PROJETO 2

UNIVERSIDADE ESTADUAL DA PARAÍBA - UEPB
PRÉ-PROCESSAMENTO DE DADOS
PROFESSOR: RAMON PAIVA SANZ
ALUNOS: AYRYSLAINE KELLE, BRENNO VALE E JEOVÁ ANDERSON
CD - TURMA 2

UEPB

LIMITE DO VISÍVEL

SUMÁRIO

01

UMA EMPRESA QUE OFERECE EMPRÉSTIMOS A PESSOAS FÍSICAS PRECISA SABER, COM BASE NOS DADOS DE SEUS CLIENTES, QUAIS NOVOS CLIENTES PODERÃO OU NÃO ACESSAR O CRÉDITO (EMPRÉSTIMO) SOLICITADO.

Importação das bibliotecas e visualização do Dataframe

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import io
import plotly.express as px
import plotly.graph_objects as go
from sklearn.model_selection import train_test_split
import time
from catboost import CatBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```

	ID	Genero	Casado	Dependentes	Escolaridade	trabalhador autônomo	Renda do requerente	Renda do co-requerente	Valor do empréstimo solicitado/aprovado	Prazo do empréstimo (em meses)	Indicador de histórico de crédito	Zona/área de residência do requerente	Status final do pedido de empréstimo
0	LP001002	Masculino	Não	0	Graduado	Não	5849	0.0	NaN	360.0	1.0	Urbano	Aprovado
1	LP001003	Masculino	Sim	1	Graduado	Não	4583	1508.0	128.0	360.0	1.0	Rural	Não aprovado
2	LP001005	Masculino	Sim	0	Graduado	Sim	3000	0.0	66.0	360.0	1.0	Urbano	Aprovado
3	LP001006	Masculino	Sim	0	Não Graduado	Não	2583	2358.0	120.0	360.0	1.0	Urbano	Aprovado
4	LP001008	Masculino	Não	0	Graduado	Não	6000	0.0	141.0	360.0	1.0	Urbano	Aprovado
...	...	...	...	...	...	...	...	...	...	...	...	...	...
609	LP002978	Feminino	Não	0	Graduado	Não	2900	0.0	71.0	360.0	1.0	Rural	Aprovado
610	LP002979	Masculino	Sim	3+	Graduado	Não	4106	0.0	40.0	180.0	1.0	Rural	Aprovado
611	LP002983	Masculino	Sim	1	Graduado	Não	8072	240.0	253.0	360.0	1.0	Urbano	Aprovado
612	LP002984	Masculino	Sim	2	Graduado	Não	7583	0.0	187.0	360.0	1.0	Urbano	Aprovado
613	LP002990	Feminino	Não	0	Graduado	Sim	4583	0.0	133.0	360.0	0.0	Semiurbano	Não aprovado

614 rows × 13 columns

PADRONIZANDO OS DADOS

01

```
df = df.rename(columns={"Loan_ID":"ID",
          "Gender":"Genero",
          "Married":"Casado",
          "Dependents":"Dependentes",
          "Education":"Escolaridade",
          "Self_Employed":"trabalhador autônomo",
          "ApplicantIncome":"Renda do requerente",
          "CoapplicantIncome":"Renda do co-requerente",
          "LoanAmount":"Valor do empréstimo solicitado/aprovado",
          "Loan_Amount_Term":"Prazo do empréstimo (em meses)",
          "Credit_History":"Indicador de histórico de crédito",
          "Property_Area":"Zona/área de residência do requerente",
          "Loan_Status":"Status final do pedido de empréstimo"
})
```

02

```
df = df.replace({"Male":"Masculino",
                 "Female":"Feminino",
                 "No":"Não",
                 "Yes":"Sim",
                 "Graduate":"Graduado",
                 "Not Graduate":"Não Graduado",
                 "Urban":"Urbano",
                 "Semiurban":"Semiurbano",
                 "Y":"Aprovado",
                 "N":"Não aprovado"
})
```

DIAGNÓSTICO DO BANCO DE DADOS

```python
def diagnostico(df):
  print(f"--- Diagnóstico do Banco de dados---")
  print("\nValores nulos por coluna:\n")
  display(df.isnull().sum())
  print("\nDuplicados:", df.duplicated().sum())

diagnostico(df)
```

```
--- Diagnóstico do Banco de dados---

Valores nulos por coluna:
```

	0
ID	0
Genero	0
Casado	0
Dependentes	15
Escolaridade	0
trabalhador autônomo	0
Renda do requerente	0
Renda do co-requerente	0
Valor do empréstimo solicitado/aprovado	22
Prazo do empréstimo (em meses)	14
Indicador de histórico de crédito	50
Zona/área de residência do requerente	0
Status final do pedido de empréstimo	0

```
dtype: int64

Duplicados: 0
```

MÉTRICAS ESTATÍSTICAS

```
df.describe()
```

	Renda do requerente	Renda do co-requerente	Valor do empréstimo solicitado/aprovado	Prazo do empréstimo (em meses)	Indicador de histórico de crédito
count	614.000000	614.000000	614.000000	614.000000	614.000000
mean	5403.459283	1621.245798	146.412162	342.000000	0.855049
std	6109.041673	2926.248369	84.037468	64.372489	0.352339
min	150.000000	0.000000	9.000000	12.000000	0.000000
25%	2877.500000	0.000000	100.250000	360.000000	1.000000
50%	3812.500000	1188.500000	129.000000	360.000000	1.000000
75%	5795.000000	2297.250000	164.750000	360.000000	1.000000
max	81000.000000	41667.000000	700.000000	480.000000	1.000000

```
df.describe(include=[object])
```

	ID	Genero	Casado	Dependentes	Escolaridade	trabalhador autônomo	Zona/área de residência do requerente	Status final do pedido de empréstimo
count	614	614	614	614	614	614	614	614
unique	614	3	3	4	2	3	3	2
top	LP002990	Masculino	Sim	0	Graduado	Não	Semiurbano	Aprovado
freq	1	489	398	360	480	500	233	422

VARIÁVEIS CATEGÓRICAS E NUMÉRICAS

Variáveis categóricas

01

```
def dados(df):
    for coluna in ["Genero","Casado", "trabalhador autônomo"]:
        df[coluna] = df[coluna].fillna("Não informado")
    return df

df = dados(df)
```

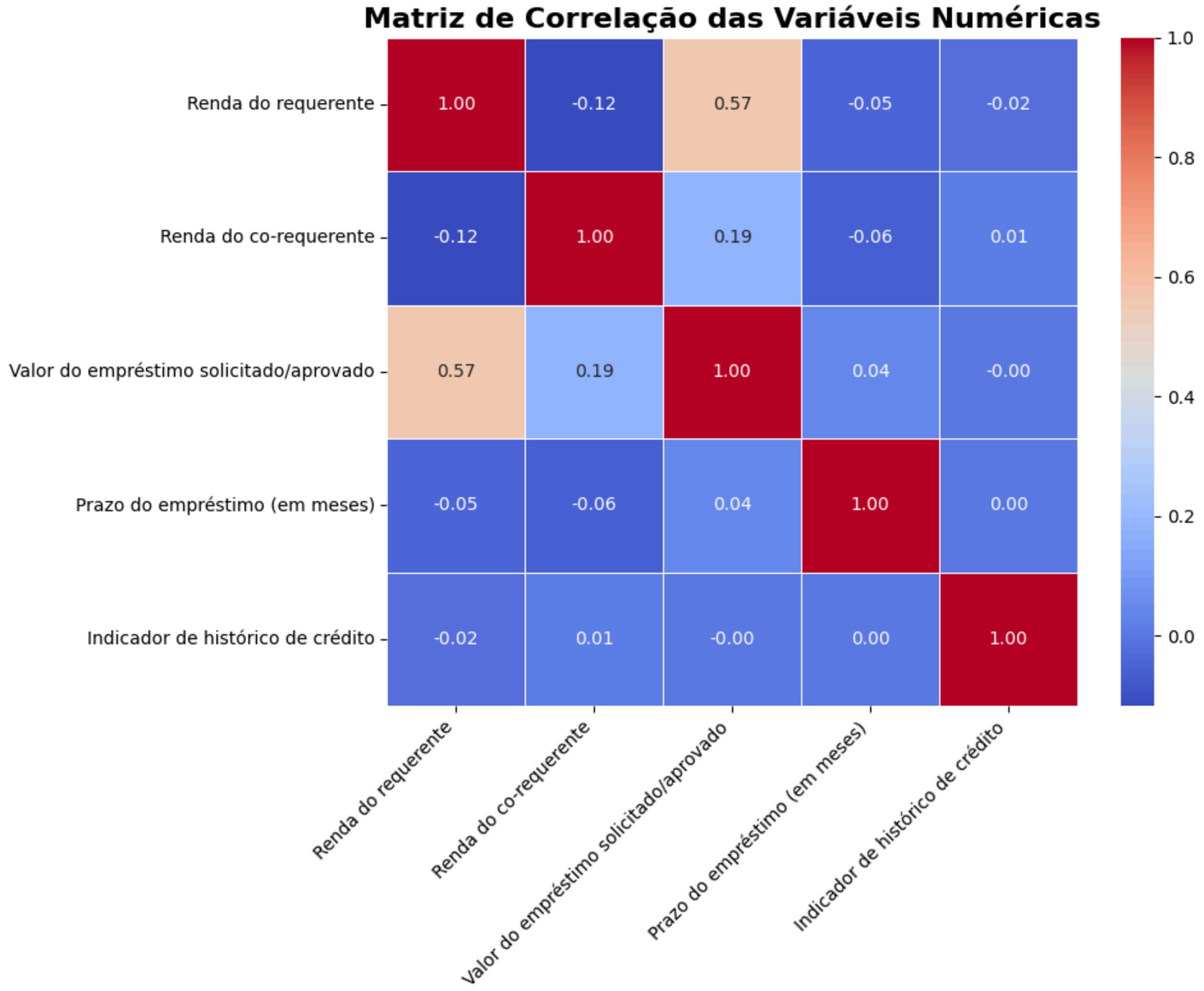
Variáveis numéricas

02

```
def media(df,coluna):
  df[coluna] = df[coluna].fillna(df[coluna].mean())

media(df, "Prazo do empréstimo (em meses)")
media(df, "Valor do empréstimo solicitado/aprovado")
```

```
def moda(df,coluna):
  df[coluna] = df[coluna].fillna(df[coluna].mode()[0])

moda(df, "Indicador de histórico de crédito")
moda(df, "Dependentes")
```

