

# **BASE DE DADOS DE REPOSITÓRIOS GITHUB**

Criação e Análise

**Lucas Jesus Santos Silva**  
**Maria Eduarda Mendes Leite**



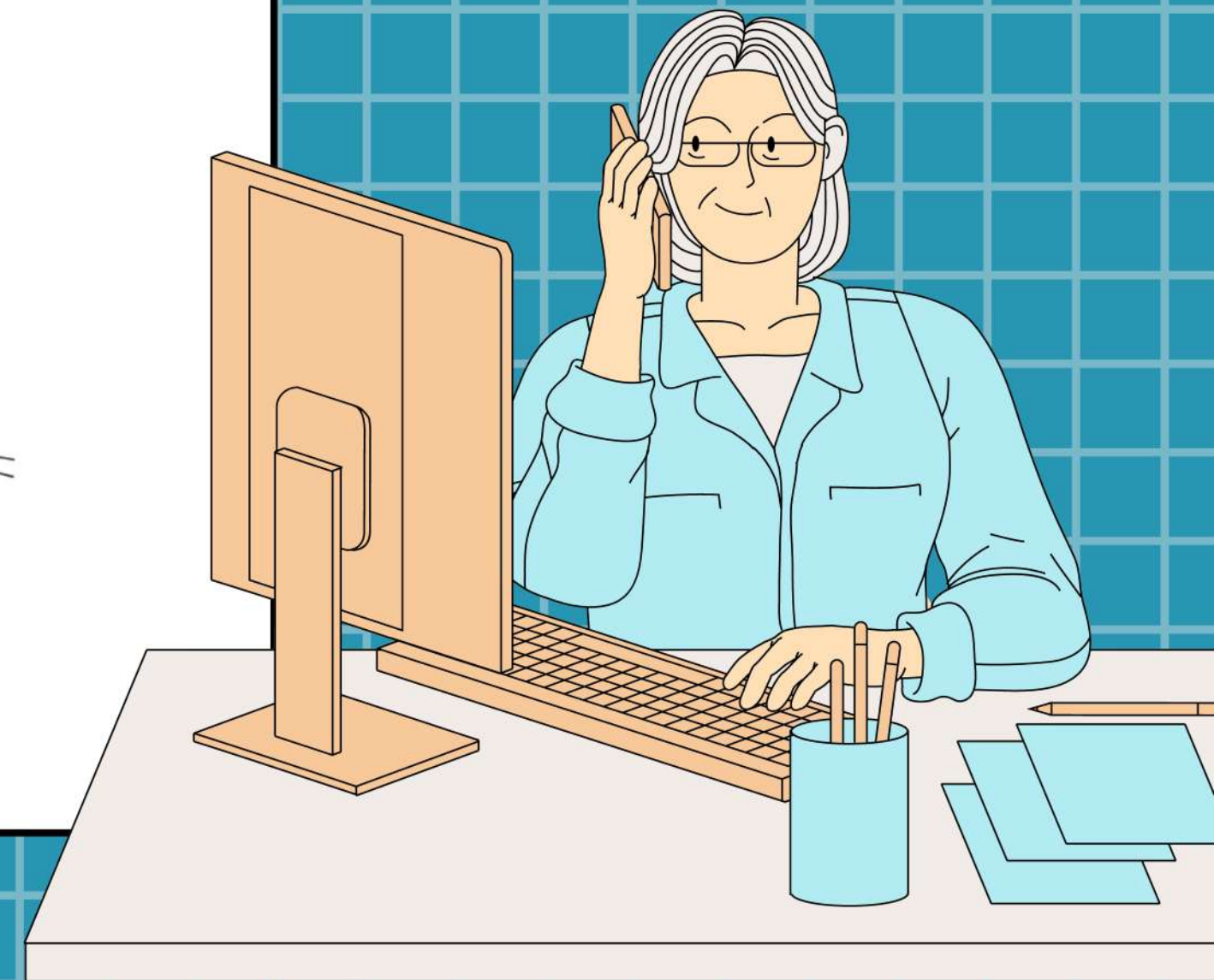
# O QUE É GITHUB?

Armazenamento de código

Organização e controle de versões - Git

Colaboração

Gerenciamento de projeto





# BASE DE DADOS

**Fonte:** extração via API oficial do GitHub

**Critério:** selecionados 1000 repositórios para cada uma das linguagens:

- Python
- JavaScript
- Java
- C#
- C++
- TypeScript
- Go
- Rust
- Kotlin
- Swift



# BASE DE DADOS

**Tamanho:** coletados 9.450 repositórios no total

**Estrutura:** 18 colunas com informações como nome, estrelas, forks, data de criação, entre outras

**Tempo de coleta:** mais de 35 horas de execução contínua, devido aos limites da API e ao volume de dados





# DICIONÁRIO

- **stars** - int, numérico, número de estrelas
- **forks** - int, numérico, número de forks
- **language** - string, categórico, linguagem principal do repositório
- **subscribers\_count** - int, numérico, número de inscritos
- **owner\_type** - string, categórico, tipo do dono (User ou Organization)



# QUALIDADE DOS DADOS

## Dados Ausentes

- **language** - 1 dado ausente
  - remoção do registro
- **owner\_location** - 3.969 dados ausentes
  - 42% do total - substituição por "Not informed"

# QUALIDADE DOS DADOS

## Dados Duplicados

- 348 registros duplicados
  - C - 347
  - Java - 1
- Exclusão de dados duplicados - 9.101



# QUALIDADE DOS DADOS

## Dados Inconsistentes

- Verificação de valores negativos em colunas numéricas ✓
- Garantindo que colunas de tempo estão em formato dateTime ✓
- Checando valores não numéricos em colunas numéricas ✓
- Verificando valores únicos em colunas categóricas
  - Potencial problema em **owner\_location** ✗



# QUALIDADE DOS DADOS

## Dados Inconsistentes

- **owner\_location** ❌

- Localizações inválidas ou fictícias
  - Variações de nomes para uma mesma localização
- ➡ exclusão da coluna

```
"beijing,china",5  
beijing,4  
hangzhou china,4  
china beijing,3  
"chengdu,china",3
```

```
earth,2
```

```
the internet,2
```


```
localhost:7000,1
```

```
the cloud,2
```

```
"[california, singapore, china]",1
```

