



## **Relatório Fase 2**

### **Estruturas de Dados Avançadas**

Aluno

João Carlos da Costa Apresentação

Docentes

Luís Ferreira

João Silva

Licenciatura em Engenharia de Sistemas Informáticos

Barcelos, Abril, 2022

## Índice

### Conteúdo

Introdução.....	3
Propósitos e Objetivos .....	4
Estruturas de Dados.....	5
Estrutura das Máquinas .....	5
Estrutura dos Jobs .....	5
Estrutura de conexões.....	6
Estrutura das Células de Plano .....	6
Testes realizados.....	7
Inserção de um job .....	8
Remover job .....	9
Inserção de uma operação em um job.....	10
Alterar operação de um job .....	11
Remoção de uma operação de um job .....	12
Tempo mínimo da realização do job .....	12
Tempo máximo da realização do job .....	13
Tempo médio de uma operação de um job .....	13
Resultado do escalonamento.....	14
Alterações .....	15
Conclusão .....	16
Bibliografia .....	17

## Introdução

Relatório da 2ª fase do projeto realizado em linguagem C utilizando ferramentas instruídas ao longo do semestre (VS Code, Doxygen, GitHub, ...) na unidade curricular Estruturas de Dados Avançadas integrada no 2 semestre do 1º ano da licenciatura Engenharia de Sistemas Informáticos.

## Propósitos e Objetivos

A realização deste projeto tem como propósito a demonstração dos conhecimentos adquiridos dentro e fora de aula, além do empenho para a sua concretização.

Como objetivos, é pretendido o desenvolvimento de uma solução para o problema de escalonamento, com o envolvimento de jobs , operações e várias máquinas de forma a minimizar o tempo de produção de um produto.

Para esta 2ª fase é pretendido o seguinte conjunto de funcionalidades:

1. Definição de uma estrutura de dados dinâmica para representação de um conjunto finito de  $m$  jobs associando a cada job um determinado conjunto finito de operações;
2. Armazenamento/leitura de ficheiro de texto com representação de um process plan (considerar obrigatoriamente para efeito de teste o process plan da Tabela 1);
3. Inserção de um novo job;
4. Remoção de um job;
5. Inserção de uma nova operação num job;
6. Remoção de uma determinada operação de um job;
7. Edição das operações associadas a um job;
8. Cálculo de uma proposta de escalonamento para o problema FJSSP (obrigatoriamente limitado a um tempo máximo de processamento configurável), apresentando a distribuição das operações pelas várias máquinas, minimizando o makespan (unidades de tempo necessárias para a realização de todos os jobs). A proposta de escalonamento deverá ser exportada para um ficheiro de texto possibilitando uma interpretação intuitiva (utilizar por exemplo um formato tabular ou representação gráfica html, ou outra);
9. Representação de diferentes process plan (variando a quantidade de máquinas disponíveis, quantidade de job, e sequência de operações, etc) associando as respetivas propostas de escalonamento. Para a realização do mesmo espera-se o recurso a manipulação de ficheiros e apontadores, e estruturas dinâmicas, além do abordado de makefile e documentação gerada por Doxygen.

## Estruturas de Dados

Para a resolução desta fase 2 foi utilizado o recurso a 4 estruturas, e tendo sofrido algumas alterações em relação á 1ª Fase, entre as quais:

- Remoção das estruturas:
  - Operation;
  - MacOp;
- Adição de novas estruturas:
  - Job;
  - Connection;
  - CelulaPlano;
- Alterações
  - A estrutura intermédia passou a ser a Connection;
  - A estrutura Machines deixou de ter o atributo “time”;

Dentre a atual estrutura de dados seguem-se especificadas as mesmas:

### Estrutura das Máquinas

Contem todas as máquinas do projeto e que como variáveis o seu “id\_mac” como identificação. Além disso ainda tem o apontador para o próximo e para o anterior nodo da lista. Este será armazenado em um ficheiro para leitura e escrita. Anteriormente, este continha o atributo “time” que foi trocada de estrutura para o Connection.

```
typedef struct list_machine
{
    int id_mac;
    struct list_machine *next, *previous;
} Machine;
```

