



**SoulCode - Eng de Dados**


Acessos dos serviços de telecomunicações  
do Brasil por município.

# Apresentação

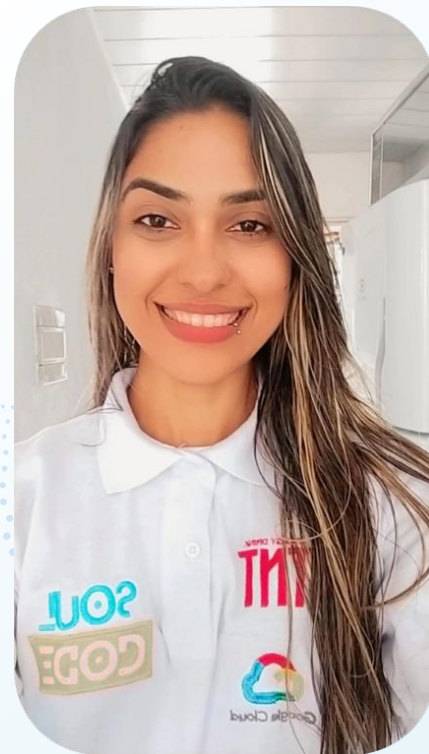
## Juliana Maciel

28 anos, Engenharia de Dados

Redes de Computadores

 [linkedin/ju-maciel](https://www.linkedin.com/in/ju-maciel)

 [github.com/ju-maciel](https://github.com/ju-maciel)



# Introdução ao Projeto

## **ETL da database**

Sobre o tema telecomunicações.

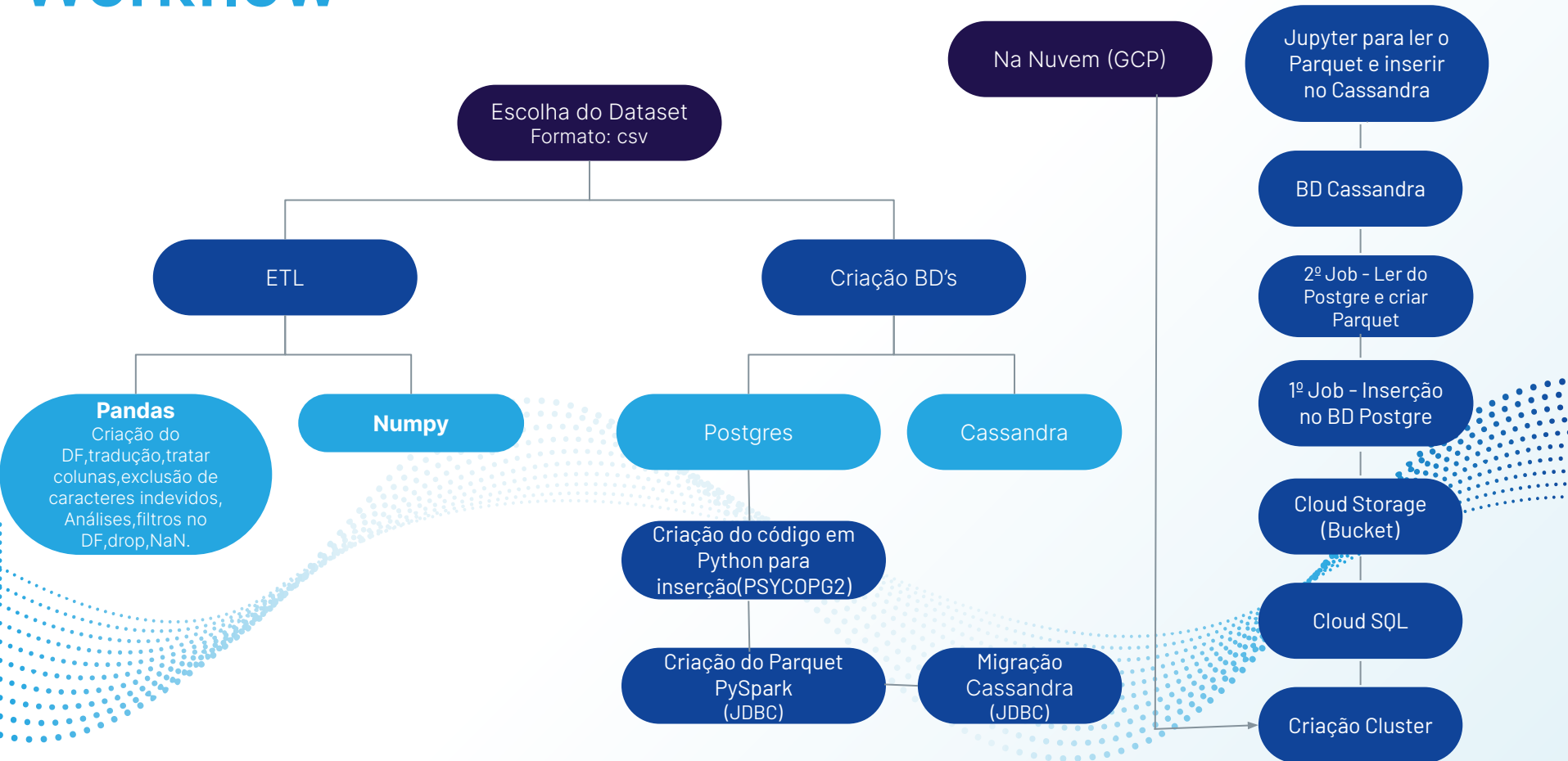
## **Migração de dados de um banco SQL para um Banco NoSQL**

