

**Relatório**

**Estruturas de Dados Avançadas**

**Aluno: André Freitas nº21112**

**Professor/es: João Carlos**

**Luís Gonzaga**

**Licenciatura em Engenharia de Sistemas Informáticos**

Barcelos, abril, 2022

**Resumo:**

Trabalho individual que visa a criação de um projeto em C de forma a resolver um problema denominado por “Flexible Job Shop Problem”.

**Índice**

[1. Introdução 1](#_Toc101209350)

[1.1. Propósito 1](#_Toc101209351)

[1.2. Objetivo 1](#_Toc101209352)

[2. Estrutura de Dados 2](#_Toc101209353)

[3. Testes realizados 3](#_Toc101209354)

[3.1. Inserção de operações 3](#_Toc101209355)

[3.2. Remoção de operações 3](#_Toc101209356)

[3.3. Alteração de operações 4](#_Toc101209357)

[3.4. Determinação da quantidade mínima de tempo 4](#_Toc101209358)

[3.5. Determinação da quantidade máxima de tempo 5](#_Toc101209359)

[3.6. Determinação da quantidade média de tempo 5](#_Toc101209360)

[4. Conclusão 6](#_Toc101209361)

[5. Bibliografia 6](#_Toc101209362)

# 1. Introdução

## 1.1. Propósito

Este trabalho prático de realização individual da Unidade Curricular (UC) Estruturas de Dados Avançadas (EDA) visa o reforço e a aplicação dos conhecimentos adquiridos ao longo do semestre.

## 1.2. Objetivo

O objetivo deste trabalho prático reside no desenvolvimento de uma solução digital para o problema de escalonamento denominado Flexible Job Shop Problem (FJSSP). A solução a implementar deverá permitir gerar uma proposta de escalonamento para a produção de um produto envolvendo várias operações e a utilização de várias máquinas, minimizando o tempo as unidades de tempo necessário na sua produção

# 2. Estrutura de Dados

Para esta fase do trabalho apenas foi criada uma estrutura que terá o nome de “Maquina”, onde será possível guardar a identificação das operações, dos procedimentos e o tempo de cada um. Além disso terá apontadores “next” e “prev” para poder percorrer a lista para a direita e para a esquerda.

Uma imagem com texto

Descrição gerada automaticamente

# 3. Testes realizados

### 3.1. Inserção de operações

Para esta tarefa o programa ira adicionar na lista as operações que o utilizador deseja.

Além disso irá também listar toda a informação, bem como registá-la num ficheiro de texto que neste caso se chama “plano\_de\_processos.txt”. Além disso também foi feita uma função encarregue ler qualquer ficheiro de texto solicitando esta informação.

Por exemplo neste caso o utilizador inserir 6 operações.

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamenteDepois de aceder o menu escolhemos a opção relativa a esta tarefa e como podem ver todas as inserções forma listadas corretamente de forma ordenada. Além disso é criado o ficheiro de texto aonde também fica registado todas as inserções.

### 3.2. Remoção de operações

**Uma imagem com texto

Descrição gerada automaticamente**Uma imagem com texto

Descrição gerada automaticamentePara remover as operações o programa pede a posição da operação de modo a poder identificar qual a operação a remover. Como demonstrado em baixo.

### 3.3. Alteração de operações

Atendendo o pedido de alteração de operações decidi alterar o tempo das operações dado que o pedido feito no enunciado do projeto é muito vago. Este procedimento é feito à semelhança da remoção de uma operação, no propósito de termos de escolher a posição da operação na qual queremos alterar o tempo. Mas tendo em atenção que o “index” das posições desta função começa em 0, o que significa que se quisermos escolher a segunda posição digitamos 1.

Isto pode ser verificado na demonstração em baixo:

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamente

### 3.4. Determinação da quantidade mínima de tempo

Esta proposta foi mais complicada que as últimas abordadas, mas sinceramente foi a mais intuitiva de programar.

Foi necessário criar duas funções para resolver isto. Uma para determinar o tempo mínimo de cada operação e uma para fazer o somatório dos tempos mínimos de todas operações de modo a determinar a quantidade de tempo mínima pra completar um job.

Por exemplo neste caso o utilizador inserir 6 operações.

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamenteComo é possível ver para a operação 1 o tempo mínimo é de 3 segundos, para a operação 2 é 3 segundos e para a operação 3 como só temos um procedimento o seu tempo mínimo é de 4 segundos. Logo somando os valores deverá dar 11 segundos.

### 3.5. Determinação da quantidade máxima de tempo

Sem grandes surpresas, o processo por detrás desta resolução é exatamente igual à proposta anterior. Mas neste caso em vez do menos iremos determinar o maior.

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamentePara a operação 1 o tempo máximo é de 4 segundos, para a operação 2 é 5 segundos e para a operação 3 como só temos um procedimento o seu tempo máximo é de 4 segundos. Logo fazendo o somatório destes valores deverá dar 13 segundos.

### 3.6. Determinação da quantidade média de tempo

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamentePara a determinação da média de tempo de operação foi preciso percorrer a lista e identificar o apontador da operação de modo a poder separar os tempos para depois fazer a média de cada um separadamente. Esta função é mais intuitiva pra o utilizador dado que ele pode determinar qual a operação este quer saber a média da quantidade de tempo. Por exemplo usando a operação 2 temos duas quantidades de tempo, 5 e 3 segundos logo a sua média será de 4 segundos.

# 4. Conclusão

Devido ao facto de ser a minha segunda vez a abordar esta disciplina foi mais fácil a realização desta fase devido aos conhecimentos que fui colecionando desde o ano passado. Mas mesmo assim o trabalho teve a sua dificuldade daí também ter demorado algum tempo para fazer.

Prova disso é o facto de duas funções não funcionarem a cem por cento, mas fico feliz com o meu desempenho neste trabalho, contando com o facto de este trabalho ser individual, algo que realmente puxou pela minha capacidade de programação. A autonomia e capacidade de pesquisa própria realmente ajuda na aprendizagem e é algo que irá ser frequente futuramente no mercado de trabalho.

# 5. Bibliografia

Repositório do professor Luís:

* [Listas Ligadas](https://github.com/luferIPCA/LESI-EDA-2122/tree/master/Aulas/Aula%204%20-%20Listas%20Ligadas)
* [GereMaquinas](https://github.com/luferIPCA/LESI-EDA-2122/tree/master/Aulas/GereMaquinas)

Eventuais dúvidas que apareceram ao longo do projeto forma resolvidas junto do professor ou com que recurso ao site [**Stack Overflow**](https://stackoverflow.com/)**.**