Distributed Music Editor

Computação distribuida 2022/2023



Arquitetura

Elementos principais



RabbitMQ

Coordena a troca de mensagens



FrontEnd

User Interface que tem a correspondência no server



Server

Envio e recepçao de audio files para e de as respetivas queues Fornecimento da informaçao para o frontend



Worker/s

Recebe o work da queue processa o mesmo e envia o resultado para a queue



ThreadPollExecutor

Gere e cria as threads das conexoes

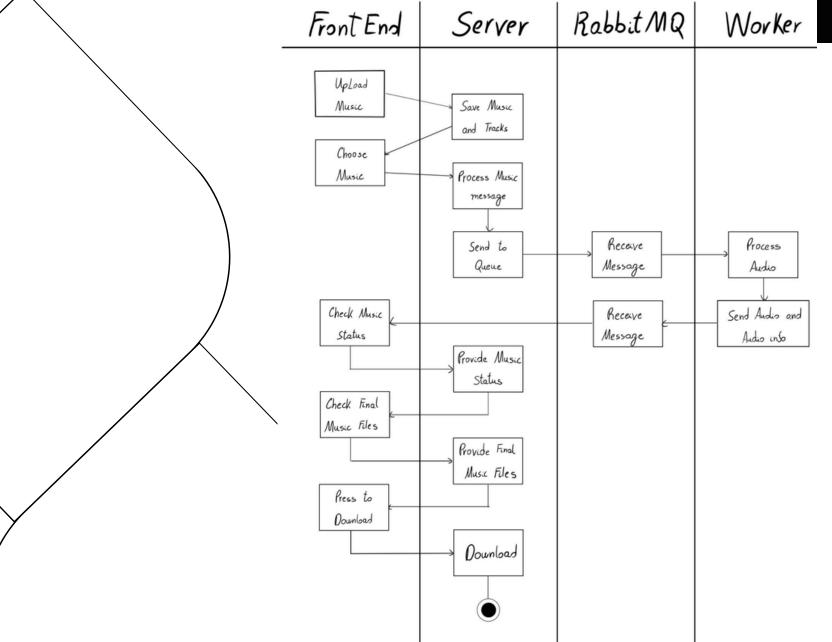


Demucs

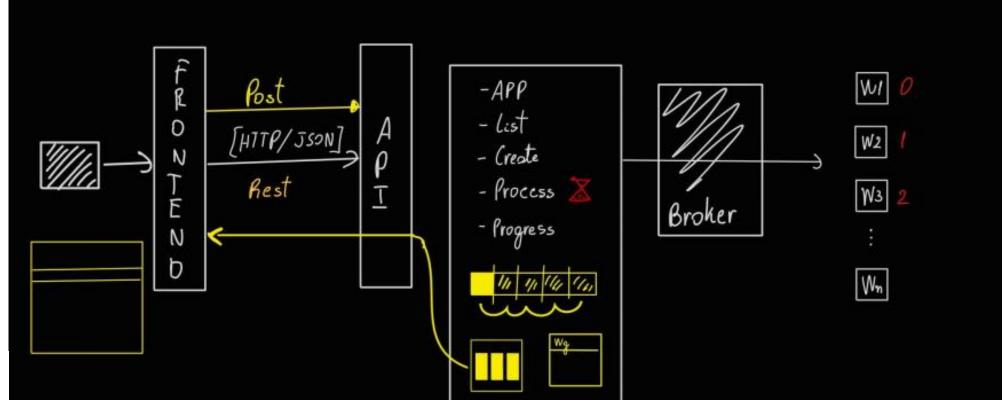
Algoritmo/Modulo/Biblioteca utilizada pelo worker para processar o audio file

Diagramas de execução

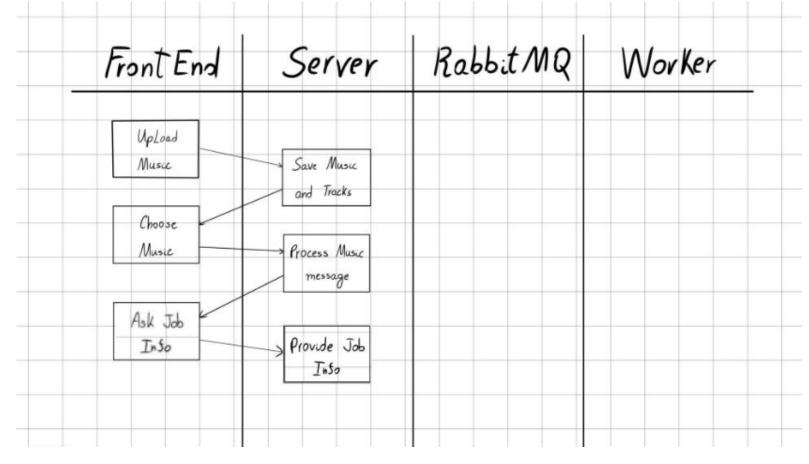
Music Request



Esboço Inicial



Job Request



Protocolo

O protocolo utilizado neste projeto é o AMQP (Advanced Message Queuing Protocol). O AMQP é o protocolo base no qual foi construído o RabbitMQ.

Este protocolo trabalha na camada de aplicação, determinando como devem ser transmitidos os dados ao longo do sistema.



Headers

Utilizados para transmitir informações relevantes sobre as tarefas e mensagens trocadas entre o server e workers.

Desta forma, é garantida a concordância entre o servidor e os workers

Linha do Desenvolvimento

2° Versão

Divisão do ficheiro de audio em 4 partes Vários workers possíveis Aumento significativo da performance

4° Versão

Divisão do ficheiro de audio em 8 partes (para permitir troca ficheiros com maior tamanho) Implementação de keep alive para a conexão do worker permanecer ativa

1° Versão

Trocas entre servidor e um worker Baixa Performance

3° Versão

Workers enviam sempre todas as tracks (merge desejado incluido)
Server em caso de modificaçao das tracks de uma musica realiza o merge Aumento significativo da performance numa execuçao posterior á primeira