



**FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA**

Departamento de Engenharia Electrotécnica

PROGRAMAÇÃO DE MICROPROCESSADORES

2018/2019

Mestrado em Engenharia Electrotécnica e de Computadores

1ºAno
1ºSemestre

Trabalho Final Fábrica de Automóveis “Vai-se fazendo Lda.”

Regente: Prof. José Barata

Pedro Prates

Francisco Nunes

Marina Gomes

Pedro Rio

Rita Lopes

Simão Gonçalves

1. Introdução

O trabalho final consiste numa sequência de procedimentos de dificuldade crescente, relacionados com a matéria apresentada em Programação de Microprocessadores, cujo objetivo é a gestão de uma fábrica e a sua respectiva produção. O problema envolve a manipulação de todos os dados e operações relacionados com o funcionamento da fábrica, assumindo o papel de administrador da mesma.

Os alunos serão encorajados a seguir um processo de implementação em que a cada etapa se incrementará a complexidade e pela qual se regerá a avaliação do trabalho. As classificações mostradas são apenas indicativas da classificação que o aluno poderá ter, sendo que outras considerações podem fazer com que a nota seja diferente:

- Avaliação do desempenho de cada aluno ao longo do semestre nas aulas de laboratório.
- Outras considerações sobre este próprio trabalho, tais como: qualidade do código; legibilidade; indentação; simplicidade; etc.
- Observações realizadas durante a discussão do trabalho.

2. Fábrica de Automóveis - “Vai-se fazendo Lda.”

O programa pretendido é um software de gestão de uma fábrica de Automóveis chamada “Vai-se fazendo Lda.”. A fábrica é constituída por um conjunto de 5 estações de trabalho automáticas capazes de realizar diferentes operações. A gestão da fábrica consiste numa série de manipulações de dados associadas por um lado às operações de funcionamento da fábrica, como a gestão do stock, de processos e de estações, e por outro lado associadas à parte administrativa da fábrica como visualização de dados estatísticos, como o estado da fábrica, o estado de cada estações (saber quais estão ativas e qual automóvel está em cada uma delas), do stock, mapa da fábrica e respectiva linha de produção. Existe ainda uma função de passar-tempo para que o estado da produção dos automóveis em cada estação seja atualizado. O programa deve começar por ler a informação dos ficheiros guardando-a nas estruturas apropriadas, ao sair do programa, este deve escrever de volta nos ficheiros de forma a garantir a atualização da informação.

Para tornar o trabalho mais interessante vai-se assumir que cada Automóvel chega à fábrica em diferentes fases da sua construção, isto é, cada Automóvel vai precisar de utilizar materiais diferentes, assim como sofrer diferentes operações ao longo da sua produção. As funcionalidades requeridas serão descritas no capítulo seguinte.

