

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
**CURSO DE ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS**  
**PROJETO INTEGRADOR DO MÓDULO 1**

## **INTRODUÇÃO**

---

O presente documento tem o objetivo de descrever o projeto envolvendo as disciplinas estudadas durante o módulo 2 do curso de especialização de Ciência de Dados pela UTFPR.

O projeto tem a proposta transversal de trabalhar todo o conteúdo abordado nas disciplinas de *Preparação Pedagógica, Linguagem de Programação para Ciência de Dados e Administração e Gerenciamento de Banco de Dados*.

O trabalho foi desenvolvido pelo seguinte grupo:

- Arturo Vaine
- Írio Moesch
- Robson Mamede

## **CONTEXTO**

---

Para este trabalho utilizamos duas bases de dados.

A primeira foi a mesma utilizada no Projeto Integrador do Módulo 1 deste curso e consiste em indicadores de desenvolvimento no mundo fornecido pelo Banco Mundial. Utilizamos uma base mais atualizada com dados até 2018 (<https://www.kaggle.com/theworldbank/world-development-indicators>). Detalhes sobre essa base podem ser consultados no documento do Projeto do Módulo 1 (UTFPR\_CECDAOS\_Projeto\_Integrador\_Modulo\_1\_4.0.pdf) que acompanha o pacote de entregas deste trabalho.

A base mais atual foi alterada de forma que os anos, antes listados na forma de registros, agora são dispostos em coluna. Assim os indicadores aparecem uma única vez por país e um total de 58 colunas, guardam seus respectivos valores de aferição. Isso se mostrou um desafio para colocar no modelo relacional, que tivemos que resolver na aplicação Python.

A segunda base é composta de dados fornecidos pela Organização Mundial de Saúde sobre a COVID-19 e disponibilizado pela Microsoft em <https://github.com/microsoft/Bing-COVID-19-Data>. Esta é uma base mais direta e sem problema de normalização ao ponto de a importação quase não ter sofrido tratamento. Dela, aproveitamos seis colunas de interesse.

```

df_covid =
    pd.read_csv(DADOS_COVID)[['Updated', 'Confirmed', 'Deaths',
                               | 'Latitude', 'Longitude', 'ISO3']]
df_covid

```

	Updated	Confirmed	Deaths	Latitude	Longitude	ISO3
0	01/21/2020	262	0.0	NaN	NaN	NaN
1	01/22/2020	313	0.0	NaN	NaN	NaN
2	01/23/2020	578	0.0	NaN	NaN	NaN
3	01/24/2020	841	0.0	NaN	NaN	NaN
4	01/25/2020	1320	0.0	NaN	NaN	NaN
...	...	...	...	...	...	...
1580285	02/28/2021	36058	1463.0	-19.00046	29.87184	ZWE
1580286	03/01/2021	36086	1463.0	-19.00046	29.87184	ZWE
1580287	03/02/2021	36115	1468.0	-19.00046	29.87184	ZWE
1580288	03/03/2021	36147	1468.0	-19.00046	29.87184	ZWE
1580289	03/04/2021	36179	1478.0	-19.00046	29.87184	ZWE

1580290 rows × 6 columns

ISO3 é uma coluna com três caracteres que identificam o país. As demais são autoexplicativas.

## DISCIPLINA DE PREPARAÇÃO PEDAGÓGICA

Aqui, preparamos um Plano de Ensino e Plano de Aula. O primeiro descreve o treinamento para o trabalho que aqui desenvolvemos com todos os elementos necessários: público-alvo, carga horária etc. No segundo, descrevemos o conteúdo detalhado, metodologia, referências entre outros.

### PLANO DE ENSINO

<b>CURSO</b>	<b>Ciência de Dados</b>
--------------	-------------------------

<b>DISCIPLINA/UNIDADE CURRICULAR</b>	<b>CÓDIGO</b>	<b>PERÍODO</b>	<b>CARGA HORÁRIA (qtde. aulas)</b>
<b>Análise de Dados e Meios de Representação Gráfica</b>	<b>ADMRG</b>	<b>2</b>	<b>20</b>

<b>PRÉ-REQUISITO</b>	<b>Computação II</b>
----------------------	----------------------

#### OBJETIVOS

Fundamentar aos alunos uma visão global sobre um processo de implementação de banco de dados e visualização de dados como um todo, destacando os conhecimentos necessários para a aplicação. Possibilitar a aprendizagem de base sobre PostgreSQL e Python.

#### EMENTA

Utilização de um SGBD: o PostgreSQL e pgAdmin 4. Linguagem de definição de dados (DDL). Linguagem de manipulação de dados (DML). Visualização de planos de consultas. Estruturas de indexação. Linguagem de controle de dados. Controle de transações em SGBD relacional. PLPGSQL. Tipos de coleções em Python. Listas e Dicionários. Funções. Tuplas e Conjuntos (Sets). Bibliotecas Pandas, NumPy e Matplotlib.

#### CONTEÚDO PROGRAMÁTICO

<b>ITEM</b>	<b>EMENTA</b>	<b>CONTEÚDO</b>
<b>1</b>	<b>Utilização de um SGBD: o PostgreSQL e pgAdmin 4.</b>	<b>Conceitos iniciais de um SGBD. Primeiras etapas de utilização do pgAdmin4.</b>
<b>2</b>	<b>Linguagem de definição de dados (DDL). Linguagem de manipulação de dados (DML).</b>	<b>Importância dos tipos de linguagens de definição e manipulação de dados. Exemplos práticos. Exercícios de fixação.</b>

3	Visualização de planos de consultas. Estruturas de indexação.	Exemplos práticos. Exercícios de fixação.
4	Linguagem de controle de dados. Controle de transações em SGBD relacional.	Sintaxe de operações. Exemplos práticos. Exercícios de fixação.
5	PLPGSQL.	Importância do uso de linguagem estrutural. Exemplos práticos. Exercícios de fixação.
6	Tipos de coleções em Python. Listas e Dicionários. Tuplas e Conjuntos (Sets).	Diferenças e semelhanças entre os tipos de coleções. Vantagens e desvantagens de cada tipo. Exemplos práticos. Exercícios de fixação.
7	Funções.	Sintaxe de funções. Motivação do uso de funções. Funções anônimas (lambda). Exemplos práticos. Exercícios de fixação.
8	Bibliotecas Pandas, NumPy.	Motivação do uso de funções das bibliotecas. Exemplos práticos. Exercícios de fixação.
9	Matplotlib.	Sintaxe de plotagem. Tipos de gráficos. Boas práticas a utilizar na representação gráfica de dados. Exemplos práticos. Exercícios de fixação.

