


| | | |
|---|---|------------------|
|  | Centro Tecnológico Departamento de Informática | |
| Disciplina: Programação II | | Código: INF09330 |
| Trabalho Prático | | |

Especificação do Trabalho

Sars-Cov-2 ES Statistics

1 Introdução

Este trabalho tem como objetivo avaliar, de forma prática, os conceitos ensinados no curso de Programação II. O trabalho pode ser feito de forma incremental, isto é, à medida que o conteúdo for ensinado. A meta deste trabalho é manipular um conjunto de dados disponibilizados pelo Governo do Estado do Espírito Santo sobre os casos de Covid-19 do Espírito Santo seguindo rigorosamente as especificações descritas neste documento e utilizando os conceitos ensinados em sala para manter boas práticas de programação.

2 Descrição do Trabalho

Neste trabalho, você deverá implementar o Sars-Cov-2 ES Statistics. O programa deverá rodar no terminal, assim como todos os outros exercícios feitos em sala.

Você terá que carregar um arquivo de extensão .csv, que será disponibilizado para download, que contém casos de Covid-19 no Espírito Santo (para reduzir o tamanho do arquivo, são considerados dados até o final de agosto).

Para o arquivo carregado, você deverá realizar uma série de análises que serão descritas ao longo deste documento.


2.1 Arquivo de Entrada

O arquivo de entrada (covid19ES.csv) é composto de 12 colunas e 202.363 linhas. Cada coluna é separada por uma vírgula e a primeira linha do arquivo contém o nome de cada coluna. As primeiras 5 linhas do arquivo são descritas a seguir:


| | DataCadastro | DataObito | Classificacao | Municipio | IdadeNaDataNotificacao | ComorbidadePulmao | ComorbidadeCardio | ComorbidadeRenal | ComorbidadeDiabetes | ComorbidadeTabagismo | ComorbidadeObesidade | FicouInternado |
|---|--------------|------------|---------------|--------------|----------------------------|-------------------|-------------------|------------------|---------------------|----------------------|----------------------|----------------|
| 0 | 2020-07-20 | 0000-00-00 | Suspeito | SAO MATEUS | 27 anos, 5 meses, 20 dias | Não | Não | Não | Não | Não | Não | Não Informado |
| 1 | 2020-07-20 | 0000-00-00 | Suspeito | LINHARES | 19 anos, 10 meses, 16 dias | Não | Não | Não | Não | Não | Não | Não |
| 2 | 2020-07-20 | 0000-00-00 | Confirmados | NOVA VENEZIA | 47 anos, 7 meses, 12 dias | Não | Não | Não | Não | Não | Não | Não Informado |
| 3 | 2020-07-20 | 0000-00-00 | Confirmados | IBIRACU | 28 anos, 4 meses, 7 dias | Não | Não | Não | Não | Não | Não | Não |
| 4 | 2020-07-20 | 0000-00-00 | Descartados | LINHARES | 40 anos, 7 meses, 22 dias | Não | Não | Não | Não | Não | Não | Não |

Info: CSV é uma sigla em inglês que significa *Comma Separated Values* (Valores Separados por Vírgula). Esse padrão é internacionalmente conhecido, o que permite que você visualize os dados do arquivo utilizando, por exemplo, Google Sheets, Excel ou LibreOffice Calc.

Cada coluna do arquivo possui um cabeçalho, que indica os dados que aquela coluna possui. A seguir, são descritos cada um deles. Observe que os nomes estão exatamente iguais aos contidos no arquivo:

| | | |
|---|---|------------------|
|  | Centro Tecnológico Departamento de Informática | |
| Disciplina: Programação II | | Código: INF09330 |
| Trabalho Prático | | |

