

Introdução

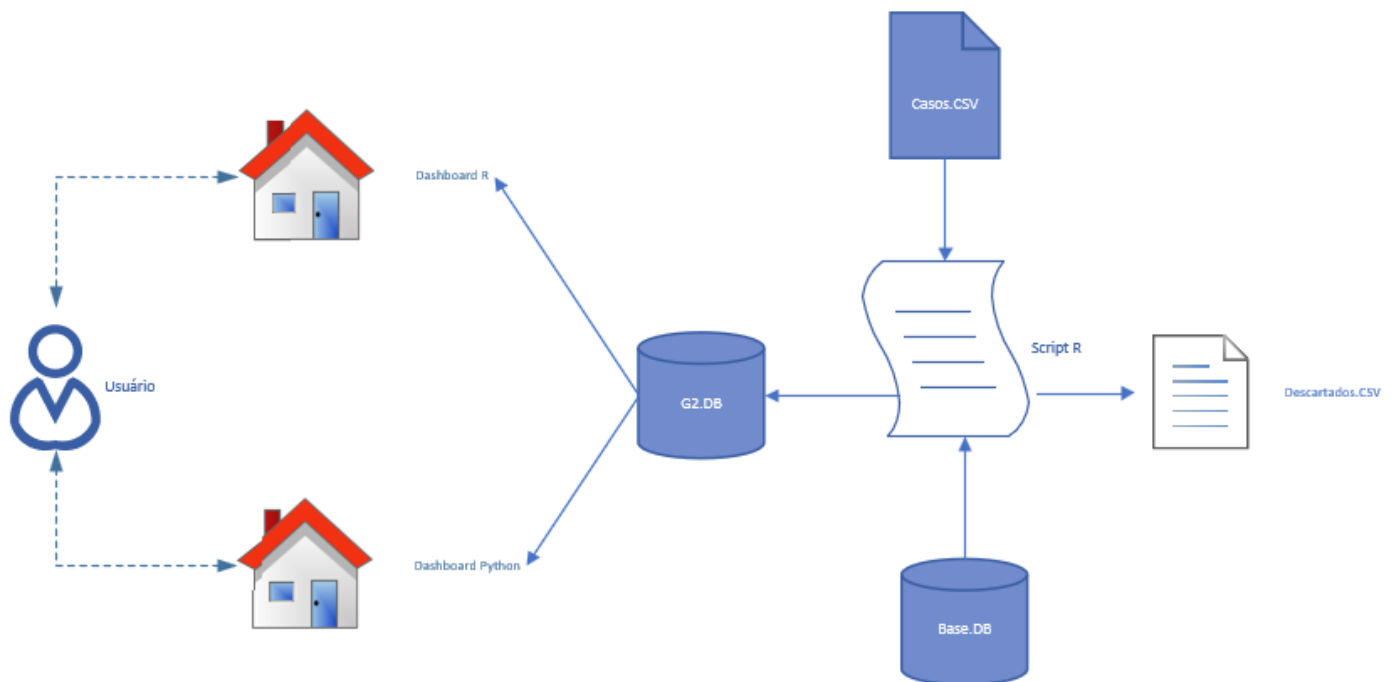
Você recebeu o enunciado do trabalho da G2 cuja solução deverá ser **enviada pela plataforma EAD**. Ela contém **duas questões** e todas as atividades relacionadas à solução do trabalho proposto devem ser realizadas respeitando-se o código de ética do CTC disponível na plataforma EAD.

Como parte do trabalho, será realizada uma avaliação oral presencial dos membros do grupo, que irá valer 30% da nota e incluirá perguntas elaboradas a partir da resolução apresentada pelo grupo. A avaliação oral irá durar em torno de **15 minutos**, sendo que qualquer um dos membros poderá ser sorteado no dia para responder aos questionamentos. Quem não estiver presente no dia e horário determinado para a avaliação oral terá falta computada penalizando a nota do grupo caso seja sorteado. O aluno sorteado que não estiver presente deverá realizar a apresentação em caráter de segunda chamada e haverá 30% de penalidade na nota, à exceção de casos previstos pelo regimento interno da PUC-RIO, mediante ingresso de solicitação via CCAR, após a devida análise e a eventual aprovação pelo Departamento.