PROFESSOR	TURMA
	S43

PROGRAMAÇÃO E CONTEÚDOS DAS AULAS (PREVISÃO)		
Dia/Mês ou Semana	Conteúdo das Aulas,	Número de Aulas
Semana 1	Introdução da disciplina. Linguagens de programação usadas em um Sistema Gerenciador de Banco de Dados (SGBD) e sua relevância na Ciência de Dados. Utilização de um SGBD: o PostgreSQL e pgAdmin 4.	2
Semana 2	Linguagem de definição de dados (DDL): criação, remoção e alterações estruturais de objetos armazenados em um banco de dados relacional.  Linguagem de manipulação de dados (DML): inserção, remoção, alteração. Conceitos básicos das linguagens de: controle de dados e controle de transações.	2
Semana 3	Visualização de planos de consultas. Estruturas de indexação. Linguagem de controle de dados. Controle de transações em SGBD relacional. PLPGSQL.	2
Semana 4	Revisão. Exercícios resolvidos.	2
Semana 5	Introdução do Python. Listas e Dicionários. Funções. Tuplas e Conjuntos (Sets).	2

Semana 6	Introdução ao Pandas (manipulação e análise de dados em Python).	<b>2</b>
Semana 7	Acessando banco de dados com Python. Introdução ao NumPy (computação científica com Python).	<b>2</b>
Semana 8	Introdução ao Matplotlib (visualização de dados em Python).	<b>2</b>
Semana 9	Revisão. Exercícios resolvidos.	<b>2</b>
Semana 10	Avaliação.	<b>2</b>

#### **PROCEDIMENTOS DE ENSINO**

##### **AULAS TEÓRICAS**

Aulas expositivas dialogadas.

##### **AULAS PRÁTICAS**

Exercícios individuais e em grupo nas aulas de revisão, solução de problemas utilizando ferramentas e conceitos expostos em aula. Aulas práticas em laboratório.

#### **PROCEDIMENTOS DE AVALIAÇÃO**

Avaliação será individual, com questões objetivas e discursivas.

#### **REFERÊNCIAS**

##### **Referências Básicas:**

CHAUHAN, C. PostgreSQL Cookbook. Birmingham, UK: Packt Publishing, 2015.

FORBELLONE, A. L. V. Lógica de programação: a construção de algoritmos e estruturas de dados. São Paulo, SP: Makron, 1993.

GUTTAG, J. Introduction to Computation and Programming Using Python: With Application to Understanding Data Second Edition. USA: MIT Press, 2016.

HUNT, J. A Beginners Guide to Python 3 Programming. Undergraduate Topics in Computer Science. Springer, 2019.

MACHADO, F. N. R. Banco de dados: projeto e implementação. São Paulo, SP: Erica, 2014.

MORAES, M. C. Educação a distância: fundamentos e práticas. Campinas, SP: Unicamp/Nied, 2002.

NELLI, F. Python data analytics: Data analysis and science using PANDAs, Matplotlib and the Python Programming Language. Apress, 2015.

QUINTELA, A. J. F.; ZAMBERLAN, M. F. Ambientação para EAD. Cuiabá, MT: Rede eTec Brasil, 2014.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. Sistema de banco de dados. Rio de Janeiro, RJ: Elsevier, 2012.

##### **Referências Complementares:**

ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. São Paulo, SP: Pearson Addison- Wesley, 2019.

LEE, K. D. Python Programming Fundamentals. Undergraduate Topics in Computer Science. Springer, 2014.

MCKINNEY, W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. Segunda edição, O'Reilly, 2017.

NASH, S.; PILJAK, Z; MOORE, M. Moodle Course Design Best Practices. England: Packt Publishing, 2014.

NIELD, T. Introdução à Linguagem SQL: Abordagem Prática Para Iniciantes. São Paulo, SP: Novatec, 2016.

<b>ORIENTAÇÕES GERAIS</b>

---

Assinatura do Professor

---

Assinatura do Coordenador do Curso

# PLANO DE AULA

## IDENTIFICAÇÃO

<b>Disciplina:</b>	Treinamento de Análise de Dados e Meios de Representação Gráfica
<b>Professor:</b>	

## TEMA DA AULA

Matplotlib.

## PRÉ-REQUISITOS

Noções em Python.

## OBJETIVOS

Fundamentar aos alunos a motivação do uso da biblioteca para a visualização de dados, destacando os requisitos necessários para uso da aplicação. Possibilitar a aprendizagem de código via exemplos práticos.

## DESENVOLVIMENTO DO TEMA

O tema proposto da aula será desenvolvido através de aula teórica para exposição da motivação e funcionalidades da biblioteca. Exposição da galeria de gráficos da documentação. Em seguida haverá atividade prática com exposição de resolução de problemas para fixação e discussão do conteúdo exposto, a fim de demonstrar aplicações práticas. Indicação de materiais complementares de pesquisa, lista de exercícios fixação.

## CONTEÚDO PROGRAMÁTICO

Sintaxe de plotagem. Tipos de gráficos. Boas práticas a utilizar na representação gráfica de dados. Exemplos práticos. Exercícios de fixação.

## AVALIAÇÃO

Avaliação será individual, com questões objetivas e discursivas ao final da disciplina.

## BIBLIOGRAFIA

GUTTAG, J. Introduction to Computation and Programming Using Python: With Application to Understanding Data Second Edition. USA: MIT Press, 2016.

NELLI, F. Python data analytics: Data analysis and science using PANDAs, Matplotlib and the Python Programming Language. Apress, 2015.



