

# ASSIGNMENT BRIEF

## BTEC (RQF)

### 2018/2019

					YEAR	1st	x	2nd	
QUALIFICATION	HND IN COMPUTING - COMPUTER SCIENCE								
UNIT NUMBER AND TITLE	EN	Nº 1	Programming			SCHEDULE PER UNIT		60 H	
	PT	Nº 1	Programação						
UNIT ASSESSOR	Nuno Frutuoso				UNIT CREDITS	15	UNIT LEVEL	4	
UNIT TUTOR	Fernando Moreira								
DEADLINE	19 / 12 / 2018								
ISSUE DATE	10 / 10 / 2018					SUBMISSION DATE	19 / 12 / 2018		
ASSIGNMENT TITLE	Introdução à programação								

#### ASSIGNMENT BRIEF AND GUIDELINES:

Nesta unidade pretende-se que o aluno adquira uma visão geral da programação, como uma ferramenta para automatizar processos, programando algoritmos. Pretende-se que o aluno entenda o que acontece quando “corre um programa” no computador, como escrever um programa para executar a função pretendida, bem como identificar e corrigir erros no código.

Em suma, pretende-se que o aluno seja capaz de resolver problemas utilizando a programação e pensando como um verdadeiro Computer Scientist.

#### SCENARIO

A startup tecnológica XPTO de mobilidade elétrica vai lançar até ao final do ano de 2018 uma aplicação BETA para gestão da sua frota e atribuição de serviços de transporte aos seus clientes. A equipa de desenvolvimento de software já reuniu os requisitos necessários e os requisitos avançados que debes corresponder da melhor forma no tempo que tens disponível.

#### PROJECT

##### Requisitos necessários:

Permite dois tipos de utilizador, o administrativo e o cliente. Ao longo do programa, as operações possíveis devem aparecer na Shell com o comando respetivo.

- O administrativo pode adicionar viaturas elétricas, que podem ser carros (até 4 lugares) ou carrinhas (até 6 lugares). Pode remover viaturas colocando o seu estado a inativo.
- Quando uma viatura é criada, deve ser-lhe atribuído um número único, atribuída uma marca/modelo, uma matrícula, um condutor, número de km percorridos, valor faturado e número de serviços efetuados.
- Quando uma viatura é criada, é-lhe atribuída a posição X=2.5 e Y=2.5, que corresponde ao centro de um quadrado com 5 km de lado onde as viaturas podem circular.
- O preço por km dos carros é 0,40€, com um valor inicial de 3€ e desloca-se a uma velocidade média de 30km/h. O preço por km das carrinhas é 0,70€, com um valor inicial de 4€ e uma velocidade média de 25km/h.

- As informações dos veículos são carregadas através do ficheiro veículos.txt. Deve ser feita a deteção de erros e a sua atualização.
- O administrador pode pesquisar a informação de todos os veículos ou a informação de um veículo em particular. Pode realizar uma procura simples por nome, marca, matrícula, valor faturado acima/abaixo de um nível, número de serviços efetuados acima/abaixo de um determinado nível.
- O administrador pode pesquisar quais são os clientes inscritos na aplicação.
- O cliente deve criar uma conta única e entrar com a sua password.
- Os clientes já listados são carregados através do ficheiro clientes.txt. Deve ser feita a deteção de erros e sua atualização.
- O cliente pode simular a distância entre o ponto A (onde está) ao ponto B (onde quer ir).
- O cliente pode procurar a viatura mais próxima para o levar do ponto A ao ponto B. A lista de todos os veículos aparecerá por ordem crescente de distância ao cliente, indicando também o custo da viagem de A para B, a duração da viagem e quanto tempo a viatura demora a chegar até ao ponto A.
- No pedido, o cliente pode indicar o número de passageiros do veículo até um máximo de 6, e caso ultrapasse o limite do carro deverão ser apresentadas apenas carrinhas.
- Ao procurar a viatura, o cliente poderá ver num gráfico a posição das viaturas e a sua própria posição.
- Após a listagem, o cliente deve confirmar qual a viatura que deseja, ou cancelar.
- O nome do cliente é único e para maior segurança, todas as passwords devem obedecer à regra de pelo menos 6 caracteres, com pelo menos uma maiúscula, um algarismo e conter pelo menos um carácter especial como @ # % etc. Um ficheiro de texto (login.txt) guardará o registo de clientes e passwords.
- De forma a tornar a plataforma mais interativa, sempre que se executar um comando “atualizar”, deve ser atribuída aleatoriamente uma nova posição à viatura, no interior de um quadrado com 5 km de lado. Se selecionado, poderá mostrar um gráfico a posição das viaturas.

