**Tabelas criadas:**

* **owid-covid-data** -> Fonte: https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owid-covid-data.csv
* **vaccinations-by-manufacturer** -> Fonte: <https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccinations/vaccinations-by-manufacturer.csv>
* **vaccinations-by-age** -> Fonte: <https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccinations/vaccinations-by-age-group.csv>
* **dim\_geo** -> Fonte: <https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owid-covid-data.csv>
* **dim\_calendar** -> Script

**Transformações no Power Query:**

**owid-covid-data –** Tabela de factos com dados gerais sobre o covid por Continente/País e data – Casos, mortes, vacinas, …

1. Carregamento do dataset  
   = Csv.Document(Web.Contents("https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owid-covid-data.csv"),[Delimiter=",", Columns=67, Encoding=65001, QuoteStyle=QuoteStyle.None])
2. Promoção dos nomes das colunas   
   = Table.PromoteHeaders(Source, [PromoteAllScalars=true])
3. Alteração dos tipos das colunas:

|  |  |
| --- | --- |
| = Table.TransformColumnTypes(#"Promoted Headers",{{"iso\_code", type text},  {"continent", type text},  {"location", type text},  {"date", type date},  {"total\_cases", Int64.Type},  {"new\_cases", Int64.Type},  {"new\_cases\_smoothed", type number},  {"total\_deaths", Int64.Type},  {"new\_deaths", Int64.Type},  {"new\_deaths\_smoothed", type number},  {"total\_cases\_per\_million", type number},  {"new\_cases\_per\_million", type number},  {"new\_cases\_smoothed\_per\_million", type number},  {"total\_deaths\_per\_million", type number},  {"new\_deaths\_per\_million", type number},  {"new\_deaths\_smoothed\_per\_million", type number},  {"reproduction\_rate", type number},  {"icu\_patients", Int64.Type},  {"icu\_patients\_per\_million", type number},  {"hosp\_patients", Int64.Type},  {"hosp\_patients\_per\_million", type number},  {"weekly\_icu\_admissions", Int64.Type},  {"weekly\_icu\_admissions\_per\_million", type number},  {"weekly\_hosp\_admissions", Int64.Type},  {"weekly\_hosp\_admissions\_per\_million", type number},  {"total\_tests", Int64.Type},  {"new\_tests", Int64.Type},  {"total\_tests\_per\_thousand", type number},  {"new\_tests\_per\_thousand", type number},  {"new\_tests\_smoothed", Int64.Type},  {"new\_tests\_smoothed\_per\_thousand", type number},  {"positive\_rate", type number},  {"tests\_per\_case", type number}, | {"tests\_units", type text},  {"total\_vaccinations", Int64.Type},  {"people\_vaccinated", Int64.Type},  {"people\_fully\_vaccinated", Int64.Type}, {"total\_boosters", Int64.Type},  {"new\_vaccinations", Int64.Type},  {"new\_vaccinations\_smoothed", Int64.Type},  {"total\_vaccinations\_per\_hundred", type number},  {"people\_vaccinated\_per\_hundred", type number},  {"people\_fully\_vaccinated\_per\_hundred", type number},  {"total\_boosters\_per\_hundred", type number},  {"new\_vaccinations\_smoothed\_per\_million", Int64.Type},  {"new\_people\_vaccinated\_smoothed", Int64.Type},  {"new\_people\_vaccinated\_smoothed\_per\_hundred", type number},  {"stringency\_index", type number},  {"population\_density", type number},  {"median\_age", type number},  {"aged\_65\_older", type number},  {"aged\_70\_older", type number},  {"gdp\_per\_capita", type number},  {"extreme\_poverty", type number},  {"cardiovasc\_death\_rate", type number},  {"diabetes\_prevalence", type number},  {"female\_smokers", type number},  {"male\_smokers", type number},  {"handwashing\_facilities", type number},  {"hospital\_beds\_per\_thousand", type number},  {"life\_expectancy", type number},  {"human\_development\_index", type number},  {"population", Int64.Type},  {"excess\_mortality\_cumulative\_absolute", type number},  {"excess\_mortality\_cumulative", type number},  {"excess\_mortality", type number},  {"excess\_mortality\_cumulative\_per\_million", type number}}) |

