

Universidade do Minho
Escola de Engenharia

José Gomes, a93083
Bruno Santos, a93087

Inteligência Artificial para as Telecomunicações

**Mestrado Integrado em Engenharia de
Telecomunicações e Informática
2022/2023**

Índice de Conteúdos

Índice de Conteúdos	i
Lista de Figuras	ii
1. Introdução	1
2. Descrição do Problema	2
2.1 Movimento do carro	2
2.2 Custos relativos à deslocação do veículo	2
2.3 Circuito gerado	3
3. Formulação do Problema	4
3.1 Representação do Estado	4
3.2 Estado Inicial	6
3.3 Estado Objetivo	6
3.4 Operadores	6
3.5 Custo da solução	6
4. Algoritmos e estratégias adotados pelo grupo	7
4.1 Classe Node	7
4.2 Classe Car	8
4.3 Circuito	9
5. Análise de resultados	11
6. Conclusão	12

Lista de Figuras

Figura 1:	Componente gráfica da aplicação desenvolvida.	1
Figura 2:	Circuito gerado pelo grupo.	3
Figura 3:	Deslocações possíveis do veículo.	5
Figura 4:	Classe <i>Node</i>	7
Figura 5:	Classe <i>Car</i> e Enumerado <i>Acceleration</i>	8
Figura 6:	Algoritmo da função <i>determine_circuit_costs</i>	9
Figura 7:	Algoritmo da função <i>graph_from_circuit</i>	10

1. Introdução

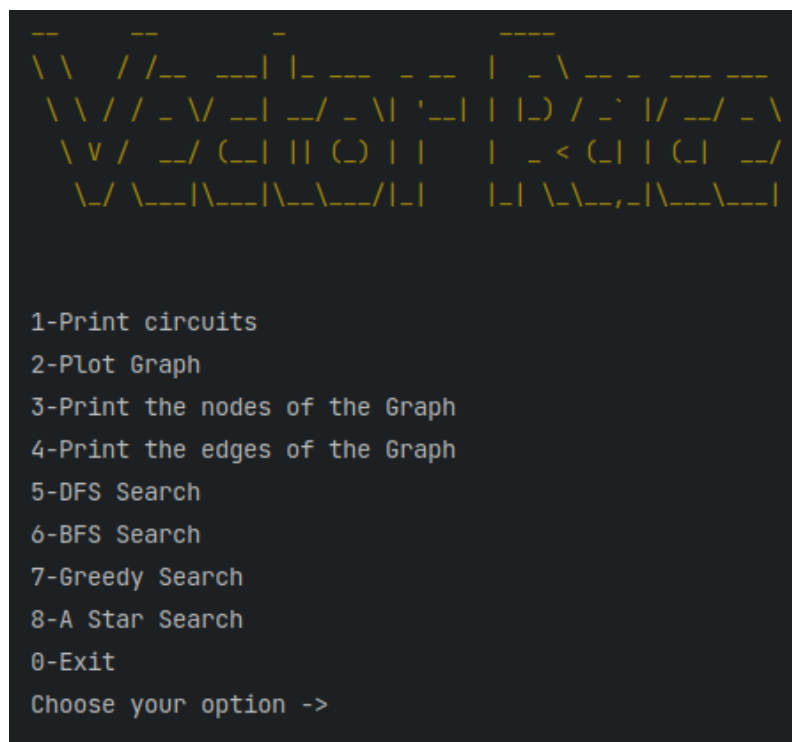
No âmbito da Unidade Curricular de Inteligência Artificial para as Telecomunicações, foi proposta a realização de um trabalho prático cujo objetivo é o desenvolvimento de diversos algoritmos de procura para a resolução de um jogo, nomeadamente o *VectorRace*.

O *VectorRace* é um jogo de simulação de uma corrida de carros, que possui um conjunto de movimentos e normas associadas.

A deslocação do carro no *VectorRace* é simples, as ações que são efetuadas pelo veículo são equivalentes a um conjunto de acelerações. Num determinado instante da corrida, o carro pode acelerar -1,0 ou 1 unidades em cada direção (linha e coluna).