MATRIZ DE CORRELAÇÃO

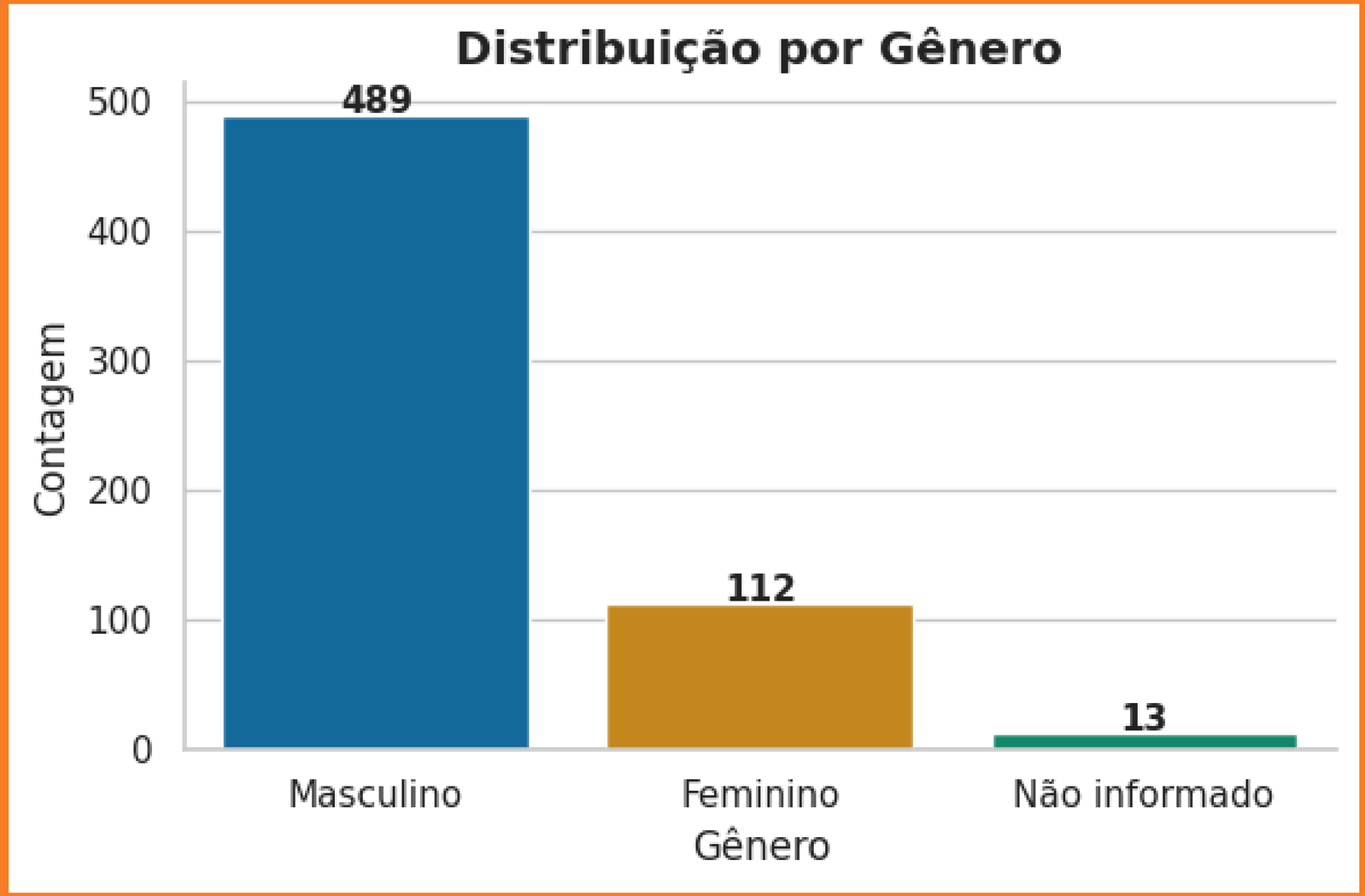


DISTRIBUIÇÃO POR GÊNERO

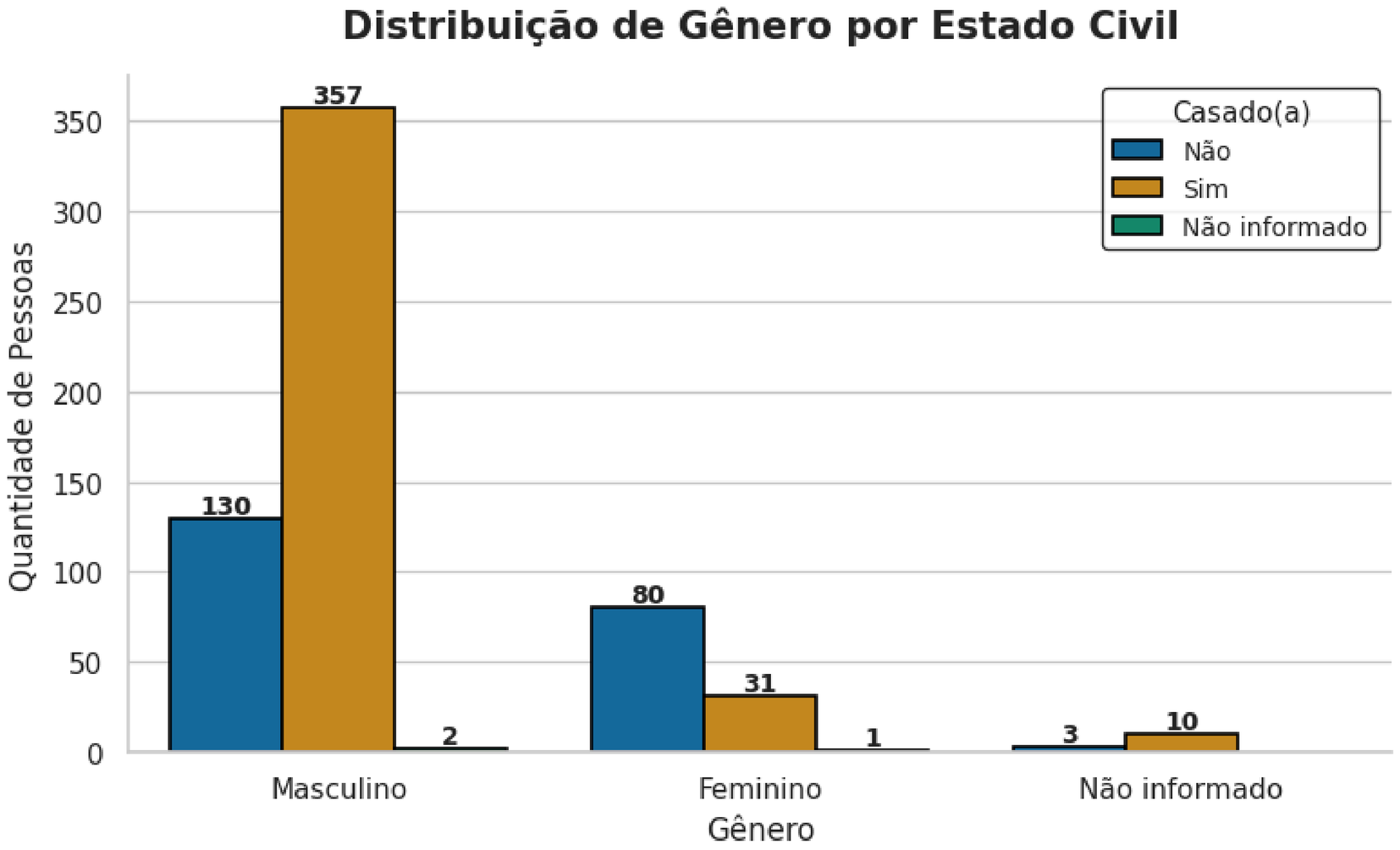
79,64%

18.24%

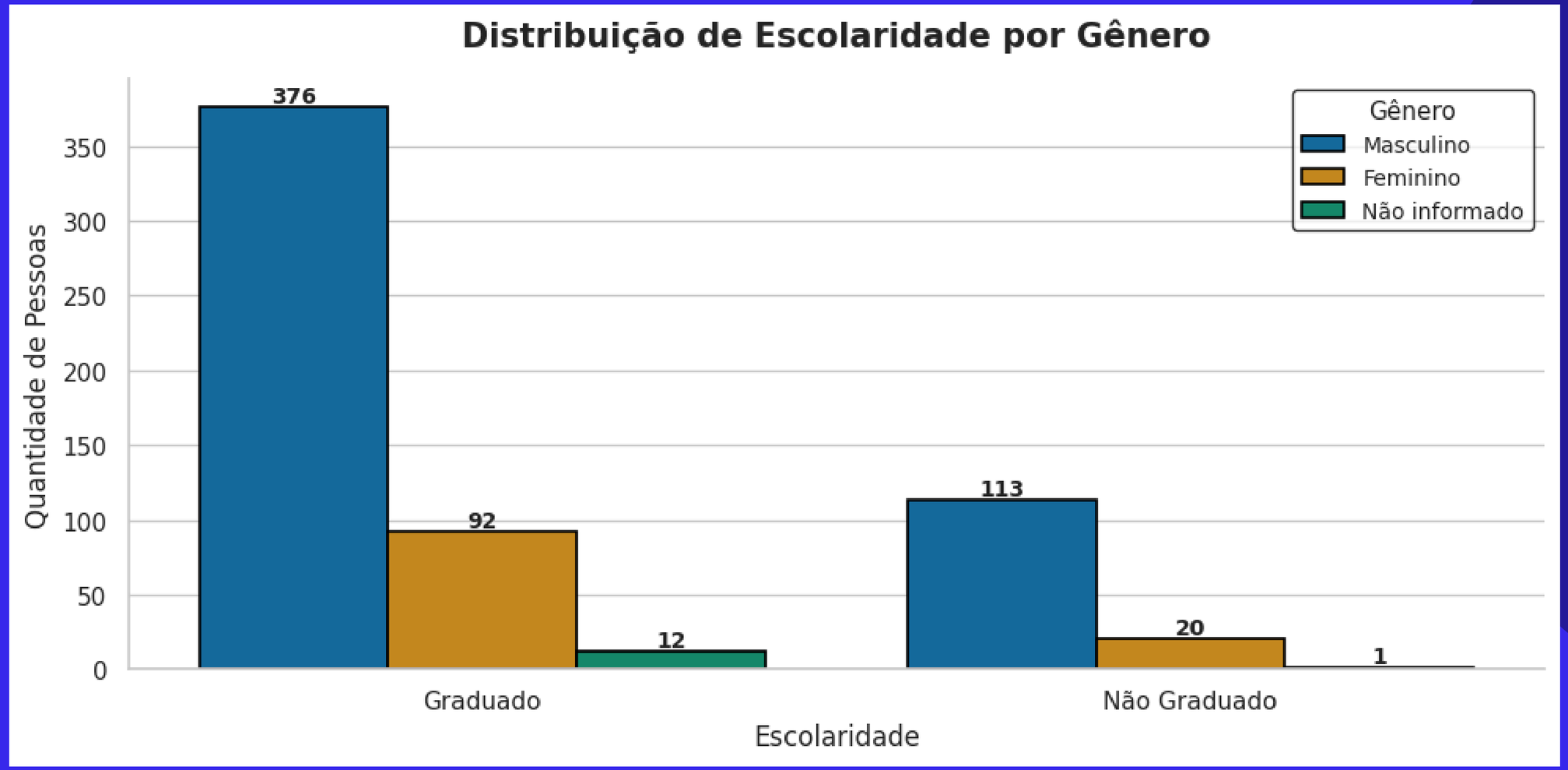
2.12%



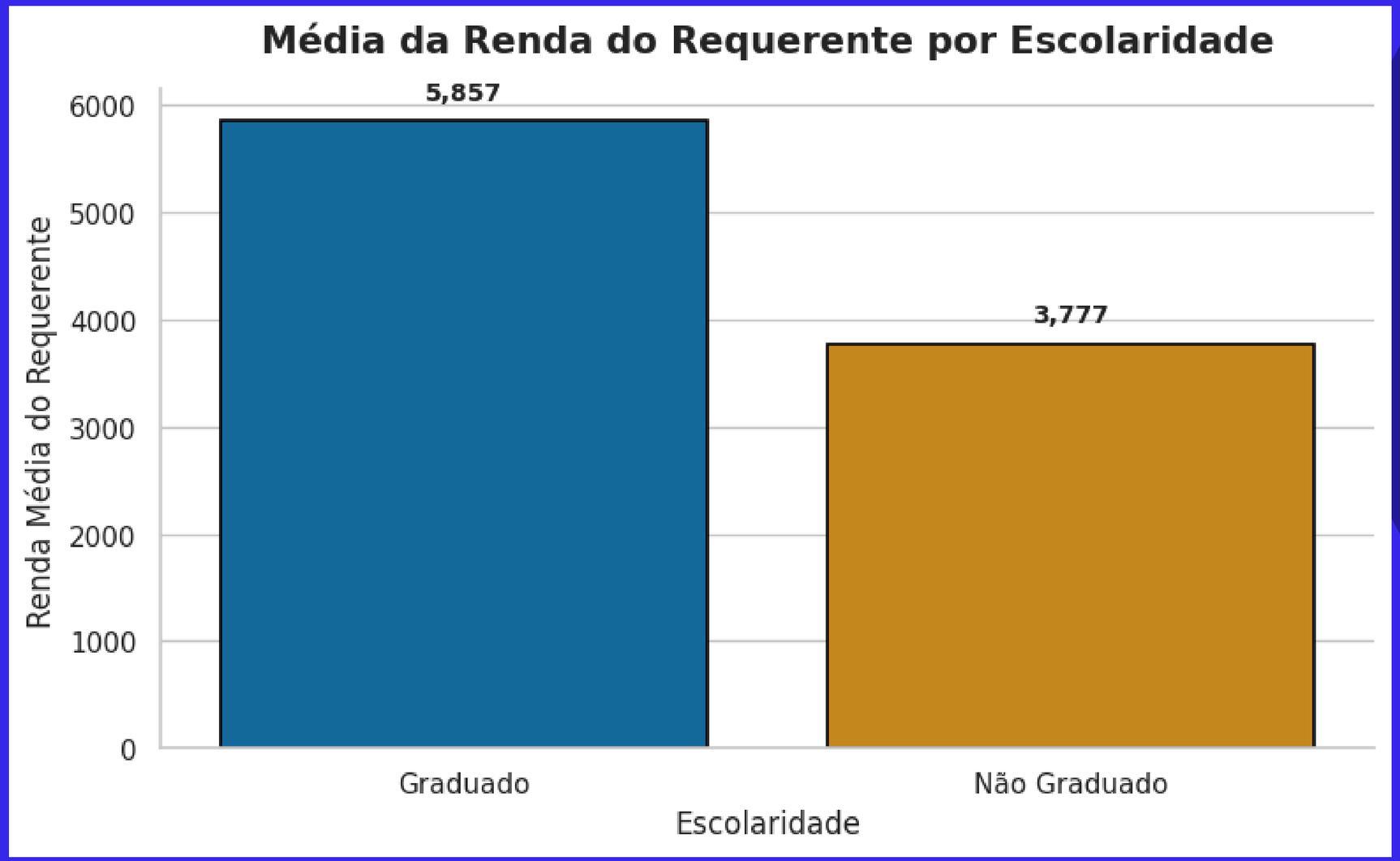
DISTRIBUIÇÃO DE GÊNERO POR ESTADO CIVIL



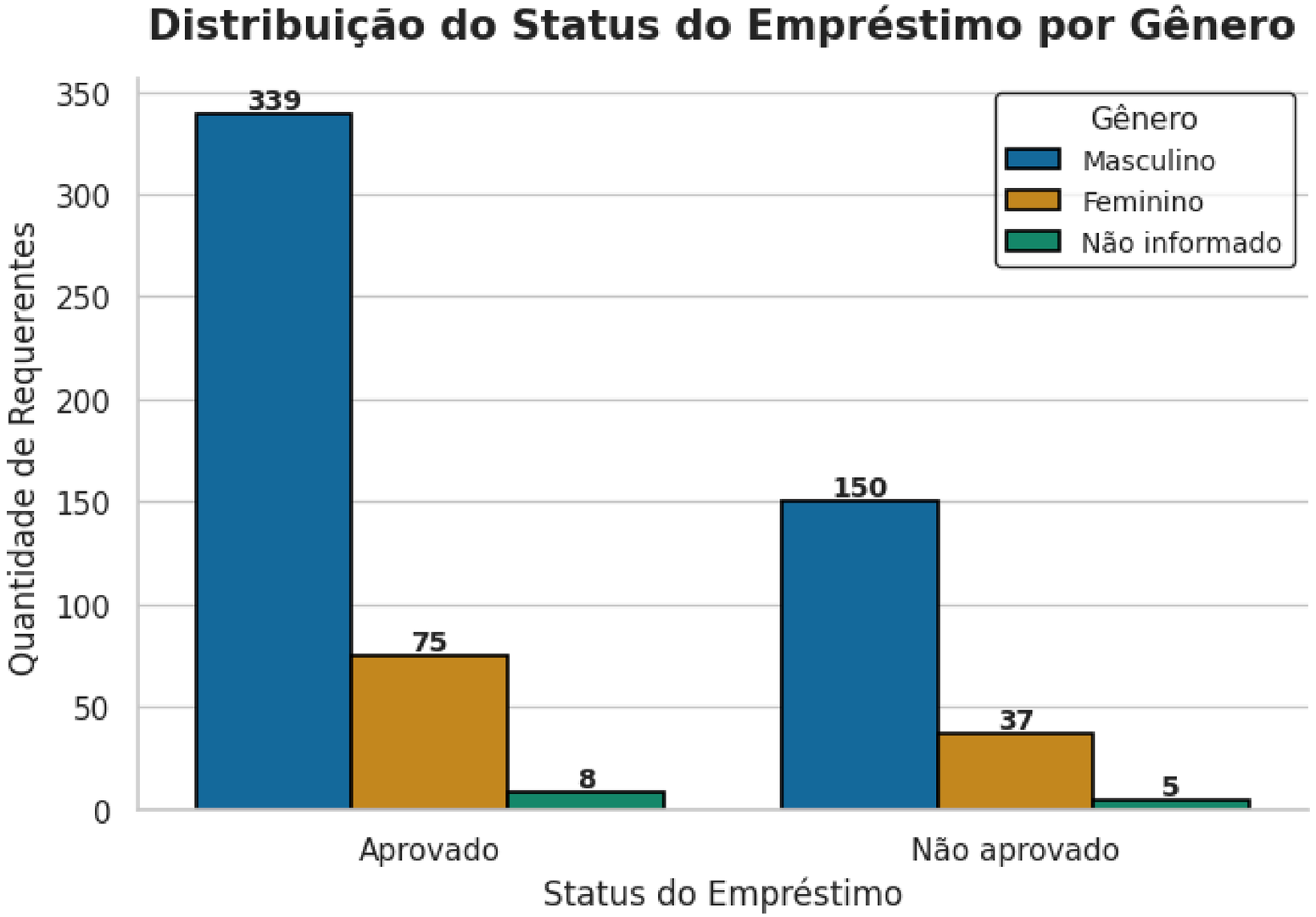
