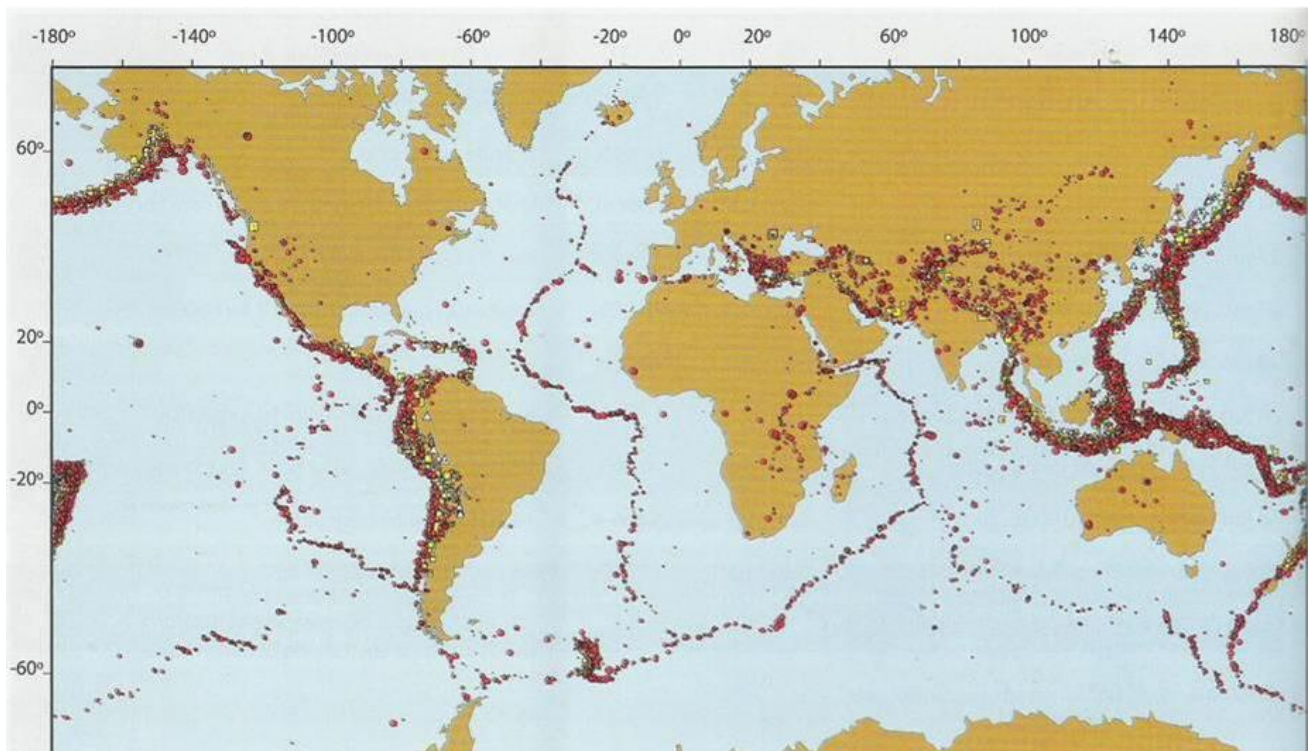


Análise de Terramotos (Processamento Estatístico e Visualização de Dados)



1. Objetivo do Projeto

Pretende-se desenvolver um programa em C para extrair/apresentar informação útil de um ficheiro com dados sobre sismos naturais e artificiais que ocorreram em vários países.

Os termos “terramoto”, “tremor de terra” e “sismo” são normalmente usados com igual significado. Sismos naturais – São aqueles que têm a sua origem na dinâmica da própria natureza; Sismos artificiais – São os sismos que tem origem nas ações do Homem sobre a natureza.

O programa consiste num interpretador de comandos que o utilizador usa para obter diversos tipos de informação, principalmente informação estatística.

1.1 Representação dos dados em Memória

Para este programa será necessário representar informação de sismos, de estatísticas de países e de localizações de países/territórios.

