COMPUTIONAL THINKING USING PYTHON

Prof. Edson de Oliveira

Objetivos:

Colocar em prática os conceitos e técnicas envolvendo o pensamento computacional em linguagem Python e que foram aprendidas nas aulas da disciplina Computional Thinking using Python.

Instruções:

Com base no estudo de caso da “Smart Mobility & Technology Solutions” a equipe deve implementar um MVP (*Minimum Viable Product*) – Produto viável mínimo, como uma versão enxuta e funcional de uma solução. Para isto, escolha uma ou mais funcionalidades de grande relevância para o projeto, realize o levantamento de requisitos e especificação (podendo ser os mesmos aplicados na disciplina DDD) e implemente uma solução em Python.

O projeto deve conter os diversos conceitos vistos em sala de aula como funções, contendo passagem de parâmetros e retorno, listas, tuplas e dicionários, aplicação de algum algoritmo de ordenação etc.

Critérios de Avaliação:

- Atender aos requisitos deste documento;
- Desenvolver os módulos com Subalgoritmos;
- Respeitar a forma de entrega descrita no capítulo abaixo;
- Aplicar usabilidade e utilizar mensagens informativas para facilitar as operações quando necessário;
- Cada item do menu valerá 2,5 pontos.

A aplicação correta dos conceitos aprendidos durante todo o ano letivo, organização do código e o seu funcionamento. Pensem sempre na aplicabilidade de técnicas que facilitem a manutenção futura do código.

Entregáveis:

1. Texto

“3.6 Até 2020, reduzir pela metade as mortes e os fetrimentos globais por acidentes em estradas. No Brasil, em 2015 ocorreram 19 mortes de trânsito para cada 100 mil habitantes. Em 2019, a taxa reduziu para 15 mortes para cada 100 mil habitantes, mas ainda longe da meta da ODS (Objetivos de Desenvolvimento Sustentável).”

2. Estrutura de dados

Armazenar dados relacionados a este problema é importante e necessário para que as estatísticas possam ser cadastradas, processadas para depois serem analisadas.

Pensando nisso, vamos criar um dicionário para armazenarmos estas informações:

REGISTRO (dicionário)

CAMPO	TIPO	FORMATO
mes_ano_referencia	str - tamanho 7	mm-aaaa
total_habitantes	int	
total_obitos	int	

Considere que este dicionário é a estrutura do elemento de uma lista. Este elemento armazenará os dados de cada registro do mês cadastrado:

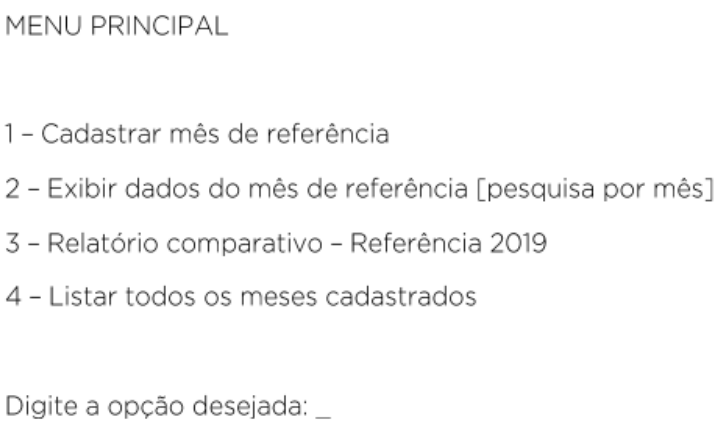
TABELA (lista do registro)

índice	elemento
0	mes_ano_referencia: 01-2021 | total_habitantes: 123456 | total_obitos: 17
1	mes_ano_referencia: 02-2021 | total_habitantes: 112321 | total_obitos: 15
2	mes_ano_referencia: 03-2021 | total_habitantes: 121345 | total_obitos: 10
...	...

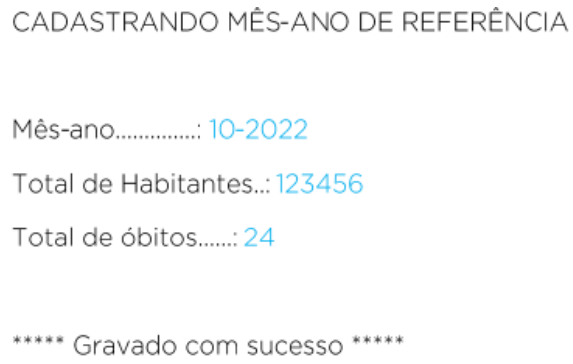
A partir destas estruturas de dados, a aplicação em Python gravará e manipulará dados para gerar relatórios.

3. Definições das entregas

Considere um menu principal com os seguintes itens:



Quando a opção escolhida for 1, a solução deverá gravar a linha do ano-mes de referência. Similar a tela abaixo:



Tela 2: Opção 1 – Cadastro do mes-ano de referência

Perguntar se o usuário quer cadastrar outro mês ou voltar para o menu anterior será uma escolha do grupo.

Quando digitada a opção 2 (exibir dados do mês de referência), o usuário deverá digitar o mes-ano desejado (no formato sugerido). Caso seja o mes-ano seja encontrado, exibir os dados cadastrados. Caso contrário, exibir “mes-ano não cadastrado”, como sugerem a tela de sucesso na pesquisa abaixo:

CONSULTANDO MÊS-ANO DE REFERÊNCIA

Digite o mês-ano desejado.....: 10-2022

Mês-ano.....: 10-2022
Total de Habitantes..: 123456
Total de óbitos.....: 24

***** Registro encontrado *****

Ao escolher a opção 3 (Exibir dados de um ano de referência), percorrer a lista com as linhas que representam este ano e exibir o relatório abaixo:

RELATÓRIO COMPARATIVO DE TAXA DE MORTALIDADE ANUAL

Digite ano a ser comparado.....: 2021

Total de Habitantes.....: 1348128
Total de óbitos.....: 217
Taxa por 100k habitantes - 2021...: 16.09
Taxa por 100k habitantes - 2019...: 15.00

Comparativo % entre 2021-2019.....: +7.3%

Sendo “Total de Habitantes” a somatória de todos os habitantes cadastrados nos meses do ano escolhido. “Total de óbitos” a somatória de todos os óbitos cadastrados nos meses do ano escolhido. “Taxa por 100k habitantes – 2021” é o cálculo envolvendo as duas somatórias acima do ano referido. “Taxa por 100k habitantes – 2019” é o dado que o texto forneceu. “Comparativo % entre 2021-2019” é o comparativo da proporção das duas últimas taxas.

Quando a opção 4 for selecionada, todas as linhas gravadas (independentemente do mês e ano) serão apresentadas na tela.

PARA DESENVOLVER ESTA APLICAÇÃO, PRIVILEGIE A UTILIZAÇÃO DE SUBALGORITMOS (FUNÇÕES E PROCEDIMENTOS).

4. Entrega

- Deverá ser efetuada via portal até a data limite;
- Comentem nas primeiras linhas do arquivo principal .py os RMs e Nomes dos integrantes;
- Deverá ser compactado em um arquivo .zip este documento e o(s) arquivo(s) .py desta aplicação;
- As entregas devem ser feitas via Portal até a data estabelecida;
- Uma vez cadastrado o grupo no Portal da FIAP, o representante deverá postas as entregas;
- Uma vez inserida a nota pelo professor, o portal aplicará a mesma para todos os integrantes do grupo.