DISTRIBUIÇÃO DE ESCOLARIDADE POR GÊNERO



MÉDIA DA RENDA DO REQUERENTE POR ESCOLARIDADE



STATUS DO EMPRÉSTIMO POR GÊNERO



01

Status: Aprovado (Total: 422)

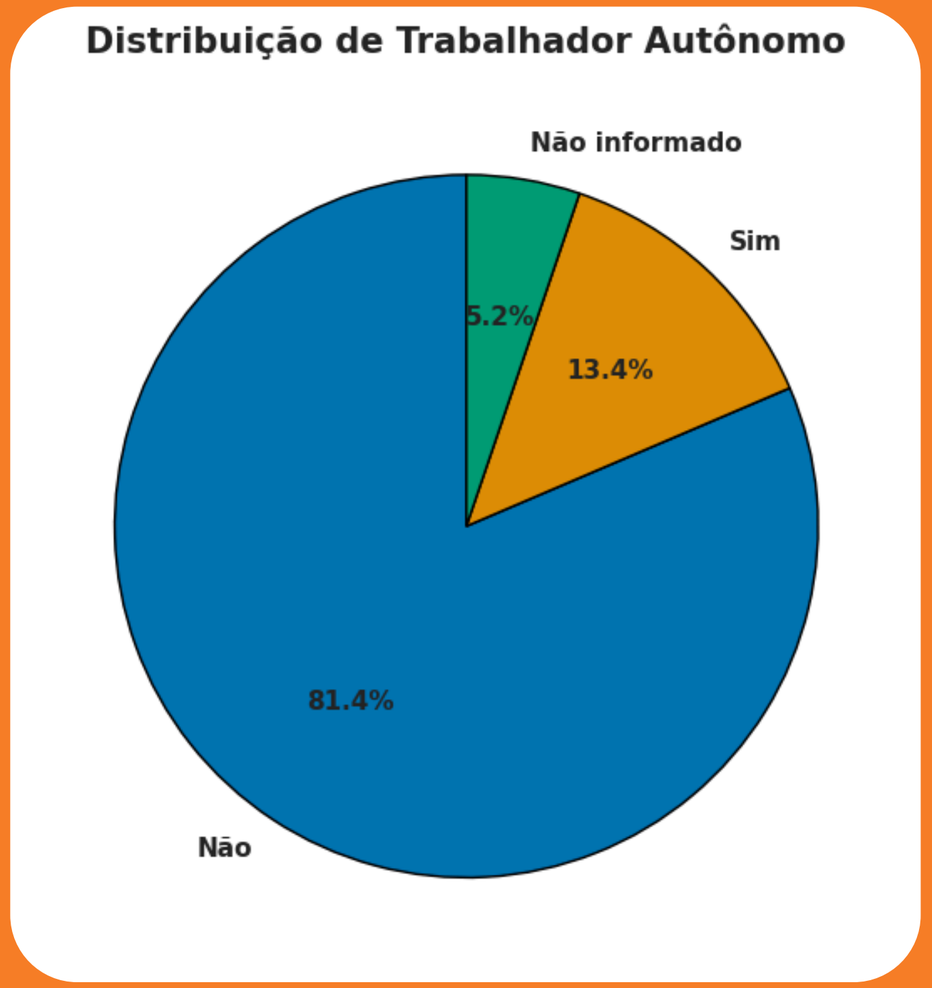
- **Masculino: (339 / 422) = 80,3%**
- **Feminino: (75 / 422) = 17,8%**
- **Não informado: (8 / 422) = 1,9%**

