

Guia das Etapas Realizadas no Projeto

Descrição documento	1
Estruturação dos diretórios	1
ETL implementado no projeto.....	2
Consumindo as Informações do Projeto no Power BI	6
Demonstração das análises criadas no Power BI.....	7







Descrição documento

Este arquivo contém todo o racional utilizado na construção do projeto, considerando o ETL, scripts SQL para criação de tabelas, consultas e criação do Dashboard utilizando o Power BI.

Estruturação dos diretórios

Inicialmente, criou-se a seguinte estruturação dos diretórios:

Figura 1 - Estruturação dos diretórios do projeto

	data	08/10/2021 18:13	Pasta de arquivos
	documentacao-projeto	11/10/2021 09:28	Pasta de arquivos
	ETL	09/10/2021 08:54	Pasta de arquivos
	pbix	09/10/2021 10:47	Pasta de arquivos
	scripts-sql	09/10/2021 08:38	Pasta de arquivos
	utilitarios	08/10/2021 18:35	Pasta de arquivos

Cada diretório possui o objetivo de armazenar arquivos específicos do projeto, sendo eles:

- **data:** os arquivos disponibilizados para realização do desafio estão neste diretório.
- **documentação-projeto:** o diretório em questão possui este arquivo.
- **ETL:** todo o racional utilizando o software Pentaho Data Integration (PDI) na versão 9.2 para construção das transformações e job.
- **pbix:** Este diretório possui o projeto criado em Power BI bem como as imagens utilizadas.
- **scripts-sql:** possui os scripts utilizados para criação das tabelas.

- **utilitários:** possui links auxiliares para realizar o download e configurar o Pentaho Data Integration, além disso, possui alguns drivers de conexão com o MSSQL.

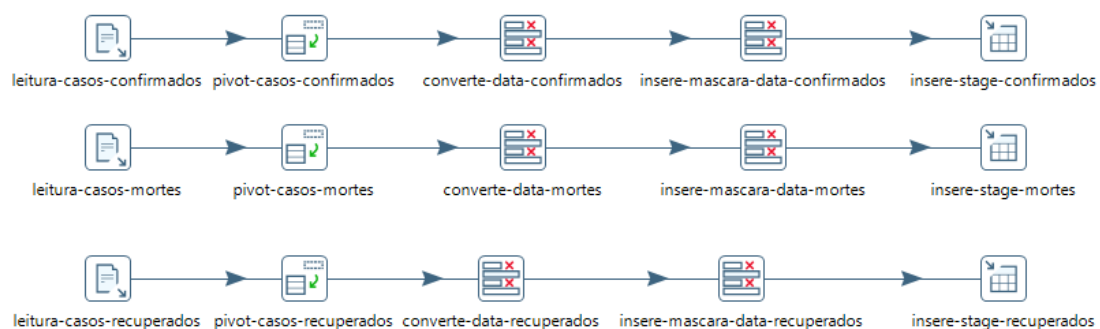
ETL implementado no projeto

Pensando no fluxo do ETL para este projeto, foi criado um banco de dados denominado DESAFIO-RADIX. Neste banco foram criados dois schemas (embora a arquitetura pudesse ser bancos de dados em servidores diferentes, dentre outros modelos) sendo eles o schema stage e o schema dw.

Os scripts para essa primeira etapa podem ser visualizados no arquivo **1.script-criacao-database-schemas** localizado no diretório de scripts-sql.

Após a criação do database e dos schemas, foi elaborado o fluxo abaixo, no PDI para busca dos dados brutos disponibilizados no desafio e posterior carga nas tabelas presentes no schema stage:

Figura 2 - Fluxo de Ingestão Dados Brutos para Stage

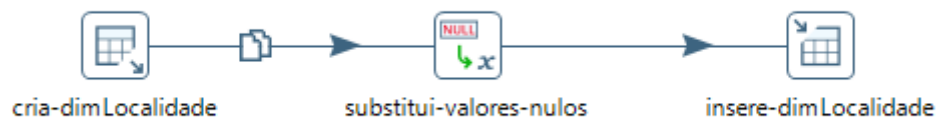


Este fluxo, basicamente busca os dados em formato csv, realiza um unpivot das colunas de data transformando-as em linhas e, posteriormente, realiza uma conversão nas datas e a atribuição de uma máscara (yyyy-MM-dd) para salvá-las no formato date nas respectivas tabelas.