- **DataCadastro:** string no formato ANO-MÊS-DIA (Ex: 2020-07-20) que representa a data de cadastro no sistema.
- **DataObito:** string no formato ANO-MÊS-DIA (Ex: 2020-05-12) que representa a data de óbito do paciente. Caso o paciente não tenha vindo à óbito, a string que representa a data será composta por: 0000-00-00
- **Classificacao:** string que representa a classificação do paciente para Covid19. Pode ter 3 diferentes valores: “Suspeito”, caso o paciente tenha informado sintomas mas não fez o teste; “Confirmados”, caso o paciente tenha feito o teste e o resultado tenha dado positivo; ou “Descartados”, caso o paciente tenha feito o teste e o resultado tenha sido negativo.
- **Município:** string que representa o município de residência do paciente. A string município está sempre em caixa alta. Ex: SAO MATEUS.
- **IdadeNaDataNotificacao:** string no formato “X anos, Y meses, Z dias” (Ex: "27 anos, 5 meses, 20 dias") que representa a idade do paciente na data em que foi cadastrado no sistema. **Obs:** Em alguns poucos casos, o dado não apresenta os meses ou dias, ou apresenta um valor incoerente. Porém, isso não afeta as respostas, uma vez que vamos utilizar apenas a idade em anos.
- **ComorbidadePulmao:** string que indica se o paciente possui ou não alguma comorbidade pulmonar. Possui dois possíveis valores “Sim”, caso o paciente tenha a comorbidade, “Não” caso não tenha e ‘-’ caso a informação é desconhecida.
- **ComorbidadeCardio:** string que indica se o paciente possui ou não alguma comorbidade cardíaca. Possui dois possíveis valores “Sim”, caso o paciente tenha a comorbidade, “Não” caso não tenha e ‘-’ caso a informação é desconhecida.
- **ComorbidadeRenal:** string que indica se o paciente possui ou não alguma comorbidade renal. Possui dois possíveis valores “Sim”, caso o paciente tenha a comorbidade, “Não” caso não tenha e ‘-’ caso a informação é desconhecida.
- **ComorbidadeDiabetes:** string que indica se o paciente possui ou não diabetes. Possui dois possíveis valores “Sim”, caso o paciente tenha a comorbidade, “Não” caso não tenha e ‘-’ caso a informação é desconhecida.
- **ComorbidadeTabagismo:** string que indica se o paciente é ou não é fumante. Possui dois possíveis valores “Sim”, caso o paciente tenha a comorbidade, “Não” caso não tenha e ‘-’ caso a informação é desconhecida.
- **ComorbidadeObesidade:** string que indica se o paciente é ou não é obeso. Possui dois possíveis valores “Sim”, caso o paciente tenha a comorbidade, “Não” caso não tenha e ‘-’ caso a informação é desconhecida.
- **FicouInternado:** string que indica se o paciente ficou ou não internado. Possui três possíveis valores “Sim”, caso o paciente tenha a ficado internado; “Não” em caso

| | | |
|---|---|------------------|
|  | Centro Tecnológico Departamento de Informática | |
| Disciplina: Programação II | | Código: INF09330 |
| Trabalho Prático | | |

negativo, “Não Informado” e “Ignorado” caso não haja este registro no sistema.

2.2 Desenvolvimento do código

Seu trabalho deverá cumprir os sete requisitos listados a seguir. Os dois primeiros são relacionados a manipulação do arquivo e das datas. Os 5 restantes, são análises dos dados da incidência de Covid-19 no nosso estado. Todas as porcentagens, médias e desvio padrão solicitados nos itens deverão utilizar **precisão de 3 casas decimais**. Além disso, é **fortemente recomendado** que você implemente cada item como uma função (que pode chamar outras funções, caso queira).

1. **Leitura do arquivo CSV:** você deve construir uma função responsável por realizar a leitura do arquivo CSV. Você deve utilizar uma ou mais *struct* para armazenar os dados carregados.

Importante: essa função deverá ser executada uma única vez para cada arquivo de teste! caso você carregue várias vezes por teste, seu código poderá demorar demais para rodar.

2. **Filtro de datas:** você deve desenvolver uma função responsável por filtrar as datas para cumprir os itens que envolve leituras de período de datas (itens 4, 5 e 7). A filtragem tem como objetivo ler uma string “2020-07-20” e interpretar que ela é o dia 20 de julho de 2020.