A parte escrita do presente trabalho irá valer 70% da nota que deverá ser entregue até as **23hrs** do dia **18/06**. Trabalhos entregues com até 2 horas de atraso sofrerão uma penalidade de 10% no valor da nota. Trabalhos entregues com até 3 horas de atraso sofrerão 15% de penalidade. Trabalhos entregues com até 4 horas de atraso sofrerão 25% de penalidade. Trabalhos entregues com até 12 horas de atraso sofrerão 35% de penalidade. Não serão aceitos trabalhos com mais de 12 horas de atraso.

Questão 1)

A figura abaixo apresenta os componentes envolvidos nesta questão.



Componentes:

- Usuário: usuário das aplicações a serem desenvolvidas;
- Dashboard R: dashboard R a ser desenvolvido;
- Dashboard Python: dashboard Python a ser desenvolvido;
- G2.DB: base de dados que armazena os dados utilizados pelos dashboards;
- Casos.CSV: arquivo no formato csv contendo dados a serem carregados na base de dados **G2.DB**;
- Base.DB: base de dados contendo dados a serem carregados na base de dados **G2.DB**.
- Script R: script na linguagem R que deverá carregar na base de dados **G2.DB** os dados armazenados no arquivo **Casos.CSV** e na base de dados **Base.DB**.

O painel apresentado no **Anexo A** deverá ser reproduzido no formato de dashboard dinâmico em **R** com o nome **"INF1514_G2_GRUPO"** e no formato de dashboard dinâmico em Python com o nome **"INF1514_G2_GRUPO"**, substituindo o texto **"GRUPO"** pelo número do grupo. Os dashboards criados deverão conter todas as implementações realizadas para o cálculo de cada valor numérico bem como de cada tabela ou gráfico gerado. Também deverá fazer parte do dashboard, em forma de comentário: as considerações feitas sobre os dados; o passo a passo adotado para a resolução da questão; a descrição das variáveis; a descrição das funções, parâmetros e valores de retorno. Deverão ser entregues, além dos dashboards, um arquivo com o nome **"INF1514_G2_R_Painel.PDF"**, correspondente ao dashboard R, e outro arquivo com o nome **"INF1514_G2_Python_Painel.PDF"**, correspondente ao dashboard Python, representando o painel do **Anexo A** preenchido com as informações solicitadas.

Os dashboards deverão ler os dados armazenados na base de dados **G2.DB**, a qual, deverá ser carregada com os dados do arquivo **Casos.CSV** e da base de dados **Base.DB**, os quais permitem gerar informações a respeito de casos novos e óbitos novos referentes à infecção causada por um vírus no Brasil ao longo do tempo. Para realizar a carga da base de

dados **G2.DB** deverá ser implementados um script em R. A definição das tabelas a serem criadas na base de dados **G2.DB** faz parte do trabalho. (8,0 pontos)

