



SOUL CODE



PROJETO TELECOM

Artur Venturoli • Maurício da Silva Barros
Renan Marques • Renato Romão

Integrantes

Contate cada um dos integrantes deste projeto:

Artur Venturoli

<https://br.linkedin.com/in/arturventuroli>

Maurício da Silva Barros

<https://www.linkedin.com/in/mauriciodasilvabarros/>

Renan Marques

www.linkedin.com/in/renan-marques-rodrigues

Renato Romão

www.linkedin.com/in/renato-cm-romao

Insights

Estes foram os insights trabalhados em nosso projeto:

- 1) Por que a Oi não oferece tecnologia 5G?**
- 2) Comparativo entre telefonia Fixa e Móvel.**
- 3) Qual o domínio dos tipos de serviço por operadora?**
- 4) Qual o crescimento de acesso nas pequenas prestadoras de serviço banda larga?**

Procedimentos

1. Instalação

```
In [ ]: pip install pymysql # Necessário para fazer a conexão / envio com o MySQL
```

```
In [ ]: pip install mysql-connector-python # Necessário para fazer a conexão com o MySQL
```

```
In [ ]: pip install pymongo[srv] # Necessário para fazer a conexão com o MongoDB
```

```
In [ ]: pip install xlrd # Necessário fazer instalações para abertura de arquivo xlsx
```

```
In [ ]: pip install openpyxl # Necessário fazer instalações para abertura de arquivo xlsx
```

2. Importação de Bibliotecas

```
In [ ]:
```

```
from pyspark.sql import SparkSession
import pyspark.sql.functions as F
from pyspark.sql.types import *
import pandas as pd
from google.cloud import storage
import pymongo
from pymongo import MongoClient
import pymysql
import mysql.connector
from sqlalchemy import create_engine # é responsável por realizar a conexão entre o
```

3. Iniciando Sessão Spark

```
In [ ]:
```

```
spark = (SparkSession.builder
    .master("local")
    .appName("ProjetoFinal")
    .config('spark.ui.port', '4050')
    .config("spark.jars", 'https://storage.googleapis.com/hadoop-lib/gcs/gcs-con
    .getOrCreate())
```

4. Importação dos DataSets

4.1. Criação DataFrame | Pandas

4.1.1. Criação do DataFrame “ERBs - antenas” utilizando arquivo “.xlsx”

```
In [ ]:
```

```
dfa = pd.read_excel('gs://projetofinal/Originais/Antenas/ERBs-Fev22.xlsx', sheet_nam
dfa.head()
```

Out[]:

	NumEstacao	NomeEntidade	EnderecoEstacao	SiglaUf	CodMunicipio	NomeMunicipio	Latitu
0	1000959098	CLARO	LINHA 148 S/N DISTRITO DE SANTO ANTONIO	RO	1100015	Alta Floresta D'Oeste	12S0424
1	689454236	OI	AV. SÃO PAULO S/NºCENTRO	RO	1100015	Alta Floresta D'Oeste	11S5537
2	684862131	TIM	AV. AMAZONAS C/ RUA CEARA SNCENTRO	RO	1100015	Alta Floresta D'Oeste	11S5543
3	696172968	CLARO	Avenida Nilo PeçanhaS/NTucano	RO	1100015	Alta Floresta D'Oeste	11S5558
4	442752075	VIVO	AVENIDA SANTA CATARINA 1000 CENTRO	RO	1100015	Alta Floresta D'Oeste	11S5537

4.1.2. Criação do DataFrame “Regiões” utilizando arquivo “.xlsx”

In []:

```
dfr = pd.read_excel('gs://projetofinal/Originais/Regioes/Estados_Regioes.xlsx')
dfr.head()

# Criação do DF de Regiões
```

Out[]:

	Capitais	Estados	Siglas dos Estados	Regiões
0	Rio Branco	Acre	AC	Norte
1	Maceió	Alagoas	AL	Nordeste
2	Macapá	Amapá	AP	Norte
3	Manaus	Amazonas	AM	Norte
4	Salvador	Bahia	BA	Nordeste

4.2. Criação DataFrame | PySpark

4.2.1. Análise dos DataSets contidos na Bucket para composição dos DataFrames “Banda Larga Fixa”, “Telefonia Móvel”, “Telefonia Fixa”

Conectando na bucket e criando uma lista com o nome dos arquivos para composição

In []:

```
bucket = storage.Client().get_bucket('projetofinal')

bl_names = [file.name for file in bucket.list_blobs(prefix='Originais/Banda Larga/A')]
movel_names = [file.name for file in bucket.list_blobs(prefix='Originais/Movel/A')]
fixa_names = [file.name for file in bucket.list_blobs(prefix='Originais/Telefonia Fi

print('Arquivos de Banda Larga Fixa:')
for i in bl_names:
```

```

print(i.split('/')[ -1])

print('\nArquivos de Telefonia Movel:')
for i in movei_names:
    print(i.split('/')[ -1])

print('\nArquivos de Telefonia Fixa:')
for i in fixa_names:
    print(i.split('/')[ -1])

```

4.2.2. Criação DataFrame "Banda Larga Fixa"

```
In [ ]: dfbl = spark.read.csv('gs://projetofinal/Originais/Banda_Larga/Acessos_Banda_Larga_F
dfbl2 = spark.read.csv('gs://projetofinal/Originais/Banda_Larga/Acessos_Banda_Larga_
```

4.2.3. Criação DataFrame "Telefonia Móvel"

Dropando colunas indesejáveis antes de concatenar para não onerar o processo

Concatenando todos os Datasets do diretório do Datalake, resultando em DataFrame único

```
In [ ]: if movei_names == []:
    print('Não encontrado nenhum arquivo para ser inserido no DataFrame.')
else:
    print('Iniciando inserção de DataSets...')
    dfm = spark.read.csv('gs://projetofinal/' + movei_names[0], sep=';', header=True)
    dfm = dfm.drop('Código IBGE Município', 'CNPJ', 'Código Nacional (Chip)', 'Porte')
    print(f"DataSet '{movei_names[0][16:]}' incluído com sucesso.")
    if len(movei_names) > 1:
        movei = movei_names[1:]
        for i in movei:
            dfnovo = spark.read.csv('gs://projetofinal/' + i, sep=';', header=True)
            dfnovo = dfnovo.drop('Código IBGE Município', 'CNPJ', 'Código Nacional (')
            dfm = dfnovo.union(dfm)
            print(f"DataSet '{i[16:]}' incluído com sucesso.")
```

Iniciando inserção de arquivos ao DataFrame...

DataFrame 'Acessos_Telefonia_Movel_201901-201906.csv' incluído com sucesso.

DataFrame 'Acessos_Telefonia_Movel_201907-201912.csv' incluído com sucesso.

