



# Informática Industrial

Mestrado Integrado em Engenharia Eletrotecnia e de Computadores

2019/2020

## *Automação de Linha de Produção*

## 1 Apresentação

Este documento descreve o trabalho que irá realizar ao longo da disciplina de Informática Industrial. De forma simplificada, consiste:

**Requisito 1:** na automatização do simulador de uma linha de produção flexível,

**Requisito 2:** desenvolvimento de uma plataforma de controlo e monitorização do mesmo (i.e. um MES – Manufacturing Execution System).

Tipicamente o controlo do nível mais baixo (requisito 1) irá ser alcançado com recurso a PLCs ou equipamento equivalente. Para o controlo do nível imediatamente superior (requisito 2, MES) aconselha-se a utilização de ferramentas de programação mais versátil (C++, Java, Python, C#, ...), sendo ainda obrigatório a utilização de uma base de dados gerida por um sistema de gestão de base de dados (PostgreSQL, MariaDB, ...).

A descrição dos equipamentos desta linha, e a forma de interagir com eles, é feita num documento anexo. Neste documento será apresentado apenas os requisitos que os sistemas de controlo e monitorização deverão cumprir. Para compreender estes requisitos é fundamental conhecer os equipamentos sob controlo, pelo que se sugere a leitura prévia da descrição da linha de produção.

## 2 Descrição da Linha de Produção Flexível

A Linha de Produção Flexível que deverá automatizar é composta por 8 células: 1 Armazém automático, 3 células de máquinação (com 3 maquina cada), 3 células de carga e descara das células de máquinação, e 1 célula para carga de matéria prima e descarga de peças processadas.

