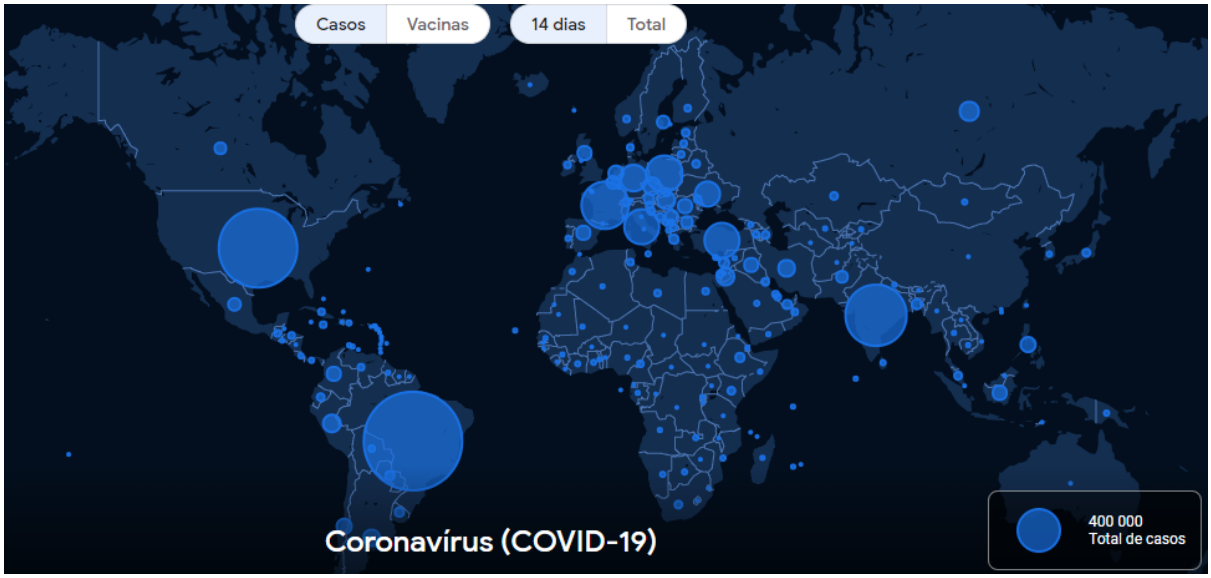
DEEC
DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES
TÉCNICO LISBOA

PROGRAMAÇÃO

(MEEC 20/21)

Enunciado do Projecto

Fase II: Projecto Final



Pandemia

1. Introdução

O trabalho que se descreve corresponde ao projeto final da UC de Programação para o MEEC em 2020/2021. O projeto a realizar corresponde à implementação de mecanismos de organização, manipulação e análise de um conjunto de dados de elevada dimensão. O objetivo do projeto é promover e avaliar a prática dos conceitos e técnicas de programação estudadas no âmbito desta UC. O projeto deverá ser realizado ao longo do semestre acompanhando a matéria teórica lecionada nesta UC. Nas secções que se seguem descrevem-se os dados a considerar e as funcionalidades que devem obrigatoriamente ser implementadas. Na secção de avaliação descreve-se como deve ser feita a entrega e como será realizada a avaliação do projeto.

2. Descrição Geral do Projecto

Pretende-se desenvolver um programa que leia dados reais sobre a evolução da pandemia em todos os países e realize um conjunto de análises definidas pelo utilizador. De acordo com a Fig. 1, o programa deve receber a especificação da análise a realizar via linha de comando, deve ainda ler dados diários e semanais de ficheiros cujo formato será detalhado adiante. As análises produzidas serão para o stdout ou para ficheiro conforme descrito nas secções seguintes.