DataFrame 'Acessos_Telefonia_Movel_202001-202006.csv' incluído com sucesso.

DataFrame 'Acessos_Telefonia_Movel_202007-202012.csv' incluído com sucesso.

DataFrame 'Acessos_Telefonia_Movel_202101-202106.csv' incluído com sucesso.

[Stage 27:===== (24 + 1) / 25] DataFrame 'Acessos_Telefonia_Movel_202107-202112.csv' incluído com sucesso.

4.2.4. Criação DataFrame "Telefonia Fixa"

```
In [ ]: dff = spark.read.csv('gs://projetofinal/Originais/Telefonia_Fixa/Acessos_Telefonia_F
```

```
dff2 = spark.read.csv('gs://projetofinal/Originais/Telefonia Fixa/Acessos_Telefonia_
```

5. Inserindo os dados originais no CloudStorage / Importando na instância do MySQL

O procedimento foi realizado por incompatibilidade da tipagem do Dataset original no Banco de Dados MySQL

5.1. Enviando DataSet "Banda Larga Fixa"

```
In [ ]: dfbl.repartition(1).write.format("csv").option("header", "false").save("gs://projeto  
print('DataSet \"Acessos_Banda_Larga_Fixa_2019-2020.csv\" enviado com sucesso.')  
  
dfbl2.repartition(1).write.format("csv").option("header", "false").save("gs://projeto  
print('DataSet \"Acessos_Banda_Larga_Fixa_2021.csv\" enviado com sucesso.')
```

```
DataFrame "Acessos_Banda_Larga_Fixa_2019-2020.csv" enviado com sucesso.  
[Stage 3:=====] (5 + 1) / 6  
DataFrame "Acessos_Banda_Larga_Fixa_2021.csv" enviado com sucesso.
```

5.2. Enviando DataSet "Telefonia Móvel"

```
In [ ]: for i in movei_names:  
    dfmysql = spark.read.csv('gs://projetofinal/' + i, sep=';', header=True, inferSc  
    dfmysql.repartition(1).write.format("csv").option("header", "false").save("gs://  
    print(f"DataFrame '{i[16:]}' enviado com sucesso.")
```

```
DataFrame 'Acessos_Telefonia_Movel_201901-201906.csv' enviado com sucesso.  
DataFrame 'Acessos_Telefonia_Movel_201907-201912.csv' enviado com sucesso.  
DataFrame 'Acessos_Telefonia_Movel_202001-202006.csv' enviado com sucesso.  
DataFrame 'Acessos_Telefonia_Movel_202007-202012.csv' enviado com sucesso.  
DataFrame 'Acessos_Telefonia_Movel_202101-202106.csv' enviado com sucesso.  
DataFrame 'Acessos_Telefonia_Movel_202107-202112.csv' enviado com sucesso.
```

5.3. Enviando DataSet "Telefonia Fixa"

```
In [ ]: dff.repartition(1).write.format("csv").option("header", "false").save("gs://projetoof  
print('DataFrame \"Acessos_Telefonia_Fixa_Concessionarias.csv\" enviado com sucesso.')  
  
dff2.repartition(1).write.format("csv").option("header", "false").save("gs://projeto  
print('DataFrame \"Acessos_Telefonia_Fixa_Autorizadas.csv\" enviado com sucesso.')
```

DataFrame "Acessos_Telefonia_Fixa_Concessionarias.csv" enviado com sucesso.

DataFrame "Acessos_Telefonia_Fixa_Autorizadas.csv" enviado com sucesso.

6. Envio do DataSet 'ERBs - Antenas' original para MySQL | Via Pandas

6.1. Criando conexão com MySQL

```
In [ ]: con = mysql.connector.connect(host='34.73.255.107', user='root', password='123456', db=cur = con.cursor()

engine = create_engine("mysql+pymysql://root:123456@34.73.255.107/telecom")
```

6.2. Convertendo DataFrame para linguagem SQL e salvando na tabela ERBs_Antenas no MySQL

```
In [ ]: dfsql = dfa.to_sql('ERBs_Antenas', con = engine, if_exists='replace', index = False)
```

6.3. Verificação da quantidade de linhas dos Dados enviados para MySQL

```
In [ ]: cur.execute('SELECT count(*) FROM ERBs_Antenas')
cur.fetchall()
```

Out[]: [(96567,)]

7. Criação de Backup dos DataFrames

```
In [ ]: dfabkp = dfa
print('Backup DF "ERBs - Antenas" criado com sucesso.')

dfblbkp = dfbl
dfbl2bkp = dfbl2
print('Backup DF "Banda Larga Fixa" criado com sucesso.')

dfmbkp = dfm
print('Backup DF "Telefonia Móvel" criado com sucesso.')

dffbkp = dff
dff2bkp = dff2
print('Backup de "Telefonia Fixa" criado com sucesso.)
```

8. Tratamentos

8.1. Com PySpark

8.1.1. DataFrame "Banda Larga"

8.1.1.1. Dropando colunas e concatenando DataFrames de Banda Larga para unificar informações

```
In [ ]: dfbl = dfbl.drop('Grupo Econômico', 'CNPJ', 'Código IBGE Município')
dfbl2 = dfbl2.drop('Grupo Econômico', 'CNPJ', 'Código IBGE Município', 'Tipo de Pess
dfbl = dfbl.union(dfbl2)
```

8.1.1.2. Dropando linhas duplicadas

```
In [ ]: dfbl = dfbl.dropDuplicates()
```

8.1.1.3. Renomeando colunas

Retirar acentos, caracteres especiais e espaços

```
In [ ]: dfbl = dfbl.withColumnRenamed('Mês', 'Mes').withColumnRenamed('Grupo Econômico', 'Gr
.withColumnRenamed('Porte da Prestadora', 'Porte_da_Prestadora').withColumnRenamed('
.withColumnRenamed('Faixa de Velocidade', 'Faixa_de_Velocidade').withColumnRenamed('
```

8.1.1.4. Verificando Schema do DataFrame

```
In [ ]: dfbl.printSchema()
```

```
root
|-- Ano: integer (nullable = true)
|-- Mes: integer (nullable = true)
|-- Empresa: string (nullable = true)
|-- Porte_da_Prestadora: string (nullable = true)
|-- UF: string (nullable = true)
|-- Municipio: string (nullable = true)
|-- Faixa_de_Velocidade: string (nullable = true)
|-- Tecnologia: string (nullable = true)
|-- Meio_de_Acesso: string (nullable = true)
|-- Acessos: integer (nullable = true)
```

8.1.1.5. Verificando valores nulos

```
In [ ]: for i in dfbl.columns:
    print(i, dfbl.filter(F.col(i).isNull()).count())
```