# QUALIDADE DOS DADOS

## Dados Inconsistentes

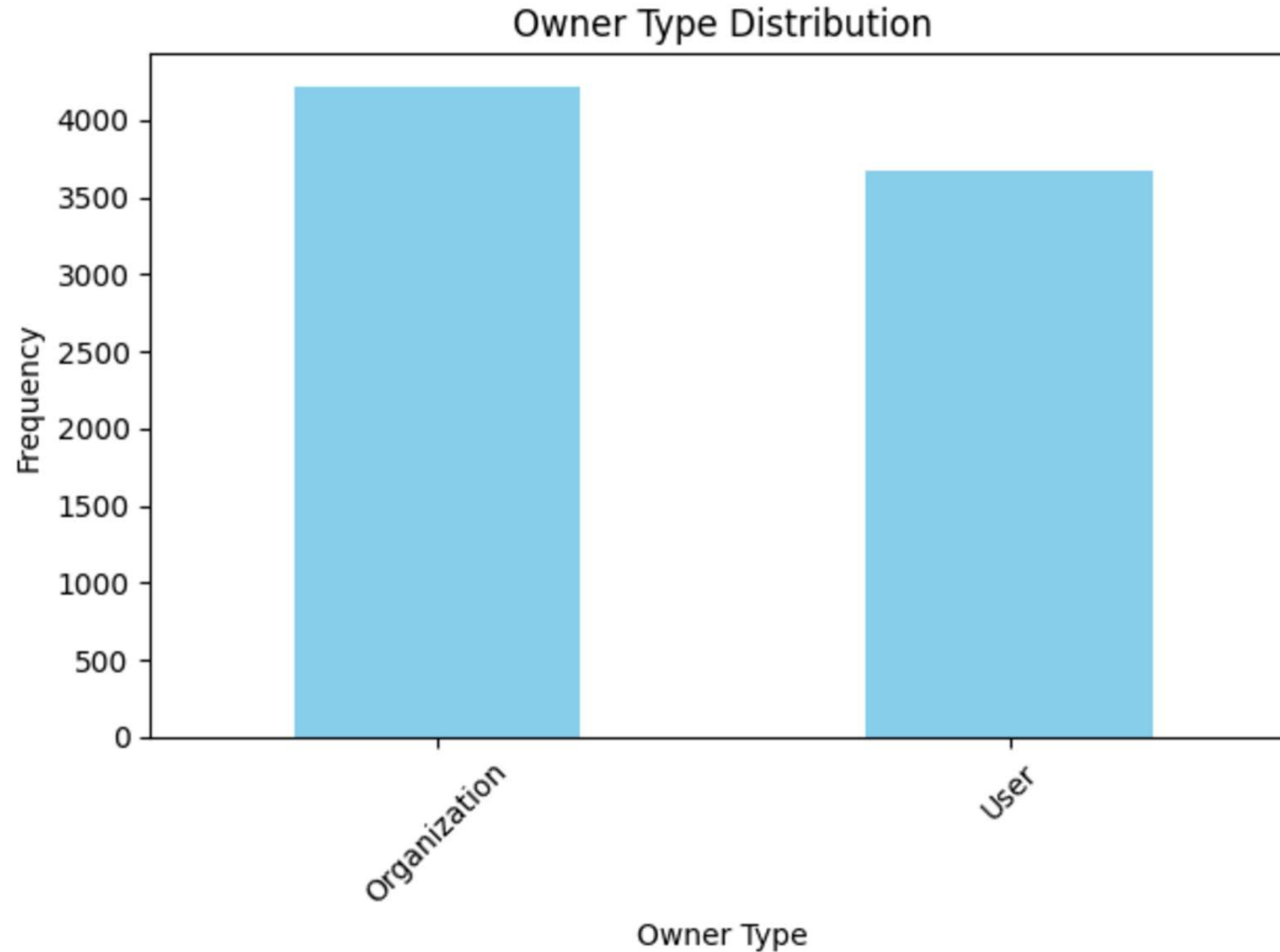
- **watchers\_count** 
  - possui as mesmas informações de **stars**
    - ↪ exclusão da coluna



