Proyecto 1 entrega 1 - Arquitectura, conclusiones y consideraciones

Grupo 8

Arquitectura

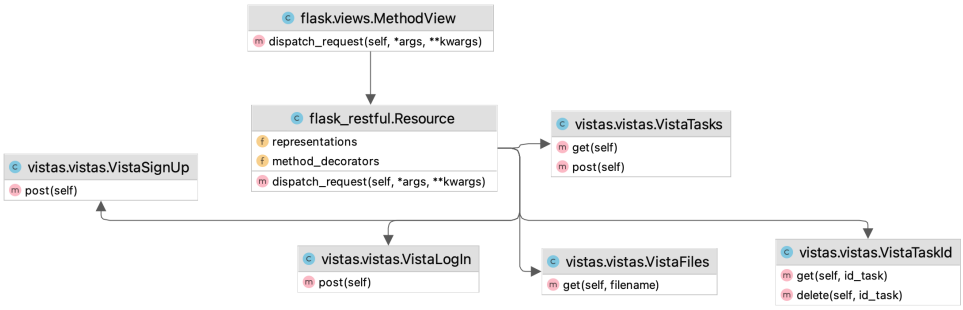
El modelo general de funcionamiento de la aplicación se basa en comprimir archivos cargados por un usuario. Para dicha funcionalidad cada usuario debe crear una cuenta en la aplicación y cargar los archivos que desea comprimir, los formatos a los cuales se puede comprimir son zip, tar.gz y tar.bz2.

Ya que la aplicación no cuenta con una interfaz gráfica el usuario deberá realizar las funcionalidades de registro, inicio de sesión y cargue de archivos a través de un API expuesto usando POSTMAN.

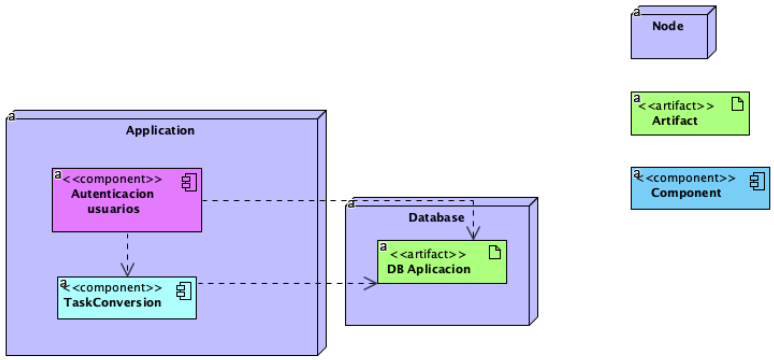
El desarrollo de la aplicación se realizó en Python usando el Flask como framework, se creó un endpoint por cada funcionalidad, cada uno de estos endpoints salvo SignUp y LogIn requieren un Json Web Token (JWT) para su consumo, este token se genera cada vez que el usuario inicia sesión en la aplicación.

Al realizar el cargue del archivo se genera una nueva tarea asociada al usuario que la cargó, esta tarea genera una nueva entrada en Redis que funciona como broker entre Celery Flask, esto permite realizar la compresión de los archivos de forma asíncrona. El estado de la tarea se actualiza cuando la tarea sea procesada. Para la persistencia de los datos se usó Postgres como motor de base de datos.

- Modelo de dominio



- Diagrama de despliegue



Para una siguiente iteración en la que se quiera aumentar el número de usuarios de forma concurrente, se recomiendo escalar horizontalmente la aplicación y así balancear la carga de usuarios por servicio.