É obrigatória a manutenção em memória da informação importada utilizando as estruturas de dados e de ADTs definidos seguidamente:

Para os sismos:

- **Estrutura de dados:**

Cada registo de sismo corresponde a um evento sismológico que ocorreu num dado instante (data e hora) e num dado local (latitude e longitude). Essa informação é guardada **obrigatoriamente** na estrutura `Eartquake` apresentada na figura 1; a data e hora devem ser guardados respetivamente nas estruturas `Date` e `Time`.

- **ADT:** a coleção destes itens numa instância do ADT List, sendo **ListElem** o tipo `Eartquake`

Para as estatísticas dos países:

- **Estrutura de dados:**

As estatísticas de cada país são representadas, **obrigatoriamente**, pela estrutura de dados `CountryStatistics` apresentada na figura 1. Esta estrutura permite guardar alguns dos seus dados estatísticos. Só são considerados nesta estrutura registos de países considerados politicamente independentes.

- **ADT:** a coleção destes itens numa instância de ADT Map, sendo **ValueElem** do tipo `CountryStatistics` (figura 1) e o **KeyElem** de um tipo apropriado que permita guardar uma *string* (código do país - a chave do dicionário);

Para as localizações dos países/territórios:

- **Estrutura de dados:**

Para poder determinar de forma automática qual o país ou região em que ocorreu um sismo a informação da localização de países/territórios, guarda-se, **obrigatoriamente**, na estrutura `CountryLocation` apresentada na figura 1. Esta estrutura guarda informação sobre países ou territórios dependentes/associados a outros países.

- **ADT:** o conjunto destes itens num array (alocado dinamicamente) do tipo apropriado para guardar essa informação.

Não é permitido (nem necessário) alterar as interfaces lecionadas dos ADT, nomeadamente os ficheiros `list.h` e `map.h`. Estas instâncias serão designadas doravante por “coleções”. Por outras palavras, o conteúdo destes ficheiros deve permanecer inalterado, face aos originais fornecidos.

Não é também necessário ou permitido alterar as estruturas apresentadas na figura 1.

Na implementação dos comandos descritos neste enunciado podem definir/utilizar outros tipos de dados auxiliares que se achem úteis para a resolução dos problemas (por vezes é até recomendado que o façam na descrição do comando a desenvolver).

```

typedef struct date {
    int day, month, year;
} Date;

typedef struct time {
    int hour, min, sec;
} Time;

typedef struct earthquake {
    int id;
    // Código de 2 letras do país ou território mais próximo (countryLocation)
    char countryCode[3]; // Calculado a partir das localizações durante importação
    Date date;
    Time time;
    float latitude;
    float longitude;
    char type[20]; // Earthquake | Explosion | Nuclear Explosion | Rock Burst
    float depth;
    float magnitude;
    char magnitudeType[3]; // MB | MD | MH | ML | MS | MW
} Earthquake;

typedef struct countryStatistics {
    char code[3]; // Código de 2 letras que representa o país
    char name[40];
    char region[30]; // UN Regions + ANTARCTICA + SOUTH ATLANTIC + INDIAN OCEAN
    long population;
    long area;
    long gdp_capita; // Produto interno bruto per capita. Contém valores N.A.
    float literacy; // Percentagem de literacia. Contém valores N.A.
    float birthrate; // Número de nascimentos por 1000 pessoas por ano. Contém valores N.A.
    float deathrate; // Número de mortes por 1000 pessoas por ano. Contém valores N.A.
} CountryStatistics;

typedef struct countryLocation {
    char code[3]; // Igual a countryStatistics se país mas diferente se for dependente
    float latitude;
    float longitude;
    char territoryName[45]; // Nome do território
    char countryName[45]; // Nome do país associado se território dependente, se não N.A.
    char territoryRegion[30]; // Nome da região se for território dependente, se não N.A.
    // Existem duas novas regiões INDIAN OCEAN e SOUTH ATLANTIC
} CountryLocation;

```

Figura 1 – Definição de Tipos de dados.

1.2 Dados de entrada

São disponibilizados 3 ficheiros de dados:

- `earthquakes.csv` - Ficheiro de dados sobre os sismos e outras explosões;
- `world_country_locations.csv` - Ficheiro de dados que contém a localização – em termos de latitude e longitude média - de todos os países/territórios;

- `world_country_statistics.csv` - Ficheiro de dados sobre os dados estatísticos dos países;

Todos os ficheiros se encontram em formato CSV; a primeira linha dos ficheiros é uma linha com os cabeçalhos e não contém dados.

