



DEEC
DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA E DE COMPUTADORES
TÉCNICO LISBOA

Licenciatura em Engenharia
Electrotécnica e de Computadores
(LEEC)

ALGORITMOS E ESTRUTURAS DE DADOS

ENUNCIADO DO PROJECTO



TOUR**I**/**T**s

Versão 4.0 (04/Setembro/2025 – primeira versão pública.)

2025/2026
1º Semestre

Conteúdo

1	Introdução	2
2	A aplicação TOUR <i>I</i> f <i>T</i> S	2
3	Fase intermédia do projecto	4
3.1	Formato de entrada	5
3.1.1	Ficheiro .map	5
3.1.2	O ficheiro .quests	6
3.2	Formato de saída	6
4	Fase final do projecto	7
4.1	Formato de entrada de dados	8
4.1.1	Ficheiro .cli	8
4.2	Formato de saída de dados	9
5	Execução do programa	10
6	Avaliação do Projecto	11
6.1	Gestor de Projecto	11
6.2	Prazos de entrega	12
6.3	Fases de avaliação	13
6.3.1	Oral de projecto	13
6.4	Prazos e comunicação de notas	14
6.5	Funcionamento	14
6.5.1	Código	15
6.5.2	Critérios de avaliação	15
6.5.3	Para uma utilização racional do site de submissões	16
7	Código de Honestidade Académica	17

1 Introdução

Este enunciado descreve as especificações do projecto da disciplina de Algoritmos e Estruturas de Dados que está decomposto em duas fases. A primeira fase consiste no desenvolvimento de algumas funcionalidades que, apesar de não determinarem em absoluto a solução final, poderão ter utilidade na implementação final da fase final. Em ambos os casos a aplicação deve ser escrita em linguagem C, cumprir todos objectivos descritos neste texto e ser computacionalmente eficiente (tempo e memória). A entrega do código fonte em ambas as fases é feita através de um sistema automático de submissão que verificará a sua correcção sintática e testará a execução do programa para algumas instâncias do problema.

Aconselha-se todos os alunos a lerem com a maior atenção o enunciado. Será obrigatória a adesão sem variações, ou interpretações livres ou discricionárias, a todas as especificações aqui descritas. A falha em cumprir alguma(s) especificação(ões) e/ou regra(s) constante(s) neste enunciado acarretará necessariamente alguma forma de desvalorização do trabalho apresentado. Por esta razão, tão cedo quanto possível e para evitar contratempos tardios, deverão os alunos esclarecer com o corpo docente qualquer aspecto que esteja menos claro neste texto, ou qualquer dúvida que surja após leitura atenta e cuidada deste enunciado.

2 A aplicação TOURIfTs

Neste projecto pretende-se desenvolver um programa para determinação de rotas e meios de transporte para que um qualquer turista possa viajar de uma qualquer cidade de partida até ao destino desejado.

Os pontos de interesse (cidades de partida, de chegada e eventualmente de passagem) estão definidos numa rede de ligações, na qual se estabelecem o conjunto de cidades a considerar, a(s) ligação(ões) que existe(m) entre pares de cidades, bem como o(s) tipo(s) de transporte que as liga(m).

Para cada ligação estão definidos o tempo de duração da viagem entre as duas cidades ligadas, bem como o custo dessa viagem para o cliente da aplicação TOURIfTs.

O programa pretendido deverá fornecer para cada cliente a sequência de ligações e o respectivo meio de transporte que permita fazer a viagem entre a cidade de partida e a de chegada de forma óptima, de acordo com o critério especificado pelo cliente: tempo ou custo.

A aplicação TOURIfTs possui um menu de destinos ligados por um ou mais meios de transporte. Neste projecto consideram-se quatro meios de transporte: autocarro, comboio, avião e barco. A Figura 1 mostra um exemplo de um mapa de uma rede de destinos com as ligações (e respectivo meio de transporte) existentes. Neste exemplo, entre Lisboa e Santarém existe apenas uma ligação através de autocarro. Já entre Faro e Beja existem duas ligações, uma por comboio e a outra por autocarro.

Note-se por exemplo que para ir do Porto a Madrid existem várias rotas possíveis: ligação directa por comboio ou autocarro, via Chaves (comboio ou autocarro) e depois Madrid necessariamente de avião, e ainda via Aveiro, Faro (barco ou comboio), Lisboa (os 4 meios), e finalmente Madrid (comboio ou avião).

A escolha de qual destas rotas entre o Porto e Madrid é a melhor depende da preferência do cliente: se pretende minimizar o tempo de viagem ou o custo da mesma. Para esse efeito, existem associados a cada ligação o respectivo tempo e custo da viagem.



Figura 1: Exemplo de um mapa oferecido pela aplicação TOURITS.

As ligações têm uma determinada periodicidade ao longo de um dia. Para este efeito, considera-se que um dia tem uma duração de 1440 minutos, sendo o minuto a unidade de tempo padrão, nunca se considerando fracções da mesma.

Assim, na definição de uma ligação, para além do tempo e custo da viagem, definem-se três parâmetros adicionais:

- o instante de tempo t_i em que sai o primeiro transporte dessa ligação - um valor entre 0 e 1440;
- o instante de tempo t_f a partir do qual não sai mais nenhum transporte dessa ligação - um valor entre 0 e 1440 que satisfaça $t_f \geq t_i$, e
- a periodicidade do transporte p .

Neste projecto, considera-se que as ligações são bidireccionais, ou seja o transporte que existe entre A e B também existe entre B e A , e os parâmetros definidos para a ligação são iguais quer a viagem seja entre A e B quer entre B e A .

