

## Índice

1. [Resumen del proyecto](#)
2. [Visión y objetivos del proyecto](#)
3. [Integrantes del equipo](#)
  - a. [Tomas Alberdi - Desarrollador FrontEnd](#)
  - b. [Tomas Dopaso - Analista de Infraestructura](#)
  - c. [Santiago Ortigoza - Analista de control de calidad/ QA](#)
  - d. [Jonatan Bordon - Desarrollador BackEnd](#)
  - e. [Silvestre Novaro - Analista de Base de Datos](#)
4. [Cómo Scrum ayudó a nuestro equipo a desarrollar el proyecto](#)
5. [Roles del equipo y responsabilidades en el proyecto](#)
6. [Acuerdos del equipo para trabajar de manera efectiva](#)
7. [Bitácora y retrospectiva del equipo](#)
  - a. [Experiencia en el desarrollo de los sprints y abordaje de los requerimientos en el proyecto](#)
  - b. [Fortalezas del equipo y éxito en la entrega de un módulo crítico del proyecto](#)
  - c. [Aprendizajes del equipo en situaciones difíciles y trazando el camino hacia la mejora continua](#)
8. [Tecnologías utilizadas para la gestión del proyecto](#)
9. [Tecnologías utilizadas para mejorar la productividad en el desarrollo del proyecto](#)
10. [Tecnologías y herramientas utilizadas en el desarrollo del sitio](#)
11. [Cómo configurar y ejecutar el ambiente de desarrollo del proyecto](#)
  - a. [Instrucciones para el desarrollador BackEnd](#)
  - b. [Instrucciones para el Analista de datos](#)
  - c. [Instrucciones para el Desarrollador FrontEnd](#)
  - d. [Instrucciones para el Analista de control de calidad/ QA](#)
  - e. [Instrucciones para el Analista de Infraestructura](#)
12. [Técnicas y acuerdos clave para el desarrollo del proyecto](#)
  - a. [Implementación de buenas prácticas en el control de versiones](#)
  - b. [Commits significativos](#)
  - c. [Mensajes claros de commit](#)
  - d. [Uso de ramas \(branches\)](#)
  - e. [Code reviews](#)
  - f. [Merge con cuidado](#)
13. [Conexión de las partes de la aplicación](#)
  - a. [Diagrama de infraestructura para la aplicación con MySQL, Java, React y LocalHost](#)
  - b. [Estructura y diagramas de la base de datos](#)
  - c. [Scripts / instrucciones para la creación de la base de datos](#)
14. [Documentación de BackEnd y script de MySQL \(links\)](#)
15. [Documentación de Testing \(links\)](#)

### Resumen del proyecto:



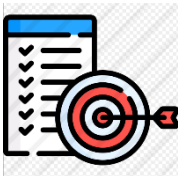
El proyecto de reservas de hoteles de alojamiento consiste en el desarrollo de un sistema de gestión de reservas de hoteles, departamentos y casas en línea. El objetivo principal del proyecto es proporcionar una plataforma de reservas en línea que sea fácil de usar para los clientes y que permita a los gerentes de hotel gestionar las reservas de manera eficiente.

El sistema se centrará en la facilidad de uso y la accesibilidad para el cliente, con una interfaz de usuario intuitiva y amigable. Los clientes podrán buscar y reservar alojamiento en línea de forma rápida y sencilla, lo que reducirá el tiempo de espera y mejorará la satisfacción del cliente.

Por otro lado, el sistema permitirá a los gerentes de hotel [Administrador] gestionar las reservas de manera más eficiente, lo que reducirá los errores y mejorará la eficiencia en la gestión del hotel. Con esta plataforma, los gerentes de hotel podrán supervisar todas las reservas en tiempo real, realizar cambios y actualizaciones en las reservas y tener una mejor visibilidad de la ocupación del hotel.

En resumen, el proyecto de reservas de hoteles de alojamiento tiene como objetivo desarrollar un sistema de gestión de reservas en línea que proporcione una mejor experiencia de usuario para los clientes y una mayor eficiencia para los gerentes de hotel. Este sistema permitirá mejorar la atención al cliente y la eficiencia en la gestión del hotel.

### Visión y Objetivos del Proyecto:



El objetivo principal del equipo es desarrollar un sistema de reservas de hotel de alta calidad que proporcione a los clientes una experiencia de reserva de hotel fluida y sin problemas. El equipo está enfocado en crear una plataforma intuitiva y fácil de usar para los clientes, lo que permitirá la realización de reservas de alojamiento de manera rápida y sencilla.

Además, el equipo busca mejorar la eficiencia de la gestión del hotel mediante la automatización de procesos y la integración de herramientas de análisis de datos. El sistema permitirá a los gerentes de hotel supervisar todas las reservas en tiempo real, realizar cambios y actualizaciones en las reservas, y tener una mejor visibilidad de la ocupación del hotel. Todo esto mejorará la eficiencia y la efectividad en la gestión del hotel.

Otro objetivo importante del proyecto es aumentar la tasa de ocupación de los hoteles, departamentos y casas que ofrecen los servicios de DigitalBooking. Al proporcionar una plataforma de reservas en línea fácil de usar, los clientes tendrán acceso a una amplia variedad de

opciones de alojamiento, lo que aumentará las posibilidades de que eligen reservar con DigitalBooking. Esto, a su vez, ayudará a aumentar la tasa de ocupación de los hoteles, departamentos y casas asociados.

Por último, el equipo busca mejorar la satisfacción del cliente al proporcionar una experiencia de reserva sin problemas y mejorar la calidad del servicio en general. Se espera que el sistema de reservas de hotel de DigitalBooking proporcione una experiencia de usuario satisfactoria y aumente la satisfacción general del cliente.