## DISCIPLINA DE ADMINISTRAÇÃO DE BANCO DE DADOS

Utilizamos duas bases de dados como origem. Uma base COVID com dados do mundo inteiro e a base de dados do Banco Mundial para desenvolvimento.

A base da COVID tem em torno de 1,5M de registros. Importamos todo o conteúdo.

A base de desenvolvimento mundial é mais complexa, pois além de não normalizada, o volume de dados é muito grande. Nossa estimativa era de que a base completa, levaria 21M de linhas, levando em conta que ela tem um histórico de aproximadamente 60 anos, 1500 indicadores, e 240 países. Assim, o filtro ficou apenas para o ano de 2018, que nos forneceu ~400K linhas.

Embora a criação da base e a importação dos dados sejam feitas pela aplicação Python, fornecemos um dump da base já pronta no PostgreSQL e se encontra no arquivo *dump\_bd\_pi\_modulo2.zip* que compõe o pacote de entregáveis.

Levando isso em conta, trabalhamos nos seguintes pontos:

### 1. ÍNDICES

Para os índices, pensamos nos seguintes:

- a. Um índice particionado na tabela *afericao* para o país 'BRA'. Como é o filtro de principal interesse e a base completa foi estimada em 21 milhões de linhas, é crucial seu uso.

```
288 ANALYSE afericao;
289
290 EXPLAIN
291 SELECT * FROM afericao a WHERE a.codigo_pais = 'BRA'
292
```

Data Output Explain Messages Notifications

QUERY PLAN	
	text
1	Gather (cost=1000.00..6894.50 rows=1592 width=45)
2	Workers Planned: 2
3	-> Parallel Seq Scan on afericao a (cost=0.00..5735.30 rows=663 width=45)
4	Filter: (codigo_pais = 'BRA'::bpchar)

Criação do índice:

```
295 CREATE INDEX idx_pais_bra ON afericao(codigo_pais)
296 WHERE (codigo_pais = 'BRA');
297
```

Data Output Explain Messages Notifications

CREATE INDEX

Query returned successfully in 211 msec.

Resultado:

292	EXPLAIN
293	SELECT * FROM afericao a WHERE a.codigo_pais = 'BRA'
294	
Data Output Explain Messages Notifications	
QUERY PLAN	
text	
1	Index Scan using idx_pais_bra on afericao a (cost=0.28..57.16 rows=1592 width=45)

292	EXPLAIN
293	SELECT * FROM afericao a WHERE a.codigo_pais = 'USA'
294	
Data Output Explain Messages Notifications	
QUERY PLAN	
text	
1	Gather (cost=1000.00..6894.50 rows=1592 width=45)
2	Workers Planned: 2
3	-> Parallel Seq Scan on afericao a (cost=0.00..5735.30 rows=663 width=45)
4	Filter: (codigo_pais = 'USA')::bpchar

- b. Índice de chave composta para a tabela de *afericao*. Como a trinca código do país, ano da aferição e indicador devem ser únicos para esta tabela, o índice é criado automaticamente quando os campos foram definidos com chave primária.

288	EXPLAIN
289	SELECT * FROM afericao a
290	WHERE 1 = 1
291	AND a.codigo_indicador = 'FB.BNK.CAPA.ZS'
292	AND a.codigo_pais = 'BRA'
293	AND a.ano_afericao = 2018
Data Output Explain Messages Notifications	
QUERY PLAN	
text	
1	Index Scan using pk_codigos_indicador_pais_ano on afericao a (cost=0.42..8.44 rows=1 width=45)
2	Index Cond: ((codigo_indicador = 'FB.BNK.CAPA.ZS')::bpchar) AND (codigo_pais = 'BRA')::bpchar

- c. Índice com busca de texto

294	EXPLAIN
295	SELECT * FROM afericao a
296	WHERE 1 = 1
297	AND a.codigo_indicador LIKE '%BNK%'
298	
Data Output Explain Messages Notifications	
QUERY PLAN	
text	
1	Gather (cost=1000.00..6842.38 rows=42 width=30)
2	Workers Planned: 1
3	-> Parallel Seq Scan on afericao a (cost=0.00..5838.18 rows=25 width=30)
4	Filter: ((codigo_indicador)::text ~~ '%BNK%':text)

299	CREATE INDEX idx_parte_cod_indicador
300	ON afericao USING GIN(codigo_indicador gin_trgm_ops);
Data Output Explain Messages Notifications	
CREATE INDEX	
Query returned successfully in 3 secs 356 msec.	

Resultado:

294	EXPLAIN
295	SELECT * FROM africao a
296	WHERE 1 = 1
297	AND a.codigo_indicador LIKE '%BNK%'
Data Output Explain Messages Notifications	
QUERY PLAN	
text	
1	Bitmap Heap Scan on africao a (cost=12.32..165.26 rows=42 width=30)
2	Recheck Cond: ((codigo_indicador)::text ~~ '%BNK%':text)
3	-> Bitmap Index Scan on idx_parte_cod_indicador (cost=0.00..12.31 rows=42 width=0)
4	Index Cond: ((codigo_indicador)::text ~~ '%BNK%':text)

d. Agora, um índice para busca dos demais países diferentes de Brasil.

Resultado sem o índice:

291	EXPLAIN
292	SELECT * FROM africao a
293	WHERE 1 = 1
294	AND a.codigo_pais = 'USA'
295	
Data Output Explain Messages Notifications	
QUERY PLAN	
text	
1	Gather (cost=1000.00..6997.18 rows=1590 width=30)
2	Workers Planned: 1
3	-> Parallel Seq Scan on africao a (cost=0.00..5838.18 rows=935 width=30)
4	Filter: (codigo_pais = 'USA':bpchar)

296	CREATE INDEX idx_cod_pais_geral ON africao(codigo_pais)
297	
Data Output Explain Messages Notifications	
CREATE INDEX	
Query returned successfully in 341 msec.	