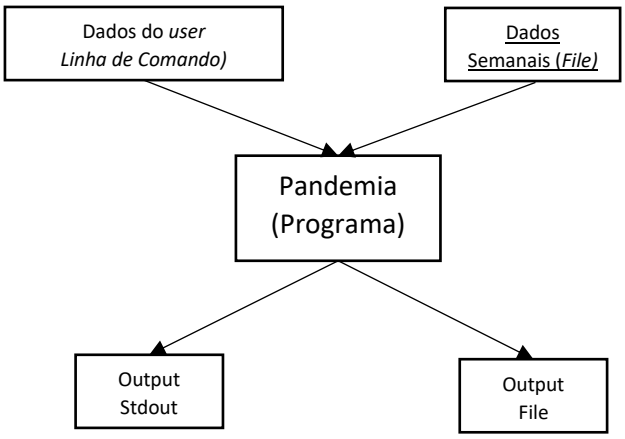


Fig. 1: Diagrama de blocos da interface do programa a desenvolver.

3. Descrição do Formato dos Ficheiros de Entrada

Nesta secção, descreve-se o formato do ficheiro de texto a considerar. Mais, juntamente com o enunciado são disponibilizados exemplos destes ficheiros que devem ser considerados para teste durante o desenvolvimento do projeto. No final, os projetos serão avaliados com os ficheiros de teste disponibilizados e com outros ficheiros com as mesmas características, isto é, com o mesmo formato, mas podendo ter menos ou mais linhas de dados. Portanto, todo o código a desenvolver deve ser robusto de modo a conseguir ler ficheiros de outra dimensão e a processar os respetivos dados.

O ficheiro de dados fornecido tem dados semanais relativos à evolução da pandemia de COVID-19. Os dados semanais são relativos a cada país. De seguida apresenta-se o detalhe dos dados e o formato do ficheiro a considerar.

3.1. Descrição dos Dados Semanais de Infetados e Mortes de COVID-19

O ficheiro de entrada terá dados semanais organizados em 9 colunas[1], conforme ilustrado na fig. 2, com a seguinte informação:

- Coluna 1 (*country*): Nesta coluna, está indicado o nome do país. O nome dos países pode conter apenas uma palavra ou várias palavras, por exemplo, "Antigua and Barbuda".
- Coluna 2 (*country_code*): Nesta coluna, indica-se o código do país.
- Coluna 3 (*continent*): Nesta coluna, indica-se o continente a que pertence cada país.
- Coluna 4 (*population*): Nesta coluna, indica-se a população do país.
- Coluna 5 (*indicator*): Nesta coluna, especifica-se através das palavras "cases" ou "deaths" o significado dos valores das colunas 6, 8 e 9. No caso da palavra "cases" os valores indicados nas restantes colunas referem-se a infectados por COVID-19 e no caso da palavra "deaths" os valores referem-se a mortes por COVID-19.
- Coluna 6 (*weekly_count*): Nesta coluna, indica-se o número de infectados ou mortes, de acordo com a palavra indicada na coluna 5, para a semana indicada na coluna 7.
- Coluna 7 (*year_week*): Nesta coluna, indica-se a semana a que se referem os dados. A semana é identificada no formato yyyy-ww, onde yyyy refere-se ao ano e ww à semana nesse ano, por exemplo, 2010-10 refere-se à 10ª semana de 2010.
- Coluna 8 (*rate_14_day*): Nesta coluna, indica-se o rácio de infectados nos últimos 14 dias (2 semanas) por 100 mil habitantes, ou o rácio de mortes por milhão de habitantes nos últimos 14 dias. A interpretação do valor depende da palavra indicada na coluna 5. A inexistência de dados para países e continentes nesta coluna tem o significado de dados não disponíveis.
- Coluna 9 (*cumulative_count*): Nesta coluna, apresenta-se o valor acumulado de infectados ou mortes até à semana indicada. A interpretação do valor depende da palavra indicada na coluna 5.

Nota, a 1ª linha do ficheiro tem a designação de cada coluna, respetivamente, *country, country_code, continent, population, indicator, weekly_count, year_week, rate_14_day, cumulative_count*.