En resumen, el objetivo principal del grupo al realizar el proyecto de reservas de hoteles de alojamiento es proporcionar una solución completa y efectiva para la gestión de reservas de hoteles. Se espera que el sistema de reservas proporcione una experiencia de usuario satisfactoria, mejore la eficiencia en la gestión del hotel, aumente la tasa de ocupación de los hoteles, departamentos y casas asociados y, en general, mejore la calidad del servicio ofrecido por DigitalBooking.

### Integrantes del Equipo:



- ✚ Tomas Alberdi - Desarrollador FrontEnd
- ✚ Tomas Dopaso - Analista de Infraestructura
- ✚ Santiago Ortigoza - Analista de control de calidad/ QA
- ✚ Jonatan Bordon - Desarrollador BackEnd
- ✚ Silvestre Novaro - Analista de Base de Datos



Tomas Alberdi - Desarrollador FrontEnd:

Desarrollador FrontEnd con gran interés por la creación de interfaces de usuario intuitivas y atractivas. Poseo experiencia en el uso de React, JavaScript, CSS y HTML para crear aplicaciones web interactivas y dinámicas. Con deseos de perfeccionar mi habilidad para crear diseños de interfaz de usuario atractivos y funcionales que ofrecen una excelente experiencia de usuario. Me considero una persona detallista y perfeccionista, lo que me permite trabajar en los pequeños detalles que hacen una gran diferencia en la experiencia del usuario. Además, me gusta mantener una comunicación abierta y constante con el resto del equipo para asegurarnos de estar alineados en los objetivos del proyecto.



Tomas Dopaso - Analista de Infraestructura:

Analista de infraestructura con habilidades en la implementación y mantenimiento de infraestructuras de sistemas con la intención de que sean de alta disponibilidad y escalables. Poseo experiencia en el uso de herramientas de virtualización y en la implementación de soluciones para garantizar la continuidad del negocio. Además, estoy en proceso de formación en la creación de entornos de desarrollo y prueba. Mi habilidad principal es la capacidad de organización y planificación. Siempre estoy pensando en cómo podemos mejorar nuestra

infraestructura para hacerla más eficiente y escalable. También me destaco por mi capacidad para resolver problemas y encontrar soluciones creativas.



Santiago Ortigoza - Analista de control de calidad/ QA:

Analista de control de calidad/ QA con habilidades en el desarrollo y ejecución de planes de pruebas. Experiencia en la creación de casos de prueba y en la ejecución de pruebas manuales y automatizadas. Además, descubrí mi habilidad para identificar y documentar problemas de calidad y para trabajar en estrecha colaboración con el equipo de desarrollo para garantizar la calidad de los productos entregados. Me gusta asegurarme de que cada aspecto del sistema esté funcionando correctamente y que la experiencia del usuario sea óptima. Además, me destaco por mi capacidad para trabajar en equipo y colaborar con el resto del equipo para solucionar cualquier problema que pueda surgir.



Jonatan Bordon - Desarrollador BackEnd:

Desarrollador BackEnd con habilidades en el desarrollo de aplicaciones web escalables y en proceso de aprendizaje para buscar que sean de alta calidad. Experiencia en el uso de Java Spring Boot para crear aplicaciones web eficientes y escalables y en el desarrollo de APIs y en la integración de servicios web de terceros. Mi habilidad principal es la resolución de problemas. Me gusta desafiarme a mí mismo y encontrar soluciones creativas y eficientes para los problemas que surgen en el desarrollo. También me destaco por mi capacidad para trabajar en equipo y colaborar en la definición de los requerimientos del proyecto.



Silvestre Novaro - Analista de Base de Datos:

Analista de base de datos con habilidades en la gestión de bases de datos y en el diseño de estructuras de datos escalables. Experiencia en el uso de MySQL y en la creación de consultas eficientes y optimizadas para obtener y manipular datos. Además, en la creación de estructuras de datos complejas y en la gestión de grandes volúmenes de datos. Mi habilidad principal es la capacidad de analizar datos y encontrar patrones y tendencias. Me gusta trabajar en la optimización de bases de datos para asegurarme de que el sistema funcione de manera eficiente. También me destaco por mi capacidad para comunicar información de manera clara y efectiva al resto del equipo.

En resumen, cada uno de los integrantes de nuestro equipo tiene habilidades y experiencia valiosas que contribuyen al éxito del proyecto DigitalBooking. Desde el desarrollo FrontEnd hasta la gestión de la base de datos, cada miembro del equipo aporta habilidades y conocimientos específicos que se complementan para crear una solución completa y efectiva para la gestión de reservas de hoteles.

### Cómo Scrum Ayudó a Nuestro Equipo a Desarrollar el Proyecto



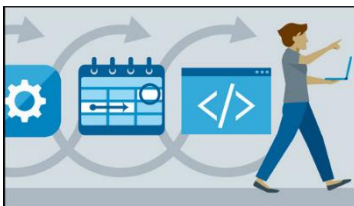
Como equipo destacamos que la metodología Scrum ha fomentado la comunicación constante y el trabajo en equipo dentro del grupo. Las reuniones regulares de Scrum nos permiten mantenernos actualizados sobre los avances y los obstáculos, y también nos da la oportunidad de colaborar en la resolución de problemas. Esta comunicación constante ha ayudado a mantener una mentalidad ágil y adaptable a lo largo del proyecto.

