

PROYECTO INTEGRADOR - GRUPO 6

¿Quiénes somos?

Somos un sitio web de reservas de alojamientos en el cual un usuario puede buscar y/o filtrar una serie de productos de acuerdo a ciertos criterios y necesidades, tales como rango de fechas y/o una ciudad, entre otros.

Nuestro objetivo

Buscamos ofrecer una plataforma intuitiva, funcional, accesible y visualmente agradable a aquellos usuarios que deseen encontrar y reservar un alojamiento de manera sencilla y versátil.

Índice

Nuestro equipo	4
Nuestras habilidades	6
Metodología de Trabajo	7
Bitácora del proyecto	8
Tecnologías utilizadas	10
Documentación JLH	12
Diagrama de redes	12
Diagrama BD	13
Documentación API	14
Documentación Testing	14

Nuestro equipo



Isabela Acosta

Tengo 21 años y soy de Bogotá, Colombia. Inicié esta carrera con conocimientos básicos en algoritmia y Python, algo que pude implementar durante este proyecto, en donde estuve a cargo de Backend y Base de Datos. Trabajé principalmente con lenguaje Java y SQL, y tecnologías como SpringBoot y Workbench.



Pilar Bettini

Tengo 23 años y soy de Córdoba, Argentina. Inicié esta carrera luego de mi licenciatura en Publicidad, por lo cual pude dar mi toque estético y redaccional al proyecto, en el cual me encargué de Front End y utilicé herramientas como Javascript y React.



Victoria Calabró

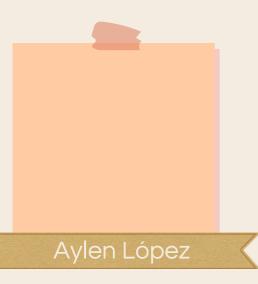
Tengo 19 años y soy de Buenos Aires, Argentina. Cuando finalicé el colegio secundario tenía mucha curiosidad por aprender programación, por eso arranqué esta carrera.



Tengo 20 años y soy de Buenos Aires, Argentina.

Empecé esta carrera con conocimientos basicos de html y javascript, los cuales se fueron desarrollando y afianzando a medida que cursaba.

Principalmente me centré en aportar al proyecto en el área de Infraestructura, desplegando los recursos en AWS, y en el área de Frontend, desarrollando nuestro sitio web en React.



Tengo 26 años y soy de Buenos Aires, Argentina. Actualmente estudio Diseño Web.

Antes de la carrera no tenía ningún conocimientos sobre programación, lo que sé es por lo aprendido en DH.

A lo largo del proyecto me desempeñé en el área de Front End. En general me encargué en la funcionalidad de los componentes.

Tengo 25 años, soy ingeniera electrónica bilingüe, de Barranquilla, Colombia. En el año 2020 realicé un

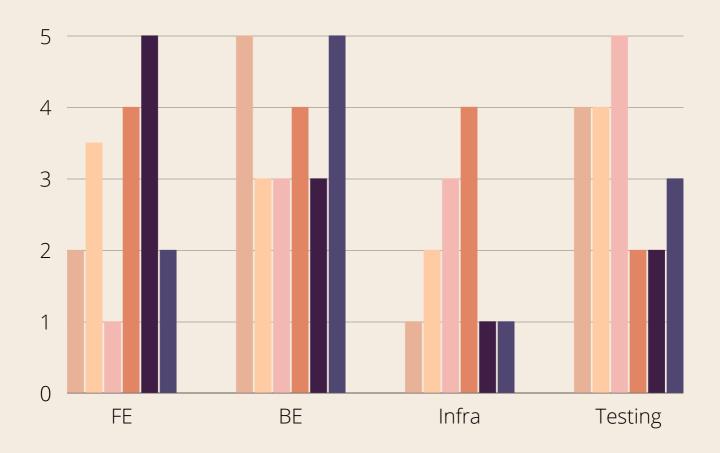
NodeJS y MongoDB, desde entonces supe que me quería especializar en Backend, gracias a Digital House tuve la oportunidad de desempeñarme en dicho rol durante este proyecto integrador, ahora puedo decir que manejo Java,

bootcamp enfocado en HTML, CSS, JavaScript, React,

Spring Boot, SQL, Git, entre otras tecnologías.

Gina Rodríguez

Nuestras habilidades





Metodología de trabajo

Durante todo el PI implementamos la metodología de trabajo Scrum. El proyecto estaba dividido en 4 Sprints, cada uno de estos tenia una duración de poco menos de dos semanas.

