

**UNIVERSIDADE AUTÓNOMA DE LISBOA LUÍS DE CAMÕES**  
**Departamento de Ciências e Tecnologias**  
**Paradigmas de Programação 2020/2021**  
**Projeto**

## Conteúdo

<b>Datas relevantes</b>	<b>1</b>
<b>Descrição</b>	<b>2</b>
Distribuição k-Erlang . . . . .	2
<b>Objetivo</b>	<b>3</b>
Regras . . . . .	3
<b>Entrega</b>	<b>3</b>
Relatório . . . . .	3
<b>Prova de autoria</b>	<b>3</b>
<b>Avaliação</b>	<b>4</b>

## Datas relevantes

Data	Evento
2021/05/03	Disponibilização do enunciado
2021/06/27, às 23:59:59 GMT	Entrega final do trabalho
De 2021/06/28 a 2021/07/06	Provas de autoria

## Descrição

Uma fábrica de automóveis pretende estudar a eficácia das suas linhas de montagem, em função do volume de encomendas de veículos e do equipamento disponível.

A fábrica produz 3 modelos, designadamente,  $M1$ ,  $M2$ , e  $M3$ . A Tabela 2 mostra o ritmo de chegada de encomendas para cada modelo.

Tabela 2: Ritmo de chegada de novas encomendas, para cada modelo.

Modelo	M1	M2	M3
Nova encomenda cada	3 a 7 dias	4 a 6 dias	2 a 5 dias

A fábrica tem 5 zonas especializadas,  $Z1, \dots, Z5$ . Cada modelo, de acordo com o plano de construção, tem que passar por várias zonas, por uma determinada ordem. Cada zona tem um número de linhas de trabalho concorrentes, de acordo com a Tabela 3.

Tabela 3: Número de linhas de trabalho por zona.

Zona	1	2	3	4	5
Número de linhas	3	2	4	3	1

O plano de construção de cada modelo implica que o veículo passe por cada zona numa determinada ordem. A Tabela 4 apresenta a sequência de zonas por modelo, e o tempo médio de trabalho por zona.

Tabela 4: Sequência de zonas por modelo. O tempo médio de trabalho, em horas, para cada modelo, em cada zona, está indicado entre parêntesis.

Modelo	1	2	3
$Z1$	2º (0.80)	2º (0.60)	3º (0.70)
$Z2$	-	3º (0.85)	1º (1.20)
$Z3$	3º (0.75)	1º (0.50)	5º (1.00)
$Z4$	1º (1.10)	-	4º (0.90)
$Z5$	-	4º (0.50)	2º (0.25)

Por exemplo, para construir o modelo  $M1$  é necessário passar por 3 zonas distintas, na ordem indicada (1º  $Z4$ , 2º  $Z1$ , e 3º  $Z3$ ).

A Tabela 4 mostra o tempo médio de trabalho por modelo/zona. O tempo é modelado como uma variável aleatória que segue uma distribuição 2-Erlang.

## Distribuição k-Erlang

A função de densidade de probabilidade da distribuição é

$$f(x) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!} \quad \forall x, \lambda \geq 0$$

onde  $x$  é um valor aleatório, uniformemente distribuído entre 0 e 1,  $k$  é a *forma* da distribuição ( $k=2$ , no problema atual). A média da distribuição é determinada por  $\frac{k}{\lambda}$ .

# Objetivo

Pretende-se uma simulação da operação da fábrica com vista a caracterizar:

- O tempo médio que um modelo demora a ser construído;
- O tempo médio que cada tipo de modelo ( $M1$ ,  $M2$ ,  $M3$ ) tem que esperar entre zonas;
- O número médio de veículos em espera, em cada zona;
- O número médio de veículos em espera, em cada linha de trabalho;
- A percentagem de tempo de utilização de cada linha de trabalho (vs. tempo parado, sem serviço).

O simulador deve colecionar e mostrar estatísticas para encomendas realizadas num período de um ano (365 dias), com operação de 24 horas por dia, todos os dias.

Idealmente, todos os parâmetros do problema devem ser configuráveis. No mínimo, deve ser possível indicar o número de linhas de trabalho por zona.

## Regras

A simulação deve ser construída em Java (JDK 11).

A solução proposta deve aplicar:

- Orientação a objetos
  - Herança
  - Polimorfismo
- Programação concorrente
  - *Threads*
  - Sincronização

A estratégia de interação e visualização de informação é livre, mas é valorizada a implementação de interfaces gráficas.

O trabalho deve ser realizado em grupos previamente registados.

## Entrega

A entrega do projeto é feita no *e-learning*, e estão previstos três ficheiros:

- Ficheiro `.zip`, com o código fonte do projeto;
- Ficheiro `.jar`, para execução do programa;
- Ficheiro `.pdf`, com o relatório.

## Relatório

O relatório deve indicar a composição do grupo de trabalho, e a descrição da aplicação (i.e., modelo de dados, e decisões arquiteturais), incluindo manual de utilização.

**A ausência de identificação individual no relatório implica a anulação da participação individual no projeto.**

## Prova de autoria

Todos os projetos entregues serão sujeitos a prova de autoria. Para esse efeito, cada grupo terá que efetuar uma discussão com a docência, de forma a demonstrar que o código entregue foi de facto feito pelo grupo, e que a distribuição de trabalho foi equilibrada.

O resultado a prova de autoria é uma nota individual de participação, express em percentagem, onde 100% implica que a pessoa participou de forma equilibrada no projeto, e 0% a anulação da participação individual (resultando em zero no projeto).

O calendário das provas de autoria será disponibilizado no *e-learning*, após o prazo de entrega da implementação do projeto.

**A não comparência na prova de autoria implica a anulação da participação individual no projeto.**

## Avaliação

O projeto é avaliado com base em duas componentes: quantitativa ( $A$ ), e qualitativa ( $B$ ). A nota final do projeto é determinada por  $(0.5 \times A) + (0.5 \times B)$ . A nota individual de cada membro do grupo de trabalho é determinada pela multiplicação entre a nota individual de participação e a nota final do projeto.

O cumprimento estrito dos objetivos neste enunciado assegura, sem prejuízo de reprovação por irregularidade acadêmica, a nota mínima de 10 valores.

As notas finais do projeto serão disponibilizadas no *e-learning*.