O presente relatório contém a descrição das estratégias e algoritmos que foram adotados pelo grupo. Em junção com a análise das componentes referidas, o mesmo também inclui a comparação dos resultados, que foram obtidos pela aplicação dos diferentes tipos de estratégias de procura utilizados e a justificação das heurísticas que foram aplicadas nos algoritmos de procura informada.

A figura 1 ilustra a componente gráfica da aplicação desenvolvida.



```

--  --  -  ----
\ \ / / _ _ _ | | _ _ _ _ _ | | _ _ _ _ _
\ \ / / _ \ _ _ | | / _ \ | | | | / _ \ | | / _ \
 \ v / _ / ( _ | | ( ) | | | | < ( _ | | ( _ _ /
  \ / \ _ _ \ _ _ \ _ _ \ _ _ / | | | | \ _ _ \ _ _ \ _ _

1-Print circuits
2-Plot Graph
3-Print the nodes of the Graph
4-Print the edges of the Graph
5-DFS Search
6-BFS Search
7-Greedy Search
8-A Star Search
0-Exit
Choose your option ->

```

Figura 1: Componente gráfica da aplicação desenvolvida.

2. Descrição do Problema

No âmbito da U.C. de Inteligência Artificial para as Telecomunicações foi proposta a elaboração de um trabalho prático relativo a um jogo de corridas, nomeadamente o *VectorRace*. O trabalho desenvolvido consiste na resolução dos problemas propostos, através da conceção e implementação de algoritmos de procura.

2.1 Movimento do carro

O movimento de um carro no *VectorRace* é simples, qualquer movimento efetuado pelo veículo resultará numa variação do seu conjunto de acelerações.

Num determinado instante, o carro poderá acelerar -1, 0 ou 1 unidades em cada direção (linha e coluna). O carro movimenta-se ao longo do plano cartesiano, isto é, desloca-se segundo o eixo (positivo) das abcissas e segundo o eixo das ordenadas.

O movimento referido pode ser descrito pelas equações seguintes:

$$p(l)^j * p(l) = p(l)^j + v(l)^j + a(l)$$

$$p(c)^j * p(c) = p(c)^j + v(c)^j + a(c)$$

Relativamente à velocidade do mesmo, esta pode ser determinada pelo conjunto de equações seguinte:

$$v(l)^j * v(l) = v(l)^j + a(l)$$

$$v(c)^j * v(c) = v(c)^j + a(c)$$

2.2 Custos relativos à deslocação do veículo

A aplicação desenvolvida simula a corrida de um carro, logo, existe a probabilidade do mesmo sair fora dos limites da pista. Caso ocorra a situação citada, o carro terá que voltar para a posição anterior com velocidade nula.

Para uma determinada jogada, cada deslocação do carro de uma certa posição para outra, terá o custo de 1 unidade e caso saia fora dos limites da pista, o custo aplicado será de 25 unidades.

Apesar de não constar no enunciado do projeto, o grupo considerou relevante atribuir um custo de 10 unidades, caso o carro atinga a meta. A introdução desta nova funcionalidade revelou ter uma enorme importância e impacto na implementação deste projeto.

2.3 Circuito gerado

O circuito gerado pelo grupo encontra-se representado pela figura 2.

X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	-	-	-	X	X	-	-	-	-	-	-	-	X	X	-	-	X	
X	-	-	-	-	-	-	-	-	-	X	X	X	-	-	-	-	-	-	F
X	P	-	-	X	X	X	X	-	-	-	-	-	-	X	-	-	-	-	F
X	-	-	-	-	-	-	-	-	-	X	X	X	-	-	-	-	-	-	F
X	X	X	X	-	-	X	X	-	-	-	-	-	-	-	-	-	-	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Figura 2: Circuito gerado pelo grupo.

Através da análise da figura 2 é possível concluir que este circuito é formado por 20 colunas e 7 linhas, onde o caractere '-' representa a pista, o caractere 'X' um obstáculo/fora da pista, o caractere 'P' a posição inicial e o caractere 'F' representa as posições finais (meta).

3. Formulação do problema

Através da análise do enunciado foi possível proceder à formulação do problema. Esta formulação foi desenvolvida com recurso a cinco componentes.

- Representação do Estado
- Estado Inicial
- Estado Objetivo
- Operadores (Nome, Pré-Condições, Efeitos, Custo)
- Custo da solução

3.1 Representação do Estado

Esta componente revelou ser a mais propulsora e crucial para o desenvolvimento deste projeto.

