

## **Algoritmos e Fundamentos de Programação**

### **Projeto**

#### **Gestão de Reservas de Parque de Estacionamento**

**Ano Letivo: 2020/2021**

**Época Normal**

## Índice

<b>1</b>	<b>Introdução .....</b>	<b>3</b>
<b>2</b>	<b>Detalhes de Implementação .....</b>	<b>3</b>
2.1	Campista .....	3
2.2	Lote e Registo de Reserva .....	3
2.3	Mapa do Parque.....	4
2.4	Menu.....	4
<b>3</b>	<b>Regras de Desenvolvimento e Entrega do Projeto .....</b>	<b>5</b>
3.1	Entrega.....	5
3.2	Implementação e codificação.....	5
3.3	Constituição de grupos.....	5
3.4	Entrega do projeto.....	6
<b>4</b>	<b>Regras e Critérios de Avaliação do Projeto .....</b>	<b>6</b>
4.1	Regras de Avaliação.....	6
4.2	Critérios de Avaliação .....	7

## 1 Introdução

O objetivo deste projeto é desenvolver uma aplicação em consola, utilizando a linguagem C ANSI e a abordagem procedimental e modular, de forma a implementar um sistema de gestão de reservas de lotes de um parque de campismo.

No problema a modelar destacam-se as seguintes entidades:

- Campista;
- Lote e Registo de Reserva;
- Mapa do Parque;
- Menu.

## 2 Detalhes de Implementação

### 2.1 Campista.

Um **campista** representa uma pessoa que faz parte de uma reserva de um determinado lote do parque. Deve registar-se o nome e a idade de cada campista.

### 2.2 Lote e Registo de Reserva

Para cada **lote** deve ser registada informação sobre:

1. A reserva atual do lote;
2. O histórico das últimas 3 reservas efetuadas nesse lote (ordenadas da mais recente para a mais antiga).

Por sua vez, uma **reserva** deve registar:

1. O tipo de alojamento (tenda, caravana ou autocaravana);
2. Se tem ligação à rede elétrica;
3. Os dados do(s) campista(s);
4. A tarifa a pagar (pode assumir que a tarifa representa um pagamento único por reserva, ou assumir que a tarifa é aplicada tendo em conta o número de dias reservados).

Cada reserva pode alocar no máximo 6 campistas, que devem estar ordenados nos registos por ordem decrescente de idade.

O total da tarifa a pagar por cada reserva depende da sua composição e é calculada segundo a seguinte tabela:

<b>TENDA</b>	4,50 €
<b>CARAVANA</b>	6,00 €
<b>AUTOCARAVANA</b>	14,90 €
<b>ELECTRICIDADE</b>	3,00 €
<b>CAMPISTA</b>	3,50 €

e.g. uma reserva para uma tenda, eletricidade e 2 campistas pagará uma tarifa de:  $4,50 + 3,00 + (2 \times 3,50) = 14,50$  €.

## 2.3 Mapa do Parque

O mapa do parque deverá mostrar todos os lotes existentes assinalando os lotes livres/reservados, bem como a contagem total de lotes livres e a percentagem de ocupação atual do parque.

Considere que o parque disponibiliza inicialmente 25 lotes representando-os, por exemplo, com uma matriz 5x5. No entanto, considere ainda que estes valores devem poder ser facilmente alterados por forma a permitir a possibilidade de representar um parque com outras dimensões.

Sugere-se, de seguida, uma apresentação simplista do mapa contendo a informação mais relevante das características de cada reserva atual de cada lote:

Mapa de Lotes					
	1	2	3	4	5
1	...	Tx2	...	...	...
2	Te4	Tx1	...	...	Ce5
3	...	...	...	...	...
4	...	...	...	...	...
5	...	...	Ce5	Ce3	Ce4
Lotes Livres:		18			
Tx Ocupação:		28,0%			

Sugestão de apresentação do Mapa de Lotes.

No mapa apresentado, os lotes são identificados por coordenadas **Linha****Coluna** (correspondentes às posições da matriz) e mostra a seguinte informação: tipo de alojamento; existência de ligação à rede elétrica; número de campistas na reserva atual (e.g. o lote 2 1 (**Te4**) está reservado para uma **T**enda com **e**letricidade e **4** campistas).

## 2.4 Menu

O Menu deverá disponibilizar opções para:

1. Visualizar o mapa atualizado do parque;
2. Reservar um lote e registar os seus detalhes;
3. Consultar os detalhes de um determinado lote (detalhes da **reserva atual** e do **histórico**);
4. Pesquisar lote por nome de campista (i.e. localizar em que reserva de lote está atualmente alojado determinado campista);
5. Editar a informação da **reserva atual** de um lote (tipo de alojamento, ligar eletricidade, adicionar/remover campistas);
6. Libertar a **reserva atual** de um lote para que fique disponível para nova reserva (adicionando essa reserva atual ao histórico);
7. Sair da aplicação;
8. Outras funcionalidades que considere que se justifiquem.

NOTA: Deverão ser efetuadas todas as validações necessárias (e.g. verificar se opções introduzidas são válidas e se têm o formato esperado; verificar se o lote a reservar existe e se está livre; etc.).

