

Processo Seletivo - Base dos Dados

Abrindo Pacote basedosdados e demais

```
In [1]: import basedosdados as bd
```

```
In [2]: import matplotlib.pyplot as plt
```

```
In [3]: import seaborn as sbs
```

```
In [4]: import pandas as pd
```

```
In [5]: import plotly.express as px
```

Avaliando Produtividade Agrícola de Arroz e Feijão

Datalake Censo Agropecuário

A produtividade agrícola pode ser definida como a razão entre **produção total (em toneladas)** e **area total (em hectares)**. Dessa maneira, é um parâmetro capaz de indicar que regiões são mais produtivas que as outras, levando em consideração a produção gerada em relação à área plantada.

Alguns pontos importantes a serem analisados:

- Correlação entre produção total e área total plantada de arroz e feijão;
- Produtividade total de arroz e feijão por estado brasileiro em 2017;
- Produtividade municipal média de arroz e feijão por estado brasileiro em 2017.

A seguir, os comandos serão realizadas com a finalidade de gerar visualização dos resultados em gráficos e, por fim, são realizados comentários dos principais insights.

In [7]: *# Acessar o datalake Censo Agropecuário*

```
df = bd.read_table(dataset_id='br_ibge_censo_agropecuario',  
table_id='municipio', billing_project_id="basedosdados-360312")
```

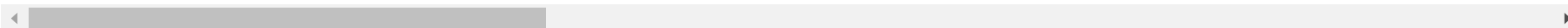
Downloading: 100%|██████████| 20738/20738 [00:26<00:00, 776.89rows/s]

In [9]: *# Conhecer dados*

```
display(df)
```

	ano	sigla_uf	id_municipio	area_total	area_proprietario	area_arrendatario	area_parceiro	area_ocupante	area_lavoura_permanente	area_lavoura_tem
0	1985	RO	1100023	NaN	682505.25	NaN	NaN	NaN	NaN	
1	1985	RO	1100049	NaN	NaN	NaN	NaN	NaN	NaN	
2	1985	RO	1100056	NaN	NaN	NaN	NaN	NaN	NaN	
3	1985	RO	1100064	NaN	NaN	NaN	NaN	NaN	NaN	
4	1985	RO	1100080	NaN	NaN	NaN	NaN	NaN	NaN	
...	
20733	2017	GO	5222005	79967.0	74387.00	5225.0	0.0	0.0	154.0	4
20734	2017	GO	5222054	62341.0	57153.00	2683.0	2213.0	0.0	1265.0	;
20735	2017	GO	5222203	84770.0	80518.00	0.0	0.0	0.0	16.0	
20736	2017	GO	5222302	186118.0	174196.00	5665.0	0.0	6116.0	5337.0	4
20737	2017	DF	5300108	257047.0	140248.00	33640.0	3851.0	72983.0	9304.0	10

20738 rows × 116 columns



In [10]: *# Criando tabela de dados*

```
dados = df[["ano", "sigla_uf", "id_municipio", "producao_total_arroz" , "area_arroz", "producao_total_feijao" , "area_feijao"]]
```

In [11]: *# Deletando linhas vazias*

```
dados = dados.dropna()
```

In [12]: *# Filtrando dados de 2017*

```
dados_2017 = dados.loc[(dados['ano'] == 2017)]
```