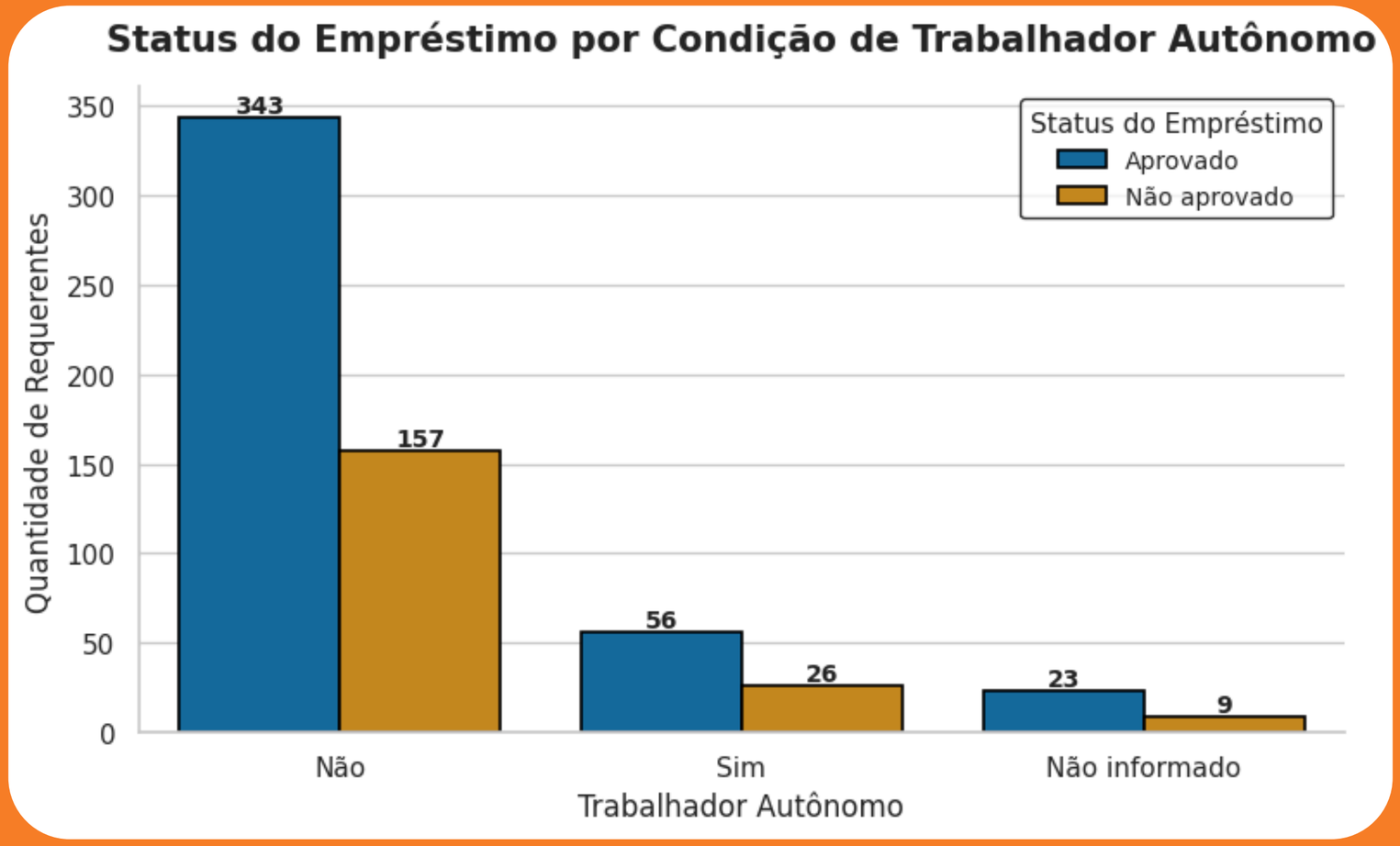
02

Status: Não Aprovado (Total: 192)

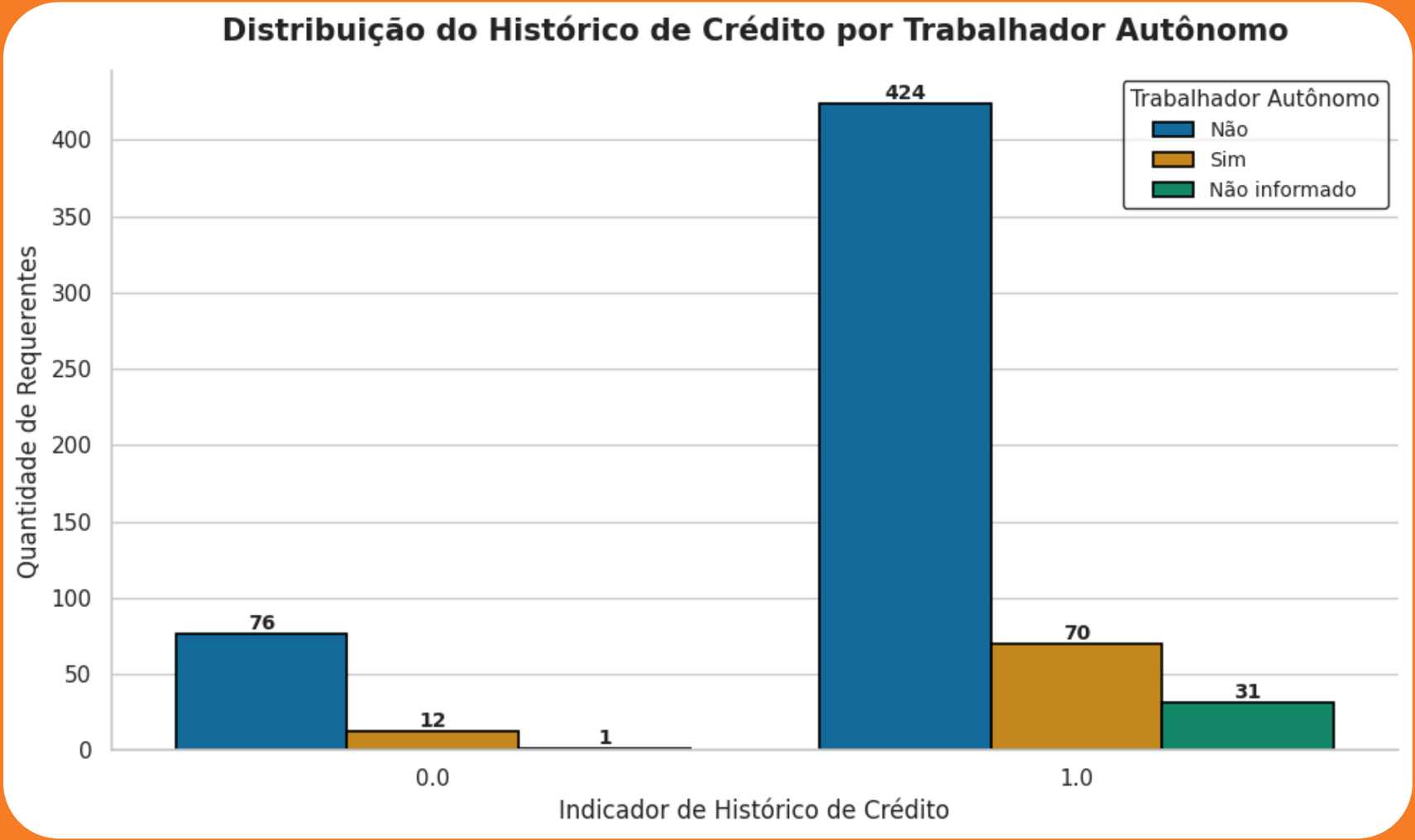
- **Masculino: (150 / 192) = 78,1%**
- **Feminino: (37 / 192) = 19,3%**
- **Não informado: (5 / 192) = 2,6%**