#### Requisitos avançados:

- No pedido, o cliente pode indicar o tempo máximo que pretende chegar ao destino e só devem aparecer as viaturas que correspondam a esse pedido.
- Após aceitar um cliente, a viatura deve ficar ocupada. As viaturas ocupadas não aparecem na listagem. Quando é executado o comando “atualizar”, deve ser atribuída aleatoriamente uma nova posição à viatura, o número de km percorridos deve ser incrementado. Aleatoriamente é atribuído se a viatura estava ocupada ou livre, caso estivesse livre, o valor faturado deve ser incrementado.
- No pedido, o cliente deve indicar o número de passageiros do veículo até um máximo 8 e caso ultrapasse o limite do carro deverão ser apresentadas as viaturas mais próximas que combinadas satisfaçam o pedido.

#### FASE 1.

Demonstra que dominas os conceitos básicos da programação.

1 - Define por palavras tuas o que é formalmente um algoritmo.

Sabendo que neste projeto será necessário criares um programa e algoritmos.

2 - Descreve quais são genericamente os passos necessários para se criar programas/software? Que modelos de desenvolvimento de software conheces?

Um passo importante é o desenho da aplicação antes de começar a programar.

3 - Define um algoritmo para o cliente entrar na aplicação e solicitar a viatura mais próxima. Descreve-o em pseudocódigo e em flow-chart.

O pseudocódigo é uma ótima forma de planeares a escrita do código. Mas o computador não entende a nossa linguagem, por isso utilizamos uma linguagem de programação.

**Deadline:**  
**Ver o Plano de**  
**Aulas**

<p>4 - O que é uma linguagem de programação? Determina quais são os passos que existem entre a escrita do código e a sua execução pelo computador.</p> <p>5 - Dado um algoritmo em pseudocódigo e a sua respetiva implementação em linguagem C e em Python, examina a implementação do algoritmo na linguagem respetiva. Faz correspondências e avalia a relação entre o algoritmo e a variante do código.</p>	
FASE 2.	
<p>Hoje em dia, para implementar numa linguagem de programação é crucial para programador utilizar um IDE (Integrated Development Environment).</p> <p>1 - Sabes porquê? Apresenta quais as principais ferramentas de um IDE. Numa breve pesquisa, apresenta dois exemplos de IDEs programar em C, em C++, em Python e em JAVA.</p> <p>Existem vários paradigmas de programação, consoante a natureza do problema a resolver.</p> <p>2 - Compara os 3 paradigmas procedural, object-orientated e event-driven. Descreve os seus aspetos principais e a relação entre eles. Dá exemplos de linguagens de programação adequadas para implementar cada modelo de programação.</p> <p>3 - Tens acesso ao código fonte de 3 aplicações que implementam os 3 paradigmas simplificados em python, compara-os e critica-os, justificadamente, em relação à estrutura e características.</p>	<p><b>Deadline:</b> Ver o Plano de Aulas</p>
FASE 3.	
<p>1 - A partir de um pseudocódigo, escreve um programa que implemente um algoritmo proposto em aula. Utiliza o IDE para gerires o desenvolvimento do programa.</p> <p>Escreveste algoritmos e observaste que nem sempre o resultado é o esperado. Como é possível? Tens um “bug” no código.</p> <p>2 - Explica por palavras tuas o que é um bug, o que é o processo de debugging e que ferramentas de debugging tens disponíveis no IDE que utilizas.</p> <p>3 – Na aula desenvolveste algoritmos com/sem recurso ao IDE. Avalia e comenta a utilização ou não do IDE para desenvolver uma aplicação.</p> <p>4 - Tens acesso a 3 programas (prog_bug1.py, prog_bug2.py, prog_bug3.py) que têm bugs, encontra-os e explica como os resolveste?</p>	<p><b>Deadline:</b> Ver o Plano de Aulas</p>
FASE 4.	
<p>1 - Avalia como o processo de debugging é útil para desenvolver programas mais seguros e robustos.</p> <p>Cada linguagem de programação tem standards de escrita e organização do código, são muito úteis os programadores.</p> <p>2 – Enumera os standards que utilizas no desenvolvimento do teu código?</p> <p>3 – Avalia criticamente a necessidade de standardização na programação, tanto para o desenvolvimento de software em equipa como para o próprio indivíduo. Descreve também a tua experiência neste tema.</p>	<p><b>Deadline:</b> Ver o Plano de Aulas</p>