Ano 0

Mes 0

Empresa 0

Porte_da_Prestadora 0

UF 0

Municipio 0

```
Faixa_de_Velocidade 0
```

```
Tecnologia 0
```

```
Meio_de_Acesso 0
```

```
[Stage 126:===== (9 + 1) / 10]  
Acessos 0
```

8.1.2. DataFrame "Telefonia Móvel"

8.1.2.1. Renomeando colunas

Retirar acentos, caracteres especiais e espaços

```
In [ ]:  
dfm = dfm.withColumnRenamed('Mês', 'Mes').withColumnRenamed('Município', 'Municipio')  
.withColumnRenamed('Modalidade de Cobrança', 'Modalidade_de_Cobrança').withColumnRen  
.withColumnRenamed('Tipo de Pessoa', 'Tipo_de_Pessoa').withColumnRenamed('Tipo de Pr
```

8.1.2.2. Atualizado Dataframe por meio de filtro contendo valores de 'DADOS', 'VOZ', 'VOZ+DADOS'

```
In [ ]:  
dfm = dfm.filter(F.col('Tipo_de_Produto').isin('DADOS', 'VOZ', 'VOZ+DADOS'))
```

8.1.2.3. Dropando valores duplicados

```
In [ ]:  
dfm = dfm.dropDuplicates()
```

8.1.3. DataFrame "Telefonia Fixa"

8.1.3.1. Filtrando DataFrame por ano, para obter apenas os anos de 2019, 2020 e 2021

```
In [ ]:  
dff = dff.filter(F.col('Ano').isin(2019, 2020, 2021))  
dff2 = dff2.filter(F.col('Ano').isin(2019, 2020, 2021))
```

8.1.3.2. Filtrando o DataFrame para ter a coluna "Tipo do Acesso" igual a "Individual em Serviço"

Procedimento para alinhar os valores

```
In [ ]:  
dff = dff.filter(F.col('Tipo do Acesso') == 'Individual em Serviço')
```

8.1.3.3. Dropando colunas desnecessárias / indesejáveis

```
In [ ]:  
dff = dff.drop('Tipo do Acesso', 'Grupo Econômico', 'CNPJ', 'Código IBGE Município')  
dff2 = dff2.drop('Grupo Econômico', 'CNPJ', 'Código IBGE Município')
```

8.1.3.4. Concatenando Dataframes para unificação de dados

```
In [ ]:  
dff = dff.union(dff2)
```

8.1.3.5. Conferência de dados nulos no DataFrame

- Foi constatado um grande número de valores nulos, porém o restante dos valores são relevantes para o tipo de análise que será realizada.

```
In [ ]: for i in dff.columns:  
    print(i, dff.filter(F.col(i).isNull()).count())
```

Ano 0

Mês 0

Empresa 0

Porte da Prestadora 0

UF 0

Município 0

Tipo de Outorga 0

Tipo de Pessoa 181593

Tipo de Atendimento 181593

```
[Stage 159:=====] (4 + 1) / 5  
Acessos 0
```

8.1.3.6. Renomeando colunas

```
In [ ]: dff.show(5)
```

```
+---+---+---+---+---+  
| Ano|Mês|Empresa|Porte da Prestadora|UF|  
| de Pessoa|Tipo de Atendimento|Acessos|  
+---+---+---+---+---+  
| 2021| 12| VIVO| Grande Porte| SP|  
| oa Física| | URBANO| 1| Juquitiba| Concessão| Pess  
| 2021| 12| VIVO| Grande Porte| SP| Lençóis Paulista| Concessão| Pess  
| oa Física| | URBANO| 1292|  
| 2021| 12| VIVO| Grande Porte| SP| Santa Bárbara d'O...| Concessão| Pess  
| oa Física| | URBANO| 2485|  
| 2021| 12| VIVO| Grande Porte| SP| São Roque| Concessão| Pess  
| oa Física| | URBANO| 936|  
| 2021| 12| VIVO| Grande Porte| SP| Avaré| Concessão| Pess  
| oa Física| | URBANO| 1904|  
+---+---+---+---+  
only showing top 5 rows
```

```
In [ ]: dff = dff.withColumnRenamed('Mês', 'Mes').withColumnRenamed('Porte da Prestadora', '  
.withColumnRenamed('Município', 'Municipio').withColumnRenamed('Tipo de Outorga', 'T  
.withColumnRenamed('Tipo de Pessoa', 'Tipo_de_Pessoa').withColumnRenamed('Tipo de At
```

8.2. Com Pandas

8.2.1. DataFrame ERBs - Antenas

In []:

```
dfa.head()
```

Out[]:

	NumEstacao	NomeEntidade	EnderecoEstacao	SiglaUf	CodMunicipio	NomeMunicipio	Latitu
0	1000959098	CLARO	LINHA 148 S/N DISTRITO DE SANTO ANTONIO	RO	1100015	Alta Floresta D'Oeste	12S0424
1	689454236	OI	AV. SÃO PAULO S/NºCENTRO	RO	1100015	Alta Floresta D'Oeste	11S5537
2	684862131	TIM	AV. AMAZONAS C/ RUA CEARA SNCENTRO	RO	1100015	Alta Floresta D'Oeste	11S5543
3	696172968	CLARO	Avenida Nilo PeçanhaS/NTucano	RO	1100015	Alta Floresta D'Oeste	11S5558
4	442752075	VIVO	AVENIDA SANTA CATARINA 1000 CENTRO	RO	1100015	Alta Floresta D'Oeste	11S5537

8.2.1.1. Dropando colunas

In []:

```
dfa.drop(['NumEstacao', 'EnderecoEstacao', 'CodMunicipio', 'Latitude', 'Longitude'],  
dfa.head(10))
```

Out[]:

	NomeEntidade	SiglaUf	NomeMunicipio	2G	3G	4G	5G
0	CLARO	RO	Alta Floresta D'Oeste	SIM	SIM	SIM	NaN
1	OI	RO	Alta Floresta D'Oeste	SIM	NaN	NaN	NaN
2	TIM	RO	Alta Floresta D'Oeste	SIM	NaN	NaN	NaN
3	CLARO	RO	Alta Floresta D'Oeste	SIM	SIM	SIM	NaN
4	VIVO	RO	Alta Floresta D'Oeste	SIM	SIM	SIM	NaN
5	CLARO	RO	Alta Floresta D'Oeste	SIM	SIM	SIM	NaN
6	CLARO	RO	Ariquemes	NaN	SIM	NaN	NaN
7	CLARO	RO	Ariquemes	NaN	SIM	NaN	NaN
8	OI	RO	Ariquemes	SIM	SIM	NaN	NaN
9	OI	RO	Ariquemes	SIM	NaN	NaN	NaN