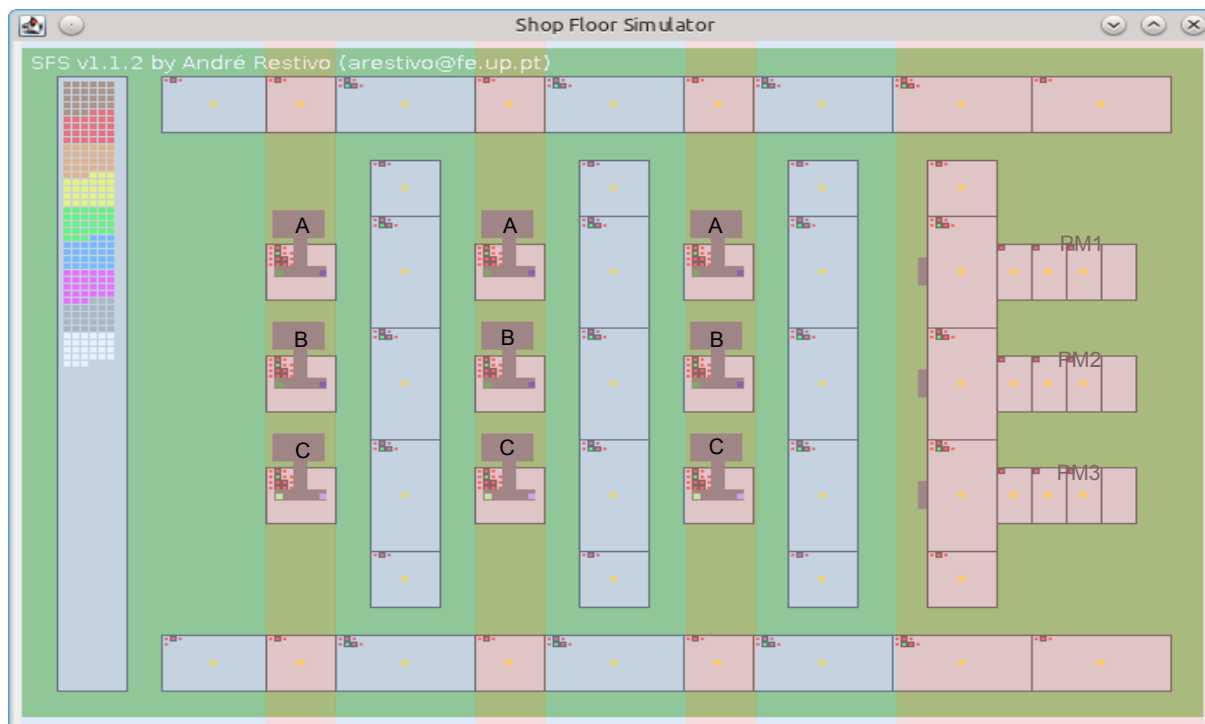


Figura 1: Linha de produção flexível – disposição dos equipamentos.

Esta linha de produção será simulada pelo software SFS (Shop Floor Simulator) o qual, sendo desenvolvido em Java, é compatível com Windows, Linux e OSX. O acesso aos sinais lógicos (sensores e atuadores) é feito através do protocolo Modbus/TCP. O simulador SFS implementa um servidor Modbus/TCP, no qual os sensores (sinais booleanos) são mapeados como Input Discretos, e os atuadores (sinais booleanos) como Coils. Os endereços de cada sinal são disponibilizados num ficheiro io.csv (formato comma separated values, que pode ser aberto como folha de cálculo). Este ficheiro é gerado pelo próprio simulador com o comando “java -jar sfs.jar –csv io.csv”.

A descrição dos sinais disponíveis para controlar cada equipamento (máquina, tapete linear, tapete rotativo, pusher e armazém) encontra-se descrito noutro documento.

## 2.1 O Processo de Transformação

A linha de produção processa tipos diferentes de peças (**P1**, **P2**, ...), representadas com cores diferentes. A convenção de cores utilizada (igual à convenção de cores das resistências elétricas) está definida na tabela seguinte.

Peça	Cor
P1	Castanho
P2	Vermelho
P3	Laranja
P4	Amarelo
P5	Verde
P6	Azul
P7	Violeta
P8	Cinza
P9	Branco

*Tabela 1: Convenção de cores e tipos de peças.*

Ao processar uma peça de determinado tipo numa das máquinas, utilizando uma dada ferramenta, por um certo período de tempo, esta peça muda de tipo.

Cada máquina dispõe internamente de um armazém de três ferramentas (**T1**, **T2**, e **T3**), entre as quais poderá alternar no sentido de produzir diferentes peças de acordo com a matéria-prima fornecida. As transformações de peças que poderão ocorrer, e que depende do tipo de máquina e do tipo de ferramenta, está descrita na tabela seguinte:

Matéria Prima	Peça final	Tipo de Máquina	Ferramenta	Tempo Processamento
P1	P2	Ma	T1	15 s
P2	P3	Ma	T1	15 s
P2	P6	Ma	T2	15 s
P6	P9	Ma	T3	15 s
P1	P3	Mb	T1	20 s
P3	P4	Mb	T1	15 s
P3	P7	Mb	T2	20 s
P7	P9	Mb	T3	20 s
P1	P4	Mc	T1	10 s
P4	P5	Mc	T1	30 s
P4	P8	Mc	T2	10 s
P8	P9	Mc	T3	10 s

*Tabela 2: Sequências de transformações possíveis*

Se tentar efetuar operações que não se encontram na tabela acima (por exemplo: processar **P8**, na máquina **Ma**, com ferramenta **T3**) a peça ficará inutilizada, passando a ficar representada com a cor preta. De forma semelhante se efetuar uma operação descrita na tabela acima, mas por um período de tempo superior ao indicado, a peça fica também inutilizada. Se começar a efetuar uma operação de transformação, mas parar antes de decorrido o tempo necessário, a transformação fica incompleta, ficando a peça com as duas cores (da peça original, e da peça final) na proporção correta.