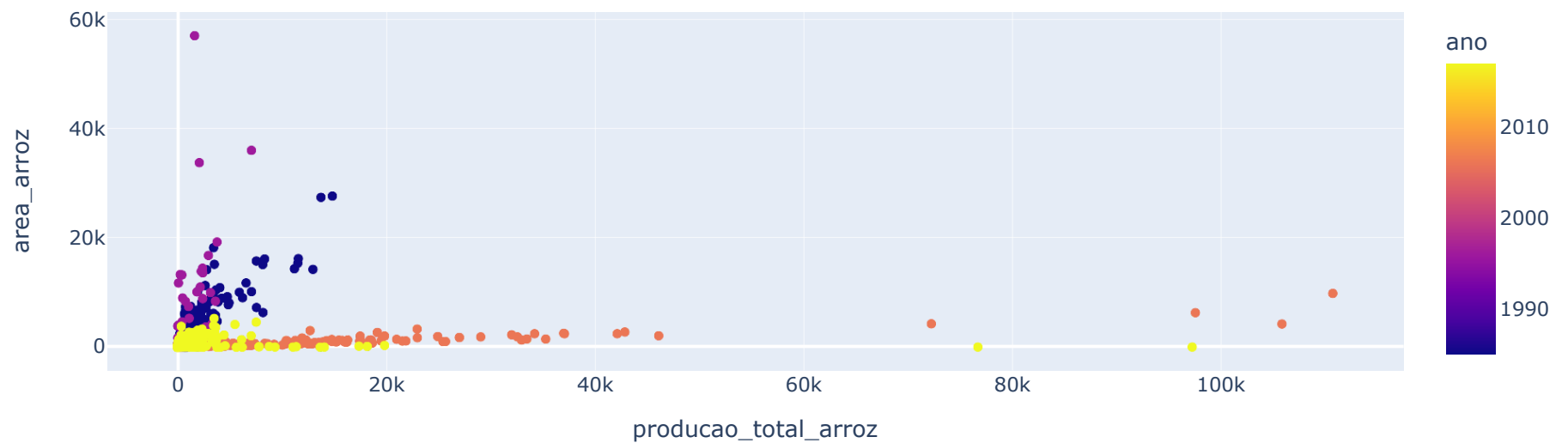
```
In [13]: #Convertendo ano em string  
dados_2017 = dados_2017.replace([2017], "2017")
```

PRODUTIVIDADE DE ARROZ

```
In [14]: # Criando a variavel de produtividade  
dados_2017["produtividade_arroz"] = (dados_2017["producao_total_arroz"] / dados_2017["area_arroz"])
```

```
In [18]: #Gráfico de dispersão
Graf_dis_arroz = px.scatter(dados,
                           x="producao_total_arroz",
                           y="area_arroz",
                           color="ano",
                           height= 400,
                           title= "GRÁFICO DE DISPERSÃO PRODUÇÃO TOTAL X ÁREA PLANTADA - ARROZ")
Graf_dis_arroz.show()
```

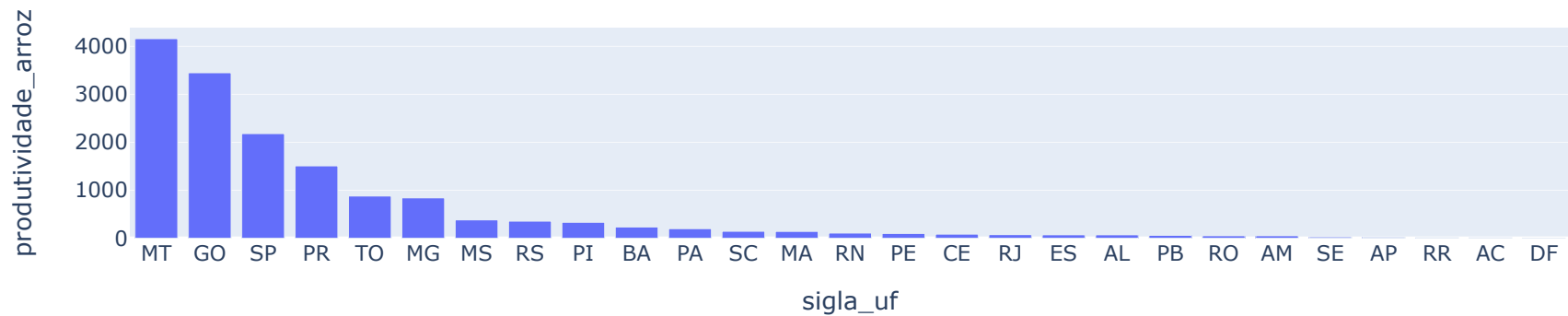
GRÁFICO DE DISPERSÃO PRODUÇÃO TOTAL X ÁREA PLANTADA - ARROZ



```
In [20]: #PRODUTIVIDADE TOTAL DE ARROZ POR ESTADO
#É o somatório de produtividade dos municípios de cada estado
produtividade_total_arroz= dados_2017.groupby('sigla_uf').sum()
produtividade_total_arroz= produtividade_total_arroz[['produtividade_arroz']].sort_values(by= 'produtividade_arroz', ascending=

#GRÁFICO DE BARRAS
graf_total_arroz = px.bar(produtividade_total_arroz,
                           x = produtividade_total_arroz.index,
                           y = "produtividade_arroz",
                           height= 300,
                           title = "PRODUTIVIDADE TOTAL DE ARROZ POR ESTADO (2017)")
graf_total_arroz.show()
```