1. Ordenação dos resultados por ordem ascendente por location e date   
   Table.Sort(#"Changed Type",{{"location", Order.Ascending}, {"date", Order.Ascending}})
2. Remoção de todas as colunas para além das necessárias:  
   = Table.SelectColumns(#"Sorted Rows",{"iso\_code", "continent", "location", "date", "total\_cases", "new\_cases", "new\_cases\_smoothed", "total\_deaths", "new\_deaths", "new\_deaths\_smoothed", "reproduction\_rate", "people\_fully\_vaccinated", "people\_fully\_vaccinated\_per\_hundred", "total\_vaccinations", "people\_vaccinated", "population", "new\_vaccinations", "total\_tests", "new\_tests"})
3. Substituição de zeros por null para “new\_cases” e “new\_deaths”  
   = Table.ReplaceValue(#"Removed Other Columns",0,null,Replacer.ReplaceValue,{"new\_cases"})  
   = Table.ReplaceValue(#"Replaced Value",0,null,Replacer.ReplaceValue,{"new\_deaths"})

**vaccinations-by-manufacturer –** Tabela de factos com quantidade de vacinas por marca por país e data

1. Carregamento do dataset  
   = Csv.Document(Web.Contents("https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccinations/vaccinations-by-manufacturer.csv"),[Delimiter=",", Columns=4, Encoding=65001, QuoteStyle=QuoteStyle.None])
2. Promoção dos nomes das colunas  
   = Table.PromoteHeaders(Source, [PromoteAllScalars=true])
3. Alteração dos tipos das colunas:  
   = Table.TransformColumnTypes(#"Promoted Headers",{{"location", type text}, {"date", type date}, {"vaccine", type text}, {"total\_vaccinations", Int64.Type}})
4. Ordenação dos resultados por ordem ascendente por location e date   
   Table.Sort(#"Changed Type",{{"location", Order.Ascending}, {"date", Order.Ascending}})

**vaccinations-by-age –** Tabela de factos com percentagem de vacinados por faixa etária por país e data

1. Carregamento do dataset  
   = Csv.Document(Web.Contents("https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccinations/vaccinations-by-age-group.csv"),[Delimiter=",", Columns=6, Encoding=65001, QuoteStyle=QuoteStyle.None])
2. Promoção dos nomes das colunas  
   = Table.PromoteHeaders(Source, [PromoteAllScalars=true])
3. Alteração dos tipos das colunas:

= Table.TransformColumnTypes(#"Promoted Headers",{{"location", type text}, {"date", type date}, {"age\_group", type text},

{"people\_vaccinated\_per\_hundred", type number},

{"people\_fully\_vaccinated\_per\_hundred", type number},

{"people\_with\_booster\_per\_hundred", type number}})}

1. Ordenação dos resultados por ordem ascendente por location e date   
   Table.Sort(#"Changed Type",{{"location", Order.Ascending}, {"date", Order.Ascending}})

**dim\_geo –** Tabela de dimensão geográfica – ISO Code, Continente e País

1. Carregar apenas 3 primeiras colunas do dataset  
   = Csv.Document(Web.Contents("https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owid-covid-data.csv"),[Delimiter=",", Columns=3, Encoding=65001, QuoteStyle=QuoteStyle.None])
2. Promoção dos nomes das colunas  
   = Table.PromoteHeaders(Source, [PromoteAllScalars=true])
3. Remoção de todas as colunas para além das necessárias:  
   = Table.TransformColumnTypes(#"Removed Columns",{{"iso\_code", type text}, {"continent", type text}, {"location", type text}})
4. Filtrar linhas mantendo apenas registos com a coluna continent preenchida  
   = Table.SelectRows(#"Changed Type", each [continent] <> null and [continent] <> "")
5. Eiminar registos duplicados  
   = Table.Distinct(#"Filtered Rows")

**dim\_calendar –** Tabela de dimensão temporal   
Tabela manualmente criada com todas as datas desde 01/01/2020 até à data actual

= List.Dates(#date(2020,1,1), Duration.Days(Date.From(DateTime.LocalNow()) - #date(2020,1,1)) +1 , #duration(1,0,0,0) )  
  
Adicionadas as colunas - Data, Ano, Mês, Nome do Mês, Quarter, Dia e Nome do Dia

**Transformações em DAX:  
  
owid-covid-data**

Adicionadas as colunas xxx\_fill em que as linhas vazias da tabela original são populadas através da propagação do ultimo valor encontrado nas linhas anteriores.