DISTRIBUIÇÃO DE TRABALHADORES AUTÔNOMOS

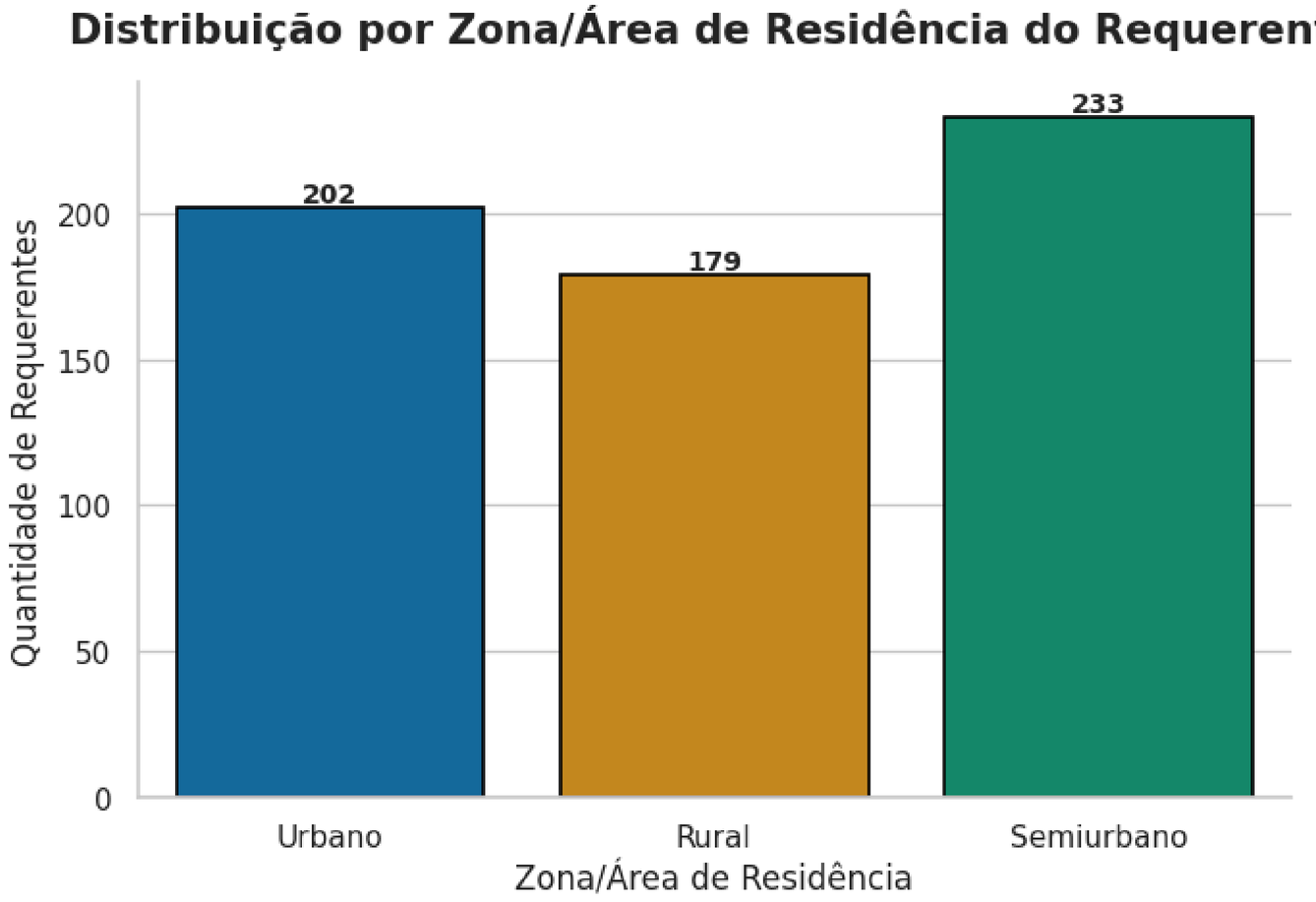




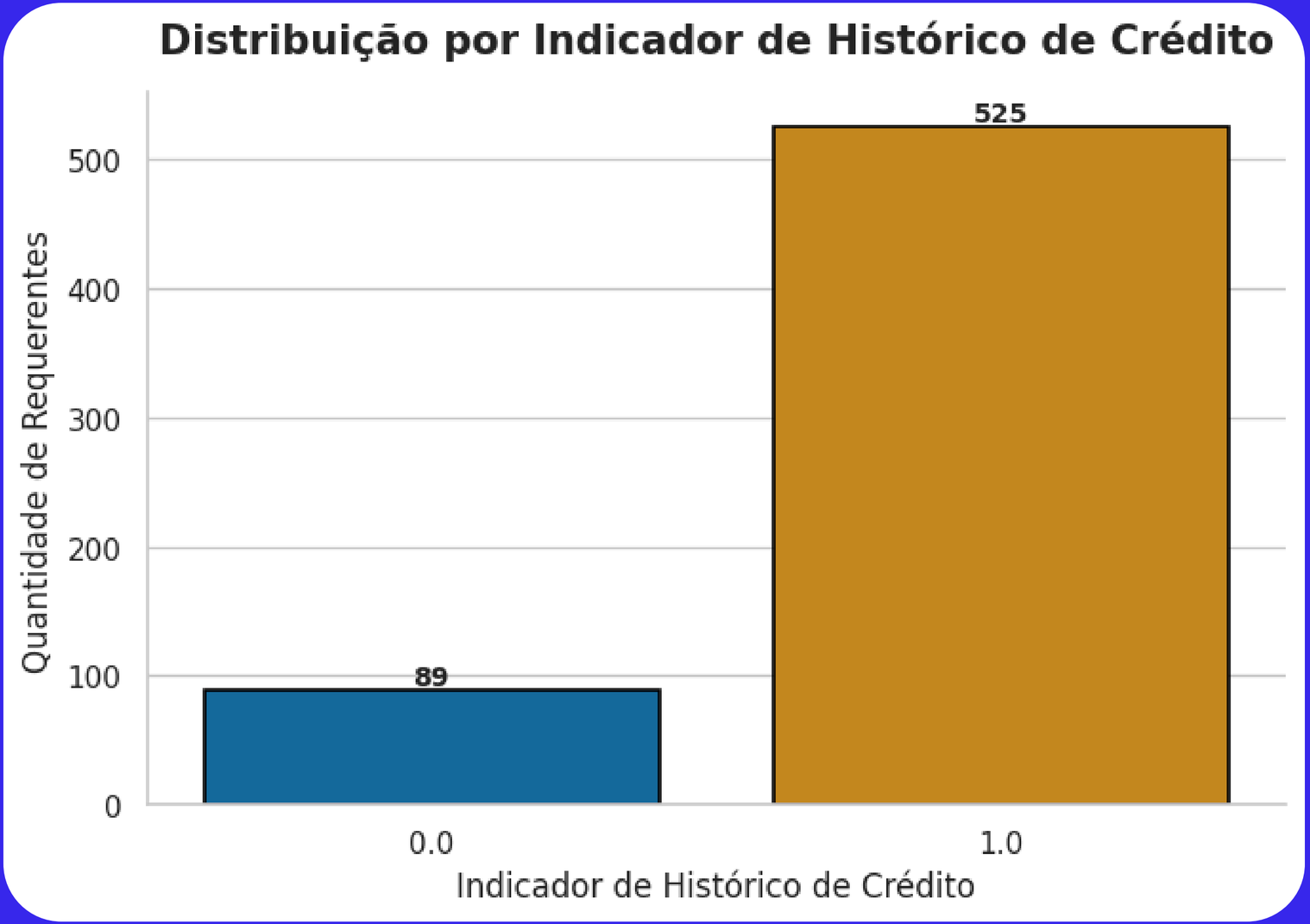
DISTRIBUIÇÃO DE TRABALHADORES AUTÔNOMOS



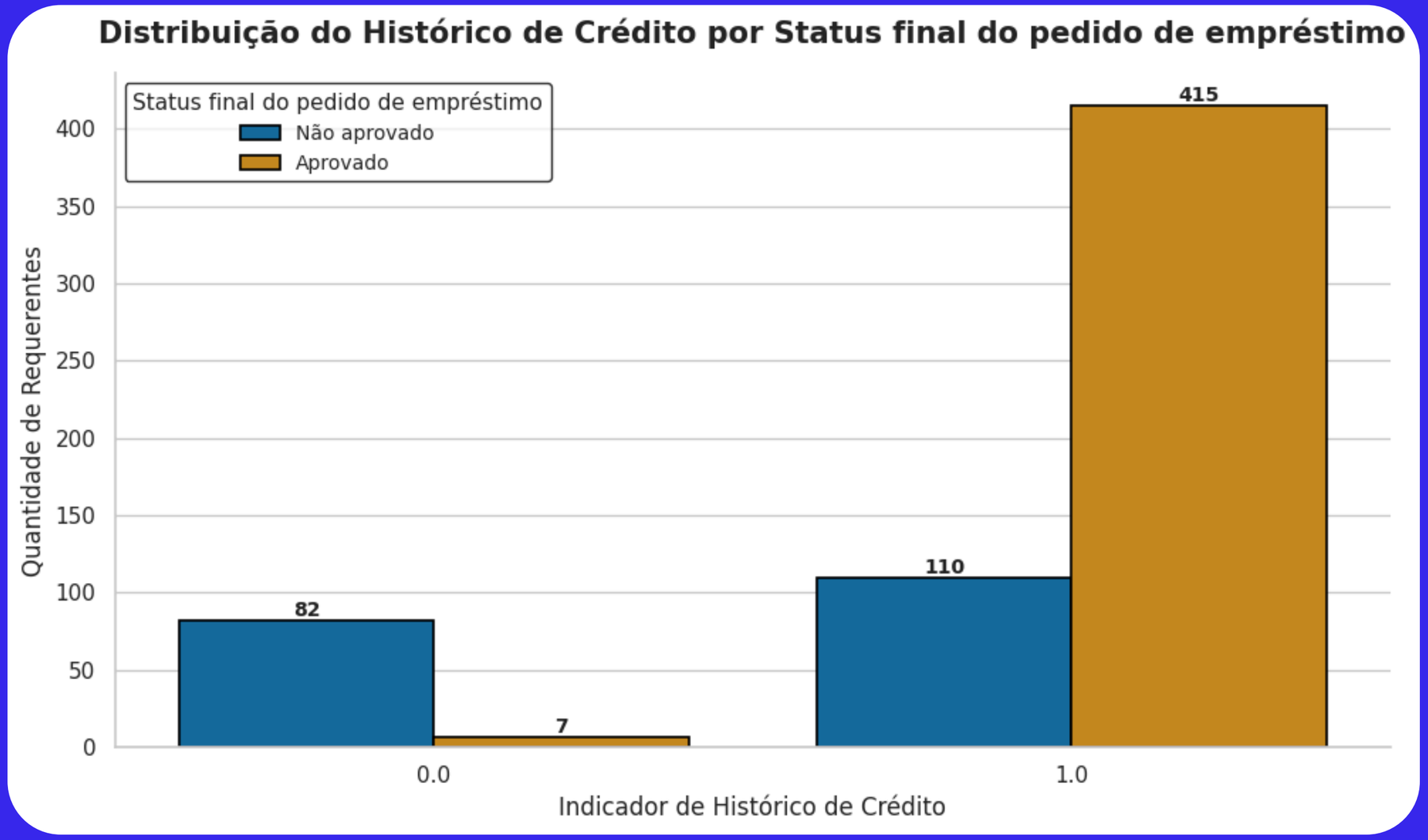
ZONA/ÁREA DE RESIDÊNCIA DO REQUERENTE



Indicador de histórico de crédito



Indicador de histórico de crédito



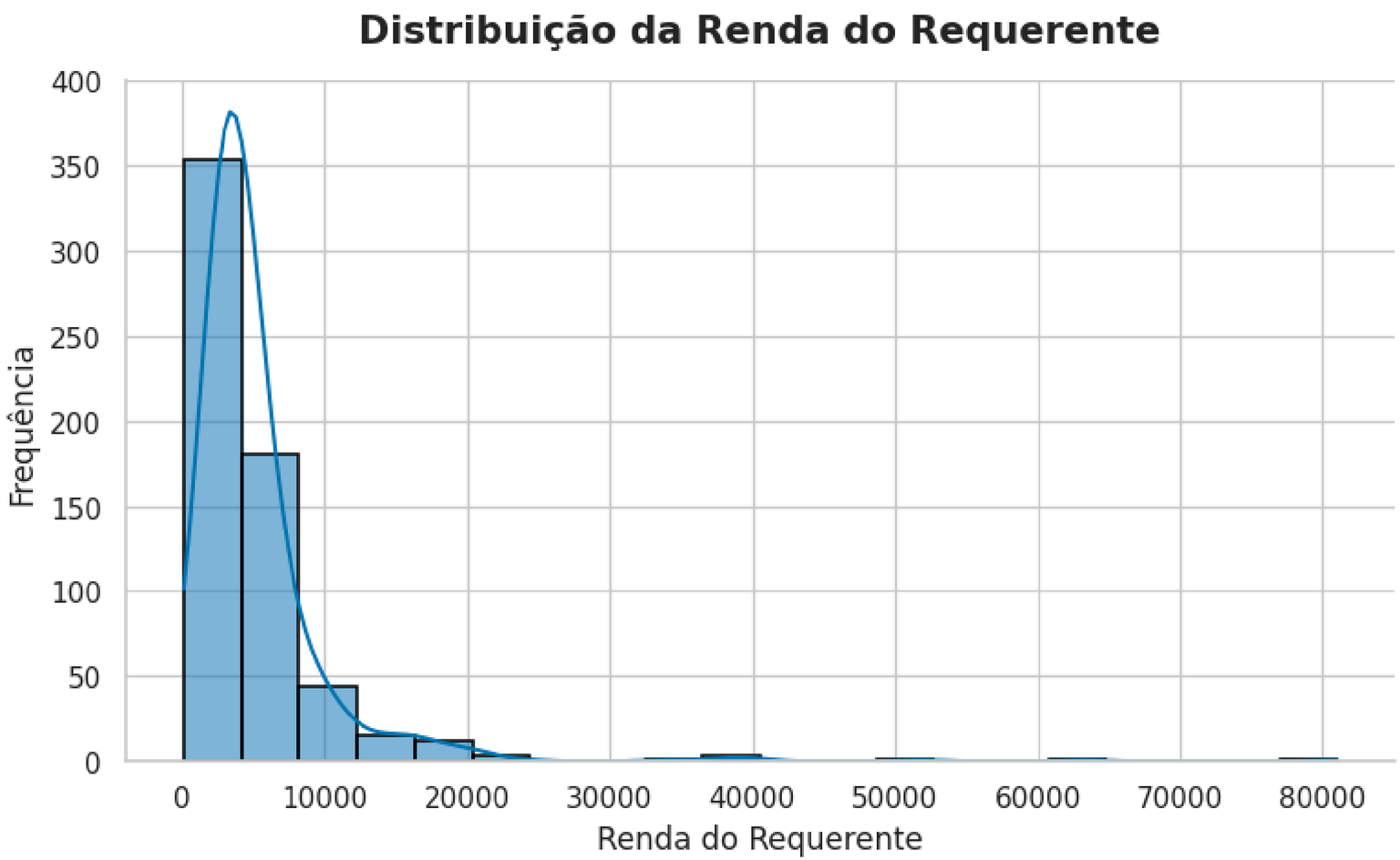
01 **Histórico de Crédito Inadequado (0.0)**