### Estrutura dos Jobs

Contem todos os jobs do projeto e que como variáveis tem o seu “id\_job” como seu identificador. Além disso ainda tem o apontador para o próximo e para o anterior nodo da lista. Este será armazenado em um ficheiro para leitura e escrita.

```
typedef struct list_job
{
    int id_job;
    struct list_job *next, *previous;
} Job;
```

### Estrutura de conexões

Lista que contem todas as ligações desde um job até uma máquina. Este contem como atributos o “id\_job”, “id\_op” (id da operação), “id\_mac” e o “time”. Além disso ainda tem o apontador para o próximo e para o anterior nodo da lista. Este será armazenado em um ficheiro para leitura e escrita.

```
typedef struct list_conection
{
    int id_job, id_op, id_mac, time;
    struct list_conection *next, *previous;
} Connection;
```

### Estrutura das Células de Plano

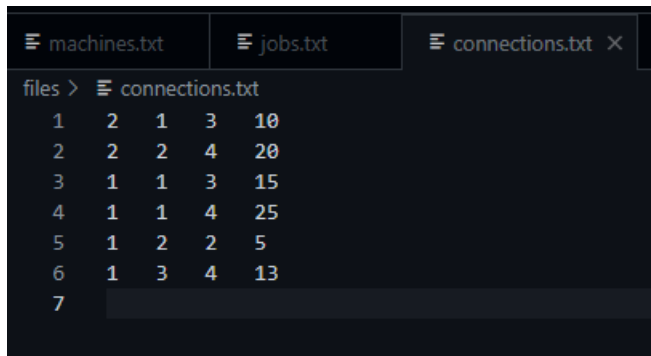
Este representa uma célula do plano/simulação para resolver o problema de escalonamento e este ser apresentado em formato tabelar no seu conjunto. Este guarda como seus atributos, “id\_job”, “id\_op”, “id\_mac” e “t” (instante).

```
typedef struct
{
    int id_job, id_op, id_mac, t;
} CelulaPlano;
```

## Testes realizados

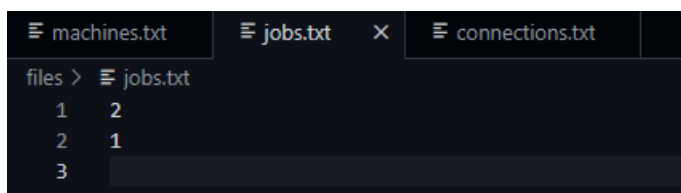
Para a realização destes testes os files encontram-se povoados da seguinte forma:

(id\_job / id\_op / id\_mac / time)



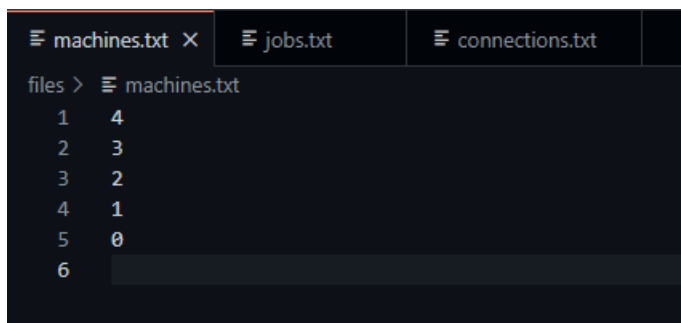
	id_job	id_op	id_mac	time
1	2	1	3	10
2	2	2	4	20
3	1	1	3	15
4	1	1	4	25
5	1	2	2	5
6	1	3	4	13
7				

(id\_job)



	id_job	id_op
1	2	
2	1	
3		

(id\_mac)



	id_mac	id_op
1	4	
2	3	
3	2	
4	1	
5	0	
6		

Seguem-se testes realizados ao código de modo a apresentar a resolução das funcionalidades pretendidas:

## Inserção de um job

Ao seleccionar a opção 2, ele automaticamente adiciona um novo job sem qualquer conexão:

```
-----  
Menu Principal  
[1]-Visualizar Jobs  
[2]-Inserir Job  
[3]-Remover Job  
[4]-Inserir Operacao em Job  
[5]-Remover Operacao em Job  
[6]-Editar Operacao em Job  
[7]-Tempo Minimo Job  
[8]-Tempo Maximo Job  
[9]-Media de Tempo (operacao)  
[0]- Sair  
-----  
-->2
```