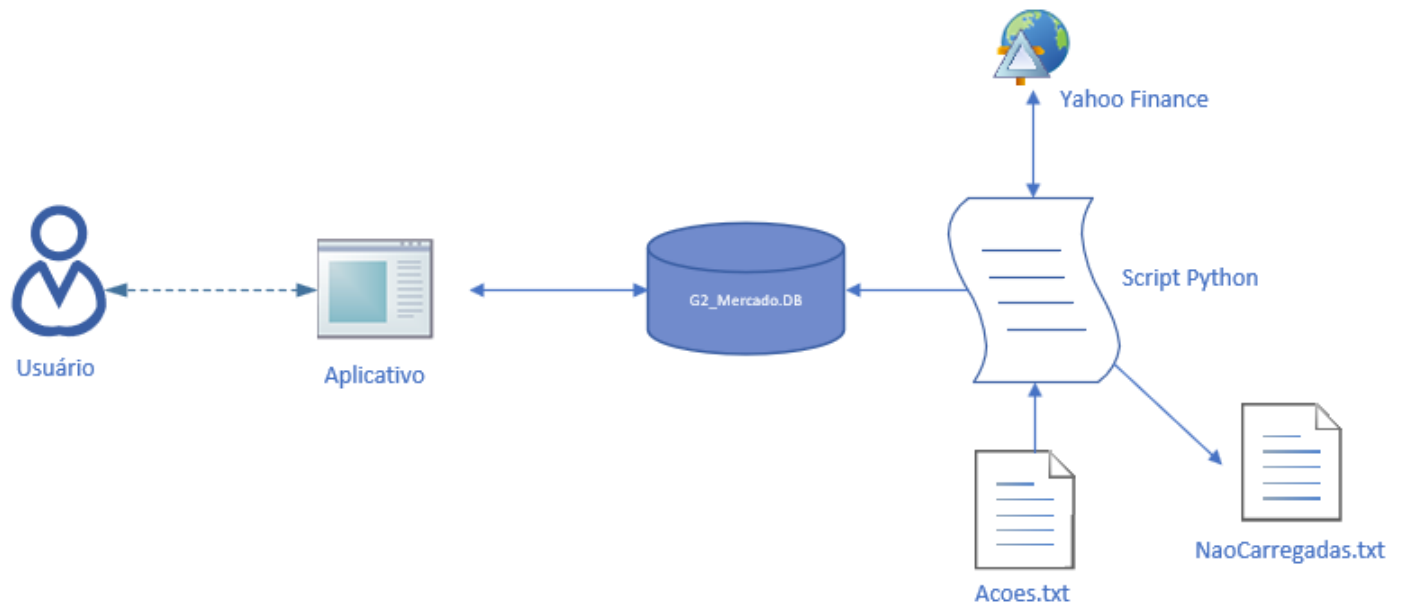
Observações:

- As tabelas a serem criadas na base de dados **G2.DB** deverão tomar como base o uso de **regras de normalização de dados** com o objetivo de reduzir a redundância de dados além de garantir a consistência e a integridade dos mesmos.
- Caso o script de carga encontre algum registro do arquivo **Casos.CSV** inconsistente e que deva ser desconsiderado, este deverá ser descartado e seu conteúdo salvo no arquivo **Descartados.CSV**.
- O script de carga deverá sempre remover todos os dados armazenados na base dados **G2.DB** antes de realizar a sua carga.
- Os dashboards deverão obrigatoriamente fazer uso de funções responsáveis por gerar as informações, tabelas e gráficos utilizados para compor o painel.
- O arquivo **Base.DB** contém uma tabela chamada **Regiao** com dados das regiões do Brasil sendo eles: **CodigoRegiao** que é o identificador da região e **NomeRegiao** que é o nome da região.
- O arquivo **Base.DB** contém uma tabela chamada **Estado** com dados dos Estados do Brasil, sendo eles: **CodigoRegiao** que corresponde ao conjunto de dados **CodigoRegiao** da tabela **Regiao** e indica a qual região o estado pertence; **CodigoIBGE** que é o código do IBGE para o Estado; **UF** que corresponde à UF do Estado; **NomeEstado** que corresponde ao nome do Estado.
- O arquivo **Casos.CSV** contém dados diários de Estados e cidades do Brasil do dia 25/02/2020 a 27/09/2020, sendo eles: **nomelocalidade** que indica o nome do Estado ou cidade, sendo que no caso de Estado, este campo não está preenchido; **codigoibge** que é o código do IBGE para o Estado ou cidade (no caso de um Estado, o **codigoibge** corresponde ao **codigoibge** da tabela Estados que está no banco **Base.DB**); **data** (formato dd/mm/yyyy) que indica a data da contagem; **populacao** que contém a população estimada para o Estado ou cidade no ano de 2019; **tipo** que identifica se o registro se refere a uma informação de Estado (possui valor **state**) ou de cidade (possui valor **city**); **uf** que corresponde à UF do Estado ou à UF à qual a cidade pertence; **casosnovos** que indica o total de casos novos da doença no dia; **obitosnovos** que indica o total de óbitos novos da doença no dia.
- Devido a inconsistências na divulgação de dados por parte de órgãos estaduais, o arquivo **Casos.CSV** apresenta algumas inconsistências: se for feita a soma de todos os casos (ou óbitos) de todas as cidades de um determinado Estado ao longo de todo o período de coleta não necessariamente este valor será igual à soma de todos os casos (ou óbitos) para o Estado ao longo de todo o período de coleta; podem ocorrer valores negativos de casos e de óbitos em determinadas datas, sendo estes, valores corretos de ajustes para lançamentos realizados a mais em datas anteriores; podem ocorrer lançamentos de cidades sem identificação de **códigoibge**, sendo estes dados inconsistentes. Faz parte da solução da questão decidir o que fazer com as inconsistências.