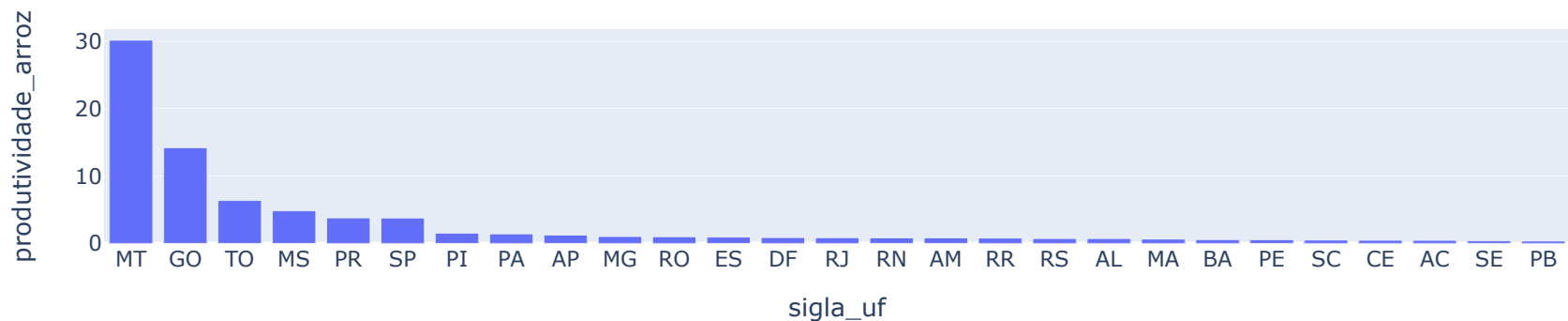
PRODUTIVIDADE TOTAL DE ARROZ POR ESTADO (2017)



```
In [21]: #PRODUTIVIDADE MÉDIA MUNICIPAL DE ARROZ POR ESTADO
produtividade_media_arroz= dados_2017.groupby('sigla_uf').mean()
produtividade_media_arroz= produtividade_media_arroz[['produtividade_arroz']].sort_values(by= 'produtividade_arroz', ascending=

#GRÁFICO DE BARRAS
graf_media_arroz = px.bar(produtividade_media_arroz,
                           x = produtividade_media_arroz.index,
                           y = "produtividade_arroz",
                           height= 300,
                           title = "PRODUTIVIDADE MÉDIA MUNICIPAL DE ARROZ POR ESTADO (2017)")
graf_media_arroz.show()
```

PRODUTIVIDADE MÉDIA MUNICIPAL DE ARROZ POR ESTADO (2017)