Foi adicionado o job 3 sem conexões:

```
-----  
Lista das Jobs  
-----  
ID Job: 3  
ID Job: 1  
ID Job: 2  
-----  
Lista das Maquinas  
-----  
ID Maquina: 0  
ID Maquina: 1  
ID Maquina: 2  
ID Maquina: 3  
ID Maquina: 4  
-----  
Lista Conexoes  
-----  
ID Job: 1 / ID Operacao: 3 / ID Maquina: 4 / Duracao: 13  
ID Job: 1 / ID Operacao: 2 / ID Maquina: 2 / Duracao: 5  
ID Job: 1 / ID Operacao: 1 / ID Maquina: 4 / Duracao: 25  
ID Job: 1 / ID Operacao: 1 / ID Maquina: 3 / Duracao: 15  
ID Job: 2 / ID Operacao: 2 / ID Maquina: 4 / Duracao: 20  
ID Job: 2 / ID Operacao: 1 / ID Maquina: 3 / Duracao: 10  
-----  
Press any key to continue . . .
```



## Remover job

É seleccionado a opção 3 de forma a escolher qual job remover:

```
-----
      Menu Principal
[1]-Visualizar Jobs
[2]-Inserir Job
[3]-Remover Job
[4]-Inserir Operacao em Job
[5]-Remover Operacao em Job
[6]-Editar Operacao em Job
[7]-Tempo Minimo Job
[8]-Tempo Maximo Job
[9]-Media de Tempo (operacao)
[0]- Sair
-----
-->3
```

É seleccionado o job 3, recentemente criado:

```
-----
      Lista das Jobs
-----
ID Job: 3
ID Job: 1
ID Job: 2
-----

Selecione o id a remover: 3
```

Este foi removido como se pode confirmar:

```
-----
      Lista das Jobs
-----
ID Job: 1
ID Job: 2
-----

      Lista das Maquinas
-----
ID Maquina: 0
ID Maquina: 1
ID Maquina: 2
ID Maquina: 3
ID Maquina: 4
-----

      Lista Conexoes
-----
ID Job: 1 / ID Operacao: 3 / ID Maquina: 4 / Duracao: 13
ID Job: 1 / ID Operacao: 2 / ID Maquina: 2 / Duracao: 5
ID Job: 1 / ID Operacao: 1 / ID Maquina: 4 / Duracao: 25
ID Job: 1 / ID Operacao: 1 / ID Maquina: 3 / Duracao: 15
ID Job: 2 / ID Operacao: 2 / ID Maquina: 4 / Duracao: 20
ID Job: 2 / ID Operacao: 1 / ID Maquina: 3 / Duracao: 10
-----
Press any key to continue . . .
```

## Inserção de uma operação em um job

Ao inserir uma operação em um job selecionado, é escolhido qual máquina/s associa-se e após isso é visível a alteração ao nível do terminal e dos ficheiros após terminar o programa.

```
-----
      Menu Principal
[1]-Visualizar Jobs
[2]-Inserir Job
[3]-Remover Job
[4]-Inserir Operacao em Job
[5]-Remover Operacao em Job
[6]-Editar Operacao em Job
[7]-Tempo Minimo Job
[8]-Tempo Maximo Job
[9]-Media de Tempo (operacao)
[0]- Sair
-----
-->4
```

```
-----
      Lista das Jobs
-----
ID Job: 1
ID Job: 2
-----

Selecione o id do job: 1
-----
      Lista das Maquinas
-----
ID Maquina: 0
ID Maquina: 1
ID Maquina: 2
ID Maquina: 3
ID Maquina: 4
-----

(Insira numero negativo para terminar de associar mais maquinas)
Selecione o id da maquina a associar a operacao: 0

Insira o tempo:10
```

```
-----
      Lista das Jobs
-----
ID Job: 1
ID Job: 2
-----

      Lista das Maquinas
-----
ID Maquina: 0
ID Maquina: 1
ID Maquina: 2
ID Maquina: 3
ID Maquina: 4
-----

      Lista Conexoes
-----
ID Job: 1 / ID Operacao: 4 / ID Maquina: 0 / Duracao: 10
ID Job: 1 / ID Operacao: 3 / ID Maquina: 4 / Duracao: 13
ID Job: 1 / ID Operacao: 2 / ID Maquina: 2 / Duracao: 5
ID Job: 1 / ID Operacao: 1 / ID Maquina: 4 / Duracao: 25
ID Job: 1 / ID Operacao: 1 / ID Maquina: 3 / Duracao: 15
ID Job: 2 / ID Operacao: 2 / ID Maquina: 4 / Duracao: 20
ID Job: 2 / ID Operacao: 1 / ID Maquina: 3 / Duracao: 10
-----
Press any key to continue . . .
```