3. Liste em, **ordem alfabética**, as cidades do estado com mais de N casos de Covid-19.


Exemplo:

Entrada:

N = 500

Saída:

- ANCHIETA: 502 casos
- ARACRUZ: 2142 casos
- CACHOEIRO DE ITAPEMIRIM: 3355 casos
- CARIACICA: 8571 casos
- CASTELO: 688 casos
- COLATINA: 3778 casos

| | | |
|---|---|------------------|
|  | Centro Tecnológico Departamento de Informática | |
| Disciplina: Programação II | | Código: INF09330 |
| Trabalho Prático | | |

- GUARAPARI: 1924 casos
- ITAPEMIRIM: 676 casos
- LINHARES: 3931 casos
- MARATAIZES: 869 casos
- MARECHAL FLORIANO: 562 casos
- NOVA VENECIA: 557 casos
- PRESIDENTE KENNEDY: 550 casos
- SAO GABRIEL DA PALHA: 777 casos
- SAO MATEUS: 1264 casos
- SERRA: 11251 casos
- VIANA: 1376 casos
- VILA VELHA: 12551 casos
- VITORIA: 11290 casos

4. Determine quantos casos de Covid-19 foram diagnosticados entre as datas d1 e d2, na qual d1 será sempre menor que d2.

Dica: essa função chamará a função de filtro de datas

Exemplo:

Entrada:

d1 = 2020-07-10
d2 = 2020-08-10

Saída:

- Total de pessoas: 11345

5. Liste as cidades top N de casos de covid entre a data d1 e d2, na qual d1 será sempre menor que d2. A saída deverá ser ordenada em ordem decrescente por número de casos.


Dica: essa função chamará a função de filtro de datas

Exemplo:

Entrada:

N = 10
d1 = 2020-07-10
d2 = 2020-08-10

Saída:

| | | |
|---|---|------------------|
|  | Centro Tecnológico Departamento de Informática | |
| Disciplina: Programação II | | Código: INF09330 |
| Trabalho Prático | | |

VILA VELHA: 1372 casos
SERRA: 1271 casos
VITORIA: 1094 casos
CARIACICA: 957 casos
LINHARES: 856 casos
COLATINA: 778 casos
CACHOEIRO DE ITAPEMIRIM: 634 casos
ARACRUZ: 469 casos
GUARAPARI: 322 casos
SAO MATEUS: 234 casos

6. Determine a % de pessoas confirmadas que ficaram internadas, % de pessoas que morreram e a % de pessoas internadas que morreram, de um dado município ou de todas as cidades.

OBS: Caso o município receba a string “TODAS”, deverão ser considerados todos os municípios que obedecem a descrição da função. A string poderá ser passada como parâmetro em qualquer formato, por exemplo, ViToRia, ViToria, vitOria, porém sua função deverá converter todas as letras da string para maiúsculo.

Exemplo:

Entrada:

municipio = TODAS

Saída:

- Resultados para TODAS:
- A % de pessoas com Covid-19 que ficaram internadas: 3.011%
- A % de pessoas com Covid-19 que morreram: 3.390%
- A % de pessoas que ficaram internadas e morreram: 46.741%

OBS 2: Note que é utilizado 3 casas decimais


7. Determinar a média e desvio padrão da idade das pessoas que morreram e a % de pessoas que morreram e não possuíam nenhuma comorbidade entre a data d1 e d2 onde d1 será sempre menor que d2.

Entrada:

d1 = 2020-07-10
d2 = 2020-08-10

Saída:

A média e desvio padrão da idade: 70.212 -- 16.070
A % de pessoas que morreram sem comorbidade: 27.799%

| | | |
|---|---|------------------|
|  | Centro Tecnológico Departamento de Informática | |
| Disciplina: Programação II | | Código: INF09330 |
| Trabalho Prático | | |

2.3 Execução do Código

Serão disponibilizados arquivos de teste públicos juntamente com o arquivo de entrada de dados. O primeiro passo é ler o arquivo de entrada de dados contendo os casos de Covid-19 do Espírito Santo (covid19ES.csv) e armazená-lo em uma estrutura de dados.