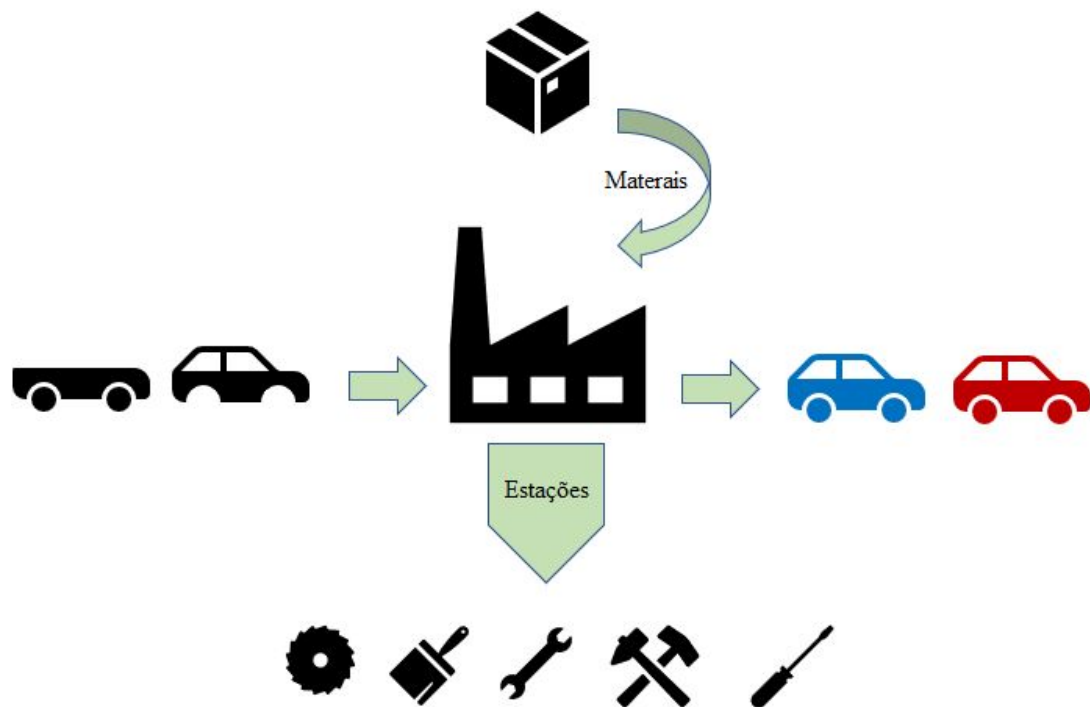


Fig. 1 Visão geral da fábrica. Automóveis em diferentes estados da sua produção chegam. Os materiais são usados nas estações automáticas, para completar a produção de cada automóvel.

```
***Factory Admin Menu***

1 - Show stock
2 - Refill stock
3 - Build car
4 - Factory State
5 - Stats
6 - Time skip
E - Exit

Option: 
```

Fig.2 Exemplo do menu principal.

3. Descrição do Trabalho

3.1. Leitura do ficheiro *stock.txt*.

A fábrica possui uma base de dados contendo o registo de todos os materiais disponíveis em stock. Cada material é constituído por um ID, Tipo e Preço.

Esta base de dados está contida no ficheiro de texto *stock.txt* no qual cada linha diz respeito a uma única unidade de material respeitando o seguinte formato:

ID – Tipo – Preço

```
1 Ferro 1.40
1 Ferro 1.40
1 Ferro 1.40
4 Plastico 2.20
2 Alumínio 7.80
3 Vidro 12.50
4 Plastico 2.20
2 Alumínio 7.80
1 Ferro 1.40
1 Ferro 1.40
4 Plastico 2.20
5 Borracha 1.5
6 Aço 3.30
```

O **ID** de cada material, representa o seu código interno na fábrica, que é único. O **Tipo**, tal como o próprio nome indica, diferencia o tipo de material, simultaneamente servindo como descrição; O **Preço** representa o custo de cada unidade de material. O ficheiro *stock.txt* deve ser lido para uma estrutura apropriada que contabilize para cada tipo de material a quantidade disponível.

A partir da informação lida deve ser apresentada uma lista com todos os materiais, quantidades disponíveis de cada e ainda o valor total em euros das mesmas. Deve ainda existir uma opção para a partir de um ficheiro, cujo o nome é pedido ao administrador, repor o stock aumentando as quantidades de material disponível.

3.2. Produção de Automóveis

A fábrica produz automóveis para 5 marcas distintas. Tal como referido anteriormente, cada marca entrega à fábrica os seus automóveis em diferentes estado de produção, o que significa que cada automóvel necessita, tanto de quantidades de material diferentes como de passar por operações distintas ao longo da sua produção. Quando o administrador coloca um novo automóvel em produção, o programa deve ser capaz de verificar se existem materiais suficientes em stock, e caso não haja apresentar uma mensagem de aviso.

	Ferro	Alúminio	Vidro	Plástico	Borracha	Aço
Renault	2	0	1	3	0	0
Mercedes	0	3	2	0	0	2
Honda	4	0	1	3	2	0
Ford	0	1	2	3	1	3
Ferrari	2	2	3	0	0	2

Tabela 1. Lista de quantidades de materiais que cada **Tipo** de automóvel precisa.