# ANÁLISE EXPLORATÓRIA

Frequência das Colunas Categóricas

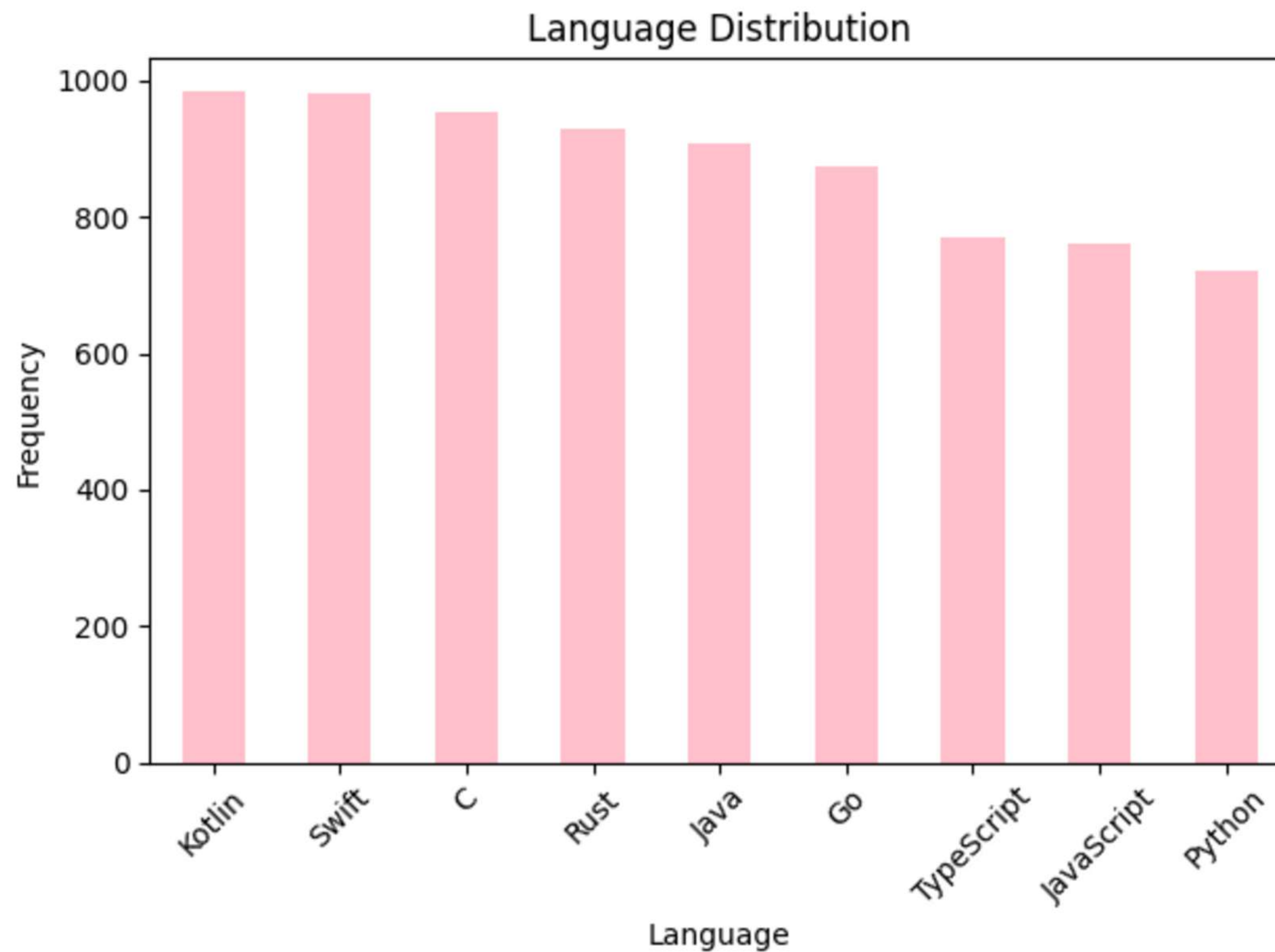
- **owner\_type**



# ANÁLISE EXPLORATÓRIA

Frequência das Colunas Categóricas

- **language**