3.2. Exemplos de Dados Semanais de Infetados e Mortes de COVID-19

Na fig. 2 ilustra-se parte do ficheiro de texto, aberto no Excel. Nesta imagem pode-se ver as 9 colunas descritas na secção anterior e, assim, conferir a descrição dos dados presentes em cada coluna.

country	country_code	continent	population	indicator	weekly_count	year_week	rate_14_day	cumulative_count
Andorra	AND	Europe	76177	cases	5	2020-11		5
Andorra	AND	Europe	76177	cases	108	2020-12	148,3387374	113
Andorra	AND	Europe	76177	cases	221	2020-13	431,8888903	334
Andorra	AND	Europe	76177	cases	167	2020-14	509,3400895	501
Andorra	AND	Europe	76177	cases	137	2020-15	399,0705856	638
Andorra	AND	Europe	76177	deaths	0	2020-11		0
Andorra	AND	Europe	76177	deaths	0	2020-12	0	0
Andorra	AND	Europe	76177	deaths	6	2020-13	78,76393137	6
Andorra	AND	Europe	76177	deaths	12	2020-14	236,2917941	18
Andorra	AND	Europe	76177	deaths	11	2020-15	301,9284036	29
Angola	AGO	Africa	32866268	cases	2	2020-12		2
Angola	AGO	Africa	32866268	cases	5	2020-13	0,021298433	7
Angola	AGO	Africa	32866268	cases	7	2020-14	0,036511599	14

Fig. 2: Amostra de dados visualizada em Excel.

[1] O ficheiro de entrada tem o formato CSV (Comma-Separated Values) que aberto no Excel apresenta os valores organizados em colunas e como ficheiro de texto os valores separados por vírgulas.

Na fig. 3 apresenta-se um ficheiro em formato de texto utilizando a vírgula como separador entre colunas (CSV). Este será o formato que deverá ser considerado para ler os dados ao abrir o ficheiro de texto. De notar que a primeira linha contém a identificação das colunas e não dados.

```
country,country_code,continent,population,indicator,weekly_count,year_week,rate_14_day,cumulative_count
Afghanistan,AFG,Asia,38928341,cases,0,2020-01,,0
Afghanistan,AFG,Asia,38928341,cases,0,2020-02,0,0
Afghanistan,AFG,Asia,38928341,cases,0,2020-03,0,0
Afghanistan,AFG,Asia,38928341,cases,0,2020-04,0,0
Afghanistan,AFG,Asia,38928341,cases,0,2020-05,0,0
Afghanistan,AFG,Asia,38928341,cases,0,2020-06,0,0
Afghanistan,AFG,Asia,38928341,cases,0,2020-07,0,0
Afghanistan,AFG,Asia,38928341,cases,0,2020-08,0,0
Afghanistan,AFG,Asia,38928341,cases,1,2020-09,0.002568823,1
Afghanistan,AFG,Asia,38928341,cases,3,2020-10,0.01027529,4
```

Fig. 3: Amostra do ficheiro de texto no formato CSV.

4. Descrição das Funcionalidades a Implementar

Nesta secção, descrevem-se as funcionalidades a implementar neste trabalho, as quais serão especificadas através da linha de comando conforme descrito mais adiante neste enunciado.

4.1. Representação dos Dados

Os dados devem ser lidos do ficheiro acima descrito (inicialmente será fornecido um ficheiro de exemplo, mas o projeto será testado com diferentes ficheiros no mesmo formato). Os dados dos países **devem ser armazenados numa lista criada dinamicamente**. Os dados a considerar devem ser organizados por cada país (conforme descrito na coluna 1 do ficheiro) e devem, obrigatoriamente, conter a seguinte informação:

- **dados fixos**
  - nome do país (dado da coluna 1)
  - sigla do país (dado da coluna 2)
  - continente (dado da coluna 3)
  - população (dado da coluna 4)
- **dados variáveis**
  - semana e ano a que respeitam os dados (dado da coluna 7).
  - número de infectados na semana (dados da coluna 6, quando coluna 5 contém “cases”).
  - número de mortes na semana (dados da coluna 6, quando coluna 5 contém “deaths”).
  - rácio de infectados por 100 mil habitantes (dados da coluna 8, quando coluna 5 contém “cases”).
  - rácio de mortes por milhão de habitantes (dados da coluna 8, quando coluna 5 contém “deaths”).
  - número de acumulado de infectados (dados da coluna 9, quando coluna 5 contém “cases”).
  - número de acumulado de mortes (dados da coluna 9, quando coluna 5 contém “deaths”).