Para se produzir uma peça final é necessário ter em consideração a sequência de processamentos das matérias-primas. Esta sequência resulta da combinação sucessiva entre matéria-prima e ferramenta, que poderão gerar subprodutos para posterior processamento. Desta forma, caso se deseje produzir uma peça **P5** a partir de uma outra peça **P3** no armazém, será necessário aplicar **T1** na máquina **Mb** (sobre a peça **P3**) resultando num subproduto do tipo **P4**, seguido de **T1** na máquina **Mc** resultando em **P5**.

A linha de produção inicia com 54 peças de cada tipo no armazém automático.

## 2.2 O Processo de Carga e Descarga

A célula mais à direita é utilizada para a carga e descarga de peças. A **descarga** efectua-se por um dos 3 postos de descarga (PM1, PM2 e PM3). A **carga** efectua-se por um dos dois tapetes extremos das linhas de transporte superior e inferior (i.e. tapetes com identificador C7T1b e C7T7b).

Os postos de carga funcionam como tapetes normais sobre os quais serão manualmente depositadas peças, as quais deverão ser transportadas para o armazém.

Cada posto de descarga é composto por um *pusher* e um *slider*. O slider é uma rampa inclinada sobre a qual cabem um máximo de 4 peças. O *pusher* é um mecanismo que empurra a peça do tapete alinhado com o *slider*, para o *slider* respectivo. As peças serão retiradas manualmente do extremo direito de cada slider. Cada posição do slider (excepto a última, mais à direita) dispõe de um sensor de presença de peça.

## 3 Controlo dos equipamentos de produção

Para o seu trabalho deverá desenvolver o software de controlo dos equipamentos que fazem parte das células da linha de produção acima descrita (requisito 1 mencionado na secção §1). No entanto terá ainda de desenvolver o MES que supervisiona e monitoriza a linha de produção (requisito 2).

Aconselha-se vivamente que o controlo do nível inferior não seja feito de forma holística, mas antes de forma modular. Sugere-se que comece por organizar/dividir a linha de produção em células de controlo – células essas que deverão conseguir operar de forma o mais independente possível das restantes células. Note que as células que definir para o seu controlo não têm que coincidir com a divisão em células que foi feita aquando da descrição da linha de produção (figura 1).

Deverá de seguida definir a forma como as duas componentes de controlo irão interagir. Em termos tecnológicos deverá adotar a tecnologia de comunicação **OPC-UA**.

## 4 Serviços da Linha de Produção Flexível

Assume-se que a empresa dispõe de software de gestão de recursos (ex. ERP – Enterprise Resource Planning, MRP – Material Requirements Planning, ...) que toma a decisão das ordens de produção que devem ser executadas, e que serão comunicadas aos MES que irá desenvolver. Assim, o seu MES deverá aceitar e executar os comandos que recebe do ERP.

Caberá ao seu sistema de controlo gerir as existências de peças no armazem. Por vezes o ERP irá perguntar ao seu software de controlo qual o numero de peças no armazem naquele momento.

### 4.1 Ordens

O MES irá receber as ordens do ERP codificadas em ficheiros XML, os quais serão enviados para a porta 54321 do protocolo UDP/IP. Cada ficheiro poderá incluir uma ou mais ordens de fabrico, codificadas de acordo com os exemplos seguintes.

#### – Transformação de peças:

O ficheiro XML irá conter:

**<Order Number="nnn">**

**<Transform From="Px" To="Py" Quantity="XX" MaxDelay="YY"/>**

**</Order>**

Em que:

**nnn** – número de ordem

**Px** – tipo de peça de origem

**Py** – tipo de peça final

**XX** – quantidade a produzir

**YY** – atraso máximo (em segundos) para terminar de executar esta ordem

Ao receber uma ordem de transformação, o seu sistema de controlo deverá transformar **XX** peças do tipo **Px** (peças origem, sendo x um número de 1 a 9) em peças do tipo **Py** (peças finais, com y de 1 a 9). As peças iniciais (finais) terão sempre como origem (destino) o armazém.

