```
!pip install polars
!pip install fastexcel
```

```
Requirement already satisfied: polars in /usr/local/lib/python3.12/dist-packages (1.25.2)
Collecting fastexcel
  Downloading fastexcel-0.16.0-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (7.0 kB)
Downloading fastexcel-0.16.0-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.3/3.3 MB 42.1 MB/s eta 0:00:00
Installing collected packages: fastexcel
Successfully installed fastexcel-0.16.0
```

```
import polars as pl
```

```
aeroportos = pl.read_csv("airports.csv",
                         columns = ["IATA_CODE", "CITY", "STATE"])
aeroportos.head(2)
```

shape: (2, 3)

IATA_CODE	CITY	STATE
str	str	str
"ABE"	"Allentown"	"PA"
"ABI"	"Abilene"	"TX"

```
wdi = pl.read_excel("WDIEXCEL.xlsx", sheet_name = "Country",
                    columns = ["Short Name", "Region"])
wdi.head(2)
```

shape: (2, 2)

Short Name	Region
str	str
"Aruba"	"Latin America & Caribbean"
"Afghanistan"	"South Asia"

```
df = pl.DataFrame({
    "grupo": ["A", "A", "B", "B", "C"],
    "valor1": [10, 15, 10, None, 25],
    "valor2": [5, None, 20, 30, None]
})
df
```

shape: (5, 3)

grupo	valor1	valor2
str	i64	i64
"A"	10	5
"A"	15	null
"B"	10	20
"B"	null	30
"C"	25	null

```
df["valor1"]
```

shape: (5,)

valor1
i64
10
15
10
null
25

```
df["valor1"].mean()
```

```
15.0
```

```
df["valor1"].drop_nulls()
```

shape: (4,)

valor1
i64
10
15
10
25

```
df["valor1"].drop_nulls().mean()
```

```
15.0
```

```
df.select([
  pl.col("valor1").mean().alias("media_v1"),
  pl.col("valor2").mean()
])
```

shape: (1, 2)

media_v1	valor2
f64	f64
15.0	18.333333

```
df.group_by("grupo").agg([
  pl.col("valor1").mean().alias("media_valor1"),
  pl.col("valor2").min().alias("min_valor2")
]).sort("grupo")
```

shape: (3, 3)

grupo	media_valor1	min_valor2
str	f64	i64
"A"	12.5	5
"B"	10.0	20
"C"	25.0	null

```
voos = pl.read_csv("flights.csv",
                   columns = ["AIRLINE", "ARRIVAL_DELAY", "DESTINATION_AIRPORT"],
                   dtypes = {"AIRLINE": pl.Utf8,
                             "ARRIVAL_DELAY": pl.Int32,
                             "DESTINATION_AIRPORT": pl.Utf8})
voos.shape
```

```
/tmp/ipython-input-2284422991.py:1: DeprecationWarning: The argument `dtypes` for `read_csv` is deprecated. It ha
  voos = pl.read_csv("flights.csv",
(93070, 3)
```

```
voos.head(3)
```

shape: (3, 3)

AIRLINE	DESTINATION_AIRPORT	ARRIVAL_DELAY
str	str	i32
"AS"	"SEA"	-22
"AA"	"PBI"	-9
"US"	"CLT"	5

```
resultado = (
  voos.drop_nulls(["AIRLINE", "DESTINATION_AIRPORT", "ARRIVAL_DELAY"])
  .filter(
    pl.col("AIRLINE").is_in(["AA", "DL"]) &
    pl.col("DESTINATION_AIRPORT").is_in(["SEA", "MIA", "BWI"])
    )
    .group_by(["AIRLINE", "DESTINATION_AIRPORT"])
    .agg([
      (pl.col("ARRIVAL_DELAY") > 30).mean().alias("atraso_medio")
      ])
)
resultado.sort("atraso_medio")
```

shape: (6, 3)

AIRLINE	DESTINATION_AIRPORT	atraso_medio
str	str	f64
"DL"	"BWI"	0.120879
"DL"	"MIA"	0.168919
"DL"	"SEA"	0.169312
"AA"	"BWI"	0.195652
"AA"	"MIA"	0.226119
"AA"	"SEA"	0.373333

Comece a programar ou gere código com IA.

```
from datetime import datetime
import pytz

# Define o fuso horário de Brasília
brasilia_timezone = pytz.timezone('America/Sao_Paulo')

# Obtém a data e hora atuais no fuso horário de Brasília
now_brasilia = datetime.now(brasilia_timezone)

# Imprime a data e hora
print(now_brasilia.strftime('%Y-%m-%d %H:%M:%S %Z%z'))
```

```
2025-10-02 10:58:31 -03-0300
```