Ficheiro dos sismos e outras explosões (cada linha corresponde a um sismo/explosão)

```
<id>;<date>;<time>;<latitude>;<longitude>;<type>;<depth>;<magnitude>;<magnitude type>  
...
```

Ficheiro com os dados dos países/território (cada linha corresponde a informação de localização informação sobre um país/território)

```
<country_code>;<latitude>;<longitude>;<country_territory>;<associated_country>;<region>  
...
```

Ficheiro com os dados estatísticos dos países

```
<country_code>;<country>;<region>;<population>;<area>;<gdp_capita>;<literacy>;<birthrate>;<deathrate>  
...
```

O valor `<date>` (data) encontra-se no formato "mm/dd/aaaa" e o valor `<time>` no formato "hh:min:ss"

Pode-se assumir que não existem ficheiros "mal-formatados".

A. No caso do ficheiro de dados sobre os países, podem existir campos sem dados que estarão identificados como "N.A.".

- No caso de um campo do tipo *string* ser igual a "N.A." deve manter o valor lido;
- No caso de um campo do tipo inteiro ou real for igual a "N.A." deve ser inicializado com um valor igual a -1 (um negativo).

1.3 Comandos

Há exatamente **15 comandos que o programa deve implementar**, que serão apresentados de seguida; 3 comandos para carregamento de dados, 10 comandos para mostrar resultados de cálculos sobre os dados, 1 comando para sair da aplicação e 1 comando para limpeza dos dados em memória.

Os comandos têm o seguinte grau de dificuldade previsto: **BAIXA**, **MÉDIA** e **ALTA**

Notas:

- Cada comando é representado por uma palavra que pode ser escrita pelo utilizador em maiúsculas ou em minúsculas, não importa.
- Sempre que um comando necessitar de algum input, e.g., nome do país, este deve ser solicitado ao utilizador.
- Sempre que um comando necessitar de informação que não está carregada, o comando deve indicar que informação está em falta, i.e., **"No earthquake data available..."** e/ou **"No country data available..."**.

A forma exata como os resultados devem ser mostrados no ecrã será descrita em seguida. **Não se deve assumir que os ficheiros de entrada estão ordenados.**

A. Os comandos base são os seguintes:

✓ **LOADCL**

- Abre o ficheiro "world_country_locations.csv" e carrega-o em memória (ver 1.2), mostrando o número de países importado, e.g., "<N> country location records imported".

Se o ficheiro não puder ser aberto, escreve "File not found" e a coleção respetiva fica vazia.

✓ **LOADEA**

- Abre o ficheiro "earthquakes.csv" e carrega-o em memória (ver Secção 1.2), mostrando o número de dados de terremotos importados, e.g., "<N> earthquake records imported".

Este comando necessita da estrutura com as localizações (carregada a partir do ficheiro "world_country_locations.csv") para poder calcular o campo countryCode (código do país/território mais próximo do terremoto) da estrutura **Earthquake** (Secção 1.1).

Deve utilizar a distância euclidiana:

https://pt.wikipedia.org/wiki/Dist%C3%A2ncia_euclidiana

Se o ficheiro não puder ser aberto, escreve "File not found" e a coleção fica vazia. Se os registos dos países ainda não tiverem sido lidos, apresenta a mensagem "Please load country location data first".

✓ **LOADST**

- Abre o ficheiro "world_country_statistics.csv" e carrega-o em memória (ver 1.2), mostrando o número de países importado, e.g., "<N> country records imported".

Se o ficheiro não puder ser aberto, escreve "File not found" e a coleção respetiva fica vazia.

✓ **CLEAR**

- Limpa a informação atualmente em memória. Deverá indicar o número de registos que foram descartados, e.g., "<N> records deleted from <Earthquakes | Countries | Country locations>".

✓ **QUIT**

- Sai do programa, libertando toda a memória alocada para as coleções.

- B.** Os comandos de indicadores simples para terremotos (os cálculos requeridos só precisam de processar informação da coleção de terremotos) são os seguintes:

Comandos **SHOW** para a coleção de terremotos.