3.3. Ficheiro de Processos (*processing.txt*).

O ficheiro de processos contém a informação sobre todos o automóveis, que já foram ou ainda estão a ser produzidos. Cada linha representa um determinado automóvel num certo estado de produção. O ficheiro possui a seguinte estrutura:

Tipo - Estado - Local - Tempo Restante - Operações Restantes

```
Ferrari Finished - 0 ---
Renault Finished - 0 ---
Ford Processing 2 10 -BD
Mercedes Processing 3 20 ACE
Mercedes Processing 1 20 --E
Honda Processing 4 40 BCE
Ford Waiting - 0 ABD
```

O **Tipo** define qual o tipo de automóvel (marca) em questão; O **Estado** diz em que fase de produção está, tendo 3 valores possíveis (*Waiting, Processing, Finished*); **Local** é o ID da estação em que se encontra no momento, '-' caso não esteja em nenhuma; **Tempo Restante** indica o tempo que o automóvel ainda tem de estar na estação indicada pelo **Local**, é 0 caso não esteja em nenhuma estação; **Operações Restantes** lista os IDs das operações que o automóvel ainda tem de sofrer para ser dado como terminado.

A título de exemplo: A 3ª linha do ficheiro, representa um automóvel da marca Ford, que se encontra a ser processado na máquina Nº2. Faltam 10 minutos para a operação corrente acabar, sendo que essa operação é do tipo B. De seguida vai ser ainda necessário realizar a operação D para terminar a sua construção.

A informação presente no ficheiro *processing.txt* deve ser lida e guardada numa estrutura apropriada. Esta estrutura vai estar no centro da lógica da fábrica uma vez que a partir dela é possível saber que automóveis estão em produção, que estações estão ocupadas, quanto tempo resta para as operações correntes terminarem, etc.

Cada vez que um novo automóvel é colocado em produção uma nova entrada deve ser feita na estrutura.

	Soldar	Cortar	Pintar	Montar	Acabamentos
Renault	2º	1º		3º	
Mercedes	1º		3º		2º
Honda		1º	2º		3º
Ford	1º	2º		3º	
Ferrari			2º	1º	3º

Tabela 2. Operações que cada **Tipo** de automóvel tem que realizar e a sua ordem.

3.4. Leitura de ficheiro binário (*stations.dat*).

Na fábrica existem 5 estações distintas cada uma capaz de executar um conjunto distinto de operações. Cada operação demora um certo tempo a ser executada, dependendo da estação e cada minuto que uma estação trabalha tem um determinado custo para a fábrica. Cada estação apenas consegue executar uma operação a um automóvel de cada vez.

A informação das estações está contida num ficheiro binário (*stations.dat*) com a seguinte estrutura:

- 1 inteiro para o ID da estações (1-5).
- 3 caracteres para as operações que a estação realiza (A-E).
- 3 inteiros para o tempo que cada operação demora.
- 1 double com o custo de utilização da estação por minuto.
- 1 inteiro com o acumulado de tempo de execução da estação.

O ficheiro deve ser lido para uma estrutura apropriada no início da execução do programa. Ao terminar o programa, o tempo que cada máquina esteve em execução deve ser atualizado, escrevendo-o de volta no ficheiro.

As operações estão codificadas no ficheiro binário (*stations.dat*) segundo a seguinte tabela:

	Soldar	Cortar	Pintar	Montar	Acabamentos
ID	A	B	C	D	E

Tabela. 3 Codificação de cada operação, no ficheiro *stations.dat*

3.5. Funcionamento geral da fábrica

O administrador têm dois modos de interagir com o seu funcionamento:

- Colocar um automóvel em produção.
- Fazer passar o tempo.

3.5.1. Automóvel em produção

Quando o administrador escolhe a opção de produzir um automóvel deve-lhe ser apresentada a lista de opções (cada marca). Depois de escolher o automóvel a produzir, o programa deve realizar os seguintes passos:

- Verificar se existem materiais suficientes. Caso não haja, o programa deve emitir uma mensagem de erro e voltar ao menu anterior. Caso hajam, deve ser criada uma nova entrada na estrutura de processos referida na secção 3.3.

