & Bernardino, J. (2018). An Analysis of Public REST Web Service APIs. *IEEE Transactions on Services Computing*. DOI: 10.1109/TSC.2018.2847344.
- [21] Truskowski, W., Klewek, R., & Skublewska-Paszkowska, M. (2020). Comparison of MySQL, MSSQL, PostgreSQL, Oracle databases performance, including virtualization. *Journal of Computer Sciences Institute*, 16, 279–284.
- [22] Wijaya, A., Kefry, Wihalim, W., & Gunawan, A. A. S. (2021). The Effect of UI/UX Design on User Satisfaction in Online Art Gallery. *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*, 1, 120–125.

- [23] Yong-ye, J. (2009). A review and some suggestions on hotel knowledge management. *Journal of Wuhan Institute of Shipbuilding Technology*.
- [24] Zhao, G., Chadwick, D., & Otenko, S. (2007). Obligations for role-based access control. *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, 1, 424–431.

ANEXO A: URL's del proyecto

- **Link al proyecto Frontend:** <https://github.com/elrichi31/hotel-app-frontend>
- **Link al proyecto Backend:** <https://github.com/elrichi31/hotel-app-backend>
- **Link a la página web desplegada:** <https://hotelapp.bethalabs.com/>

ANEXO B: Capturas de pantalla del sistema

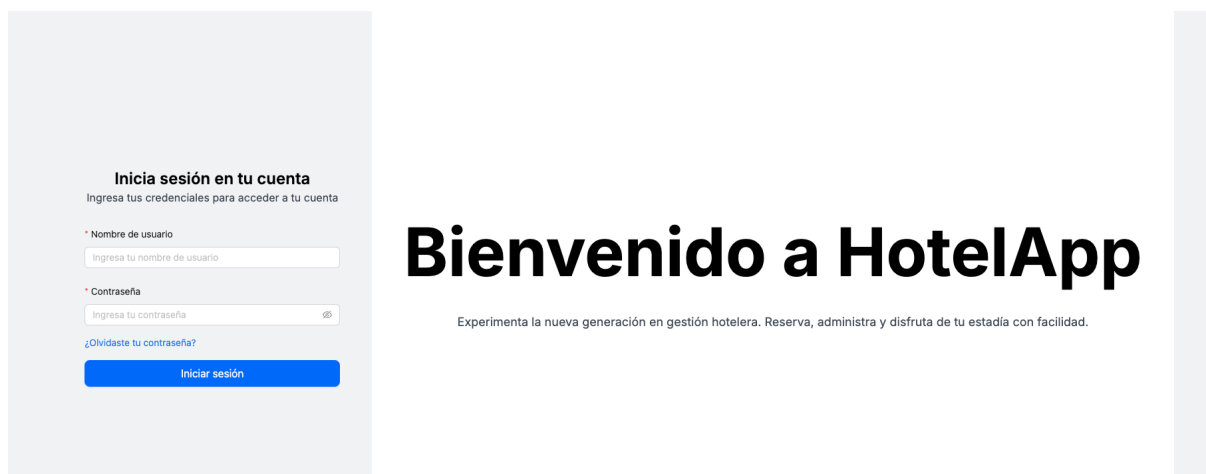


Figure 3.1: Inicio de sesión del sistema.

admin admin

Dashboard

Nueva Venta

Habitaciones

Ventas

Facturas

Reservas

Usuarios

Detalles de las Ventas

Todas Start date End date Aplicar Filtro

Buscar...

ID	Cliente	Fecha de Inicio	Fecha de Fin	# Facturas	Subtotal	Descuento	Total	Acciones
7	Nicolas Molina	09/12/2024 03:37	12/12/2024 10:00	0	69.00	0.00	69.00	Editar Facturas X
6	Nicolas Molina	09/12/2024 03:27	14/12/2024 10:00	2	60.00	0.00	60.00	Editar Facturas X
5	Juan Aldaz	08/12/2024 13:03	10/12/2024 05:00	0	50.00	0.00	50.00	Editar Facturas X
4	Nicolas Molina	08/12/2024 05:06	11/12/2024 00:00	2	36.00	0.00	36.00	Editar Facturas X
3	David Lopez	07/12/2024 22:22	09/12/2024 19:00	1	30.00	0.00	30.00	Editar Facturas X

< 1 >

Figure 3.2: Página de ventas.

admin admin

Dashboard

Nueva Venta

Habitaciones

Ventas

Facturas

Reservas

Usuarios

Lista de Usuarios

Buscar usuarios...

+ Crear Usuario

ID	Nombre	Apellido	Nombre de Usuario	Correo Electrónico	Rol	Estado	Acciones
1	admin	admin	admin	admin@example.com	admin	Activo	Editar Eliminar
5	Nicolas	Molina	n Molina	nicolawsmolina@gmail.com	empleado	Activo	Editar Eliminar

< 1 >

Figure 3.3: Página de usuarios.