Para este fluxo, foi definida sempre uma carga full, onde a tabela sempre é truncada e posteriormente os registros são inseridos, embora fosse possível outros tipos de carga, tais como a carga incremental. Após a conclusão da definição do fluxo foi criado um script SQL para criação das tabelas. Esses scripts podem ser visualizados no arquivo **2.script-criacao-tabelas-stage**.

Após a finalização da stage, o primeiro passo foi criar a tabela dimensão localidade. O seu fluxo foi definido reunindo as informações de estado, país, latitude e longitude de todas as três tabelas salvas no schema da stage.

Figura 3 - Fluxo criação dimensão localidade



No fluxo mencionado, a dimLocalidade foi criada através um script SQL que realiza o UNION das tabelas criadas na stage pegando valores distintos e os unindo (não há valores de países e estados diferentes nas tabelas, mas por precaução se futuramente houver, a estrutura estaria pronta para armazenar dados mais completos nesta dimensão).

Figura 4 - Query utilizada para montagem da dimensão localidade

```

SELECT DISTINCT
    LATITUDE,
    LONGITUDE,
    ESTADO,
    PAIS
FROM STAGE.RECUPERADOS

UNION

SELECT DISTINCT
    LATITUDE,
    LONGITUDE,
    ESTADO,
    PAIS
FROM STAGE.CONFIRMADOS

UNION

SELECT DISTINCT
    LATITUDE,
    LONGITUDE,
    ESTADO,
  
```

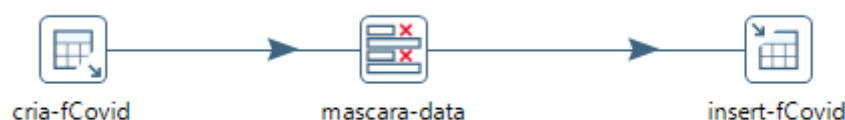
O script SQL pode ser visualizado ao abrir a transformação no PDI conforme imagem anterior. Vale ressaltar que o PDI poderia realizar o DISTINCT e o UNION internamente através de alguns steps, mas em alguns testes e com base na minha experiência profissional, certas modificações possuem um ganho de desempenho maior se realizadas via SQL, sem falar que é uma linguagem de fácil entendimento, capaz de ser absorvida rapidamente por outras pessoas.

Após isso, percebendo que diversos estados não foram informados, foi realizado um tratamento substituindo valores nulos pelo valor “Não Informado”. E por fim, os dados foram inseridos na dimLocalidade.

Por fim, foi realizado o fluxo para a tabela fato de covid. Vale ressaltar que a dimensão calendário optei por criar diretamente no Power BI, e será abordada posteriormente.

O fluxo da tabela fato, foi bem simples e ficou da seguinte forma:

Figura 5 - Fluxo para criação da tabela fato



Na criação da fcovid, tem-se um script SQL que une os dados necessários através de JOINS com as colunas de longitude, latitude, data e país com o objetivo de concentrar em apenas uma tabela as informações de casos confirmados, recuperados e óbitos.

Segue abaixo uma amostra da consulta utilizada para junção de todos os dados:

Figura 6 - Query utilizada para construção da tabela fato

```

USE [DESAFIO-RADIX]

SELECT DISTINCT
LOCALIDADE.FK_LOCALIDADE,
CONFIRMADOS.DATA,
--UTILIZAÇÃO DA FUNÇÃO LAG PARA DESCOBRIR OS CASOS NO DIA EM QUESTÃO PEGANDO A DATA ANTERIOR E REALIZANDO A SUBTRAÇÃO DO ACUMULADO POR ELA--
CONFIRMADOS.QTD_CONFIRMADOS_ACUMULADOS - ISNULL(LAG (CONFIRMADOS.QTD_CONFIRMADOS_ACUMULADOS, 1) OVER ( PARTITION BY LOCALIDADE.FK_LOCALIDADE ORDER BY LOCALI
CONFIRMADOS.QTD_CONFIRMADOS_ACUMULADOS,
MORTES.QTD_MORTES_ACUMULADAS - ISNULL(LAG (MORTES.QTD_MORTES_ACUMULADAS, 1) OVER ( PARTITION BY LOCALIDADE.FK_LOCALIDADE ORDER BY LOCALIDADE.FK_LOCALIDADE,
MORTES.QTD_MORTES_ACUMULADAS,
RECUPERADOS.QTD_RECUPERADOS_ACUMULADOS - ISNULL(LAG (RECUPERADOS.QTD_RECUPERADOS_ACUMULADOS, 1) OVER ( PARTITION BY LOCALIDADE.FK_LOCALIDADE ORDER BY LOCALI
RECUPERADOS.QTD_RECUPERADOS_ACUMULADOS

FROM STAGE.CONFIRMADOS CONFIRMADOS
INNER JOIN STAGE.MORTES MORTES
ON CONFIRMADOS.LATITUDE = MORTES.LATITUDE
AND CONFIRMADOS.LONGITUDE = MORTES.LONGITUDE
AND CONFIRMADOS.DATA = MORTES.DATA
AND CONFIRMADOS.PAIS = MORTES.PAIS
  
```

Vale ressaltar que os casos informados são os casos acumulados, desse modo, foram avaliadas algumas formas de descobrir a quantidade de novos casos em uma determinada data.

Alguns modos avaliados, foram via DAX no Power BI, via linguagem M no Power Query do Power BI, via Steps no Pentaho e via consulta SQL. Optei por uma consulta SQL, visto que, se futuramente a ferramenta de ETL ou a ferramenta de visualização de dados for substituída, toda a inteligência por trás dessas regras permanecerão imutáveis no SQL. Para isso, foi utilizada a função LAG para descobrir os casos da data anterior e após essa descoberta, seu valor é subtraído pela data atual.

Para demonstração, foi elaborado um exemplo, conforme tabela abaixo.

Tabela 1 - Demonstração do funcionamento da lógica adotada utilizando a função LAG

Data	Quantidade de Casos Acumulados
01/05/2021	10
02/05/2021	15
03/05/2021	22

Com base nos dados acima, na data do dia 02/05/2021, tem-se 15 casos acumulados, desse modo, com a lógica aplicada via SQL, é possível capturar os dados da data anterior (agrupando por país, estado, latitude, longitude, etc. No exemplo, os dados foram agrupados pela FK da dimensão localidade) e após essa captura, torna-se possível a subtração:

03/05/2021 → 22 casos

- Aplica-se o cálculo: 03/05/2021 (22 casos) – 02/05/2021 (15 casos) = 7 novos casos.

02/05/2021 → 15 casos

- Aplica-se o cálculo: 02/05/2021 (15 casos) – 01/05/2021 (10 casos) = 5 novos casos.

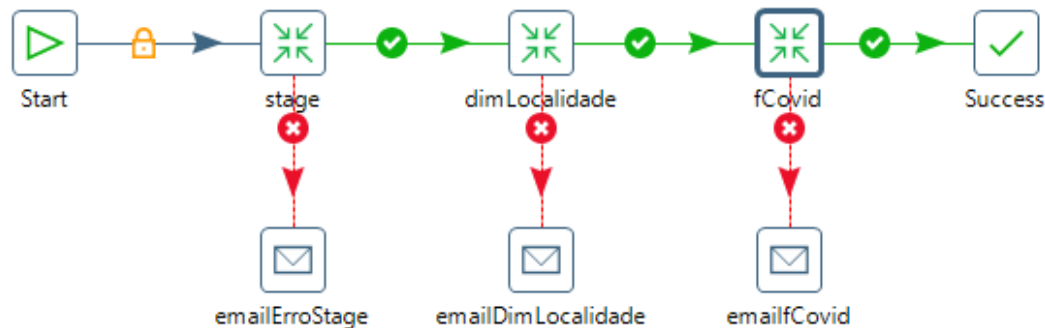
01/05/2021 → 10 casos

Aplica-se o cálculo: 01/05/2021 (10 casos) – Não possui data anterior = primeira data, logo 10 casos.