Insights

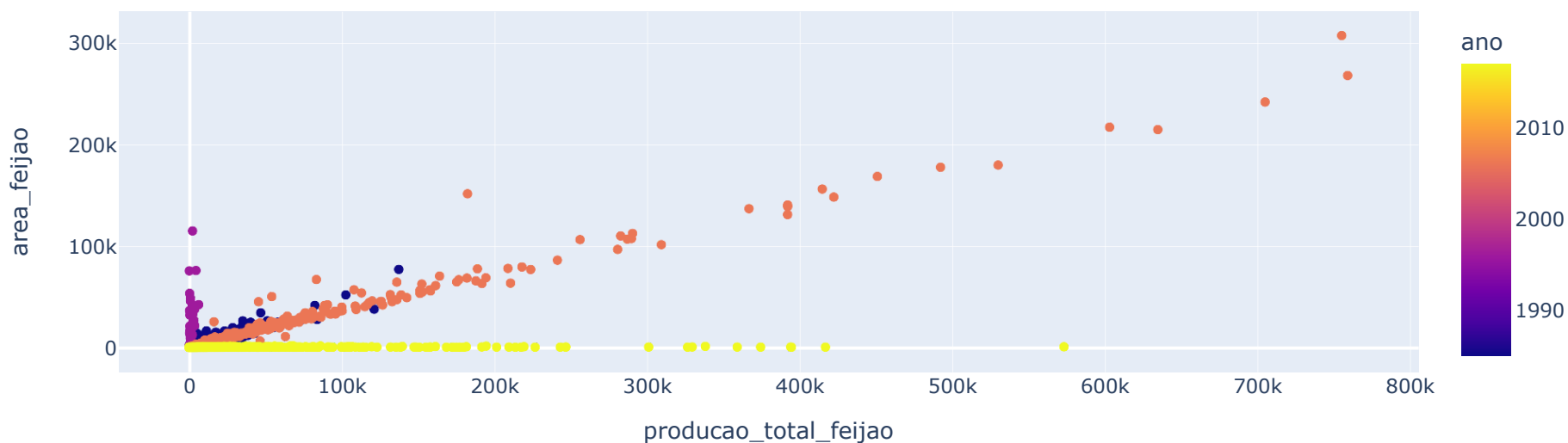
- O gráfico de dispersão indica forte correlação positiva entre as variáveis *produção total de arroz* e *área total destinada ao plantio de arroz*, mostrando que as duas variáveis tendem a diminuir ou aumentar juntas;
- O estado com maior produtividade total de arroz foi o Mato Grosso (MT), com produção média de 4159,41 T/Ha ;
- O estado com menor produtividade total de arroz foi o Distrito Federal (DF), com produção média de 0,87 T/Ha ;
- O estado com maior produtividade média municipal de arroz foi o Mato Grosso (MT), com produção de 30,14 T/Ha ;
- O estado com menor produtividade média municipal de arroz foi o Paraíba (PB), com produção de 0,36 T/Ha .

PRODUTIVIDADE DE FEIJÃO

```
In [22]: # Criando a variavel de produtividade
dados_2017["produtividade_feijao"] = (dados_2017["producao_total_feijao"] / dados_2017["area_feijao"])
```

```
In [23]: #Gráfico de dispersão
Graf_dis_feijao = px.scatter(dados,
                             x="producao_total_feijao",
                             y="area_feijao",
                             color="ano",
                             height= 400,
                             title= "GRÁFICO DE DISPERSÃO PRODUÇÃO TOTAL X ÁREA PLANTADA - FEIJÃO")
Graf_dis_feijao.show()
```