## Alterar operação de um job

É selecionado o que acabou de ser criado, e vai ser alterado o tempo da sua máquina:

```
-----
      Menu Principal
[1]-Visualizar Jobs
[2]-Inserir Job
[3]-Remover Job
[4]-Inserir Operacao em Job
[5]-Remover Operacao em Job
[6]-Editar Operacao em Job
[7]-Tempo Minimo Job
[8]-Tempo Maximo Job
[9]-Media de Tempo (operacao)
[0]- Sair
-----
```

```
-->6
```

```
-----
      Lista das Jobs
-----
ID Job: 1
ID Job: 2
-----

Selecione o id do job: 1
-----
      Lista Intermedia
      (Job 1)
-----
ID Operacao: 4
ID Operacao: 3
ID Operacao: 2
ID Operacao: 1
-----

Selecione o id da operacao:4
Selecione a maquina a ser alterado o seu tempo:
-----
      Lista Intermedia
      (Job 1 / Operacao 4)
-----
ID Maquina: 0 / Time: 10
-----
0
-----
Insira o novo tempo: 20
```

```
-----
      Lista das Jobs
-----
ID Job: 1
ID Job: 2
-----

      Lista das Maquinas
-----
ID Maquina: 0
ID Maquina: 1
ID Maquina: 2
ID Maquina: 3
ID Maquina: 4
-----

      Lista Conexoes
-----
ID Job: 1 / ID Operacao: 4 / ID Maquina: 0 / Duracao: 20
ID Job: 1 / ID Operacao: 3 / ID Maquina: 4 / Duracao: 13
ID Job: 1 / ID Operacao: 2 / ID Maquina: 2 / Duracao: 5
ID Job: 1 / ID Operacao: 1 / ID Maquina: 4 / Duracao: 10
ID Job: 1 / ID Operacao: 1 / ID Maquina: 3 / Duracao: 15
ID Job: 2 / ID Operacao: 2 / ID Maquina: 4 / Duracao: 20
ID Job: 2 / ID Operacao: 1 / ID Maquina: 3 / Duracao: 10
-----
Press any key to continue . . .
```

## Remoção de uma operação de um job

Agora será removido uma operação de um job, neste caso a criada recentemente:

```

-----
      Lista das Jobs
-----
ID Job: 1
ID Job: 2
-----

Selecione o id do job: 1
-----
      Lista Intermedia
      (Job 1)
-----
ID Operacao: 3
ID Operacao: 2
ID Operacao: 1
-----

Selecione o id da operacao a ser removida:2

```

```

-----
      Lista Conexoes
-----
ID Job: 1 / ID Operacao: 2 / ID Maquina: 4 / Duracao: 13
ID Job: 1 / ID Operacao: 1 / ID Maquina: 4 / Duracao: 25
ID Job: 1 / ID Operacao: 1 / ID Maquina: 3 / Duracao: 15
ID Job: 2 / ID Operacao: 2 / ID Maquina: 4 / Duracao: 20
ID Job: 2 / ID Operacao: 1 / ID Maquina: 3 / Duracao: 10
-----
Press any key to continue . . .

```

## Tempo mínimo da realização do job

A funcionalidade que prevê o tempo mínimo para realizar um job testando todas as possibilidades. Para cada id foi escolher a máquina com menor tempo de trabalho e obteve se o resultado seguinte.

```

-----
      Lista das Jobs
-----
ID Job: 1
ID Job: 2
-----

Selecione o id do job: 1

Tempo minimo do job 1: 33
Press any key to continue . . .