- **Total de Requerentes com Histórico 0.0: 82 + 7 = 89**
- **Não Aprovado (0.0): 92.13%**
- **Aprovado (0.0): 7.87%**

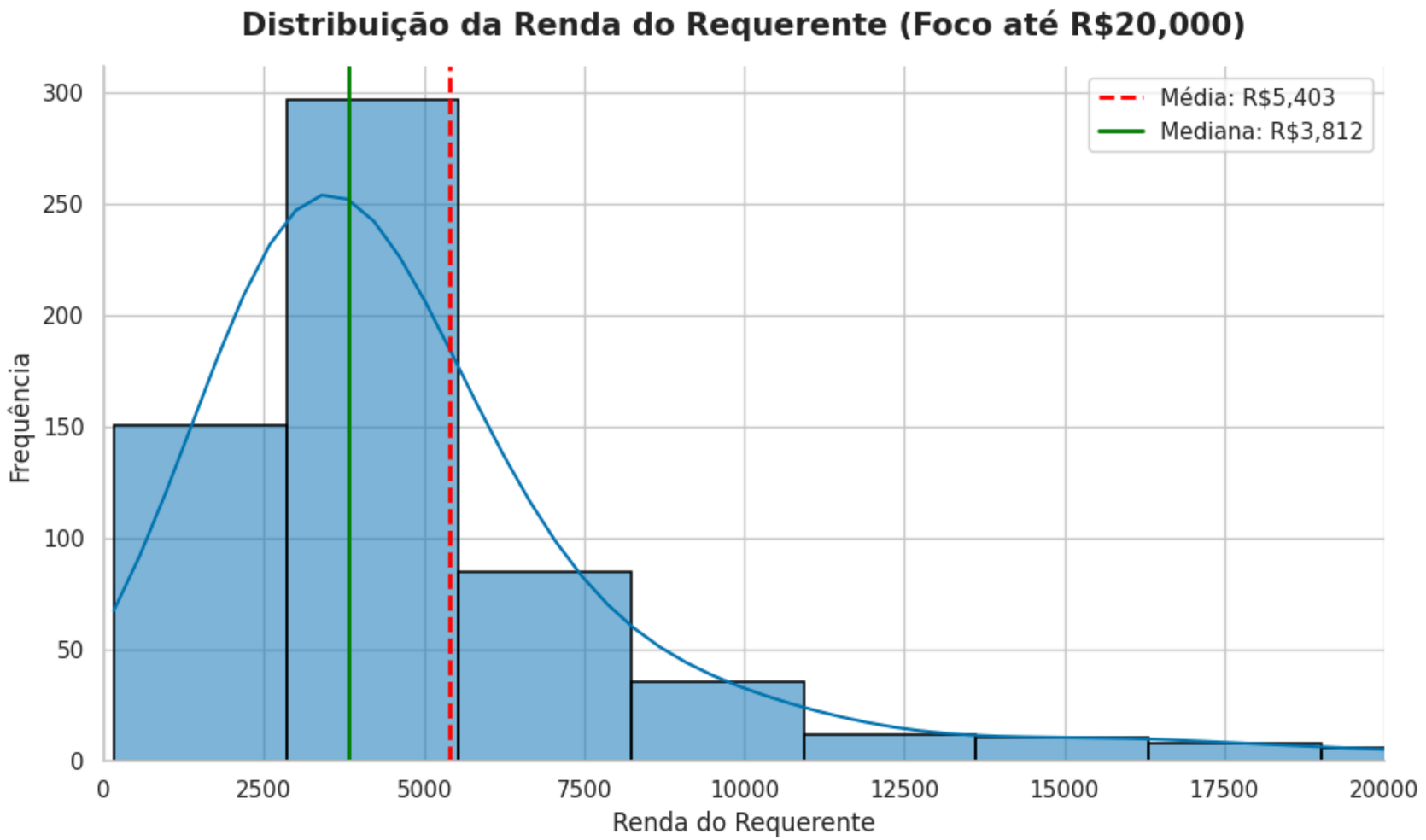
02 **Histórico de Crédito Adequado (1.0)**

- **Total de Requerentes com Histórico 1.0: 110 + 415 = 525**
- **Não Aprovado (1.0): 20.95%**
- **Aprovado (1.0): 79.05%**

DISTRIBUIÇÃO DA RENDA DO REQUERENTE



DISTRIBUIÇÃO DA RENDA DO REQUERENTE



MACHINE LEARNING

Nesta etapa, aplicamos técnicas de Machine Learning para prever se novos clientes terão seus empréstimos aprovados.

A proposta é treinar modelos capazes de aprender padrões a partir de dados cadastrais e financeiros, permitindo avaliar o risco de crédito e a probabilidade de aprovação.

```
categorical_cols = df.select_dtypes(include=['object']).columns.tolist()
categorical_cols.remove('ID')

df = pd.get_dummies(df, columns=categorical_cols, drop_first=True)

display(df.head())
```

	ID	Renda do requerente	Renda do co-requerente	Valor do empréstimo solicitado/aprovado	Prazo do empréstimo (em meses)	Indicador de histórico de crédito	Genero_Masculino	Genero_Não informado	Casado_Não informado	Casado_Sim	Dependentes_1	Dependentes_2	Dependentes_3+	Escolaridade_Não Graduado	trabalhador autônomo_Não informado	trabalhador autônomo_Sim
0	LP001002	5849	0.0	146.412162	360.0	1.0	True	False	False	False	False	False	False	False	False	False
1	LP001003	4583	1508.0	128.000000	360.0	1.0	True	False	False	True	True	False	False	False	False	False
2	LP001005	3000	0.0	66.000000	360.0	1.0	True	False	False	True	False	False	False	False	False	True
3	LP001006	2583	2358.0	120.000000	360.0	1.0	True	False	False	True	False	False	False	True	False	False
4	LP001008	6000	0.0	141.000000	360.0	1.0	True	False	False	False	False	False	False	False	False	False

Os dados categóricos, como gênero, escolaridade e área de residência, foram transformados em variáveis numéricas através do One-Hot Encoding, permitindo que os algoritmos compreendam essas informações.

A variável-alvo foi definida como o status do empréstimo (aprovado ou não).

TREINAMENTO

```
# corrigindo o nome da coluna alvo apos one-hot encoding

X = df.drop("Status final do pedido de empréstimo_Não aprovado", axis=1)
y = df["Status final do pedido de empréstimo_Não aprovado"]


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# separando id do df teste pra posteriormente conseguirmos identificar quem é quem
X_test_no_id = X_test.drop('ID', axis=1)
X_train_no_id = X_train.drop('ID', axis=1)

print("Shape of X_train_no_id:", X_train_no_id.shape)
print("Shape of X_test_no_id:", X_test_no_id.shape)
print("Shape of y_train:", y_train.shape)
print("Shape of y_test:", y_test.shape)
```

```
Shape of X_train_no_id: (491, 17)
Shape of X_test_no_id: (123, 17)
Shape of y_train: (491,)
Shape of y_test: (123,)
```

Os dados foram divididos em 80% para treino e 20% para teste.

A coluna de identificação (ID) foi removida das entradas, pois não contribui para a previsão.

TREINAMENTO

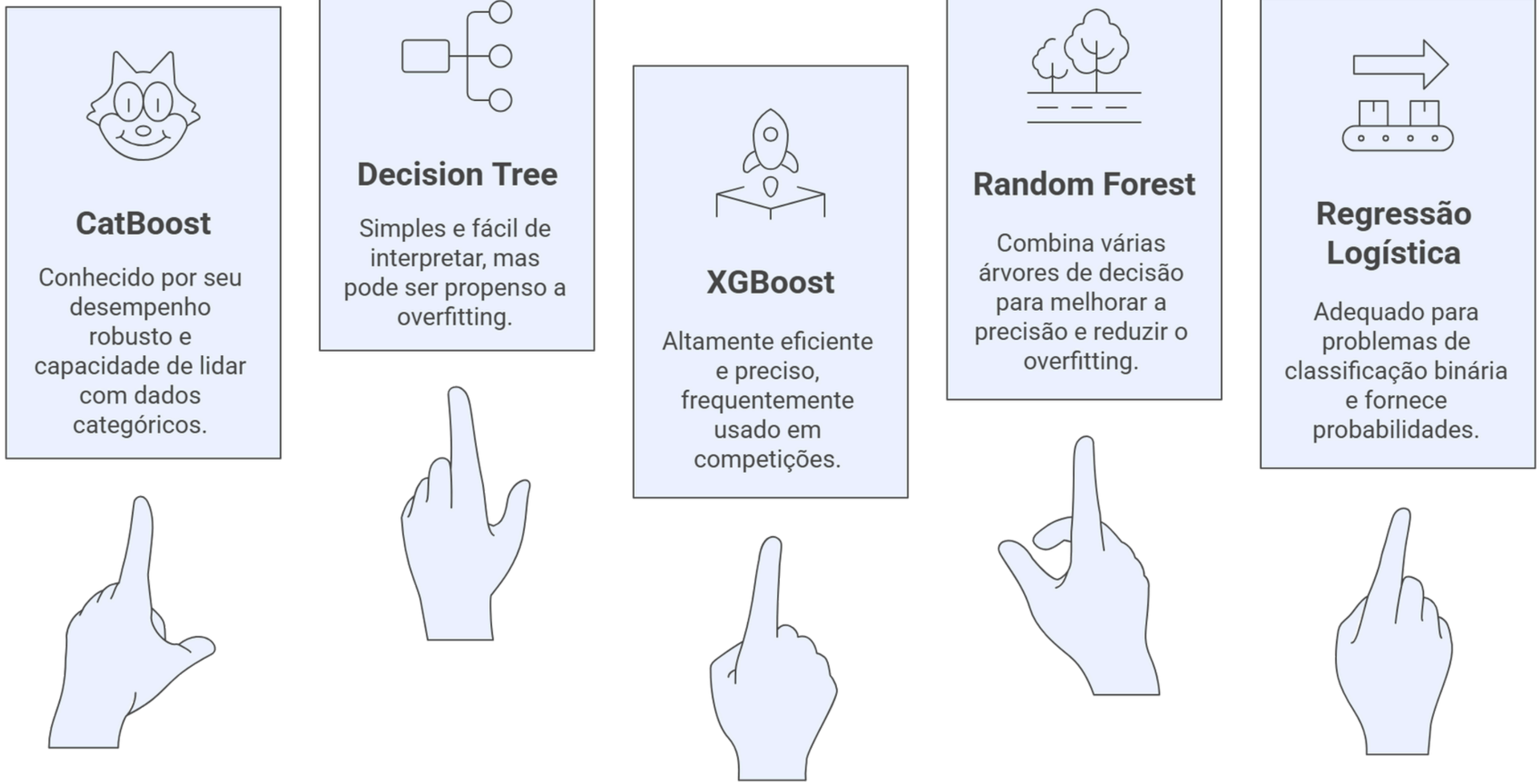
Escolhendo os modelos e treinando

```
models = {}
trained_models = {}
training_times = {}

models['CatBoost'] = CatBoostClassifier(random_state=42, verbose=0)
models['Decision Tree'] = DecisionTreeClassifier(random_state=42)
models['XGBoost'] = XGBClassifier(random_state=42, use_label_encoder=False, eval_metric='logloss')
models['Random Forest'] = RandomForestClassifier(random_state=42)
models['Logistic Regression'] = LogisticRegression(random_state=42)

for name, model in models.items():
    start_time = time.time()
    model.fit(X_train_no_id, y_train)
    end_time = time.time()
    training_times[name] = end_time - start_time
    trained_models[name] = model

print("Training Times:")
for name, train_time in training_times.items():
    print(f"{name}: {train_time:.4f} seconds")
```