- Devem ser verificadas as operações necessárias consoante o **Tipo** e colocadas na entrada da estrutura. O automóvel passa assim a um estado de “*Waiting*”.
- Verificar quais as estações não ocupadas que realizam a 1ª operação necessária. Escolher a estação que terá um custo menor (tempo x custo/min). Atualizar a entrada na estrutura com o local correcto e passando o estado do automóvel a “*Processing*”.
- Caso não haja nenhuma estação disponível, o local será representado por ‘-’ e o automóvel continuará em “*Waiting*”.

3.5.2. Passagem do tempo

A opção de passagem do tempo deve perguntar ao administrador quanto tempo, em minutos, quer passar. O tempo restante de cada um dos processos em “*Processing*” deve ser atualizado e o tempo de utilização das estações deve ser adicionado na estrutura das Estações. Caso o tempo restante de alguma operação chegue a 0, o processo correspondente deve passar ao estado de “*Waiting*” e verificar qual a próxima operação e estação disponível fazendo as atualizações necessárias, tal como quando se coloca um novo automóvel em produção. Caso não haja mais nenhuma operação a ser realizada, deve passar ao estado de “*Finished*”, finalizando a sua produção.

O programa deve ainda aproveitar o tempo ao máximo tendo em conta os tempos de término das operações. Ex: O administrador manda passar 30 minutos, mas existe um automóvel a realizar uma operação que acaba em 10 mins. Deve ser considerado que passam 10 mins apenas, verificar o próximo destino do automóvel que acabou o seu processamento e ainda verificar se algum dos automóveis em estado de “*Waiting*” pode prosseguir com a sua produção. De seguida devem passar os restantes 20 minutos verificando mais uma vez se algum processo acaba num tempo menor.

3.6. Estado da fábrica

A opção de estado de fábrica deve apresentar as seguintes opções:

- Listar todos os processos: Listagem de toda a informação contida na estrutura de processos.
- Listar processos por tipo: Listagem de toda a informação contida na estrutura de processos relativa a automóveis de um determinado tipo, pedido ao administrador.
- Listar processos por estado: Listagem de toda a informação contida na estrutura de processos relativa a automóveis num determinado estado de produção, pedido ao administrador.
- Estado das estações: Listagem de todas as estações, indicado quais se encontram a trabalhar. E caso estejam a trabalhar que tipo de automóvel estão a processar e qual a operação que estão a realizar.

3.7. Estatística

A opção de estado de fábrica deve apresentar as seguintes opções:

- Histograma com a quantidade de cada tipo de automóvel já terminado, em percentagem.
- Histograma com número de processos em cada um dos estados (número absoluto).
- Contagem do tempo que cada estação esteve em funcionamento, custo acumulado de cada uma e o custo total da fábrica.

```
****State Chart****  
  
Processing:      *****  
Waiting:         *****  
Finished:        **
```

Fig. 3 Exemplo de um histograma com o número de processos em cada estado.

3.8. Mapa

Deve existir ainda um mapa simples representativo da fábrica, com as estações em funcionamento assinaladas com as operações que estão a executar naquele momento.

1	2	3	4	5
---	---	---	---	---
A	-	E	-	C
---	---	---	---	---

Fig. 4 Exemplo de mapa da fábrica. Cada letra representa a operação que a estação está a realizar.

4. Avaliação

Como foi especificado na introdução, o trabalho está dividido de maneira a ser possível percorrer um percurso de etapas que vão ser de complexidade crescente e que aconselhamos a serem seguidas:

0-10 Valores: O programa deve ser capaz de ler, processar e carregar a informação que se encontra no ficheiro referente ao stock inicial de materiais com recurso ao ficheiro fornecido pelos docentes (*stock.txt*). A operação referente à listagem de todos os materiais existentes em stock deve estar totalmente completa e funcional.

A criação de um novo automóvel deve estar parcialmente implementada, sendo necessário conseguir verificar se há materiais que chegue para fazer o automóvel e retirar os materiais de disponibilidade (de maneira a não usar o mesmo material na criação de outro automóvel).

Ao sair do programa este deve escrever no ficheiro *stock.txt* fazendo a atualização dos materiais disponíveis.