```

```

-----
                Lista Conexoes
-----
ID Job: 1 / ID Operacao: 3 / ID Maquina: 4 / Duracao: 13
ID Job: 1 / ID Operacao: 2 / ID Maquina: 2 / Duracao: 5
ID Job: 1 / ID Operacao: 1 / ID Maquina: 4 / Duracao: 25
ID Job: 1 / ID Operacao: 1 / ID Maquina: 3 / Duracao: 15
ID Job: 2 / ID Operacao: 2 / ID Maquina: 4 / Duracao: 20
ID Job: 2 / ID Operacao: 1 / ID Maquina: 3 / Duracao: 10
-----
Press any key to continue . . .

```

Este obtém-se de  $13 + 5 + 15 = 33$  (25 não selecionado pois comparado com o mesmo id de operação do mesmo job, no qual é selecionado o menor que é 15)

### Tempo máximo da realização do job

Mesma lógica do de cima mas para valores maiores:

```

-----
                Lista das Jobs
-----
ID Job: 1
ID Job: 2
-----

Selecione o id do job: 1

Tempo maximo do job 1: 43
Press any key to continue . . .

```

### Tempo médio de uma operação de um job

Para testar a funcionalidade final foi inserido o id 1 devido a este ter mais que uma máquina associada e desta forma obteve-se a média do tempo dessa mesma operação.

```

-----
                Lista das Jobs
-----
ID Job: 1
ID Job: 2
-----

Selecione o id do job: 1
-----
                Lista Intermedia
                (Job 1)
-----
ID Operacao: 3
ID Operacao: 2
ID Operacao: 1
-----

Selecione o id da operacao: 1

Media de tempo (Job 1 / Operacao 1): 20.00
Press any key to continue . . .

```

## Resultado do escalonamento

## Alterações

Relativamente á fase 1, o projeto sofreu algumas alterações, desde melhorias de forma a tornar o código mais coerente e eficiente, a adições de novas funcionalidades e estruturas.

Nota: todas as funcionalidades sofreram alterações de forma a suportar as novas estruturas Jobs e Connections

- O código era um pouco incoerente ao nível de ter a nomenclatura das funções, variáveis, etc de forma diferente, então de forma a evitá-lo foi corrigido;
- Assim como referido já acima as estruturas de dados também sofreram alterações de forma a facilitar certas funcionalidades e por algumas serem desnecessárias ao código atual.
- Funções alteradas:
  - Insert\_operation:
    - Continha um problema de não conseguir inserir mais que uma máquina, então foi corrigido adicionando um **head\_macop = \*list\_macops** no fim de cada **head\_insert** par assim avançar na lista;
    - Foi implementado com o time de forma a ser inserido com a máquina;
  - Change\_operation:
    - Alterado completamente, em que agora este serve para alterar o tempo de uma máquina pertencente a uma operação.
- Todas as funcionalidades agora retornam algum valor ou lista;
- Ocorreu uma correção geral da documentação;
- **IfNDef** aplicado;
- Removidos alguns dos mallocs desnecessários;
- Adicionadas novas condições em algumas das funções;
- Adicionada condição para verificar files vazios;
- Na função min, agora este é inicializado com o primeiro valor time da lista.
- Novo na fase 2:
  - Estruturas
  - Manipulação de ficheiros
  - Adaptação com as novas estruturas
  - Novas funções adaptadas

## Conclusão

Com a conclusão deste projeto foi possível sedimentar todos os elementos aprendidos interna e externamente, assim como o desenvolvimento de código de forma independente e de forma a cumprir alguns dos objetivos pretendidos. Foi possível a melhoria significativa do código além de implementação de novas funcionalidades, levando á melhoria das capacidades algorítmicas.



## Bibliografia

(Luís Ferreira, IV - Structs e Apontadores, 2022)

(Luís Ferreira, VI - Structs e Apontadores, 2022)

(James Aspnes, Notes on Data Structures and Programming Techniques, 2016)