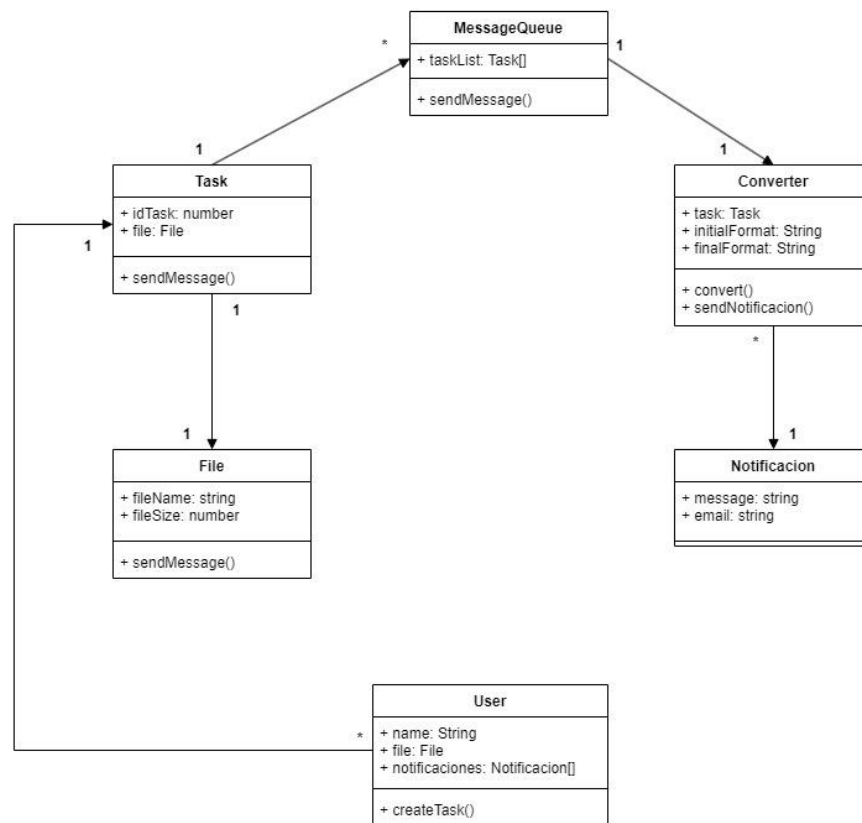


# Proyecto 1 entrega 1

## Arquitectura, conclusiones y consideraciones

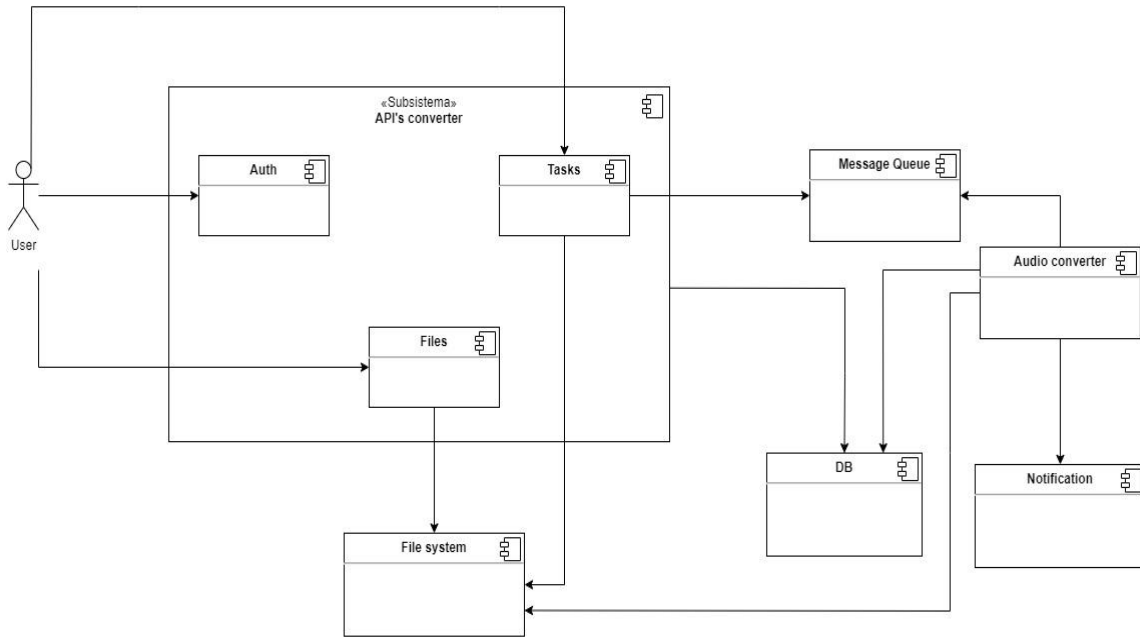
### Arquitectura

#### Modelo de Dominio



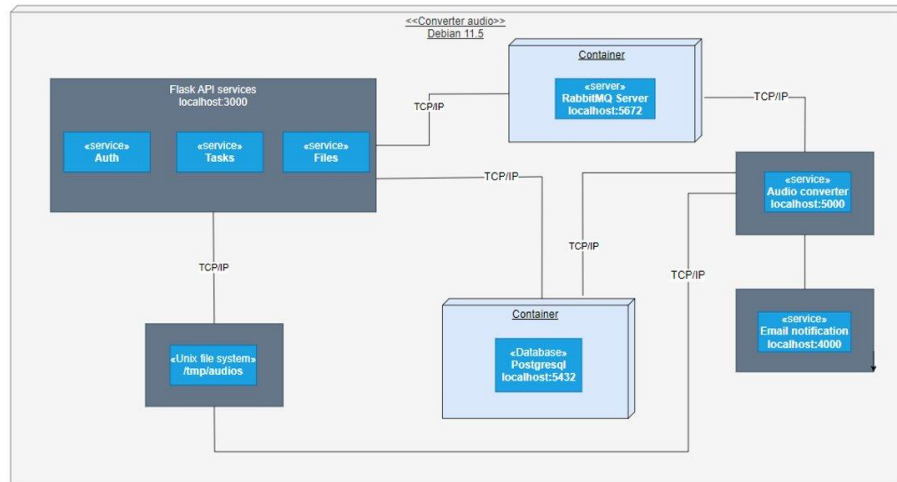
En el modelo de dominio se pueden ver las relaciones, dependencias y funciones básicas del sistema, se resaltan las relaciones 1 a 1 entre un archivo y una tarea y entre la cola de mensajes y el convertidor, del lado de la funcionalidad el Converter resalta ya que es en donde se realiza el proceso central de la aplicación (convert) y también es el único capaz de enviar notificaciones

## Modelo de Componentes



En el modelo de componentes resalta el subsistema API's converter, que es la API REST por donde se recibe toda la información, el resto de sistemas se mantienen de manera independiente y estos son la cola de mensajes, el Audio converter, el sistema de notificaciones y las persistencias de datos y de archivos

## Vista de despliegue



Esta aplicación se basa en un API REST que realiza la conversión de formato de un archivo de audio, el sistema usa una cola de mensajes de la cual el convertidor de archivos obtiene el archivo y la información relacionada para luego convertir el archivo

**Application server:** es donde reside la API REST y por ende por donde se recibe toda la información para realizar el procesamiento de los archivos de audio, desde acá también es posible obtener archivos convertidos de lo cual es necesaria la comunicación con la cola de mensajes, la base de datos y el File System

**File System:** es