También hemos aprendido a ser más flexibles y a adaptarnos a los cambios a medida que surgen, lo que consideramos que es una habilidad importante en el mundo del desarrollo de software. Scrum nos ha enseñado a ser más autónomos en nuestra forma de trabajar y a tomar decisiones informadas sobre cómo priorizar nuestro trabajo en función de los objetivos del proyecto.

En cuanto a habilidades blandas, Scrum ha fomentado la colaboración, el liderazgo y la responsabilidad individual y colectiva. Como equipo, hemos aprendido a trabajar juntos hacia un objetivo común y a ayudarnos mutuamente a superar los obstáculos. También hemos desarrollado habilidades de liderazgo al asumir responsabilidades específicas dentro del equipo y al tomar decisiones informadas sobre cómo priorizar nuestro trabajo.

En resumen, Scrum ha sido una metodología de trabajo eficaz que nos ha ayudado a trabajar de manera más enfocada y colaborativa. Aunque hay desafíos y puntos negativos a considerar, hemos logrado superarlos y hemos desarrollado habilidades importantes en el proceso.

### Roles del equipo y responsabilidades en el proyecto:



Nuestro equipo de desarrollo se reunió para discutir cómo podríamos utilizar nuestras fortalezas y habilidades para garantizar el éxito del proyecto. En base a esas conversaciones, se asignaron roles específicos, como desarrollador frontend, desarrollador backend, analista de bases de datos, analista de infraestructura y Analista de control de calidad/ QA o testing analyst.

Una vez asignados los roles, discutimos las tareas necesarias para completar el proyecto y las distribuimos entre los miembros del equipo según sus habilidades y experiencia. Utilizamos un software de gestión de proyectos llamado GitLab para crear tareas y asignarlas a diferentes miembros del equipo, lo que nos permitió realizar un seguimiento del progreso y asegurarnos de que no hubiera solapamiento en las tareas asignadas.

Además, utilizamos una metodología ágil llamada Scrum, en la que trabajamos en sprints de corta duración para entregar un conjunto de funcionalidades acordadas al principio del sprint. Al final de cada sprint, evaluamos lo logrado y planificamos el siguiente sprint.

En general, nos sentimos satisfechos con la asignación de roles y tareas dentro del equipo. Esto nos permitió aprovechar al máximo las habilidades y fortalezas de cada miembro para garantizar el éxito del proyecto. Utilizar un software de gestión de proyectos también fue muy útil para trabajar de manera eficiente y asegurarnos de que todos estuvieran al tanto de sus responsabilidades.



#### **Nuestros acuerdos para trabajar de manera efectiva como equipo:**

Como equipo, hemos acordado algunas prácticas adicionales para trabajar de manera más efectiva.

Para la gestión del control de versiones, hemos elegido utilizar GitLab y GitHub y establecimos un flujo de trabajo para trabajar en ramas separadas y fusionarlas en la rama principal del repositorio. También hemos acordado seguir unas convenciones de codificación para normalizar el código y garantizar que sea legible y coherente. Además, escribimos pruebas de unidad para cada funcionalidad y revisamos el código diariamente para asegurarnos de que cumpla con los estándares acordados y sea de alta calidad.

Estos acuerdos adicionales nos han permitido trabajar de manera más eficiente y coherente, asegurando que el código producido sea de alta calidad y que todos los procesos de desarrollo sean consistentes.

Nuestro enfoque iterativo y colaborativo para cumplir con los requerimientos del proyecto

#### **Bitácora y Retrospectiva del Equipo:**



##### **"Experiencia en el desarrollo de los sprints y abordaje de los requerimientos en el proyecto"**

Durante el proyecto, como equipo de desarrollo, cada sprint fue un reto que nos permitió mejorar nuestro trabajo en equipo. Cada sprint comenzó con una planificación de objetivos y tareas a completar.

Durante las dailies, aprovechamos la oportunidad de compartir nuestro progreso y abordar cualquier problema que surgiera en el camino.

Uno de los mayores desafíos que enfrentamos fue abordar los requerimientos obligatorios y opcionales de manera eficiente, por lo que trabajamos juntos para establecer prioridades y dividir el trabajo según la capacidad del equipo.

Mantuvimos un enfoque iterativo en el desarrollo, lo que nos permitió ajustar y cambiar nuestro enfoque según las necesidades y requisitos emergentes.

En las retrospectivas, reflexionamos sobre nuestro desempeño en el sprint y encontramos maneras de mejorar en el futuro, considerando las retroalimentaciones del ProductOwner y nuestras experiencias personales.

En resumen, cada sprint fue una oportunidad para mejorar nuestro trabajo y enfoque en el cumplimiento de los objetivos y requerimientos del proyecto, y como equipo, logramos superar los desafíos y abordar todos los requerimientos de manera efectiva.



#### **“Destacando las fortalezas del equipo y su éxito en la entrega de un módulo crítico del proyecto.”**

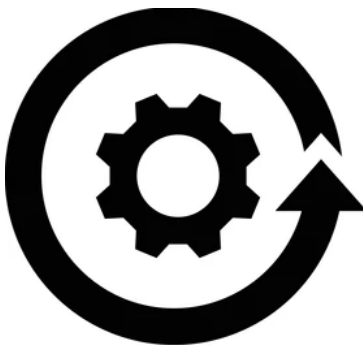
Como equipo de desarrollo, consideramos que una de las principales fortalezas que hemos tenido ha sido la comunicación efectiva.

Desde el comienzo del proyecto, establecimos una comunicación abierta y constante para asegurarnos de que todos estuviéramos en la misma página y para abordar cualquier problema o desafío que surgiera en el camino.