É possível que receba ordens de transformação para as quais não exista no armazém peças **Px** em quantidade suficiente para completar a totalidade do pedido. Nestes casos deverá começar por executar a ordem de transformação até esgotar as peças **Px** no armazem, ficando as restantes peças da ordem de transformação em suspenso até que possam ser executadas.

Sendo assim, caberá ao MES (que irá implementar) gerir a existência de peças no armazém. Em contra-partida, pode assumir que haverá sempre lugares disponíveis no armazém para armazenar as peças **Py** produzidas.

A ordem indica também o tempo máximo de termino da ordem. Este tempo é contado a partir do momento em que a ordem é entregue (i.e. um tempo relativo).

#### – Descarga de peças

O ficheiro XML irá conter:

```
<Order Number="nnn">  
<Unload Type="Px" Destination="Dy" Quantity="XX"/>  
</Order>
```

Em que:

**nnn** – número de ordem  
**Px** – tipo de peça a descarregar  
**Py** – Destino / Local de descarga  
**XX** – quantidade de peças a descarregar

Indica um pedido para depositar no local de descarga **Dy** indicado (y=1 → PM1, y=2 → PM2, y=3 → PM3), **XX** exemplares de peças do tipo **Px**. As peças depositadas serão posteriormente removidas manualmente. Caso o local de descarga fique completo, e o operador se atrasar a retirar as peças, as restantes peças do pedido deverão permanecer nos tapetes (caso já tenham sido retiradas do armazém) ou no armazém, até que haja novamente disponibilidade no local de destino;

#### – Existência de peças

O ERP irá perguntar ao seu software qual o numero de peças de cada tipo existentes naquele momento no armazem. Para tal irá enviar um ficheiro XML que inclui:

```
<Request_Stores/>
```

O seu software deverá responder, para o mesmo IP/porto que enviou o pedido, com um ficheiro XML que deverá conter:

```
<Current_Stores>  
<WorkPiece type="Px" quantity="XX"/>  
<WorkPiece type="Px" quantity="XX"/>  
...  
</Current_Stores>
```

Em que:

**Px** – tipo de peça  
**XX** – quantidade de peças no armazem

#### – Carga de peças

O seu sistema de controlo deverá armazenar no armazém todas as peças que sejam depositadas em qualquer um dos dois locais de carga (i.e. os dois tapetes mais à direita). Esta ordem deverá ser despoletada pela presença de uma peça em qualquer um destes tapetes.

As peças depositadas no tapete imediatamente acima de PM1 serão sempre do tipo P1. As peças depositadas no tapete imediatamente abaixo de PM3 serão sempre do tipo P2.

Todo os pedidos acima descritos deverão ser tratados e processados em simultâneo, desde que a disposição da fábrica e os recursos disponíveis o permitam. Isto não quer dizer que a conclusão das ordens seja a sempre a mesma pela qual estas deram entrada (por exemplo,

uma ordem que requeira apenas as máquinas A poderá acabar antes de outra ordem que recorra apenas à máquina B, embora esta última tenha chegado primeiro).

Deverá tentar dar prioridade às ordens que tenham prazo de entrega mais curto. O prazo das ordens que não são de transformação (carga, descarga, ...) deverá ser considerado como sendo imediato (i.e., prazo relativo = 0s). Os prazos deverão ser considerados como objetivos. Por vezes, devido ao excesso de ordens, poderá não ser possível cumprir todos os prazos.

## 4.2 Estatísticas:

Para permitir aos utilizadores acompanhar o processo produtivo, o MES deverá disponibilizar a seguinte informação:

### – Lista de Ordens

Uma lista das ordens pendentes, em execução, ou já executadas. Para cada uma destas ordens, deverá ser disponibilizado

- o número da ordem (de produção, montagem ou descarga)
- estado atual da ordem (pendente, em produção, ou completa)
- o número de peças já produzidas/montadas/descarregadas,
- o número de peças em produção/montagem/descarga,
- o número de peças ainda pendentes,
- a hora de entrada da ordem,
- a hora de início da sua execução,
- a hora do fim da sua execução.
- a folga atual na sua execução  
(i.e., para ordens em execução, quanto tempo falta para o prazo de entrega terminar, e para ordens já completas o tempo que faltava para o prazo no momento que a ordem terminou).

