

PROYECTO 1 ENTREGA 1

APLICACIONES WEB ESCALABLES EN UN ENTORNO TRADICIONAL

Integrantes: Cristian M. Amaya (201325778) – Nicolas Hernandez (201412420)

Arquitectura de la aplicación

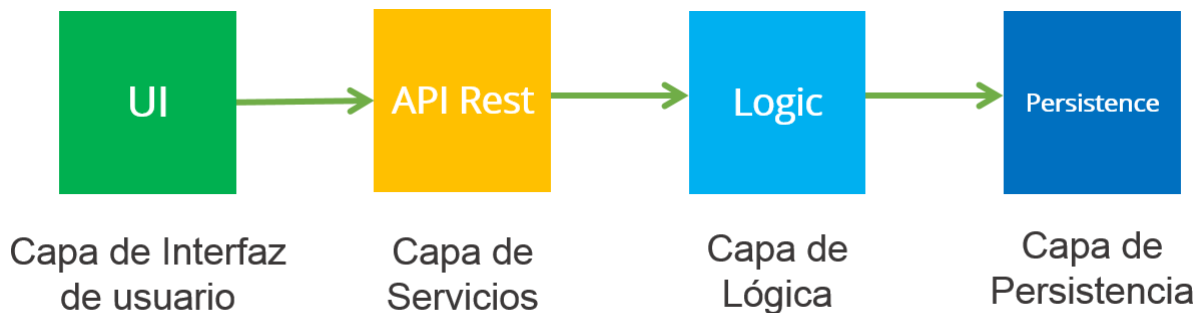


Ilustración 1. Capas de la Arquitectura

La aplicación web base se basó una arquitectura funcional de cuatro capas:

- Capa de Interfaz Gráfica: al Frontend desarrollado en React (JavaScript).
- Capa de Servicios: Backend desarrollado en Node.JS, por medio de un API Rest se ofrece las operaciones CRUD sobre los recursos de la aplicación para el Frontend.
- Capa de Lógica: Conexión entre los servicios y la capa de persistencia, realizada por medio de modelos y controladores desarrollados con la librería Sequelize. Adicionalmente permite la conexión entre los servicios, controladores y el proceso batch, para realizar la verificación de archivos a procesar en la base de datos y el encolamiento de nuevos archivos para ser procesados.
- Capa de Persistencia: Base de datos relacional PostgreSQL corriendo en una máquina virtual separada.

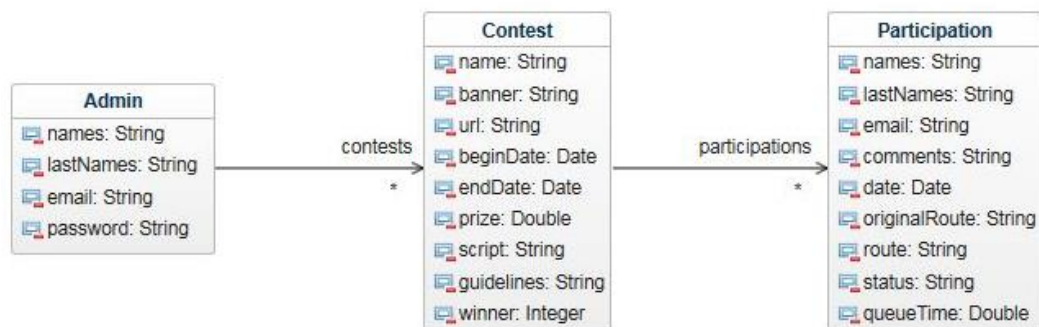
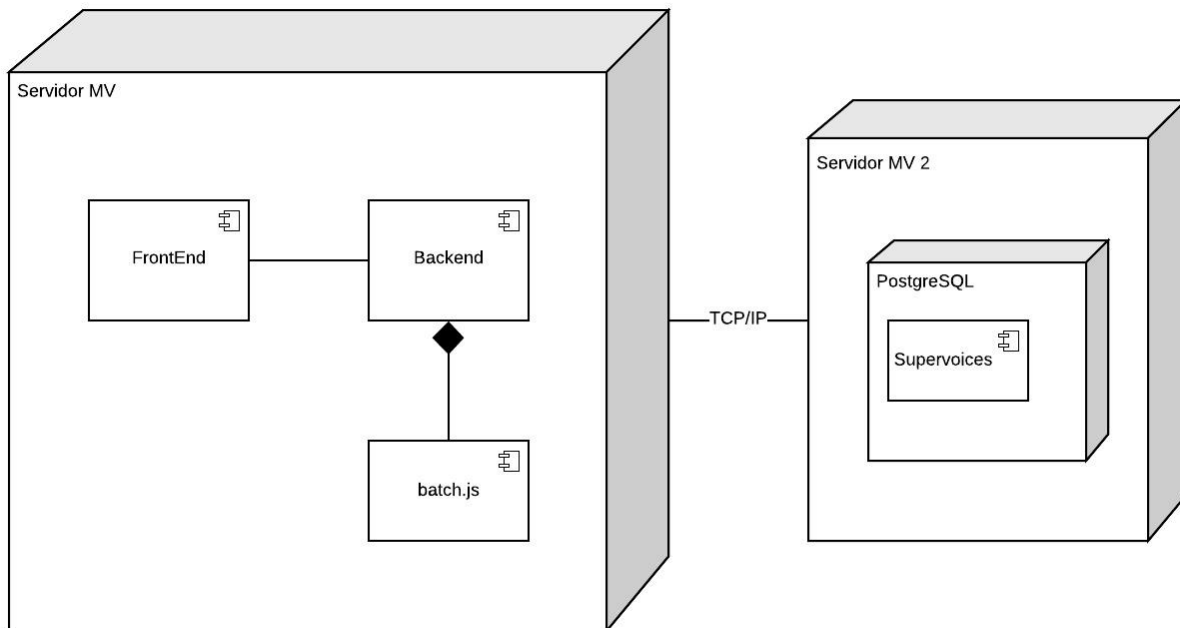