Os valores devem ser mostrados paginados de 50 em 50 linhas com a possibilidade de passar para próxima página ou terminar a listagem.

Aconselha-se que façam funções para filtrar e uma função para imprimir a lista filtrada.

✓ **SHOWALL**

- Mostra os dados de todos os terremotos de todos os países. Antes de apresentar a listagem deve apresentar o número de registos encontrados.

✓ **SHOW_Y**

- Mostra os dados de terremotos disponíveis **de um ano solicitado ao utilizador**. Antes de apresentar a listagem deve apresentar o número de registos encontrados.
- Caso o ano que inseriu não tenha dados disponíveis na coleção, escreve "**Earthquake data not available for year <year>**".

✓ **SHOW_T**

- Mostra os dados de terremotos disponíveis **de um país/território solicitado ao utilizador (código de país/território)**. Antes de apresentar a listagem deve apresentar o número de registos encontrados.
- Caso o país/território que inseriu não tenha dados disponíveis na coleção, escreve "**Earthquake data not available for <Country/Territory Code>**".

NOTA: O comando deve ser **case insensitive**, i.e., aceitar letras maiúsculas ou minúsculas.

✓ **SHOW_YT**

- Mostra os dados de terremotos disponíveis **de um país/território solicitado ao utilizador (código de país/território) num determinado ano solicitado ao utilizador**. Antes de apresentar a listagem deve apresentar o número de registos encontrados.
- Deve fazer as mesmas verificações que fez anteriormente para o ano e país/território.

✓ **LIST_T**

- Mostra a lista de códigos de países/territórios que têm dados de terremotos, sem repetições de códigos. Antes de apresentar a listagem deve apresentar o número de países/territórios encontrados.

✓ **COUNT**

- Deve pedir ao utilizador:
 - Um intervalo de magnitude (mínimo e máximo);
 - Um código de país/território.
- Mostra as contagens para um determinado **intervalo de magnitude pedido ao utilizador**:
 - Do número de terremotos que o país/território escolhido teve nesse intervalo de magnitude

- Da percentagem de terremotos do país/território escolhido de cada tipo de terremoto nesse intervalo de magnitude.
- Caso o intervalo que inseriu não tenha dados disponíveis na coleção, escreve "Earthquake data not available for magnitude between [<min>,<max>]"
- Caso o país/território que inseriu não tenha dados disponíveis na coleção (nenhum terremoto, independentemente do intervalo), escreve "Earthquake data not available for <Country/Territory Code>"

✓ HISTOGRAM

- Deve pedir ao utilizador quantos países/territórios quer ver no histograma (até um máximo de 3). Caso o valor seja menor que 1 ou maior que 3 deve dar uma mensagem de erro e voltar a pedir.
- Deve depois pedir os códigos dos países/territórios
- Deve criar uma função que calcula e devolve, para cada país/território, o número de terremotos para os intervalos de magnitude:
 - ≤ 6
 -] 6 , 6.5]
 -] 6.5 , 7]
 -] 7 , 7.5]
 -] 7.5 , 8]
 - > 8
- Caso algum dos países não tenha terremotos deve escrever "Earthquake data not available for <Country/Territory Code>"
- Em baixo é apresentado um exemplo com 2 países (PT e JP):

Code | Magnitude | Number of Earthquakes (logarithmic scale)

PT	≤ 6	##### 23
JP	≤ 6	##### 2093
PT] 6.0, 6.5]	##### 6
JP] 6.0, 6.5]	##### 470
PT] 6.5, 7.0]	### 1
JP] 6.5, 7.0]	##### 150
PT] 7.0, 7.5]	0
JP] 7.0, 7.5]	##### 48
PT] 7.5, 8.0]	##### 2
JP] 7.5, 8.0]	##### 21
PT	> 8	0
JP	> 8	##### 8

Para cada país/território escolhido aparece:

- Código do país/território
- Designação do intervalo (igual para os N países dentro do mesmo intervalo)
- Barra do histograma que indica para aquele país/território quantos terremotos houve nesse intervalo de magnitudes
- Valor numérico que indica para aquele país/território quantos terremotos houve nesse intervalo de magnitudes