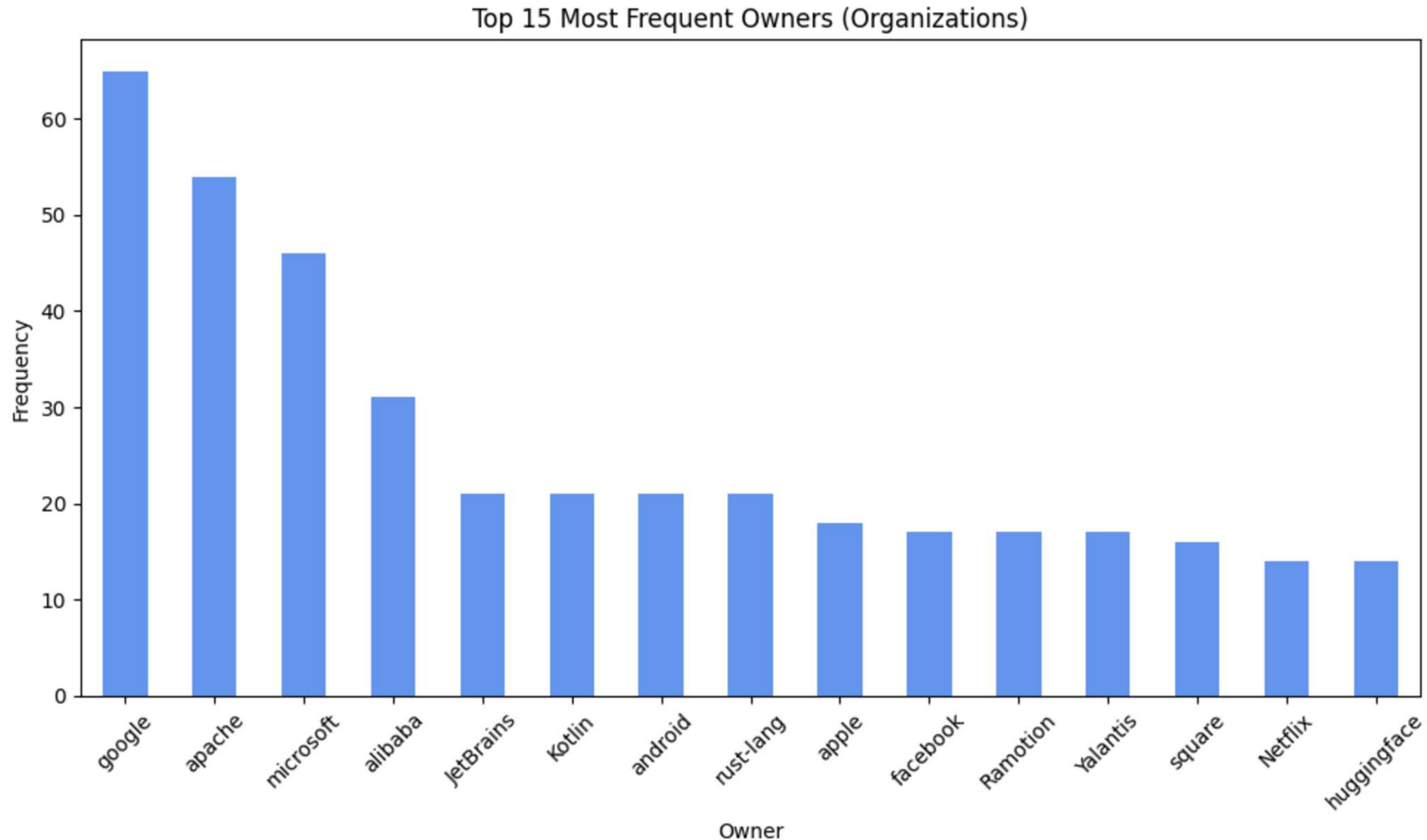




# ANÁLISE EXPLORATÓRIA

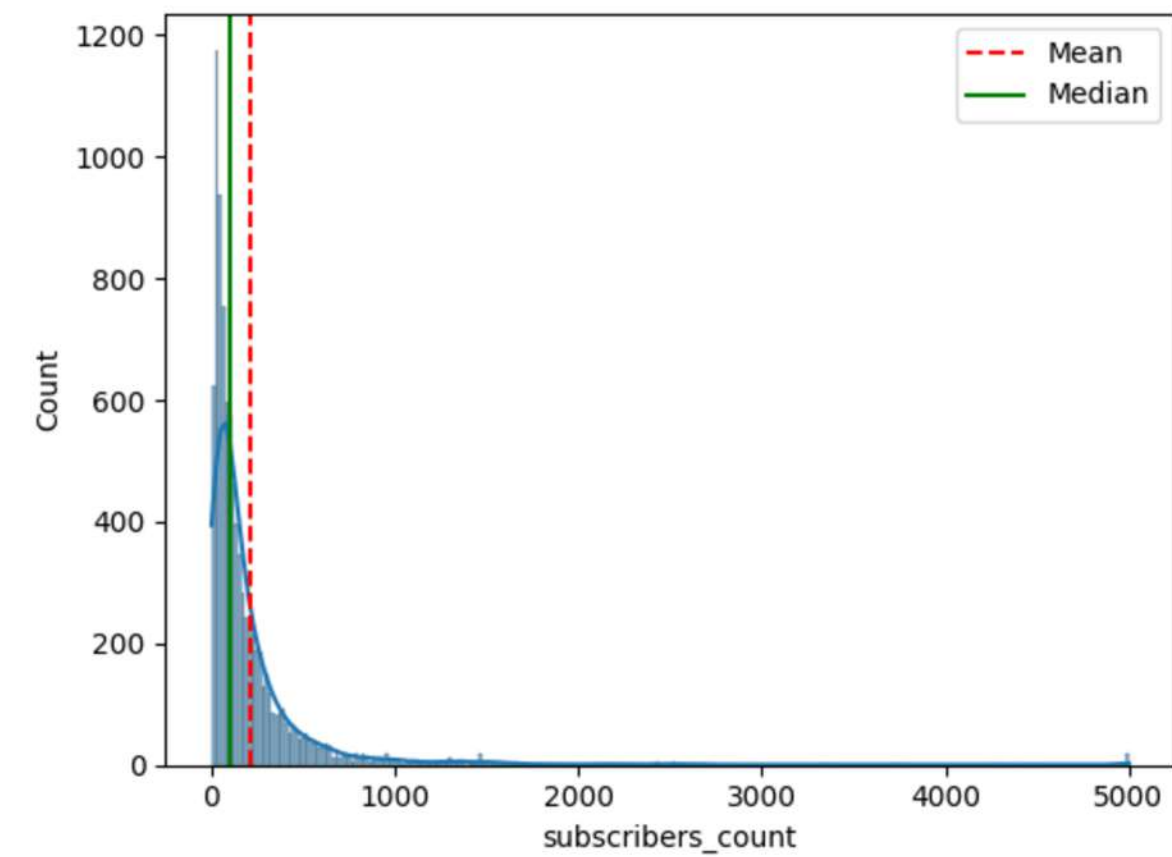
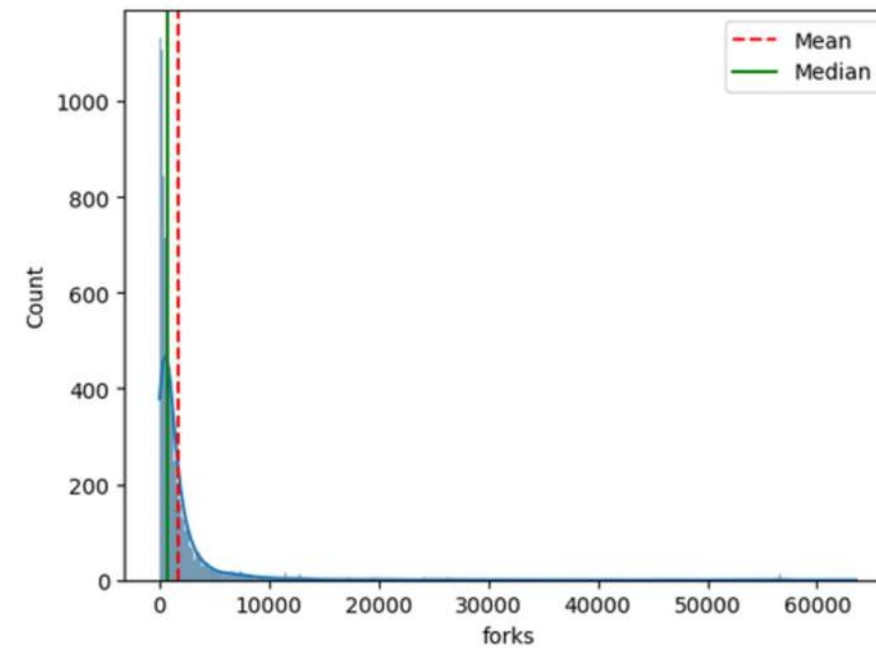
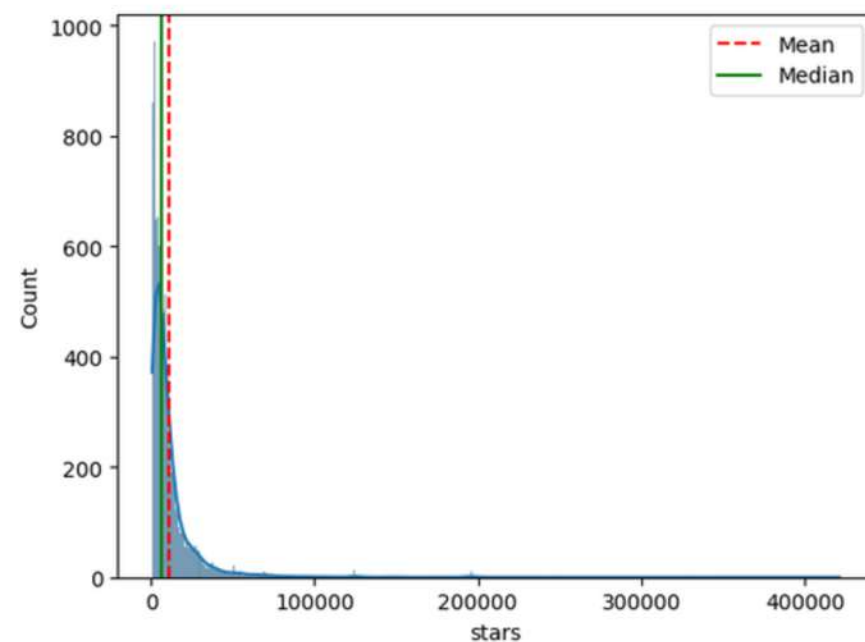
## Frequência das Colunas Categóricas

