



# Projeto Final

Computação Distribuída

Prof. Diogo Gomes

Prof. Alfredo Matos

2022/2023

Roberto Castro

107133

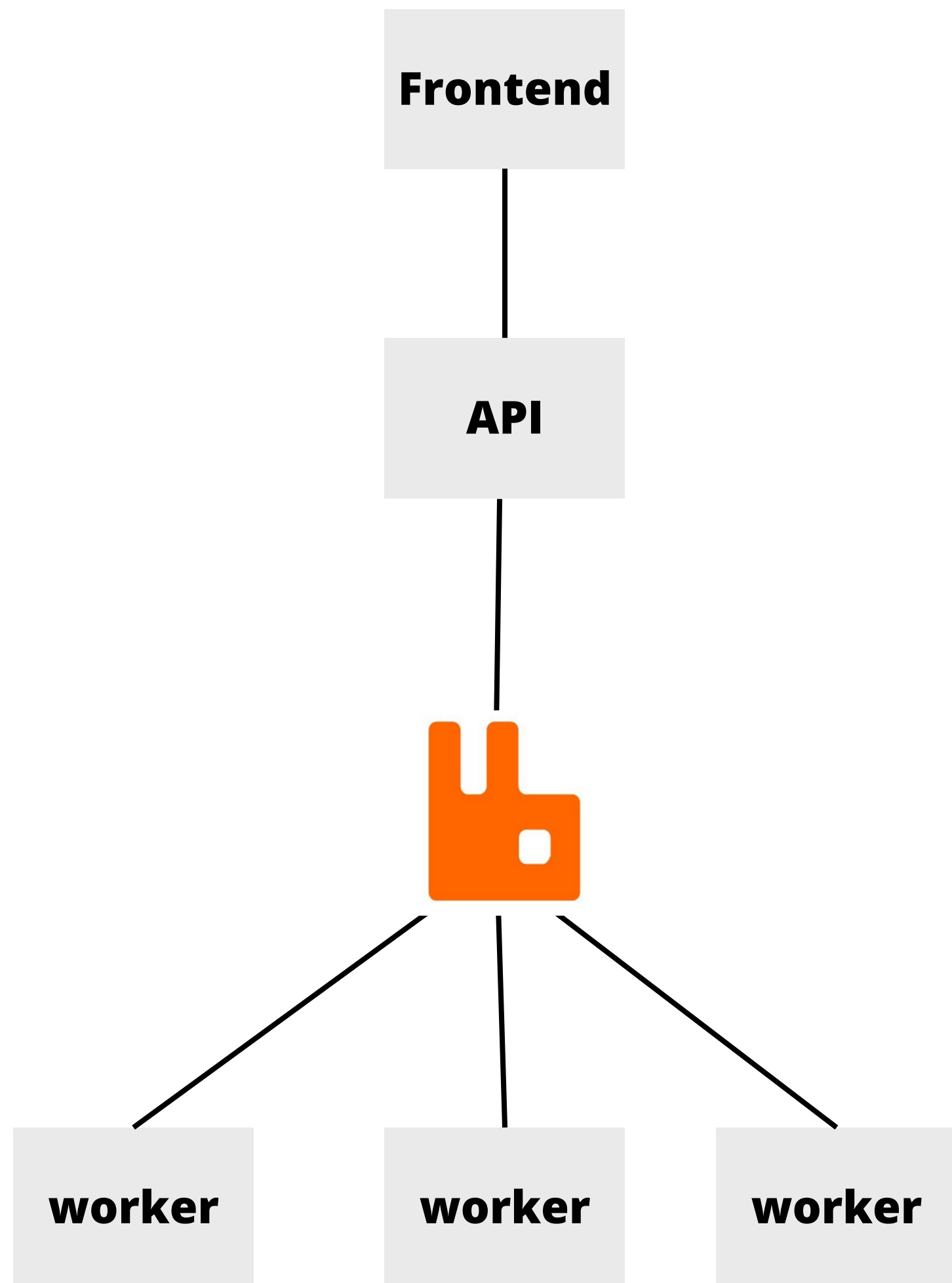
Tiago Gomes

108307



# Arquitetura Client-Server

- Implementação do serviço web foi feito com API FLASK
- O servidor é responsável por dividir a música em diferentes tracks e de seguida enviar cada parte para o respetivo *worker* que irá processar a track. Assim que os *workers* acabarem de processar enviam o resultado obtido para a queue '*join\_worker*', para que esta junte o que foi pedido pelo utilizador no *frontEnd*.



# Message-broker Rabbit.MQ

Comunicação através de um message Broker (Rabbit.MQ) que estabelece a ligação entre o server (API) e os workers (também conhecidos como consumidores), estes recebem mensagens das filas e processam as tarefas associadas



# Resultados obtidos

- Implementação do serviço web com uma API
- O processamento da musica submetida:
  - Divisão da música em diferentes tracks
  - Criação de um novo ficheiro consoante as opções do utilizador
- Apagar todos ficheiros temporários e músicas

## Upload de Música

Choose file No file chosen

Enviar

Ver Músicas Submetidas

## Selecione os instrumentos a serem juntados:

Vocals ☐ Bass ☐ Drums ☐ Other ☐

Processar Música

RESET

▼ 6a2d07f9-0f7a-47e3-adfb-dc874f6e0091

> instrument\_worker1

> instrument\_worker2

> instrument\_worker3

🔊 combined\_music\_bass.wav

🔊 combined\_music\_drums.wav

🔊 combined\_music\_other.wav

🔊 combined\_music\_vocals.wav

🔊 combined\_musicSELECTED.mp3

# O que falta implementar...

- A introdução e apresentação correta das informações que a API possui
- A criação de disponibilização dos links para download das tracks
- Multiproxessing mais eficiente
- Atender a múltiplos clientes em simultâneo
- Tolerância a falhas