Resultado com o índice:

291	EXPLAIN
292	SELECT * FROM africao a
293	WHERE 1 = 1
294	AND a.codigo_pais = 'USA'
295	
Data Output Explain Messages Notifications	
QUERY PLAN	
text	
1	Index Scan using idx_cod_pais_geral on africao a (cost=0.42..50.25 rows=1590 width=30)
2	Index Cond: (codigo_pais = 'USA':bpchar)

Resultado utilizando o filtro para país igual a 'BRA'. Aqui vemos que ele ignorou o índice particionado criado anteriormente exclusivo para o Brasil.

291	EXPLAIN
292	SELECT * FROM afericao a
293	WHERE 1 = 1
294	AND a.codigo_pais = 'BRA'
295	
Data Output Explain Messages Notifications	
QUERY PLAN	
text	
1	Index Scan using idx_cod_pais_geral on afericao a (cost=0.42..50.25 rows=1590 width=30)
2	Index Cond: (codigo_pais = 'BRA':bpchar)

- e. Os demais índices foram mais simples para atividades de apoios às nossas análises na base da COVID.

330	CREATE INDEX idx_covid_casos ON afericao_covid(casos_novos);
331	CREATE INDEX idx_covid_mortes ON afericao_covid(novas_mortes);
332	CREATE INDEX idx_covid_pais ON afericao_covid(codigo_pais);
Data Output Explain Messages Notifications	
CREATE INDEX	
Query returned successfully in 5 secs 42 msec.	

## 2. TRIGGERS

Criamos duas triggers que evitam operações de delete nas principais tabelas afericao e aferição\_covid.

```
create or replace function bloquear_operacao_geral()
returns trigger language plpgsql as $$
begin
return null;
end $$;
```

```
create trigger tg_evitar_delete before delete on afericao_covid for each row
execute procedure bloquear_operacao_geral();
```

```
create trigger tg_evitar_delete before delete on afericao for each row execute
procedure bloquear_operacao_geral();
```

## 3. USUÁRIOS E PERMISSÕES

Seguem as roles e os usuários criados para o projeto. Duas roles, geral e gestores. Para geral, permissão de leitura nas tabelas e para gestores, permissão de leitura, atualização e exclusão de registros. Inserção está de fora das permissões, pois isso é feito em batch pela aplicação Python. Temos um usuário na role geral; e 2 usuários como gestores.

```
CREATE ROLE geral;
GRANT SELECT ON afericao TO geral;
GRANT SELECT ON indicador TO geral;
GRANT SELECT ON categoria TO geral;
GRANT SELECT ON regioao TO geral;
GRANT SELECT ON pais TO geral;
GRANT SELECT ON afericao_covid TO geral;
```

```

CREATE USER usuario1 WITH PASSWORD '12354';
GRANT geral TO usuario1;

-----

CREATE ROLE gestores;
GRANT SELECT, UPDATE, DELETE ON pais TO gestores;
GRANT SELECT, UPDATE, DELETE ON afericao TO gestores;
GRANT SELECT, UPDATE, DELETE ON indicador TO gestores;
GRANT SELECT, UPDATE, DELETE ON categoria TO gestores;
GRANT SELECT, UPDATE, DELETE ON regiao TO gestores;
GRANT SELECT, UPDATE, DELETE ON afericao_covid TO gestores;

CREATE USER gestor1 WITH PASSWORD '12354';
CREATE USER gestor2 WITH PASSWORD '12354';

GRANT gestores TO gestor1;
GRANT gestores TO gestor2;

```

#### 4. TRANSAÇÕES.

Para este projeto, utilizamos duas transações no nível de bloqueio *read committed*, ou seja, em caso de concorrência de atualização, uma transação lerá a atualização da outra somente quando esta confirmar os dados. Em caso de atualização na mesma linha, a que chegar primeiro bloqueia o registro para as demais até que ela finaliza a sua participação.

No exemplo a seguir, as duas transações querem alterar a unidade monetária do país Brasil, que tem o valor original definido em 'Brazilian Real'

6

select \* from pais where codigo\_pais = 'BRA';

7

Data Output

Explain

Messages

Notifications

	codigo_pais [PK] character (3)	nome_pais character (50)	nome_completo character (150)	unidade_monetaria character (50)	codigo_regiao integer
1	BRA	Brazil	... Federative Republic ...	Brazilian real	3

T1 está no arquivo de script transação1.sql

T2 está no arquivo de script transação2.sql

Utilizamos a tabela 'países' para a demonstração. Segue a sequência de eventos que deve ser seguida.

- T1: abre os trabalhos iniciando a transação e define o nível de isolamento.

```

begin;
set transaction isolation level read committed;

```

- T2: também inicia sua transação e define seu nível de isolamento.

```
begin;
set transaction isolation level read committed;
```

- T1: atualiza a unidade monetária do país Brasil para Cruzeiro

```
update pais set unidade_monetaria = 'Cruzeiro' where codigo_pais = 'BRA';
```

- T2: tenta atualizar a unidade monetária do mesmo país, **mas é bloqueada pela T1**, pois estão trabalhando no mesmo registro. Neste momento T2 aguarda T1 terminar sua missão ou abortar.

```
update pais set unidade_monetaria = 'Cruzado' where codigo_pais = 'BRA';
```



- T1: se um SELECT for feito em T1 para ver o resultado do update, este é mostrado com o novo valor (Cruzeiro) de T1, mas ainda não está confirmado.

```
select * from pais where codigo_pais = 'BRA';
```

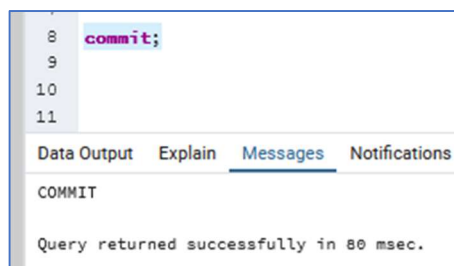
The screenshot shows a SQL client window with the following SQL code:

```
6 select * from pais where codigo_pais = 'BRA';
7
```

The window has tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the following columns and data:

	codigo_pais [PK] character (3)	nome_pais character (50)	nome_completo character (150)	unidade_monetaria character (50)	codigo_regiao integer
1	BRA	Brazil	Federative Republic of Brazil	Cruzeiro	3

- T1: ao confirmar a alteração, T1 terá enviado os dados para o disco ...



e liberado o registro para a T2 seguir sua vida.

```

3
4 update pais set unidade_monetaria = 'Cruzado' where codigo_pais = 'BRA';
5

```

Data Output Explain Messages Notifications

UPDATE 1

Query returned successfully in 4 min 41 secs.