Deve ainda existir uma opção para a partir de um ficheiro, cujo o nome é pedido ao utilizador, adicionar stock aumentando as quantidades de materiais disponível.

10-12 Valores: O programa deve ser capaz de ler, processar e carregar a informação que se encontra no ficheiro referente aos automóveis em produção com recurso ao ficheiro fornecido pelos docentes (*processing.txt*).

Deve ser possível ver uma lista de todos os processos presentes assim como poder escolher para apresentar, só os processos de um determinado tipo de automóvel ou processos num determinado estado.

Ao mandar produzir num novo automóvel o programa deve acrescentar à lista de processos um novo processo automaticamente com o estado “Finished”.

Ao sair do programa o ficheiro *processing.txt* deve ser atualizado com os processos adicionados ao longo da sua execução.

12-14 Valores: O programa deve ser capaz de ler, processar e carregar a informação que se encontra no ficheiro referente às estações existentes na fábrica com recurso ao ficheiro fornecido pelos docentes (*stations.dat*).

Ao produzir um novo automóvel este deve ser colocado numa estação livre que realize a 1ª tarefa necessário à sua produção, passando o seu estado a “Processing”. A função “passar tempo” deve imediatamente acabar todas as operações em curso movendo os automóveis para a seguinte estação na sua produção, sem preocupações temporais. Ao acabar todas as tarefas o automóvel deve ser dado como “Finished”. Ao sair do programa o ficheiro *station.dat* deve ser atualizado com o tempo que cada estação esteve em execução.

14-16 Valores: O programa deve ter a opção de apresentar o estado de cada estação, “Processing ou “Idle” e caso esteja em funcionamento qual a operação que está a realizar.

O programa deve agora passar o tempo e ter em conta o tempo restante de cada operação. Apenas terminando a execução de uma operação quando tempo suficiente tiver passado e continuando com a execução das operações que necessitem de mais tempo na estação em que se encontram. Nas estatísticas deve ser possível ver quanto cada estação já custou à fábrica.

Devem ainda ser apresentados histogramas representativos das quantidades de automóveis terminados por marca e da quantidade de processos em cada estado.

16-18 Valores: A escolha das estações para cada produto deve agora ter atenção a relação custo por minuto de operação, escolhendo sempre a estação, que realize a operação necessária, com o menor custo.

Deve existir a opção de apresentar um mapa da fábrica sendo que este deve representar o estado de cada estação visualmente.

18-19 Valores: Aproveitamento total do tempo na função “passar tempo”, tendo em conta os tempos de término das operações (secção 3.5.2).

20 Valores: O trabalho deve ser feito utilizando as melhores práticas de programação. Código genérico e flexível capaz de ser escalável caso seja necessário. Existência de funcionalidades extras e ainda usar apontadores sempre que faça sentido.

5. Postura dos Alunos

Cada grupo deve ter em consideração o seguinte:

- Não perca tempo com a estética de entrada e saída de dados
- Programe de acordo com os princípios gerais de uma boa codificação (utilização de indentação, apresentação de comentários, uso de variáveis com nomes conformes às suas funções...) .
- Proceda de modo a que o trabalho a fazer fique equitativamente distribuído pelos dois membros do grupo.
- Vá fazendo vários *backups* ao longo do desenvolvimento do programa de forma a evitar contratempos.

6. Datas Limites

A data limite para entrega do trabalho de avaliação é no dia **16 de Dezembro de 2018**, às **23:59**. Não são permitidas entregas posteriores, sendo a nota do trabalho de zero valores. Tenha especial atenção a esta data de 16 de Dezembro por causa da pressão dos testes e outros trabalhos de outras disciplinas. Adiante o trabalho para não sofrer muita pressão no final do prazo. O trabalho deve ser entregue, num arquivo .rar/.zip com o ficheiro de código C, pelo correio eletrónico do seu docente das aulas práticas. O nome do ficheiro deve indicar o turno do grupo e os números dos alunos da seguinte forma: **PX_XXXXX_XXXXX.rar**. Ex. P1_39757_39758.

As discussões dos trabalhos realizar-se-ão na semana seguinte, tanto quanto possível.