Como exemplo, suponha-se que numa ligação entre as cidades A e B os parâmetros são $t_i = 0$, $t_f = 1010$ e $p = 200$. Isto significa que para essa ligação, num dia, há transportes a saírem da cidade A para a cidade B , e da cidade B para a cidade A , nos instantes $t = 0$, $t = 200$, $t = 400$, $t = 600$, $t = 800$, e $t = 1000$. Note-se que esta situação não significa que a viagem tem que durar menos de 200, porque se assume que a frota disponível é infinita. Ou seja, não é necessário considerar que um transporte que vá de A para B , a seguir regressa a A .

Outro exemplo de parametrização de uma ligação: $t_i = 360$, $t_f = 1080$ e $p = 800$. Neste caso, há apenas um transporte por dia que sai no instante 360. Ou seja, sempre que numa ligação sucede que $p > t_f - t_i$, a ligação em causa tem apenas um transporte diário. Para este caso, o transporte seguinte será apenas no dia seguinte, i.e. no instante $1440 + 360 = 1800$.

Adicionalmente, os clientes podem colocar algumas restrições relativamente às ligações que aceitam para construir a rota desejada. Neste projecto, devem-se considerar as seguintes restrições (**tipo A**), mas apenas uma para cada cliente:

- A.1 indicação de não mais do que um meio de transporte que o cliente **não quer** utilizar;
- A.2 tempo máximo para uma ligação;
- A.3 custo máximo para uma ligação.

As restrições acima condicionam as ligações. Como exemplo, considere-se um cliente que estabelece como restrição que não quer andar de avião. Isto significa que a rota a determinar para este cliente não deve incluir nenhuma ligação por avião; mas se o cliente definir como restrição que o tempo máximo por ligação é de 1000, então a rota pode ter um tempo total superior a 1000, mas cada uma das ligações que a constituem não podem ter um tempo de viagem superior a 1000. Idêntico raciocínio para o custo.

Finalmente, ainda para cada cliente, este pode definir uma restrição adicional (**tipo B**):

- B.1 se escolher como preferência (critério de optimização) o tempo, o cliente pode estabelecer que não aceita rotas cujo tempo total seja superior a um valor especificado; ou
- B.2 se escolher como preferência (critério de optimização) o custo, o cliente pode estabelecer que não aceita rotas cujo custo total seja superior a um valor especificado.

O objectivo do projecto é determinar para um conjunto de clientes da aplicação TOURISTS, que indicam a cidade de partida, a cidade de chegada, o critério de preferência, e eventualmente uma restrição do tipo A e/ou uma restrição do tipo B, qual a melhor rota entre as duas cidades, identificando as ligações que a constituem (i.e., qual o meio de transporte para cada uma).

3 Fase intermédia do projecto

Nesta secção explicitam-se os objectivos, especificações e funcionalidades que deverão estar operacionais na primeira fase de submissões. Nesta fase devem ser implementadas as seguintes funcionalidades:

1. Task1: Dadas duas cidades, contabilizar quantos meios de transporte diferentes permitem uma viagem directa entre elas;
2. Task2: Dadas duas cidades, determinar qual o menor tempo de viagem entre elas para todos os meios de transporte que as liguem directamente;
3. Task3: Dadas duas cidades, determinar qual o mais baixo custo de viagem entre elas para todos os meios de transporte que as liguem directamente;
4. Task4: Dadas duas cidades e uma hora do dia qual a hora mais cedo a que se pode chegar de uma à outra, desde que possuam alguma ligação directa;
5. Task5: Dadas duas cidades que não possuam qualquer ligação directa, indicar se existe uma terceira cidade que permita um caminho entre elas, independentemente dos meios de transporte usados em cada uma das ligações.

3.1 Formato de entrada

Na primeira fase de submissões haverá dois ficheiros de entrada. O primeiro, de extensão `.map`, identifica o menu disponível em termos de destinos, meios de transporte, tempos de viagem, custos e horários. O segundo, extensão `.quests`, concretiza uma série de indagações sobre o primeiro de acordo com as tarefas anteriormente explicitadas.

```
12 21
4 3 comboio 60 10 0 1440 100
4 1 comboio 100 14 500 1200 500
4 3 autocarro 70 9 600 1440 300
1 4 aviao 10 30 480 1440 1440
4 1 barco 90 21 0 1000 200
4 1 autocarro 90 15 0 1000 200
1 5 autocarro 20 5 0 1000 200
2 6 aviao 80 14 0 1000 200
7 1 aviao 200 100 0 1000 200
1 7 comboio 1111 75 0 1000 200
4 8 barco 220 93 0 1000 200
4 8 comboio 132 61 0 1000 200
9 8 barco 22 9 0 1000 200
9 8 comboio 13 13 0 1000 200
9 10 autocarro 5 2 0 1000 200
11 9 comboio 19 9 0 1000 200
11 9 autocarro 18 8 0 1000 200
9 7 comboio 302 101 0 1000 200
7 9 autocarro 1221 90 0 1000 200
11 7 aviao 2001 110 0 1000 200
5 6 barco 5 5 0 1000 200
```

Figura 2: Exemplo do ficheiro `.map` para a rede apresentada na Figura 1.