* People\_fully\_Vaccinated\_fill
* People\_Vaccinated\_fill
* Total\_Vaccinations\_fill

ex:

Total\_Vaccinations\_fill =

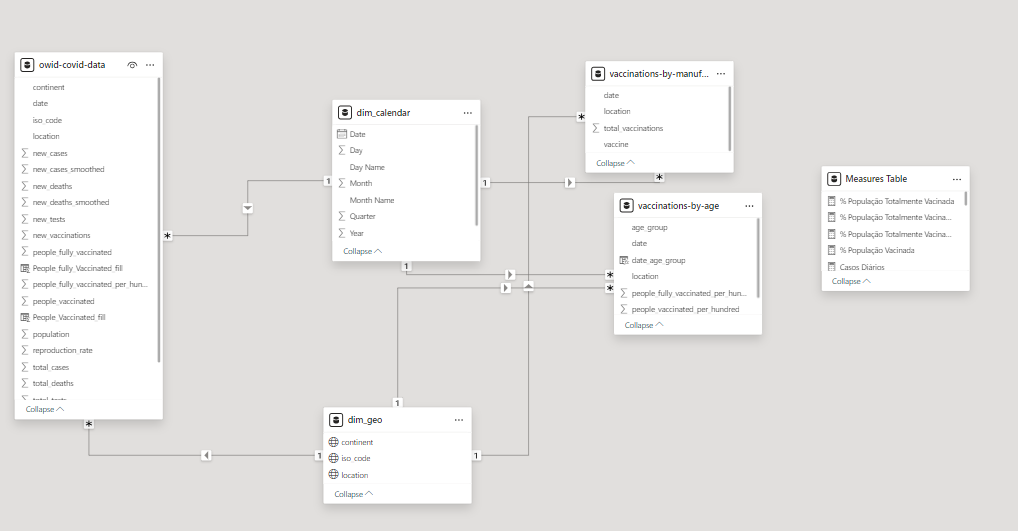
Var \_Last\_date = maxx(filter('owid-covid-data',[date]<earlier([date])&&[total\_vaccinations] > 0&&[iso\_code]=earlier([iso\_code])),[date])

var \_Last\_value = COALESCE([total\_vaccinations],maxx(filter('owid-covid-data',[date]=\_Last\_date&&[total\_vaccinations] > 0&&[iso\_code]=earlier([iso\_code])),[total\_vaccinations]))

return

\_Last\_value

**Dim\_calendar**Adicionada a coluna Year\_Month com a concatenação das colunas Ano e MêsYear\_Month = CONCATENATE(dim\_calendar[Year]&"-",dim\_calendar[Month])

**Modelo de Dados:**  
  
A propriedade Cross-filter direction foi alterada de “both” para “single” em todas as relações entre tabelas.  
  
Desta forma garantimos que os filtros geográficos ou temporais são corretamente aplicados e evitamos dependência circular entre duas ou mais tabelas (referencias cíclicas)  
  
  
  
**Medidas**Nas medidas saliento as seguintes expressões:  
 **1**

População Totalmente Vacinada =

VAR SummaryTablePais =

    ADDCOLUMNS (

        VALUES ( 'dim\_geo'[location]),

        "@value",

        CALCULATE (

            LASTNONBLANKVALUE(

                'dim\_calendar'[Date],

                MAX('owid-covid-data'[People\_fully\_Vaccinated\_fill])

            )

        )

    )

    VAR SummaryTableContinent =

    ADDCOLUMNS (

        VALUES ( 'owid-covid-data'[location]),

        "@value",

        CALCULATE (

            LASTNONBLANKVALUE(

                'dim\_calendar'[Date],

                MAX('owid-covid-data'[People\_fully\_Vaccinated\_fill])

            )

        )

    )

VAR Mundo =

        CALCULATE (

            LASTNONBLANKVALUE(

                'dim\_calendar'[Date],

                MAX('owid-covid-data'[People\_fully\_Vaccinated\_fill])

            ) ,

            'owid-covid-data'[location] = "World")

return (

if([Count Paises Filtrados] > 0, SUMX ( SummaryTablePais, [@value] ), if([Count Continentes Filtrados] > 0 , SUMX ( SummaryTableContinent, [@value] ),Mundo))

)

Esta expressão permite calcular corretamente a soma das vacinas dependentemente da seleção que se faça nas slicers dos países e/ou continentes.

Doutra forma, ao selecionar mais do que 1 país, ou mais do que 1 continente ou removendo qualquer seleção, a agregação não era corretamente realizada originando valores incorretos.  
  
(A expressão foi replicada de forma semelhante noutras medidas)

**2**

**Dt. 80% Imun. =** MINX( FILTER('dim\_calendar', 'Measures Table'[% População Totalmente Vacinada] >= {80}), dim\_calendar[Date])

% População Totalmente Vacinada = 100\*(DIVIDE([População Totalmente Vacinada],[População]))

Esta expressão permite obter, para uma dada seleção a data mínima em que a % de vacinação atingiu os 80% (imunidade de grupo).  
  
Nota que, por forma a simplificar a expressão, fazemos uso de medidas previamente calculadas

**3**LASTNONBLANKVALUE(

  'dim\_calendar'[Date],

       MAX('owid-covid-data'[People\_fully\_Vaccinated\_fill])

     )  
  
Esta expressão permite obter o último valor não vazio numa determinada coluna