- **owner** (Organizações)



# ANÁLISE EXPLORATÓRIA

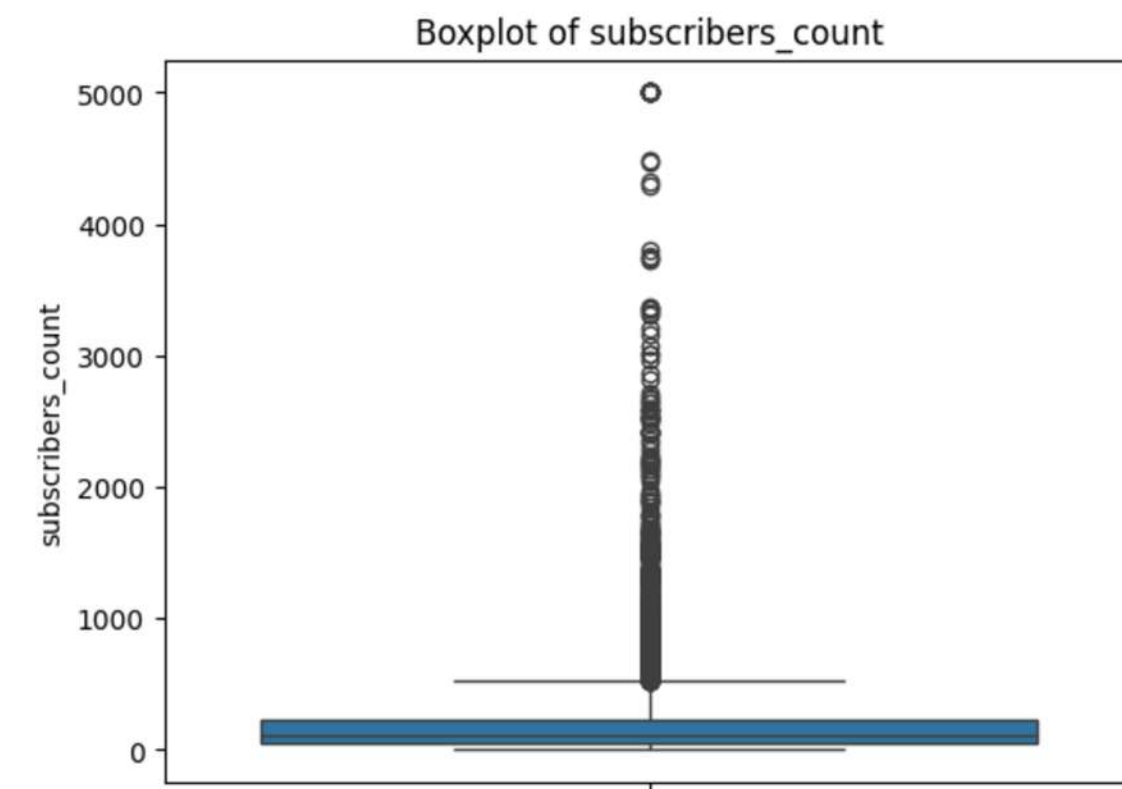
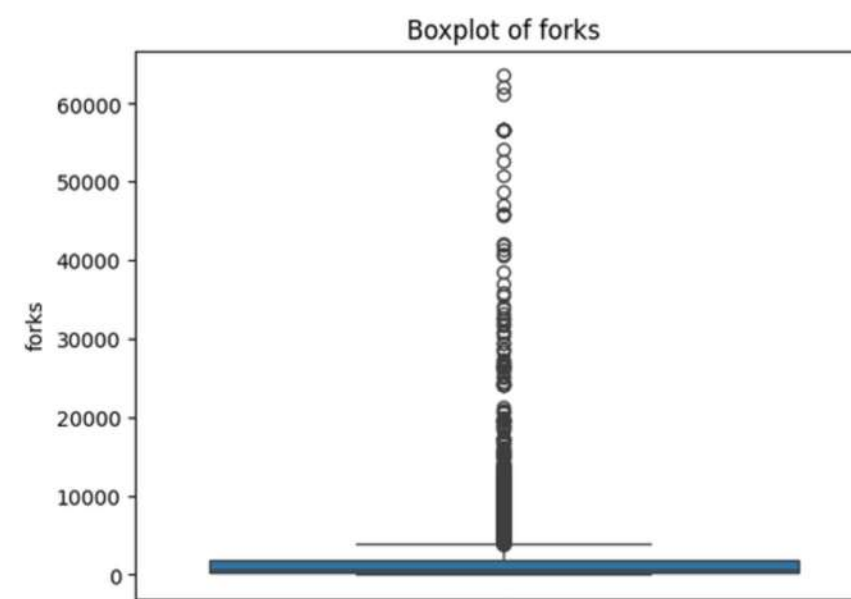
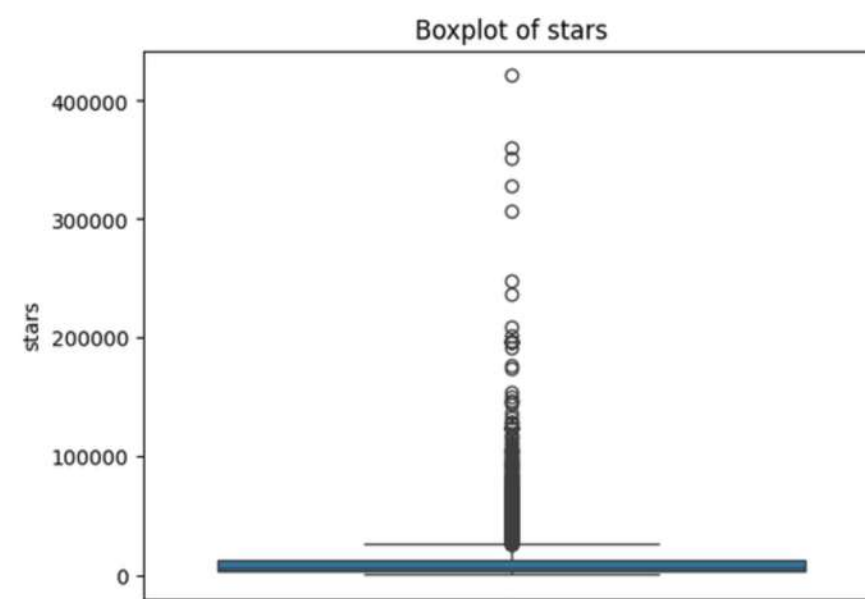
## Estatísticas de posição