3.1.1 Ficheiro `.map`

A descrição da rede de cidades é feita num ficheiro com extensão `.map`, obedecendo às seguintes regras:

- Um cabeçalho contendo dois números inteiros, N e L , em que N representa o número de cidades e L o número de ligações.
- Após o cabeçalho haverá, garantidamente, a informação associada com as L ligações existentes. Cada ligação contém oito elementos separados por, pelo menos¹, um espaço em branco:
 - dois números inteiros correspondentes à identificação das duas cidades ligadas;
 - o meio de transporte da ligação, que pode ser uma das seguintes palavras: **aviao**, **comboio**, **barco** ou **autocarro**;
 - o tempo de duração da ligação;

¹O espaçamento entre elementos pode ser qualquer coisa, como mais que um espaço em branco, tabulações ou mudanças de linha. O procedimento de leitura não deve preocupar-se com estas separações. Basta-lhe saber que cada ligação é composta por uma sequência de 2 inteiros, uma string e mais 5 inteiros.

- o custo da ligação;
- o instante da primeira partida num dia (t_i);
- o instante do dia após o qual não haverá mais partidas (t_f); e
- a periodicidade diária (p).

Para a definição das cidades de uma ligação, considera-se que as cidades estão numeradas de 1 a N .

Nota: todos os tempos e custos referidos atrás são representados por números inteiros.

A título de exemplo, o ficheiro que descreve a rede² da Figura 1, é o indicado na Figura 2. Considerou-se a seguinte numeração das cidades: 1 - Lisboa, 2 - Évora, 3 - Beja, 4 - Faro, 5 - Santarém, 6 - Guarda, 7 - Madrid, 8 - Aveiro, 9 - Porto, 10 - Braga, 11 - Chaves, 12 - Dili.

3.1.2 O ficheiro .quests

A descrição dos problemas que se pretende resolver é feita num ficheiro de extensão `.quests`, em que cada problema é caracterizado por uma string e dois inteiros, excepto para a **Task4** que requer 3 inteiros. A string identifica a tarefa e os dois inteiros as cidades às quais se pretende aplicar a tarefa explicitada. Quando tenha de existir, o terceiro inteiro representa um instante de tempo restringido a valores entre 0 e 1440. Na Figura 3 apresenta-se um exemplo do que pode ser o conteúdo de um ficheiro de entrada para a fase intermédia.

```
Task1 4 3
Task5 1 2
Task3 12 8
Task2 4 13
Task4 1 4 500
```

Figura 3: Exemplo do ficheiro `.quests` para a rede apresentada na Figura 1.

Para cada problema existente no ficheiro de extensão `.quests`, a aplicação **TOURIfTs** deverá produzir uma resposta. Havendo P problemas nele, uma solução correcta terá de conter P respostas de acordo com o formato de saída que se explicita em seguida.

3.2 Formato de saída

Cada invocação da aplicação **TOURIfTs** terá se produzir como resultado um ficheiro de extensão `.results`. Qualquer problema para todas as variantes admite uma resposta numérica ou, em certas circunstâncias, uma string. Uma resposta para cada problema tem sempre que repetir o seu enunciado e indicar qual o resultado desse problema a seguir ao enunciado.

Existem dois tipos de problemas para qualquer das 5 tarefas: problemas admissíveis e problemas não admissíveis. Um problema é admissível se e somente se a string inicial for uma de cinco (**Task1**, **Task2**, **Task3**, **Task4**, **Task5**) e se os dois inteiros após essa string pertencerem ambos ao mapa que esteja a ser processado. Adicionalmente, na

²À qual se acrescenta a cidade de Dili, sem nenhuma ligação às restantes.

```
Task1 4 3 2
Task5 1 2 1
Task3 12 8 infinito
Task2 4 13 -1
Task4 1 4 600
```

Figura 4: Exemplo do ficheiro `.results` para a rede apresentada na Figura 1 e para o ficheiro `.quests` apresentado na Figura 3.

Task4, para que o problema seja admissível é necessário que o terceiro inteiro, $t_0 \in [0, 1440[$.

Para a **Task1** as respostas possíveis em problemas admissíveis são 0, 1, 2, 3 ou 4. Para as **Task2**, **Task3** e **Task4** as respostas possíveis para problemas admissíveis são um inteiro positivo ou a string `infinito`. Esta última resposta possível nestas três tarefas acontece quando não existe qualquer ligação directa entre as duas cidades dadas.

A admissibilidade em **Task5** possui uma condição extra, a saber. Considera-se admissível um problema em que as duas cidades dadas não possuam qualquer ligação entre si. Nestas circunstâncias as respostas possíveis são 0, quando não existe uma cidade que as ligue, ou 1, no caso contrário.

Qualquer problema que seja não admissível tem como resposta a repetição do seu enunciado seguida do inteiro -1.

Se o ficheiro de extensão `.quests` se chamar `luisinho.quests`, então o ficheiro produzido pela aplicação, resultante do seu processamento, tem obrigatoriamente que chamar-se `luisinho.results`. Ou seja, é o nome do ficheiro de extensão `.quests` que determina o nome do ficheiro de extensão `.results`³.

4 Fase final do projecto

O programa a desenvolver deve conseguir ler a configuração de uma rede de cidades, contendo informação sobre as cidades e as ligações que existem, bem como a informação sobre os clientes, e gerar para cada cliente a rota que melhor serve os seus interesses.

Naturalmente, o programa deve estar também preparado para lidar com um cliente que pretende viajar entre duas cidades para as quais não existe qualquer rota, ou coloca restrições que são impossíveis de satisfazer.

Neste projecto não se deve assumir qualquer pressuposto que os dados fornecidos estejam em consonância com a realidade. Por exemplo, não é necessário que uma viagem de avião demore menos tempo e/ou custe mais que a mesma viagem feita por qualquer dos outros meios de transporte.

Nas secções seguintes descreve-se a forma como o programa deve ser invocado, a forma como a informação da rede e dos clientes é descrita e o formato que têm de ter os dados de saída, ou seja a forma de descrever as rotas encontradas.