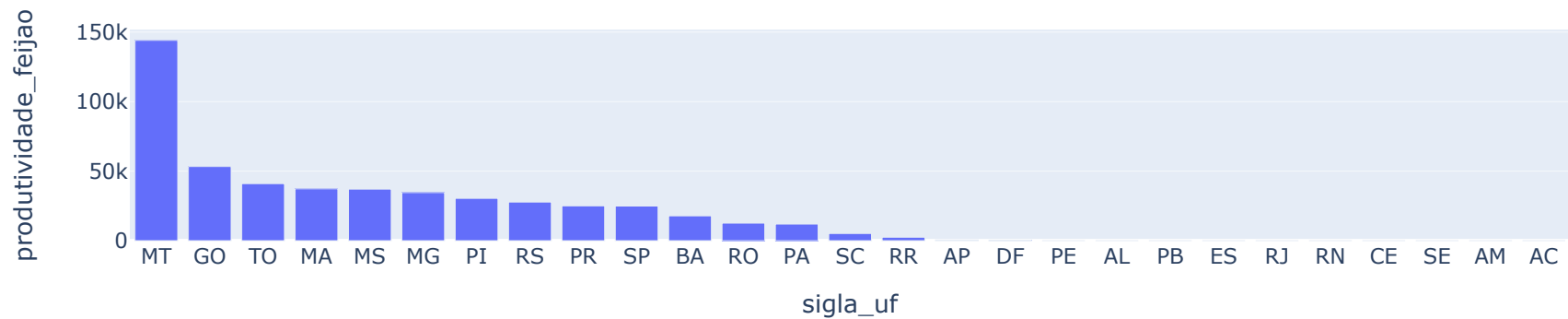
GRÁFICO DE DISPERSÃO PRODUÇÃO TOTAL X ÁREA PLANTADA - FEIJÃO



```
In [25]: #PRODUTIVIDADE TOTAL DE FEIJÃO POR ESTADO
produtividade_total_feijao= dados_2017.groupby('sigla_uf').sum()
produtividade_total_feijao= produtividade_total_feijao[['produtividade_feijao']].sort_values(by= 'produtividade_feijao', ascend:

#GRÁFICO DE BARRAS
graf_total_feijao = px.bar(produtividade_total_feijao,
                           x = produtividade_total_feijao.index,
                           y = "produtividade_feijao",
                           height= 300,
                           title = "PRODUTIVIDADE TOTAL DE FEIJÃO POR ESTADO (2017)")
graf_total_feijao.show()
```

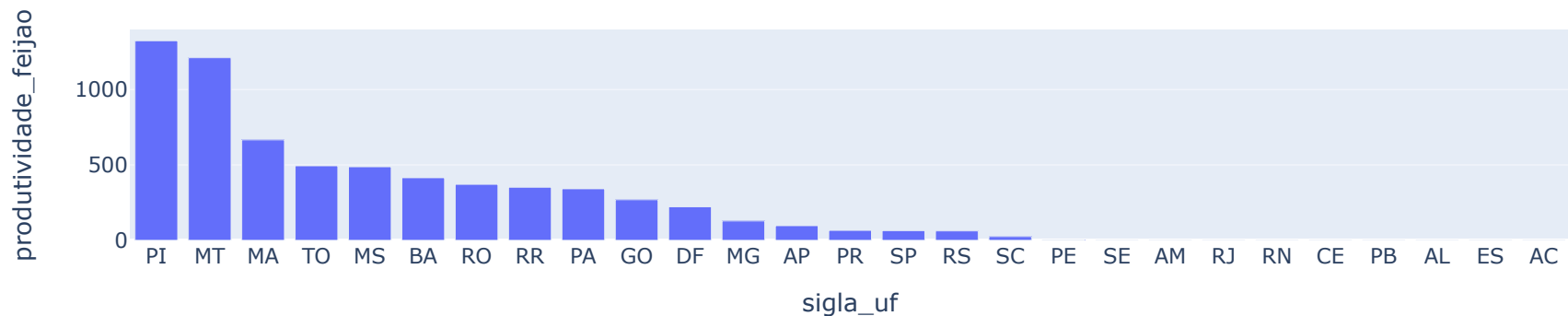
PRODUTIVIDADE TOTAL DE FEIJÃO POR ESTADO (2017)




```
In [31]: #PRODUTIVIDADE MÉDIA MUNICIPAL DE FEIJÃO POR ESTADO
produtividade_media_feijao= dados_2017.groupby('sigla_uf').mean()
produtividade_media_feijao= produtividade_media_feijao[['produtividade_feijao']].sort_values(by= 'produtividade_feijao', ascend:

#GRÁFICO DE BARRAS
graf_media_feijao = px.bar(produtividade_media_feijao,
                           x = produtividade_media_feijao.index,
                           y = "produtividade_feijao",
                           height= 300,
                           title = "PRODUTIVIDADE MÉDIA MUNICIPAL DE FEIJÃO POR ESTADO (2017)")
graf_media_feijao.show()
```