O arquivo de teste será denominado <teste_N.txt> e deverá ser lido na entrada padrão (da mesma maneira que são lidos os arquivos de entrada do BOCA). Esse arquivo contém as seguintes informações:

```
<PASTA_DE_SAIDA>
<ENTRADA_ITEM_3>
<ENTRADAS_ITEM_4>
<ENTRADAS_ITEM_5>
<ENTRADA_ITEM_6>
<ENTRADAS_ITEM_7>
```


Onde:

- **<PASTA_DE_SAIDA>**: caminho para a pasta onde o seu código deverá salvar os resultados para cada item descrito previamente.
- **<ENTRADAS_ENTRADA_ITEM_N>**: representa os parâmetros de entrada para os itens de 3 a 7, respectivamente. Caso haja mais de uma entrada, as mesmas estarão separadas por um espaço.

Um exemplo do arquivo de teste:

```
./teste1/
500
2020-07-10 2020-08-10
10 2020-07-10 2020-08-10
TODAS
2020-07-10 2020-08-10
```

Em outras palavras: as saídas do seu código (vamos falar delas na próxima seção) deverão ser salvas na pasta *teste1*. O item 3 tem como entrada N=500. O item 4, tem como entrada as datas d1=2020-07-10 e d2=2020-08-10. Assim por diante.

| | | |
|---|---|------------------|
|  | Centro Tecnológico Departamento de Informática | |
| Disciplina: Programação II | | Código: INF09330 |
| Trabalho Prático | | |

2.4 Arquivo de Saída

Para cada um dos itens de 3 a 7, considerando os parâmetros lidos no arquivo entrada, você deve criar um arquivo de saída chamado <item_N.txt>. Os padrões de saída estão descritos dentro dos itens na seção 2.2. Por exemplo, para o item 6, seu código deve resultar um arquivo *item6.txt* contendo, por exemplo, os seguintes valores:

- Resultados para TODAS:
- A % de pessoas com Covid-19 que ficaram internadas: 3.011%
- A % de pessoas com Covid-19 que morreram: 3.390%
- A % de pessoas que ficaram internadas e morreram: 46.741%

Esse arquivo deverá ser salvo na pasta indicada no arquivo nos parâmetros de entrada. No fim das contas, a pasta deverá conter 5 arquivos .txt como resultado do seu código.

Importante: sigam estritamente os padrões de saída indicados nos itens da seção 2.2. A correção será realizada de maneira automática, assim como no BOCA.

2.5 Implementação


A implementação deverá seguir os conceitos de modularização e abstração apresentados em sala. O trabalho terá uma componente subjetiva que será avaliada pelo professor para verificar o grau de uso dos conceitos ensinados. Portanto, além de funcionar, o código deverá estar bem escrito para que o aluno obtenha nota máxima.

Dica: é extremamente recomendado utilizar algum programa para fazer as comparações do resultado final do programa, isto é, os arquivos de saída gerados, poderão ser comparados com o arquivo de saída esperada (fornecido pelo professor) utilizando o comando *diff*, como visto em sala. O *meld* é uma alternativa gráfica para o *diff*, se você preferir. Esse programa faz uma comparação linha a linha do conteúdo de 2 arquivos. Diferenças na formatação poderão impossibilitar a comparação e consequentemente a correção do trabalho. O programa será considerado correto se gerar a saída esperada IDÊNTICA à fornecida com os casos de teste.

3 Regras Gerais

O trabalho deverá ser feito em **dupla** e pelo próprios alunos, isto é, os alunos deverão necessariamente conhecer e dominar **todos** os trechos de código criados.

Cada dupla deverá trabalhar independente das outras, não sendo permitido a cópia ou compartilhamento de código. O professor irá fazer verificação automática de plágio. Trabalhos

| | | |
|---|---|------------------|
|  | Centro Tecnológico Departamento de Informática | |
| Disciplina: Programação II | | Código: INF09330 |
| Trabalho Prático | | |

identificados como iguais, em termos de programação, serão penalizados com a **nota zero**. Isso também inclui a dupla que forneceu o trabalho, sendo, portanto, de sua obrigação a proteção de seu trabalho contra cópias ilícitas. **Proteja seu trabalho e não esqueça cópias do seu código nas máquinas de uso comum (ex. laboratório).**

3.1 Entrega do Trabalho

O trabalho deverá ser entregue pelo AVA até as 23:59 do dia **04/12/2020**. O arquivo deve seguir o padrão: PROG2-2020_MATRICULA1_MATRICULA2.c, substituindo a matrícula pelos respectivos números de matrícula da dupla.