³NOTA: Estava a ponderar inovar aqui e propor que o nome do ficheiro de saída fosse a concatenação do nome do `.map` com o nome do `.quests`, a que depois se acrescenta `.results`. Desta forma poderia aplicar os mesmos `.quests` a diferentes `.map` sem baralhar os nomes.

4.1 Formato de entrada de dados

Na fase final também são necessários dois ficheiros. O primeiro possui extensão `.map` e segue o mesmo formato anteriormente descrito. O segundo ficheiro possui extensão `.cli`.

4.1.1 Ficheiro `.cli`

A descrição dos clientes é feita num ficheiro com extensão `.cli`, obedecendo às seguintes regras:

- Existe um primeiro inteiro, C , que representa o número de clientes contidos no ficheiro.
- Após aquela informação existem C conjuntos de dados, que correspondem à informação de cada cliente da aplicação TOURISTS. Cada cliente contém seis elementos obrigatórios separados por, pelo menos, um espaço em branco:
 - um número inteiro correspondente ao número de cliente (um número entre 1 e C);
 - dois números inteiros correspondentes à identificação da cidade de partida e da cidade de chegada desejada;
 - um número inteiro não negativo com o instante mínimo de disponibilidade para a partida do cliente;
 - a preferência do tipo de rota, identificada por uma de duas palavras: **tempo** ou **custo**. A primeira palavra significa que o cliente pretende a rota de menor tempo, enquanto a segunda significa que pretende a rota de menor custo; e
 - o número de restrições do cliente (0, 1 ou 2).

Para além dos elementos obrigatórios, a informação correspondente a um cliente pode incluir uma ou duas restrições, i.e., um ou dois pares opcionais de elementos:

- um par constituído por uma das seguintes palavras: **A1**, **A2** ou **A3**, seguida de uma das seguintes palavras: `aviao`, `comboio`, `autocarro` ou `barco`, caso a primeira seja **A1**, ou um número inteiro caso a primeira seja **A2** ou **A3**.
- O outro par possível é constituído por uma das seguintes palavras: **B1** ou **B2**, seguida de um número inteiro.

Nota: todos os tempos e custos referidos atrás são representados por números inteiros.

A título de exemplo, apresenta-se na Figura 5 a informação relativa a um conjunto de 5 clientes.

```
5
1 1 4 0 tempo 1 A1 aviao
2 2 3 250 custo 1 B2 10000
3 1 12 90 custo 0
4 9 10 0 tempo 2 A2 500 B1 1440
5 5 9 1000 tempo 2 A3 200 B1 2000
```

Figura 5: Exemplo do ficheiro `.cli`.

No exemplo de ficheiro `.cli` apresentado na Figura 5, a informação associada aos clientes significa o seguinte:

- o primeiro cliente, que pretende ir da cidade 1 para a cidade 4 o mais rapidamente possível, está disponível para partir no instante 0, e tem como restrição a não utilização do avião,
- o segundo cliente, que quer ir da cidade 2 para a 3 da forma mais barata possível, está disponível no instante 250, e não aceita rotas cujo custo total seja superior a 10000,
- o terceiro cliente, que pretende ir da cidade 1 para a 12 da forma mais barata possível, está disponível para partir no instante 1550, e não tem qualquer restrição,
- o quarto cliente, que quer ir da cidade 9 para a 10 o mais rapidamente possível, está disponível para partir no instante 0, e não quer ligações cujo tempo de viagem seja superior a 500 nem rotas cujo tempo total seja superior a 1440, e finalmente
- o quinto cliente pretende ir da cidade 5 para a cidade 9 o mais rapidamente possível, pode partir no instante 1000, e não quer ligações cujo custo de viagem seja superior a 200 nem rotas cujo tempo total seja superior a 1000.

4.2 Formato de saída de dados

Na fase final, o resultado da execução do programa de TOURISTs consiste em determinar para cada cliente qual a rota que melhor satisfaz a sua preferência, em tempo ou custo.

A solução deve ser colocada num ficheiro de saída, cujo nome deve ser o mesmo do ficheiro de entrada `.cli` mas com extensão `.sol`. Este ficheiro deve ser criado e aberto pelo programa. Por exemplo, se o ficheiro de entrada de clientes se chama `input231.cli`, o ficheiro de saída deve chamar-se `input231.sol`.

O ficheiro de saída deverá ter C linhas, uma por cada cliente, com a seguinte informação, separada por um espaço em branco:

- o número do cliente, a rota determinada para o cliente, o tempo total da rota e o custo total da mesma. O tempo total da rota é a diferença entre o instante de chegada do cliente à cidade final e o instante de disponibilidade de partida da cidade inicial (ver definição do ficheiro `.cli`). O custo total da rota é simplesmente a soma dos custos das diferentes ligações que compõem a rota.
- a especificação da rota deve seguir o seguinte formato: começa sempre pelo número da cidade de partida (número inteiro de 1 a N), e termina no número da cidade de chegada. Entre estes dois números deve-se colocar, sempre separados por um espaço em branco, o meio de transporte (uma das seguintes palavras: aviao, comboio, barco ou autocarro) e o número da cidade para as várias ligações que constituem a rota.

Por exemplo, se uma rota entre a cidade 33 e 18 compreende as seguintes ligações: comboio entre a cidade 33 e a cidade 67, aviao entre a cidade 67 e a cidade 101, e barco entre a cidade 101 e a cidade 18, então a linha correspondente do ficheiro de saída é a seguinte⁴:

⁴O número 12 inicial corresponde ao número do cliente.

12 33 comboio 67 aviao 101 barco 18 1267 8197

Neste exemplo, a rota em causa tem um tempo total de 1267 e um custo total de 8197.