Esto para nosotros fue de suma importancia ya que trabajamos en diferentes áreas del proyecto y en diferentes ubicaciones geográficas [Argentina, Colombia].

Otra fortaleza que logramos como equipo es la colaboración, dispuestos a ayudar y trabajar juntos para lograr los objetivos del proyecto. Siempre con una actitud positiva y enfocados en soluciones, esto nos ayudó a lograr la mayor eficiencia en el desarrollo y a la resolución rápida de problemas.

Coincidimos que, uno de los mayores éxitos del equipo ha sido la entrega de un módulo crítico del proyecto cumpliendo los plazos establecidos, intentando siempre buscar una calidad óptima. Aprovechamos nuestras fortalezas en comunicación y colaboración para trabajar de manera eficiente y efectiva en el módulo, abordando cualquier problema de manera oportuna y manteniéndonos enfocados en los objetivos del proyecto.



#### **"Trazando el camino hacia la mejora continua: Aprendizajes del equipo en situaciones difíciles"**

Como equipo en nuestro primer proyecto, hemos enfrentado varias debilidades y situaciones difíciles. Uno de los mayores desafíos fue la falta de experiencia y conocimientos técnicos en ciertas áreas. Esto resultó en errores, pero no así en retrasos en el desarrollo, lo que no afectó la calidad y el tiempo de entrega siendo para nosotros muy importante respecto a las funcionalidades solicitadas en cada sprint.

Para abordar esta situación, nos aseguramos de trabajar en equipo y compartir nuestros conocimientos y habilidades para cubrir las brechas de conocimiento. También nos aseguramos de buscar ayuda externa [compañeros de la camada] y consultar con otros desarrolladores más experimentados como los TL cuando nos encontramos con problemas técnicos.

En un momento específico, nos encontramos con un problema de integración, en lugar de enfocarnos en culpar a alguien por el problema, nos pusimos a trabajar juntos para encontrar una solución. Hicimos una lluvia de ideas sobre posibles soluciones, realizamos pruebas y probamos varias opciones antes de encontrar la solución adecuada. Aprendimos que trabajar juntos en lugar de señalar con el dedo a alguien fue la mejor manera de abordar el problema.

En general, aprendimos que trabajar juntos como un equipo y comunicarnos claramente son claves para superar nuestras debilidades y resolver situaciones difíciles en un proyecto. Es importante aprender de nuestros errores y tomar las medidas necesarias para mejorar continuamente como equipo y como individuos.

#### **Tecnologías utilizadas para la gestión del proyecto.**



En nuestro primer proyecto como desarrolladores, utilizamos varias herramientas para gestionar el proyecto de forma efectiva. Entre ellas se encuentran GitLab-Boards, GitHub, GitLab y Google Drive.

GitLab-Boards fue fundamental para crear y administrar nuestro tablero de proyecto, en el cual definimos todas las tareas necesarias y las organizamos según su estado. Con esto, pudimos mantener una visión clara de las actividades pendientes y avanzar de manera organizada.

Además, GitHub y GitLab fueron nuestras plataformas de control de versiones y colaboración, lo que nos permitió trabajar en equipo y mantener un registro completo de los cambios realizados. Cada miembro del equipo trabajó en su propia rama y realizó un pull request para revisar y fusionar el código en el repositorio principal.

Finalmente, Google Drive nos permitió compartir y colaborar en documentos importantes, como la documentación del proyecto y las especificaciones de requisitos.

En general, estas herramientas fueron accesibles y fáciles de usar, permitiéndonos trabajar de manera efectiva y organizada en el proyecto.

#### **"Tecnologías utilizadas para mejorar la productividad en el desarrollo del proyecto"**



Como equipo de desarrollo en nuestro primer proyecto juntos como, hemos estado utilizando diferentes herramientas y tecnologías para llevar a cabo el trabajo de manera efectiva.



En cuanto a las IDEs, hemos estado utilizando principalmente IntelliJ IDEA, popular para el desarrollo en Java con una gran cantidad de herramientas y funcionalidades para mejorar nuestra productividad en el desarrollo.



En cuanto a los sistemas de control de versiones, hemos estado utilizando Git junto con la plataforma de alojamiento de repositorios GitHub/GitLab.

Esto nos ha permitido colaborar de manera efectiva en el código y mantener un registro claro de los cambios realizados en el proyecto.

Además, GitLab nos ha permitido revisar y aprobar el código de los demás miembros del equipo antes de ser fusionado con el código principal.



En cuanto a la simulación y virtualización, utilizamos diferentes herramientas para probar nuestro código en diferentes entornos y simulaciones de diferentes sistemas operativos y entornos de desarrollo, lo que nos ha permitido probar y depurar el código en diferentes configuraciones sin tener que configurar físicamente cada entorno.

En cuanto a herramientas de virtualización tenemos conocimiento en VirtualBox para crear máquinas virtuales y simular diferentes entornos de hardware y software, pero no nos vimos en la necesidad de implementarlo.

En general, estas herramientas y tecnologías nos han ayudado mucho en el desarrollo del proyecto y nos han permitido colaborar de manera efectiva.

### **Tecnologías y herramientas utilizadas en el desarrollo del sitio**

Durante el desarrollo del sitio, se emplearon diversas tecnologías y herramientas. En el frontend, se utilizaron HTML5, CSS3, JavaScript, React, y las librerías de pruebas Jest y React Testing Library. En cuanto al backend, se utilizó Java con Spring Boot y REST API, y se integró la herramienta de pruebas unitarias JUnit. Además, se implementó Mock Service Worker para simular servicios de backend y probar el frontend de forma aislada.

Para la base de datos se utilizó MySQL, y para la infraestructura se implementó Amazon Web Services (AWS) para el despliegue de la aplicación.