- T2: ao verificar os dados em T2, apesar de T1 ter confirmado sua alteração, a transação de T2 ainda está ativa e com o seu update executado. Ou seja, o valor dela é o 'Cruzado'.

```

7 select * from pais where codigo_pais = 'BRA';

```

	codigo_pais [PK] character (3)	nome_pais character (50)	nome_completo character (150)	unidade_monetaria character (50)	codigo_regiao integer	
1	BRA	Brazil	Federative Republic ...	Cruzado	...	3

O valor de T1 é o que está valendo para o banco, 'Cruzeiro'.

```

6 select * from pais where codigo_pais = 'BRA';
7

```

	codigo_pais [PK] character (3)	nome_pais character (50)	nome_completo character (150)	unidade_monetaria character (50)	codigo_regiao integer	
1	BRA	Brazil	Federative Republic ...	Cruzeiro	...	3

- T2: ao confirmar sua transação, o valor 'Cruzado' passa a ser o valor de situação do banco.

```

5 commit;
6
7 select * from pais where codigo_pais = 'BRA';

```

	codigo_pais [PK] character (3)	nome_pais character (50)	nome_completo character (150)	unidade_monetaria character (50)	codigo_regiao integer	
1	BRA	Brazil	Federative Republic ...	Cruzado	...	3

- T1:

```

6 select * from pais where codigo_pais = 'BRA';
7

```

	codigo_pais [PK] character (3)	nome_pais character (50)	nome_completo character (150)	unidade_monetaria character (50)	codigo_regiao integer	
1	BRA	Brazil	Federative Republic ...	Cruzado	...	3

## DISCIPLINA DE LING. DE PROG. PARA CIÊNCIA DE DADOS

---

Conforme já explicitado, utilizou-se duas bases de dados para análise, uma do Banco Mundial (2021) e outra da Organização Mundial de Saúde (2021) disponibilizada na plataforma GitHub no perfil da empresa Microsoft (2021), para a visualização de alguns dos dados disponibilizados em colunas conforme tabela abaixo.

Column header	Description
ID	Unique identifier
Updated	Datetime in UTC
Confirmed	Confirmed case count for the region
ConfirmedChange	Change of confirmed case count from the previous day
Deaths	Death case count for the region
DeathsChange	Change of death count from the previous day
Recovered	Recovered count for the region
RecoveredChange	Change of recovered case counts from the previous day
Latitude	Latitude of the centroid of the region
Longitude	Longitude of the centroid of the region
ISO2	2 letter country code identifier
ISO3	3 letter country code identifier
Country_Region	Country/region
AdminRegion1	Region within Country_region
AdminRegion2	Region within AdminRegion1

Neste contexto de pandemia de COVID-19, pode-se destacar como um dos pontos relevantes de análise de condução do tratamento médico nos países, a densidade de médicos por 1000 habitantes nos países. A recomendação da Organização Mundial de Saúde (OMS, 2021) é de que haja a proporção mínima de 1 médico por 1000 habitantes, ou 10 por 10000 habitantes (KUMAR e PAL, 2018). As estatísticas disponíveis mostram que mais de 40% dos Estados Membros da OMS relatam ter menos de 10 médicos por 10.000 habitantes (mais de 26% relatam ter menos de 3).

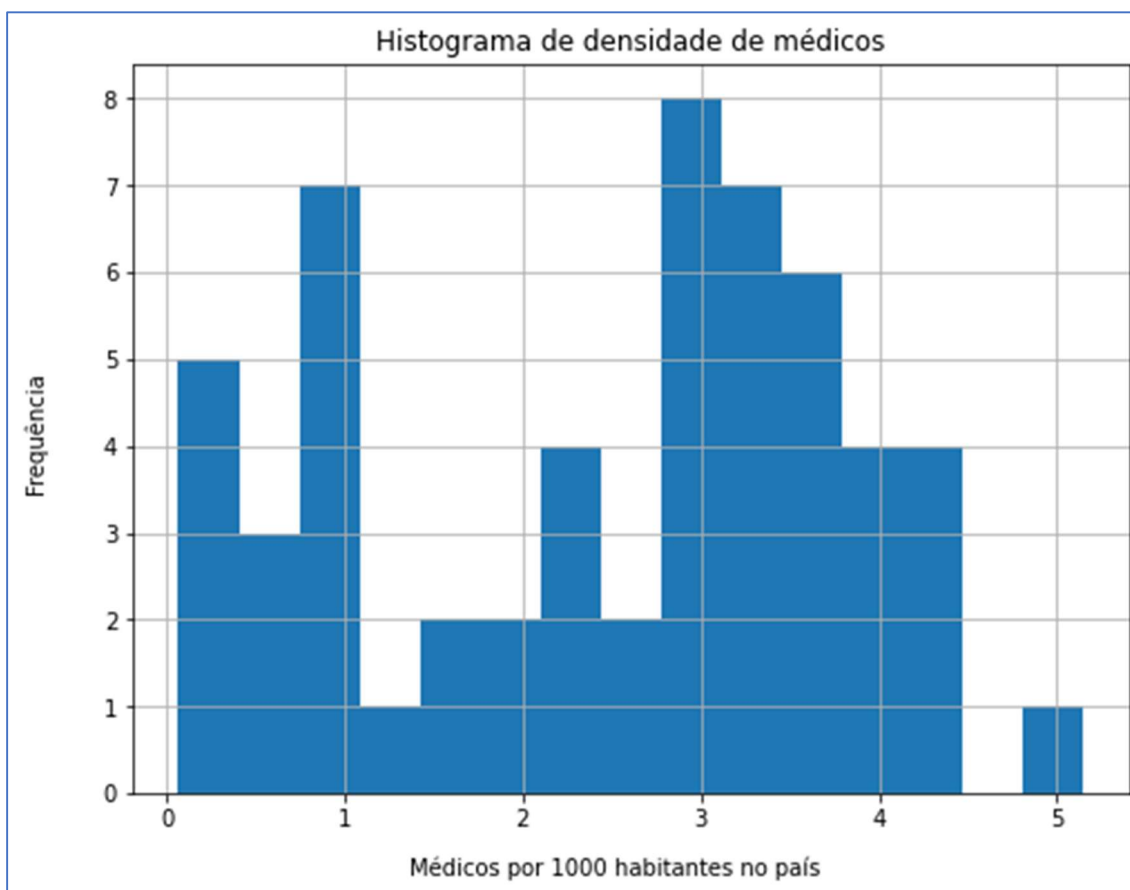
Os profissionais de saúde estão distribuídos de maneira bastante desigual em todo o mundo. Os países com as necessidades relativas mais baixas têm o maior número de profissionais de saúde, enquanto aqueles com maior carga de doenças devem se contentar com uma força de trabalho de saúde muito menor. A Região Africana sofre mais de 22% do fardo global de doenças, mas tem acesso a apenas 3% dos profissionais de saúde e a menos de 1% dos recursos financeiros mundiais (OMS, 2021).