- no caso particular de não existir qualquer rota entre a cidade de partida e a cidade de chegada e/ou que satisfaça as restrições definidas por um cliente, a linha correspondente a esse cliente no ficheiro de saída deve ter apenas -1.
- qualquer cliente que define um pedido não admissível terá como resposta -2 (Ou então... não exploramos estas singularidades que já as há na primeira fase. Ou seja, todos os problemas em .cli são admissíveis: existem as cidades, é sempre custo ou tempo, e as restrições estão sempre bem.)

Considere a rede apresentada na Figura 1 e o ficheiro de entrada da Figura 5. Neste caso, um exemplo de ficheiro de saída é o seguinte:

```
1 1 autocarro 4 90 15
2 2 aviao 6 barco 5 autocarro 1 comboio 4 autocarro 3 1020 47
3 -1
4 9 autocarro 10 5 2
5 5 autocarro 1 autocarro 4 comboio 8 comboio 9 853 94
```

Figura 6: Exemplo do ficheiro de saída para a rede apresentada na Figura 1 e os clientes apresentados na Figura 5.

5 Execução do programa

O programa TOURISTS deverá ser invocado na linha de comandos da seguinte forma:

aed\$./tourists <nome1>.map <nome2>.quests – para a primeira fase.

aed\$./tourists <nome1>.map <nome2>.cli – para a fase final.

onde:

tourists designa o nome do ficheiro executável contendo o programa TOURISTS;

<nome1>.map indica o nome do ficheiro contendo a descrição da rede de cidades, de acordo com o formato indicado na Secção 3.1.1 e em que **<nome1>** pode ser uma qualquer string em conteúdo e tamanho.

<nome2>.quests indica o nome do ficheiro de problemas da primeira fase, de acordo com o formato indicado na Secção 3.1.2, em que **<nome2>** pode ser uma qualquer string em conteúdo e tamanho.

<nome2>.cli indica o nome do ficheiro contendo a informação dos clientes, de acordo com o formato indicado na Secção 4.1.1, em que **<nome2>** pode ser uma qualquer string em conteúdo e tamanho.

Os ficheiros utilizados para a descrição da informação de partida terão de ter a extensão `.map` para a rede de cidades, extensão `.quests` para as tarefas da fase intermédia e `.cli` para os clientes na fase final, mas poderão ter qualquer nome, nada sendo assumido para esse efeito.

Se se pedir a execução do programa indicando-se o nome de um ficheiro que não exista, ou que exista mas não possui a extensão adequada, ou sem indicar qualquer nome ou com número errado de argumentos, o programa deve simplesmente abortar a execução sem emitir qualquer mensagem. Sobre a produção de texto de saída, sublinha-se que a única forma aceitável e correcta que a aplicação tem para comunicar com o mundo exterior é sempre via ficheiros de saída. Se a invocação estiver errada, não há lugar à produção de qualquer ficheiro de saída e não há qualquer comunicação.

Assume-se que os ficheiros de entrada estão sempre correctos de acordo com o formato descrito anteriormente. O programa a desenvolver não tem de se preocupar com a detecção de erros nos dados de entrada.

6 Avaliação do Projecto

O projecto está dimensionado para ser feito por grupos de dois alunos, não se aceitando grupos de dimensão superior nem inferior. Para os alunos que frequentam o laboratório, o grupo de projecto não tem de ser o mesmo do laboratório, mas é aconselhável que assim seja.

Do ponto de vista do planeamento do projecto, os alunos deverão ter em consideração que o tempo de execução e a memória usada serão tidos em conta na avaliação do projecto submetido. Por essas razões, a representação dos dados necessários à resolução dos problemas deverá ser criteriosamente escolhida tendo em conta o espaço necessário, mas também a sua adequação às operações necessárias sobre eles.

Serão admissíveis para avaliação versões do programa que não possuam todas as funcionalidades, seja no que à primeira fase de submissões diz respeito, como para a fase final. Naturalmente que um menor número de funcionalidades operacionais acarretará penalização na avaliação final.

Quando os grupos de projecto estiverem constituídos, os alunos devem obrigatoriamente inscrever-se no sistema Fénix, no grupo designado como “Projecto de AED”, que será oportunamente criado. Para além desta inscrição, cada grupo terá de seleccionar um horário de Gestão de Projecto.

6.1 Gestor de Projecto

No ano lectivo presente dar-se-á continuidade à figura do Gestor de Projecto, que funcionou pela primeira vez no ano lectivo de 2024/2025.

Todos os grupos inscritos a projecto terão acesso a um dos docentes que desempenhará, com maior proximidade, as funções de gestor do seu projecto de AED. Serão disponibilizados 7 horários de 2 horas cada um e cada grupo terá de seleccionar um desses 7 horários. Durante as sete semanas do período lectivo, cada grupo terá de agendar, pelo menos, um total de quatro reuniões com o seu Gestor de Projecto nesse horário.

Em cada uma das reuniões, com duração de cerca de cinco minutos, o grupo deverá apresentar os seus planos de trabalho e relatórios de progresso, explicitando as suas propostas de abordagem ao problema definido no enunciado.

Compete ao Gestor de Projecto oferecer alguma orientação sobre as opções de cada grupo e ajudar a estruturar as várias etapas e a definir calendarizações para a conclusão das tarefas necessárias à concretização do projecto, ao mesmo tempo que avalia o grau de equilíbrio/homogeneidade desse grupo e qualidade do trabalho desenvolvido.

A participação nestas reuniões, e qualidade dessa participação, conta para a avaliação do projecto com um peso global de 10% da cotação do mesmo. Acrescenta-se que cada grupo deverá obrigatoriamente realizar duas reuniões por cada fase de submissões e deverá também garantir que a sua última reunião se realiza até ao dia 17 de Outubro, o mais tardar.