Usando arquivos do formato Parquet e jobs PySpark.

## **Realizar o Processo em Nuvem**

Google Cloud Platform

# Workflow



# Processos do Projeto

Acessos dos serviços de telecomunicações  
do Brasil por município.

# Acessos dos serviços de telecomunicações do Brasil por município.

66.837 dados.

Período: 2019 a 2021.

Fonte: <https://dados.gov.br/dataset/meu-municipio-anatel>





# ETL

```
[1] import pandas as pd
import numpy as np

# separador; #codificação do texto
df = pd.read_csv('//content/drive/MyDrive/Meu_Municipio_Acessos.csv', sep=';', encoding='Utf-8')
```

```
[11] df.head(2)
```

	Ano	Mes	Acessos	Servico	Cod_IBGE	Municipio	UF	Regiao
0	2021	6	5597	Telefonia Móvel	4118709	Paulo Frontin PR	PR	Sul
1	2021	6	713	Banda Larga Fixa	4118709	Paulo Frontin PR	PR	Sul

```
[7] # renomeando colunas
df.columns=['Ano', 'Mes', 'Acessos', 'Servico', 'Densidade', 'Cod_IBGE', 'Municipio', 'UF', 'Nome_UF', 'Regiao', 'Cod_nacional']
```

```
[8] # excluindo colunas
df = df.drop(columns=['Densidade', 'Nome_UF', 'Cod_nacional'])
```

```
[9] # tamanho do dataframe
df.shape
```

(66837, 8)

```
[10] # verificação de dados faltantes
df.info()
```

```
[ ] # verificação de dados nulos
df.isnull().sum()
```

```
[ ] # substituir nulos por 0
df.fillna(0, inplace = True)
```

# SGBD PostgreSQL

Descrição das Etapas.



PostgreSQL



CREATE TABLE IF NOT EXISTS operadora (

id\_op serial constraint PK\_operadoras primary key,

Ano int not null,

Mes varchar (100) not null,

Acessos int not null,

Servico varchar (400) not null,

Cod\_IBGE int not null,

Municipio varchar (400) not null,

UF varchar (100) not null,

Regiao varchar (200) not null

);

operadora	
123	id_op
123	ano
ABC	mes
123	acessos
ABC	servico
123	cod_ibge
ABC	municipio
ABC	uf
ABC	regiao

# Código de Conexão com o Banco Relacional

```
import psycopg2
```

```
class Conectar:
```

```
    def __init__(self, host, database, user, password):  
        self.host = host  
        self.database = database  
        self.user = user  
        self.password = password
```

```
    def conectar(self):  
        conect = psycopg2.connect(host=self.host,  
                                   database=self.database,  
                                   user=self.user, password=self.password)  
        return conect
```

```
    def executar(self, query):  
        con = self.conectar()  
        cursor = con.cursor()  
        cursor.execute(query)  
        cursor.close()  
        con.commit()  
        return "Acao Feita!"
```

```
    def inserir_array(self, table, parametros, valores):  
        query = f"INSERT INTO {table}  
        ({parametros}) VALUES {valores}"  
        self.executar(query)  
        #print(query)
```