Diariamente nos reuniamos entre una hora y media y 3 horas, con el objetivo principal de comentar al resto del grupo nuestro progreso en las issues que teníamos asignadas de manera individual; adicional a esto, utilizábamos ese tiempo para encontrar respuestas como grupo a problemas que frenaban el progreso de algún area en especifico: FrontEnd, BackEnd/BD, Infraestructura o Testing.



La distribución de roles fue un proceso sencillo, ya que cada una de nosotras tenia afinidad en cosas distintas, de modo que no era incomodo para ninguna repetir un mismo rol durante los siguientes sprints.

A pesar de que cada una tuviera un rol y una serie de tareas afines asignadas, no fue impedimento para que conociéramos y entendiéramos el código en su totalidad en cada una de sus areas.

La distribución de tareas fue un proceso al que probablemente debimos invertirle algo mas de tiempo y análisis, sin embargo era una división equitativa y afín a los roles que cada una ocupaba. En el caso de que algún miembro terminara sus tareas asignadas antes de tiempo, generalmente ayudaba y/o tomaba algún issue asignado a otra persona, esto con el objetivo de alivianar la carga.

Entre los muchos aprendizajes que este proceso nos dejó como equipo, fue poder entender que para aprender hay que hacer desastres de vez en cuando. Es por esto que cada vez que alguien quería intentar algo nuevo sin afectar el trabajo de las demás, simplemente se creaba una rama, nombrada bajo la audaz persona que quería implementar este nuevo cambio.

Bitácora del proyecto

Primer Sprint

Como equipo nos fue sencillo organizarnos, definiendo eficazmente los roles dependiendo inicialmente de nuestras fortalezas.

Desde la primera semana pudimos decidir que queríamos que nuestro proyecto transmitiera: Elegancia, simpleza y sofisticación.

Teníamos a nuestro favor las ganas de aprender y la disposición de ayudarnos entre todas en cualquier momento. Sin embargo la falta de comunicación fue un impedimento grande para poder terminar todas las tareas asignadas.

De los aprendizajes del Sprint anterior, priorizamos un poco mas aquellas tareas obligatorias, concentrándonos mas en las funcionalidades, sin dejar de lado los estilos.

Sin embargo, a pesar de que cumplimos con casi todas las tareas asignadas, la falta de comunicación y la dificultad elevada de ciertas issues, hicieron que nuevamente tuvieramos Carry Over.

Tercer Sprint

Siendo este el sprint mas complejo, decidimos iniciar el desarrollo de código un poco antes que el resto de los equipos. Fue bastante desafiante, pues el nivel de dificultad de estas nuevas tareas era mucho mas elevado que en los anteriores Sprints.

Nuestras fortalezas permanecieron: la colaboración y disposición a ayudar fueron cosas de nunca cambiaron; sin embargo nuestro error mas grande el fue no priorizar correctamente.

5

Sprint Final

4

Para el cuarto y último sprint, ya teníamos completamente identificadas nuestras fortalezas y debilidades como equipo. Decidimos enfocarnos en las tareas que eran obligatorias para el correcto funcionamiento de nuestro proyecto, dejando en segundo plano funcionalidades que, a pesar de ser un plus, no eran esenciales.

Tecnologías utilizadas

Trabajamos las diferentes areas de Just Like Home en tres IDE's (Integrated Development Environment) principales:

- FrontEnd se desarrolló en Visual Studio Code principalmente en JavaScript y HTML.
- o BackEnd se desarrolló en IntelliJ IDEA, en lenguaje Java.
- En Base de Datos se hizo uso de MySQL Workbench, en lenguaje SQL.

Durante todo el proyecto hicimos uso de Git como sistema de control de versiones y GitLab como plataforma para gestionar el código e implementar la parte de DevOps/Infraestructura.

Cabe resaltar que la lógica y estructura de todo el proyecto está en inglés, permitiendo así que el código sea mucho mas accesible.