6.2 Prazos de entrega

As datas de entrega referentes aos vários passos da avaliação do projecto estão indicadas na Tabela 1.

O site para as submissões electrónicas estará disponível em datas e condições a indicar posteriormente na página da disciplina e serão aceites trabalhos entregues até aos instantes finais indicados na tabela já referida.

Os alunos não devem esperar qualquer extensão nos prazos de entrega, pelo que devem organizar o seu tempo no sentido de estarem em condições de entregar a versão final até aos instantes finais, embora devam tentar realizar submissões com alguma antecedência para que surpresas, como erros de compilação infantis ou outros, não atrapalhem a sua capacidade de submeter implementações funcionais.

Note-se que, na versão final, o projecto só é considerado entregue aquando da entrega da ficha de autoavaliação. As submissões electrónicas do código não são suficientes para concretizar a entrega. A ausência de ficha de autoavaliação é a forma que os alunos têm de informar os docentes que não pretendem que a sua implementação seja avaliada. Não se consideram admissíveis a avaliação implementações sem submissões na fase final ou que nela obtenham pontuação nula. Mais sobre isto na Secção 1.4.2.

Tabela 1: Datas importantes do Projecto

Data	Documentos a Entregar
1 de Setembro de 2025, 2 ^a feira	Enunciado do projecto disponibilizado na página da disciplina.
até 12 de Setembro de 2025 (18h)	Inscrição dos grupos no sistema Fénix.
26 de Setembro de 2025, 6 ^a feira (23:59)	1^a Etapa: – Conclusão da primeira fase de submissões
22 de Outubro de 2025, 4 ^a feira (23:59)	Conclusão da fase final de submissões
23 de Outubro de 2025, 5 ^a feira (18:00)	2^a Etapa – Submissão da ficha de autoavaliação (Via Fénix).
Mais tardar até 24 de Janeiro de 2026	3^a Etapa – Comunicação de notas ou convocatória para oral.
Mais tardar até 30 de Janeiro de 2026	4^a Etapa – Oraís de projecto.

6.3 Fases de avaliação

A avaliação do projecto decorre em três ou quatro etapas distintas. O primeira coincide com a primeira submissão electrónica, onde os projectos serão avaliados automaticamente com base na sua capacidade de cumprir as especificações e funcionalidades definidas na Secção 3 deste documento. Para esta fase existe apenas uma data limite de submissão (ver Tabela 1) e não há qualquer entrega de relatório. A segunda etapa corresponde à submissão electrónica do código na sua versão final e à entrega de uma ficha de autoavaliação em moldes a definir. A entrega desta ficha ratifica e lacra a submissão electrónica realizada.

Só serão avaliados projectos para os quais haja submissão de código e ficha de autoavaliação. Assim, caso um grupo não obtenha pontuação suficiente no conjunto das duas submissões para que a nota mínima seja atingível com significativa probabilidade deverá evitar submeter-se a avaliação. A forma de o fazer é não entregando a ficha de autoavaliação.

Na terceira etapa há uma proposta enviada pelo corpo docente que pode conter a indicação de convocatória para a discussão e defesa do trabalho ou uma proposta de nota para a componente de projecto. Caso os alunos aceitem a nota proposta pelo docente avaliador, a discussão não é necessária e a nota torna-se final. Se, pelo contrário, os alunos decidirem recorrer da nota proposta, será marcada uma discussão de recurso para data posterior. Sobre as orais de recurso devem os alunos estar conscientes que a contestação da proposta de nota só faz sentido se o relatório de avaliação enviado pelo docente avaliador explicitar observações sobre a implementação que a desvalorizam e não correspondem à verdade e/ou ser omissos na apreciação de aspectos que os alunos entendam relevantes à luz da grelha de avaliação.

A quarta, e última, etapa acontece apenas caso haja marcação de uma discussão, seja por convocatória do docente, seja por solicitação dos alunos. Nestas circunstâncias, a discussão é obrigatoriamente feita a todo o grupo, sem prejuízo de as classificações dos elementos do grupo poderem vir a ser distintas. A falta de um ou de ambos os alunos à oral convocada pelos docentes acarreta reprovação automática de quem faltar ou, eventualmente, de todo o grupo se se verificar que o trabalho foi feito por apenas um aluno embora o relatório seja do grupo, pese embora ser necessário justificar a(s) razão(ões) dessa falta.

6.3.1 Oral de projecto

Poderão ter de realizar oral de projecto os grupos em que se verifique uma diferença entre a classificação provisória do projecto e a obtida em laboratório superior a três valores a favor do projecto, ou grupos em que os dois elementos possuam uma diferença na classificação do laboratório superior a três valores ou uma diferença de desempenho no LAI superior ou igual a 8 valores, considerando que o LAI é avaliado em 20 valores. O Gestor de Projecto de cada grupo poderá propor dispensa de oral nestes casos, em função da avaliação que tenha feito ao longo do período lectivo, assim como poderá propor realização de oral a grupos que não cumpram aquelas condições, também com base na sua avaliação do desempenho individual dos alunos nas reuniões com o seu GP.

A discussão, a ser efectuada, será sempre feita em grupo embora possam ser colocadas questões distintas aos dois elementos de cada grupo e os mesmos possam vir a obter nota diferente no projecto, quando tal se justificar.

Outra razão para a convocatória de um grupo para oral decorre da eventual suspeita

de não ter sido observado o Código de Honestidade Académica.