Como o número de terremotos decresce exponencialmente com o aumento da magnitude a barra deve estar em escala logarítmica (base 2), colocando-se 3 símbolos (###) por cada aumento na escala logarítmica. Ou seja:

- 3 símbolos para 1 terremoto ($\log_2(1) = 0$)
- 6 símbolos para 2 ou 3 terremotos ($\log_2(2) = 1$)
- 9 símbolos para 4 a 7 terremotos ($\log_2(4) = 2$)
- 12 símbolos para 8 a 15 terremotos ($\log_2(8) = 3$)
- 15 símbolos para 16 a 31 terremotos ($\log_2(16) = 4$)
- ...

NOTA: Pode usar a função **log21** da biblioteca **math.h** para calcular o valor de logaritmo de base 2 de um número (não existe $\log_2(0)$, mas o resultado deverá ser 0 também).

C. Os comandos de indicadores simples para estatísticas de países (os cálculos requeridos só precisam de processar informação da coleção de estatísticas de países) são os seguintes:

✓ COUNTRY_S

- Mostra os dados dos países ordenados por % (percentagem) de literacia e por GDP (per capita)

Deve solicitar ao utilizador:

- Qual o critério principal (sendo o segundo utilizado para desempate).
- Se a ordem será crescente ou decrescente.

Caso ainda exista empate após os dois critérios deve usar o código de país para desempatar.

✓ REGION_AVG

Deve pedir ao utilizador o nome da região para qual pretende os cálculos.

Mostra, **para a região escolhida**, os seguintes dados agregados países:

- nome da região
- área total
- população total
- densidade populacional
- média da literacia, ponderada com a população
- *gdp per capita* da região
- *gdp* (total) da região

NOTA: Deve considerar apenas os itens CountryStatistics contidos na instância de ADT Map. No caso de não existir alguma das estatísticas de um país, a totalidade da informação desse país deve ser ignorada. Após apresentar os cálculos, deve indicar quantos países foram ignorados nesses cálculos.

Bonificação: Mostre no final, os dados anteriores para todos os países (input/region: "ALL").

D. Os comandos de indicadores complexos (os cálculos requeridos precisam dos dados da coleção de terremotos e das coleções dos países) são os seguintes:

✓ **TOPN (Lista os países com mais terremotos num intervalo de anos ou todos por defeito)**

- O valor N deve ser solicitado ao utilizador.

Mostra de forma decrescente os dados de terremotos de N países/territórios, sendo o critério **decrescente** pelo número de terremotos registados nesse país/território.

Deve mostrar os seguintes dados (aconselha-se a criação de uma estrutura auxiliar com estes dados):

- *code*
- *territoryName*
- *numberOfEarthquakes*
- *averageDepth*
- *averageMagnitude*

Sugere-se que divida a implementação em 3 funções:

- uma que calcule a lista com cada um dos países e dados calculados
- outra que obtenha os N elementos com mais terremotos ordenados por ordem decrescente
- outra que mostre os dados dos N países.

1.4 Git Classroom e repositório template

Todos os projetos deverão ser obrigatoriamente versionados através do Git Classroom. O link do *assignment* encontra-se no Moodle junto com este enunciado.

2 Relatório e Documentação

2.1 Documentação

Todo o código deve ser documentado utilizando a **documentação Doxygen**.

A mesma deve ser gerada para formato HTML e entregue a respetiva pasta "html" junto com o projeto.

2.2 Relatório

No relatório deverão constar as seguintes secções (para além de capa com identificação dos alunos e índice). Pode utilizar o template de relatório fornecido no Moodle.

a) **Divisão de trabalho** – Descrição da participação de cada elemento:

- Para cada comando a percentagem de participação de cada elemento.
- Percentagem de participação de cada elemento para o projeto como um todo.

- b) **ADTs Utilizados** - Descrição breve dos ADTs utilizados, qual a implementação escolhida e porquê (comparação de eficiências para o problema de aplicação).
- c) **Complexidades Algorítmicas** - Para cada comando (exceto CLEAR, e QUIT) fornecer:
- A complexidade algorítmica da respetiva implementação, tendo em conta as complexidades algorítmicas das funções dos ADTs utilizadas (dependem da implementação escolhida).
- d) **Algoritmos** - Escolha de 3 funcionalidades do tipo B, C e D, onde apresentam o algoritmo implementado em pseudo-código;
- e) **Limitações** - Quais os comandos que apresentam problemas ou não foram implementados;
- f) **Conclusões** - Análise crítica do trabalho desenvolvido.