A componente mencionada resultou do processamento do circuito, a ser percorrido pelo automóvel. Dentro do processamento do circuito procedeu-se à abstração do mesmo, ou seja, mapeou-se o circuito no plano cartesiano, logo, o trajeto será efetuado sobre duas dimensões (segundo o eixo das abcissas e o eixo das ordenadas).

O mapeamento do circuito no plano cartesiano foi repartido em duas etapas:

1. Recolha dos custos consoante as diferentes deslocações na pista:
 - (a) O automóvel parte da posição inicial com um custo de **0** unidades.
 - (b) Cada deslocação de uma certa posição para outra, terá o custo de **1** unidades.
 - (c) Caso saia fora dos limites da pista, o custo será de **25** unidades.
 - (d) Para atingir a posição final, o custo será de **10** unidades.
2. Estudo das deslocações possíveis do automóvel:
 - (a) Após o processamento dos custos, foi necessário estudar as deslocações possíveis, inerentes ao veículo. O grupo chegou à conclusão que dado um carro, existem 3 deslocamentos possíveis, para os nodos seguintes: 'B', 'C' e 'D':

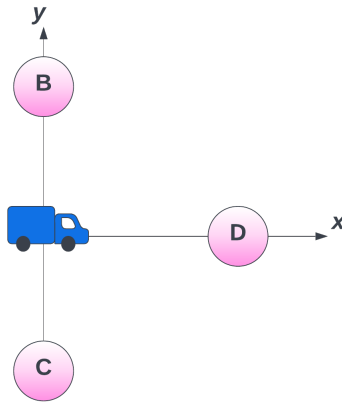


Figura 3: Deslocações possíveis do veículo.

- (b) O movimento do carro, que é exercido sobre o plano cartesiano, encontra-se personificado na figura 3. O eixo das ordenadas é descrito pela deslocação do carro para os nodos B ou C e o eixo das abcissa é exposto pela deslocação veículo para o nodo D. Tal como a imagem sugere, o carro não pode inverter o sentido de marcha (voltar para trás).
- (c) As funções *determine_circuit_costs* e *graph_from_circuit*, que se encontram presentes no módulo *circuit.py*, abordam os dois tópicos referidos.

3.2 Estado Inicial

O estado inicial é determinado pela caractere 'P', a posição do mesmo varia consoante a fisionomia do circuito. É de relembrar, que o circuito gerado pelo grupo possui uma posição inicial e 3 metas/posições finais.

3.3 Estado Objetivo

O estado objetivo é determinado pelo caractere 'F', que por consequência da fisionomia do circuito, poderá equivaler a um ou mais destinos possíveis.

3.4 Operadores

Os operadores englobam o conjunto de ações possíveis do veículo, estabelecem normas/pré-condições e determinam os efeitos/consequências dessas ações e, por consequência, associam custos. No domínio do problema, os operadores presentes encontram-se listados nos tópicos abaixo:

- O carro pode acelerar -1, 0, 1 unidades em cada direção (linha e coluna).
- O carro pode sair da pista: terá que voltar para a posição anterior com velocidade nula.
- O custo de deslocação do carro de uma posição x para uma posição y é de 1 unidade. Fora dos limites da pista, o custo será de 25 unidades.

3.5 Custo da solução

O custo da solução pode ser representado pela fórmula seguinte:

$$C(s, a, s')$$

- C representa o custo da solução.
- s representa o estado **atual**.
- a representa a ação que foi executada pelo agente para atingir o próximo estado.
- s' representa o **próximo** estado.

4. Algoritmos e estratégias adotados pelo grupo

Esta secção abordará os algoritmos e estratégias, que foram adotados pelo grupo, para a construção e implementação do projeto respetivo. O grupo construiu fluxogramas e diagramas de classe para ilustrar os organismos essenciais para o bom funcionamento do *VectorRace*.

4.1 Classe Node

A classe *Node* representa um nodo, que é inerente a um determinado grafo. Cada posição na pista é refletida por um nodo, logo, esta apresentará os estados e comportamentos, que são intrínsecos a um nodo.

A figura 4 descreve os estados e comportamentos relativos à classe *Node*.