Foram testados cinco modelos:

CatBoost, **Decision Tree**, **XGBoost**, **Random Forest** e **Regressão Logística**.

TREINAMENTO



RESULTADOS

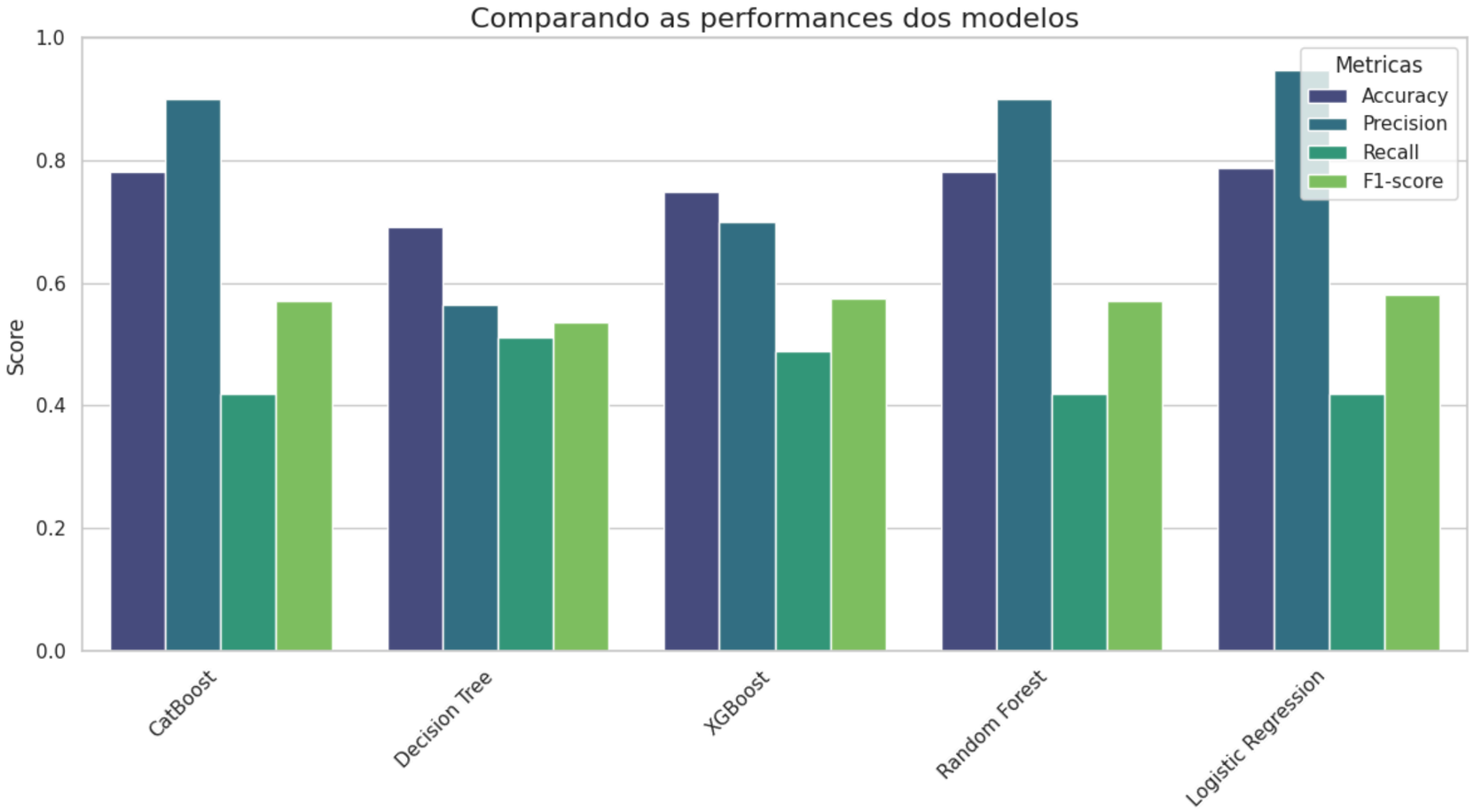
	Model	Accuracy	Precision	Recall	F1-score
0	CatBoost	0.780488	0.900000	0.418605	0.571429
1	Decision Tree	0.691057	0.564103	0.511628	0.536585
2	XGBoost	0.747967	0.700000	0.488372	0.575342
3	Random Forest	0.780488	0.900000	0.418605	0.571429
4	Logistic Regression	0.788618	0.947368	0.418605	0.580645

1º – Regressão Logística: melhor desempenho geral, com alta precisão (94,7%).

2º – CatBoost: boa acurácia (78%) e precisão (90%), robusto e estável, mas menos interpretável e com maior tempo de treinamento.

3º – XGBoost: bom equilíbrio entre recall e F1-score, detecta mais aprovações reais, mas gera mais falsos positivos.