O código fonte deverá estar bem comentado, isto é, todas as structs e funções devem ter descrições claras do que ela faz, o que representa cada parâmetro e o que ela retorna (se houver). Pode-se adicionar também comentários que ajude a entender a lógica da questão.


IMPORTANTE: O código fonte será compilado e testado APENAS com o arquivo de dados que está disponível para download no AVA.

3.2 Pontuação e Processo de Correção

Trabalhos entregues após o prazo **não serão corrigidos** (nota zero). O trabalho será pontuado de acordo com sua implementação e a tabela abaixo. Os pontos descritos na tabela não são independentes entre si, isto é, alguns itens são pré-requisitos para obtenção da pontuação dos outros. Por exemplo, gerar o arquivo de estatísticas depende de realizar a simulação corretamente. Código com falta de legibilidade e modularização pode perder ponto conforme informado na tabela. Erros gerais de funcionamento, lógica ou outros serão descontados como um todo.

Percebam que no melhor dos casos os pontos da tabela abaixo somam 11 ao invés de 10. Isso foi feito propositalmente para ajudar os alunos esforçados com um ponto extra. Esse ponto, caso obtido, irá complementar uma das notas, do trabalho ou das provas parciais do semestre. Prioridade será dada para a nota com maior peso, porém não ultrapassando a nota máxima.

| Item | Quesitos | Ponto |
|--------------------|--|-------------|
| Inicialização | Ler o arquivo de entrada e armazená-lo corretamente em uma estrutura de dados (Item 1) | 0,5 |
| Lógica das Funções | Filtro de datas funcionando corretamente (Item 2) | 0,5 |
| Lógica das Funções | Itens 3 a 7 funcionando corretamente | 1,8 cada |

| | | |
|---|---|------------------|
|  | Centro Tecnológico Departamento de Informática | |
| Disciplina: Programação II | | Código: INF09330 |
| Trabalho Prático | | |

| | | |
|------------------------------|--|----|
| Legibilidade e Modularização | Falta de: <ul style="list-style-type: none"> • Uso de comentários; • Identação do código; • Uso de funções; • Uso de tipos de dados definidos pelo usuário | -2 |
|------------------------------|--|----|

Processo de Correção: O trabalho será corrigido em dois turnos: correção do professor e entrevista. A correção do professor (CP) gerará uma nota entre 0 e 10 e as entrevistas (E) serão realizadas posteriormente à data de entrega do trabalho e será atribuída uma nota entre 0 e 1. Alunos que fizeram o trabalho em dupla farão a entrevista juntos porém terão seus desempenhos na entrevista avaliados de forma individual. Pequenas e pontuais correções de código serão permitidas no momento da entrevista, o que poderá acarretar em uma CP maior.

A nota final (NF) do trabalho será dada pela seguinte fórmula: $NF = E * CP$

Importante: apesar de ser em dupla, as notas serão individuais e podem ser diferentes. É importante que domine toda lógica do código entregue e não apenas a “parte que desenvolveu”. As perguntas durante a entrevista podem ser direcionadas a qualquer parte do código e a qualquer componente da dupla.

4 Considerações Finais

Não utilizar acentos no trabalho.

5 Erratas

Esse documento descreve de maneira geral as regras de implementação do trabalho. É de responsabilidade do aluno garantir que o programa funcione de maneira correta e amigável com o usuário. Qualquer alteração nas regras do trabalho será comunicada em sala e no portal do aluno. É responsabilidade do aluno frequentar as aulas e se manter atualizado em relação às especificações do trabalho. Caso seja notada qualquer tipo de inconsistência nos arquivos de testes disponibilizados, comunique imediatamente ao professor para que ela seja corrigida e reenviada para os alunos.