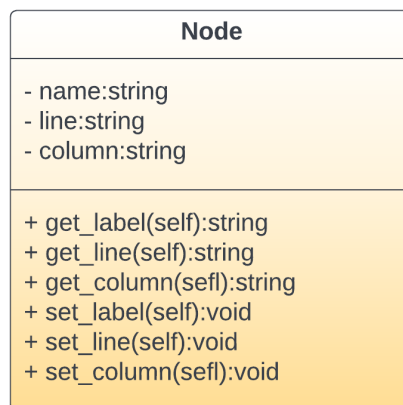


Figura 4: Classe *Node*.

Um nodo é caracterizado por um nome, por uma linha e por uma coluna. O nome do nodo corresponde a um caractere (P,-,X ou Z), a linha e a coluna formam o par de coordenadas cartesianas (linha,coluna) do mesmo. De forma a proteger a entidade do objeto e a garantir o seu encapsulamento, as variáveis de instância são privadas, logo, apenas poderão ser acedidas com recurso às funções *getters* do mesmo.

4.2 Classe Car

A classe *Car* representa o carro que percorre a pista. Dentro da pista, cada carro é identificado pela sua matrícula, logo, não pode existir mais que um carro com a mesma matrícula.

A figura 4 descreve os estados e comportamentos relativos à classe *Node*.

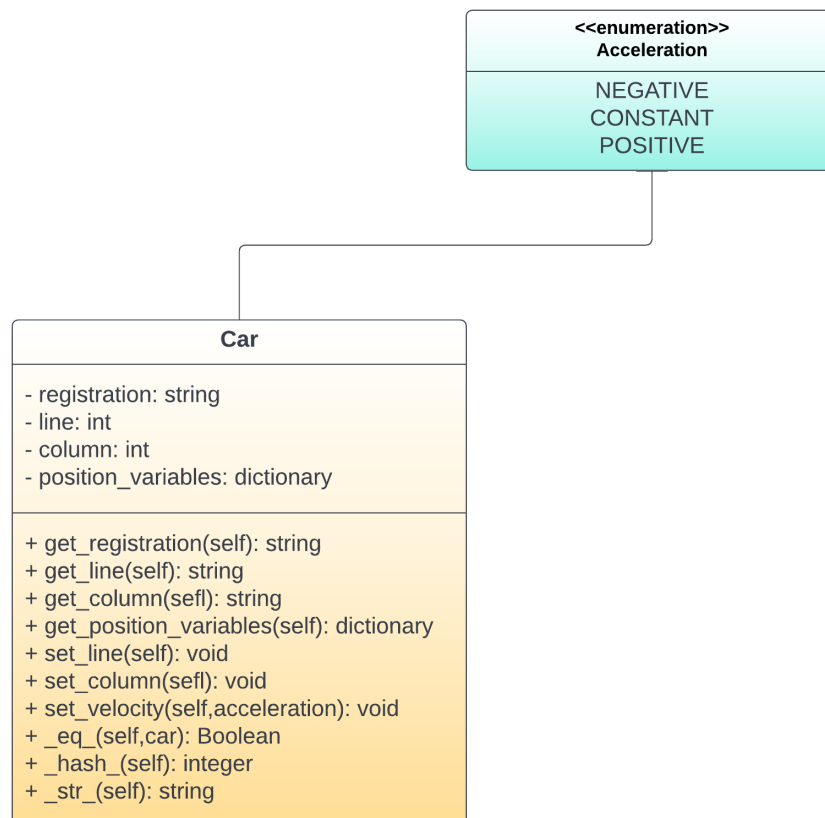


Figura 5: Classe *Car* e Enumerado *Acceleration*.

Um carro é caracterizado pela sua matrícula, por uma linha, por uma coluna e por um conjunto de variáveis que ditam a velocidade e aceleração do veículo relativamente às suas coordenadas (linha e coluna). O grupo considerou relevante definir um enumerado para a aceleração, uma vez que esta variável possui um conjunto de valores restrito. Em paralelo com a classe *Node*, também na classe *Car* as variáveis de instância só podem ser manipuladas através das funções *setters*.

4.3 Circuito

O módulo *circuit.py* é composto por um conjunto de funções, que são responsáveis pelo funcionamento e processamento do circuito.