3 Tabela de Cotações e Penalizações

A avaliação do trabalho será feita de acordo com os seguintes princípios:

- **Estruturação:** o programa deve estar estruturado de uma forma modular e procedimental;
- **Correção:** o programa deve executar as funcionalidades, tal como pedido.
- **Legibilidade e documentação:** o código deve ser escrito, formatado e comentado de acordo com o standard de programação definido para a disciplina.
- **Desempenho:** Os algoritmos implementados devem ter em conta a complexidade do mesmo, valorizando-se a implementação de algoritmos com menor complexidade. A gestão da memória deverá ser feita corretamente, garantindo que a mesma é libertada quando não está a ser utilizada. Utilização da ferramenta *Valgrind*, para validar a correta gestão de memória.

A nota final obtida, cuja tabela de cotações se apresenta a seguir, será ponderada de acordo com os princípios acima descritos.

Descrição	Cotação (valores)
Leitura de comandos, tratamento de situação de ficheiro inexistente/vazio, limpeza de memória (CLEAR) e saída do programa (QUIT)	1
Importação de dados (comandos LOADCL, LOADEA e LOADST)	2
Comandos SHOWALL, SHOW_Y, SHOW_T e SHOW_YT	2,5
Comando LIST_T	1,5
Comando COUNT	2
Comando HISTOGRAM	2,5
Comando COUNTRY_S	2
Comando REGION_AVG	1,5
Comando TOPN	2
Relatório e Documentação	3
TOTAL	20

A seguinte tabela contém penalizações a aplicar:

Descrição	Penalização (val.)
Uso de variáveis globais	até 2
Não separação de funcionalidades em funções/módulos	até 3
Não libertação de memória	até 3
Não comentar o programa	até 2
Não utilização dos ADTs obrigatórios	Anulado
Não geração de documentação doxygen (pasta html)	Até 1

4 Instruções e Regras Finais

O não cumprimento das regras a seguir descritas implica uma penalização na nota do trabalho prático. Se ocorrer alguma situação não prevista nas regras a seguir expostas, essa ocorrência deverá ser comunicada ao respetivo docente de laboratório de ATAD.

Regras:

- O Projeto deverá ser elaborado por **grupos entre 3 e 4 alunos do mesmo docente de laboratório (ou 2 + 2 de docentes diferentes)**. Exceções têm de ser validadas pelo RUC.
- A nota do Projeto será atribuída individualmente a cada um dos elementos do grupo após a discussão. As discussões poderão ser orais e/ou com perguntas escritas. As orais poderão ser feitas com todos os elementos do grupo presentes em simultâneo ou individualmente. E poderão ser feitas remotamente via plataforma zoom.
 - Os *commits* no repositório individual do grupo serão tidos em conta na avaliação individual.
- A apresentação de relatórios ou implementações plagiadas leva à imediata atribuição de nota zero a todos os trabalhos com semelhanças, quer tenham sido o original ou a cópia.**
- No rosto do relatório e nos ficheiros de implementação deverá constar o número, nome e turma dos autores e o nome do docente a que se destina.
- O trabalho deverá ser submetido no Moodle, no link do respetivo docente de laboratórios criado para o efeito, **até às 11.00 do dia 17 de junho: 12h do dia 19 de junho.**

Para tal terão que criar uma pasta com o nome: **nomeAluno1_númeroAluno1-nomeAluno2_númeroAluno2-...**, onde colocarão o ficheiro do relatório em formato **pdf** e uma pasta com o projeto VS Code (cópia do repositório, versão para submissão). Os alunos terão de submeter essa **pasta compactada em formato ZIP**. Apenas será permitido submeter um ficheiro.
- Não serão aceites trabalhos entregues que não cumpram na íntegra o ponto anterior.
- As datas das discussões serão publicadas após a entrega dos trabalhos.

(fim de enunciado)