- Para complementar o dashboard, também poderão ser usados:
 - *shapefiles* do Brasil e seus estados no formato .shp, disponibilizados pelo IBGE no link <https://www.ibge.gov.br/geociencias/downloads-geociencias.html>, na estrutura “organizacao_do_territorio / malhas_territoriais / malhas_municipais / municipio_2022 /”.
 - *shapefiles* do Brasil e seus estados no formato .geojson, disponibilizados pelo IBGE no link <https://servicodados.ibge.gov.br/api/docs/malhas?versao=3>.
- No arquivo **Casos.CSV** a coluna **codigoibge** corresponde:
 - ao conjunto de dados **CD_UF** do *shapefile* .shp ou **codarea** do *shapefile* .geojson do Brasil obtido no site do IBGE, caso o registro tenha para a coluna **tipo** o valor **state**.
 - ao conjunto de dados **CD_MUN** do *shapefile* .shp ou **codarea** do *shapefile* .geojson do Estado obtido no site do IBGE, caso o registro tenha para a coluna **tipo** o valor **city**.
- Deve ser utilizado como base para a implementação do dashboard em R o pacote **shinydashboard**.
- Deve ser utilizado como base para a implementação do dashboard em Python o pacote **dash**.

Questão 2)

A figura abaixo apresenta os componentes envolvidos nesta questão.



Componentes:

- Usuário: usuário do aplicativo a ser desenvolvido;
- Aplicativo: aplicativo Python a ser desenvolvido;
- G2_Mercado.DB: base de dados que armazena os dados utilizados pelo Aplicativo;
- Yahoo Finance: site que contém dados históricos de ações que deverão ser recuperados, via API (*Application Programming Interface*), e armazenados na base de dados **G2_Mercado.DB**;
- Acoes.txt: arquivo txt contendo as ações que deverão ter seus dados históricos recuperados do Yahoo Finance e armazenados na base de dados **G2_Mercado.DB**;
- Script: script Python que deverá ler as ações armazenadas no arquivo Acoes.txt (cada linha deverá conter a sigla de uma única ação) e para cada uma recuperar o nome da ação e os dados históricos (abertura, fechamento, máximo, mínimo, etc) dos últimos 360 dias, contados a partir da data corrente de execução do script, do Yahoo Finance e armazená-los na base de dados **G2_Mercado.DB**. Caso os dados da ação não existam no Yahoo Finance, a sua sigla deverá ser armazenada no arquivo NaoCarregadas.txt (uma linha por sigla de ação);
- NaoCarregadas.txt: a cada execução do script este arquivo deverá conter apenas as siglas das ações, uma por linha, cujos dados não foram encontrados no Yahoo Finance na execução corrente.

Deverá ser criado um aplicativo em Python, utilizando Tkinter, PyQt ou Flet, que disponibilize quatro análises gráficas, como candlestick, o critério do grupo, iterativas e úteis na avaliação de ações. As análises deverão ser apresentadas na tela após uma ação e um período (uma da data de início e fim) serem selecionados na interface do aplicativo. Observe que o banco de dados G2_Mercado.DB já contém uma tabela chamada Acao que deverá fazer parte do modelo de dados final da solução sendo também carregada pelo script Python. (2,0 pontos)

Total de casos confirmados

- Brasil
<XXX>
- Estado do Rio de Janeiro
<XXX>
- Cidade do Rio de Janeiro
<XXX>

Total de óbitos

- Brasil
<XXX>
- Estado do Rio de Janeiro
<XXX>
- Cidade do Rio de Janeiro
<XXX>

Letalidade (total de óbitos / total de casos confirmados)