Segundo os dados consolidados na OMS, há apenas informação da densidade no Brasil até o ano de 2013. Em comparação, para os outros países selecionados, há dados do ano de 2014, com valor de 3 a pouco mais de 4,1 médicos por 1000 habitantes.

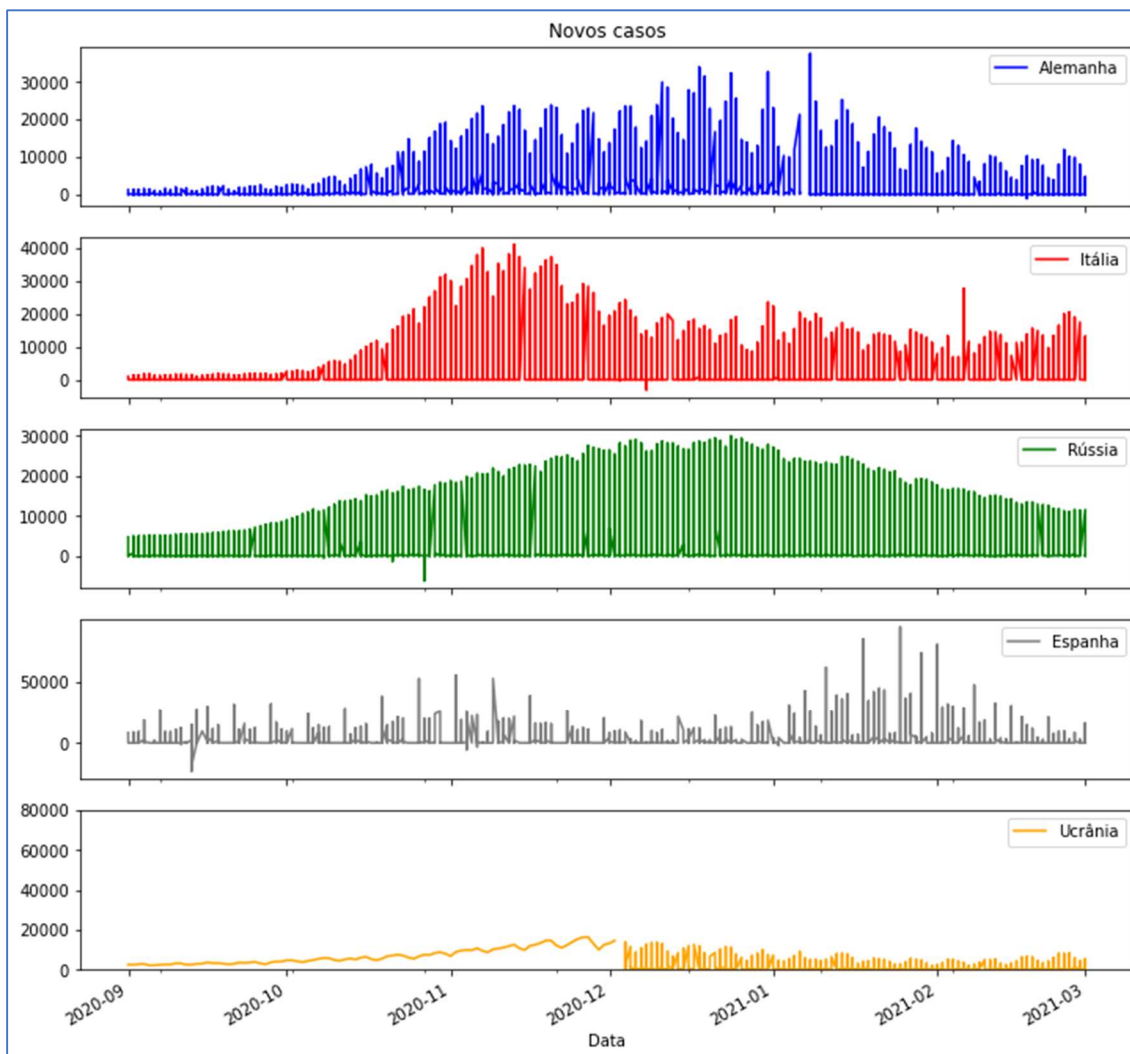


Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2010	2011	2012	2013	2014	2015	2016	2017	2018
Brazil	BRA	Physicians (per 1,000 people)	SH.MED.PHYS.ZS	0.374	0.0	0.0	0.0	0.0	0.4	...	1.787	1.821	0.000	1.852	0.000	0.000	0.000	0.0	0.0
Germany	DEU	Physicians (per 1,000 people)	SH.MED.PHYS.ZS	0.000	0.0	0.0	0.0	0.0	0.0	...	3.775	3.870	3.944	4.039	4.125	4.191	0.000	0.0	0.0
Italy	ITA	Physicians (per 1,000 people)	SH.MED.PHYS.ZS	0.000	0.0	0.0	0.0	0.0	0.0	...	0.000	0.000	3.861	3.930	3.945	3.898	4.021	0.0	0.0
Russian Federation	RUS	Physicians (per 1,000 people)	SH.MED.PHYS.ZS	0.000	0.0	0.0	0.0	0.0	0.0	...	5.000	4.244	4.031	4.022	4.163	3.975	0.000	0.0	0.0
Spain	ESP	Physicians (per 1,000 people)	SH.MED.PHYS.ZS	0.000	0.0	0.0	0.0	0.0	0.0	...	3.756	3.838	3.835	3.824	3.819	3.872	0.000	0.0	0.0
Ukraine	UKR	Physicians (per 1,000 people)	SH.MED.PHYS.ZS	0.000	0.0	0.0	0.0	0.0	0.0	...	3.494	3.495	3.501	3.506	3.000	0.000	0.000	0.0	0.0