PRODUTIVIDADE MÉDIA MUNICIPAL DE FEIJÃO POR ESTADO (2017)



Insights

- O gráfico de dispersão indica forte correlação positiva entre as variáveis *produção total de feijão* e *área total destinada ao plantio de feijão*, mostrando que as duas variáveis tendem a diminuir ou aumentar juntas;
- O estado com maior produtividade total de feijão foi o Mato Grosso (MT), com produção média de 144.090,82 T/Ha ;
- O estado com menor produtividade total de feijão não nula foi o Pernambuco (PE), com produção média de 1.33 T/Ha ;
- O estado com maior produtividade média municipal de feijão foi o Piauí (PI), com produção de 1323,22 T/Ha ;
- O estado com menor produtividade média municipal de feijão foi o Pernambuco (PE), com produção de 0,16 T/Ha .

Avaliando a distribuição de bolsas CAPES em 2019

Datalake Bolsas CAPES

```
In [33]: # Acessar o datalake Bolsas Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)
df2 = bd.read_table(dataset_id='br_capes_bolsas',
                    table_id='mobilidade_internacional',
                    billing_project_id="basedosdados-360312")
```

Downloading: 100%|██████████| 146036/146036 [01:08<00:00, 2117.14rows/s]

```
In [34]: # Conhecer base de dados
display(df2)
```

...
146031	2019	2	2019	8	88881.172162/2018-01	***.732.087- **	THIAGO DA SILVEIRA ALVARES	PVEX - PROGRAMA DE PROFESSOR VISITANTE NO EXTE...	ESTADOS UNIDOS	USD	...	
146032	2019	2	2020	1	88887.289030/2018-00	***.800.368- **	THIAGO DE MORAIS MARIANO	CAPES/COFECUB (COMITÉ FRANCÊS DE AVALIAÇÃO DA ...	FRANÇA	EUR	...	
146033	2019	2	2023	1	88881.284303/2018-01	***.617.361- **	TITOS MOAMBA	PROGRAMA ESTUDANTES CONVÊNIO DE PÓS- GRADUAÇÃO	BRASIL	BRL	...	
146034	2019	2	2020	1	88881.162278/2017-01	***.386.226- **	UELINTON MANOEL PINTO	PROGRAMA CAPES/HARVARD	ESTADOS UNIDOS	USD	...	
				VITOR								

```
In [35]: # Criando tabela de dados
dados_capes = df2[["ano_inicial", "programa_capes", "pais_destino", "sigla_moeda", "valor_recebido_total"]]
```