### – Estatística das Máquinas

Uma lista com a estatística da operação de cada máquina. Para cada máquina, disponibilizar:

- tempo total de operação,
- número de peças operadas (total e por cada tipo).

### – Peças Descarregadas

Para cada zona de descarga, disponibilizar:

- o número de peças descarregadas (total e por tipo).

Não se encontra especificada a interface através da qual é feita a monitorização, pelo que está livre de a implementar como achar mais conveniente. Se desejar, poderá estender o protocolo de comunicação por forma a que sejam disponibilizadas em formato XML no porto UDP.

## 4.3 Persistência

A informação gerida pelo MES deverá ser persistente, i.e. não deverá ser perdida caso o MES tenha de ser desligado. Este requisito deverá ser atingido recorrendo a uma base de dados gerida por um sistema de gestão de bases de dados (e.x.: MariaDB, MySQL, PostgreSQL, ...).

A informação a colocar na base de dados deverá ser tal de forma a permitir que o MES seja desligado e retomado mais tarde sem que haja perda de informação relativa a ordens pendentes, nem a estatísticas elaboradas até essa altura. Durante o período em que o MES se encontra desligado é aceitável que se deixe de aceitar novas ordens vindas do ERP. Com o MES desligado as operações de produção poderão também vir a ser afetadas, sendo que o nível em que são afetadas irá depender da forma como está organizada a arquitectura do sistema de produção.

## 5 Resultados Esperados

Este projeto deverá ser executado por grupos de 4 alunos.

Toda a lógica de funcionamento por detrás do controlo da linha de produção deverá ser analisada, projetada e implementada pelos alunos, que deverão identificar, para além dos componentes descritos anteriormente, todos os componentes que serão necessários ao funcionamento da linha. Recomenda-se que comece primeiro por não considerar a possível otimização de utilização dos recursos disponíveis. A otimização de recursos não deixará no entanto de ser valorizada aquando da avaliação do projeto.

### Os resultados esperados do trabalho são:

1. **Modelo da arquitetura da aplicação**, composto, no mínimo, pelos diagramas UML de classes, objetos, e sequência. A ser entregue até **15 de Março 2020**;
2. **Relatório Final** (50% da nota), a entregar até **14 de Junho 2020**.
3. **Demonstração** e discussão da aplicação (50% da nota), na semana de **25 a 29 de Maio 2020**.

### Distribuição de Esforço Entre Elementos do Grupo:

As notas dos trabalhos serão atribuídas ao grupo. O relatório deverá indicar como o grupo deseja que esta nota seja distribuída por cada elemento do grupo. Esta distribuição deverá ser feita em forma de percentagem de tal forma que a soma das percentagens seja 100.

Calculo da nota final individual	
Nota Final do Grupo: G (0..20)	$A = \text{Max}[0; \sin(\pi/2 * (Px/P - 1) / (1/P - 1))]$
Pesos dos elementos do Grupo: P1, P2, P3, P4	$B = \text{Min}[0; (Px/P - 1)]$
$P = \text{Average}(P1, P2, P3, P4)$	$C = (20-G)*G/20 * A$
Nota Individual $Gx = G + C + D$	$D = G * B$

### Relatório:

O relatório deverá ser escrito em letra *arial* com tamanho 11pt e espaçamento entre linhas de 1.5 linha, e uma só coluna, com 2cm de margem a toda a volta de paginas A4.

