



Universidade do Minho

Mestrado Integrado de Engenharia Informática

# PROJETO Aurras: Processamento de Ficheiros de Audio (MIEI –SO 20/21)

## Grupo N°40:

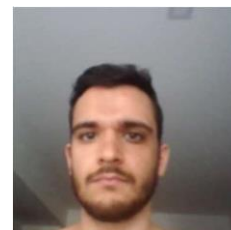
- Pedro Aquino Martins de Araújo - A90614



- Rafael dos Anjos Arêas - A86817



- Vitor Lelis Noronha Leite - A90707



## ÍNDICE

- Introdução..... 3
- Desenvolvimento.....4
- Conclusões e desafios.....5

# Introdução

## Projeto:

### - Objetivo?

- Desenvolver dois programas , um para o usuário, outro para o servidor, que se comuniquem . (Aurras e Aurrasd consecutivamente)
- O usuário é suposto passar um ficheiro de áudio, uma sequencia de filtros que deseja aplicar neste, e o ficheiro output desta transformação.
- O servidor deve receber as requisições do user e executar elas. Deve dar respostas ao usuário do estado da execução.

### - Competências presentes e necessárias:

- Pipes com nomes : para comunicação entre servidor e usuário
- Pipes anônimos: para execução de mais de um filtro
- Fork: para execução paralela e geração de diferentes processos
- Tratamento de sinais: para uma saída graciosa do programa

# Desenvolvimento

## User/Aurras:

- É responsável por criar a pipe com nome e escrever nela os argumentos passados na argv, passando-os para o server.
- Fica de standby, aguardando respostas do server até que o usuário pressione `ctrl^ \`, que irá permitir que o programa seja encerrado.

## Server/Aurrasd:

- Possui papel principal no projeto, responsável por fazer toda execução.
- Começa-se, por ler do argv o path para onde está os filtros, e o path das configurações do servidor. Foi criado uma struct que guarda as linhas ( assumi-se que são 5 ), e separa o campos de cada linha.
- Entra em um loop para receber os pedidos do user. É feito o parse dos argumentos para recolha dos dados.
- Depois de possuir os argumentos necessários, verifica qual tipo de pedido que o usuário fez, caso seja um transform, encaminha para a execução dos filtros. Durante este processo, o usuário é acompanhado de como está o estado do pedido.
- Após a execução completa do pedido, fica à espera de novos pedidos, ou até que seja invocado `ctrl^ \`, que irá sair graciosamente do programa.

## **Conclusões e desafios**

O principal desafio foi saber balancear o tempo com outras obrigações curriculares simultaneamente. Mas em relação às dificuldades encontradas no projeto, algumas que custou-se um bom tempo e vale destacar: realização do parse das strings corretamente, utilização do read com um buffer correto, além de conseguir fazer perfeitamente a sincronização entre user e servidor.

Conclui-se, que não foi cumprido com todos os objetivos requisitados do projeto, mas conseguimos cumprir e trabalhar com todas as competências que foram estudadas na cadeira. Contudo, o projeto serviu como benefício, no sentido de nos obrigar a superar as dificuldades, adquirindo conseqüentemente, conhecimento.