En general, estas tecnologías y herramientas permitieron un desarrollo más eficiente y efectivo, y la implementación de pruebas unitarias y de integración contribuyó a la calidad del sitio desarrollado.

## Cómo configurar y ejecutar el ambiente de desarrollo del proyecto

En esta sección se proporcionarán instrucciones detalladas para que un nuevo integrante del equipo pueda configurar y ejecutar el ambiente de desarrollo del proyecto de manera exitosa.

### Instrucciones para el desarrollador BackEnd:

Descarga e instala la versión Community de IntelliJ IDEA (actualmente la versión 2022.2.3).

Descarga e instala MySQL (puede ser la versión comunitaria o cualquier otra versión que soporte la conexión JDBC).

Clona el repositorio del proyecto desde la plataforma de control de versiones que se esté utilizando (por ejemplo, GitLab, GitHub, etc.).

Abre IntelliJ IDEA e importa el proyecto desde el directorio donde se clonó el repositorio.

Configura las siguientes dependencias del proyecto en el archivo pom.xml :

org.springframework.boot:spring-boot-starter-data-jpa

org.springframework.boot:spring-boot-starter-validation

org.springframework.boot:spring-boot-starter-web

com.mysql:mysql-connector-j

org.projectlombok:lombok (opcional)

org.springframework.boot:spring-boot-starter-test (scope: test)

com.fasterxml.jackson.core:jackson-databind (version: 2.13.1)

org.springframework.boot:spring-boot-starter-mail

org.springframework.boot:spring-boot-starter-security

org.springframework.security:spring-security-crypto (version: 6.0.2)

org.springframework.security:spring-security-oauth2-resource-server (version: 6.0.2)

io.jsonwebtoken:jjwt-api (version: 0.11.5)

io.jsonwebtoken:jjwt-impl (version: 0.11.5, scope: runtime)

io.jsonwebtoken:jjwt-jackson (version: 0.11.5, scope: runtime)

org.springdoc:springdoc-openapi-starter-webmvc-ui (version: 2.0.0)

Configura las propiedades de conexión a la base de datos en el archivo `application.properties` según la siguiente especificación:

- Especifica la URL de la base de datos, que incluye el protocolo JDBC, el servidor y la base de datos a la que se conectará la aplicación.

```
spring.datasource.url=jdbc:mysql://digitalhouse.cluster-cnyqegjgetrv.us-east-2.rds.amazonaws.com/0223TDPROM1C2LAED0522FT_GRUPO7?useSSL=true&verifyServerCertificate=false
```

- Nombre de usuario y contraseña para acceder a la base de datos.



```
spring.datasource.username=0223TDPROM1C2LAED0522FT_GRUPO7
```

```
spring.datasource.password=hae9Eibu
```

- Establece el dialecto de Hibernate para MySQL y especifica que Hibernate debe actualizar automáticamente la estructura de la base de datos a medida que se realizan cambios en las entidades de la aplicación.

### Hibernate properties



```
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
```

```
spring.jpa.hibernate.ddl-auto=update
```

Configura cualquier otra propiedad del proyecto que sea necesaria para su correcto funcionamiento (por ejemplo, las credenciales para el envío de correos electrónicos).

Ejecuta el proyecto y verifica que se puedan realizar las operaciones básicas (como registrar un usuario y obtener un token de autenticación) mediante la interfaz Swagger.

Además de estas instrucciones, es importante que el nuevo integrante tenga un conocimiento básico de Java, Spring Boot, JPA, MySQL, Swagger, y de las herramientas de testing como JUnit

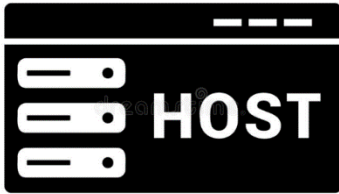


### Instrucciones para el Analista de datos:

Para armar tu ambiente de desarrollo en MySQL Workbench, deberás seguir los siguientes pasos:

Descarga e instala MySQL Workbench en tu equipo. Puedes descargarlo desde el sitio oficial de MySQL.

Configura una conexión a la base de datos. Para hacerlo, abre MySQL Workbench y selecciona la opción "New Connection" en la pantalla de bienvenida. En la ventana de configuración, completa la información requerida, incluyendo el host, puerto, nombre de usuario y contraseña. En este caso, deberás utilizar la información de conexión proporcionada en las propiedades de Backend:



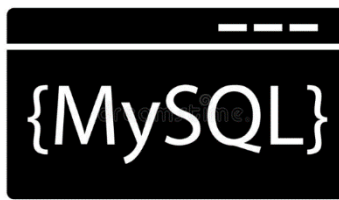
Host: digitalhouse.cluster-cnyqegjgetrv.us-east-2.rds.amazonaws.com

Puerto: 3306 (por defecto para MySQL)

Nombre de usuario: 0223TDPROM1C2LAED0522FT\_GRUPO7

Contraseña: hae9Eibu

Una vez que hayas configurado la conexión, podrás conectarte a la base de datos haciendo clic en el botón "Test Connection". Si la conexión es exitosa, haz clic en "OK" para guardar la configuración.



Una vez que hayas establecido la conexión, podrás acceder a la base de datos y realizar consultas utilizando MySQL Workbench. Es importante que tengas conocimientos básicos de SQL para poder comprender la estructura de la base de datos y realizar las consultas necesarias.

Además, es posible que necesites utilizar otras herramientas o lenguajes de programación para el análisis de datos, como Python o R. En ese caso, deberás asegurarte de tener las herramientas adecuadas instaladas en tu ambiente de desarrollo.