O relatório deverá ainda seguir a seguinte estrutura:



- **(2%) Introdução (até meia página de texto)**

Resumo da abordagem seguida com identificação do problema, tecnologias usadas, algumas características especiais da vossa solução, alguns resultados que queiram salientar e estrutura do relatório. Para isto caber em meia página, terão que ser mesmo sucintos. Aqui interessa apenas dar uma ideia rápida do que fizeram e que vão mostrar a seguir.

- **(7%) Estrutura do código (até 1,5 página de texto, excluindo figuras)**

Descrição da estrutura (estática) do código utilizando diagramas de classes e/ou objetos. O texto deve salientar os componentes principais do código e fazer referência aos diagramas respetivos.

- **(7%) Funcionamento do código (até 1,5 página de texto)**

Descrição da interação (dinâmica) entre os elementos do código utilizando diagramas de interação, de atividades e/ou de estado. Mais uma vez deverá usar o texto para explicar algumas interações mais elaboradas ou alguns pormenores que queiram salientar, fazendo sempre referência aos diagramas respetivos.

- **(7%) Implementação (até 1,5 página de texto)**

Justificar a escolha das tecnologias usadas, descrever a forma utilizada para mapear classes em cada tecnologia, descrever a arquitetura de alto nível do programa (ciclo infinito vs threads, etc...), explicar como fazem invocação de métodos entre MES e PLC (variáveis Modbus, etc...)

- **(5%) Estrutura final vs inicial (1,5 página de texto)**

Discutir as principais alterações introduzidas entre a solução proposta inicialmente e a que foi implementada.

- **(7%) Comparação com Padrões Existentes (até 1,5 página de texto)**

Comparação crítica da arquitectura utilizada no projeto com os padrões (design patterns) propostos pela norma ISA 95, o RAMI 4.0, e o artigo "Service Granularity in Industrial Automation and Control Systems".

- **(5%) Organização da equipa (até 1 página de texto)**

Descrever a atribuição de tarefas dentro da equipa, o modelo de desenvolvimento, comentários críticos ao desempenho dos elementos da equipa e eventuais alterações realizadas na atribuição de tarefas. Indicar como o grupo deseja que a nota seja distribuída por cada elemento do grupo

- **(5%) Testes e resultados (1 página de texto)**

Descrever os testes efetuados, em particular que situações complexas foram testadas. O desempenho também pode ser quantificado (e.g., a ordem XXXX demorou xx segundos a ser processada; a sequência de ordens X, Y e Z demorou yy segundos...). Apreciação crítica destes resultados.

- (5%) **Conclusões** (1 página de texto)

Resumo do que foi dito no relatório, salientando alguns resultados considerados mais relevantes.

## 6 Competições

Serão lançadas duas competições (A e B), cada uma das quais será ganha pelo grupo que conseguir completar uma determinada sequência de ordens de fabrico mais rapidamente. O grupo que ganhar cada competição terá um bónus de 1 valor na nota pratica. Se a competição for ganha pelo mesmo grupo a bonificação acumulada poderá ser reduzida a 1.5 valores (decisão final ao critério do docente).

Para a competição A a sequência de ordens de fabrico que será utilizada na competição é fornecida no início do projecto. Para a competição B a sequência de ordens de fabrico será entregue após o prazo de entrega do projecto. Ambas as sequências deverão ser executadas pelo mesmo algoritmo/programa de controlo desenvolvido e submetido pelo grupo.

Sendo assim poderão ter em conta a sequência A aquando do desenvolvimento dos algoritmos de otimização. Têm no entanto de ter em conta de que estes mesmos algoritmos terão de executar a sequência B, desconhecida durante o desenvolvimento do projeto.

As sequencias são disponibilizadas na forma de *script* por forma a que as ordens sejam lançadas a instantes pré-determinados (e não todas de uma vez no arranque do programa).

Os grupos interessados em competir deverão incluir no **início** do seu relatório o tempo total de processamento de cada sequência. O(s) grupo(s) vencedor(es) terá(ão) ainda de demonstrar ao docente a execução das ordens ao vivo (i.e. com o software de controlo a correr).