- Brasil
<XXX>
- Estado do Rio de Janeiro
<XXX>
- Cidade do Rio de Janeiro
<XXX>

Mortalidade (total de óbitos / total de habitantes)

- Brasil
<XXX>
- Estado do Rio de Janeiro
<XXX>
- Cidade do Rio de Janeiro
<XXX>

Casos no Brasil

<Mapa do Brasil por estados com padrão de cor degradê baseado no total de casos confirmados até o momento.>

Casos no Estado do Rio de Janeiro

<Mapa do RJ por cidades com padrão de cor degradê baseado no total de casos confirmados até o momento>

Casos por região

<Gráfico de setores por região do Brasil com o % de casos confirmados (acumulados) de cada região. O gráfico deverá apresentar legenda.>

Casos por estados da região Sudeste

<Gráfico de setores por estados e % do total de casos confirmados (acumulados) de cada um com legenda.>

10 cidades do Brasil com maior taxa de casos confirmados (casos confirmados / habitantes)

<Tabela com 5 colunas: UF, Nome (nome da cidade), Taxa (total de casos confirmados / habitantes), Letalidade (total de óbitos / total de casos) e Mortalidade (total de óbitos / habitantes). Os valores devem tomar como base o total de casos acumulados. As cidades devem estar ordenadas por UF e a seguir pelo Nome. >

10 cidades do Estado do RJ com maior taxa de casos confirmados (casos confirmados / habitantes)

<Tabela com 4 colunas: Nome (nome da cidade), Taxa (total de casos confirmados / habitantes), Letalidade (total de óbitos / total de casos confirmados) e Mortalidade (total de óbitos / habitantes). Os valores devem tomar como base o total de casos acumulados. As cidades devem estar ordenadas pelo Nome.>

Casos novos por dia

<Gráfico de dispersão, tendo os pontos ligados, com o eixo X representando os dias e o eixo Y o total de casos novos (não confundir com o valor acumulado) por dia de todo o Brasil e do estado do RJ. No eixo X deve aparecer um contador de 1 ao total de dias e não a informação de data. As linhas para Brasil e RJ devem possuir cores diferentes e o gráfico uma legenda. Para os dias em que não existir uma coleta para o Brasil ou para o Rio de Janeiro deve ser considerado o valor 0.>

Média móvel de óbitos do Brasil e do Estado do Rio de Janeiro

<Gráfico de dispersão formado pela média móvel com uma janela de 6 dias (dia atual + 5 anteriores) do total de óbitos do Brasil e do estado do RJ. No eixo X deve aparecer um contador de dias e não a informação de data. Para os dias em que não existir uma coleta para o Brasil ou para o Rio de Janeiro deve ser considerado o valor 0. Faça uma análise de tendência utilizando a variação percentual do valor da média móvel do último dia contra o valor da média móvel do décimo quarto dia anterior e verifique se: está “Estável” se a variação do último dia para o primeiro for de até 15% positivos ou negativos; está “Em crescimento” se a variação for acima de 15%; está “Em queda” se for mais do que 15% negativos. Aplique a análise ao Brasil e ao estado do RJ.>

Casos acumulados e óbitos do Estado do Rio de Janeiro

<Gráfico de dispersão, tendo os pontos ligados, com o eixo X representando os dias e o eixo Y o total de casos acumulados e o total de óbitos acumulados até o último dia de todo o estado do Rio de Janeiro. No eixo X deve aparecer um contador de 1 ao total de dias e não a informação de data. A linha para total de casos acumulados deve ter a cor “azul” e a linha para o total de óbitos acumulados deve ter a cor “vermelha”.>

Casos por Estado

<Tabela com 5 colunas: UF do estado, Confirmado (total de casos), Óbitos (total de óbitos), Letalidade (total de óbitos / total de casos) e Mortalidade (total de óbitos / total de habitantes). Os valores devem tomar como base o total de casos acumulados. Os estados devem estar ordenados pela UF. >