# Inserção no Postgre da Nuvem

Código que apresenta a Conexão com o Postgres da Nuvem e a inserção dos Dados:

```
from conector_postgres import Conectar

import pandas as pd
import numpy as np

if __name__ == '__main__':

    conexao = Conectar(host='34.151.208.93',
database='operadora2', user='postgres',password='ROOT')

    df = pd.read_csv(r'gs://pasta-scripts/operadoras2/
Meu_Municipio_Acessos.csv', sep=';', encoding = 'utf-8')

    array = np.array(df)

    # iteração
    lista = []
    for i in array:
        x = (i[0], i[1], i[2], i[3], i[4], i[5], i[6], i[7])
        lista.append(x)
    # print(lista)
    lista = str(lista)[1:-1]
```



```
# inserção por array
conexao.inserir_array('operadora', 'Ano,
Mes, Acessos, Servico, Cod_IBGE, Municipio,
UF, Regiao', lista)
```

```
print(c1.selecionar("select count(*) from operadora"))

... [(66837,)]
```

# SGBD Cassandra

Descrição das Etapas.



```
create keyspace if not exists operadora2 with replication = {'class':  
'SimpleStrategy', 'replication_factor': 1};
```

```
CREATE TABLE IF NOT EXISTS "operadora2"."operadora" (  
  id_op int primary key,  
  Ano int,  
  Mes text,  
  Acessos int,  
  Servico text,  
  Cod_IBGE int,  
  Municipio text,  
  UF text,  
  Regiao text  
);
```

# Transformando os dados em Parquet

Código que apresenta a extração dos dados no Postgre da nuvem e a inserção como Parquet no Cloud Storage:



```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName('Postgres para Parquet').getOrCreate()

def lendo_transformando_parquet(nome_tabela, caminho_parquet):
    url = 'jdbc:postgresql://34.151.208.93:5432/operadora2'
    properties = {
        'user': 'postgres',
        'password': 'ROOT',
        'driver': 'org.postgresql.Driver'
    }
    df = spark.read.jdbc(url=url, table=nome_tabela, properties=properties)

    df.write.parquet(caminho_parquet)
    return f"Parquet add em {caminho_parquet}"

lendo_transformando_parquet('operadora', 'gs://pasta-scripts/parquetofic')
```

6,17 MB (csv) ➡ 770 KB (parquet)

# Inserção no Cassandra

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import *

spark = SparkSession \
    .builder \
    .appName("Spark Cassandra App") \
    .config("spark.jars.packages",
"com.datastax.spark:spark-cassandra-connector_2.12:3.1.0") \
    .config("spark.sql.extensions",
"com.datastax.spark.connector.CassandraSparkExtensions") \
    .config("spark.cassandra.connection.host",
"34.71.101.86") \
    .config("spark.cassandra.connection.port", "9042") \
    .getOrCreate()

keyspace = "operadora2"

def inserindo_cassandra(df, table):
    df.write \
        .format("org.apache.spark.sql.cassandra") \
        .option("keyspace", keyspace) \
        .option("table", table) \
        .mode('append') \
        .save()
```

Jupyter  
Na GCP

Código da Conexão com o Cassandra e a inserção dos Dados