Con estos pasos, deberías estar listo para trabajar en el proyecto como Analista de datos utilizando MySQL Workbench.

#### Instrucciones para el Desarrollador FrontEnd:

Para poder trabajar en el proyecto necesitarás tener instalado Node.js, Visual Studio Code, Git y NPM en tu ambiente de desarrollo. Para instalar estas herramientas, debes seguir los siguientes pasos:



Node.js: Lo primero que necesitarás es tener instalado Node.js en tu sistema. Puedes descargar la versión más reciente desde el sitio web oficial de Node.js.



Visual Studio Code: Este es un editor de código gratuito y muy popular entre desarrolladores. Puedes descargarlo desde su sitio web oficial.



Git: Git es un sistema de control de versiones que utilizamos para colaborar en el proyecto. Si aún no lo tienes instalado, puedes descargarlo desde su sitio web oficial.

NPM: NPM es el gestor de paquetes de Node.js que utilizamos para instalar las dependencias del proyecto. Se instala automáticamente cuando instalas Node.js.

Una vez que hayas instalado estas herramientas, debes seguir estos pasos:

Clona el repositorio del proyecto desde Github en tu equipo utilizando Git.

Abre una terminal de línea de comandos y navega hasta el directorio del proyecto clonado.

Ejecuta el comando `npm install` para instalar todas las dependencias del proyecto.

Instala las librerías necesarias para el proyecto ejecutando los siguientes comandos en la terminal:

React Router Dom: Es la librería que utilizamos para manejar el enrutamiento en la aplicación. Para instalarla, ejecuta el siguiente comando en la terminal:

```
npm install react-router-dom
```

Fslightbox React: Es la librería que utilizamos para implementar la galería de fotos en la versión de escritorio. Para instalarla, ejecuta el siguiente comando en la terminal:

```
npm install fslightbox-react
```

Swiper: Es la librería que utilizamos para implementar el slider de fotos en las versiones móvil y tablet. Para instalarla, ejecuta el siguiente comando en la terminal:

```
npm install swiper
```

React DatePicker: Es la librería que utilizamos para implementar el seleccionador de fechas en el filtro y reserva. Para instalarla, ejecuta el siguiente comando en la terminal:

```
npm install react-datepicker
```

Reactjs Availability Calendar: Es la librería que utilizamos para implementar el calendario de disponibilidad de productos. Para instalarla, ejecuta el siguiente comando en la terminal:

```
npm install reactjs-availability-calendar
```

Una vez que se hayan instalado todas las dependencias y librerías, puedes ejecutar el proyecto utilizando el comando `npm run dev`.

Ahora puedes acceder al proyecto en tu navegador web en la dirección `http://localhost:3000`.

Buenas prácticas y acuerdos en el desarrollo de proyectos



**Instrucciones para el Analista de control de calidad/ QA:**

Instrucciones para configurar y ejecutar el ambiente de desarrollo para Analista de Control de Calidad / QA en React con Jest, React Testing Library y JUnit

Como Analista de Control de Calidad / QA en React, es importante que configures correctamente tu ambiente de desarrollo para poder realizar pruebas efectivas en el proyecto.

A continuación, se detallan las instrucciones para configurar y ejecutar el ambiente de desarrollo con las herramientas necesarias:



Instalar Node.js: Descarga e instala la última versión estable de Node.js desde el sitio web oficial.



Instalar un editor de código: Recomendamos utilizar Visual Studio Code como editor de código, ya que es gratuito y cuenta con una gran cantidad de extensiones útiles para el desarrollo en React.

Clonar el repositorio del proyecto: Utiliza Git para clonar el repositorio del proyecto en tu equipo.

Instalar las dependencias: Abre una terminal de línea de comandos y navega hasta el directorio del proyecto clonado. Ejecuta el comando "npm install" para instalar todas las dependencias del proyecto, incluyendo Jest, React Testing Library y JUnit.

Configurar las pruebas:



+



Crea una carpeta llamada "tests" en la raíz del proyecto y agrega los archivos de prueba con la extensión ".test.js" dentro de ella. Utiliza Jest y React Testing Library para escribir las pruebas unitarias y de integración necesarias.

Ejecutar las pruebas: Para ejecutar las pruebas, utiliza el comando "npm run test". Esto iniciará Jest y ejecutará todas las pruebas en la carpeta "tests".

Verificar los resultados: Una vez que las pruebas hayan finalizado, verifica los resultados en la terminal. Jest te informará si alguna prueba ha fallado y te proporcionará información detallada sobre el error.

Con estas instrucciones, podrás configurar y ejecutar correctamente el ambiente de desarrollo para realizar pruebas en React con Jest, React Testing Library y JUnit. Recuerda que es importante realizar pruebas de forma regular para asegurarte de que el proyecto funciona correctamente y cumple con los estándares de calidad necesarios.



#### Instrucciones para el Analista de Infraestructura:

Para implementar Amazon Web Services (AWS) para el despliegue de la aplicación, sigue los siguientes pasos:

Crea una cuenta de AWS si aún no la tienes. Puedes hacerlo en el sitio web de AWS.

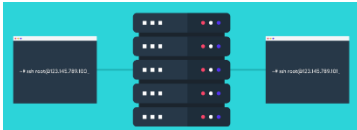
En el panel de AWS, crea una instancia EC2. Asegúrate de seleccionar la región adecuada para el despliegue.



Selecciona una imagen de máquina virtual (AMI) que sea compatible con tu aplicación. Puedes elegir entre varias opciones, como Amazon Linux, Ubuntu, etc.