8.2.1.2. Renomeando colunas

In []:

```
dfa.rename(columns={'NomeEntidade': 'Empresas', 'SiglaUF': 'UF', 'NomeMunicipio': 'Muni'})
```

8.2.1.3. Contando quantidade de valores nulos do DataFrame

In []:

```
dfa.isna().sum()
```

```
Out[ ]: Empresas      77
UF          0
Municipio    0
2G        37538
3G        21976
4G        15322
5G        94864
dtype: int64
```

8.2.1.4. Dropando linhas que o nome da empresa seja valor nulo

Matematicamente foi constatado que as 77 valores nulos na coluna 'empresas' representavam 0,05% e que a exclusão das mesmas não prejudicaria, pelo contrário, possibilitaria a precisão nas análises realizadas posteriormente.

```
In [ ]: dfa.dropna(subset=['Empresas'], axis=0, inplace=True)
```

8.2.1.5. Substituindo valores nulos

Foi identificado que o valor nulo significava "NÃO"

```
In [ ]: dfa = dfa.fillna('NÃO')
dfa.head()
```

```
Out[ ]:   Empresas  UF       Municipio  2G  3G  4G  5G
0     CLARO  RO  Alta Floresta D'Oeste  SIM  SIM  SIM  NÃO
1       OI  RO  Alta Floresta D'Oeste  SIM  NÃO  NÃO  NÃO
2       TIM  RO  Alta Floresta D'Oeste  SIM  NÃO  NÃO  NÃO
3     CLARO  RO  Alta Floresta D'Oeste  SIM  SIM  SIM  NÃO
4      VIVO  RO  Alta Floresta D'Oeste  SIM  SIM  SIM  NÃO
```

8.2.2. DataFrame Regiões

```
In [ ]: dfr.head(5)
```

```
Out[ ]:   Capitais  Estados  Siglas dos Estados  Regiões
0  Rio Branco        Acre            AC      Norte
1   Maceió        Alagoas           AL  Nordeste
2   Macapá        Amapá            AP      Norte
3   Manaus      Amazonas           AM      Norte
4  Salvador        Bahia            BA  Nordeste
```

8.2.2.1. Renomeando colunas

```
In [ ]: dfr.rename(columns={'Siglas dos Estados':'UF', 'Regiões':'Regioes'}, inplace=True)
dfr.head()
```

	Capitais	Estados	UF	Regioes
0	Rio Branco	Acre	AC	Norte
1	Maceió	Alagoas	AL	Nordeste
2	Macapá	Amapá	AP	Norte
3	Manaus	Amazonas	AM	Norte
4	Salvador	Bahia	BA	Nordeste

8.2.2.2. Contando quantidade de valores nulos do DataFrame

In []:

```
dfr.isna().sum()
```

Out[]:

Capitais	0
Estados	0
UF	0
Regioes	0
dtype: int64	

8.2.3. Unificando a tabela 'ERBs - Antenas' com 'Regiões'

In []:

In []:

```
dfa = dfa.merge(dfr, on='UF', how='left')
dfa.head()
```

Out[]:

	Empresas	UF	Municipio	2G	3G	4G	5G	Capitais	Estados	Regioes
0	CLARO	RO	Alta Floresta D'Oeste	SIM	SIM	SIM	NÃO	Porto Velho	Rondônia	Norte
1	OI	RO	Alta Floresta D'Oeste	SIM	NÃO	NÃO	NÃO	Porto Velho	Rondônia	Norte
2	TIM	RO	Alta Floresta D'Oeste	SIM	NÃO	NÃO	NÃO	Porto Velho	Rondônia	Norte
3	CLARO	RO	Alta Floresta D'Oeste	SIM	SIM	SIM	NÃO	Porto Velho	Rondônia	Norte
4	VIVO	RO	Alta Floresta D'Oeste	SIM	SIM	SIM	NÃO	Porto Velho	Rondônia	Norte

8.2.3.1. Renomeando valores para realização de consultas

Alterando valores SIM e NÃO para 1 e 0 para a realização de consultas

In []:

```
dfa.replace('SIM', 1, inplace=True)
dfa.replace('NÃO', 0, inplace=True)
```

9. Plotagem - Consultas iniciais

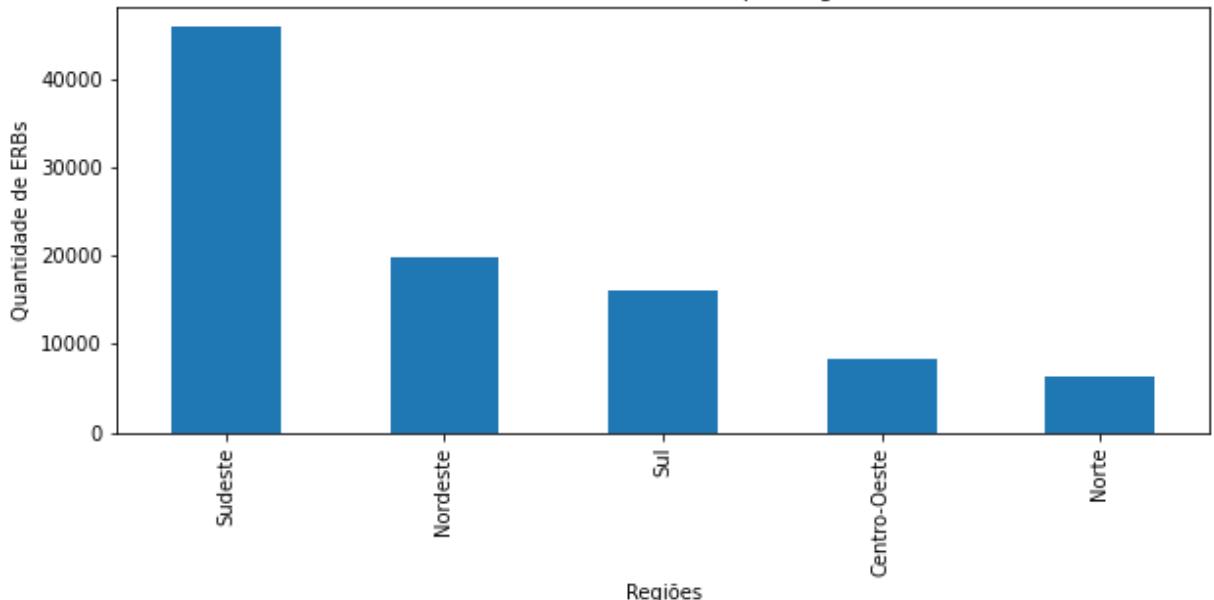
9.1. Qual região possui mais ERBs?