donde se guardan los archivos de audio originales y convertidos

**Database Server:** dado que se requiere persistir el estado de un archivo que ha sido subido para ser convertido se optó por usar una base de datos donde se guarda y actualiza dicha información

**Queue Server:** Cola de mensajes usada para enviar la información al Audio Converter de manera asíncrona

**Container (Audio Converter):** encargado de convertir los archivos, guardar los archivos nuevos, notificar la finalización de una conversión y actualizar la base de datos una vez se ha terminado el proceso

## Limitaciones

El proyecto tiene un alcance definido, es una api con autenticación y operaciones CRUD en una base de datos que también gestiona archivos con funcionalidades asíncronas y notificaciones, todas actividades comunes en diferentes tipos de backend, las pruebas se limitaran a 2 escenarios y se harán pruebas con Postman, de igual manera los sistemas se pondrán en ejecución en máquinas virtuales proveídas por Google Cloud haciendo uso de recursos que serán pagados con créditos ofrecidos por el mismo Google.

Respecto a elementos que no van a ser incluidos en el desarrollo están: el front de la aplicación, respecto a funcionalidades del back facturación, pagos, entre otros no serán tampoco incluidos.

## Instrucciones para replicar ambiente (Opcional)

### Pruebas de carga

Documentación pruebas Postman

[Pruebas AudioConverter \(getpostman.com\)](https://getpostman.com)

Escenario y Pruebas de Estrés API REST y Batch

(cambiar a link pdf)

[Escenario y Pruebas de Estrés API REST y Batch](#)