A definição dos nós (ou elementos) da lista ou listas criadas para representar os dados é uma opção de projeto de cada grupo e deve ser devidamente justificada. Contudo, a lista principal deve ter apenas um nó para cada país presente na coluna 1.

4.2. Opções de Leitura de Dados

A leitura de dados do ficheiro deve ser feita de modo selectivo de acordo com as seguintes opções (tendo em consideração os valores da coluna 1 e 3):

- -L all
    - Leitura integral do ficheiro
- -L *continente nome_do_continente*
    - Apenas lê os dados relativos aos países do continente especificado com a opção.

Em todos os casos, acima descritos, devem ser lidos todos os dados de acordo com a opção e armazenados de modo a conterem a informação descrita em 4.1.

4.3. Opções de Ordenação de Dados

Devem ser implementadas as seguintes opções de ordenação sobre os dados lidos do ficheiro.

- -S alfa
    - ordem alfabética de países
- -S pop
    - ordem decrescente de população de países
- -S inf yyyy-ww
    - ordem decrescente do número total de infetados por países numa determinada data yyyy-ww
- -S dea yyyy-ww
    - ordem decrescente do número total de mortes por países numa determinada data yyyy-ww

Nota: a ordenação da lista deve manter a estrutura de dados de cada nó da lista original. Em caso de "empate" no fator de ordenação deve-se sempre considerar como segundo critério a ordenação por ordem alfabética do nome do país.

4.4. Opções de Seleção de Dados

Deve ser possível selecionar dados de acordo com as seguintes opções:

- -D inf
    - selecionar para cada país a semana com mais infectados
- -D dea
    - selecionar para cada país a semana com mais mortes
- -D racioinf
    - selecionar para cada país a semana com maior rácio de infectados por 100000 habitantes.
- -D raciodea
    - selecionar para cada país a semana com maior rácio de mortes por milhão de habitantes.

4.5. Opções de Restrição de Dados

Deve ser possível restringir as análises das restantes opções a países que cumpram as seguintes restrições:

- -P min *n*
    - apenas dados de países com mais de n mil habitantes (sendo n um inteiro)
- -P max *n*
    - apenas dados de países com menos de n mil habitantes (sendo n um inteiro)
- -P date yyyy-ww
    - apenas dados relativos à semana indicada
- -P dates yyyy-ww yyyy-ww
    - apenas dados entre as semanas indicadas (**a ordem pela qual são especificadas a semana inicial e final deve ser irrelevante**)

4.6. Opções de Leitura e Escrita em Ficheiros

Deve ser possível ler e escrever em ficheiros de texto (formato csv) e ficheiros binários de acordo com as seguintes opções.

- -i *filename*.csv
    - leitura de dados de ficheiro de texto
- -i *filename*.dat
    - leitura de dados de ficheiro binário (compatível com a opção de escrita em ficheiros de dados desenvolvida). Esta opção funciona apenas para importação de dados sem qualquer especificação de outra opção do tipo -L, -S, -D ou -P.
- -o *filename*.csv
    - escrita de dados em ficheiro de texto
- -o *filename*.dat
    - escrita de dados em ficheiro binário. A forma como são armazenados os dados é uma opção de desenvolvimento (garantido que o que é escrito corresponde à forma como os dados estão codificados em memória e não em formato de texto) e deve ser compatível com a opção de leitura do ficheiro .dat .

Todas as situações não especificadas podem ser consideradas opções de projecto.

5. Modos de Execução do Programa

Nesta secção, descrevem-se os modos de execução do programa de acordo com as funcionalidades implementadas. Para efeitos de avaliação, as **entradas** serão especificadas em **linha de comando** e as **saídas** deverão ser escritas **em ficheiro**.

5.1. Especificação dos Ficheiros de Entrada e Saída

O ficheiro de texto com os dados de entrada e o ficheiro de saída devem ser especificados, respectivamente, com as opções -i e -o, conforme descrito na secção 4.