6.4 Prazos e comunicação de notas

A avaliação dos projectos nesta disciplina é um processo complexo que, face ao volume estimado de grupos em avaliação, tendo também em conta as outras obrigações lectivas da equipa docente, é impossível avaliar da ordem das 150 implementações no espaço de uma semana. Esta é a razão que nos leva a propor um calendário de avaliação diferenciado para os vários grupos em avaliação.

Terminada a fase final de submissões, cada grupo possui uma pontuação que corresponde à soma dos pontos obtidos nas duas fases de submissão. Alguns dos grupos poderão encontrar-se em situação de incerteza relativa a estarem ou não em condições de receber uma nota na componente de Projecto superior ou igual à nota mínima (7.00 valores).

Por exemplo, um grupo que tenha uma nota devida a testes passados igual a 4.50 valores está nestas circunstâncias. Por outro lado e por exemplo também, um grupo que tenha uma marca de 7.40 valores decorrente dos testes deverá estar relativamente seguro de que a sua nota na componente de Projecto venha a ser superior à nota mínima.

Após a conclusão das submissões, a equipa docente fará uma segmentação das implementações que tenham submetido ficha de autoavaliação em três categorias.

1. Grupos que, de certeza, não têm condições de atingir a nota de 7.00 valores a projecto;
2. Grupos sobre os quais não é seguro que obtenham a nota mínima;
3. Grupos sobre os quais se possui relativa segurança de virem a obter nota mínima.

Sobre a primeira categoria, não havendo dúvidas sobre o resultado da avaliação (ver Secção 6.5.2), os próprios alunos sabem que não terão nota mínima a Projecto⁵.

A equipa docente dará prioridade à avaliação dos grupos que estejam na segunda categoria no sentido se lhes ser comunicada a nota durante a semana de preparação para os exames de primeira época. As implementações dos grupos da terceira categoria seguirá um ritmo mais lento.

A tradição desta disciplina tem também contado com a compreensão de parte dos alunos nela inscritos para um processo de avaliação dos projectos tão distendido no tempo.

Sublinha-se que, nas circunstâncias em que decorre o processo de submissões e com a informação dele retirada, não se pode comparar esta situação com casos em que à data da submissão de relatórios finais de projectos não exista qualquer informação que permita aos alunos saber em que faixa a sua nota poderá vir a situar-se, na medida em que em AED, como se verá, 65% da nota é obtenível nas submissões electrónicas.

6.5 Funcionamento

A verificação do funcionamento do código a desenvolver no âmbito do projecto será exclusivamente efectuada nas máquinas do laboratório da disciplina, embora o desenvolvimento possa ser efectuado em qualquer plataforma ou sistema que os alunos escolham. Esta regra será estritamente seguida, não se aceitando quaisquer excepções. Por esta razão, é

⁵Espera-se que os grupos que estejam nestas circunstâncias optem por não submeter ficha de autoavaliação.

essencial que os alunos, independentemente do local e ambiente em que desenvolvam os seus trabalhos, os verifiquem no laboratório antes de os submeterem, de forma a evitar problemas de última hora. Uma vez que os laboratórios estão abertos e disponíveis para os alunos em largos períodos fora do horário das aulas, este facto não deverá causar qualquer tipo de problemas.

6.5.1 Código

Não deve ser entregue código em papel. Os alunos devem entregar por via electrónica o código do programa (ficheiros `.h` e `.c`) e uma `Makefile` para gerar o executável. Todos os ficheiros (`*.c`, `*.h`, `Makefile`, etc.) devem estar localizados na directoria raiz.

O código deve ser estruturado de forma lógica em vários ficheiros (`*.c` e `*.h`). As funções devem ter um cabeçalho curto mas explicativo e o código deve estar correctamente indentado e com comentários que facilitem a sua legibilidade.

6.5.2 Critérios de avaliação

Os projectos submetidos serão avaliados de acordo com a seguinte grelha:

- Testes passados na primeira submissão electrónica – 200 pontos
- Testes passados na última submissão electrónica – 1100 pontos
- Ficha de autoavaliação – 500 pontos
 - Opções de projecto e complexidade – 300 pontos
 - Estruturação do código e comentários – 100 pontos
 - Gestão de memória e tipos abstractos – 100 pontos
- Avaliação do Gestor de Projecto – 200 pontos

Considera-se que uma implementação não possui qualidade suficiente para atingir a nota mínima de 7.00 valores (700 pontos) na componente de Projecto se a soma da pontuação directa das duas fases de submissão for estritamente inferior a 400 pontos. Ou seja, implementações que não atinjam a soma de 400 pontos nas duas fases de submissão pertencem à primeira das categorias anteriormente descritas. Tanto na primeira como na submissão electrónica final, cada projeto será testado com vários ficheiros de problemas de diferentes graus de complexidade, onde se avaliará a capacidade de produzir soluções correctas dentro de limites de tempo e memória. Para o limite de tempo, cada um dos testes terá de ser resolvido em menos de x segundos⁶.

Para o limite de memória, cada um dos testes não poderá exceder 100MB como pico de memória usada⁷. Cada teste resolvido dentro dos orçamentos temporal e de memória que produza soluções correctas recebe uma pontuação de acordo com o grau de dificuldade desse mesmo teste. Um teste considera-se errado se, pelo menos, um dos problemas do ficheiro de entrada correspondente for incorrectamente resolvido.

Se o corpo docente entender necessário, face à complexidade dos problemas a resolver, poderão os limites de tempo e/ou memória ser alterados.

⁶Para a primeira fase haverá um limite, y tal que $y < x$.

⁷O limite de memória será o mesmo em ambas as fases.