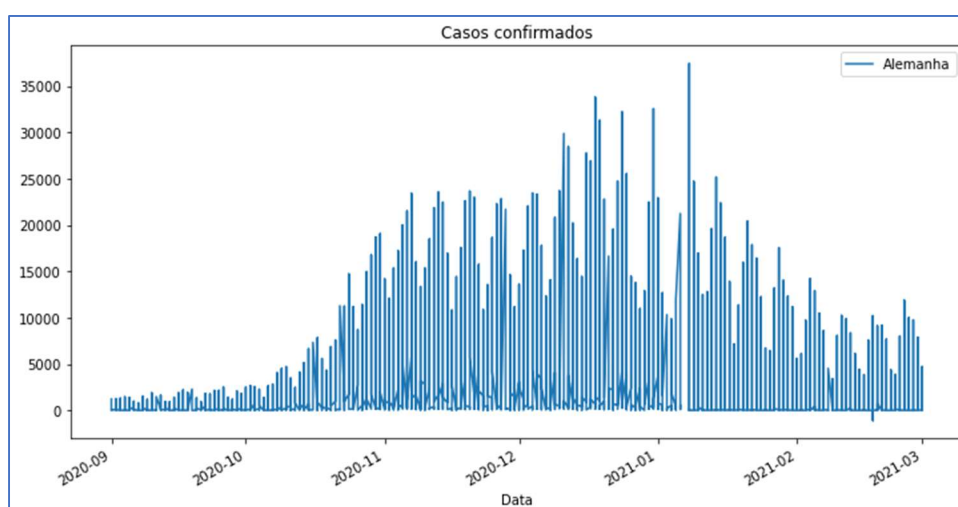
Por meio da extração do histograma de densidade médicos, é possível observar que há grande distribuição de países com 1 ou menos médicos por 1000 habitantes, assim como de 3 a 4 médicos por 1000 habitantes.

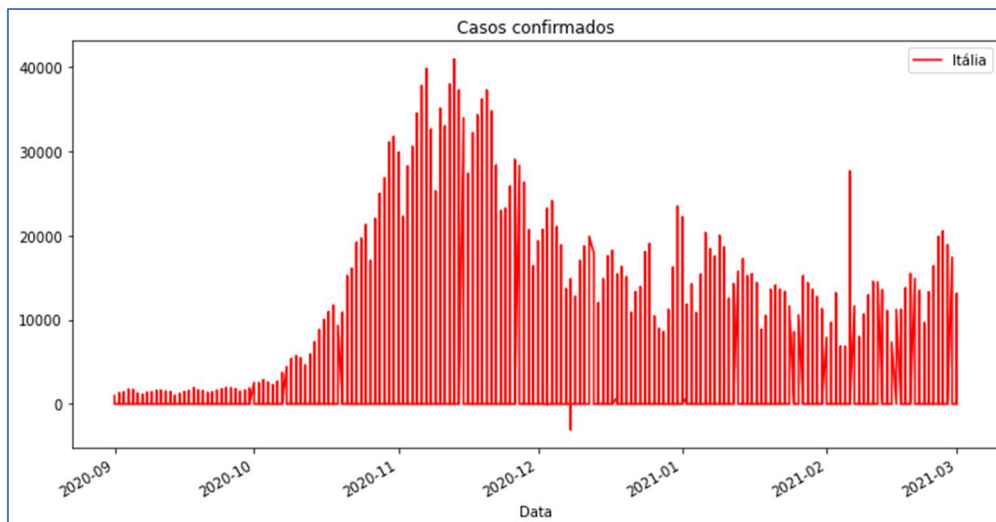


Verificando a evolução dos novos casos para países com grande proporção de médicos por 1000 habitantes, é possível observar diferentes “comportamentos” de novas infecções, em que até mesmo foram registradas possíveis inconsistências (valores negativos nos dados da Itália, Rússia e Espanha). Além disto, no caso dos registros da Ucrânia verificou-se relativa “falta de granularidade” dos dados no período de setembro a dezembro de 2020, ou ainda algum problema de consistência dos dados consolidados.

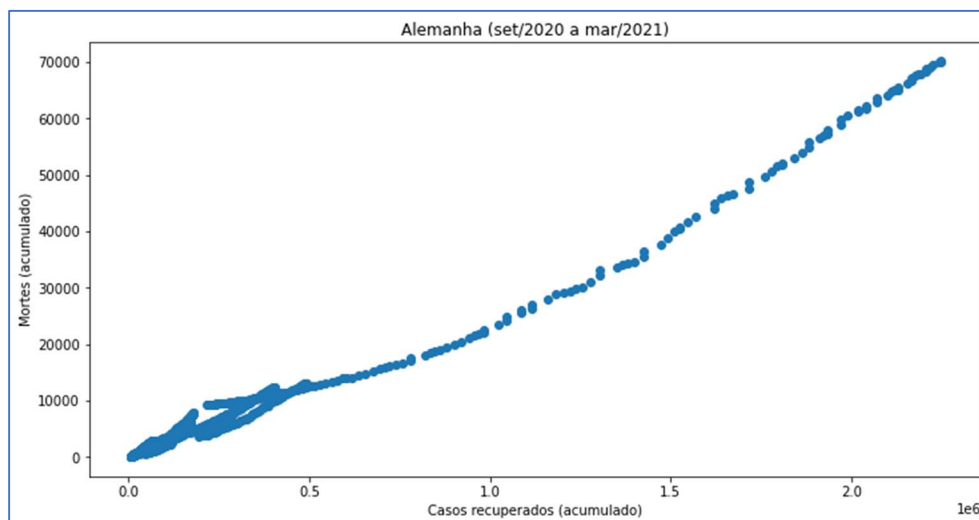


Se for observada a evolução dos novos casos apenas na Alemanha e Itália, é possível observar que a escalada foi relativamente maior na Itália, conforme também pode-se dizer que foi também percebido de senso comum pelos noticiários na época.