Dentro do módulo referido, o grupo definiu fluxogramas para traduzir o raciocínio, o funcionamento e a lógica das funções que revelam ser mais complexas e sinuosas, nomeadamente as funções: *determine_circuit_costs* e *graph_from_circuit*.

A figura 6 ilustra o algoritmo da função *determine_circuit_costs* e a figura 7 ilustra o algoritmo da função *graph_from_circuit*.

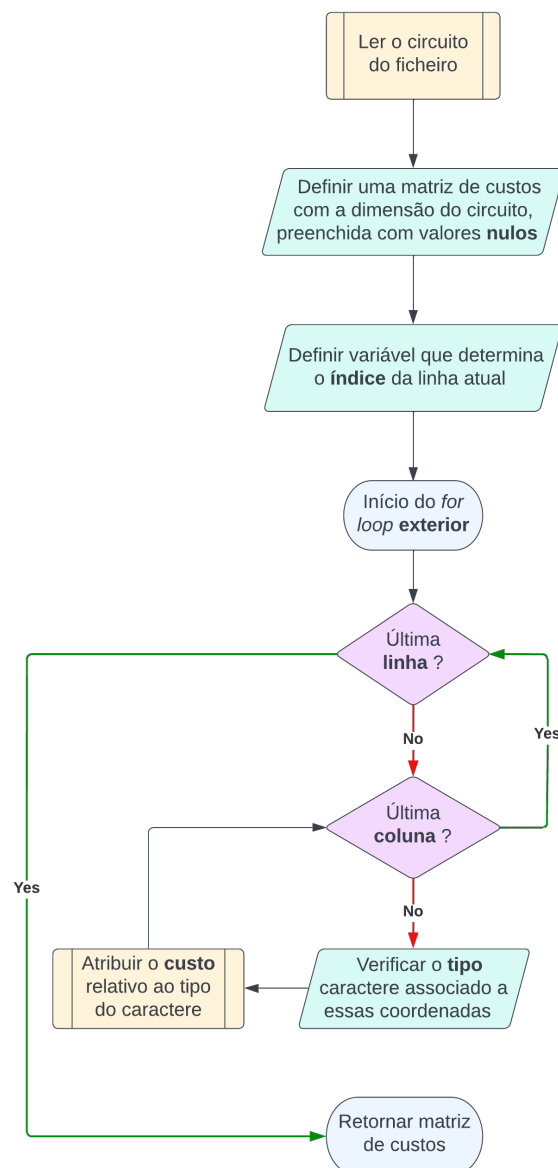
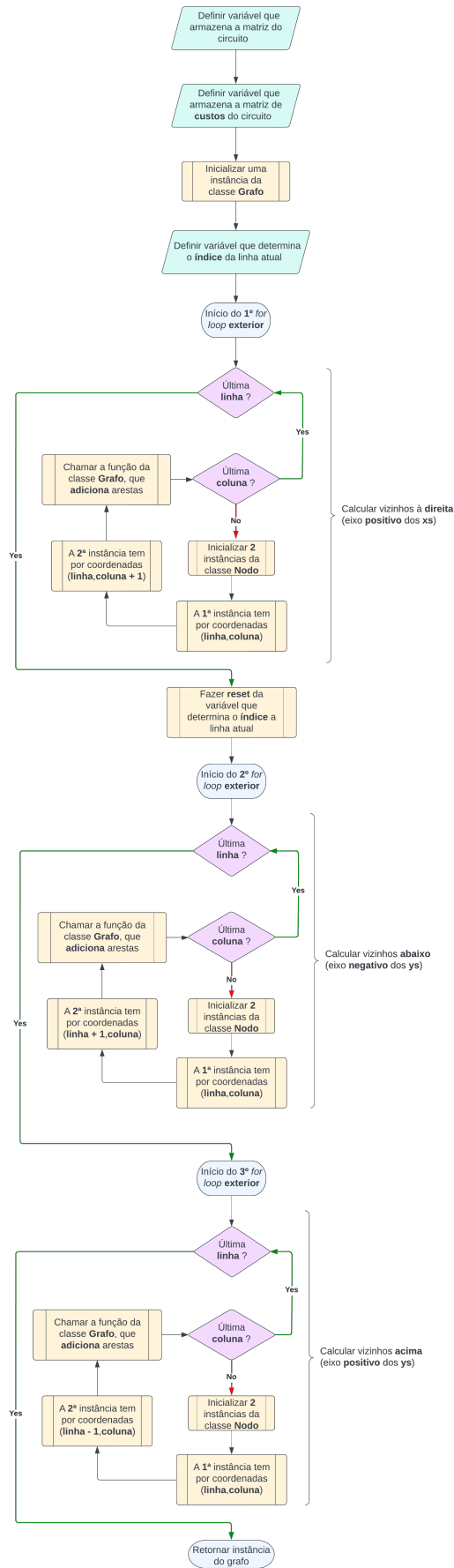