Ilustración 2. Modelo UML de las clases de la aplicación

Despliegue (máquinas) de la aplicación:

La aplicación es desplegada en dos máquinas distintas: una tiene el front-end, back-end y procesamiento batch, y la otra tiene la base de datos.



Conclusiones Pruebas de Stress

Escenario 1

La limitación de tiempo en respuesta fue la que definió el número de solicitudes que puede procesar nuestra aplicación. Alrededor de las 130 solicitudes se estipuló que la página es capaz de responder en tiempos y porcentajes de error aceptables.

Dicho bajo rendimiento se puede deber a la poca escalabilidad y capacidad que tiene la infraestructura y arquitectura que está actualmente implementada: sólo existe un servidor que procesa todas las solicitudes HTTP y, este mismo, realiza las conversiones de audio.

Escenario 2

Se pudo comprobar la capacidad del sistema, siendo capaz de manejar 555 archivos de audio por minuto antes de llegar al límite de tiempo establecido de espera en cola. Los procesos auxiliares, como lo son las escrituras y lecturas en las bases de datos y el envío de correos, pueden ser limitantes a la hora de procesar grandes volúmenes de datos.

Adicionalmente, una manera para incrementar la capacidad del sistema batch sería aumentar el número de procesos de conversión en paralelo, aunque este número de procesos

en paralelo debe ser determinado por medio de que tantos procesos de ffmpeg puede manejar el sistema.

Consideraciones de Escalabilidad

La aplicación que desarrollamos aun tiene bastante por mejor para llegar a ser escalable, en términos de limitaciones tenemos: la baja latencia de REST y de la base de datos, una única instancia de la aplicación corriendo sobre una única máquina virtual, la capacidad del proceso Batch es limitada por los recursos de la máquina, no hay mecanismos para garantizar o monitorear la disponibilidad y el mal acoplamiento del proceso Batch con el Backend y base de datos. Para solventar estas limitaciones, algunas de las posibles soluciones son las siguientes:

- Redundancia y Balanceo de Carga: Por ahora el sistema solo se ejecuta sobre una máquina virtual, el aumentar el numero de unidades que ejecutan la aplicación y un balanceador de cargas aumentaría la capacidad, la escalabilidad y la disponibilidad del sistema. Uno de los grandes problemas es el sistema de archivos de audio, el cual reside actualmente sobre la misma máquina de la aplicación, almacenar los archivos en un sistema de común acceso sería necesario.
- Descomposición: Desacoplar funciones y servicios en diferentes aplicaciones, esto para no tener una aplicación monolítica y aumentar la independencia y escalabilidad del sistema.
- Cache: Tener disponible en memoria la información mas solicitada puede ayudarnos a reducir la latencia en la respuesta de solicitudes.
- Base de datos: Estamos usando una base de datos relacional, la cual puede no ser la indicada para el manejo de grandes volúmenes de datos. Adicionalmente puede que no estemos optimizando las operaciones sobre la base de datos, ya sea por la falta de implementación de índices o por la configuración de la librería de manejo de bases de datos.