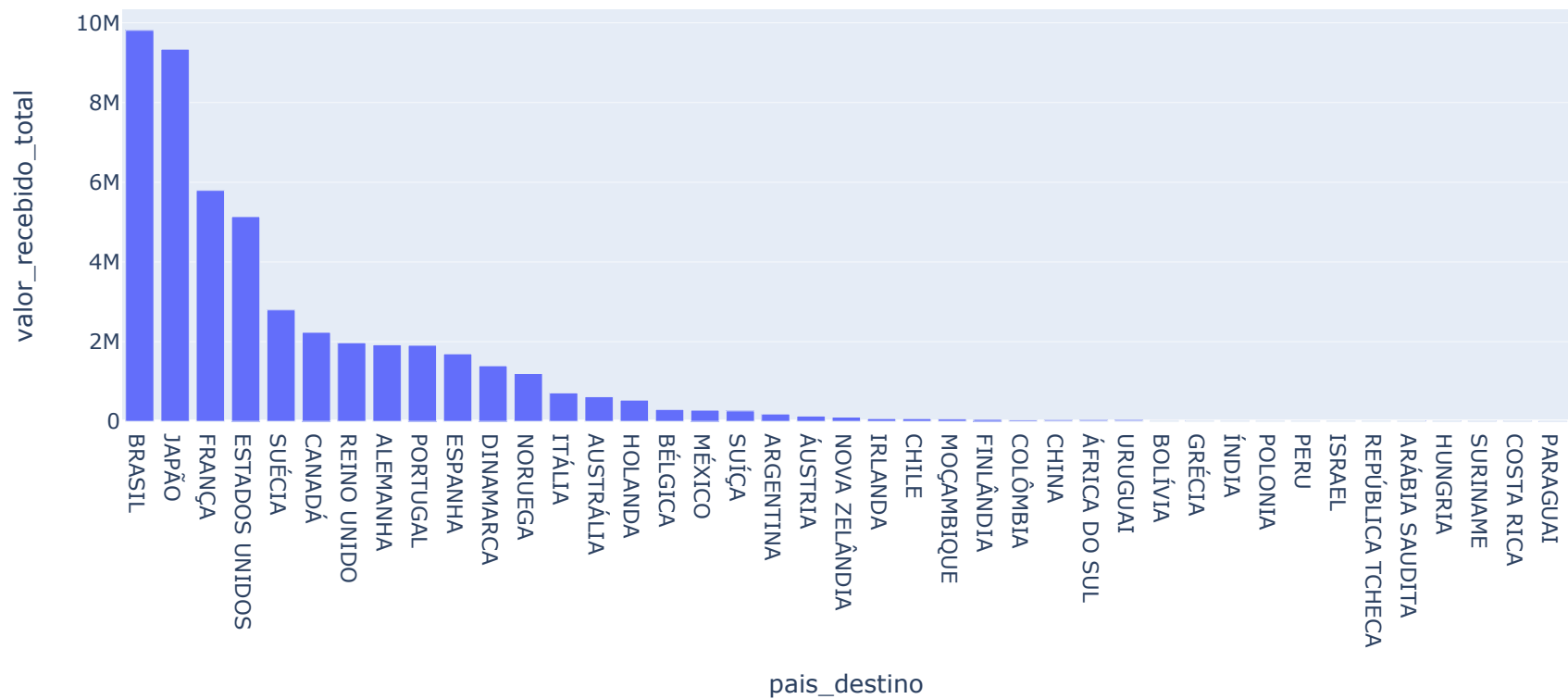
```
In [36]: # Deletando linhas vazias
dados_capes = dados_capes.dropna()
```

```
In [37]: # Filtrando dados de 2019
dados_2019 = dados_capes.loc[(dados_capes['ano_inicial'] == 2019)]
```

```
In [39]: #Agrupando por países
tabela_países= dados_2019.groupby(['pais_destino']).sum()
tabela_países= tabela_países.sort_values(by= 'valor_recebido_total', ascending= False)
#tabela_países = tabela_países[['valor_recebido_total']].sort_values(by= 'valor_recebido_total', ascending= False)

fig = px.bar(tabela_países,
             x= tabela_países.index,
             y="valor_recebido_total",
             height= 500,
             title = "Países destinos de bolsas CAPES por valor recebido total (2019) ")
fig.show()
```

Países destinos de bolsas CAPES por valor recebido total (2019)