Selecciona la instancia y crea una clave de acceso para conectarte a la instancia a través de SSH.

Configura las reglas de seguridad de la instancia. Asegúrate de permitir el tráfico entrante y saliente necesario para tu aplicación.



Conéctate a la instancia a través de SSH utilizando la clave de acceso que creaste anteriormente.

Instala las dependencias necesarias para la aplicación, como Node.js, npm, etc.

Clona el repositorio del proyecto en la instancia de EC2.

Configura la aplicación para el entorno de producción. Esto puede incluir la configuración de variables de entorno, la conexión a bases de datos, etc.

Ejecuta la aplicación en la instancia de EC2. Puedes hacerlo utilizando un servidor web como Nginx o Apache, o mediante el uso de un contenedor Docker.



Configura el balanceador de carga de AWS para distribuir el tráfico entre varias instancias de EC2 si es necesario.

Configura una base de datos en AWS, como Amazon RDS, si la aplicación lo requiere.

Configura la monitorización y los registros de la instancia de EC2 utilizando AWS CloudWatch.

Realiza pruebas en el entorno de producción para asegurarte de que todo funcione correctamente.

Con estos pasos, podrás implementar Amazon Web Services (AWS) para el despliegue de tu aplicación y asegurarte de que está funcionando correctamente en producción.

### Técnicas y acuerdos clave para el desarrollo de nuestro Proyecto

Las buenas prácticas, convenciones y acuerdos que nuestro equipo de desarrollo utilizó para asegurarnos que se desarrollara de manera efectiva y coherente son:



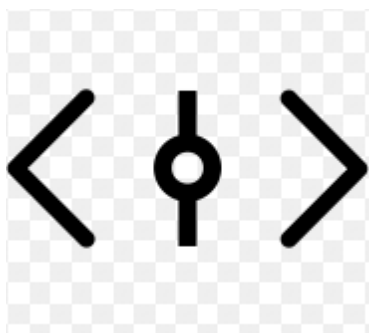
Como primer paso, establecimos un patrón de codificación coherente para mantener nuestro código limpio y fácil de entender. Esto incluyó convenciones para la nomenclatura de variables, indentación, uso de comentarios, entre otros aspectos.

Además, implementamos un sistema de control de versiones, en este caso utilizamos GitHub, para mantener un registro de los cambios realizados en el código y poder revertirlos en caso de ser necesario. También nos aseguramos de realizar pruebas de forma regular, tanto unitarias como de integración, para asegurarnos de que todas las funcionalidades funcionaran correctamente.

Por último, consideramos que una comunicación efectiva es crucial para el éxito del proyecto, por lo que establecimos reuniones regulares para discutir el progreso y cualquier problema que pudiera surgir. Además, documentamos todo el proceso en un wiki para asegurarnos de que todos los miembros del equipo estuvieran al tanto de la estructura del proyecto y cómo funcionaba. Esperamos que estas prácticas también puedan ser útiles para tu equipo de desarrollo.

### Implementando buenas prácticas en el control de versiones: Commits, merges y branches

Como equipo en nuestro primer proyecto, es fundamental establecer buenas prácticas en cuanto al control de versiones. A continuación, se presentan algunas recomendaciones que nos ayudarán a mantener un flujo de trabajo eficiente y efectivo:

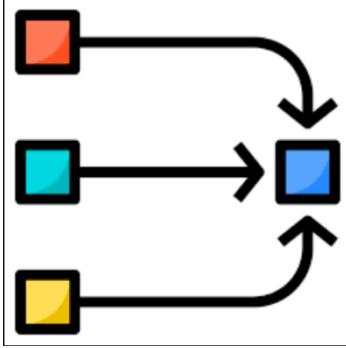


**Commits significativos:** Para rastrear los cambios de manera efectiva, es necesario que los commits sean significativos y se enfoquen en cambios específicos relacionados con una tarea o funcionalidad.

**Mensajes claros de commit:** Los mensajes de commit deben ser claros y concisos para que cualquier miembro del equipo pueda entender fácilmente el propósito del commit.

**Uso de ramas (branches):** El uso de ramas nos permitirá separar el trabajo en progreso del código que ya ha sido aprobado y se encuentra en producción. Es importante nombrar nuestras ramas de manera clara y concisa para que otros miembros del equipo puedan entender el propósito de cada una.





**Code reviews:** Antes de realizar un merge, es importante realizar revisiones de código (code reviews) para asegurarnos de que el código cumple con las mejores prácticas y sea de alta calidad. También nos permitirá detectar posibles errores o problemas antes de que lleguen a producción.

**Merge con cuidado:** Al realizar un merge, debemos asegurarnos de que todos los conflictos se hayan resuelto de manera efectiva y de que se hayan realizado las pruebas necesarias para confirmar que el código se integra correctamente.

Implementar estas buenas prácticas en el control de versiones nos ayudará a mantener un código de alta calidad, a colaborar efectivamente como equipo, a reducir los errores y a mantener una historia clara de los cambios realizados en el proyecto.

### La historia detrás de cómo nuestro equipo de desarrollo conectó las partes de nuestra aplicación

Como equipo de desarrollo detrás de nuestra aplicación DigitalBooking. Queríamos compartir la historia de cómo conectamos las partes de nuestra aplicación para brindar una experiencia coherente y eficiente para nuestros usuarios.



Primero, comenzamos con el front-end de nuestra aplicación. Diseñamos una interfaz de usuario atractiva y fácil de usar que permitiera a los usuarios interactuar con la aplicación de manera intuitiva. Sabíamos que esto era importante para que los usuarios disfrutaran usando nuestra aplicación.