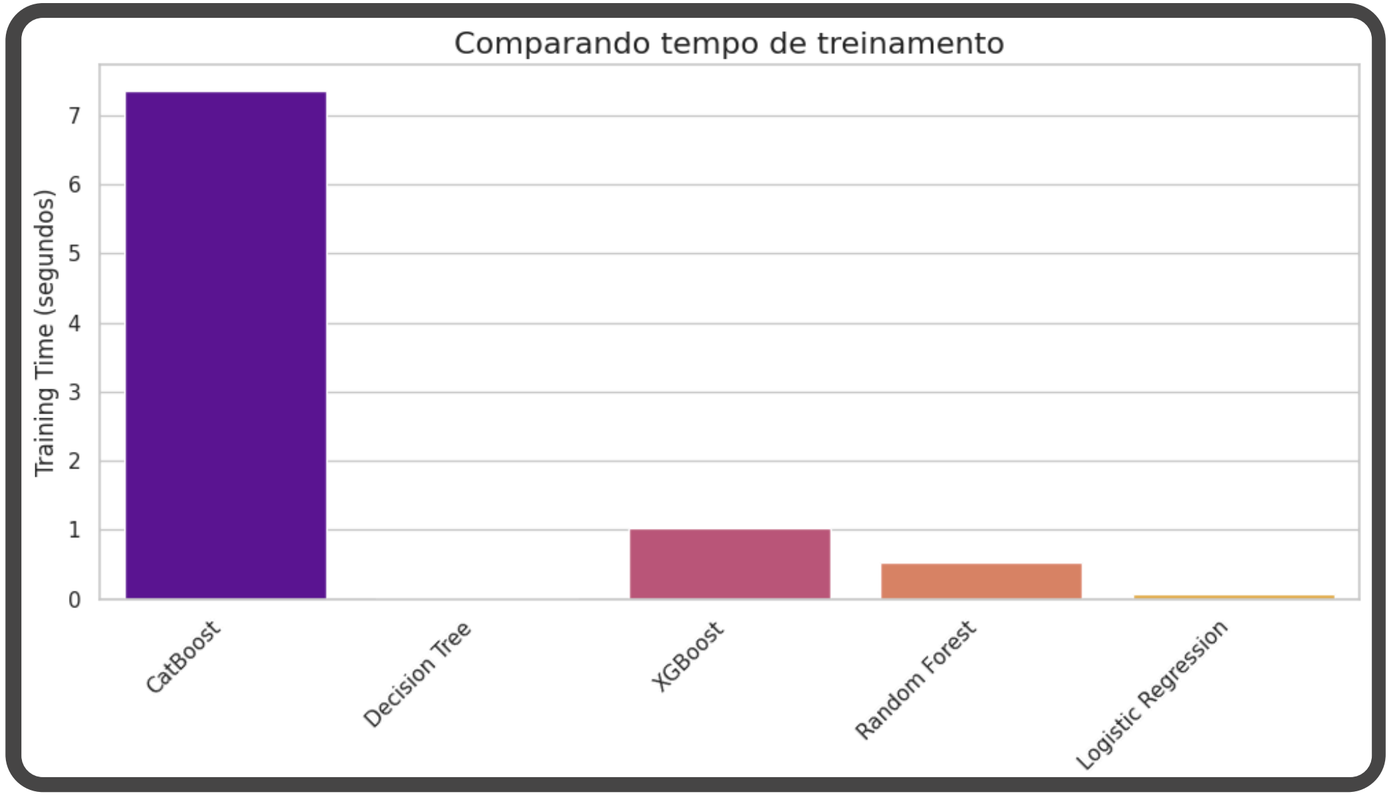
AVALIANDO MODELOS



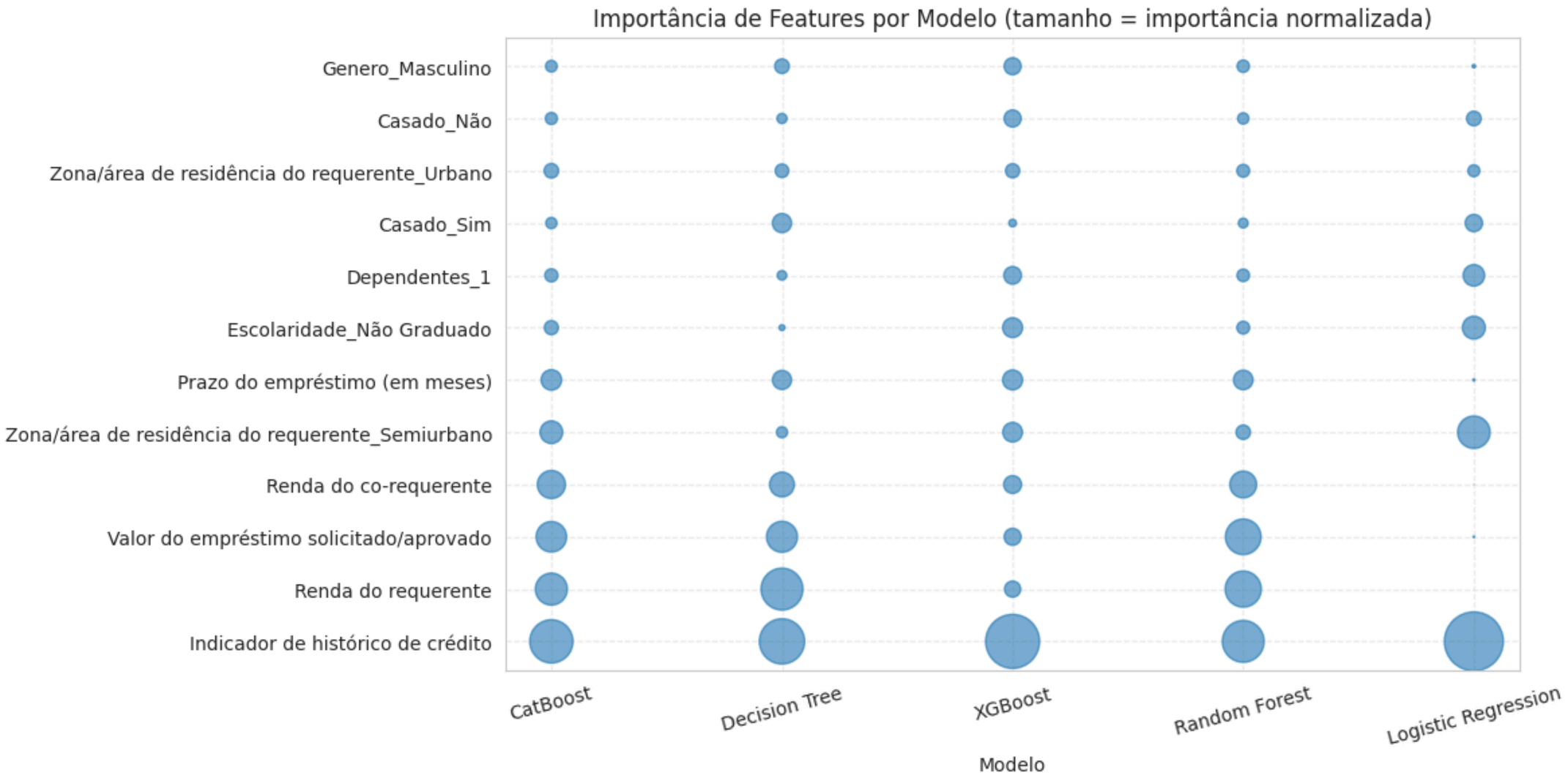
A Regressão Logística apresentou o melhor desempenho em precisão e acurácia, tornando-se o modelo mais adequado para o negócio, que prioriza minimizar falsos positivos (evitar aprovar clientes de alto risco).

TEMPO DE TREINAMENTO

	Model	Training Time (seconds)
0	CatBoost	7.359342
1	Decision Tree	0.010108
2	XGBoost	1.030791
3	Random Forest	0.522427
4	Logistic Regression	0.076239



FEATURES


Importância de Features por Modelo (tamanho = importância normalizada)
Genero_Masculino
Casado_Não
Zona/área de residência do requerente_Urbano
Casado_Sim
Dependentes_1
Escolaridade_Não Graduado
Prazo do empréstimo (em meses)
Zona/área de residência do requerente_Semiurbano
Renda do co-requerente
Valor do empréstimo solicitado/aprovado
Renda do requerente
Indicador de histórico de crédito
CatBoost
Decision Tree
XGBoost
Random Forest
Logistic Regression
Modelo

SIMULANDO “NOVOS CLIENTES”

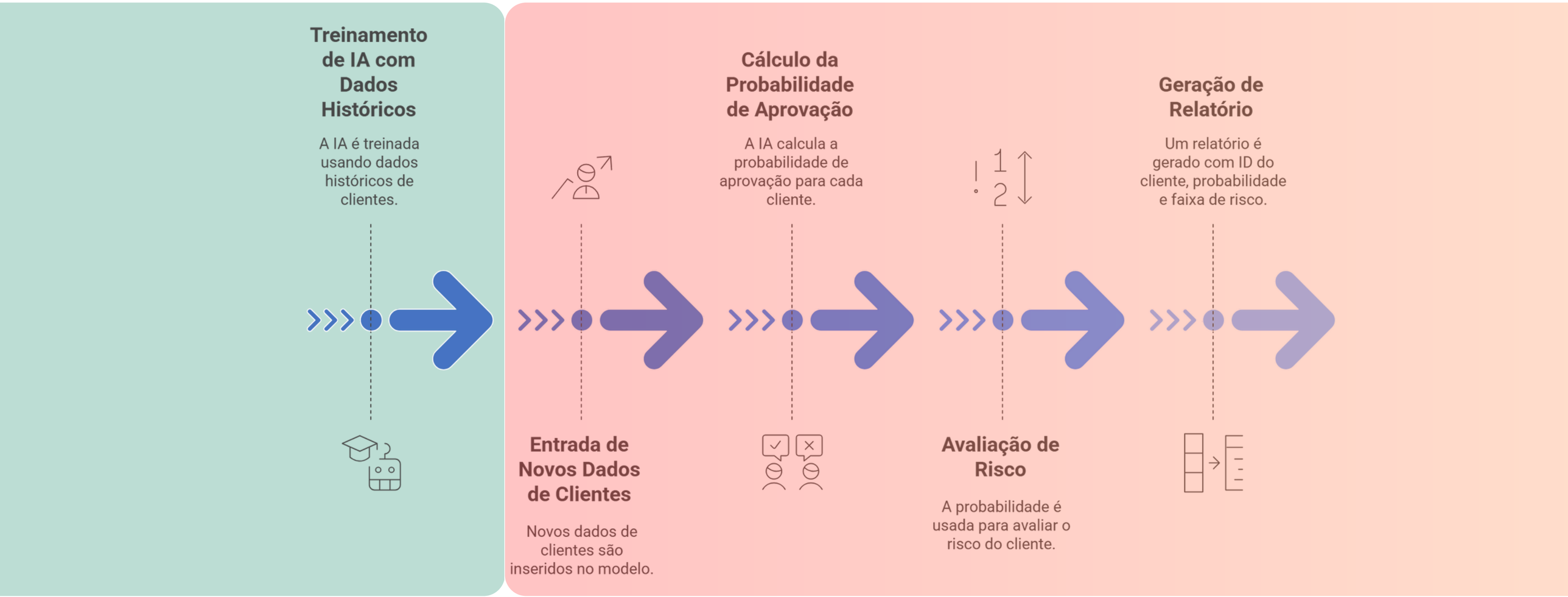
E APLICANDO PROBABILIDADE DE RISCO

```
results_df = results_df.sort_values(by='Probabilidade de Aprovação', ascending=False)
results_df
```

	ID	Probabilidade de Aprovação	Faixa de Risco
182	LP001636	0.959334	Baixo Risco
575	LP002868	0.937626	Baixo Risco
316	LP002035	0.930312	Baixo Risco
55	LP001194	0.929268	Baixo Risco
231	LP001768	0.922443	Baixo Risco
...	...	...	...
396	LP002277	0.122571	Alto Risco
584	LP002911	0.113222	Alto Risco
412	LP002328	0.103227	Alto Risco
300	LP001964	0.101877	Alto Risco
421	LP002357	0.086559	Alto Risco

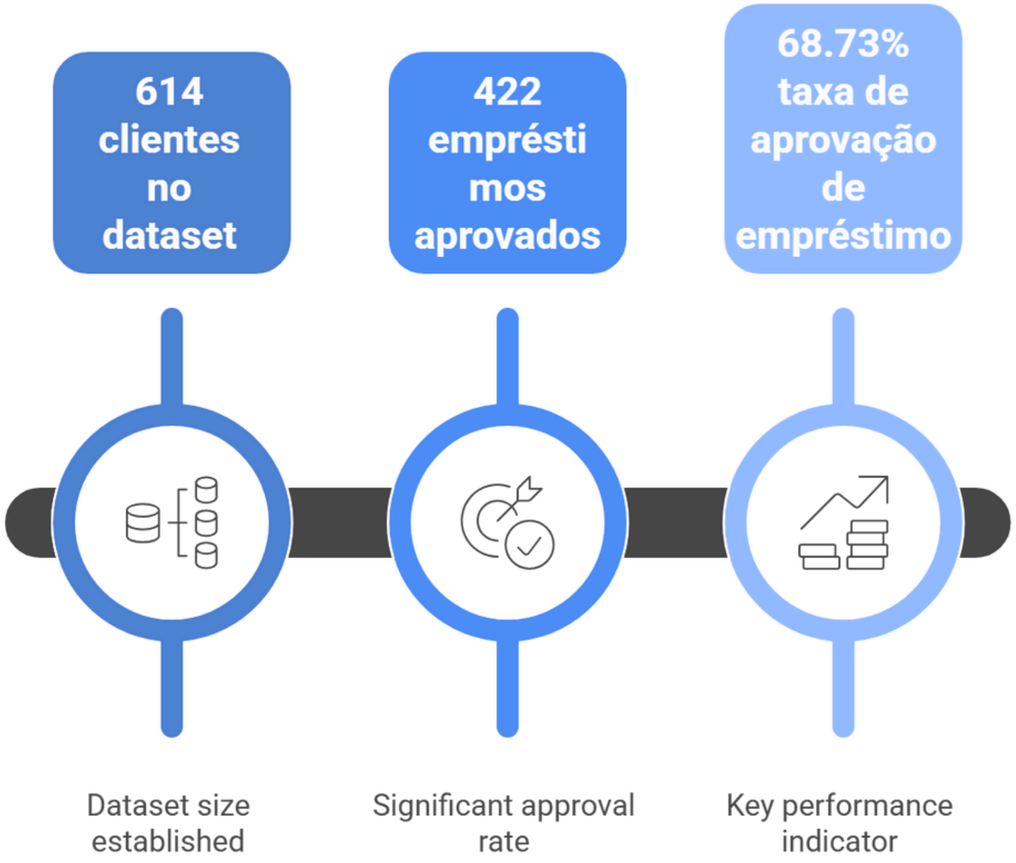
123 rows × 3 columns

UTILIZANDO O MODELO



RESULTADOS E CONCLUSÃO

Qual é a probabilidade de aprovação/risco de cada novo cliente, considerando suas informações cadastrais e financeiras?



Muito obrigado pela atenção!