In []:

```
print(dfa['Regioes'].value_counts().plot.bar(xlabel='Regiões', ylabel='Quantidade de
                                               title='Quantidade de antenas por Região')
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```

Quantidade de antenas por Região



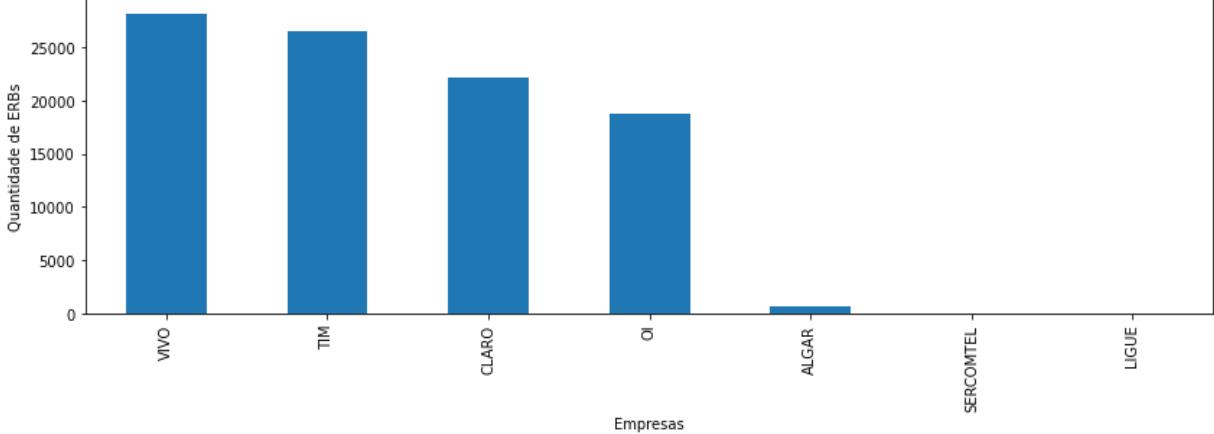
9.2. Quais empresas possuem mais ERBs?

```
In [ ]:
```

```
print(dfa['Empresas'].value_counts().plot.bar(xlabel='Empresas', ylabel='Quantidade de ERBs', figsize=(14, 4), title='Quantidade de ERBs por Empresa'))
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```

Quantidade de ERBs por Empresas

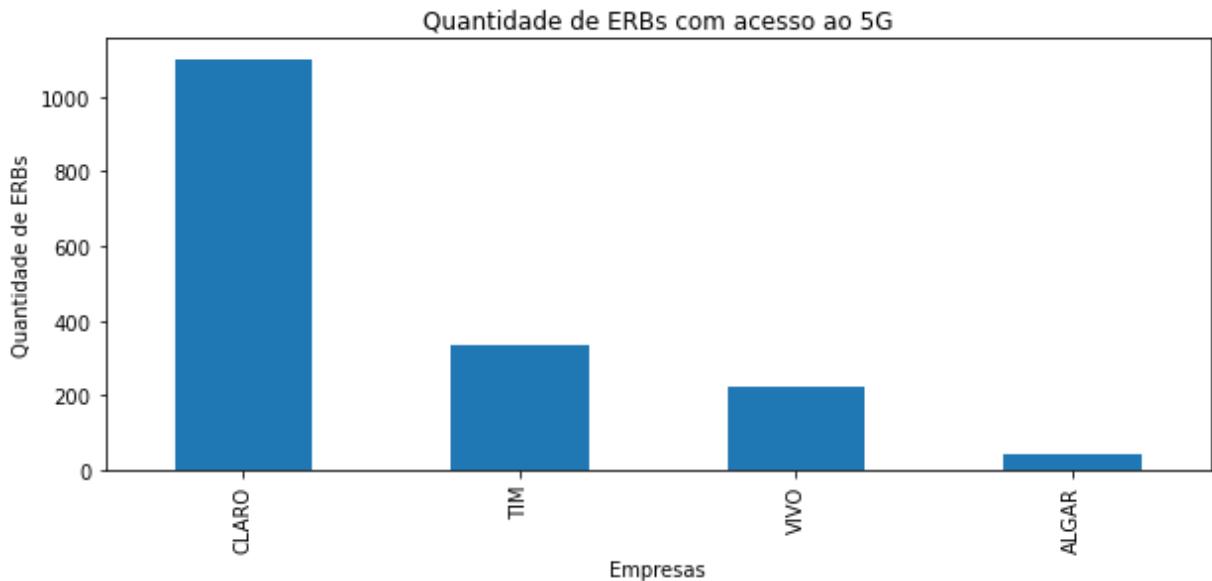


9.3. Qual empresa possui mais ERBs com tecnologia 5G?

```
In [ ]:
```

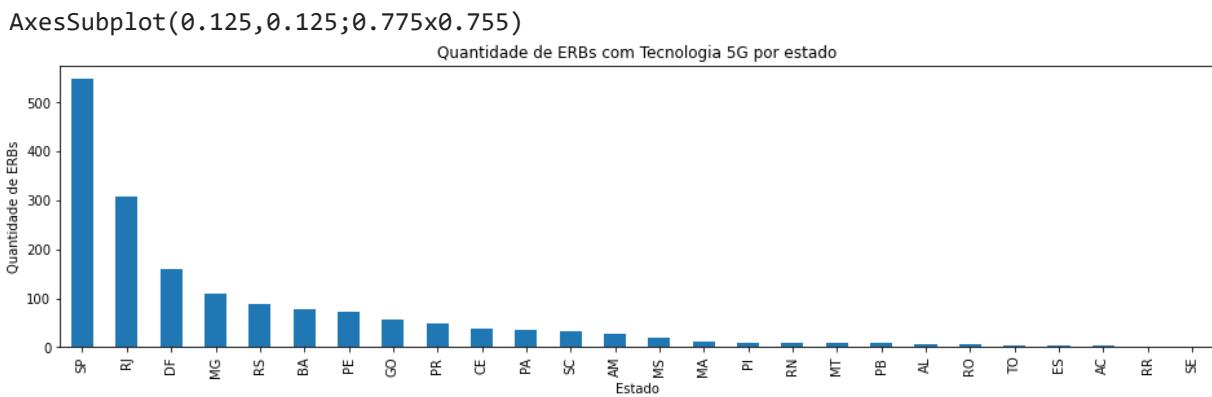
```
print(dfa[dfa['5G'] == 1]['Empresas'].value_counts().plot.bar(xlabel='Empresas', ylabel='Quantidade de ERBs', title='Quantidade de ERBs com tecnologia 5G'))
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



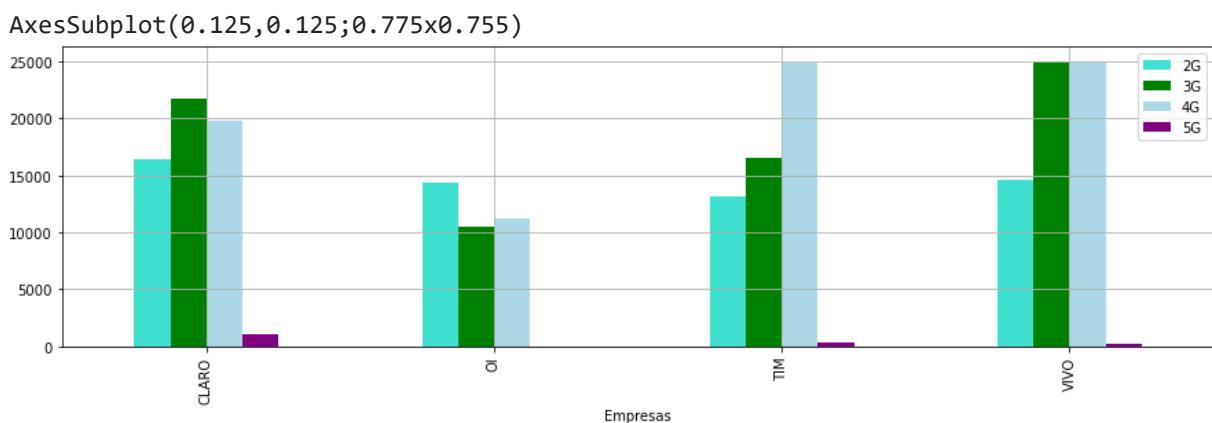
9.4. Qual estado possui mais antenas com tecnologia 5G?