5.2. Especificação das Opções Funcionais

As opções funcionais consistem nas opções descritas na secção 4 deste enunciado. As opções funcionais, tais como as opções para especificação do ficheiro de entrada e de saída, podem surgir em qualquer ordem na linha de comando.

5.3. Exemplos de Invocação do Programa

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

prog$ ./covid19 [OPTIONS]

No caso de não serem especificadas quaisquer opções ou de não serem especificadas as opções obrigatórias (descritas de seguida) o programa deve enviar para o terminal a descrição das várias opções e qual o seu significado.

A invocação do programa para permitir a execução deve obrigatoriamente incluir as opções -i e -o. As restantes opções podem ou não estar presentes. No caso de não serem especificadas devem assumir os seguintes valores por omissão:

- Valor por omissão para -L deve ser -L all
- Valor por omissão para -S deve ser -S alfa
- Na ausência de especificação da opção -D, devem ser considerados todos os dados que resultem da aplicação das outras opções
- Na ausência de especificação da opção -P, devem ser considerados todos os dados que resultem da aplicação das outras opções

Alguns exemplos de invocação na linha de comando:

- prog$ ./covid19 -i f_texto1.csv -o f_dados.dat
  - o ficheiro de texto deve ser lido e criado um ficheiro binário exactamente com a mesma informação
- prog$ ./covid19 -i f_dados.dat -o f_texto2.csv
  - o ficheiro binário criado deve ser lido e convertido num ficheiro de texto com o formato do ficheiro de texto fornecido inicialmente
- prog$ ./covid19 -i f_texto1.csv -L Africa -S pop -D dea -P min 100 -o f_texto2.csv
  - Devem ser lidos os dados do ficheiro de texto f_texto1.csv relativos aos países de Africa. Os dados devem considerar as linhas relativas a mortes em zonas com mais de 100 mil habitantes. Os dados devem ser ordenados por ordem decrescente de população. Por fim, o conjunto de dados obtido deve ser escrito no ficheiro de texto f_texto2.csv.
- prog$ ./covid19 -i f_texto1.csv -L all -S inf 2020-27 -D inf -P max 5000 -o f_dados.dat
  - Devem ser lidos os dados do ficheiro de texto f_texto1.csv. Os dados devem considerar as linhas relativas a infectados em zonas com um máximo de 5 milhões de habitantes. Os dados devem ser ordenados por ordem decrescente de infectados na semana 2020-27. Por fim, os resultados devem ser escritos no ficheiro binário f_dados.dat
- prog$ ./covid19 -i f_texto1.csv -L Asia -S alfa -D inf -o f_dados.dat
  - Devem ser lidos os dados do ficheiro de texto f_texto1.csv. Os dados devem considerar as linhas relativas a infectados em países da Ásia. Os dados devem ser ordenados por ordem alfabética. Por fim, os resultados devem ser escritos no ficheiro binário f_dados.dat
- prog$ ./covid19 -i f_texto1.csv -L all -S dea 2021-05 -D inf -o f_texto2.csv

- Devem ser lidos os dados do ficheiro de texto f_texto1.csv relativos a todos os países. Os dados devem considerar as linhas relativas a infectados. Os dados devem ser ordenados por ordem decrescente de mortes na semana 2021-05. Por fim, os resultados devem ser escritos no ficheiro de texto f_texto2.csv (**Neste caso ao selecionar apenas os dados relativos a infetados e a ordenar por número decrescente de mortes, não deve ser feita nenhuma ordenação visto que não existem dados sobre mortes seleccionados**)

No caso de o resultado ser um conjunto de dados vazio, o ficheiro de saída deve refletir esse resultado e ser um ficheiro vazio. No caso de uma incompatibilidade de opções, deve ser gerado um ficheiro vazio. No caso de especificação de opções inexistentes ou de opções com valores não válidos, deve ser enviado uma nota explicativa para o stderr.