A continuación, se mostrarán las principales herramientas utilizadas en cada área:

Infraestructura

ENCARGADA: ZOE JIMENEZ

Herramientas

- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon Simple Storage Service (S3)
- Amazon Relational Database Service (RDS)
- CI/CD

Herramientas

- JEST
- Selenium
 Casos de Prueba
- Postman

Testing

ENCARGADA: VICKY CALABRÓ

Front End

Encargadas:

- Pilar Bettini
- Aylen López
- Zoe Jiménez

Herramientas

- React
- Bootstrap
- Material UI
- Moment.js
- React Router Dom
- Axios
- jwt-decode

Herramientas

- Mayen
- SpringBoot
- SpringBoot Security
- MySQL Workbench
- Postman

Back End / BD

Encargadas:

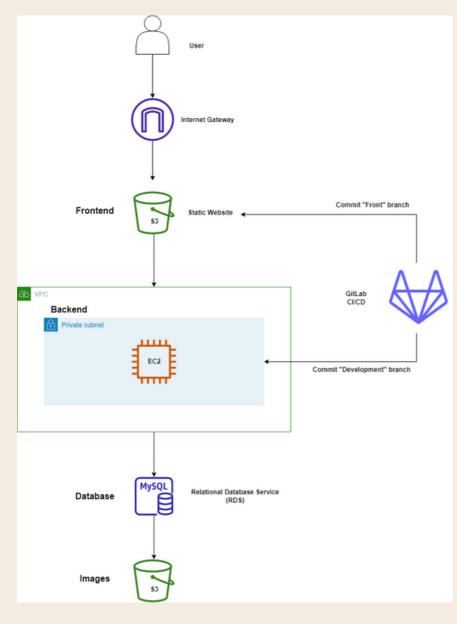
- Gina Rodríguez
- Isabela Acosta
- Zoe Jiménez

Documentación de JLH

Diagrama de redes

Nuestro proyecto hace uso de diferentes herramientas de AWS, y está desplegado en esta misma plataforma.

Nuestro frontend se encuentra alojado en un bucket S3, configurado para almacenar un sitio web estático. Tenemos también una instancia EC2 dentro de la VPC default, en la que alojamos y desplegamos nuestro código de backend. En cuánto a nuestra base de datos, contamos con una RDS provista por Digital House



en la que está almacenada toda la información nuestro proyecto. Nuestra instancia EC2 se encuentra conectada a esta base de datos, asi como también nuestra API. Por último. contamos con otro bucket S3, que se encarga de almacenar todas las imagenes que utilizamos.

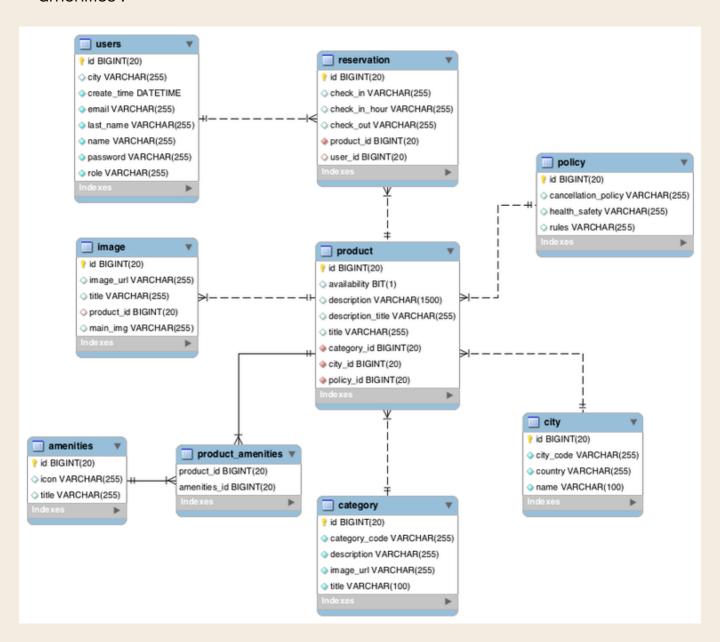
En cuánto a pipelines, contamos con dos, una para la rama de Front y otra para la rama de Development. Las pipelines son gatilladas al realizarse un push en la rama, y realizan tanto la compilación del código, como el deploy automático en AWS.

Diagrama BD Just Like Home

Hicimos uso de una base de datos relacional, compuesta de 9 tablas, una de ellas una tabla intermedia.

La tabla "product" es la tabla principal, pues es la que contiene al menos una relación con el resto delas tablas, a excepción de "user"; estas se conectan por medio de la tabla "reservation".

Existe solo una relación "muchos a muchos", esta se encuentra entre "product" y "amenities".



Documentación API

En el siguiente link se encuentra toda la documentación que contiene la información del funcionamiento de la API:

http://18.217.103.69:8080/swagger-ui.html

Informe Testing

En el siguiente link se encuentra toda la documentación relacionada a Testing:

https://docs.google.com/document/d/1ytWBeERyq7FDHSmTKq98bh6EkAJAbE p7ryliFMFzuPQ/edit?usp=sharing