

O objetivo desta lista é criar um mini-projeto, o qual contém todas as camadas (Camada de dados, camada de lógica, e camada do Usuário). O projeto consiste em fazer a análise dos dados da COVID-19.

**Questão 1 (*Criando um projeto*)***(2 pontos)*

Crie um projeto no R-Studio com a seguinte estrutura:

1. Um diretório chamado: "Database"
2. Um diretório chamado: "Graficos"
3. Baixe a base de dados do site <https://ourworldindata.org/coronavirus-source-data>
4. Salve a base no diretório "Database" com o nome "owid-covid-data.xlsx"

(Uma versão antiga da base pode ser obtida no e-class)

Caso você não esteja utilizando o R-Studio apenas organize seu diretório de maneira a seguir a estrutura sugerida.

**Questão 2 (*Camada de dados(Data Loader)*)***(2 pontos)*

Crie um script chamado "DataLoader.r", o script deve realizar as seguintes tarefas:

1. Abrir o banco de dados de nome "owid-covid-data.xlsx" que está no diretório "Database"
2. A coluna data deve ser convertida para o tipo data. (garantindo que a coluna é do tipo "Date")
3. O script deve filtrar a base e mostrar apenas os seguintes países: "Brazil", "United States", "Mexico", "Germany", "France", "United Kingdom"
4. O script deve selecionar apenas as variáveis: **location**, **date**, **total\_cases**, **new\_cases**

**Solução:**

```
# Script DataLoader.r

# Carrega a base de dados
covid_data <- read_excel("Database/owid-covid-data.xlsx")

# Converte a coluna date para uma variável do tipo data.
covid_data$date = as.Date(covid_data$date)

# filtre os países selecionados
PaísesSelecionados <- c("Brazil", "United States", "Mexico",
                        "Germany", "France", "United Kingdom")

# Filtra a base para conter apenas os países selecionados.
covid_data <- covid_data %>% dplyr::filter(location %in% PaísesSelecionados)

# Seleciona apenas as variáveis utilizadas.
covid_data <- covid_data %>% dplyr::select(location, date, total_cases,
                                           new_cases)
```

**Questão 3 (*Função*)***(2 pontos)*

Crie uma função que recebe um vetor com datas e determina o dia da semana bem como faz uma contagem das semanas.

date	weekday	week
2020-01-01	4	1
2020-01-02	5	1
2020-01-03	6	1
2020-01-04	7	1
2020-01-05	1	2
2020-01-06	2	2
2020-01-07	3	2
2020-01-08	4	2
2020-01-09	5	2
2020-01-10	6	2
2020-01-11	7	2
2020-01-12	1	3

1. A função deve retornar um data.frame com as seguintes colunas: date, weekday, week. Exemplo:

**Solução:**

```
get_weekNumber <- function(date) {  
  # Bibliotecas requeridas  
  require(lubridate)  
  
  # inicializa uma tabela com a informacao das datas e duas colunas  
  tbl <- data.frame(date=unique(date),  
                    weekday = as.integer(NA),  
                    week = as.integer(NA))  
  
  # Adiciona a informacao de dia da semana as datas  
  tbl$weekday <- lubridate::wday(tbl$date)  
  
  # Organiza os dados  
  tbl <- tbl %>% arrange(date)  
  
  for(row in 1:nrow(tbl)) {  
  
    if(row == 1){  
      # Se é a primeira linha da tabela inicializa o contador  
      ncont = 1  
    } else if (tbl$weekday[row] == 1){  
      # Se nao for a primeira linha e for domingo, incrementa o contador.  
      ncont = ncont+1  
  
      # Verifica se a data de ontem é sequencial com a data de hoje,  
      # basicamente verifica que não tem datas ausentes.  
      if(tbl$date[row] != tbl$date[row-1]+1){  
        cat(sprintf("Data nao prevista: %s", tbl$date[row]))  
        break  
      }  
    }  
  
    # indica a qual semana a data pertence.  
    tbl$week[row] <- ncont  
  }  
  
  # retorna um data.frame com as datas, o dia da semana e a contagem progressiva  
  # das semanas  
  return(tbl)  
}
```

**Questão 4 (Camada lógica)**

(2 pontos)

Crie um script chamado “main.r”, o script deve fazer as seguintes tarefas:

1. Carregar a função criada anteriormente.
2. Carregar o script DataLoader.r
3. Utilizar a função criada para determinar qual a semana associada a cada registro na base “covid\_data”. (dica utilizar *join*)
4. Agrupar os dados por semana para cada pais selecionado. (dica: utilize *group\_by* e *summarise*)

5. excluir a informação da ultima semana pois essa pode estar incompleta.

**Solução:**

```
# Script logic

# Setup -----

# Clear all
rm(list = ls())

library(dplyr)
library(ggplot2)

# Funcoes -----

source("./get_weekNumber.R")

# Data Loader -----

source("./Dataloader.r")

dates_table <- get_weekNumber(covid_data$date)

# Main -----
# Faça um inner join com a dates_table e determine a semana
covid_data <- dplyr::left_join(covid_data, dates_table, by=c("date"="date"))

# Agrupe os dados por location e semana
covid_data.grouped <- dplyr::group_by(covid_data, location, week)

# faça um resumo do total de casos, e novos casos (utilize a soma)
covid_data.grouped <- dplyr::summarise(covid_data.grouped,
                                       total = sum(total_cases, na.rm = TRUE),
                                       new = sum(new_cases, na.rm = TRUE))

# exclua a ultima semana
covid_data.grouped <- filter(covid_data.grouped, week != max(week))
```

**Questão 5 (Camada do usuário)**

(2 pontos)

Crie uma rotina que cria 2 gráficos e salva eles no diretório “Gráficos”

1. Gráfico 1: utilizando os dados agrupados, crie um gráfico de linha do total de mortes por covid-19, com uma serie para cada pais selecionado. (eixo x: Semana, eixo y: Total de mortes).
2. Gráfico 2: utilizando os dados agrupados, crie um gráfico de linha das novas de mortes por covid-19, com uma serie para cada pais selecionado. (eixo x: Semana, eixo y: Novas mortes)

3. Salve os dois gráficos no diretório “Graficos”. (Configurações recomendadas: scale=1, units = “in”, dpi = 300, width = 10.4, height = 5.85)

**Solução:**

```
# Camada do usuário -----

g1 <- ggplot(covid_data.grouped) +
  geom_line(aes(x=week,
                y=total,
                colour=location))

g2 <- ggplot(covid_data.grouped) +
  geom_line(aes(x = week,
                y = new, colour=location)) +
  geom_point(aes(x = week,
                 y = new, colour=location))

ggsave(filename = "grafico 1.png",
        path = "Graficos",
        plot = g1,
        scale=1, units = "in", dpi = 300,
        width = 10.4, height = 5.85)

ggsave(filename = "grafico 2.png",
        path = "Graficos",
        plot = g2,
        scale=1, units = "in", dpi = 300,
        width = 10.4, height = 5.85)
```