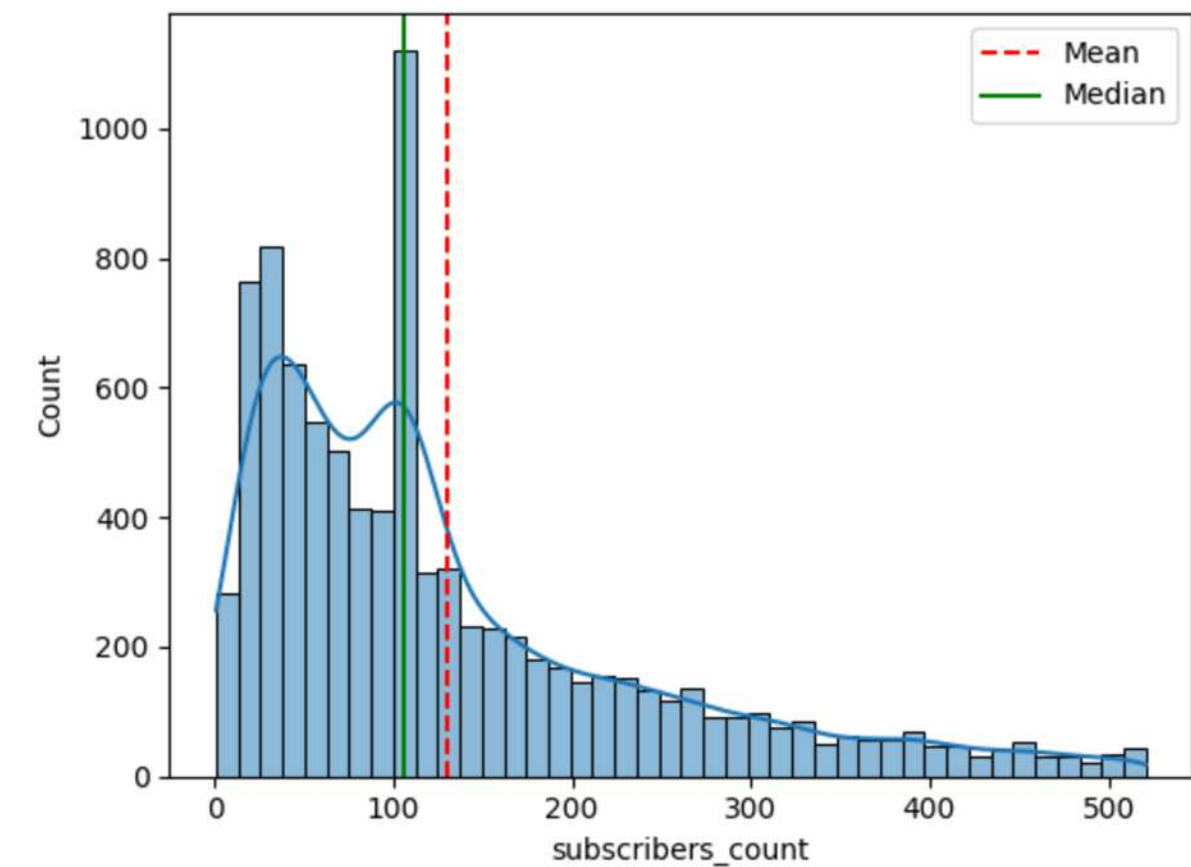
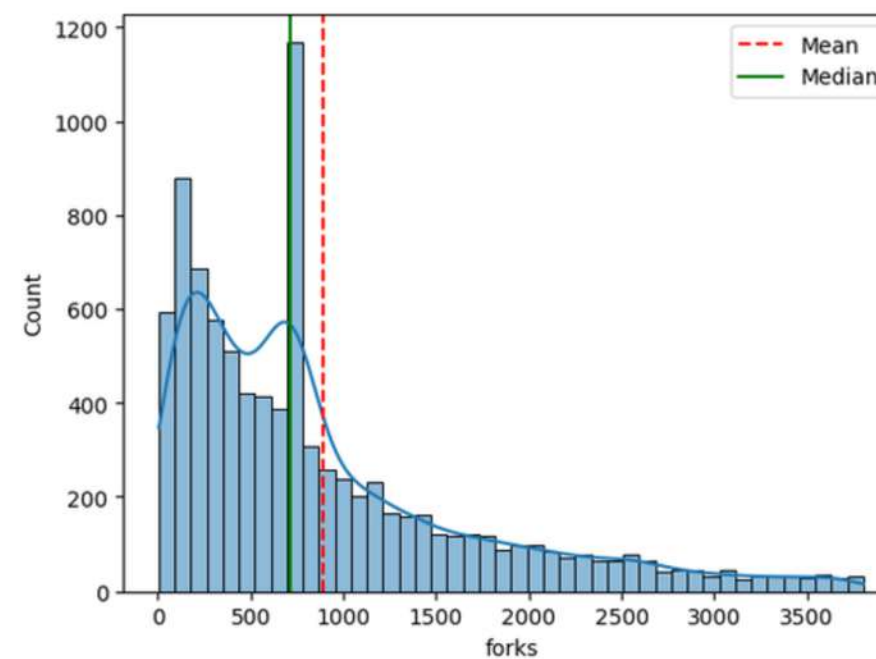
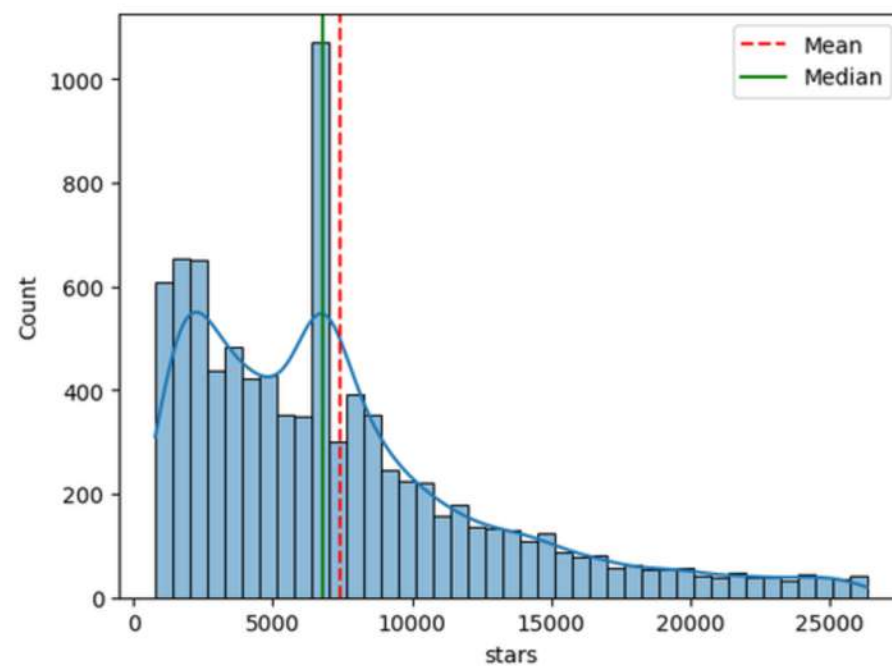
# ANÁLISE EXPLORATÓRIA

