

ESCOLA
SUPERIOR
DE MEDIA
ARTES
E DESIGN
POLITÉCNICO
DO PORTO

ALP

PROVA DE AVALIAÇÃO

ATIVIDADE LETIVA

Tecnologias e Sistemas de Informação para a Web

CURSO

2021/2022	2022/04/07	14h30	1h45
-----------	------------	-------	------

ANO LETIVO

DATA

HORA

DURAÇÃO

Programação Orientada a Objetos	1º ano
---------------------------------	--------

UNIDADE CURRICULAR

ANO

Mário Pinto / Ricardo Queirós/Daniel Carneiro	Avaliação distribuída
---	-----------------------

DOCENTE

ÉPOCA

Observações:

- O Exame é individual
- Podem consultar **apenas** as seguintes referências externas: MDN, W3Schools, Moodle e GitHub (repo da UC e repo pessoal)
- A avaliação do teste poderá implicar convocar os estudantes para uma sessão **de avaliação oral**, onde podem ter que explicar o código entregue
- Serão usadas ferramentas anti-plágio na avaliação dos testes
- **Crie uma pasta com o seu número de aluno. Guarde os exercícios nessa pasta. No final do teste submeta a pasta (compactada) no moodle, em objeto próprio de submissão**

A empresa Growdot, uma tecnológica líder de mercado no desenvolvimento de software para a área automóvel, vai inaugurar as suas novas instalações em Vila do Conde. Nestas novas instalações pretende implementar sistemas de controlo de acessos ao seu parque de estacionamento, assim como de colaboradores, pelo que pretende desenvolver as duas aplicações abaixo sugeridas:



Grupo I (50%)

Crie um programa que permita gerir os acessos ao parque de estacionamento da Growdot. Suponha que existe uma lista de matrículas autorizadas, definidas numa estrutura designada:

```
gessList = ['00-CC-00','01-CC-01','02-CC-02','03-CC-03','04-CC-04', '05-CC-05','06-CC-06','07-CC-07','08-CC-08','09-CC-09']
```

Pode criar esta lista estática de matrículas autorizadas, semelhante à acima apresentada, no seu programa.

O programa deve criar uma outra lista (um array), **parkList**, que permita gerir a ocupação do parque de estacionamento. O seu programa deve começar por solicitar a indicação da lotação máxima do parque (**lotPark**). Simule entradas e saídas de carros no parque, fazendo pedidos sucessivos de matrícula e tipo de movimento (E – para entrada, S – para saída). O programa deverá terminar com a introdução da matrícula de '00-00-00'.

A cada movimento, o seu programa deve invocar a função:

- **parkValidator(matricula, movimento)**. Função booleana que valida se o movimento é válido.
 Numa entrada: verifica se a matrícula está autorizada (**gessList**), se não foi atingida a lotação máxima do parque (**lotPark**), e se o veículo não está já no parque (**parkList**).
 Numa saída: verifica se a matrícula está no parque de estacionamento(**parkList**), para dar a sua saída.
 Quando a função **parkValidator** devolve um valor falso, deve surgir uma mensagem (genérica) ao utilizador, indicado que o movimento não é possível.

Quando o movimento é válido deve invocar, em seguida, a função:

- **parkManager(matricula, movimento).** Função que permite gerir a lista **parkList**, que contém as matrículas dos carros que estão no parque, num dado momento. Numa entrada, deve acrescentar a matrícula à lista; quando se trata de uma saída deve remover a respetiva matrícula da lista.

No final, o seu programa deve indicar, numa alert, quantos carros entraram no parque de estacionamento.

Grupo II - (50%)

Implemente uma aplicação que permita à GrowDot gerir os movimentos dos seus 10 colaboradores. O sistema de entradas/saídas do novo espaço tem um sensor de iris que permite registar as entradas e saídas dos seus colaboradores. Implemente algumas das funcionalidades do sistema:

1. Crie um array com os nomes dos 10 colaboradores: Rui, Maria, Joana, Beatriz, Pedro, Salomé, Iris, Vicente, Dinis e Ruben.
2. Crie um parágrafo no topo da página em H1 com a seguinte informação: ">> N <<" onde **N** é a percentagem do número de colaboradores dentro do edifício. Inicialmente não está ninguém no edifício
3. Crie uma tabela de forma dinâmica com 2 colunas:
 - a. Nome do colaborador
 - b. botão **ENTRAR** / **SAIR**
4. Crie uma caixa de texto e um botão **ADICIONAR** que vai permitir adicionar novos colaboradores ao array. Valide a não existência de duplicados.

>> 30% <<

NOME	E/S
Rui	ENTRAR
Maria	ENTRAR
Joana	SAIR
Beatriz	ENTRAR
Pedro	SAIR
Salomé	ENTRAR
Iris	ENTRAR
Vicente	SAIR
Dinis	ENTRAR
Ruben	ENTRAR

ADICIONAR

Algumas notas:

- Os botões comutam entre os dois valores possíveis (ENTRAR / SAIR);
- Ao clicar no botão, o valor N (% de colaboradores dentro do edifício) deve ser alterado automaticamente e o texto do botão deve alternar para o outro valor;
- No momento em que não tiver ninguém no edifício deve surgir uma caixa de alerta com a seguinte informação:

ATÉ AMANHÃ!