```
In [ ]:
print(dfa[dfa['5G'] == 1]['UF'].value_counts()\n    .plot.bar(figsize=(16,4), xlabel='Estado', ylabel='Quantidade de ERBs', \n              title='Quantidade de ERBs com Tecnologia 5G por estado'))
```



9.5. Comparando as grande empresas, quais as quantidades de ERBs de cada Tecnologia?

```
In [ ]:
print(dfa[dfa['Empresas'].isin('VIVO CLARO TIM OI'.split())].groupby('Empresas').sum\n      color={'2G':'turquoise', '3G':'green', '4G':\n
```



10. Salvando DataFrames Tratados no Cloud Storage

10.1. DataFrame ERBs_Antenas

```
In [ ]: dfa.to_csv('gs://projetofinal/Modificados/Finalizado_Antenas_Pandas.csv', index=False)
print('DF salvo com sucesso.')
```

DF salvo com sucesso.

10.2. DataFrame Banda Larga

Esse código salva o DataFrame em apenas 1 arquivo CSV

```
In [ ]: dfbl.repartition(1).write.format("csv").option("header", "true").save("gs://projetofinal/Modificados/Finalizado_Banda_Larga_PySpark.csv")
print('DF salvo com sucesso.')
```

DF salvo com sucesso.

Observação

O Código abaixo salva mais rápido, porém salva em diversos arquivos diferentes

Esse procedimento não foi utilizado no projeto

```
In [ ]: # dfdl.write.csv('gs://projetofinal/Modificados/Finalizado_Banda_Larga_PySpark.csv', header=True)
```

10.3. DataFrame Telefonia Móvel

```
In [ ]: dfm.repartition(1).write.format("csv").option("header", "true").save("gs://projetofinal/Modificados/Finalizado_Telefonia_Movel_PySpark.csv")
print('DF salvo com sucesso.')
```

DF salvo com sucesso.

10.4. DataFrame Telefonia Fixa

```
In [ ]: dff.repartition(1).write.format("csv").option("header", "true").save("gs://projetofinal/Modificados/Finalizado_Telefonia_Fixa_PySpark.csv")
print('DF salvo com sucesso.')
```

DF salvo com sucesso.

11. Envio do DataFrame para o MongoDB Atlas

11.1. Criando conexão com o Mongo DB Atlas

Chave de acesso MongoDB Atlas já com login/senha SoulCode:

```
mongodb+srv://soulcode:a1b2c3@projetofinal.356e0.mongodb.net/myFirstDatabase?  
retryWrites=true&w=majority
```

```
In [ ]: client = pymongo.MongoClient('mongodb+srv://soulcode:a1b2c3@projetofinal.356e0.mongo  
client
```

```
Out[ ]: MongoClient(host=['projetofinal-shard-00-00.356e0.mongodb.net:27017', 'projetofinal-  
shard-00-02.356e0.mongodb.net:27017', 'projetofinal-shard-00-01.356e0.mongodb.net:27  
017'], document_class=dict, tz_aware=False, connect=True, retrywrites=True, w='major  
ity', authsource='admin', replicaset='atlas-14dul3-shard-0', tls=True)
```

11.2. Criando DataBase, Coleção e Inserindo Dados

Dada a nossa limitação de espaço locado no MongoDB, foi enviado apenas os menores DataFrames.

Os DataFrames de "Telefonia Móvel" e "Banda Larga" são superiores ao espaço gratuito disponível no MongoDB (500mb).

11.2.1. Inserindo os dados do DataFrame "Antenas"

```
In [ ]: antenas = client['Projeto_Final'].Antenas  
antenas.insert_many(dfa.to_dict('records'))
```

```
Out[ ]: <pymongo.results.InsertManyResult at 0x7fe6bd8a7fd0>
```

11.2.1.1. Realizando a contagem para conferência

```
In [ ]: antenas.count_documents({})
```

```
Out[ ]: 96490
```

```
In [ ]: dfa.count()
```

```
Out[ ]: Empresas      96490  
UF            96490  
Municipio    96490  
2G            96490  
3G            96490  
4G            96490  
5G            96490  
Capitais     96490  
Estados      96490  
Regioes      96490  
dtype: int64
```

11.2.2. Inserindo os dados do DataFrame "Telefonia Fixa" em uma nova coleção

```
In [ ]: fixa = client['Projeto_Final'].Telefonia_Fixa  
fixa.insert_many(dff.toPandas().to_dict('records'))
```

```
Out[ ]: <pymongo.results.InsertManyResult at 0x7fe6b81d0460>
```

11.2.2.1. Realizando a contagem para conferência

```
In [ ]: fixa.count_documents({})
```

```
Out[ ]: 517259
```

```
In [ ]: dff.count()
```

```
Out[ ]: 517259
```

12. SparkSQL | Análises

12.1. Criar tabela em tempo de execução para realização de consulta