## Estatísticas de dispersão



# ANÁLISE EXPLORATÓRIA

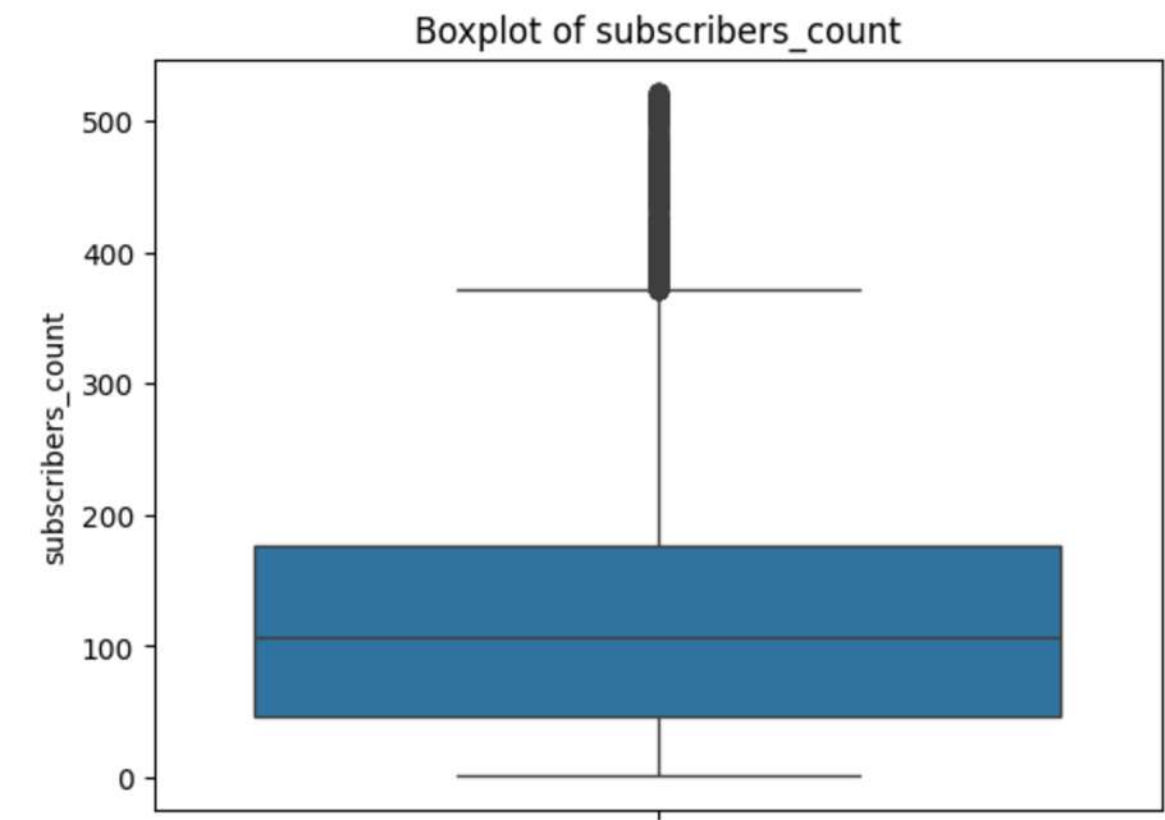
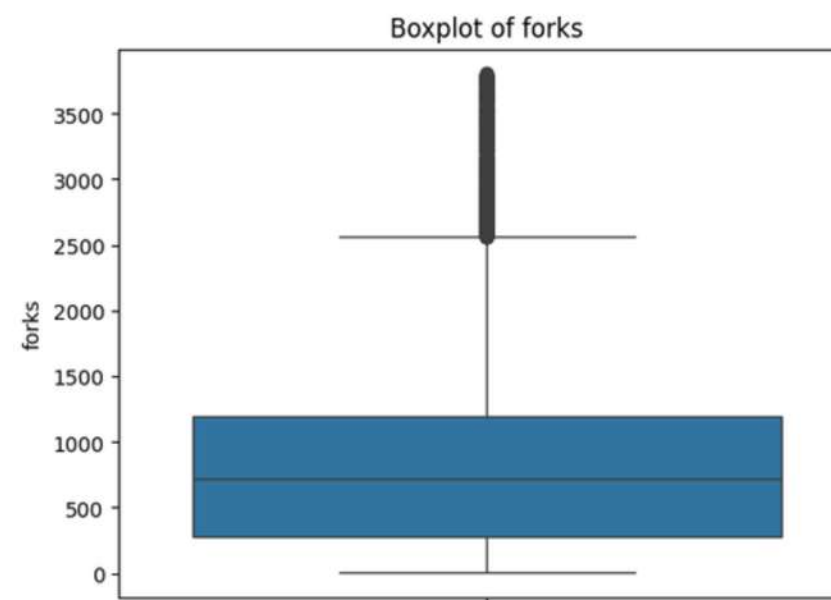
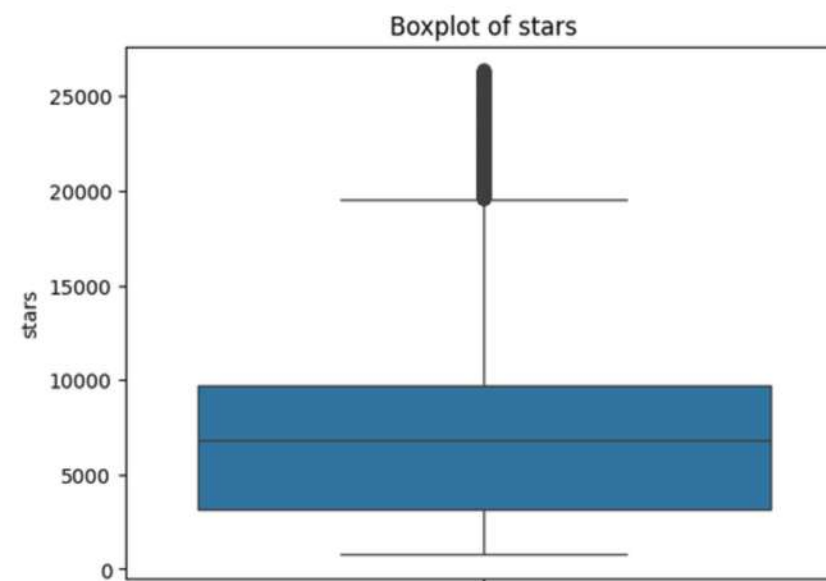
## Estatísticas de posição - Mediana







