



Instituto Superior de Engenharia de Coimbra

Licenciatura em Engenharia Informática

## Sistemas Operativos

# Trabalho Prático Meta 2- Relatório

Bruno André Rodrigues Cancela – 2020131288 LEI

Rodrigo Miguel Coelho Carvalho – 2020145787 LEI

Coimbra, 2 de janeiro de 2024

## Conteúdo

1	Requisitos implementados .....	3
2	JogoUI .....	3
3	Motor .....	4
4	Conclusão.....	6

## 1 Requisitos implementados

### Motor

- Receber e interpretar o input de vários JogoUI
- Interpretar comandos do administrador

### JogoUI

- Envia input ao Motor
- Receber e interpretar o input do Motor
- Interpretação de comandos do Jogador

## 2 JogoUI

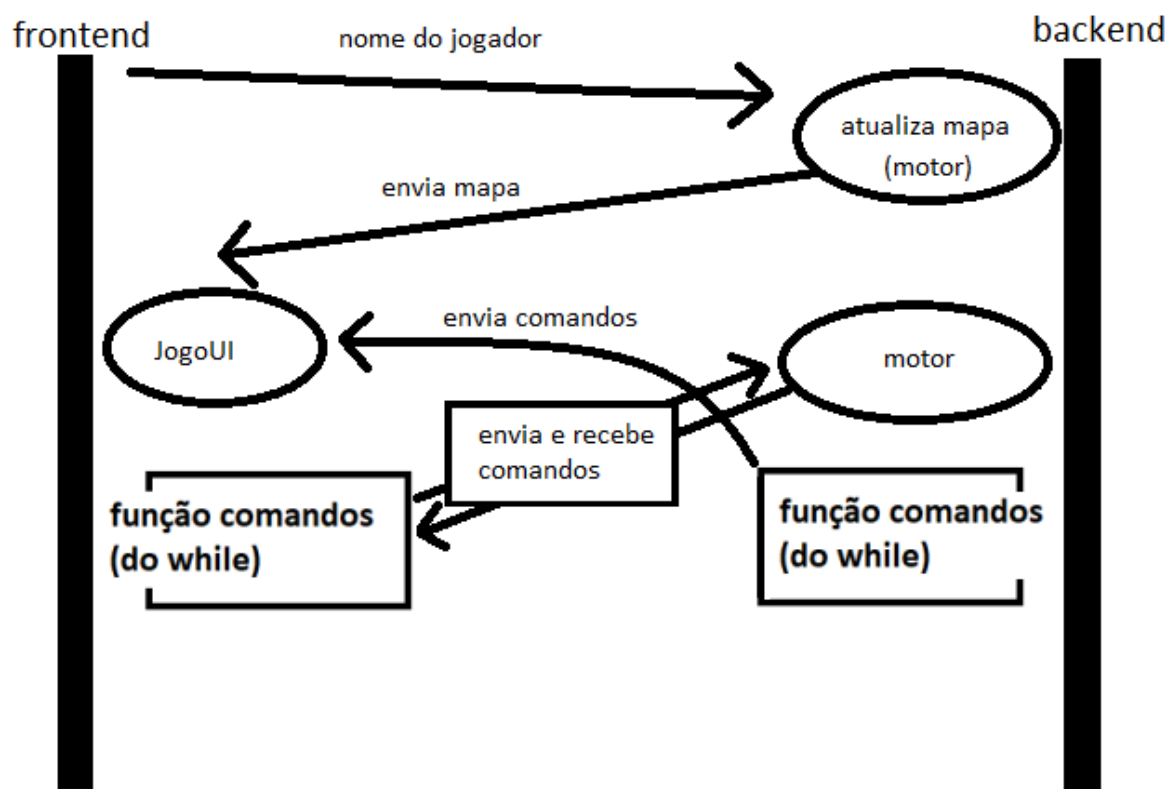
O JogoUI começa por enviar os seus dados ao Motor através do **FIFO** específico para o Motor. Se o utilizador entrar com sucesso, irá ser criado um **FIFO** para esse utilizador onde irá receber informação do motor e será criada um **THREAD** que estará num loop infinito a ouvir/ler a informação que entra no **PIPE** do utilizador.

Além disso, tem um **SIGNATION HANDLER** permitindo ao JogoUI fechar corretamente.

### 3 Motor

O Motor vai criar um **FIFO** específico só para ele, onde ira receber os comandos dos JogosUI e onde envia o mapa atualizado para o JogoUI.

Lógica de implementação



Dentro do Motor temos várias **THREADS** sendo estas as seguintes:

- Uma que fica a ler infinitamente o **PIPE** do Motor para ler os comandos do JogoUI;
- Uma para o timer de inscrição de Jogadores;
- Uma para o timer de final de jogo;
- Uma para dar setup às **THREADS** do bot(foi feito desta maneira para a **THREAD** só terminar quando a **THREAD** de cada bot terminar);
- Uma por cada bot iniciado;
- Uma para cada Pedra;
- Uma por cada bloqueio movel;

## 4 Conclusão

A base do projeto está criada com recurso a toda a matéria abordada durante as aulas de Sistemas operativos:

- Uso de FIFOs para a comunicação entre vários frontends e um backend;
- Gestão de estruturas partilhadas entre várias THREADS com o uso de MUTEX;
- Uso de sinal para impedir o frontend e backend de fechar sem segurança;
- Interpretação de todos os comandos;
- Uma função para ler ficheiros de texto;

Praticamente todo o código teve testes para se saber se algo aconteceu com sucesso (por exemplo, ao abrir um FIFO, fazer `if (fifo == -1)`, como muitos outros métodos de segurança.