```
In [ ]: dfbl.createOrReplaceTempView("blsql")
```

12.2. Banda Larga Fixa

```
In [ ]: spark.sql(  
    'Select * from blsql limit 10;'  
) .show()
```

```
[Stage 18:=====] (8 + 1) / 9  
+-----+-----+-----+-----+-----+-----+-----+  
| Ano|Mes| Empresa|Porte_da_Prestadora| UF| Municipio|Faixa_de  
_Velocidade|Tecnologia|Meio_de_Acesso|Acessos|  
+-----+-----+-----+-----+-----+-----+-----+  
| 2020| 12|NET.COM TELECOMUN...| Pequeno Porte| PE| Chã Grande| 12Mb  
ps a 34Mbps| ETHERNET| Cabo Metálico| 19|  
| 2020| 12|CENTURYLINK COMUN...| Pequeno Porte| MG| Belo Horizonte| 2Mb  
ps a 12Mbps| ETHERNET| Cabo Metálico| 32|  
| 2020| 12|VOGEL SOLUCOES EM...| Pequeno Porte| RS| Porto Alegre| 0Kbp  
s a 512Kbps| ETHERNET| Cabo Metálico| 1|  
| 2020| 12| VELOO NET EIRELI| Pequeno Porte| AL| Maribondo| 2Mb  
ps a 12Mbps| ETHERNET| Cabo Metálico| 236|  
| 2020| 12|CENTURYLINK COMUN...| Pequeno Porte| SP| Praia Grande| 512k  
bps a 2Mbps| ETHERNET| Cabo Metálico| 1|  
| 2020| 12| HUGO CHAVES PEREIRA| Pequeno Porte| BA|Vitória da Conquista| 2Mb  
ps a 12Mbps| ETHERNET| Cabo Metálico| 4|  
| 2020| 12|NOVA INTERNET E P...| Pequeno Porte| BA| Ibitiara| 2Mb  
ps a 12Mbps| ETHERNET| Cabo Metálico| 170|  
| 2020| 12|ISRAEL DE ANDRADE...| Pequeno Porte| PR| Piraquara|  
> 34Mbps| ETHERNET| Cabo Metálico| 3|  
| 2020| 12|PGI SERVICOS DE T...| Pequeno Porte| PE| Goiana| 12Mb  
ps a 34Mbps| ETHERNET| Cabo Metálico| 426|  
| 2020| 12|ACENET DO BRASIL ...| Pequeno Porte| SP| São Paulo| 512k  
bps a 2Mbps| ETHERNET| Cabo Metálico| 2|  
+-----+-----+-----+-----+-----+-----+-----+
```

```
-----+-----+-----+-----+
```

12.3. Total de acessos dos 3 últimos anos.

In []:

```
spark.sql('SELECT ANO, SUM(Acessos) FROM blsql \
GROUP BY ANO').show()
```

```
[Stage 30:=====] (8 + 1) / 9]
```

ANO	sum(Acessos)
2019	388393110
2020	414932279
2021	469976308

12.4. Comparativo de crescimento entre Prestadoras de Grande e Pequeno porte nos últimos 3 anos.

In []:

```
spark.sql('WITH bl2019 AS (SELECT Porte_da_Prestadora, Ano, SUM(Acessos) AS Soma_de_\
    AS Porcentagem FROM blsql WHERE Ano = 2019 \
    GROUP BY Porte_da_Prestadora, Ano), \
    \
    bl2020 AS (SELECT Porte_da_Prestadora, Ano, SUM(Acessos) AS Soma_de_Acessos, rou \
    WHERE Ano = 2020 \
    GROUP BY Porte_da_Prestadora, Ano), \
    \
    bl2021 AS (SELECT Porte_da_Prestadora, Ano, SUM(Acessos) AS Soma_de_Acessos, rou \
    WHERE Ano = 2021 \
    GROUP BY Porte_da_Prestadora, Ano) \
    \
    SELECT bl2019.Porte_da_Prestadora, bl2019.Ano, bl2019.Soma_de_Acessos, bl2019.Po \
    bl2020.Ano, bl2020.Soma_de_Acessos, bl2020.Porcentagem, bl2021.Ano, bl2021.Soma_ \
    FROM bl2019 LEFT JOIN bl2020 ON bl2019.Porte_da_Prestadora = bl2020.Porte_da_P \
    bl2020.Porte_da_Prestadora = bl2021.Porte_da_Prestadora').show()
```

```
[Stage 42:==> (4 + 1) / 6][Stage 45:> (0 + 0) / 1][Stage 48:> (0 + 0) / 1]1
```

Porte_da_Prestadora	Ano	Soma_de_Acessos	Porcentagem	Ano	Soma_de_Acessos	Porcentag em	Ano	Soma_de_Acessos	Porcentagem	
Grande Porte	2019	280170257	72.0	2020	267709124	6	5.0	2021	265117279	56.0
Pequeno Porte	2019	108222853	28.0	2020	147223155	3	5.0	2021	204859029	44.0

É notável o crescimento da contagem de acessos oriundo das empresas de pequeno porte.

Mesmo que o número de acessos totais das empresas de Grande porte não tenha sofrido muita alteração, elas estão caindo consideravelmente em relação ao número total de acessos.

As empresas de pequeno porte, por sua vez, praticamente dobraram a quantidade de acessos de 2019 até 2021.

12.5. Quais são as empresas destaque dentre as empresas de Pequeno Porte?

In []:

```
spark.sql('WITH b2019 as (SELECT Empresa, Ano, COUNT(Acessos) as Contagem FROM blsql
WHERE Porte_da_Prestadora = "Pequeno Porte" AND Ano = 2019 \
GROUP BY Empresa, Ano \
ORDER BY Contagem DESC), \
\
b2020 as (SELECT Empresa, Ano, COUNT(Acessos) as Contagem from blsql \
WHERE Porte_da_Prestadora = "Pequeno Porte" AND Ano = 2020 \
GROUP BY Empresa, Ano \
ORDER BY Contagem DESC), \
\
b2021 as (SELECT Empresa, Ano, COUNT(Acessos) as Contagem from blsql \
WHERE Porte_da_Prestadora = "Pequeno Porte" AND Ano = 2021 \
GROUP BY Empresa, Ano \
ORDER BY Contagem DESC) \
\
SELECT b2019.Empresa, b2019.Ano, b2019.Contagem AS Contagem_2019, b2020.Ano, b20
round((b2020.Contagem / b2019.Contagem)*100 - 100) AS Porcentagem_19_20, b2021.A
round((b2021.Contagem / b2020.Contagem)*100 - 100) AS Porcentagem_20_21 \
FROM b2019 INNER JOIN b2020 ON b2019.Empresa = b2020.Empresa INNER JOIN b2021 ON
ORDER BY b2021.Contagem DESC').show(10)
```

	Empresa	Ano	Contagem_2019	Ano	Contagem_2020	Porcentagem_19_20	Ano	Contagem_2021	Porcentagem_20_21
273660	HUGHES TELECOMUNI...	2019	99396	2020	130258	31.0	2021	110.0	
106749	ALGAR (CTBC TELECOM)	2019	38635	2020	41878	8.0	2021	155.0	
83674	BT	2019	82295	2020	80531	-2.0	2021	4.0	
44320	UNIFIQUE TELECOMU...	2019	5835	2020	7937	36.0	2021	458.0	
34370	MHNET TELECOMUNIC...	2019	4307	2020	5665	32.0	2021	507.0	
24814	BRASIL SERVICOS D...	2019	5188	2020	6970	34.0	2021	256.0	
23139	MOB SERVICOS DE T...	2019	1838	2020	2215	21.0	2021	945.0	
20825	Brisanet Servicos...	2019	7418	2020	6667	-10.0	2021	212.0	
19286	PREFEITURA DE LON...	2019	16476	2020	19801	20.0	2021	-3.0	
18380	VOGEL SOLUCOES EM...	2019	11101	2020	10755	-3.0	2021	71.0	

Todas as 10 Empresas com mais acessos em 2021 tiveram crescimento em relação a 2020 e em quase todas, esse crescimento foi expressivo.

12.6. Dentre as Grandes empresas, quais se destacou em crescimento 2021?

In []:

```
spark.sql('WITH b2019 AS (SELECT Empresa, Ano, COUNT(Acessos) AS Contagem FROM blsql
  WHERE Porte_da_Prestadora = "Grande Porte" AND Ano = 2019 \
  GROUP BY Empresa, Ano \
  ORDER BY Contagem DESC), \
\
  b2020 AS (SELECT Empresa, Ano, COUNT(Acessos) AS Contagem FROM blsql \
  WHERE Porte_da_Prestadora = "Grande Porte" AND Ano = 2020 \
  GROUP BY Empresa, Ano \
  ORDER BY Contagem DESC), \
\
  b2021 AS (SELECT Empresa, Ano, COUNT(Acessos) AS Contagem FROM blsql \
  WHERE Porte_da_Prestadora = "Grande Porte" AND Ano = 2021 \
  GROUP BY Empresa, Ano \
  ORDER BY Contagem DESC) \
\
  SELECT b2019.Empresa, b2019.Ano, b2019.Contagem AS Contagem_2019, b2020.Ano, b20
  round((b2020.Contagem / b2019.Contagem)*100 - 100) AS Porcentagem_19_20, b2021.A
  round((b2021.Contagem / b2020.Contagem)*100 - 100) AS Porcentagem_20_21 \
  FROM b2019 INNER JOIN b2020 ON b2019.Empresa = b2020.Empresa INNER JOIN b2021 ON
  ORDER BY b2021.Contagem DESC').show(10)
```

[Stage 82:=====]>		(1 + 1) / 2]		[Stage 85:>	(0 + 0) / 1]	
+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+	
Empresa Ano Contagem_2019 Ano Contagem_2020 Porcentagem_19_20 Ano Contagem_2021						
Porcentagem_20_21						
+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+
OI 2019 608908 2020 619344 2.0 2021 1169493						
89.0						
CLARO 2019 182833 2020 186969 2.0 2021 441691						
136.0						
VIVO 2019 97243 2020 97224 0.0 2021 326077						
235.0						
TIM 2019 2686 2020 2587 -4.0 2021 16719						
546.0						
SKY/AT&T 2019 3618 2020 4393 21.0 2021 6481						
48.0						
+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+
+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+	+-----+-----+

A TIM com certeza teve um crescimento grande em porcentagem, porém sua quantidade é consideravelmente baixa em relação às outras.

Considerando 2020 para 2021, a OI dobrou seu número.

FIM

Referências

Datasets:

1. **Descrição:** O Mapa de Antenas é uma ferramenta que permite verificar a cobertura do celular e dos serviços de internet móvel instalados no país, no estado ou no município, com a geolocalização de cada antena. A expansão da infraestrutura de telecomunicações e o ritmo de ampliação da cobertura dependem de leis municipais atualizadas e modernas, sem burocracia nos processos de licenciamentos.

Link: <https://conexis.org.br/numeros/mapa-de-antenas/>

2. **Descrição:** Observação: considerando que uma versão do serviço móvel de 5ª Geração (5G-DSS) já está disponível em algumas partes do país, assim como aparelhos compatíveis com a tecnologia, as prestadoras passam a informar a quantidade de acessos desta tecnologia a partir do mês de agosto de 2021.

Os dados apresentados referem-se aos acessos de Telefonia Móvel (Serviço Móvel Pessoal – SMP), enviados pelas prestadoras do serviço.

Os números de acessos por município para o serviço de Telefonia Móvel estão disponíveis somente a partir de janeiro de 2019. Por esse motivo, não são exibidos os dados por município caso seja selecionado um período anterior a 2019. Também a partir de 2019 o cálculo da densidade do serviço considera somente acessos do tipo "Padrão", não considerando em seu cômputo acessos do tipo "Ponto de Serviço" e "M2M". Por se tratar de uma nova coleta, os dados de acesso por município (2019) estão sendo validados pela Anatel junto às prestadoras do serviço móvel.

Link: <https://informacoes.anatel.gov.br/paineis/acessos/telefonia-movel>

3. **Descrição:** Os dados apresentados referem-se aos acessos de Telefonia Fixa (Serviço de Telefonia Fixa Comutada – STFC), enviados pelas prestadoras do serviço. Não estão incluídos os acessos referentes aos telefones de uso público (orelhões).

Atenção: a Anatel passa a publicar a densidade do serviço de Telefonia Fixa com base na população e não mais nos números de domicílios, o que permitirá comparações mais diretas com indicadores de organismos internacionais.

Link: <https://informacoes.anatel.gov.br/paineis/acessos/telefonia-fixa>

4. **Descrição:** Tabela de regiões.

Link: <https://www.todamateria.com.br/capitais-do-brasil/>

Notícias:

5. **Descrição:** Venda Telefonia Móvel OI

Link: <https://www.convergenciadigital.com.br/Opiniao/O-fim-da-Oi-Movel-59405.html?UserActiveTemplate=mobile>

6. **Descrição:** Recuperação judicial OI
Link: <https://www.infomoney.com.br/mercados/oi-oibr3-oibr4-super-tele-a-maior-recuperacao-judicial-do-brasil-entenda-o-que-aconteceu/>

Workflow:

7. **Descrição:** Workflow com as etapas do projeto e suas respectivas tecnologias.
Link: <https://whimsical.com/telecom-NkmvD9ZR3p4yzW6bTwinXT>

Imagens:

8. **Descrição:** Vetores e ilustrações e imagens que compõe o projeto.
Link: <https://br.freepik.com>
9. **Descrição:** Fotografia de capa.
Link: <https://www.gov.br/secretariageral/pt-br/noticias/2019/outubro/presidente-bolsonaro-sanciona-novo-marco-legal-das-telecomunicacoes>