Caso o desempenho de alguma submissão electrónica não seja suficientemente conclusivo, poderá ser sujeita a testes adicionais fora do contexto da submissão electrónica. O desempenho nesses testes adicionais poderá contribuir para subir ou baixar a pontuação obtida na submissão electrónica.

No que à avaliação da ficha de autoavaliação diz respeito, os elementos importantes são: rigor no seu preenchimento, ou seja, se os alunos não explicarem as opções de acordo com o pedido na ficha isso será descontado ou se os alunos indicarem na ficha algo diferente do código submetido, isso será descontado; apreciação da qualidade da abordagem geral ao problema e respectiva implementação, face às alternativas disponíveis; e clareza e suficiência do texto, na sua capacidade de descrever e justificar com precisão algumas das opções.

Pela análise da grelha de avaliação aqui descrita, deverá ficar claro que a ênfase da avaliação se coloca na capacidade de um programa resolver correctamente os problemas a que for submetido. Ou seja, o código de uma submissão até pode ser muito bonito e bem estruturado e o grupo até pode ter despendido muitas horas no seu desenvolvimento. No entanto, se esse código não resolver um número substancial de testes na submissão electrónica ou se necessitar de um número substancial de submissões para demonstrar a sua correcção, dificilmente terá uma nota positiva.

6.5.3 Para uma utilização racional do site de submissões

Com o objectivo de se contribuir para que os alunos façam melhor uso do seu tempo no desenvolvimento das suas implementações, introduz-se um factor correctivo associado com o número de submissões de cada grupo. O site de submissões não pode, nem deve, ser encarado como um mecanismo de apoio ao desenvolvimento do projecto, na medida em que o feedback que fornece é muito limitado. Por outro lado, é entendimento da equipa docente que se deve valorizar a capacidade dos grupos em validarem as suas implementações fora do contexto do site de submissões. Por isso, dois projectos que resolvam todos os testes não valem o mesmo se um deles atingir essa marca em, por exemplo, 5 submissões e o outro atingir a mesma marca em 18.

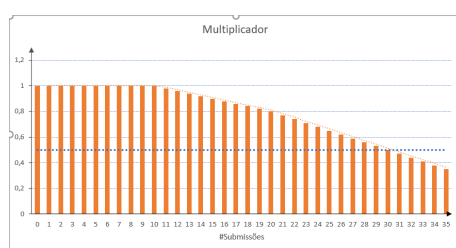


Figura 7: Multiplicador para cada uma das duas fases de submissão, como função do número total de submissões em cada uma.

Na Figura 7 apresenta-se um gráfico com o multiplicador associado ao número de submissões. Cada grupo dispõe de 10 submissões livres de penalização em cada uma das duas fases. As submissões livres servem para acomodar ligeiros percalços com erros de compilação ou erros na composição dos ficheiros a submeter. Se um grupo fizer mais que 10 submissões em alguma das fases, a pontuação que recebe nessa fase é penalizada em 2% por cada submissão adicional até à vigésima submissão. A partir da vigésima, cada submissão adicional é penalizada em 3%. Por exemplo, suponha-se que um grupo atinge a pontuação ilíquida de 180 pontos na primeira fase: recebe 180 pontos se tiver

feito 10 ou menos submissões; recebe 162 pontos se tiver realizado 15 submissões; e recebe 117 pontos caso concretize 25 submissões na primeira fase. Qualquer que venha a ser o número de submissões, avaliar-se-á apenas a que tiver obtido a melhor pontuação, independentemente de ser a última, a primeira ou uma intermédia, a menos que o grupo indique pretender que outra seja avaliada. Na eventualidade de existir empate na pontuação, também cabe ao grupo indicar, na ficha de autoavaliação, quais as suas duas submissões (uma por fase) pretende que sejam contabilizadas e avaliadas.

No sentido de contribuir para que cada grupo tire o melhor partido possível de cada submissão mal sucedida, a equipa docente disponibiliza um link por fase de submissões que permite aceder ao relatório detalhado da submissão. Ignorar a informação contida nesses relatórios não contribui para uma utilização racional do site de submissões⁸. Sendo plausível que alguns grupos tenham dificuldade em decodificar algumas das mensagens de erro, sobretudo as associadas com erros de compilação, espera-se que os grupos procurem por ajuda junto do corpo docente, no contexto dos horários de dúvidas ou outros, ao invés de teimarem em submeter de 10 em 10 minutos sem endereçarem verdadeiramente as causas dos erros que obtêm.

Para este projecto os links que estarão activos assim que o site de submissões for aberto serão:

1. <https://amarguinha.scdeec.tecnico.ulisboa.pt/aed/cgi-bin/checksub1.TourISTs.pl>
2. <https://amarguinha.scdeec.tecnico.ulisboa.pt/aed/cgi-bin/checksub2.TourISTs.pl>

7 Código de Honestidade Académica

Espera-se que os alunos conheçam e respeitem o Código de Honestidade Académica que rege esta disciplina e que pode ser consultado na página da cadeira. Lembramos igualmente que a verificação de potenciais violações a este código é feita de forma automática com recurso a sofisticados métodos de comparação de código, que envolvem não apenas a comparação directa do código mas também da estrutura do mesmo. Esta verificação é feita com recurso ao software disponibilizado em

<http://moss.stanford.edu/>

Em 2025/2026 em AED os alunos estão **PROIBIDOS** de usar código gerado por meios automáticos tipo LLM (ChatGPT, etc). Qualquer violação desta proibição constituirá, naturalmente, violação do Código de Honestidade Académica⁹.

⁸Há registos de grupos com meia dúzia de submissões consecutivas todas com "Erro de Compilação" para o mesmo erro.

⁹Sem CHA não há bolinhos.