## FASE 5.

Está a chegar a altura de lançares a aplicação! Deverás publicar o código fonte que desenvolveste, colocando-o na plataforma ETIC com a extensão .py. Uma nova equipa ficará responsável pela manutenção do software, certifica-te de que comentas corretamente o teu código e que utilizas os standards de programação em Python.

Submete um documento anexo ao projeto onde expliques como utilizar (executar) o teu programa, que funcionalidades tens implementadas, que bugs existem detetados. Comenta brevemente a tua experiência na programação, como por exemplo: desafios de criar algoritmos, aprender uma linguagem de programação, desafios de utilizar um IDE, desafios do processo de debugging.

**Deadline:**  
**Ver o Plano de Aulas**

## UNIT LEARNING OUTCOMES :

- LO1. Define basic algorithms to carry out an operation and outline the process of programming an application.
- LO2. Explain the characteristics of procedural, object-orientated and event-driven programming, conduct an analysis of a suitable Integrated Development Environment (IDE).
- LO3. Implement basic algorithms in code using an IDE.
- LO4. Determine the debugging process and explain the importance of a coding standard.

## SUBMISSION FORMAT

A submissão dos trabalhos escrito e prático, deve ser feita de forma individual pelo aluno.

Part 1: Entrega em formato PDF, não deve exceder 5 páginas

Part 2: Entrega em formato PDF, não deve exceder 5 páginas

Part 3: Entrega em formato PDF, não deve exceder 5 páginas

Part 4: Entrega em formato PDF, não deve exceder 5 páginas

Part 5: Entrega do projeto em extensão .py. Documento anexo ao projeto em formato PDF, não deve exceder 5 páginas

## INSTRUCTIONS AND GUIDENCE

- Os documentos devem ser elaborados com linguagem clara e concisa.
- O projeto é individual, mas o debate de ideias entre colegas é saudável

**Note: All information sources and resources, including websites used to complete this assignment must be stated and correctly referenced. No credit will be given for wholesale copying from information sources and checks for plagiarism may be made on your submitted report**

\*Please access HN Global for additional resources support and reading for this unit. For further guidance and support on report writing please refer to the Study Skills Unit on HN Global [www.highernationals.com](http://www.highernationals.com)