5.4. Formato dos Ficheiros de Saída

O formato dos ficheiros de saída deve ser:

- [ficheiro de texto com extensão csv] semelhante ao formato do ficheiro de entrada mas contendo apenas as linhas que resultem do processamento realizado com base nas opções indicadas em linha de comando. Mais, tal como o ficheiro original, o ficheiro de resultados deve conter a linha inicial com identificação das colunas.
- [ficheiro binário com extensão dat] semelhante em termos de informação ao ficheiro de texto CSV mas em formato binário, isto é, deve ser guardado de acordo com a sua codificação e não caracter a caracter. O ficheiro de dados não deve conter a informação de descrição das colunas existente na 1ª linha do ficheiro CSV.

6. Teste do Programa

Nesta secção descreve-se o modo como deve ser executado e testado o programa com recurso aos ficheiros de exemplo fornecidos. O programa será avaliado da mesma forma, mas com outros ficheiros de idêntica formatação mas de dimensão diferente (em número de linhas).

O programa desenvolvido deve ser robusto no sentido em que deverá garantir a validade dos dados em cada uma das colunas do ficheiro de entrada. Por exemplo, a existência de texto onde deveria existir um valor numérico, a existência de algarismos nos nomes do países, a falta de colunas em alguma linha do ficheiro são erros que devem ser detectados e que devem abortar a execução do programa com uma mensagem de erro para o terminal no formato “-1 Erro de Leitura …”.

As operações de alocação dinâmica realizadas no programa serão ainda validadas com recurso ao programa Valgrind, que se sugere seja utilizado para verificar a consistência das mesmas.

7. Processo de Submissão

Os trabalhos submetidos deverão obrigatoriamente ser compilados com as seguintes opções -Wall -std=c11 -O3 e correr na máquina virtual fornecida.

- **Entrega final (28 de maio de 2021)**
  Submissão no Fénix do: (1) código comentado com as funcionalidades indicadas para a entrega final; (2) Descrição da estrutura de dados utilizada para representar os dados lidos de ficheiro (máx: 4 páginas); (3) ficha de auto-avaliação a preencher no GoogleForms.

É importante reforçar que certos modos de operação do programa serão avaliados de forma automática, pelo que é imperativo que o programa respeite a geração das saídas para ficheiro conforme descritas. A falha na execução dos mesmos poderá levar a uma penalização na nota final.

Por fim, para o projeto final 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 corretamente indentado e com comentários que facilitem a sua legibilidade. A submissão final, por via eletrónica, deverá conter o código do programa (ficheiros .h e .c), uma Makefile para gerar o executável e o relatório final em formato PDF. Todos os ficheiros (*.c, *.h, Makefile) e relatório devem ser comprimidos num único ficheiro, de acordo com as instruções que serão publicadas no Fénix quando se abrir o processo de submissões.

8. Avaliação

A avaliação do trabalho terá a seguinte distribuição de cotações:

- Leitura de dados e criação de lista e estrutura de dados adequada (representação dos dados de forma eficiente e económica): 20%
- Funcionalidades de ordenação das listas: 10%
- Funcionalidades de seleção de elementos das listas: 10%
- Funcionalidades de restrição das listas: 10%
- Correta utilização da memória dinâmica: 20%
- Comentários: 10%
- Qualidade do código: 20%
- Não utilização de listas: **penalização de 50%**
- Não utilização de alocação dinâmica: **penalização de 100%**
- Relatório ou guia de implementação: Obrigatório, cotação embebida nos tópicos em avaliação.

Avaliação oral a todos os grupos (**realizada após a entrega final**)

- A avaliação obtida resultante da ponderação do projecto intermédio (1/3) e do projeto final (2/3), antes da discussão, será o ponto de partida e limite máximo da nota após a discussão oral.

9. 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. O projeto é para ser planeado e executado por grupos de dois alunos e é nessa base que será avaliado. Quaisquer associações de grupos ou outras, que eventualmente venham a ocorrer, serão obviamente interpretadas como violação do Código de Honestidade Académica e terão como consequência a anulação do projeto aos elementos envolvidos.

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 direta do código, mas também da estrutura do mesmo.