## **Proyecto Integrador**

* Nombre del proyecto: Stay Easy
* Resumen general del proyecto:

El proyecto consiste en el desarrollo de una aplicacion web de servicios de alojamiento , aplicando todos los conocimientos adquiridos en la carrerta,

La aplicacion permite filtrar los alojamientos por categoria, por ciudad y tambien por fechas disponibles;

Parmite la navegacion libre de usuarios no registrados mostrando de forma aleatoria los alojamientos disponibles, y solo permite realizar la reservacion a usuarios registrados.

Para la realizacion de el proyecto utilizamos metodologias agiles utilizando principios de scrum y camban dividiendo el desarrollo de las funcionalidades en isues y realizando sprint de 2 semanas para realizarlos.

* Objetivo que persigue el grupo al realizar el proyecto:

El objetivo principal es poner en practica los conceptos aprendidos, tambien aprender a calcular el tiempo necesario para realizar cada issue, aprender a distribuir el trabajo y manejar el trabajo de una forma colaborativa.

## **Equipo**

**Mauricio Abril**

**Centrado en el area de frontend tambien realizando tareas en testing**

**Santiago Betancur**

**Centrado en el area de frontend**

**Silvia Guardo**

**Centrada en el area de frontend**

**Cristhian Perez**

**Centrado en el area de infraestructura y realizando tareas en frontend**

**Daniel Ramírez**

**Centrado en el area de backend y realizando tareas de testing**

**Pablo Novoa**

**Centrado en el area de backend y realizando tareas de frontend**

## **Metodología de trabajo**

Utilizamos la metodologia Scrum ya que nos permite dibidir las tareas y los tiempos para realizarlos de una manera muy manejable ayudandonos a coordinar el flujo de trabajo.

La asignacion de roles se realizo deacuerdo a las areas mas afines a la persona para poderlas realizar en un tiempo prudente dividiendo en consecuencia las tareas procurando no sobrecargar a las personas.

Tambien realizamos acurerdos de cuales dependencias utilizariamos y los momentos de hacer merge a las ramas en el git para evitar en lo posible los problemas con todo el codigo realizado.

## **Tecnologías utilizadas**

* IDEs
* Intellij Idea, Visual Studio code
* Sistemas de control de versiones

GitLab

* Simulación / Virtualización

Amazon Web Services (AWS)

* Tecnologías utilizadas para el desarrollo:
* Front end

React

Fontawesome

Styled components

* Back end

Java

* + Spring
  + Spring Boot
* Swagger
* Base de datos

MySQL

MySQL workbench

* Infraestructura

AWS

* + Deploy en AWS
  + Servicio RDS para alojar la base de datos MySQL
  + Amazon Elastic Compute Cloud (EC2) para alojar el backend (API)
  + S3 (Amazon Simple Storage Service) para alojar las imágenes.
* Testing

Pruebas de caja negra y caja blanca

Selenium

Postman

## **Documentación técnica del proyecto**

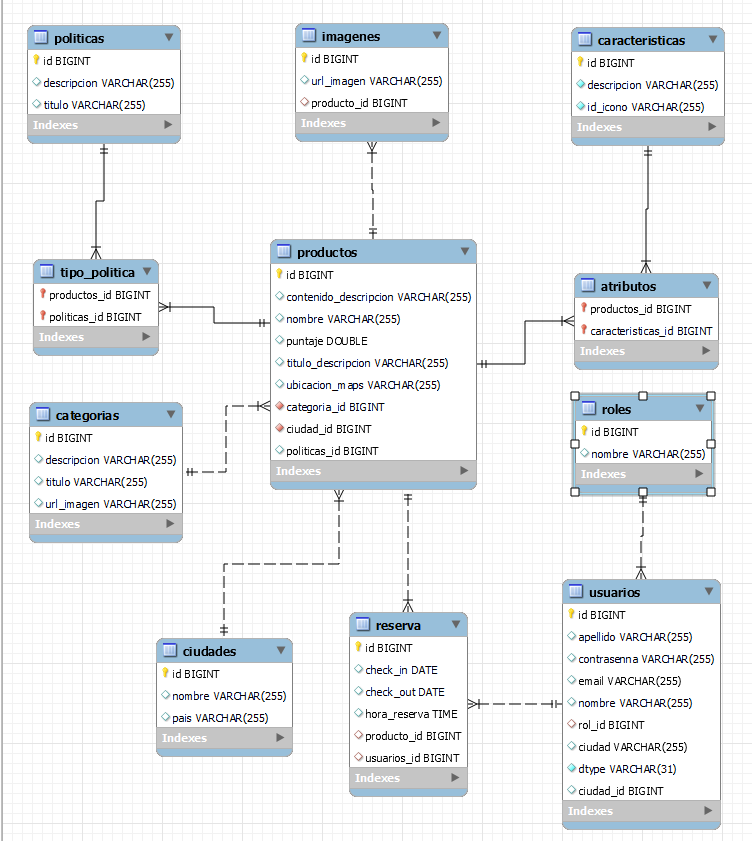
### **Ambiente de desarrollo**

**Instrucciones para armar el ambiente de desarrollo**

* Instalar los IDEs IntelliJ IDEA y Visual Studio Code.
* Instalar MySQL Workbench
* Instalar Postman
* Clonar el repositorio

**Base de datos:**

* Una vez abierto MySQL Workbench crear una nueva conexión o utilizar una ya existente.
* Desde el apartado File seleccionar “Run SQL Script” u “Open SQL Script” y correr el script con el **Configuracion\_MYSQL.txt** que creará la estructura de tablas e ingresara los datos iniciales.
* Estructura y diagrama



**Frontend:**

Desde Visual Studio Code, abriremos la terminal dentro de la carpeta de frontend.

Posicionado dentro la carpeta 'ProyectoX', usar npm install para instalar las dependencias necesarias.

$ npm install

Una vez completo el paso anterior, ejecutar el siguiente codigo en la consola:

$ npm run dev

* Esto nos iniciará el proyecto por default en localhost con el puerto 3000. [http://localhost:5173](http://localhost:3000/)

**Backend:**

* En el archivo de application.properties debemos colocar los datos de nuestra base de datos (MySQL Workbrench) local. Es decir, el username y password.
* Posteriormente ejecutar con el botón de RUN el proyecto backend desde el IntelliJ.
* Para probar los distintos endpoints podemos utilizar Postman ingresamos a la url [Swagger UI](http://localhost:8080/swagger-ui/index.html#/) donde se encuentran los endponts con la estructura de el JSON que se debe enviar, Se recomienda crear variables de entorno (environments) con el nombre 'url' para [http://localhost:8080](http://localhost:8080/) y 'token' para poner poner el token que devuelva el endpoint /authenticate, ya que así se encuentran definidos.
* Testing y calidad

Para la documentacion global de el testing se realizo un Informe Final de testing el cual se encuentra en el repositorio.

Ademas de que la documentacion de cada sprint está alojada en su respectiva carpeta,

Los test realizados incluyen :

* Pruebas de caja negra y caja blanca
* Pruebas en Selenium Web Driver
* Pruebas automatizadas en Postman para el consumo a la API