### **3 Regras de Desenvolvimento e Entrega do Projeto**

#### **3.1 Entrega**

A entrega do projeto será em duas fases:

1ª fase – até 17 de janeiro de 2021 às 23h59

Todo o código deverá estar comentado de forma detalhada.

Deve ser entregue um relatório, explicando todo o processo de desenvolvimento do projeto.

#### **3.2 Implementação e codificação**

O programa deve ser desenvolvido utilizando a linguagem C ANSI, colocando em prática os conceitos fundamentais do paradigma procedimental e modular, é ainda permitida a utilização das bibliotecas disponíveis.

Em relação às regras de codificação, siga as convenções adotadas normalmente no decorrer da UC de AFP:

- A notação *lowerCamelCase* para os nomes de subprogramas e dos seus parâmetros, de variáveis locais e de atributos;
- Não utilize o símbolo ‘\_’ nos identificadores, nem abreviaturas no início das mesmas.

É necessário que o projeto cumpra o que é pedido no seu enunciado, sendo deixado ao critério do programador qualquer pormenor de implementação que não seja referido, o qual deverá ser devidamente documentado.

Nas funcionalidades desenvolvidas, deverão ser incluídas todas as validações necessárias para impedir um comportamento incorreto do sistema.

Sempre que fizer sentido, os subprogramas deverão de ser responsáveis por apresentar no ecrã mensagens de informação e/ou erro, indicando o processamento que foi feito.

As mensagens apresentadas pela aplicação podem ser apresentadas em português e/ou inglês.

#### **3.3 Constituição de grupos**

Cada projeto deverá ser elaborado em grupos de dois alunos, podendo ser desenvolvido individualmente em casos pontuais devidamente justificados. Não serão permitidos, em caso algum, grupos com mais do que dois alunos.

Os grupos são propostos pelos alunos e têm de ser previamente aceites pelo professor aquando da apresentação do projeto UC de AFP.

### 3.4 Entrega do projeto

- O projeto deverá ser entregue até à data limite especificada por **via exclusivamente eletrónica, utilizando a área dos trabalhos na plataforma Moodle**. Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado em **formato ZIP**. Em caso de dificuldades no acesso à plataforma Moodle, o envio dos ficheiros poderá ser feito por correio eletrónico para o respetivo docente, dentro do prazo acima indicado.
- **Não serão aceites quaisquer projetos entregues fora do prazo.**
- Todos os materiais do projeto devem ser devidamente identificados com nome, número e endereço de correio eletrónico dos alunos.

Os materiais do projeto deverão incluir:

- Relatório de acordo com o template disponibilizado, que contenha o manual de utilizador
- O código fonte do programa na forma de projeto Code::Blocks, com todas as funcionalidades implementadas.
- Antes de compactarem o projeto não se esqueçam de eliminar os ficheiros que se encontram dentro das pastas <debug> e <release>.
- Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado em formato ZIP cujo nome deverá ter a seguinte nomenclatura:  
**<curso>\_<PrimeiroNomeUltimoNomedoAluno1>\_<PrimeiroNomeUltimoNomedoAluno2>.zip**  
(exemplo: TPSI\_AndreEsteves\_MarioGuedes).
- **O incumprimento das normas de entrega está sujeito a uma penalização de até 1 valor.**

## 4 Regras e Critérios de Avaliação do Projeto

### 4.1 Regras de Avaliação

A avaliação do projeto está sujeita às seguintes regras:

- Não serão aceites quaisquer projetos entregues fora do prazo.
- A classificação do programa terá em conta a qualidade da programação e a estrutura do código criado segundo os princípios da Programação Procedimental e Modular.
- Serão premiadas a imaginação e a criatividade.
- O projeto terá uma componente de avaliação oral obrigatória com classificação individual dos elementos do grupo.
- As datas das discussões serão publicadas após a entrega dos projetos.
- Os alunos que não comparecerem à discussão serão classificados com zero. Nesta discussão será apurada a capacidade do aluno de produzir o código apresentado. Nos casos em que essa capacidade não for demonstrada, a nota atribuída será zero.
- A avaliação oral é realizada pelo respetivo professor e irá ser feita uma marcação prévia para cada grupo de trabalho.
- Todos os projetos serão submetidos a um sistema automático de deteção de cópias. Os projetos que forem identificados como possíveis cópias, e verificando-se serem efetivamente cópias, serão anulados.

#### 4.2 Critérios de Avaliação

Este projeto será avaliado segundo os seguintes critérios:

<b>Implementação</b>	<b>16.0</b>
Interação com o Utilizador – Menu	3.0
Mapa dos Lotes	3.0
Criar, Consultar, Editar e Apagar Reserva	4.0
Registo de Histórico de Reservas por Lote	2.0
Pesquisa de Lote por Nome de Campista	2.0
Utilização de Ficheiros	2.0
<b>Avaliação qualitativa</b>	<b>4.0</b>
Bom estilo (identificadores, comentários, indentação)	2.0
Qualidade geral, detalhes de implementação e funcionalidades adicionais	2.0
<b>Valor máximo admissível</b>	<b>20.0</b>

Bom trabalho!