```
df1_operadoras =
spark.read.parquet(r"gs://pasta-scripts/parquetof
ic")

inserindo_cassandra(df1_operadoras, "operadora")

df1_operadoras.count()
```



# Google Cloud Platform

Descrição das Etapas.



## Dataproc

Jobs em clusters

Clusters

Jobs

Fluxos de trabalho

Políticas de escalonamento ...

Sem servidor

Lotes

Utilitários

Troca de componentes

Metastore

Workbench

Notas de lançamento

&lt;|

## ← Detalhes do cluster

+ ENVIAR JOB

↻ ATUALIZAR

▶ INICIAR

■ INTERROMPER

🗑 EXCLUIR

☰ VER REGISTROS

Região	southamerica-east1
Zona	southamerica-east1-a
Escalonamento automático	Desativado
Metastore do Dataproc	Nenhum
Exclusão programada	Desativado
Nó mestre	Padrão (1 mestre, N workers)

Tipo de máquina	n1-standard-2
Número de GPUs	0
Tipo de disco principal	pd-standard
Tamanho do disco principal	30 GB
SSDs locais	0

Nós de trabalho	2
Tipo de máquina	n1-standard-2
Número de GPUs	0
Tipo de disco principal	pd-standard
Tamanho do disco principal	30 GB
SSDs locais	0

Nós de trabalho secundários	0
Inicialização segura	Desativada
VTPM	Desativada
Monitoramento de integridade	Desativada
Bucket de preparação do Cloud Storage	<a href="#">pasta-scripts</a>

Rede	default
Tags de rede	Nenhum
Apenas IP interno	Não

Versão da imagem	2.0.30-debian10
Criado em	11 de fev. de 2022 13:53:20

Componentes opcionais JUPYTER

Propriedades [Mostrar propriedades](#)

Nome

cluster-c5b5

UUID do cluster

00537c6a-dd23-45a1-8b28-ea6fd3538548

Tipo

Cluster do Dataproc

Status

✓ Em execução

Cloud  
Dataproc

## Conectar-se a esta instância

Endereço IP público

34.151.233.62



Endereço IP de saída

34.151.208.93



Nome da conexão

projeto-energia-340918:southamerica-east1:post1234



### Precisa de ajuda com a conexão?

Leia a documentação para saber mais sobre as diversas maneiras de se conectar à instância. [Saiba mais](#)

Para se conectar usando o gcloud,

[ABRIR O CLOUD SHELL](#)

## Configuração

vCPUs

4

Memória

26 GB

Armazenamento SSD

100 GB



Google Cloud Platform

projeto-energia



SQL

Bancos de dados

#### INSTÂNCIA PRINCIPAL



Visão geral



Insights de consulta



Conexões



Usuários



Bancos de dados



Backups



Réplicas



Operações

Todas as instâncias > post1234

✓ **post1234**

PostgreSQL 14



**CRIAR BANCO DE DADOS**

Nome ↑

Compilação

Conjunto d

energia

en\_US.UTF8

UTF8

operadora2

en\_US.UTF8

UTF8

postgres



Cloud SQL

Google Cloud Platform

projeto-energia

Pesquisa

34.151.233.62

Cloud Storage

Navegador

Monitoramento

Configurações

Marketplace

Notas de lançamento

Detalhes do bucket

pasta-scripts

Local

Classe de armazenamento

Acesso público

Proteção

us (várias regiões nos Estados Unidos)

Standard

Não público

Nenhum

OBJETOS

CONFIGURAÇÃO

PERMISSÕES

PROTEÇÃO

CICLO DE VIDA

Intervalos

pasta-scripts

operadoras2

FAZER UPLOAD DE ARQUIVOS

CARREGAR PASTA

CRIAR PASTA

GERENCIAR RETENÇÕES

FAZER O DOWNLOAD

EX

Filtrar apenas pelo prefixo do nome

Filtro

Filtrar objetos e pastas

	Nome	Tamanho	Tipo	Criado
	<div></div> Meu_Municipio_Acessos.csv	6 MB	application/vnd.ms-excel	16 de fe...
	<div></div> dataset/	—	Pasta	—
	<div></div> scripts_banco/	—	Pasta	—
	<div></div> scripts_python/	—	Pasta	—

Google Cloud Storage

🔗

Dataproc

Jobs em clusters

^

🌐

Clusters

☰

Jobs

👤