Figura 6: Algoritmo da função *determine_circuit_costs*.


 Figura 7: Algoritmo da função *graph_from_circuit*.

5. Análise de resultados

De acordo com o circuito presente no ficheiro *circuits.txt*, procedeu-se à comparação dos resultados obtidos, para os diferentes tipos de estratégias de procura utilizados, e à escolha da heurística, que foi introduzida pelo grupo nos algoritmos de procura informada.

A tabela 1 demonstra os resultados obtidos para uma posição inicial (P,3,1), e para uma posição final (F,2,19).

Tabela 1: Análise dos resultados obtidos para os diferentes algoritmos de procura.

(P,3,1)→(F,2,19)		
Algoritmo	Trajeto	Custo
DFS	(P,3,1), (-,3,2), (-,3,3), (-,4,3), (-,4,4), (-,4,5), (-,4,6), (-,4,7), (-,4,8), (-,4,9), (-,5,9), (-,5,10), (-,5,11), (-,5,12), (-,5,13), (-,5,14), (-,5,15), (-,5,16), (-,5,17), (-,4,17), (-,4,18), (-,3,18), (-,2,18), (F,2,19)	32
BFS	(P,3,1), (-,2,1), (-,2,2), (-,2,3), (-,2,4), (-,2,5), (-,2,6), (-,2,7), (-,2,8), (-,2,9), (-,3,9), (-,3,10), (-,3,11), (-,3,12), (-,3,13), (-,3,14), (-,2,14), (-,2,15), (-,2,16), (-,2,17), (-,2,18), (F,2,19)	30
Greedy	(P,3,1), (-,2,1), (-,2,2), (-,2,3), (-,2,4), (-,2,5), (-,2,6), (-,2,7), (-,2,8), (-,2,9), (-,3,9), (-,3,10), (-,3,11), (-,3,12), (-,3,13), (-,3,14), (-,4,14), (-,4,15), (-,5,15), (-,5,16), (-,5,17), (-,4,17), (-,3,17), (-,3,18), (-,2,18), (F,2,19)	34
A*	(P,3,1), (-,2,1), (-,2,2), (-,2,3), (-,2,4), (-,2,5), (-,2,6), (-,2,7), (-,2,8), (-,2,9), (-,3,9), (-,3,10), (-,3,11), (-,3,12), (-,3,13), (-,3,14), (-,2,14), (-,2,15), (-,2,16), (-,2,17), (-,2,18), (F,2,19)	30

O algoritmo de procura informada *Greedy*, é o algoritmo que possui o custo mais elevado, logo, não representa a solução ideal. A solução ideal é discutida entre os algoritmos BFS (*Breadth-First Search*) e A* (A Estrela), uma vez que possuem os custos mais baixos. Relativamente aos algoritmos de procura informada, a heurística adotada pelo grupo foi a minimização do custo, do percurso efetuado pelo veículo, entre o estado inicial e o estado objetivo.

6. Conclusão

Concluído este projeto, é inegável o carácter didático do mesmo.

O grupo encarou alguns atritos associados ao processamento do circuito e à construção do grafo. Graças à documentação da linguagem *Python* e de outros *websites*, que abordam a manipulação de matrizes e definição de grafos, foi possível contornar essas dificuldades.

É de realçar, que as aulas práticas foram decisivas no esclarecimento de dúvidas e na resolução dos problemas, que foram surgindo ao longo da construção deste projeto.

Em suma, é de referir a importância deste projeto no crescimento da capacidade e autonomia de aplicação, de conhecimentos adquiridos no decorrer da Unidade Curricular, até ao presente momento.