A continuación, construimos el back-end de nuestra aplicación. Nos aseguramos de que la lógica del negocio, el almacenamiento y la manipulación de datos estuvieran perfectamente organizados. Utilizamos una API para manejar las solicitudes del front-end y enviar las respuestas adecuadas.

También nos aseguramos de que nuestra aplicación se comunicara con los servicios externos que necesitábamos, como la autenticación.

Luego, trabajamos en la base de datos de nuestra aplicación. Almacenamos los datos de manera eficiente y aseguramos que el back-end pudiera interactuar con la base de datos para leer, escribir y actualizar los datos según fuera necesario.



Finalmente, nos aseguramos de que nuestra aplicación estuviera disponible en línea y accesible para los usuarios. Utilizamos servidores para alojar nuestra aplicación y sus componentes, lo que nos permitió ofrecer una experiencia en línea sin interrupciones.

En resumen, conectamos las partes de nuestra aplicación utilizando protocolos de comunicación como HTTP, WebSocket y API. Fue un proceso de construcción paso a paso, pero gracias a la colaboración y el trabajo en equipo, logramos crear una aplicación sólida y coherente que nuestros usuarios pueden disfrutar.

### Diagrama de infraestructura para la aplicación con MySQL, Java, React y LocalHost

Basándonos en las tecnologías mencionadas, podríamos tener una estructura de infraestructura como la siguiente:



Servidor:

Máquina virtual en Amazon Web Services (AWS) con sistema operativo Linux

Dirección IP: 18.216.169.161

Servidor web Apache



Base de datos:

Servidor de base de datos MySQL en la misma máquina virtual de AWS

Puertos abiertos para permitir el acceso remoto desde aplicaciones externas

Esquema de base de datos diseñado y creado en MySQL

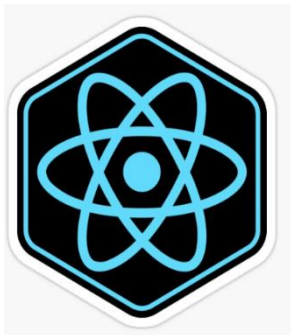


Back-end:

Aplicación Java que proporciona una interfaz de programación de aplicaciones (API) para el front-end

Se comunica con la base de datos para leer y escribir datos

Usa el servidor Apache para alojar la aplicación



Front-end:

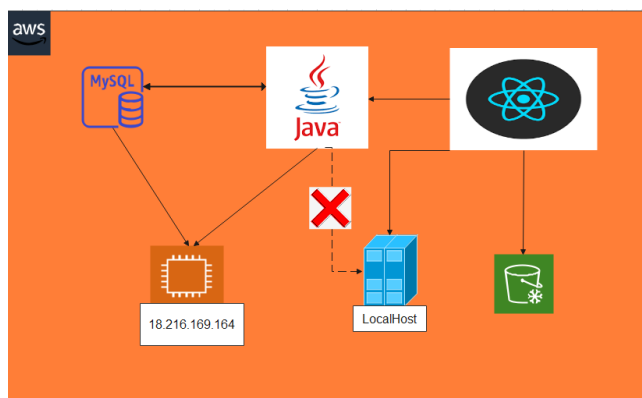
Aplicación React que se ejecuta en un navegador web

Se comunica con el back-end a través de la API para obtener y actualizar datos

Se aloja en el servidor Apache junto con el back-end

Para visualizar la estructura de la infraestructura, podríamos crear un diagrama de arquitectura como el siguiente:

Este diagrama muestra cómo los diferentes componentes de la infraestructura se conectan entre sí y cómo interactúan para proporcionar la funcionalidad de la aplicación.



La máquina virtual en AWS actúa como el servidor principal y aloja tanto el back-end como el front-end. El back-end se comunica con la base de datos para leer y escribir datos, mientras que el front-end se comunica con el back-end a través de la API para obtener y actualizar datos. Todo el software se aloja en el servidor web Apache para su entrega a los usuarios a través de un navegador web.

Este es solo un ejemplo de cómo podría ser la estructura de infraestructura para una aplicación con las tecnologías mencionadas. La infraestructura real puede variar según las necesidades de la aplicación y las preferencias del equipo de desarrollo.

## Estructura y diagramas de la base de datos

Nuestra aplicación utiliza MySQL como sistema de gestión de base de datos relacional. La estructura de la base de datos se compone de las siguientes tablas:

<b>Características</b>	<b>Políticas</b>	<b>Puntuacion</b>
<b>Categoria</b>	<b>Políticas_Descripcion</b>	<b>Reservas</b>
<b>Ciudades</b>	<b>Producto_Caracteristica</b>	<b>Roles</b>
<b>Imagen</b>	<b>Productos</b>	<b>Usuario</b>

## Scripts / instrucciones para la creación de la base de datos

Para crear la base de datos, necesitarás tener MySQL instalado en tu sistema y seguir los siguientes pasos:

Crea una nueva base de datos llamada " 0223TDPROM1C2LAED0522FT\_GRUP07 " en MySQL.

Script `create database 0223TDPROM1C2LAED0522FT_GRUP07;`

## Documentación de BackEnd y script de MySQL:

<https://humane-turnover-412.notion.site/Wikly-b66f8a7a24e540938dd703fa4c824892>

## Documentación de Testing:

[https://docs.google.com/document/d/1Bz03M7rtiiA93UwDMosmavOxHwx\\_K7ciVV1sA7C3-U/edit?usp=sharing](https://docs.google.com/document/d/1Bz03M7rtiiA93UwDMosmavOxHwx_K7ciVV1sA7C3-U/edit?usp=sharing)