## Resultados de Aprendizagem e Critérios de Avaliação / Learning Outcomes and Assessment Criteria

Aprovação		Mérito	Distinção
<b>LO1</b> Definir algoritmos básicos para realizar uma operação e delinear o processo de programação de uma aplicação. / Define basic algorithms to carry out an operation and outline the process of programming an application.			<p><b>D1</b> Examinar a implementação de um algoritmo numa linguagem adequada. Avaliar a relação entre o algoritmo escrito e a variante de código. / Examine the implementation of an algorithm in a suitable language. Evaluate the relationship between the written algorithm and the code variant.</p>
<b>P1</b> Fornecer uma definição do que é um algoritmo e delinear o processo na construção de uma aplicação. / Provide a definition of what an algorithm is and outline the process in building an application.		<b>M1</b> Determinar as etapas desde a escrita do código até à execução. / Determine the steps taken from writing code to execution.	
<b>LO2</b> Explicar as características da programação orientada para o objecto e baseada em eventos, efectuar uma análise de ambiente de desenvolvimento integrado (IDE). / Explain the characteristics of procedural, objectorientated and event-driven programming, conduct an analysis of a suitable Integrated Development Environment (IDE).			
<b>P2</b> Fornecer explicações sobre o que são paradigmas processuais, orientados para o objecto e baseados em eventos; as suas características e a relação entre os mesmas. / Give explanations of what procedural, objectorientated and eventdriven paradigms are; their characteristics and the relationship between them.		<b>M2</b> Analisar as características comuns a que tem acesso um programador num IDE. / Analyse the common features that a developer has access to in an IDE.	
<b>LO3</b> Implementar os algoritmos básicos no código utilizando um IDE. / Implement basic algorithms in code using an IDE			<p><b>D2</b> Avaliar de forma crítica o código de origem de uma aplicação que implemente os paradigmas de programação, em termos de estrutura e características do código. / Critically evaluate the source code of an application which implements the programming paradigms, in terms of the code structure and characteristics.</p>
<b>P3</b> Escrever um programa que implemente um algoritmo utilizando um IDE. / Write a program that implements an algorithm using an IDE.		<b>M3</b> Utilizar o IDE para gerir o processo de desenvolvimento do programa. / Use the IDE to manage the development process of the program.	
<b>LO4</b> Determinar o processo de depuração e explicar a importância de um padrão de codificação. / Determine the debugging process and explain the importance of a coding standard			<p><b>D3</b> Avaliar o uso de um IDE para o desenvolvimento de aplicações em comparação com a não utilização de um IDE. / Evaluate the use of na IDE for development of applications contrasted with not using an IDE.</p>
<b>P4</b> Explicar o processo de depuração e explicar os aplicativos de depuração disponíveis num IDE. / Explain the debugging process and explain the debugging facilities available in the IDE.		<b>M4</b> Avaliar como pode ser utilizado um processo de depuração para ajudar a desenvolver aplicações mais fortes e seguras. / Evaluate how the debugging process can be used to help develop more secure, robust applications.	
<b>P5</b> Delinear o padrão de codificação que se utilizou no próprio código. / Outline the coding standard you have used in your code.			<p><b>D4</b> Avaliar de forma crítica porque é necessário um padrão de codificação numa equipa bem como para um indivíduo. / Critically evaluate why a coding standard is necessary in a team as well as for the individual.</p>

**BIBLIOGRAPHY:**

- AHO, A. V. et al. (1987) Data Structures and Algorithms. 1st Ed. Addison-Wesley.
- HUNT, A. et al. (2000) The Pragmatic Programmer: From Journeyman to Master. 1st Ed. Addison-Wesley.
- MCCONNELL, S. (2004) Code Complete: A Practical Handbook of Software Construction. 2nd Ed. Microsoft Press
- Head First Programming, Paul Barry & David Griffiths - O'Reilly
- Head First C, Paul Barry & David Griffiths - O'Reilly
- Programação em Python: Introdução à programação com múltiplos paradigmas: João P. Martins 2013 IST Press
- Python for Software Design: Downey A.B 2009 Cambridge University Press

**WEBGRAPHY:**

<http://interactivepython.org/runestone/static/thinkcspy/index.html>  
<https://docs.python.org/3/>

**THE BRIEFING HAS BEEN VERIFIED AS BEING EFFECTIVE FOR THE PURPOSE:****ASSESSOR  
SIGNATURE****INTERNAL VERIFIER  
NAME**

Fernando Moreira

**INTERNAL VERIFIER  
SIGNATURE**