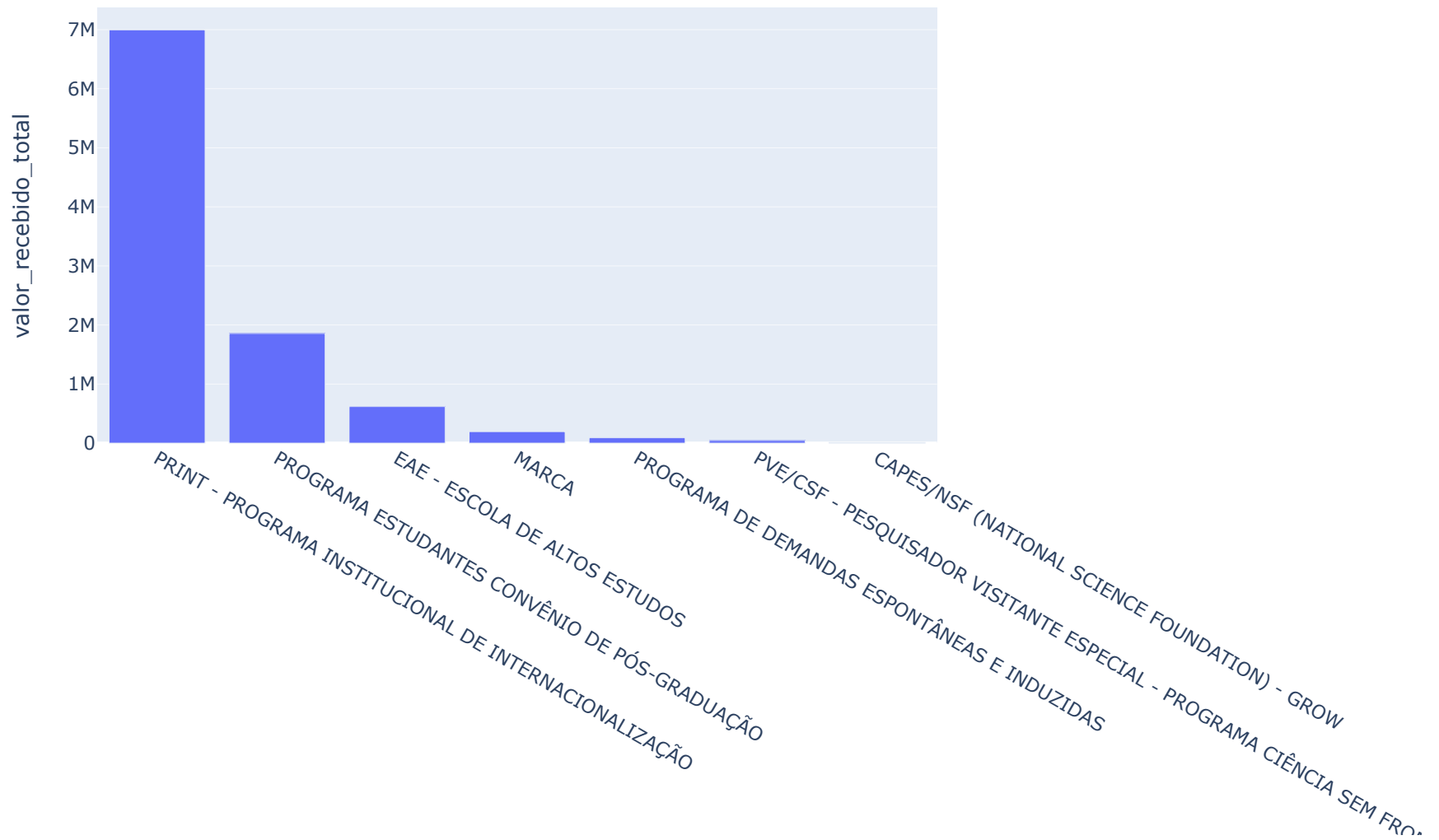


```
In [40]: # Fazendo query para Brasil
dados_brasil = dados_2019.query("pais_destino == 'BRASIL'")
```

```
In [41]: #Agrupando por tipo de programa
tabela_programas = dados_brasil.groupby(['programa_capes']).sum()
tabela_programas = tabela_programas.sort_values(by= 'valor_recebido_total', ascending= False)
```

```
In [44]: fig = px.bar(tabela_programas,
                    x= tabela_programas.index,
                    y='valor_recebido_total',
                    height=700,
                    title = "Distribuição do valor recebido total por tipo de programa CAPES (2019) ")
fig.show()
```

Distribuição do valor recebido total por tipo de programa CAPES (2019)



programa_capes

Insights

- Depois do Brasil, o Japão foi país que mais recebeu bolsas CAPES, em seguida, França e Estados Unidos em 2019;
 - O programa que a CAPES mais destinou recursos em 2019 foi o Programa Institucional de Internacionalização (PRINT), o que justifica o insight anterior.
-