# ANÁLISE EXPLORATÓRIA

## Estatísticas de dispersão - Mediana

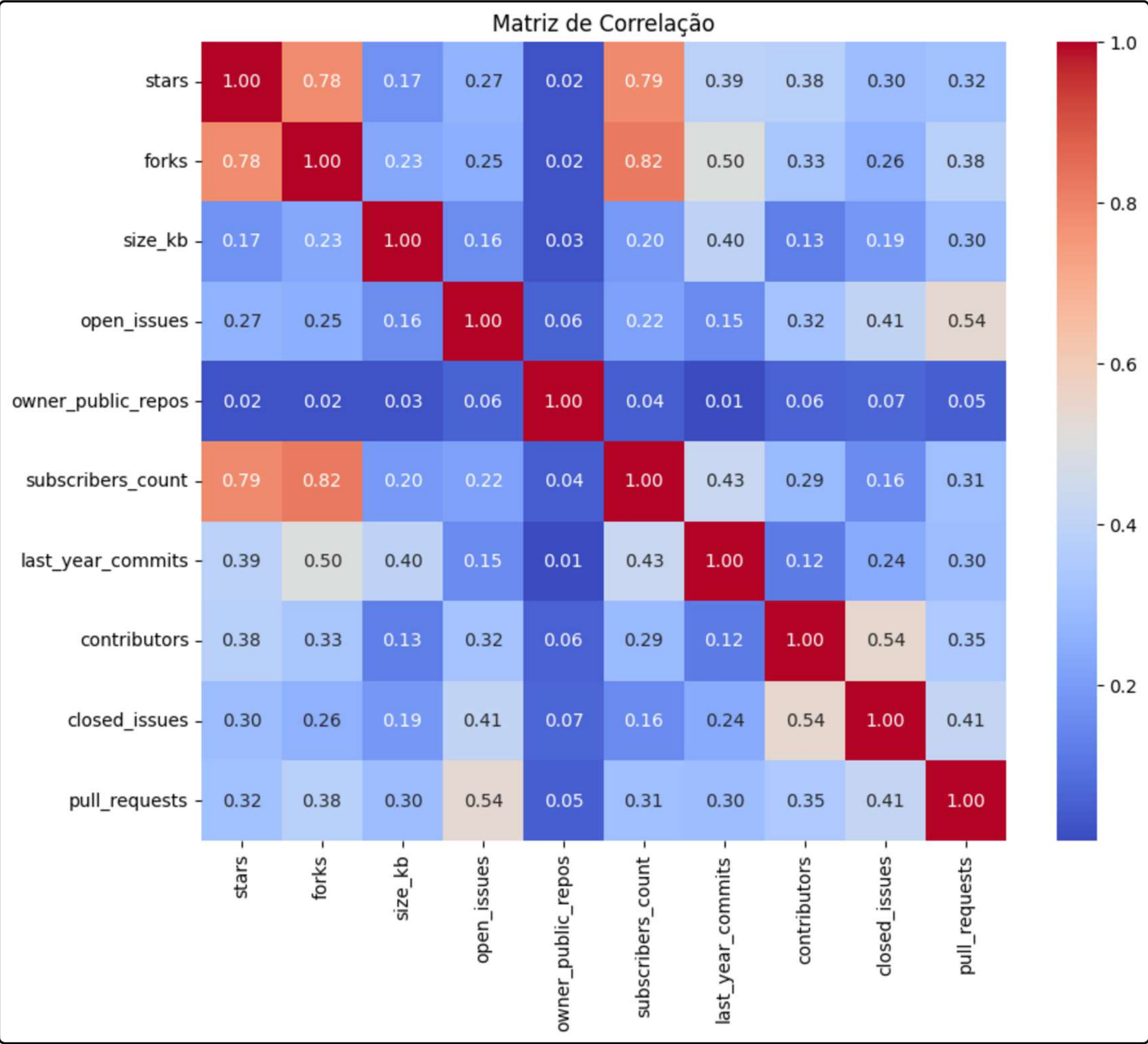


# MODELAGEM DE DADOS

## Problemas alvo

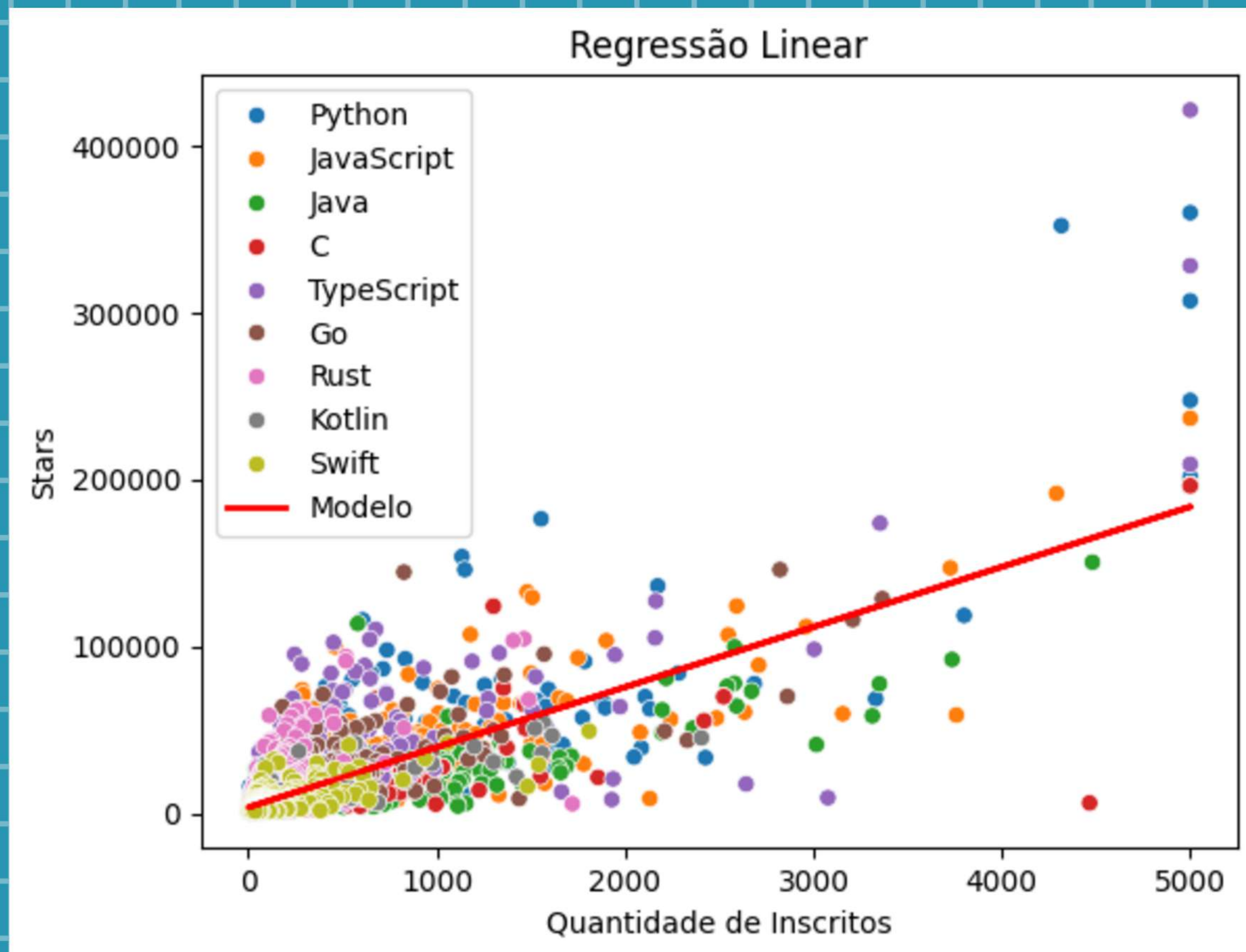
-  Prever a quantidade de estrelas de acordo com a quantidade de inscritos - **Regressão Linear**
-  Prever a popularidade do repositório de acordo com os forks, quantidade de inscritos e quantidade de commits nos últimos meses - **Classificação**

- stars
- forks
- subscribers\_count





## Resultado



# CLASSIFICAÇÃO

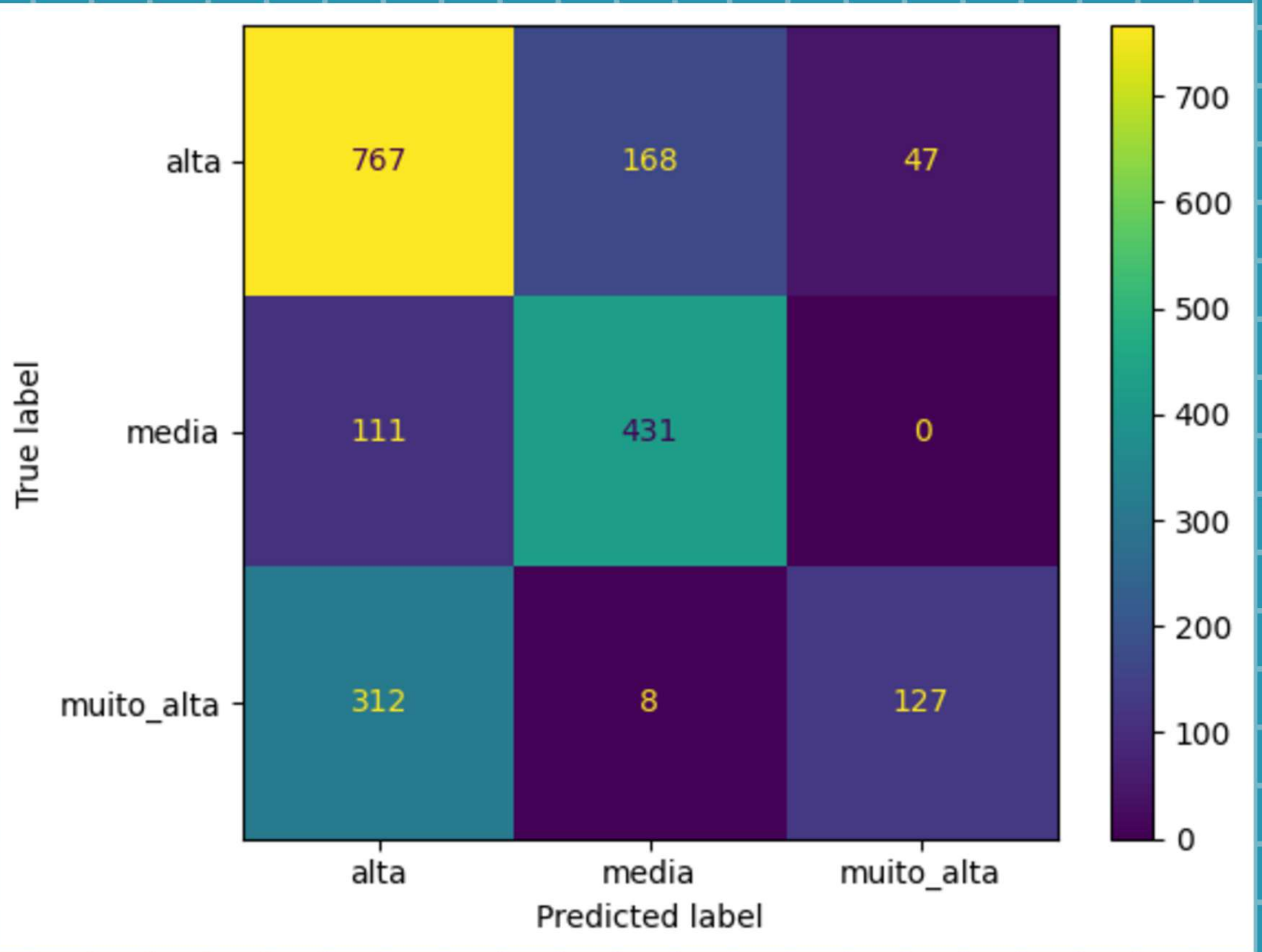
- Redução do dataframe ( até 20.000 estrelas)
- Nova coluna **popularidade**
- Baseada em **stars**
- Níveis:
  - média (770 - 3.000 estrelas)
  - alta (3.000 - 10.000 estrelas)
  - muito alta (+ 10.000 estrelas)
- forks, subscribers\_count, last\_year\_commits





# Resultado

	precision	recall	f1-score
alta	0.64	0.78	0.71
media	0.71	0.80	0.75
muito_alta	0.73	0.28	0.41





**OBRIGADO**