Fluxos de trabalho

📊

Políticas de escalonamen...

Sem servidor

^

☰

Lotes

Utilitários

^

🔧

Troca de componentes

🔗

Metastore

📁

Workbench

📝

Notas de lançamento

Jobs

+

ENVIAR JOB

↻

ATUALIZAR

■

INTERROMPER

🗑

EXCLUIR

REGIÕES ▼

+

2 ALERTAS RECOMENDADOS

O

☰

Filtro

Filtrar jobs

?

☰

<input type="checkbox"/>	ID do job	Status	Região	Tipo	Cluster	Horário de início
<input type="checkbox"/>	job-c320c59e	✅ Concluído	southamerica-east1	PySpark	cluster-c5b5	17 de fev. de 202
<input type="checkbox"/>	job-f816c36f	❌ Falha	southamerica-east1	PySpark	cluster-c5b5	17 de fev. de 202
<input type="checkbox"/>	job-6e757f49	❌ Falha	southamerica-east1	PySpark	cluster-c5b5	17 de fev. de 202
<input type="checkbox"/>	job-2609f2b5	❌ Falha	southamerica-east1	PySpark	cluster-c5b5	16 de fev. de 202
<input type="checkbox"/>	job-b446d94c	❌ Falha	southamerica-east1	PySpark	cluster-c5b5	16 de fev. de 202
<input type="checkbox"/>	job-4fb47c95	✅ Concluído	southamerica-east1	PySpark	cluster-c5b5	16 de fev. de 202
<input type="checkbox"/>	job-af0dd29e	✅ Concluído	southamerica-east1	PySpark	cluster-c5b5	16 de fev. de 202
<input type="checkbox"/>	job-155c0ec8	❌ Falha	southamerica-east1	PySpark	cluster-c5b5	16 de fev. de 202
<input type="checkbox"/>	job-da5b77f1	✅ Concluído	southamerica-east1	PySpark	cluster-c5b5	11 de fev. de 202
<input type="checkbox"/>	job-3a010a51	❌ Falha	southamerica-east1	PySpark	cluster-c5b5	11 de fev. de 202
<input type="checkbox"/>	job-e10b5d98	❌ Falha	southamerica-east1	PySpark	cluster-c5b5	11 de fev. de 202
<input type="checkbox"/>	job-43c0a876	❌ Falha	southamerica-east1	PySpark	cluster-c5b5	11 de fev. de 202
<input type="checkbox"/>	job-17bb765f	✅ Concluído	southamerica-east1	PySpark	cluster-c5b5	11 de fev. de 202
<input type="checkbox"/>	job-0245babd	✅ Concluído	southamerica-east1	PySpark	cluster-c5b5	11 de fev. de 202

Nenhum job selecionado

PERMISSÕES

MARCADORE

📘

Selecione pelo menos u

Jobs

GCP



## Deployment Manager

Implantações

Registro do tipo

cassandra-3

EXCLUIR

✓ cassandra-3 foi implantado

Overview - cassandra-3

▼ cassandra cassandra.jinja

▼ db-tier db\_tier.jinja

▼ cassandra-3-db-vm-tmpl-0 vm\_instance.py

■ cassandra-3-db-vm-0 instância de VM

■ cassandra-3-db-vm-0-data disco

▼ cassandra-3-db-vm-tmpl-1 vm\_instance.py

■ cassandra-3-db-vm-1 instância de VM

■ cassandra-3-db-vm-1-data disco

▼ cassandra-3-db-vm-tmpl-2 vm\_instance.py

■ cassandra-3-db-vm-2 instância de VM

■ cassandra-3-db-vm-2-data disco

▼ software-status software\_status.py

■ cassandra-3-db-config config

■ cassandra-3-db-software encarregado de configuração

■ cassandra-3-db-tcp-9042 firewall

✕ cassandra



## Cassandra

Solução fornecida por Google Click to Deploy

Zone us-central1-a

▼ MAIS SOBRE O SOFTWARE

### Cassandra: primeiros passos

SSH

### Próximas etapas sugeridas

- **Read the getting started guide**  
View the [Apache Documentation](#) (Start on configuring Cassandra).
- **Open TCP port 7000-7001 traffic for Cassandra**  
This firewall rule is not enabled. To allow specific network traffic from the Internet, create a firewall rule to open TCP port 7000-7001 traffic for target tag "cassandra-3-db-tier". [Learn more](#)  
If you are using Google Cloud SDK, type the following command in the terminal:

```
$ gcloud --project=projeto-energia-340918 compute firew
```

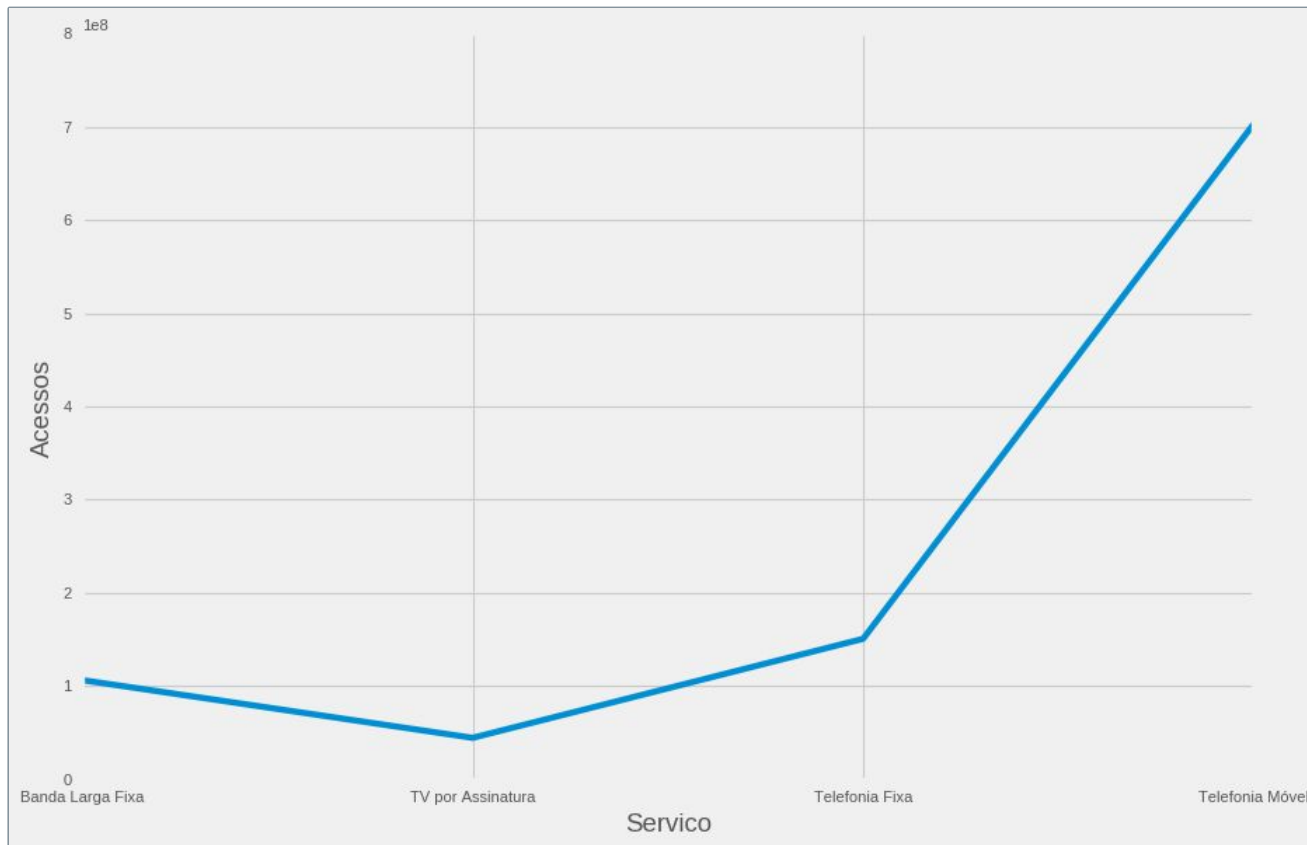
Cassandra  
GCP

# Análise Exploratória

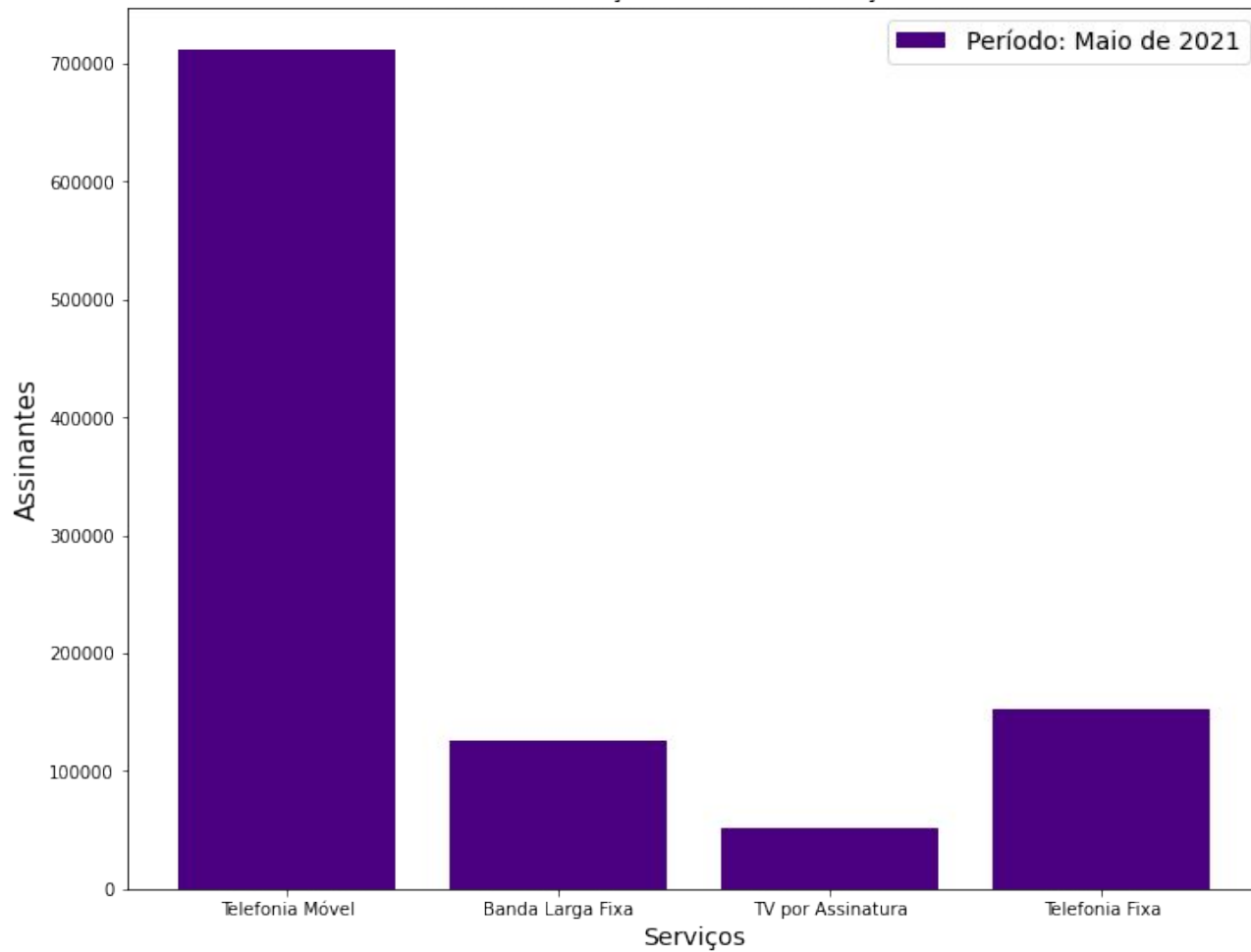
The background is a solid blue color. Overlaid on this background are several wavy, dotted lines in a darker shade of blue. These lines form a series of peaks and valleys, resembling a stylized wave or a data visualization. The dots are small and closely spaced, creating a textured effect. The waves start from the left side, move towards the center, and then curve upwards towards the right side of the image.



## Total de Assinantes por Serviços de Telecomunicações do Brasil por município



Acessos dos Serviços de Telecomunicações no ES



The background is a solid blue color. Overlaid on this are several wavy, horizontal lines composed of small, dark blue dots. These lines create a sense of motion and depth, with some lines appearing more prominent than others. The dots are arranged in a way that suggests a 3D effect, with some lines curving upwards and others downwards.

Obrigada!