Após a obtenção dos dados, realizou uma nova atribuição de máscara na data retornada pela consulta SQL e por fim, as informações foram salvas na tabela fCovid.

Finalizando esse fluxo, foi criado um job capaz de executar todas as transformações em sequência e ainda por cima, seria possível criar um alerta via e-mail se a execução falhasse. Para que o alerta funcione, seria necessário informar algumas credenciais. Neste caso não informei por serem minhas credenciais privadas, mas segue um esboço:

Figura 7 - Job para orquestração do fluxo de ETL



Em caso de sucesso, a próxima transformação é executada. Em caso de erro, um e-mail será enviado e o fluxo interrompido. Desse modo, seria mais fácil identificar em qual etapa houve o erro diminuindo assim o tempo de busca e aumentando o tempo de correção.

Consumindo as Informações do Projeto no Power BI

Após a conclusão dessa etapa, todas as tabelas do schema dw foram importadas no Power BI, possibilitando uma maior velocidade graças as propriedades de query folding.

Figura 8 - Importação das Tabelas do DW no Power BI

Consultas [5]

- Parâmetros [2]
 - Database (DESAFIO-...)
 - Servidor (localhost)
- Fatos [1]
 - fCovid
- Dimensões [1]
 - dimLocalidade
- Outras Consultas [1]
 - _Medidas

Fórmula de DAX: `= Table.RenameColumns(dw_fCovid,{{"DATA", "Data"}, {"QTD_CONFIRMADOS", "Confirmados"}},`

	PK_LOCALIDADE	Data	Confirmados	Confirmados Acumulados	Mortes
1	1	22/01/2020	0	0	0
2	1	23/01/2020	0	0	0
3	1	24/01/2020	0	0	0
4	1	25/01/2020	0	0	0
5	1	26/01/2020	0	0	0
6	1	27/01/2020	0	0	0
7	1	28/01/2020	0	0	0
8	1	29/01/2020	0	0	0
9	1	30/01/2020	0	0	0
10	1	31/01/2020	0	0	0

Config. Consulta

PROPRIEDADES

Nome: fCovid

ETAPAS APLICADAS

- Source
- Navegação
- Colunas Renomeadas

Os dados importados conforme imagem acima, ainda possuíam a adição de dois parâmetros informado o host e database utilizados, pensando na situação em que as fontes de dados precisariam ser alteradas. Desse modo, trocando essas informações apenas uma vez, todas as tabelas originadas desse servidor e base seriam atualizadas.

A tabela dimensão calendário foi criada via Dax calculando a menor data e maior data para que os dados sejam sempre dinâmicos:

Figura 9 - Script em DAX para criação da tabela dimensão calendário

×

✓

```

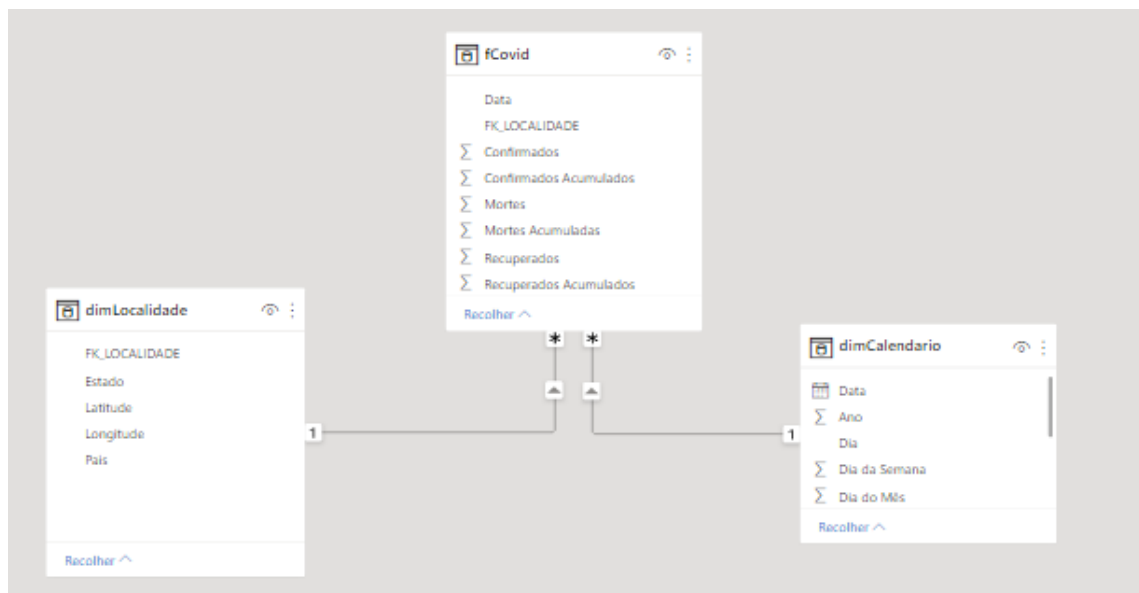
1 dimCalendario =
2
3 ADDCOLUMNS(
4     CALENDAR( MIN( fCovid[Data] ), MAX( fCovid[Data] ) ),
5     "Dia",FORMAT([Date],"dddd")
6     ,"Dia da Semana",(WEEKDAY([Date],1))
7     ,"Dia do Mês",DAY([Date])
8     ,"Mês", FORMAT([Date],"MMMM")
9     ,"Número do Mês", MONTH([Date])
10    ,"Mês e Ano", FORMAT([Date],"MMMM") & " de " & YEAR([Date])
11    ,"Ordenador Mês e Ano", FORMAT([Date], "yyymm")
12    ,"Ano",YEAR([Date])
13    ,"Dia Mês e Ano", DAY([Date]) & " de " & FORMAT([Date],"MMMM") & " de " & YEAR([Date])
14 )
15
16
17
18
19

```

Data	Dia	Dia da Semana	Dia do Mês	Mês	Número do Mês	Mês e Ano	Ano	Ordenador Mês e Ano	Dia Mês e Ano	Ordenador Dia Mês e Ano
22/01/2020	quarta-feira		4	22	janeiro	1 janeiro de 2020	2020	202001	22 de janeiro de 2020	20200122
23/01/2020	quinta-feira		5	23	janeiro	1 janeiro de 2020	2020	202001	23 de janeiro de 2020	20200123
24/01/2020	sexta-feira		6	24	janeiro	1 janeiro de 2020	2020	202001	24 de janeiro de 2020	20200124
25/01/2020	sábado		7	25	janeiro	1 janeiro de 2020	2020	202001	25 de janeiro de 2020	20200125
26/01/2020	domingo		1	26	janeiro	1 janeiro de 2020	2020	202001	26 de janeiro de 2020	20200126
27/01/2020	segunda-feira		2	27	janeiro	1 janeiro de 2020	2020	202001	27 de janeiro de 2020	20200127
28/01/2020	terça-feira		3	28	janeiro	1 janeiro de 2020	2020	202001	28 de janeiro de 2020	20200128
29/01/2020	quarta-feira		4	29	janeiro	1 janeiro de 2020	2020	202001	29 de janeiro de 2020	20200129

Desse modo, o modelo do Power BI ficou da seguinte forma:

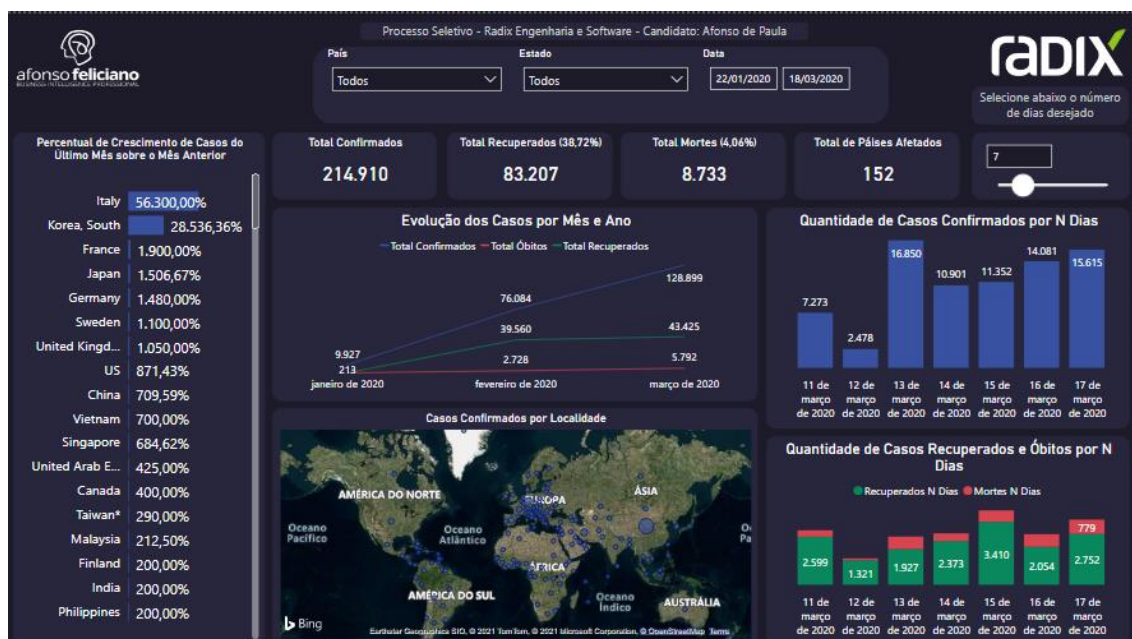
Figura 10 - Modelo star schema no Power BI



Demonstração das análises criadas no Power BI

Por fim, as análises criadas podem ser observadas conforme imagens abaixo. Recomenda-se visualizá-las abrindo o arquivo do projeto que pode ser localizado no diretório PBIX.

Figura 11 - Página 1 do Projeto de Análise de Casos de Covid



Nesta primeira página, pode-se observar os filtros de País, Estado e Data. Além disso, foi criado um filtro para realizar os cálculos da quantidade de casos confirmados, recuperados e óbito por um número N de dias.

Ademais, foram criados os indicadores para avaliar o percentual de crescimento de casos confirmados conforme data selecionada. Desse modo, esse indicador calcula os casos do último e penúltimo mês, realizando um comparativo percentual. Observou-se que no mês de fevereiro se comparado a janeiro, a Itália foi o país com o maior número de casos confirmados.

Além disso, pode-se observar um aumento na tendencia de casos confirmados para todos os meses. Um aumento no número de casos recuperados no mês de janeiro para fevereiro bem como a estabilização de fevereiro para março.

O número de óbitos, sofreu um grande aumento de janeiro para fevereiro, e após isso, dobrou, de fevereiro para março. Por fim, pode-se observar os casos confirmados através do mapa, no qual indica uma grande concentração de casos no continente asiático.

A seguir, pode-se observar outros indicadores presentes na segunda página do projeto.

Figura 12 - Página 2 do Projeto de Análise de Casos de Covid



Na segunda página do projeto, foram criados indicadores levando em consideração a média móvel de casos dos últimos 14 dias. Desse modo, pode-se observar que os óbitos e casos confirmados entraram em uma tendência de alta, considerando todo período do projeto.

Por fim, criou-se uma matriz com os casos por localidade, considerando a sua quantidade e o seu percentual. Surpreendentemente, a China, país com maior número de casos confirmados até aquele momento, já contava com mais de 86% de casos recuperados.