A fim de analisar a correlação (não necessariamente de causalidade) entre acumulado e mortes e de casos recuperados, foi gerado o gráfico correspondente ao período de 6 meses mais recentes (de setembro de 2020 a março de 2021) para os dados da Alemanha. É possível afirmar que não houve grande variação nos 3 meses mais recentes e a relação entre as duas variáveis foi de natureza aproximadamente linear.



# SUMÁRIOS DOS DATASFRAMES

## Indicadores do Banco Mundial (Tabela Principal)

[17]: df\_indicadores.describe()

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2010	2011	2012	2013
count	422136	422136	422136	422136	38296.0	42017.0	44198.0	44084.0	44611.0	47305.0	...	235880.0	226320.0	227731.0	222939.0
unique	264	264	1599	1599	33953.0	37862.0	39704.0	39704.0	40151.0	42061.0	...	178021.0	177735.0	179528.0	174721.0
top	Hungary	ALB	Labor force participation rate, male (% of mal...	FM.AST.DOMS.CN	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
freq	1599	1599	264	264	1269.0	747.0	759.0	707.0	701.0	1229.0	...	11994.0	11797.0	11545.0	11768.0

4 rows × 64 columns

< >

## Dados da COVID para Alemanha, Rússia, Espanha, Ucrânia e Itália

[18]: df\_alemanha.describe()

	casos_novos	recuperados_acumulados	mortes_acumuladas
count	22598.000000	1.606000e+03	22966.000000
mean	221.663156	1.984489e+05	557.111164
std	1406.483656	3.778855e+05	3206.697148
min	-1091.000000	4.629000e+03	2.000000
25%	9.000000	2.498000e+04	64.000000
50%	21.000000	6.760750e+04	116.000000
75%	50.000000	1.948058e+05	213.000000
max	37422.000000	2.248200e+06	70105.000000

[19]: df\_russia.describe()

	casos_novos	recuperados_acumulados	mortes_acumuladas
count	15184.000000	1.517800e+04	15184.000000
mean	420.176633	4.671057e+04	1038.250000
std	2177.015710	2.498521e+05	5562.758771
min	-6146.000000	1.000000e+00	0.000000
25%	56.000000	6.611500e+03	83.000000
50%	114.000000	1.221100e+04	184.000000
75%	195.000000	2.304375e+04	451.000000
max	29935.000000	3.823074e+06	86455.000000

```
[20]: df_espanha.describe()
```

	casos_novos	recuperados_acumulados	mortes_acumuladas
count	3954.000000	1514.000000	2798.000000
mean	1412.948660	7977.293263	4435.507148
std	5202.988931	11154.736281	11551.701950
min	-22532.000000	125.000000	0.000000
25%	0.000000	1537.000000	0.000000
50%	0.000000	3107.000000	447.000000
75%	576.500000	9204.000000	2522.750000
max	93822.000000	150376.000000	69609.000000

```
[21]: df_italia.describe()
```

	casos_novos	recuperados_acumulados	mortes_acumuladas
count	4140.000000	4.141000e+03	4141.000000
mean	655.277778	5.399390e+04	4226.908235
std	3668.100607	2.606561e+05	13273.713210
min	-3024.000000	3.730000e+02	23.000000
25%	0.000000	1.373000e+03	146.000000
50%	0.000000	4.212000e+03	472.000000
75%	0.000000	1.462600e+04	1568.000000
max	40902.000000	2.416093e+06	97945.000000

```
[22]: df_ucrania.describe()
```

	casos_novos	recuperados_acumulados	mortes_acumuladas
count	2295.000000	2.318000e+03	2318.000000
mean	782.501961	6.837250e+04	1776.847714
std	2052.939424	1.650722e+05	4043.087546
min	2.000000	2.337000e+03	91.000000
25%	100.000000	2.012075e+04	505.000000
50%	203.000000	3.334250e+04	680.500000
75%	424.000000	4.441825e+04	1056.500000
max	16294.000000	1.171724e+06	26212.000000



## REFERÊNCIAS

---

BANCO MUNDIAL. World Development Indicators. Explore country development indicators from around the world. Kaggle. Disponível em: <<https://www.kaggle.com/worldbank/world-development-indicators>>. Acesso em: 05 de mar. de 2021.

BING. Bing-COVID-19 Data. Covid-19 Tracker, 2021. Disponível em: <<https://media.githubusercontent.com/media/microsoft/Bing-COVID-19-Data/master/data/Bing-COVID19-Data.csv>>. Acesso em: 05 de mar. de 2021.

KUMAR, Raman; PAL, Ranabir. India achieves WHO recommended doctor population ratio: A call for paradigm shift in public health discourse!. Journal of family medicine and primary care, v. 7, n. 5, p. 841, 2018.

OMS. Organização Mundial de Saúde. Médicos por 1000 habitantes. Plataforma de Dados de Saúde / GHO / Indicadores, 2021. Disponível em: <[https://www.who.int/data/gho/data/indicators/indicator-details/GHO/medical-doctors-\(per-10-000-population\)](https://www.who.int/data/gho/data/indicators/indicator-details/GHO/medical-doctors-(per-10-000-population))>. Acesso em: 05 de mar. de 2021.