

PETSGRAM

Grupo P25

Integrantes:

- Cristian Leonardo Sánchez Rivera (Scrum Master)
- Geraldine Gómez Romero
- Alexander Viafara Hurtado
- Alejandro Arboleda López
- Luis Eduardo Ruz Menco

Proyecto JIRA:

- <https://petsgram.atlassian.net/jira/software/projects/PET/boards/1>

Repositorio GitHub:

- <https://github.com/Petsgram/petsgram>

Tareas Realizadas:

- Crear repositorio en GitHub
- Crear sitio y proyecto en Jira
- Definir un proyecto a realizar (desde cero)
- Crear diseño preliminar de la arquitectura de la aplicación.
- Crear documento PDF.

Explicación de código/arquitectura:

- La arquitectura planteada muestra que se va a desarrollar bajo una arquitectura basada en microservicios, donde se van a tener 3 microservicios obligatorios y uno opcional, este ultimo es opcional considerando el poco tiempo y analizando que tal vez no se pueda desarrollar en el poco tiempo que se tiene, así mismo se tienen 3 microservicios obligatorios que son:
 - Microservicio de Mascotas
 - Microservicio encargado de gestionar toda la información relacionada con las mascotas individuales de todos los usuarios de la aplicación.
 - Microservicio de Publicaciones
 - Microservicio encargado de gestionar todas las publicaciones que se realicen por las mascotas en la aplicación.
 - Microservicio de Autenticación o Autorización
 - Microservicio encargado de gestión de usuarios tanto su información personal como la generación de tokens.

Y un microservicio opcional que sería

- Microservicio de Amigos
 - Microservicio encargado de gestionar los amigos de cada mascota.

Así mismo se puede plantear crear un quinto microservicio para aislar la generación de tokens y el manejo de información personal de los usuarios en un microservicio cada uno.

Cabe resaltar que cada microservicio está conectado a una base de datos única para cada uno, lo que permite la integridad y disponibilidad de la mayor parte de datos en el peor de los casos.

Por otro lado, todos estos microservicios expondrán una API tipo REST la cual se conectará con el API Gateway el cuál será el encargado de gestionar y unificar los procesos de los distintos microservicios para que el componente de presentación pueda consumir esta API Gateway y le permita acceder a todos los datos de los distintos microservicios de igual manera.

Capturas de pantalla o imagen de arquitectura:

