Data Understanding

Bank Marketing

Discentes:

* Arthur Bezerra Calado
* Gabriel D'assumpção de Carvalho
* Pedro Henrique Sarmento de Paula

Data: 16/07/2024

Introdução

O objetivo desta fase do projeto é fornecer uma compreensão detalhada dos dados fornecidos. A importância do entendimento dos dados reside em sua capacidade de guiar as próximas etapas do projeto, garantindo que as decisões sejam fundamentadas em informações precisas e completas. Compreender a estrutura, a qualidade e os padrões dos dados ajuda a identificar possíveis desafios e oportunidades de melhoria antes de avançar para a modelagem. Esta fase está diretamente ligada ao Entendimento do Negócio, pois traduz os requisitos de negócio em um contexto de dados, assegurando que a análise esteja alinhada com os objetivos da organização. O entendimento dos dados permite identificar características importantes, detectar anomalias e garantir que as análises e modelos preditivos sejam construídos com base em informações confiáveis.

Nesta fase do projeto, realizaremos uma análise exploratória detalhada dos dados do dataset "Bank Marketing" para entender melhor a estrutura dos dados, identificar padrões, verificar a distribuição das variáveis e descobrir possíveis correlações que possam influenciar os resultados.

Coleta Inicial de Dados

A coleta inicial dos dados envolve documentar as fontes e métodos utilizados para obter os dados fornecidos. Os dados utilizados neste projeto são provenientes de campanhas de marketing direto realizadas por uma instituição bancária portuguesa. Essas campanhas foram baseadas em chamadas telefônicas para os clientes, com o objetivo de avaliar se os clientes subscreveriam um depósito a prazo. O dataset possui um total de 45.211 registros e 16 características, abrangendo variáveis categóricas, numéricas e binárias. O formato dos dados é tabular, com cada linha representando um cliente e cada coluna representando uma característica ou a variável alvo.

Biblioteca utilizadas

1. pandas -> manipulação de dados;
2. numpy -> cálculos estatísticos;
3. matplotlib -> gráficos;
4. seaborn -> gráficos;
5. scipy -> transformação de variável;
6. warnings -> remoção de avisos

In [ ]:
```
# Instalação das bibliotecas

# %pip install pandas

# %pip install ucimlrepo

# %pip install numpy

# %pip install scipy

# %pip install matplotlib

# %pip install seaborn
```

In [ ]:
```
# Importação das bibliotecas

import pandas as pd

import numpy as np

from scipy.stats import boxcox

import matplotlib.pyplot as plt

import seaborn as sns

import warnings
```

```
from ucimlrepo import fetch_ucirepo

from IPython.display import display, Markdown
```

In [ ]:
```
# Desativa todos os avisos
warnings.filterwarnings("ignore")
```

In [ ]:
```
# Configurando o modo de exibição do pandas
pd.options.display.float_format = "{:.4f}".format
```

Coletando os dados

In [ ]:
```
# Baixando os dados
bank_marketing = fetch_ucirepo(id=222)
```

In [ ]:
```
# data (as pandas dataframes)
X = bank_marketing.data.features
y = bank_marketing.data.targets
```

In [ ]:
```
# Criando variável que vai ter as variáveis transformada
Xt = X
```

Descrição dos Dados

A descrição detalhada dos dados é essencial para entender a natureza das variáveis e a distribuição dos valores. O dataset contém variáveis de diferentes tipos:

Numéricas: incluem variáveis como 'age' (idade do cliente), 'balance' (saldo médio anual), 'duration' (duração da última chamada em segundos), 'campaign' (número de contatos realizados durante esta campanha), 'pdays' (número de dias desde o último contato em uma campanha anterior) e 'previous' (número de contatos realizados antes desta campanha).

Categóricas: incluem 'job' (tipo de emprego), 'marital' (estado civil), 'education' (nível educacional), 'contact' (tipo de comunicação), 'day_of_week' (dia da semana do último contato), 'month' (mês do último contato) e 'poutcome' (resultado da campanha anterior).

Binárias: incluem 'default' (tem crédito em default?), 'housing' (tem empréstimo habitacional?), 'loan' (tem empréstimo pessoal?) e a variável alvo 'y' (o cliente subscreveu um depósito a prazo?).

As variáveis que serão exploradas incluem:

Nome da variável	Papel	Tipo	Demográfico	Descrição	Unidades	Valores ausentes
idade	Característica	Inteiro	Idade			Não
trabalho	Característica	Categórico	Ocupação	Tipo de emprego (categórico: 'Admin.', 'Blue-collar', 'Entrepreneur', 'Householdant', 'Management', 'Aposentado', 'Autônomo', 'Serviços', 'Estudante', 'Técnico', 'Desempregado', 'Desconhecido')		Sim
conjugal	Característica	Categórico	Estado civil	estado civil (categórico: 'divorciado', 'casado', 'solteiro', 'desconhecido'; nota: 'divorciado' significa divorciado ou viúvo)		Não
educação	Característica	Categórico	Nível de escolaridade	(categórico: 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'analfabeto', 'professional.course', 'university.degree', 'unknown')		Sim
inadimplência	Característica	Binário		tem crédito inadimplente?		Não
equilíbrio	Característica	Inteiro		saldo médio anual	Euros	Não
habitação	Característica	Binário		Tem crédito habitação?		Não
empréstimo	Característica	Binário		Tem empréstimo pessoal?		Não
contato	Característica	Categórico		tipo de comunicação de contato (categórico: 'celular', 'telefone')		Sim
day_of_week	Característica	Data		último dia de contato da semana		Não
mês	Característica	Data		último mês do ano de contato (categórico: 'jan', 'feb', 'mar', ..., 'nov', 'dec')		Não
duração	Característica	Inteiro		Duração do último contato, em segundos (numérico). Observação importante: esse atributo afeta fortemente o destino de saída (por exemplo, se duration=0 e y='no'). No entanto, a duração não é conhecida antes de uma chamada ser realizada. Além disso, após o fim da chamada y é obviamente conhecido. Assim, esse insumo só deve ser incluído para fins de benchmark e deve ser descartado se a intenção for ter um modelo preditivo realista.		Não
campanha	Característica	Inteiro		número de contatos realizados durante esta campanha e para este cliente (numérico, inclui último contato)		Não
pdays	Característica	Inteiro		número de dias que se passaram após o cliente ter sido contatado pela última vez a partir de uma campanha anterior (numérico; -1 significa que o cliente não foi contatado anteriormente)		Não
anterior	Característica	Inteiro		número de contatos realizados antes desta campanha e para este cliente		Não
presultado	Característica	Categórico		resultado da campanha de marketing anterior (categórico: 'fracasso', 'inexistente', 'sucesso')		Sim
y	Alvo	Binário		O cliente subscreveu um depósito a prazo?		Não

Exploração dos Dados

A exploração dos dados visa identificar padrões, tendências e anomalias que podem impactar a análise. Outliers são identificados e avaliados quanto à sua significância, enquanto correlações entre variáveis são analisadas para identificar relações importantes. A análise de variáveis categóricas inclui a avaliação das frequências e modos, e visualizações adicionais, como heatmaps e pair plots, são utilizadas para suportar a análise e identificar padrões mais complexos.

In [ ]:
```
# Verificando as 5 primeiras linhas de x
print(X.head())
```

```
   age           job  marital  education default  balance housing loan  \
0   58    management  married   tertiary      no     2143     yes   no
1   44    technician   single  secondary      no       29     yes   no
2   33  entrepreneur  married  secondary      no        2     yes  yes
3   47   blue-collar  married        NaN      no     1506     yes   no
4   33           NaN   single        NaN      no        1      no   no

  contact  day_of_week month  duration  campaign  pdays  previous poutcome
0     NaN            5   may       261         1     -1         0      NaN
1     NaN            5   may       151         1     -1         0      NaN
2     NaN            5   may        76         1     -1         0      NaN
3     NaN            5   may        92         1     -1         0      NaN
4     NaN            5   may       198         1     -1         0      NaN
```

In [ ]:
```
# Verificando as 5 primeiras linhas de x
print(y.head())
```

```
     y
0  no
1  no
2  no
3  no
4  no
```

Variáveis Explicativas

Vamos realizar uma análise exploratória das variáveis explicativas do conjunto de dados. Abaixo estão as estatísticas descritivas e visualizações para cada variável.

Idade

In [ ]:
```
# Resumo estatístico
X["age"].describe()
```

Out[ ]:
```
count   45211.0000
mean       40.9362
std        10.6188
min        18.0000
25%        33.0000
50%        39.0000
75%        48.0000
max        95.0000
Name: age, dtype: float64
```

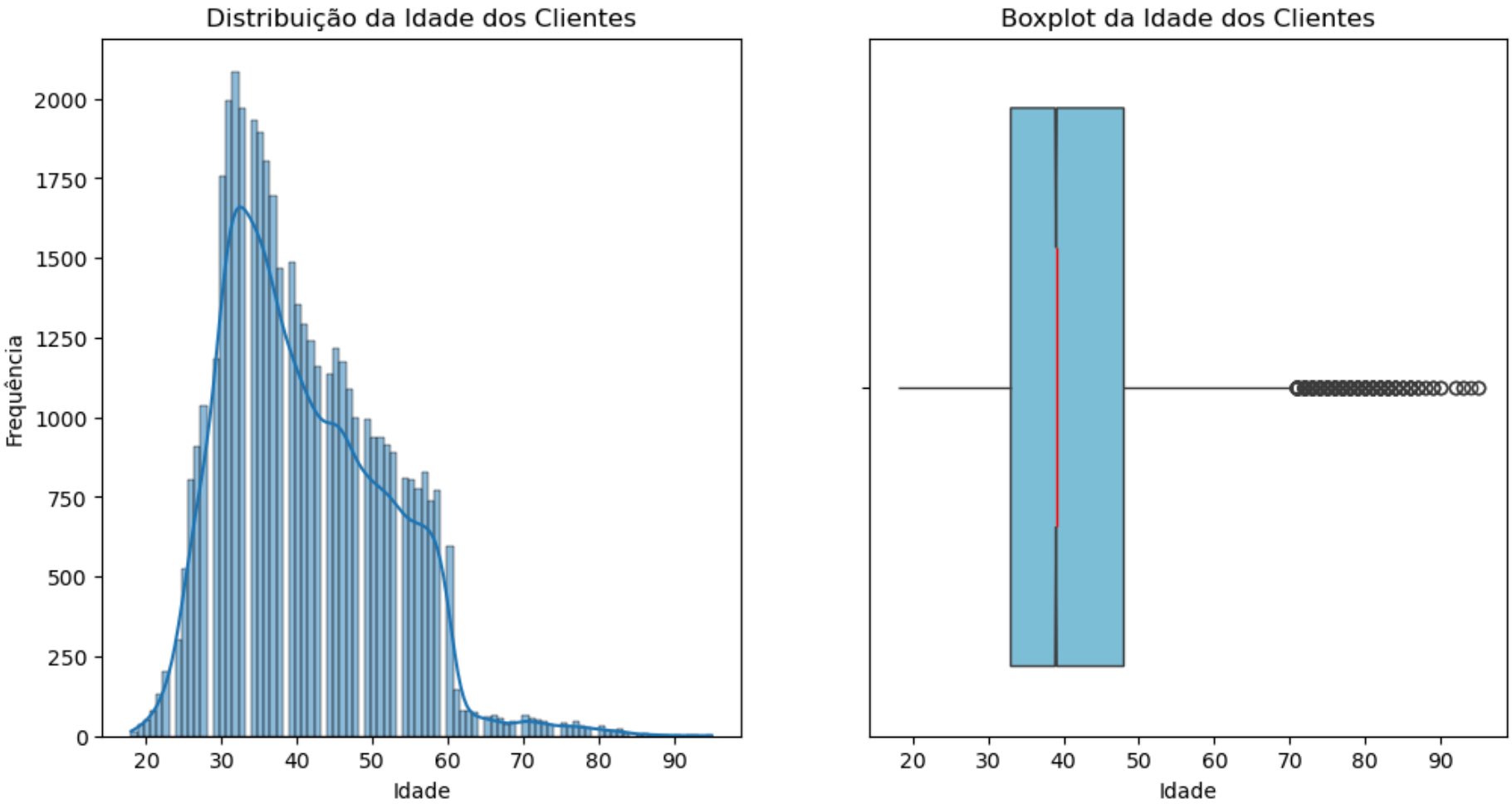
In [ ]:
```
# Visualizando a moda
X["age"].mode()
```

Out[ ]:
```
0    32
Name: age, dtype: int64
```

Ao analisarmos em detalhes os dados relativos à idade dos clientes, percebemos que a média das idades é próxima de 41, variando entre o mínimo de 18 e um máximo de 95. Além disso, podemos observar um desvio padrão de aproximadamente 11, o que indica uma dispersão significativa dos valores. Portanto, em média, os clientes têm idades compreendidas entre 30 e 51 anos.

É importante notar que a média de 41 anos supera tanto a mediana de 39 quanto a moda de 32, sugerindo uma assimetria positiva nos dados devido à presença de idades mais altas no terceiro quartil, que é de 48 anos, próximo do valor máximo.

In [ ]:
```
# Criação do grafico histograma e boxplot
plt.figure(figsize=(12,6))
plt.subplot(1, 2, 1)
sns.histplot(X["age"], kde=True)
plt.title("Distribuição da Idade dos Clientes")
plt.xlabel("Idade")
plt.ylabel("Frequência")
plt.subplot(1, 2, 2)
sns.boxplot(X["age"], orient='h', notch=True, showcaps=False,
            boxprops={"facecolor": (0, .5, .7, .5)},
            medianprops={"color": "r", "linewidth": 1})
plt.title("Boxplot da Idade dos Clientes")
plt.xlabel("Idade")
plt.show()
```



Como mencionado anteriormente, é possível observar a assimetria positiva nas idades devido à presença de uma pequena parte dos clientes que são pessoas idosas. Após analisar todas as variáveis, vamos propor algumas transformações para tentar melhorar a qualidade dos dados, reduzindo assimetrias e outliers.

Trabalho

A variável que indica o trabalho exercido por cada cliente é do tipo categórica, podendo ser:

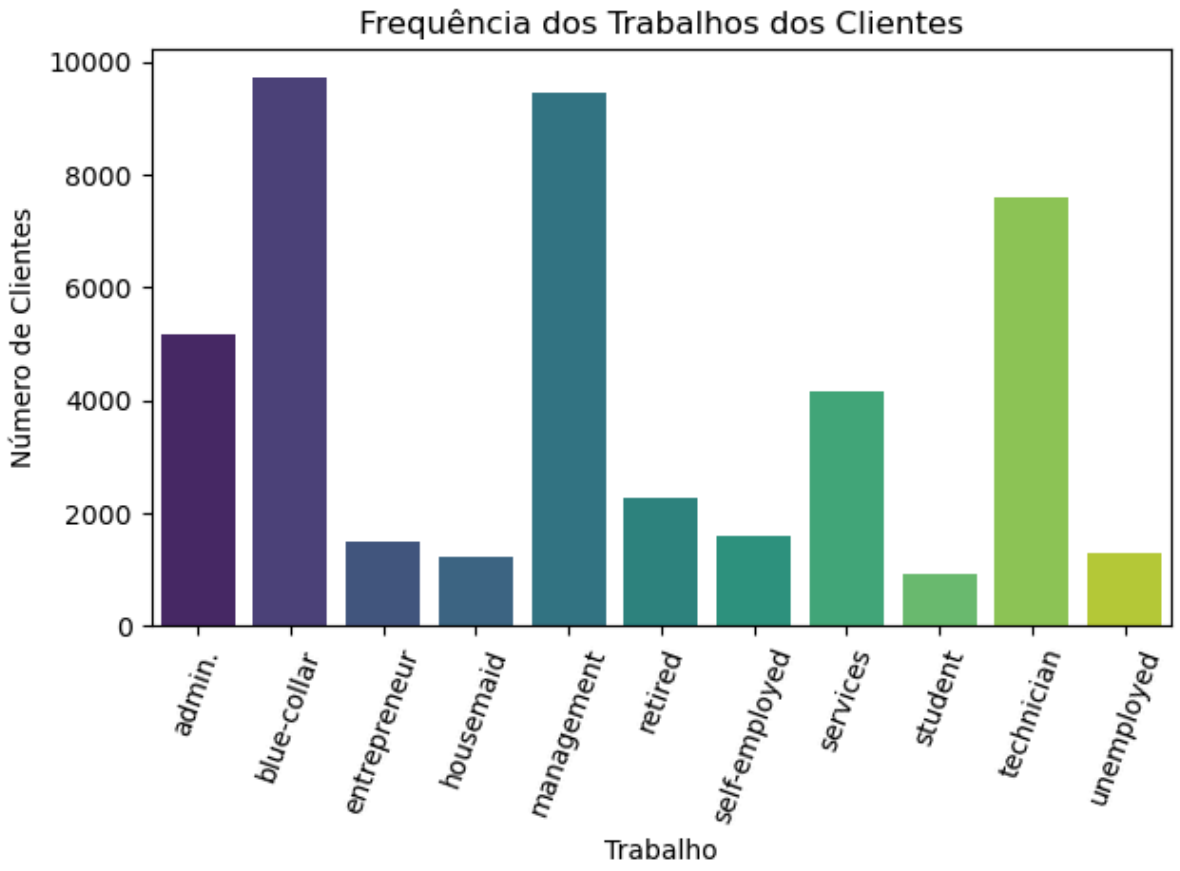
```
Admin.: Administrativo
Blue-collar: Trabalhador manual
Entrepreneur: Empresário
Householdant: Trabalhador doméstico
Management: Gerência
Retired: Aposentado
Self-employed: Autônomo
Services: Serviços gerais
student: Estudante
Technician: Técnico
Unemployed: Desempregado
Unknown: Desconhecido
```

Para verificar essa variável, vamos estar analisando a quantidade da frequência de cada classe.

In [ ]:
```python
# Criando uma tabela de frequência
job_counts = X['job'].value_counts().sort_index()
print(job_counts*100/44923)
```

```
job
admin.          11.5108
blue-collar     21.6637
entrepreneur     3.3101
housemaid        2.7603
management      21.0538
retired          5.0397
self-employed    3.5149
services         9.2469
student          2.0880
technician      16.9112
unemployed       2.9005
Name: count, dtype: float64
```

In [ ]:
```python
# Criando um gráfico de frequência
sns.barplot(x=job_counts.index, y=job_counts.values, palette='viridis')
plt.title('Frequência dos Trabalhos dos Clientes')
plt.xlabel('Trabalho')
plt.ylabel('Número de Clientes')
plt.xticks(rotation=70)
plt.tight_layout()
plt.show()
```



Podemos observar tanto na lista acima quanto no gráfico que a base de dados possui mais clientes que desempenham papéis de administradores, trabalhadores manuais, gerência e técnicos, representando 11,51%, 21,6%, 21,05% e 16,91% das 44.923 observações da variável trabalho. É importante lembrar que esta variável possui 1.303 observações classificadas como trabalho desconhecido, além de 288 observações faltantes.

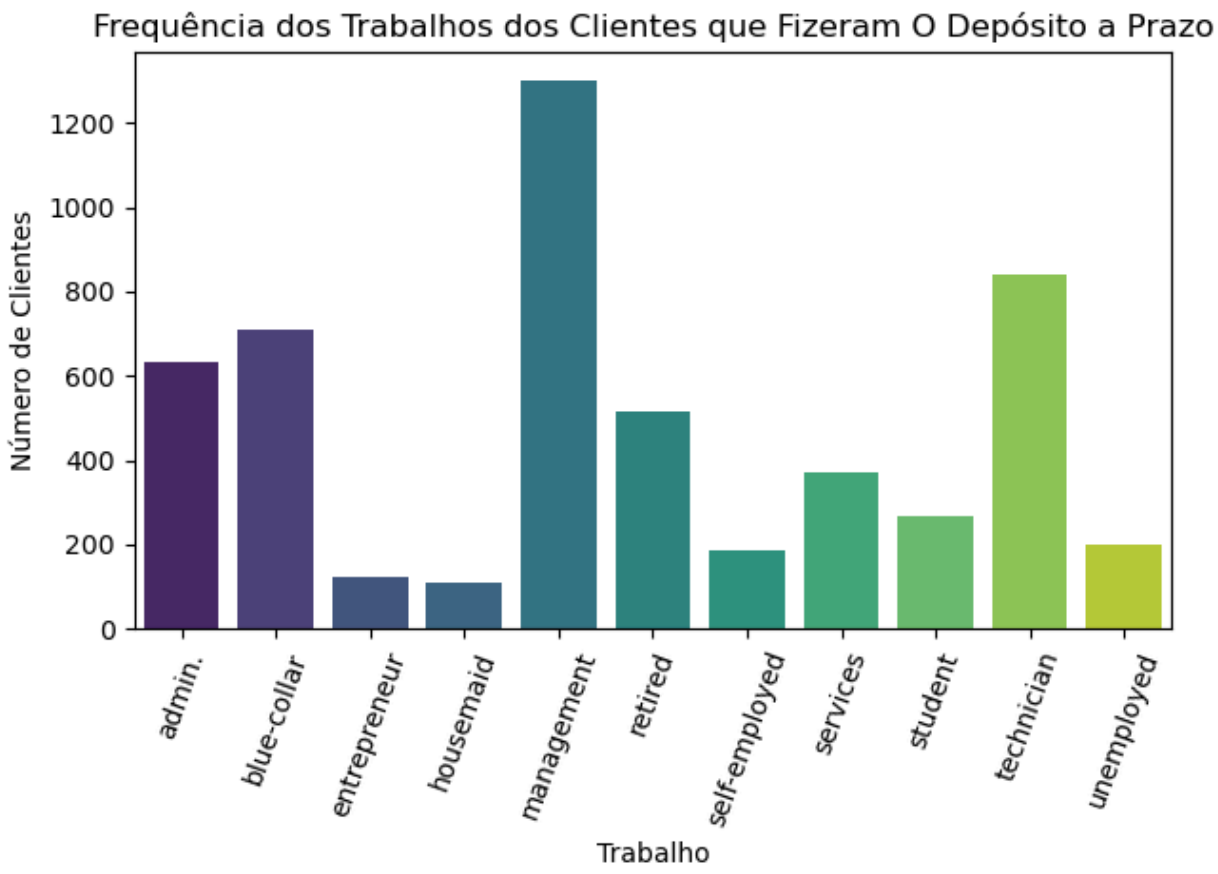
Devido aos problemas como desequilíbrio entre as classes e dados faltantes, na sessão de transformações serão abordados alguns mecanismos para o tratamento dessa variável.

Além disso, podemos verificar quais tipos de trabalhadores são mais propensos a aceitar o produto oferecido pelo banco por meio do telemarketing.

In [ ]:
```
# Criando uma tabela de frequencia para clientes que aceitaram o produto
job_yes_counts = X[y.values == 'yes']['job'].value_counts().sort_index()
job_yes_percentages = job_yes_counts * 100 / job_yes_counts.sum()
print(job_yes_percentages)
```

```
job
admin.          12.0076
blue-collar     13.4729
entrepreneur     2.3406
housemaid        2.0742
management      24.7574
retired          9.8192
self-employed    3.5585
services         7.0219
student          5.1189
technician      15.9848
unemployed       3.8440
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frequência para clientes que aceitaram o produto
sns.barplot(x=job_yes_counts.index, y=job_yes_counts.values, palette='viridis')
plt.title('Frequência dos Trabalhos dos Clientes que Fizeram O Depósito a Prazo')
plt.xlabel('Trabalho')
plt.ylabel('Número de Clientes')
plt.xticks(rotation=70)
plt.tight_layout()
```



Observando os trabalhadores que aceitaram o depósito a prazo, as coisas mudam um pouco. A maioria dos clientes que aceitam tem papéis como gerentes, técnicos, trabalhadores manuais, administradores e aposentados, representando aproximadamente 24,76%, 15,98%, 13,47% e 12,01% das 5.255 observações, respectivamente.

Essa análise é de extrema importância porque, anteriormente, vimos que o banco tem feito telemarketing para muitas pessoas que realizam trabalhos manuais, enquanto gerentes e técnicos representam cerca de 40,73% dos clientes que aceitam o produto.

Estado Civil

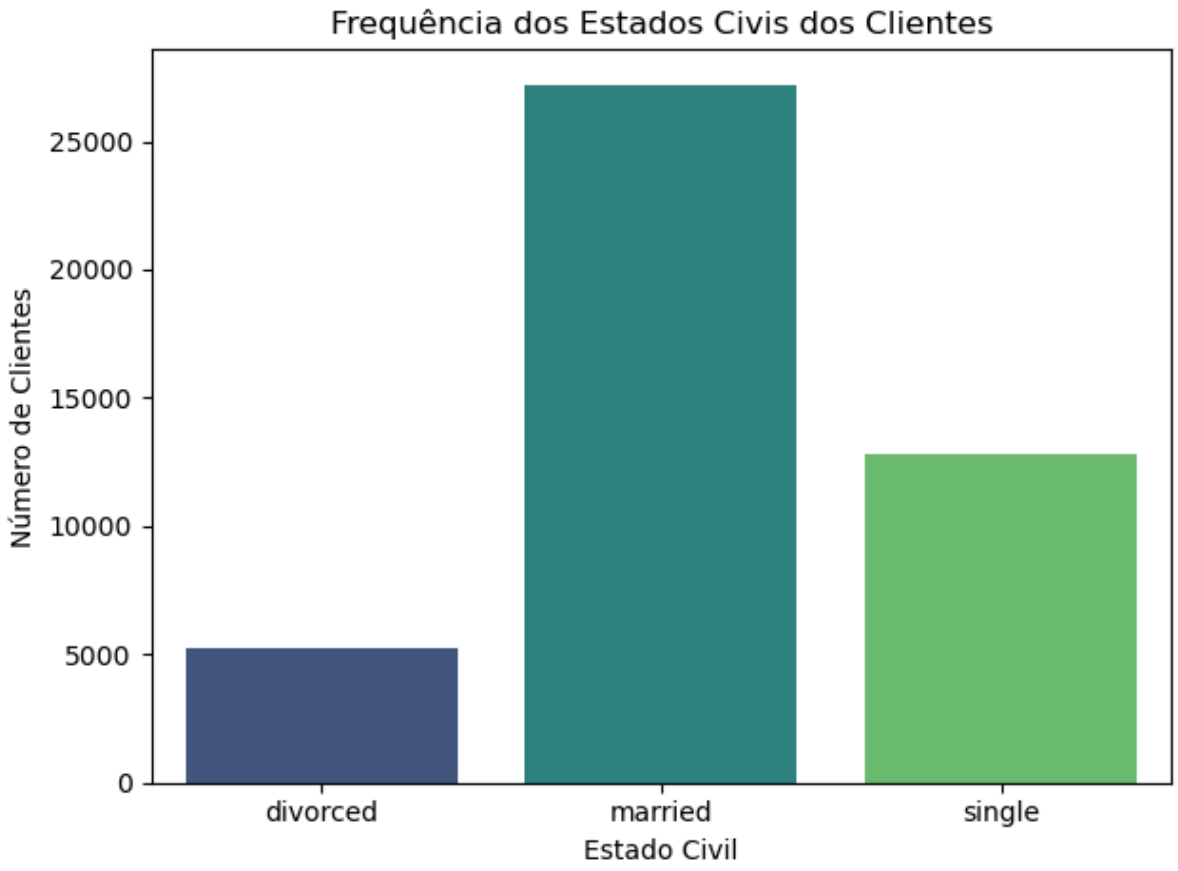
A variável *marital* que representa o estado civil do cliente pode apresentar 4 categorias, sendo elas:

```
Married: Casado
Single: Solteiro
Divorced: Divorciado ou Viúvo
Unknown: Desconhecido
```

In [ ]:
```
# Criando uma tabela de frequência
marital_counts = X['marital'].value_counts().sort_index()
marital_percentages = (marital_counts * 100 / sum(marital_counts.values))
print(marital_percentages)
```

```
marital
divorced   11.5171
married    60.1933
single     28.2896
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia
sns.barplot(x=marital_counts.index, y=marital_counts.values, palette='viridis')
plt.title('Frequência dos Estados Civis dos Clientes')
plt.xlabel('Estado Civil')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```



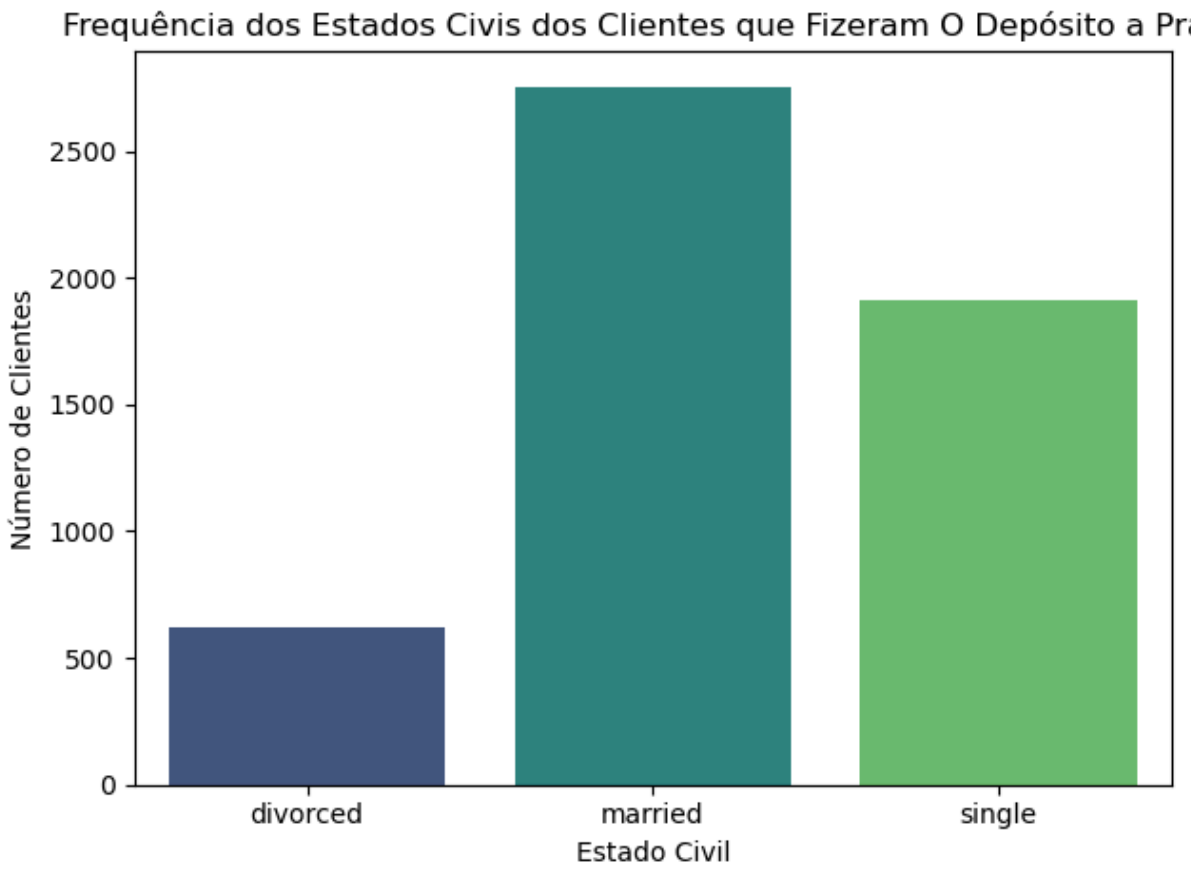
Como podemos ver tanto na lista de porcentagens quanto no gráfico de barras, os clientes casados, solteiros e divorciados representam aproximadamente 60,19%, 29,29% e 11,51% das 45.211 observações, respectivamente. Além disso, não há nenhum cliente com estado civil desconhecido.

Para entender mais sobre os clientes que aceitam o produto do banco devido ao telemarketing, podemos ver abaixo a análise feita com a variável de estado civil restrita aos clientes que realizaram o depósito a prazo.

In [ ]:
```
# Criando uma tabela de frequencia para clientes que aceitaram o produto
marital_yes_counts = X[y.values == 'yes']['marital'].value_counts().sort_index()
marital_yes_percentages = marital_yes_counts * 100 / marital_yes_counts.sum()
print(marital_yes_percentages)
```

```
marital
divorced   11.7603
married    52.0892
single     36.1505
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frequencia para clientes que aceitaram o produto
sns.barplot(x=marital_yes_counts.index, y=marital_yes_counts.values, palette='viridis')
plt.title('Frequência dos Estados Civis dos Clientes que Fizeram O Depósito a Prazo')
plt.xlabel('Estado Civil')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```



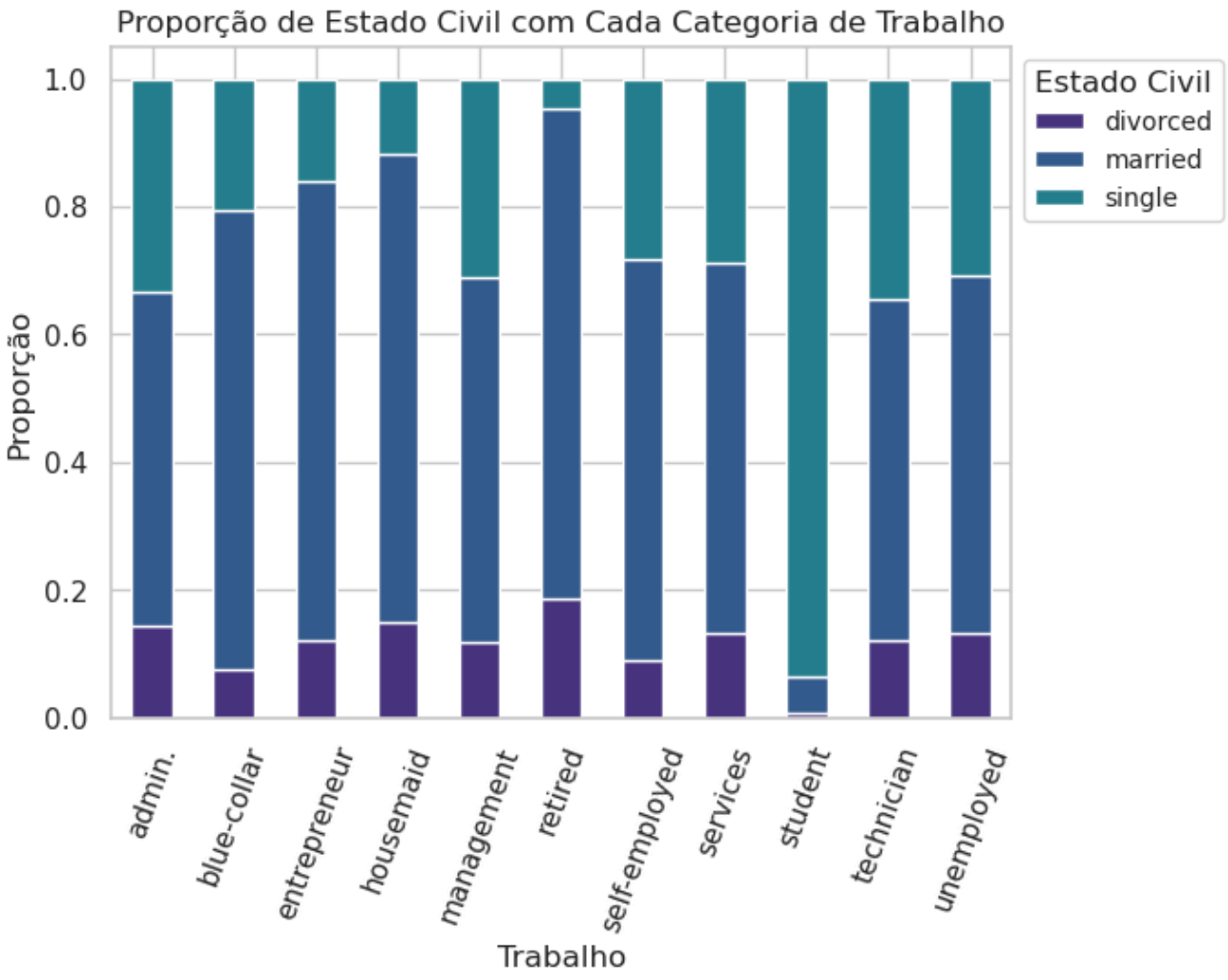
Podemos observar que os clientes que geralmente aceitam fazer o depósito a prazo não diferem significativamente do total de clientes do banco. Os clientes casados continuam sendo a maioria, seguidos por solteiros e divorciados, representando aproximadamente 52,09%, 36,15% e 11,76% dos 5.289 clientes que aceitaram o produto decorrente do marketing direto, respectivamente.

Trabalho e Estado Civil

Uma análise interessante a ser realizada é verificar a proporção de cada tipo de trabalho em relação ao estado civil. Isso pode fornecer insights valiosos sobre o perfil dos clientes antes de criar um modelo categórico. Com essas informações, a instituição financeira pode ajustar suas estratégias de telemarketing de maneira mais eficaz, mesmo antes da conclusão do modelo de aprendizado de máquina. Dessa forma, cada etapa do projeto se torna mais útil e contribui para a implementação de melhorias contínuas nas estratégias de marketing.

In [ ]:
```
#Criando uma tabela cruzada
job_marital_counts = pd.crosstab(X['job'], X['marital'])
job_marital_counts_normalized = job_marital_counts.div(job_marital_counts.sum(axis=1), axis=0)

# Plotar gráfico de proporção
sns.set_theme(style="whitegrid", palette='viridis')
job_marital_counts_normalized.plot(kind='bar', stacked=True)
plt.xlabel('Trabalho')
plt.ylabel('Proporção')
plt.title('Proporção de Estado Civil com Cada Categoria de Trabalho')
plt.legend(title='Estado Civil', bbox_to_anchor=(1, 1), loc='upper left', fontsize='small')
plt.xticks(rotation=70)
plt.show()
```



Como podemos observar no gráfico acima, que mostra a proporção de estado civil para cada categoria de trabalho, há algumas tendências interessantes. Os estudantes, por exemplo, apresentam mais de 90% de solteiros. Como era de se esperar, as pessoas aposentadas (retired) têm quase 20% de separados ou viúvos, e um pouco mais de 70% são casados.

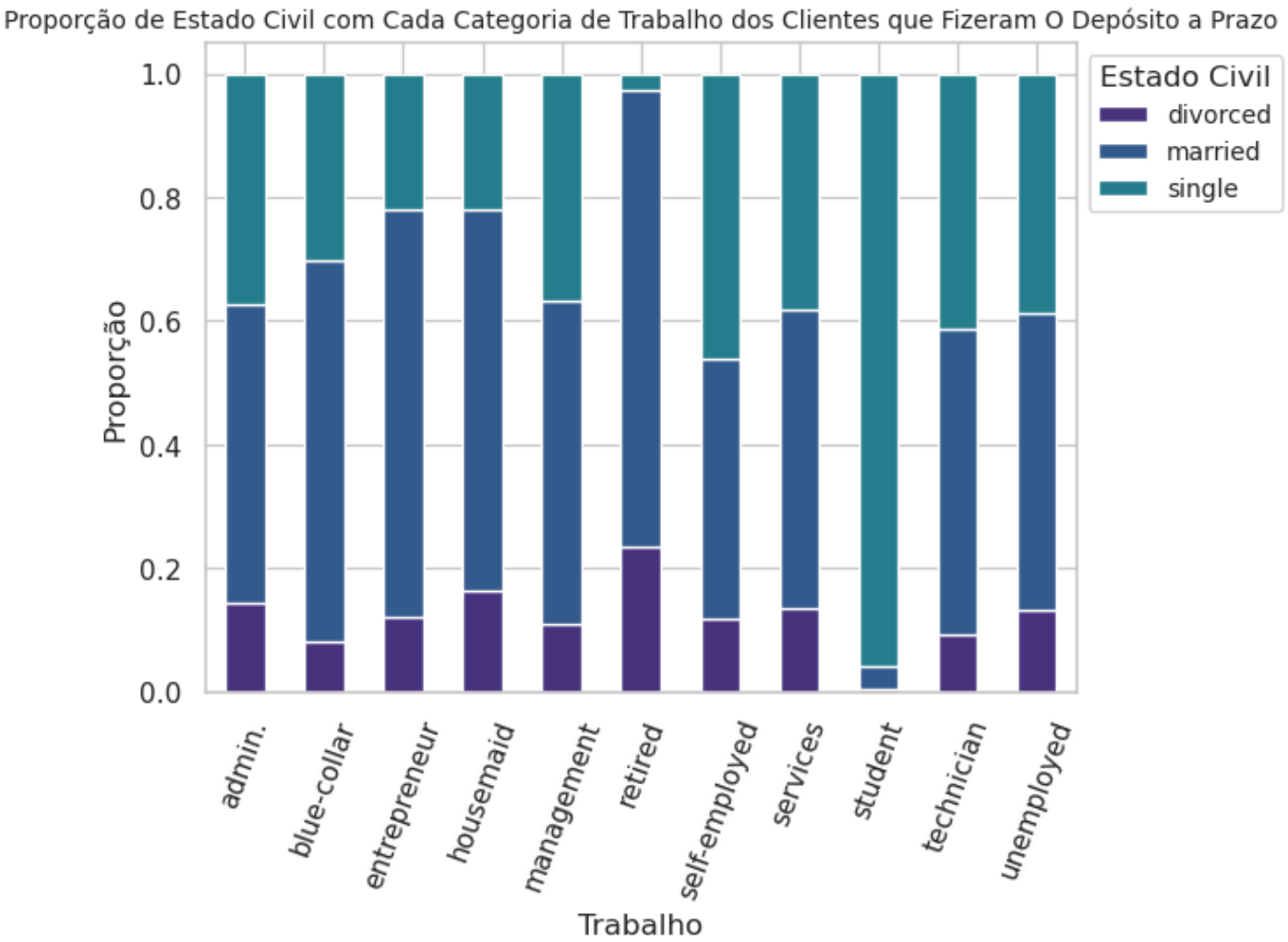
Analisando os empregos com maior aceitação do produto oferecido pelo telemarketing, podemos dizer que os gerentes e administradores têm um equilíbrio maior entre casados e solteiros. Já as pessoas que fazem trabalhos manuais (blue-collar) apresentam uma maior quantidade de casados e são a segunda ocupação com menos pessoas divorciadas.

Essas informações são valiosas para entender o perfil dos clientes e ajustar as estratégias de marketing de forma mais direcionada e eficaz. Com essas análises, a instituição financeira pode personalizar suas campanhas de telemarketing, aumentando a probabilidade de sucesso ao adaptar as abordagens às características específicas de cada grupo de clientes.

Para obter uma melhor probabilidade de sucesso, podemos verificar a mesma proporção, mas para os clientes que aceitaram fazer o depósito a prazo, conforme mostrado no gráfico abaixo.

In [ ]:
```
#Criando uma tabela cruzada dos clientes que aceitaram o produto
job_marital_y_counts = pd.crosstab(X[y.values == 'yes']['job'], X[y.values == 'yes']['marital'])
job_marital_y_counts_normalized = job_marital_y_counts.div(job_marital_y_counts.sum(axis=1), axis=0)

# Plotar gráfico de proporção dos clientes que aceitaram o produto
sns.set_theme(style="whitegrid", palette='viridis')
job_marital_y_counts_normalized.plot(kind='bar', stacked=True)
plt.xlabel('Trabalho')
plt.ylabel('Proporção')
plt.title('Proporção de Estado Civil com Cada Categoria de Trabalho dos Clientes que Fizeram O Depósito a Prazo', fontsize
plt.legend(title='Estado Civil', bbox_to_anchor=(1, 1), loc='upper left', fontsize='small')
plt.xticks(rotation=70)
plt.show()
```



Como podemos ver no gráfico acima, não há uma mudança significativa na proporção do estado civil entre os clientes que fizeram o depósito a prazo. Sendo assim, o banco poderia focar mais nos clientes que trabalham como administradores, trabalhadores manuais, gerentes e técnicos, e que são casados ou solteiros. Como vimos anteriormente, esses são os tipos de clientes que têm maior sucesso com as campanhas de telemarketing, ao considerar apenas essas duas variáveis.

Essa abordagem permitirá que a instituição financeira direcione seus esforços de marketing de maneira mais eficiente, aumentando a probabilidade de conversão e, consequentemente, a eficácia das suas campanhas de telemarketing.

Educação

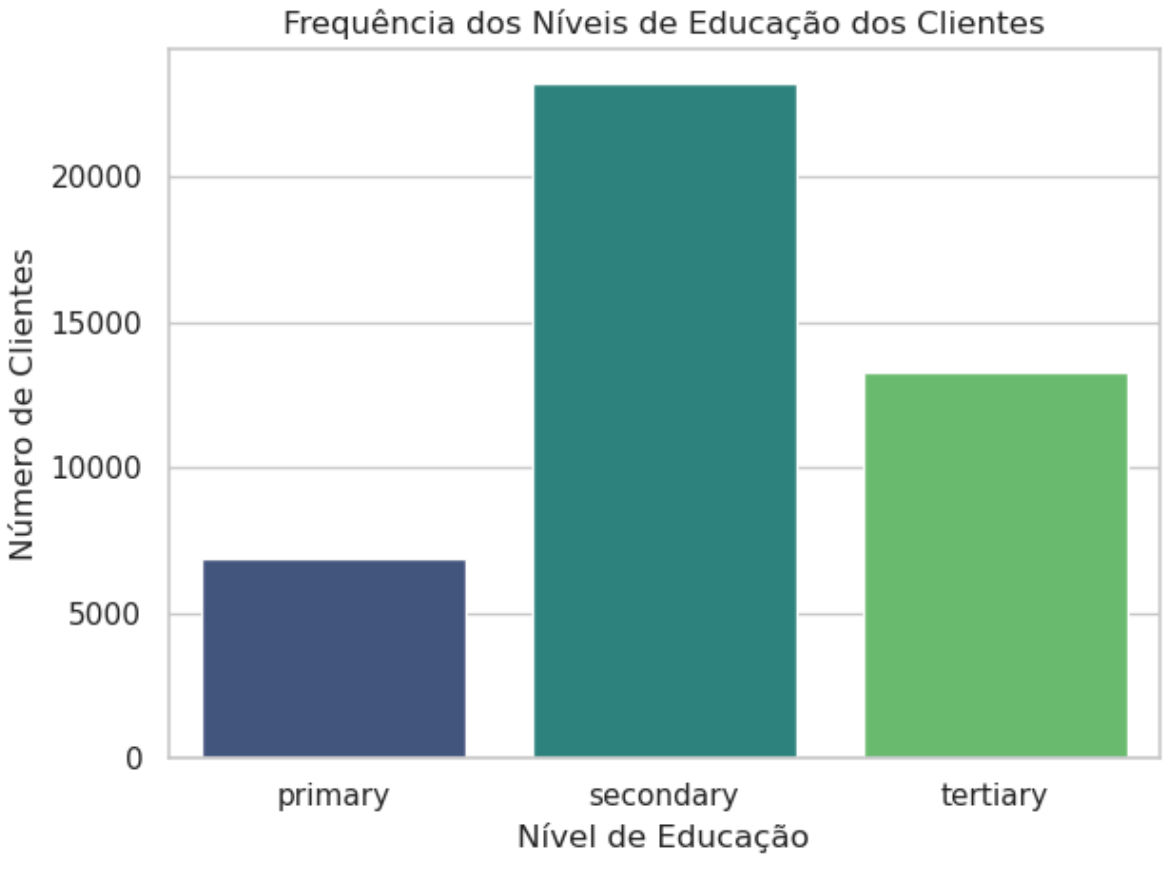
A variável *education* que representa o nível de educação do cliente pode apresentar 4 categorias, sendo elas:

```
Primary: illiterate, basic.4y, basic.6y e basic.9y
Secondary: high.school
tertiary: professional.course, university.degree
Unknown: Desconhecido
```

In [ ]:
```
# Criando uma tabela de frequência
education_counts = X['education'].value_counts().sort_index()
education_percentages = (education_counts * 100 / sum(education_counts.values))
print(education_percentages)
```

```
education
primary     15.8025
secondary   53.5176
tertiary    30.6800
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia
sns.barplot(x=education_counts.index, y=education_counts.values, palette='viridis')
plt.title('Frequência dos Níveis de Educação dos Clientes')
plt.xlabel('Nível de Educação')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```



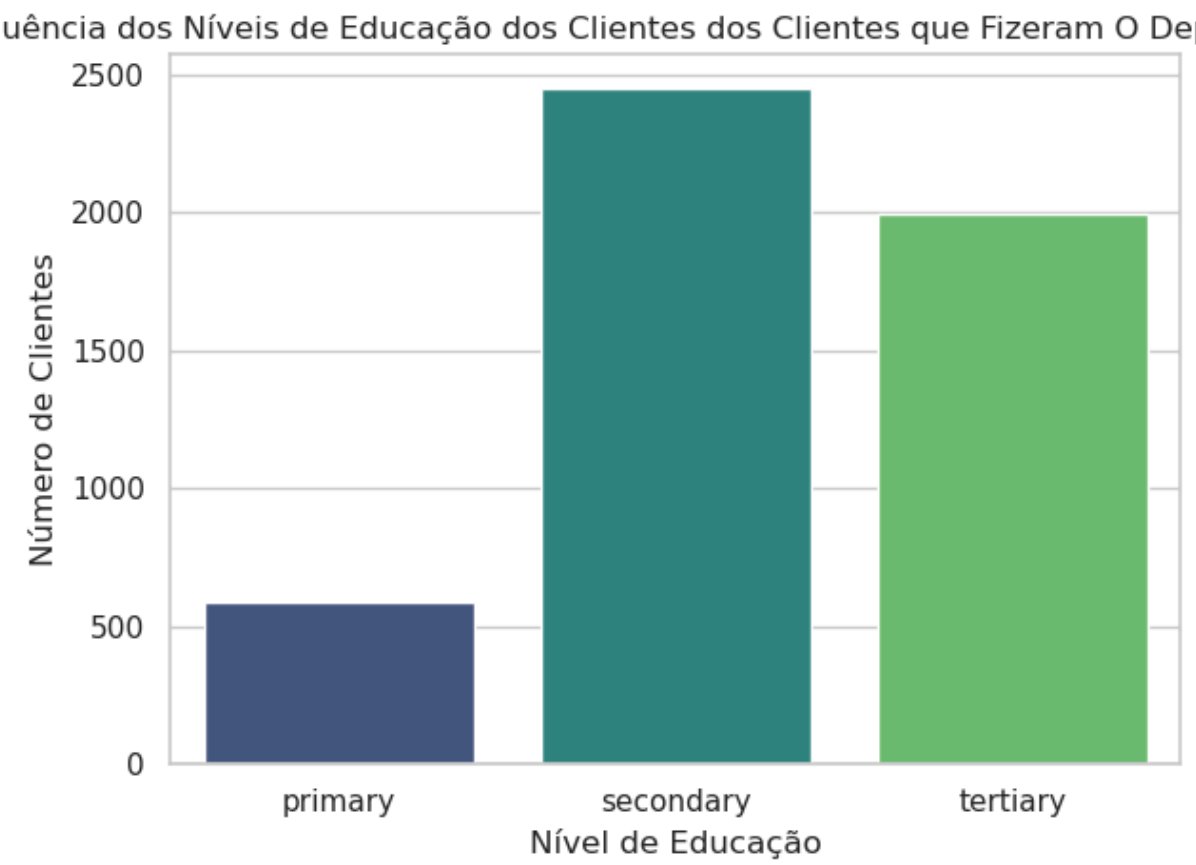
Como podemos observar na tabela de percentuais e no gráfico de frequência, aproximadamente 53,52% dos clientes possuem um nível de educação secundário (secondary). De acordo com os dados da tabela de variáveis, definimos que o nível de educação secundário corresponde a pessoas que completaram o ensino médio (*high school*). A segunda categoria mais prevalente na base de dados são os clientes com um nível de educação terciário (*tertiary*), que consideramos como pessoas que possuem um diploma universitário ou que tem um curso profissional, representando 30,68% dos clientes.

Como já mencionado, a proposta desta etapa do projeto é entender melhor quais tipos de clientes apresentam conversão por meio do marketing direto. No entanto, também podemos analisar a frequência dos níveis de educação entre os clientes que aceitaram fazer o depósito a prazo.

In [ ]:
```
# Criando uma tabela de frequência dos clientes que aceitaram o produto
education_y_counts = X[y.values == 'yes']['education'].value_counts().sort_index()
education_y_percentages = (education_y_counts * 100 / sum(education_y_counts.values))
print(education_y_percentages)
```

```
education
primary     11.7332
secondary   48.6401
tertiary    39.6268
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia dos clientes que aceitaram o produto
sns.barplot(x=education_y_counts.index, y=education_y_counts.values, palette='viridis')
plt.title('Frequência dos Níveis de Educação dos Clientes dos Clientes que Fizeram O Depósito a Prazo')
plt.xlabel('Nível de Educação')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```



Ao analisarmos o nível de educação dos clientes que aceitaram fazer o depósito a prazo, observamos uma mudança significativa. A distribuição entre clientes com nível de educação terciário e secundário se torna mais equilibrada, representando aproximadamente 39,63% e 48,64%, respectivamente. Esse equilíbrio pode estar relacionado ao perfil dos clientes mais propensos à conversão por meio do telemarketing, que inclui administradores,

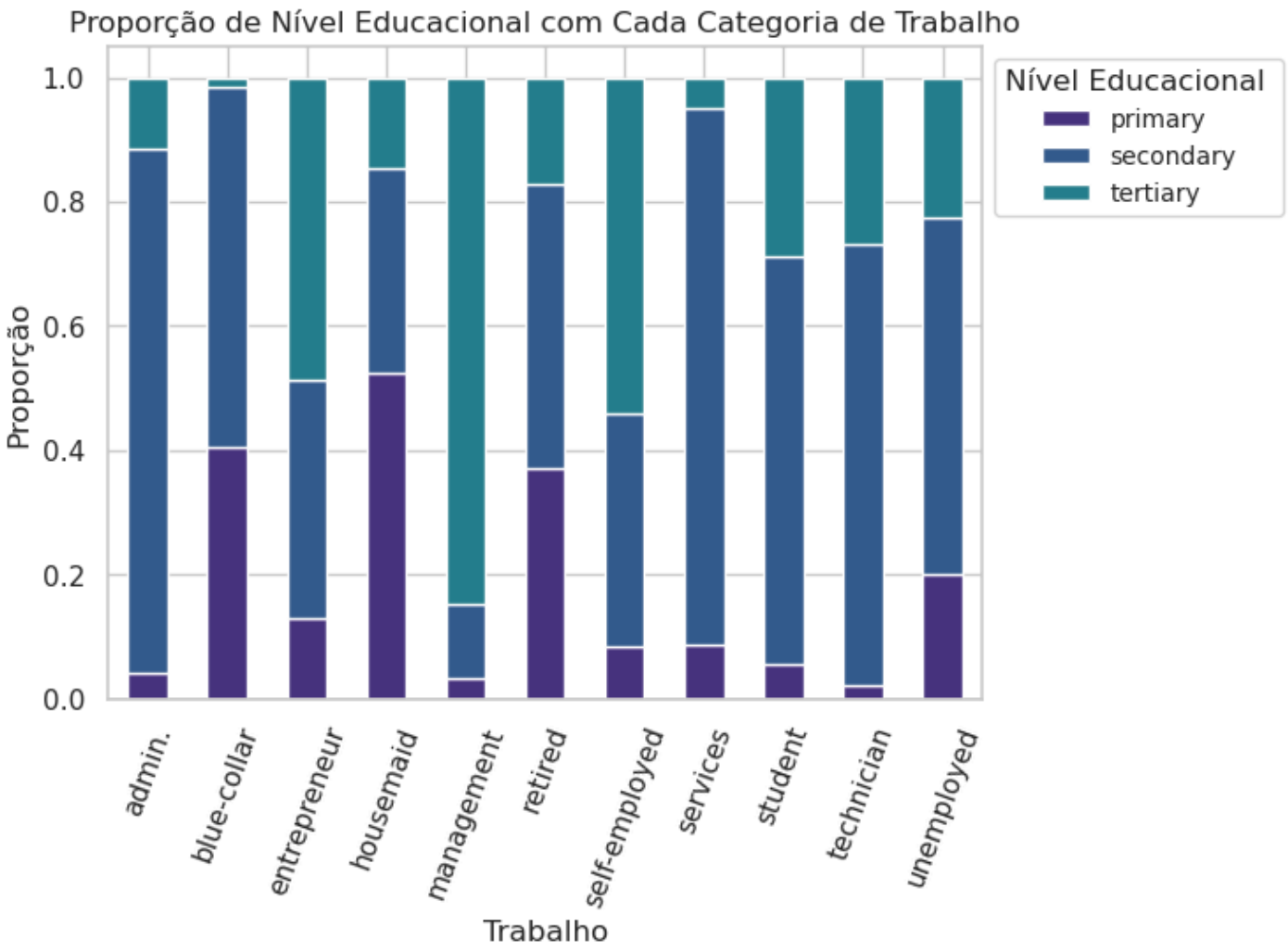
técnicos, gerentes, trabalhadores manuais e aposentados. Normalmente, trabalhadores manuais, que podem ser menos favorecidos economicamente, têm um nível de educação mais baixo, enquanto os profissionais das outras categorias mencionadas geralmente possuem níveis de educação superior.

Nível de Educação e Trabalho

Para verificar se as afirmações mencionadas são de fato reais, podemos utilizar o mesmo gráfico de proporção empregado na seção anterior. Esse gráfico mostrará a distribuição de cada nível de educação em relação às diferentes categorias de trabalho.

In [ ]:
```
#Criando uma tabela cruzada
job_education_counts = pd.crosstab(X['job'], X['education'])
job_education_counts_normalized = job_education_counts.div(job_education_counts.sum(axis=1), axis=0)

# Plotar gráfico de proporção
sns.set_theme(style="whitegrid", palette='viridis')
job_education_counts_normalized.plot(kind='bar', stacked=True)
plt.xlabel('Trabalho')
plt.ylabel('Proporção')
plt.title('Proporção de Nível Educacional com Cada Categoria de Trabalho')
plt.legend(title='Nível Educacional ', bbox_to_anchor=(1, 1), loc='upper left', fontsize='small')
plt.xticks(rotation=70)
plt.show()
```

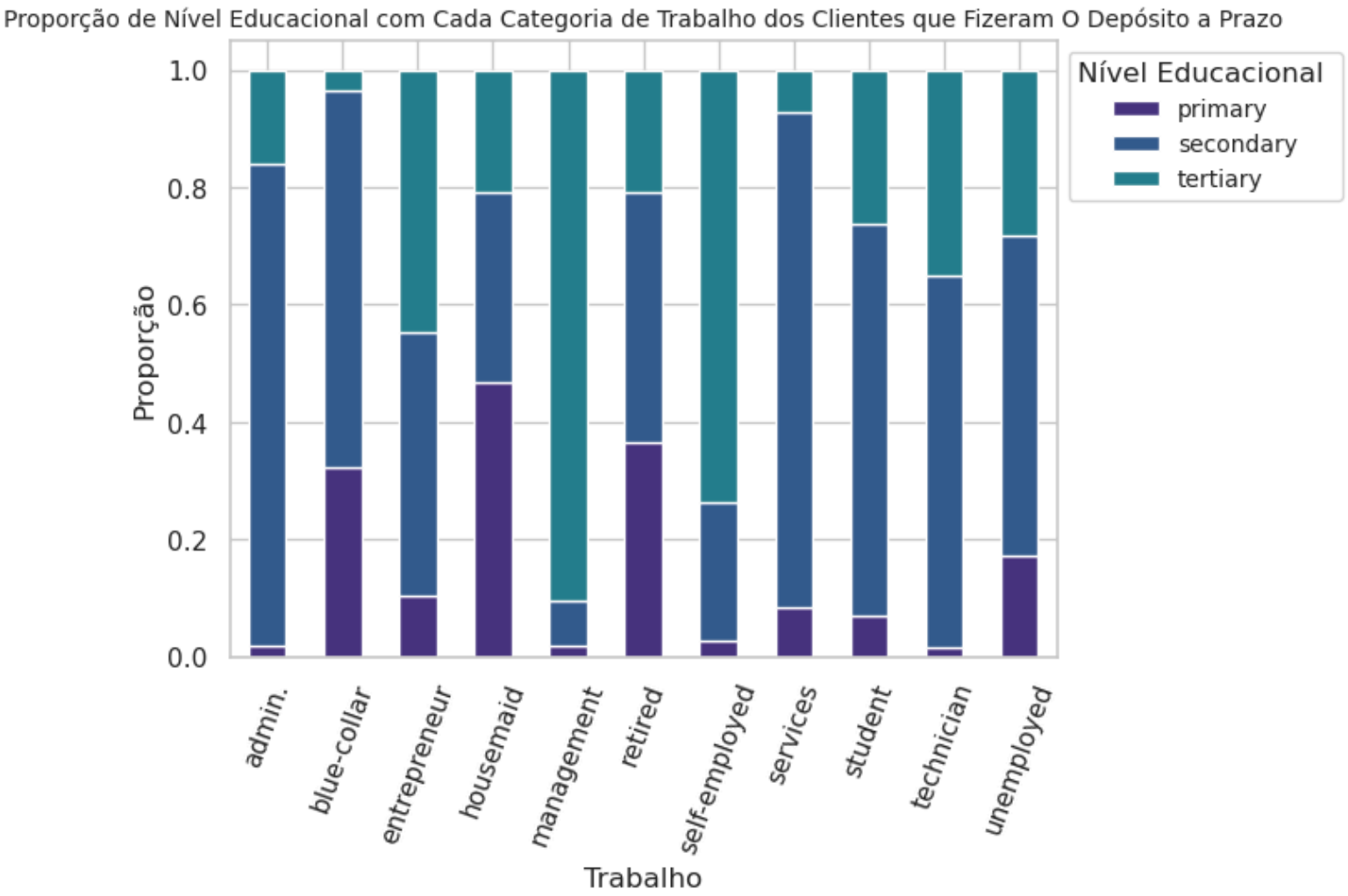


Como era de se esperar, os trabalhadores manuais e os trabalhadores domésticos são as categorias com maior proporção de pessoas com nível de educação primária, representando aproximadamente 40% e 50%, respectivamente. Por outro lado, gerentes, empresário e trabalhadores autônomos são os clientes com a maior proporção de nível de educação terciário. É interessante notar que os aposentados apresentam uma distribuição mais equilibrada entre as três classificações de nível educacional. As demais categorias de trabalho tendem a ter uma maior proporção de clientes com nível de educação secundário.

Além disso também podemos visualizar abaixo a mesma proporção mas levando em conta os clientes que aceitaram o produto.

In [ ]:
```
#Criando uma tabela cruzada dos clientes que aceitaram o produto
job_education_y_counts = pd.crosstab(X[y.values == 'yes']['job'], X[y.values == 'yes']['education'])
job_education_y_counts_normalized = job_education_y_counts.div(job_education_y_counts.sum(axis=1), axis=0)

# Plotar gráfico de proporção dos clientes que aceitaram o produto
sns.set_theme(style="whitegrid", palette='viridis')
job_education_y_counts_normalized.plot(kind='bar', stacked=True)
plt.xlabel('Trabalho')
plt.ylabel('Proporção')
plt.title('Proporção de Nível Educacional com Cada Categoria de Trabalho dos Clientes que Fizeram O Depósito a Prazo', fon
plt.legend(title='Nível Educacional ', bbox_to_anchor=(1, 1), loc='upper left', fontsize='small')
plt.xticks(rotation=70)
plt.show()
```



A diferença mais significativa entre os clientes que aceitaram fazer o depósito a prazo foi observada na classe dos trabalhadores autônomos, que apresentou uma queda considerável na proporção de pessoas com nível de educação secundário. Outro ponto interessante é que, em todas as categorias de trabalho, houve uma redução ou estagnação na proporção de clientes com nível de educação primário. Esse fenômeno será analisado mais detalhadamente quando considerarmos os salários desses clientes, pois geralmente indivíduos com menor nível educacional tendem a ter salários mais baixos e, consequentemente, optam por gastar seu capital disponível em bens essenciais, ao invés de investir.

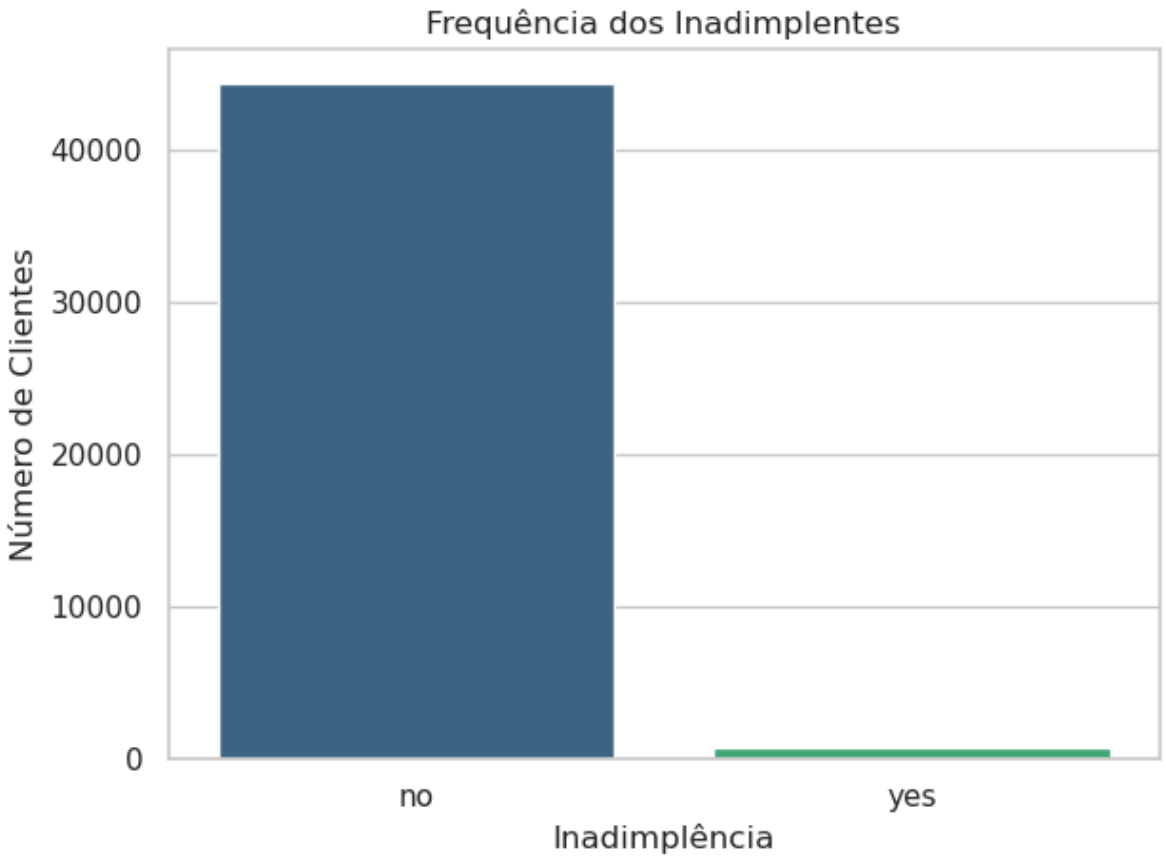
Inadimplência

A variável *default* é binária, com valores *no* ou *yes*, indicando se o cliente possui algum crédito inadimplente.

In [ ]:
```
# Criando uma tabela de frequência
inadimplencia_counts = X['default'].value_counts().sort_index()
inadimplencia_percentages = (inadimplencia_counts * 100) / sum(inadimplencia_counts)
print(inadimplencia_percentages)
```

```
default
no    98.1973
yes    1.8027
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia
sns.barplot(x=inadimplencia_counts.index, y=inadimplencia_counts.values, palette='viridis')
plt.title('Frequência dos Inadimplentes')
plt.xlabel('Inadimplência')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```



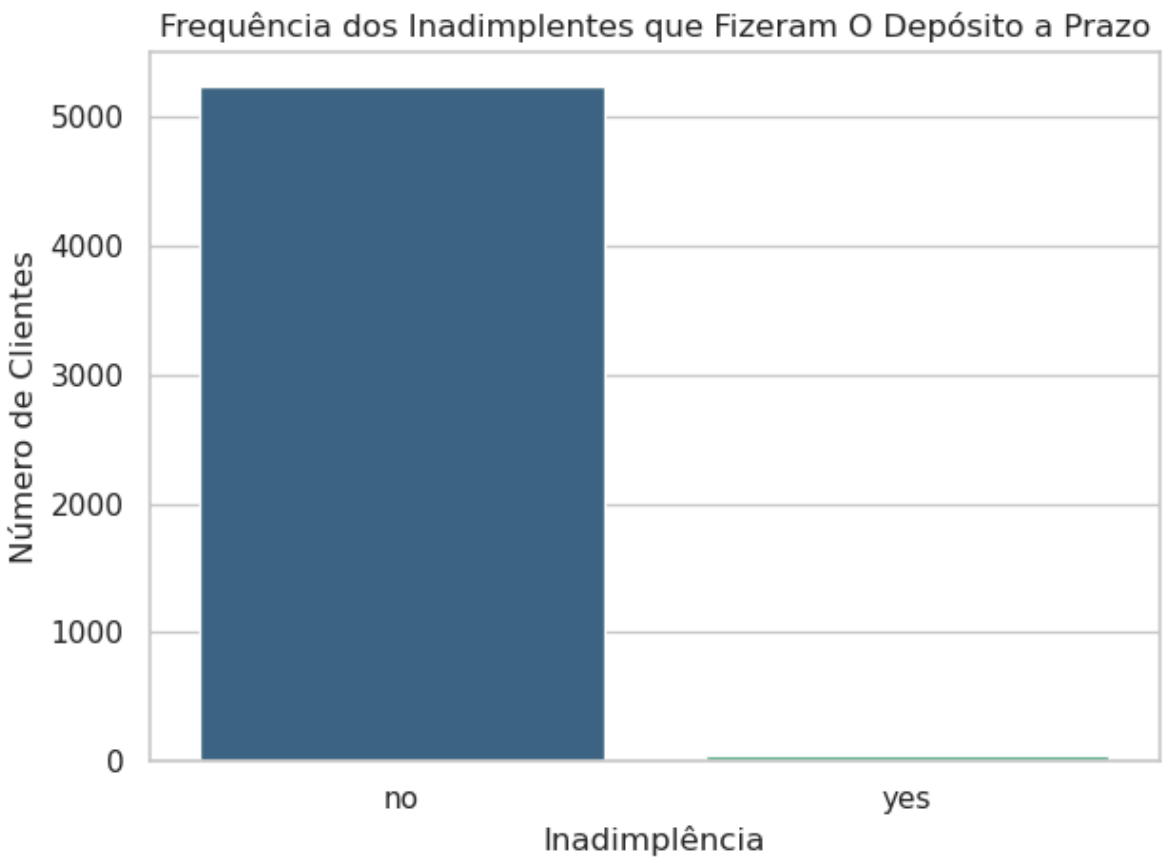
Ao analisar a tabela e o gráfico de barras dos inadimplentes, podemos observar um desequilíbrio significativo entre as classes, com aproximadamente 98,2% dos clientes sem inadimplência. Devido a esse desbalanceamento, é recomendável usar técnicas de oversampling para equilibrar a classe minoritária (inadimplentes) com a classe majoritária (não inadimplentes). Isso ajudará a garantir que o algoritmo de aprendizado de máquina não se concentre apenas nos casos de não inadimplência, proporcionando um modelo mais equilibrado e eficaz.

Abaixo, podemos ver o percentual de inadimplentes que aceitaram o depósito a prazo.

In [ ]:
```
# Criando uma tabela de frequência dos clientes que aceitaram o produto
inadimplencia_y_counts = X[y.values == 'yes']['default'].value_counts().sort_index()
inadimplencia_y_percentages = (inadimplencia_y_counts * 100) / sum(inadimplencia_y_counts)
print(inadimplencia_y_percentages)
```

```
default
no    99.0168
yes    0.9832
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia
sns.barplot(x=inadimplencia_y_counts.index, y=inadimplencia_y_counts.values, palette='viridis')
plt.title('Frequência dos Inadimplentes que Fizeram O Depósito a Prazo')
plt.xlabel('Inadimplência')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```



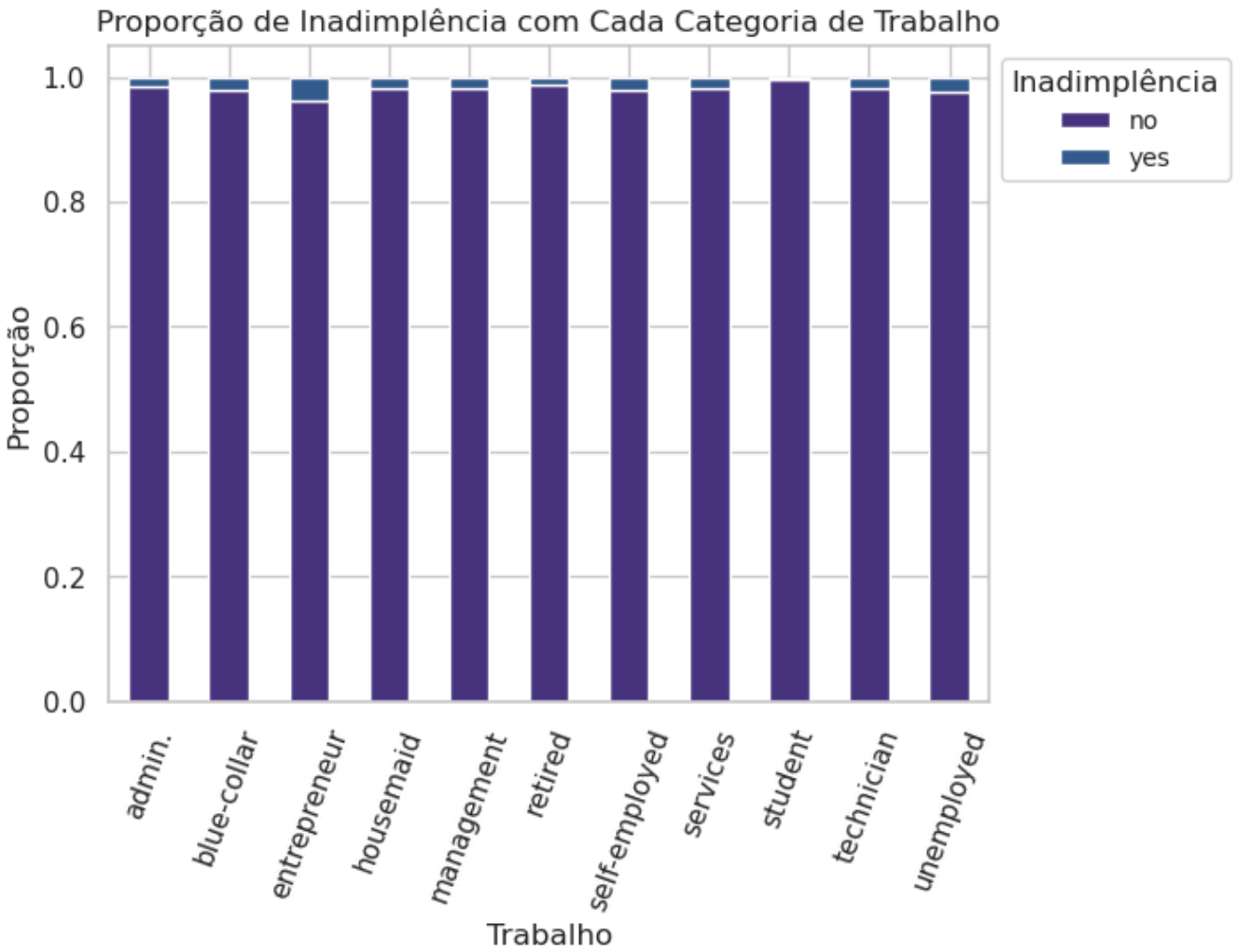
Ao analisar os clientes que aceitaram fazer o depósito a prazo, observamos uma situação ainda mais desafiadora, com os casos de inadimplência caindo praticamente pela metade, representando apenas 0,98% das observações. Esse fenômeno pode estar relacionado a um padrão similar ao identificado na seção anterior, onde indivíduos com menor capital disponível e que precisam pagar suas contas estão menos inclinados a optar por um investimento oferecido pelo banco.

Inadimplência e Trabalho

Para entender melhor os perfis dos clientes com os quais estamos lidando, é interessante analisar quais são as ocupações dos clientes e qual a proporção de inadimplentes em cada categoria de trabalho.

In [ ]:
```
#Criando uma tabela cruzada
job_default_counts = pd.crosstab(X['job'], X['default'])
job_default_counts_normalized = job_default_counts.div(job_default_counts.sum(axis=1), axis=0)

# Plotar gráfico de proporção
sns.set_theme(style="whitegrid", palette='viridis')
job_default_counts_normalized.plot(kind='bar', stacked=True)
plt.xlabel('Trabalho')
plt.ylabel('Proporção')
plt.title('Proporção de Inadimplência com Cada Categoria de Trabalho')
plt.legend(title='Inadimplência', bbox_to_anchor=(1, 1), loc='upper left', fontsize='small')
plt.xticks(rotation=70)
plt.show()
```



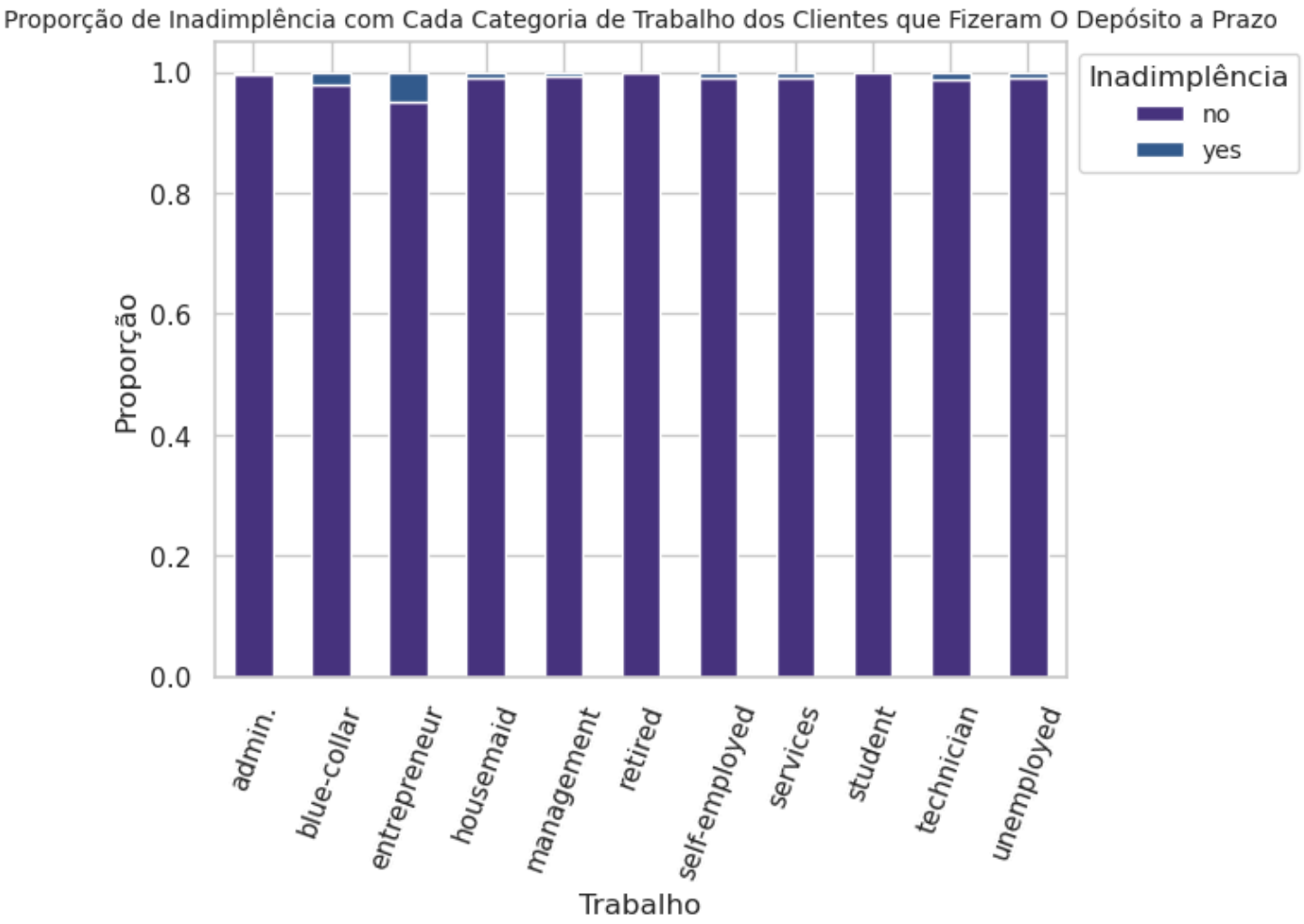
No gráfico acima, podemos observar um fenômeno interessante e comum em economias: pessoas com maior inadimplência são frequentemente empresários, que se expõem a riscos em busca de retorno financeiro. No entanto, um dado preocupante é que pessoas desempregadas também apresentam um grau significativo de inadimplência, sendo a segunda categoria mais inadimplente. Isso pode ser explicado pelo fato de que, em um sistema onde tudo gira em torno do capital, indivíduos desempregados frequentemente precisam se endividar para sobreviver.

Em contrapartida, é encorajador ver que os estudantes não apresentam inadimplência. Isso é particularmente interessante em uma economia onde os estudantes conseguem estudar sem precisar se endividar. No entanto, surge um questionamento: será que as pessoas menos favorecidas estão conseguindo estudar? Ou será que elas só conseguiriam estudar se obtivessem crédito para financiar seus estudos? E pelo fato de serem mais propensas à inadimplência, acabam não conseguindo obter esse crédito?

Como já estamos fazendo em nossa metodologia de análise, a seguir verificaremos a mesma proporção, mas focando nos clientes que aceitaram fazer o depósito a prazo.

In [ ]:
```
#Criando uma tabela cruzada dos clientes que aceitaram o produto
job_default_y_counts = pd.crosstab(X[y.values == 'yes']['job'], X[y.values == 'yes']['default'])
job_default_y_counts_normalized = job_default_y_counts.div(job_default_y_counts.sum(axis=1), axis=0)

# Plotar gráfico de proporção dos clientes que aceitaram o produto
sns.set_theme(style="whitegrid", palette='viridis')
job_default_y_counts_normalized.plot(kind='bar', stacked=True)
plt.xlabel('Trabalho')
plt.ylabel('Proporção')
plt.title('Proporção de Inadimplência com Cada Categoria de Trabalho dos Clientes que Fizeram O Depósito a Prazo', fontsiz
plt.legend(title='Inadimplência', bbox_to_anchor=(1, 1), loc='upper left', fontsize='small')
plt.xticks(rotation=70)
plt.show()
```



Já sabíamos que a maioria das categorias de trabalho teria uma redução no número de pessoas inadimplentes, com algumas classes apresentando até mesmo nenhum inadimplente entre os clientes que aceitaram fazer o depósito a prazo. No entanto, há um aspecto interessante observado entre os empresários, que experimentaram um aumento na inadimplência. Isso pode estar relacionado à crise financeira de 2008, que levou à redução das taxas de juros. Com taxas de juros mais baixas, os empresários podem ter visto uma oportunidade para pegar dinheiro emprestado e investir, tornando mais atraente manter o dinheiro aplicado do que pagar a dívida.

Equilíbrio

A variável *balance* representa o saldo médio anual em euros, considerando que a base de dados é de um banco português. Portanto, essa variável é expressa em valores inteiros.

In [ ]:
```
# Resumo estatístico
X["balance"].describe()
```

Out[ ]:
```
count    45211.0000
mean      1362.2721
std       3044.7658
min      -8019.0000
25%         72.0000
50%        448.0000
75%       1428.0000
max     102127.0000
Name: balance, dtype: float64
```

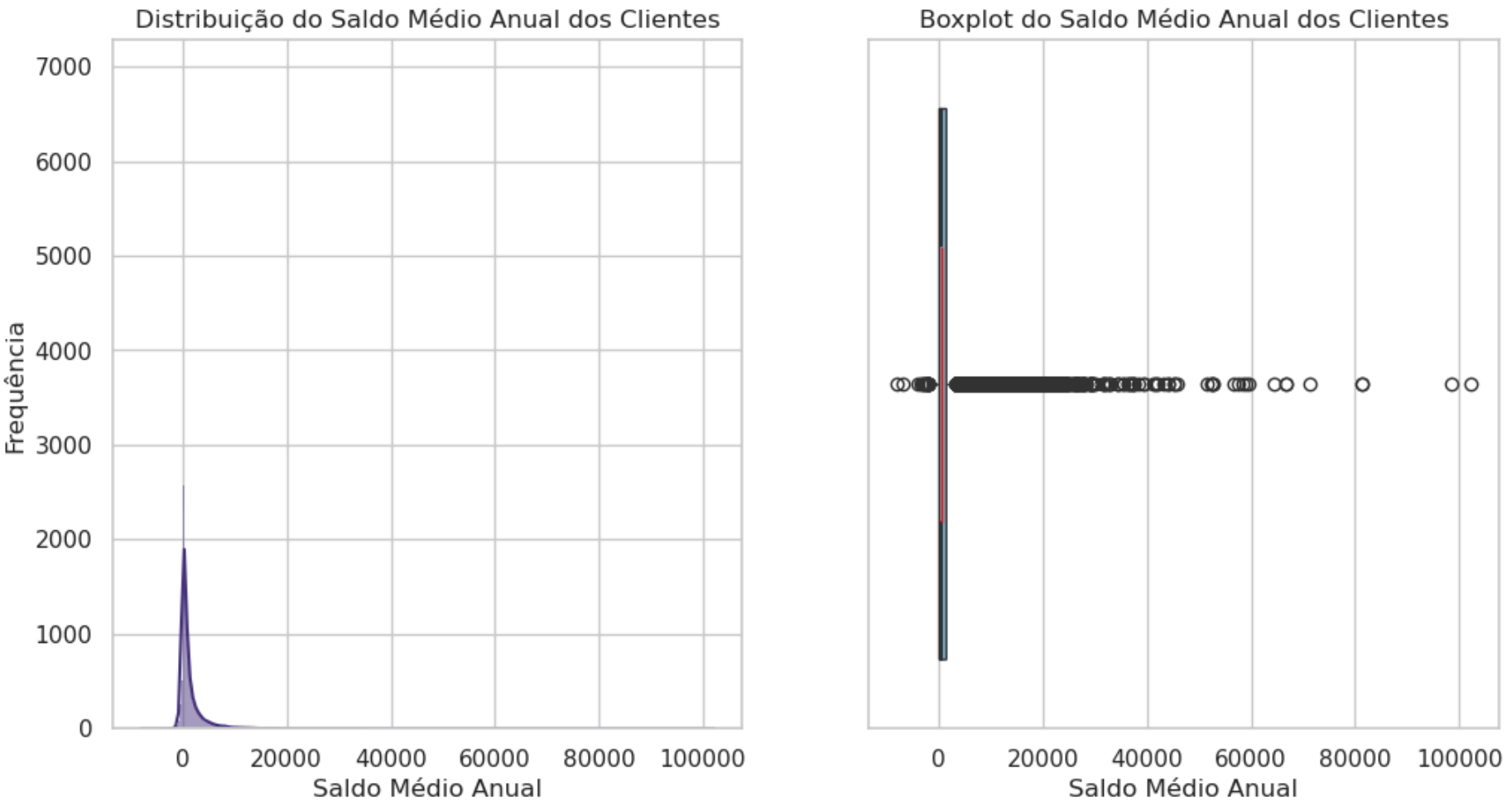
In [ ]:
```
# Visualizando a moda
X["balance"].mode()
```

Out[ ]:
```
0    0
Name: balance, dtype: int64
```

Como pode ser observado na tabela acima, o saldo médio anual (média) é de aproximadamente 1362,27 euros, enquanto a mediana é de 448 euros, e a moda é 0. Portanto, a média não é uma representação precisa da distribuição dos saldos, sendo elevada por possíveis outliers. Isso é evidenciado pelo valor máximo de 102.127,00 euros e pelo valor mínimo de -8.019,00 euros.

A diferença significativa entre a média e a mediana indica uma distribuição assimétrica positiva, onde a média é praticamente três vezes maior que a mediana. A presença de outliers, particularmente no extremo superior, contribui para essa assimetria.

In [ ]:
```
# Criação do grafico histograma e boxplot
plt.figure(figsize=(12,6))
plt.subplot(1, 2, 1)
sns.histplot(X["balance"], kde=True)
plt.title("Distribuição do Saldo Médio Anual dos Clientes")
plt.xlabel("Saldo Médio Anual")
plt.ylabel("Frequência")
plt.subplot(1, 2, 2)
sns.boxplot(X["balance"], orient='h', notch=True, showcaps=False,
            boxprops={"facecolor": (0, .5, .7, .5)},
            medianprops={"color": "r", "linewidth": 1})
plt.title("Boxplot do Saldo Médio Anual dos Clientes")
plt.xlabel("Saldo Médio Anual")
plt.show()
```

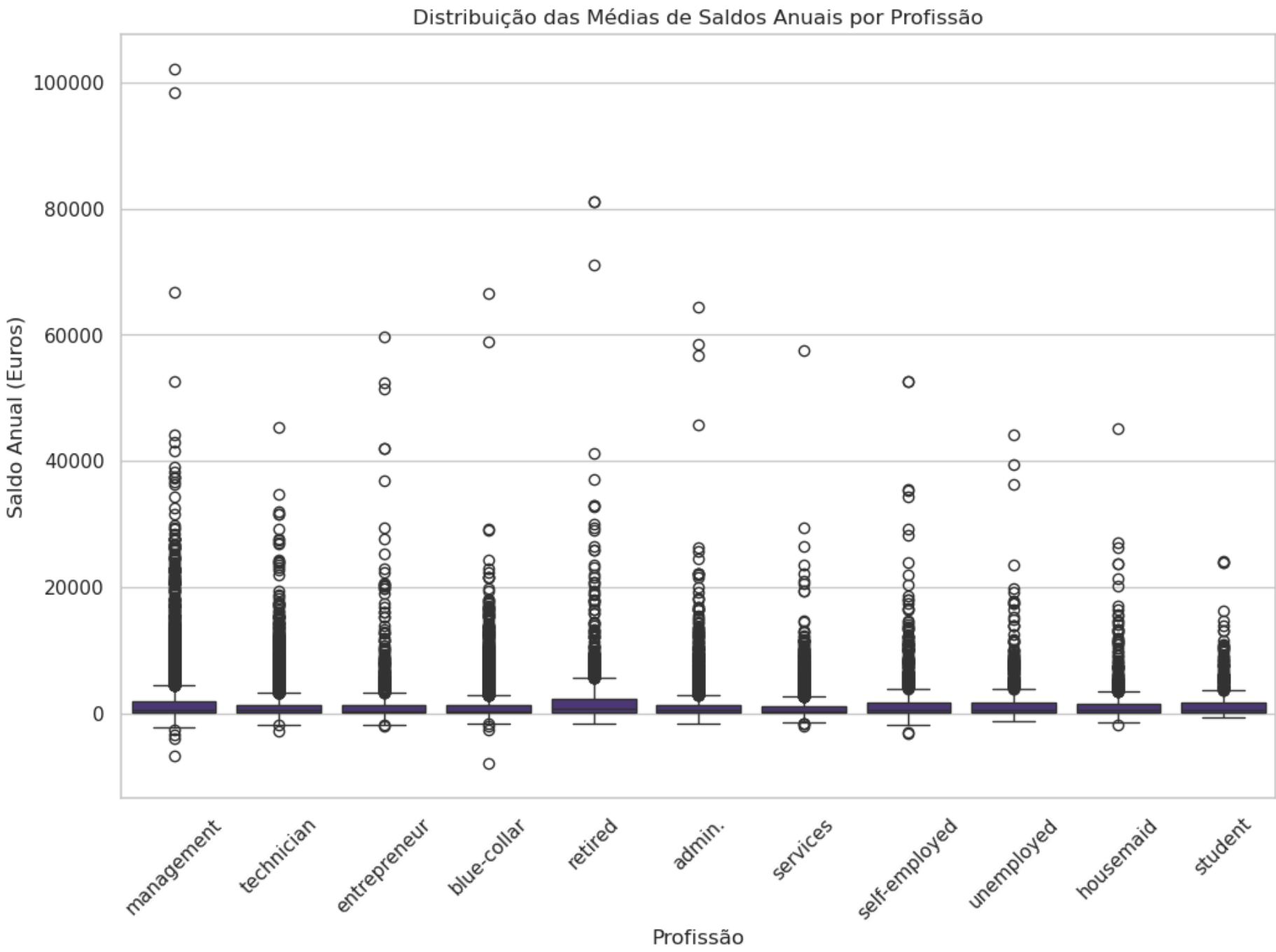


Como podemos ver nos gráficos acima, a distribuição dos saldos anuais dos clientes é bastante assimétrica, com muitos valores altos além do terceiro quartil contribuindo para essa assimetria. Portanto, será extremamente importante aplicar alguma transformação nessa variável para tentar corrigir tanto a sua assimetria quanto reduzir o impacto dos outliers.

Equilíbrio e Trabalho

A analise feita abaixo representa a distribuição dos saldos anuais por profissão.

In [ ]:
```
# Gráfico das médias de saldos anuais por profissação
plt.figure(figsize=(12, 8))
sns.boxplot(x='job', y='balance', data=X)
plt.xticks(rotation=45)  # Rotaciona os rótulos do eixo X para melhor visualização
plt.title('Distribuição das Médias de Saldos Anuais por Profissão')
plt.xlabel('Profissão')
plt.ylabel('Saldo Anual (Euros)')
plt.show()
```

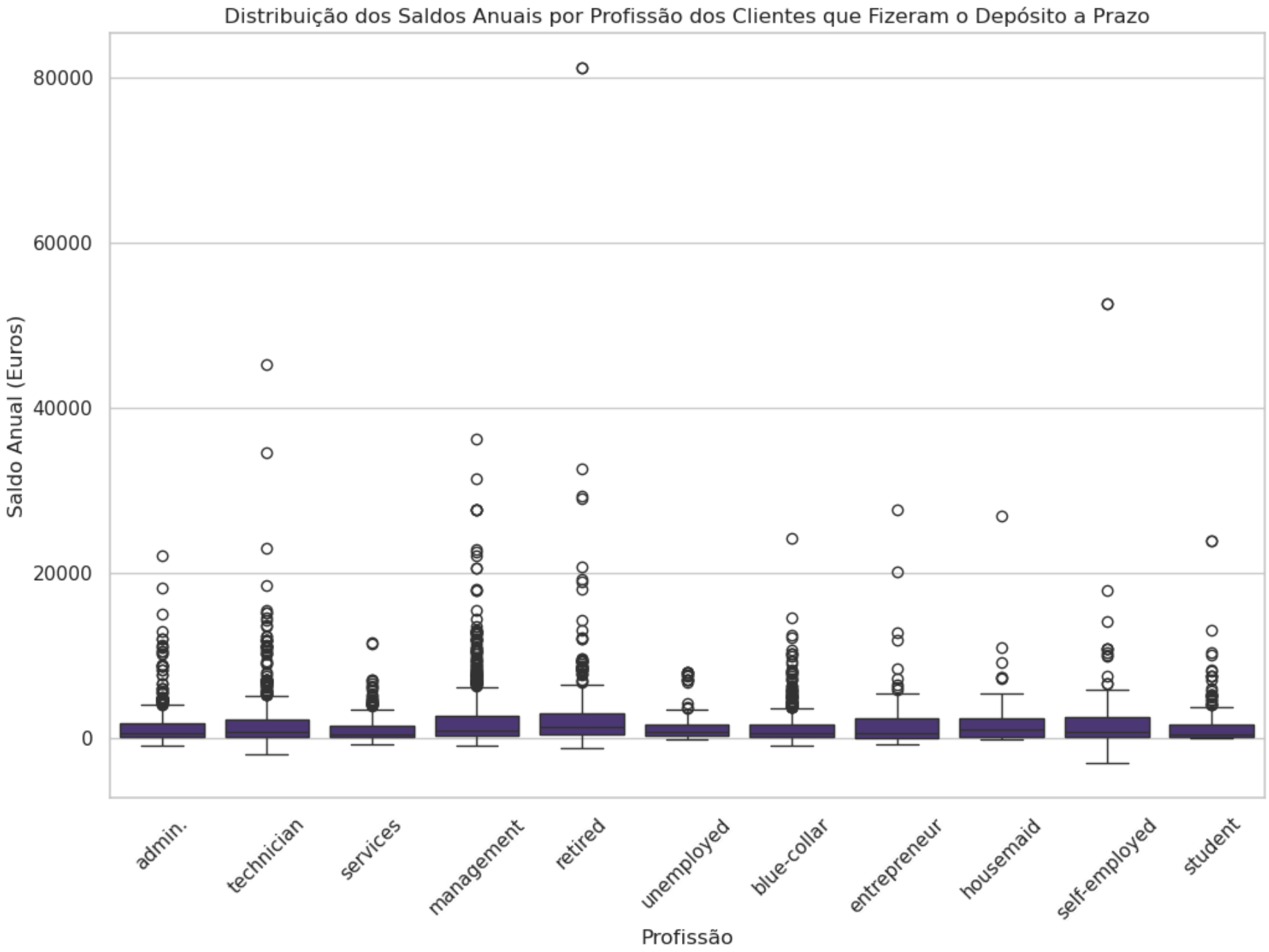


Como podemos ver no gráfico acima, que contém diversos boxplots separados por profissão, conseguimos entender um pouco do comportamento do saldo anual médio por cada classe de profissão. Observamos que os gerentes são a classe que mais apresenta outliers possitivos, indicando que pessoas nessa profissão frequentemente têm saldos anuais mais elevados e variáveis.

Outro ponto interessante é a classe de trabalhadores manuais, onde encontramos o valor mínimo de -8.019 euros, mas também duas observações próximas de 60.000 euros. A categoria de trabalho que apresenta menor variabilidade nos saldos médios anuais são os estudantes, com outliers apenas à direita (positivos) e uma concentração maior. Isso ocorre porque muitos estudantes ainda são sustentados pelos pais ou ganham dinheiro através de pequenos empreendimentos e bolsas de pesquisa.

Uma visualização também interessante é criar a mesma análise, mas focando nos clientes que fizeram o investimento no banco. Isso nos permitirá entender melhor o comportamento do saldo anual médio por profissão entre os clientes que decidiram aceitar o depósito a prazo.

In [ ]:
```
# Gráfico das médias de saldos anuais por profissação dos clientes que fizeram o depósito a prazo
plt.figure(figsize=(12, 8))
sns.boxplot(x='job', y='balance', data=X[y.values == "yes"])
plt.xticks(rotation=45)  # Rotaciona os rótulos do eixo X para melhor visualização
plt.title('Distribuição dos Saldos Anuais por Profissão dos Clientes que Fizeram o Depósito a Prazo')
plt.xlabel('Profissão')
plt.ylabel('Saldo Anual (Euros)')
plt.show()
```



Podemos observar uma melhora na distribuição dos dados quando analisamos os clientes que fizeram o depósito a prazo, apresentando uma redução dos outliers. Agora, percebemos que existem apenas três valores acima da média de saldo anual de 40 mil euros. Isso pode ocorrer porque essas pessoas, que têm uma boa renda, fazem melhores escolhas de investimento.

Além disso, como já mencionado indiretamente na análise de inadimplência, os clientes que fazem o depósito a prazo possuem uma renda maior. Por isso, o gráfico acima não apresenta nenhum outlier abaixo do primeiro bigode dos boxplots.

Interessantemente, a classe de trabalho de autônomos que aceita fazer o investimento possui o valor mais baixo de saldo médio. Isso pode ocorrer porque, muitas vezes, essas pessoas devem assumir um grau de risco por serem considerados empreendedores. Mesmo enfrentando prejuízos, elas ainda optam por fazer o investimento no banco.

Com base em toda essa análise feita por meio desse gráfico, podemos afirmar que a maioria dos clientes que aceitam fazer o depósito a prazo tem uma média anual de saldo entre 0 e 15.000 euros.

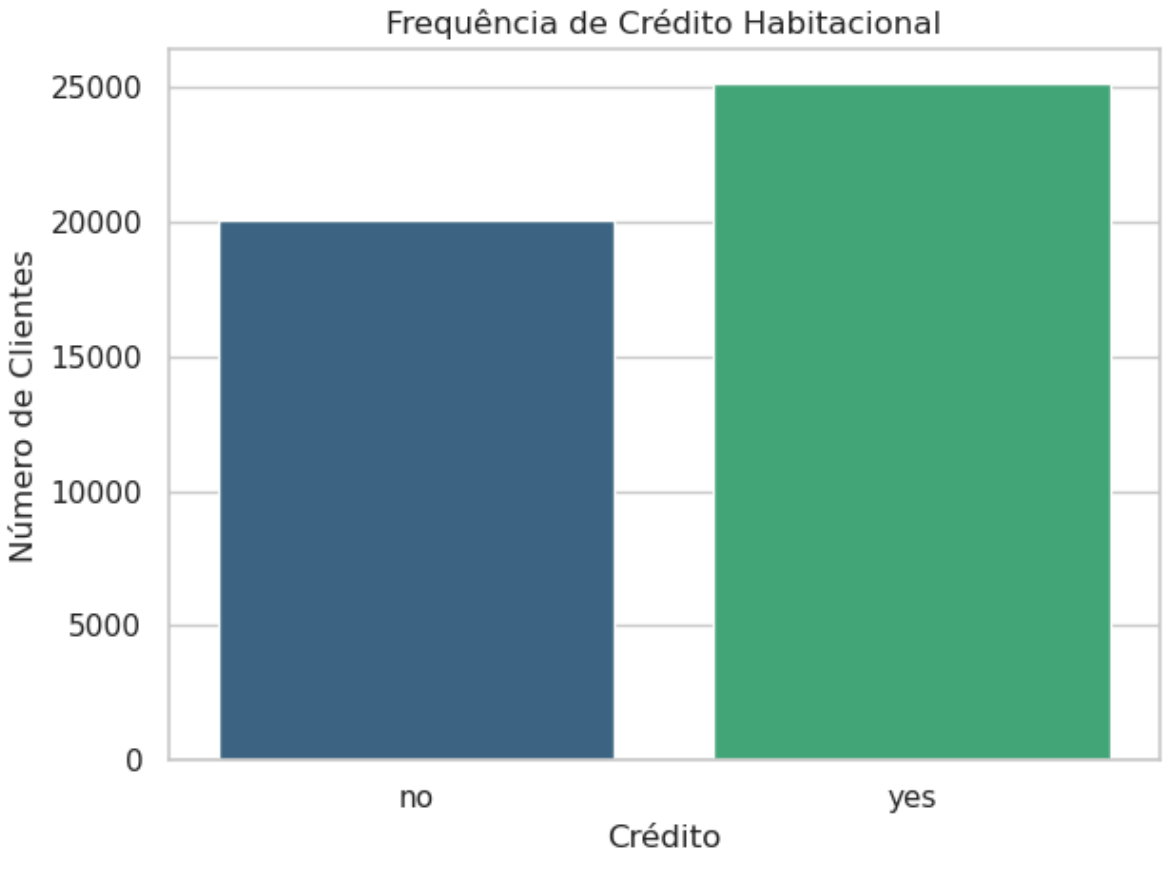
Habitação

A variável *housing* é uma variável binária que assume os valores *yes* ou *no*. Ela indica se o cliente possui um crédito de habitação.

In [ ]:
```
# Criando uma tabela de frequência
housing_counts = X['housing'].value_counts().sort_index()
housing_percentages= (housing_counts * 100) / sum(housing_counts)
print(housing_percentages)
```

```
housing
no    44.4162
yes   55.5838
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia
sns.barplot(x=housing_counts.index, y=housing_counts.values, palette='viridis')
plt.title('Frequência de Crédito Habitacional')
plt.xlabel('Crédito')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```

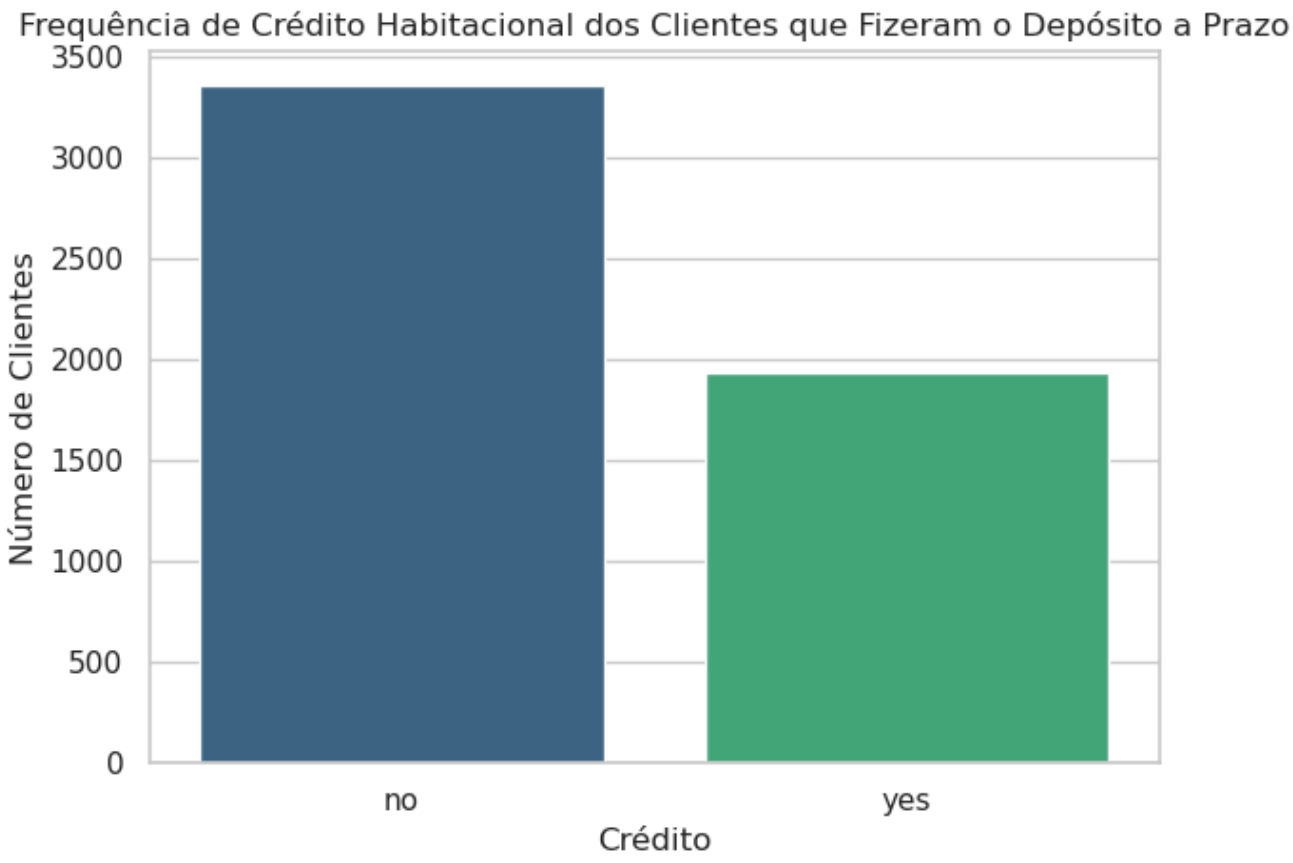


A variável *housing* apresenta um bom equilíbrio entre suas duas classes: aproximadamente 55,58% dos clientes possuem crédito de habitação, enquanto o restante não possui.

In [ ]:
```
# Criando uma tabela de frequência dos clientes que aceitaram o produto
housing_y_counts = X[y.values == 'yes']['housing'].value_counts().sort_index()
housing_y_percentages= (housing_y_counts * 100) / sum(housing_y_counts)
print(housing_y_percentages)
```

```
housing
no    63.4146
yes   36.5854
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia dos clientes que aceitaram o produto
sns.barplot(x=housing_y_counts.index, y=housing_y_counts.values, palette='viridis')
plt.title('Frequência de Crédito Habitacional dos Clientes que Fizeram o Depósito a Prazo')
plt.xlabel('Crédito')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```



Um fato interessante é que aproximadamente 63,41% dos clientes que aceitam fazer o investimento não possuem crédito habitacional. Isso pode ser explicado pelo fato de que clientes com crédito habitacional tendem a ter uma renda maior. Como vimos anteriormente, os clientes que aceitam fazer o depósito a prazo apresentam uma renda mediana, e notamos uma diminuição nos outliers positivos. Portanto, a falta de crédito habitacional entre esses clientes pode indicar que eles não têm renda tão alta quanto a dos clientes com crédito habitacional.

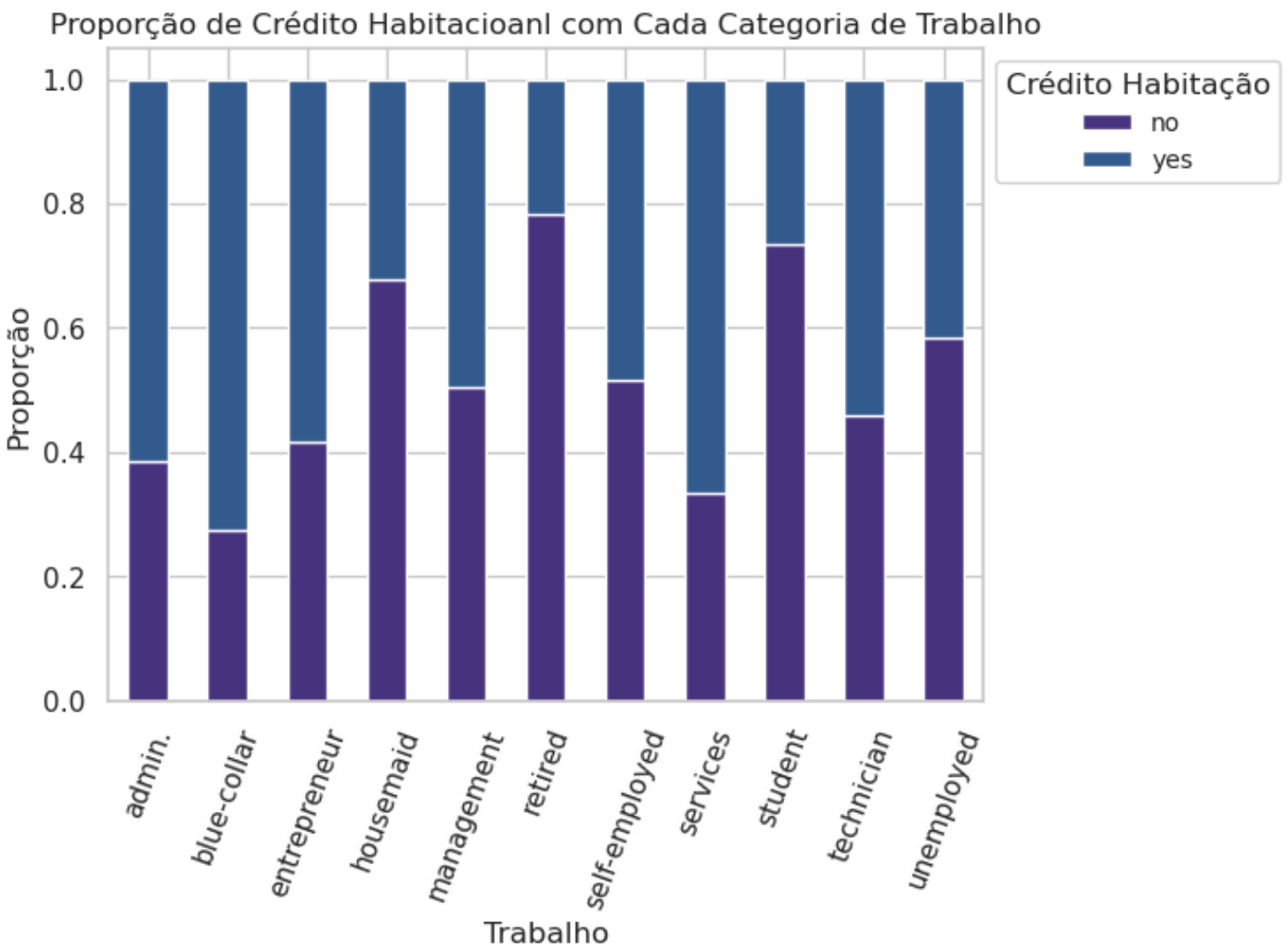
Para compreender melhor essa variável, é importante analisá-la em conjunto com outras variáveis.

Habitação e Trabalho

Abaixo, será realizada uma análise para verificar a distribuição de clientes com crédito de habitação por cada categoria de trabalho.

In [ ]:
```
#Criando uma tabela cruzada
job_housing_counts = pd.crosstab(X['job'],X['housing'])
job_housing_counts_normalized = job_housing_counts.div(job_housing_counts.sum(axis=1), axis=0)

# Plotar gráfico de proporção
sns.set_theme(style="whitegrid", palette='viridis')
job_housing_counts_normalized.plot(kind='bar', stacked=True)
plt.xlabel('Trabalho')
plt.ylabel('Proporção')
plt.title('Proporção de Crédito Habitacioanl com Cada Categoria de Trabalho')
plt.legend(title='Crédito Habitação', bbox_to_anchor=(1, 1), loc='upper left', fontsize='small')
plt.xticks(rotation=70)
plt.show()
```

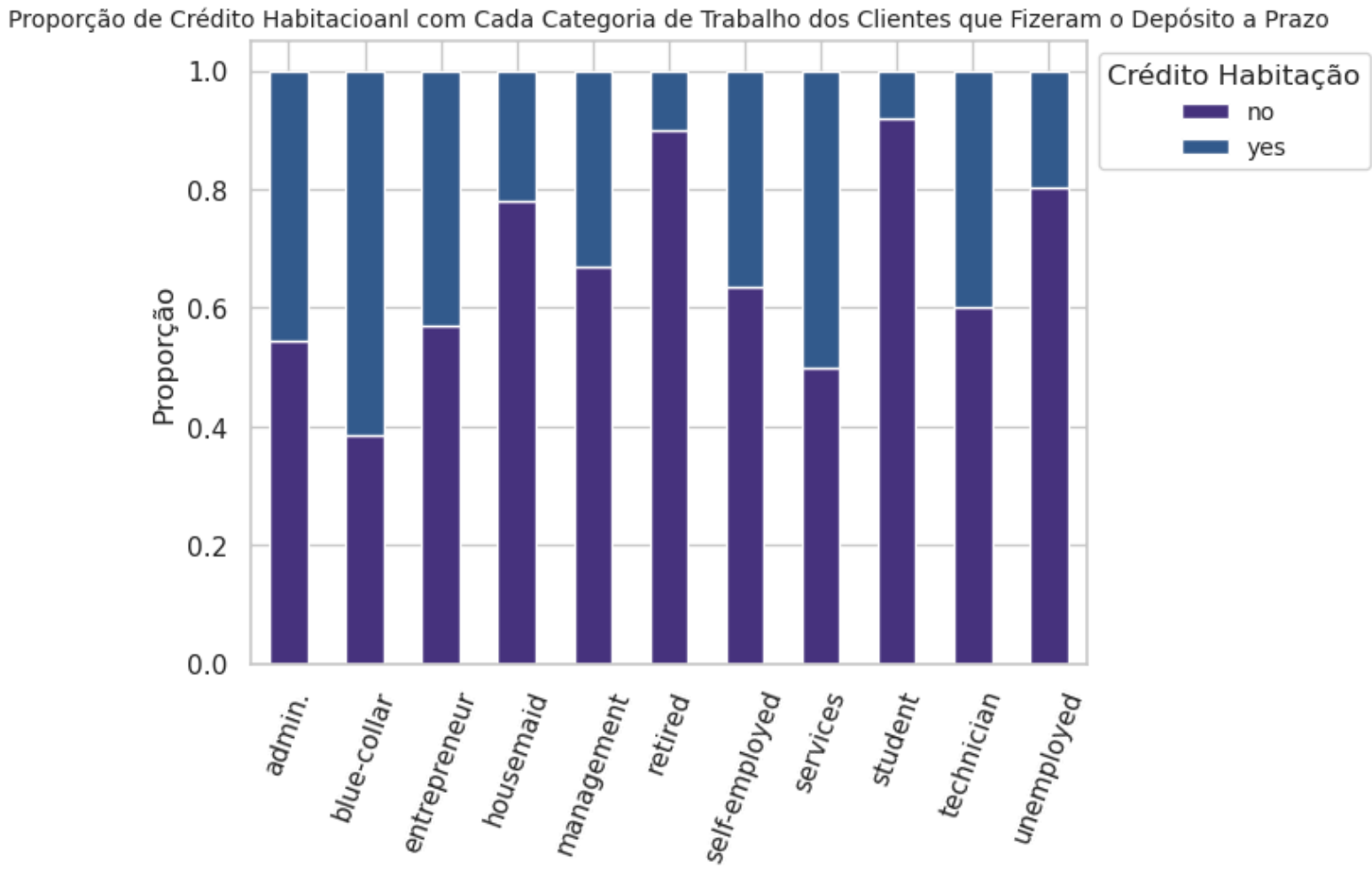


No gráfico acima, podemos observar que as categorias de trabalhadores domésticos, aposentados e estudantes são aquelas com a maior proporção de pessoas sem crédito habitacional. Isso ocorre porque muitas dessas pessoas não possuem garantias suficientes para a concessão de crédito habitacional.

Também podemos ver a seguir como fica a proporção de crédito habitacional por categoria de trabalho para os clientes que fizeram o deposito a prazo.

In [ ]:
```
#Criando uma tabela cruzada dos clientes que aceitaram o produto
job_housing_y_counts = pd.crosstab(X[y.values == 'yes']['job'],X[y.values == 'yes']['housing'])
job_housing_y_counts_normalized = job_housing_y_counts.div(job_housing_y_counts.sum(axis=1), axis=0)

# Plotar gráfico de proporção dos clientes que aceitaram o produto
sns.set_theme(style="whitegrid", palette='viridis')
job_housing_y_counts_normalized.plot(kind='bar', stacked=True)
plt.xlabel('Trabalho')
plt.ylabel('Proporção')
plt.title('Proporção de Crédito Habitacioanl com Cada Categoria de Trabalho dos Clientes que Fizeram o Depósito a Prazo',
plt.legend(title='Crédito Habitação', bbox_to_anchor=(1, 1), loc='upper left', fontsize='small')
plt.xticks(rotation=70)
plt.show()
```



Como mostrado no gráfico acima, todas as categorias de trabalho apresentaram um aumento na proporção de pessoas sem crédito habitacional. As classes de trabalhadores manuais, gerentes, técnicos e empresários têm uma proporção maior de clientes com crédito habitacional, indicando uma maior estabilidade financeira. Em contraste, as categorias de trabalhadores domésticos, aposentados, estudantes e desempregados apresentam uma proporção menor de clientes com crédito habitacional, refletindo fatores como baixa renda. É interessante observar que, mesmo entre os clientes que aceitaram o depósito a prazo, as proporções variam significativamente entre as diferentes categorias de trabalho.

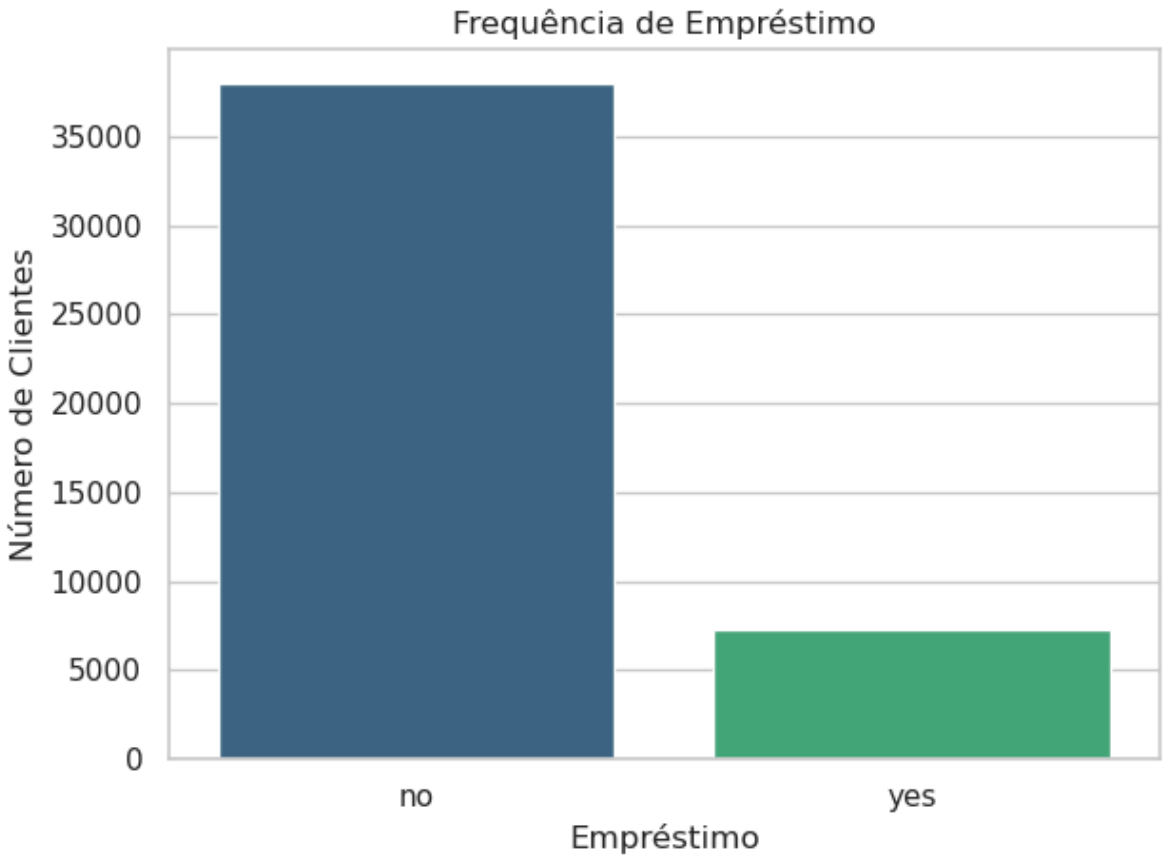
Empréstimo

A variável *loan* é uma variável categórica que assume os valores *yes* ou *no*, indicando se o cliente possui um empréstimo pessoal.

In [ ]:
```
# Criando uma tabela de frequência
loan_counts = X['loan'].value_counts().sort_index()
loan_percentages= (loan_counts * 100) / sum(loan_counts)
print(loan_percentages)
```

```
loan
no    83.9774
yes   16.0226
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia
sns.barplot(x=loan_counts.index, y=loan_counts.values, palette='viridis')
plt.title('Frequência de Empréstimo')
plt.xlabel('Empréstimo')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```



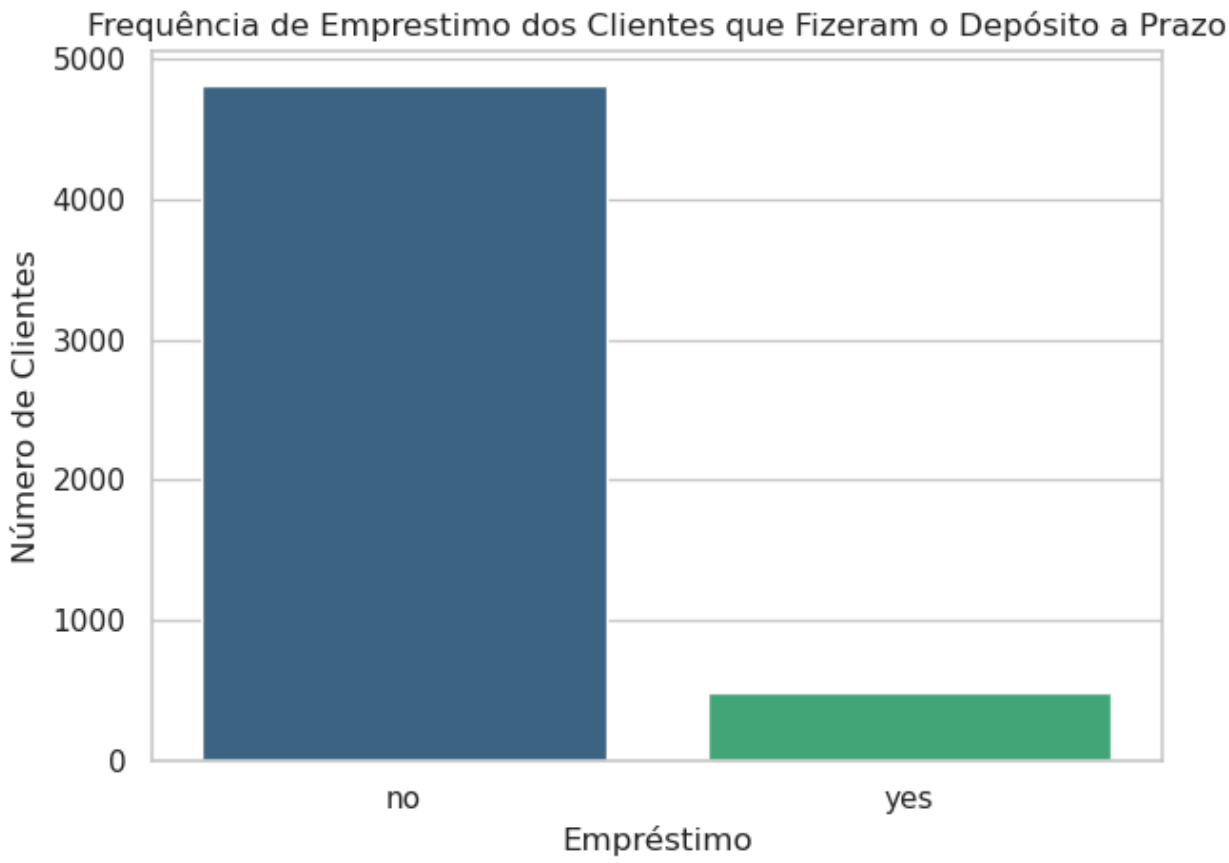
Na tabela de porcentagens acima, observamos que aproximadamente 84% dos clientes não possuem empréstimo pessoal, enquanto 16% possuem um empréstimo. Esse desequilíbrio sugere que a maioria dos clientes não está endividada com empréstimos pessoais.

Esse padrão pode ter implicações importantes para a análise de crédito e risco. Por exemplo, clientes sem empréstimos podem ter uma situação financeira mais estável ou podem estar menos propensos a assumir dívidas. Por outro lado, aqueles com empréstimos pessoais podem ter uma gestão financeira mais agressiva ou estar mais expostos a riscos financeiros.

In [ ]:
```
# Criando uma tabela de frequência dos clientes que aceitaram o produto
loan_y_counts = X[y.values == 'yes']['loan'].value_counts().sort_index()
loan_y_percentages= (loan_y_counts * 100) / sum(loan_y_counts)
print(loan_y_percentages)
```

```
loan
no    90.8489
yes    9.1511
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia dos clientes que aceitaram o produto
sns.barplot(x=loan_y_counts.index, y=loan_y_counts.values, palette='viridis')
plt.title('Frequência de Emprestimo dos Clientes que Fizeram o Depósito a Prazo')
plt.xlabel('Empréstimo')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```



É interessante observar que, ao analisar os clientes que fizeram o depósito a prazo, aproximadamente 90,85% deles não possuem um empréstimo pessoal. Isso pode ocorrer porque pessoas com empréstimos muitas vezes não optam por fazer esse tipo de investimento. É possível que essas pessoas tenham contraído empréstimos para atender necessidades imediatas ou para investir em oportunidades que oferecem uma rentabilidade mais alta.

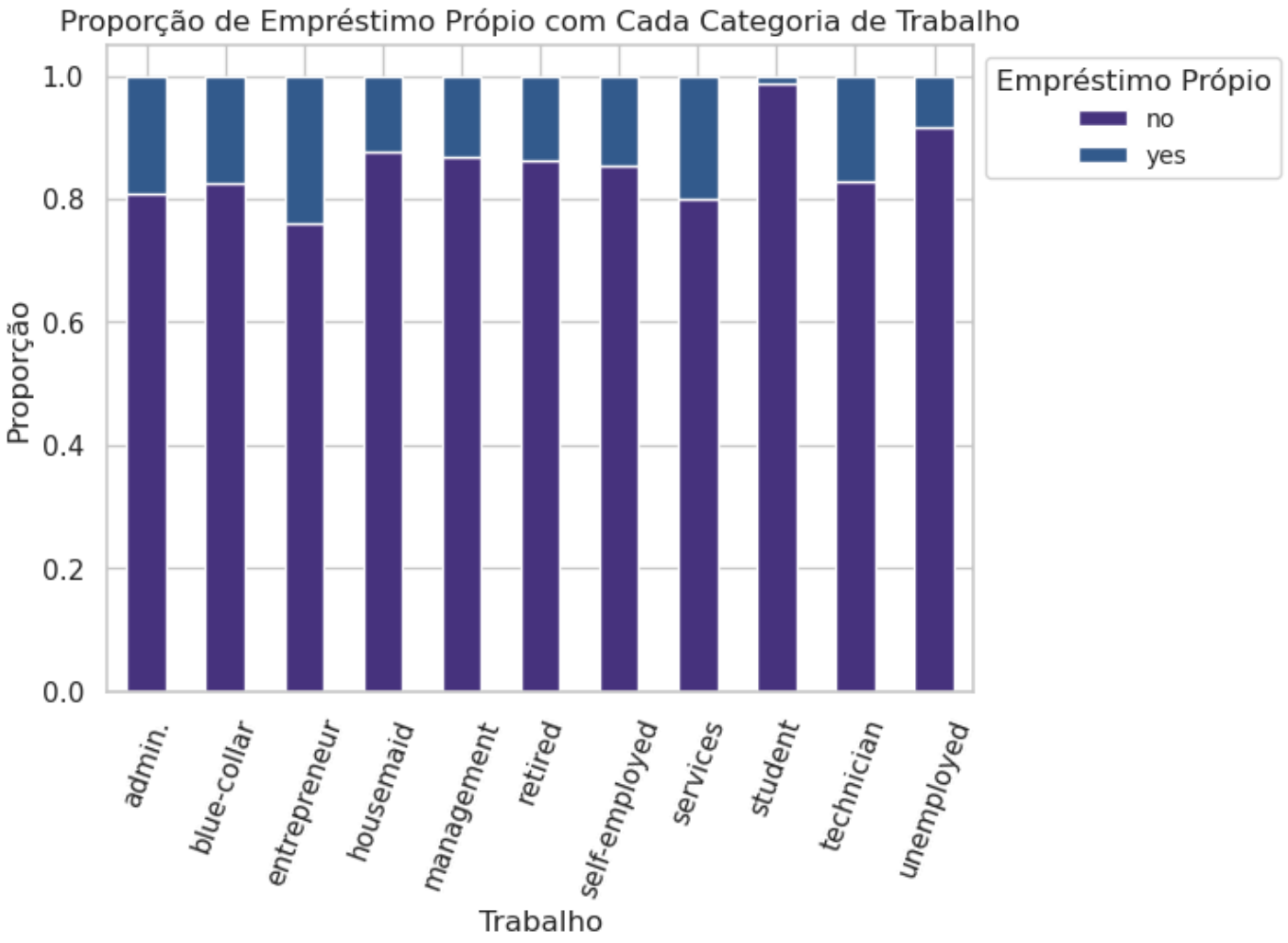
Além disso, é interessante comparar essa variável com outras características dos clientes

Empréstimo e Estado Civil

A seguir vamos verificar a propoção de clientes que tem empréstimo própio por cada categoria de trabalho.

In [ ]:
```
#Criando uma tabela cruzada
job_loan_counts = pd.crosstab(X['job'],X['loan'])
job_loan_counts_normalized = job_loan_counts.div(job_loan_counts.sum(axis=1), axis=0)

# Plotar gráfico de proporção
sns.set_theme(style="whitegrid", palette='viridis')
job_loan_counts_normalized.plot(kind='bar', stacked=True)
plt.xlabel('Trabalho')
plt.ylabel('Proporção')
plt.title('Proporção de Empréstimo Própio com Cada Categoria de Trabalho')
plt.legend(title='Empréstimo Própio', bbox_to_anchor=(1, 1), loc='upper left', fontsize='small')
plt.xticks(rotation=70)
plt.show()
```

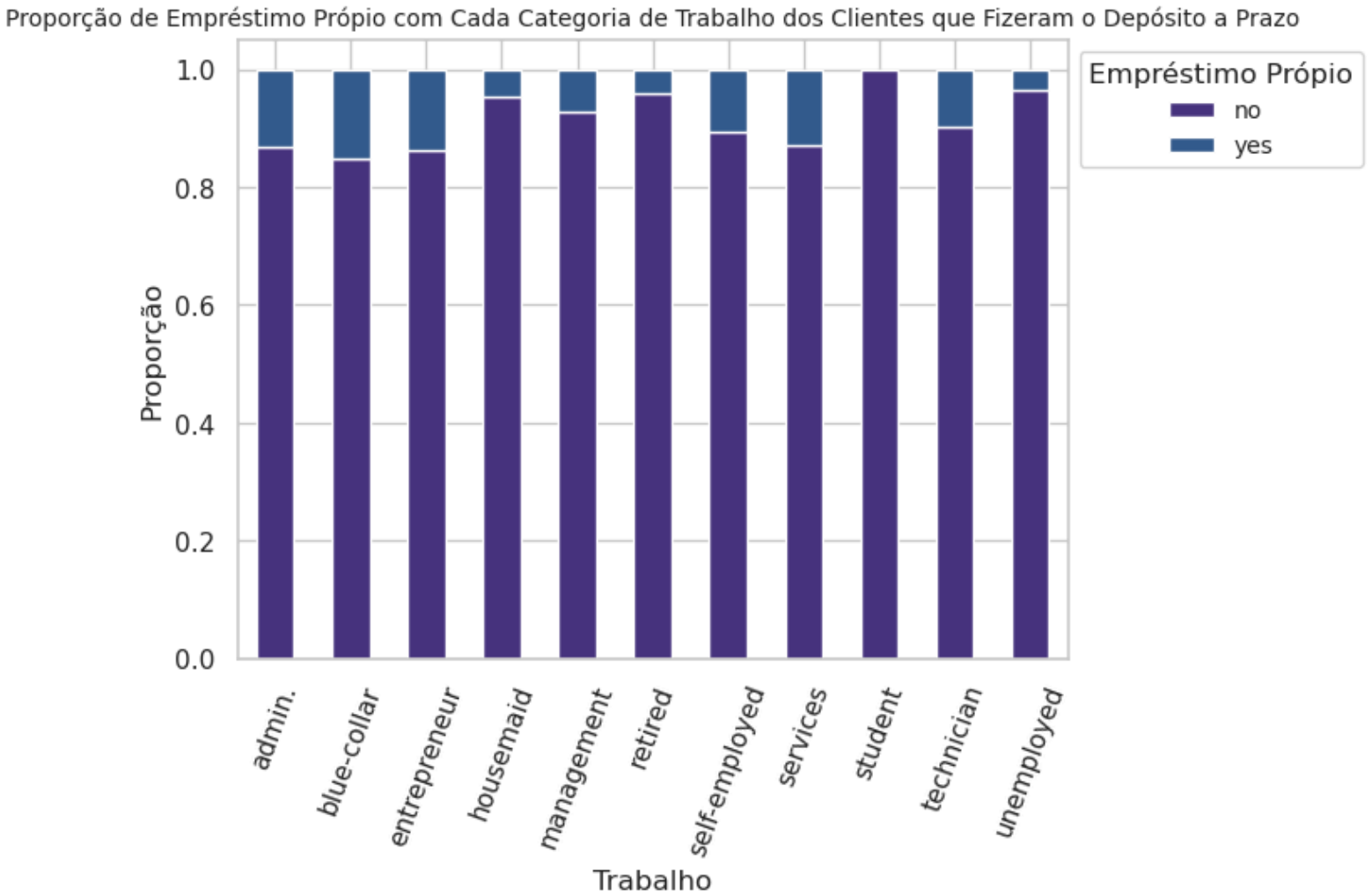


Ao analisar o gráfico acima, podemos observar que os estudantes apresentam a maior proporção de pessoas sem empréstimo pessoal. Além disso, a segunda categoria com menor proporção de empréstimos pessoais são os desempregados. No entanto, como vimos na sessão de inadimplência, os desempregados são a segunda classe com maior taxa de inadimplência. Isso sugere que o modelo de avaliação de risco dos bancos pode estar restringindo a concessão de empréstimos a essas pessoas devido ao risco elevado de inadimplência.

De acordo com as observações, a categoria de empresários tem cerca de 25% de pessoas com empréstimo pessoal. Isso indica que uma parte significativa dos empresários optou por tomar crédito devido à queda da taxa de juros na Europa entre 2008 e 2010, que é justamente o período de coleta do nosso banco de dados utilizado.

In [ ]:
```
#Criando uma tabela cruzada dos clientes que aceitaram o produto
job_loan_counts = pd.crosstab(X[y.values == 'yes']['job'],X[y.values == 'yes']['loan'])
job_loan_counts_normalized = job_loan_counts.div(job_loan_counts.sum(axis=1), axis=0)

# Plotar gráfico de proporção dos clientes que aceitaram o produto
sns.set_theme(style="whitegrid", palette='viridis')
job_loan_counts_normalized.plot(kind='bar', stacked=True)
plt.xlabel('Trabalho')
plt.ylabel('Proporção')
plt.title('Proporção de Empréstimo Própio com Cada Categoria de Trabalho dos Clientes que Fizeram o Depósito a Prazo', fon
plt.legend(title='Empréstimo Próprio', bbox_to_anchor=(1, 1), loc='upper left', fontsize='small')
plt.xticks(rotation=70)
plt.show()
```



Como era esperado, houve uma queda na quantidade de clientes que possuem empréstimo próprio e decidiram investir com o banco. Essa tendência já havia sido observada na seção de crédito habitacional. Podemos concluir, portanto, que os clientes que optam por fazer investimentos com o banco tendem a não ter empréstimo próprio ou crédito habitacional, o que os torna menos propensos à inadimplência.

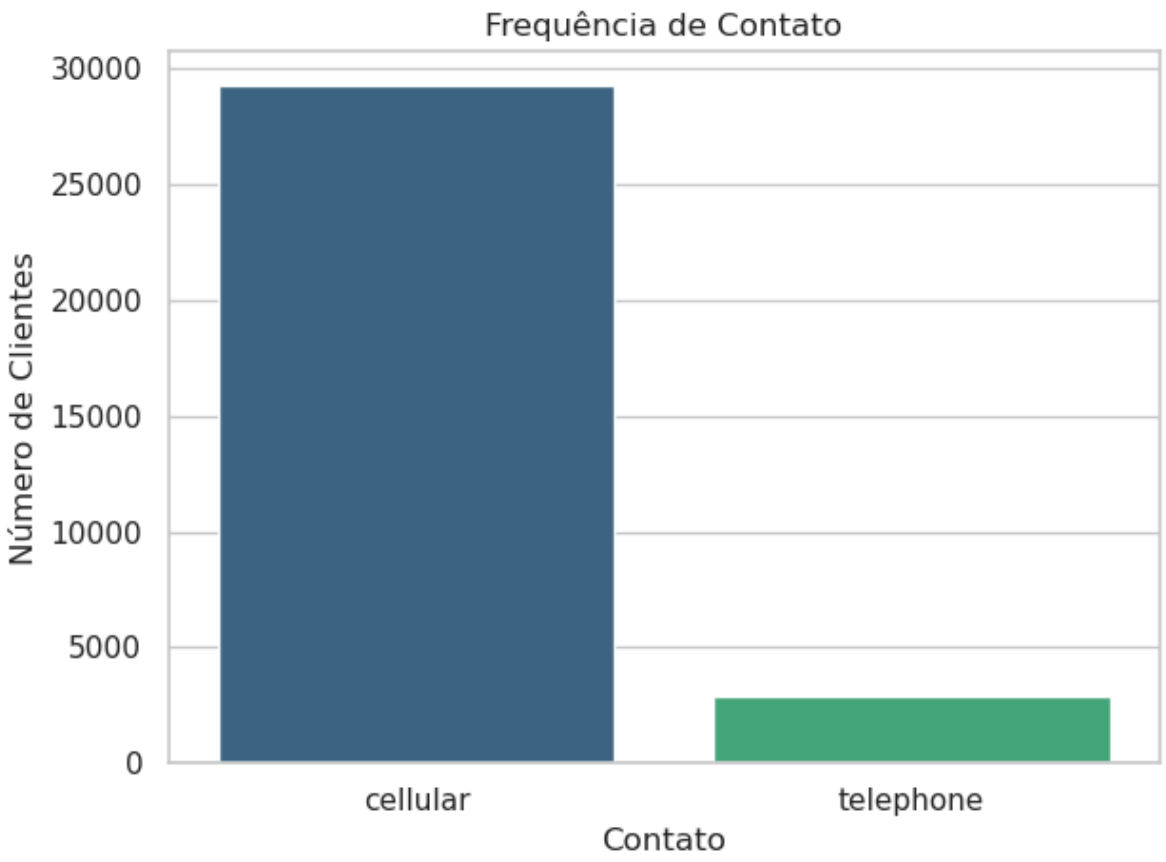
Contato

A variável *contact* é categórica e assume dois valores: *cellular* ou *telephone*. É importante destacar que essa variável possui 13.020 valores ausentes.

In [ ]:
```
# Criando uma tabela de frequência
contact_counts = X['contact'].value_counts().sort_index()
contact_percentages= (contact_counts * 100) / sum(contact_counts)
print(contact_percentages)
```

```
contact
cellular    90.9726
telephone    9.0274
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia
sns.barplot(x=contact_counts.index, y=contact_counts.values, palette='viridis')
plt.title('Frequência de Contato')
plt.xlabel('Contato')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```

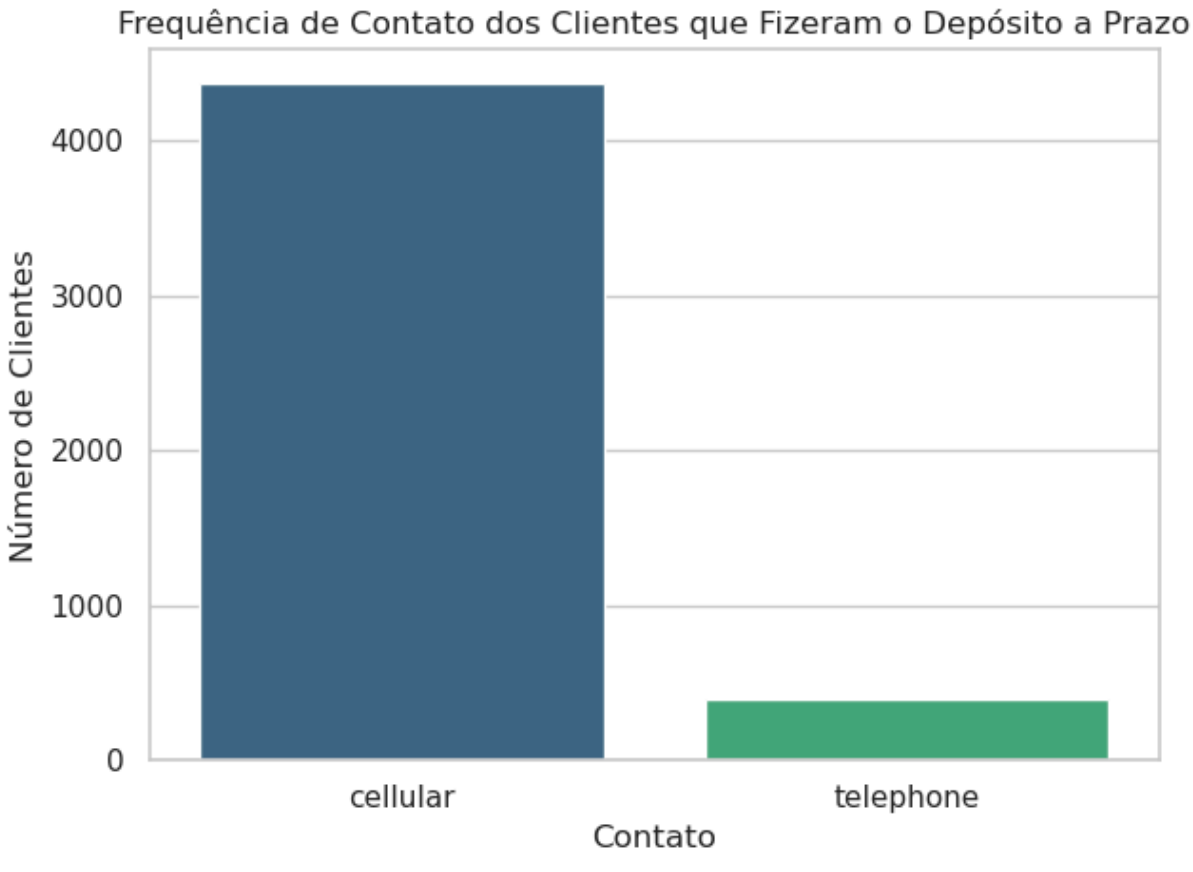


No século 21, as pessoas têm usado mais o celular do que o telefone fixo, devido à facilidade de portabilidade e à melhoria na utilidade dos aparelhos. Isso também é refletido na tabela e no gráfico apresentados acima, que mostram que cerca de 90,98% dos clientes têm contato por celular. Também é importante verificar se essa tendência aumenta entre as pessoas que aceitam fazer o investimento com o banco.

In [ ]:
```
# Criando uma tabela de frequência dos clientes que aceitaram o produto
contact_y_counts = X[y.values == 'yes']['contact'].value_counts().sort_index()
contact_y_percentages= (contact_y_counts * 100) / sum(contact_y_counts)
print(contact_y_percentages)
```

```
contact
cellular    91.8050
telephone    8.1950
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia dos clientes que aceitaram o produto
sns.barplot(x=contact_y_counts.index, y=contact_y_counts.values, palette='viridis')
plt.title('Frequência de Contato dos Clientes que Fizeram o Depósito a Prazo')
plt.xlabel('Contato')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```

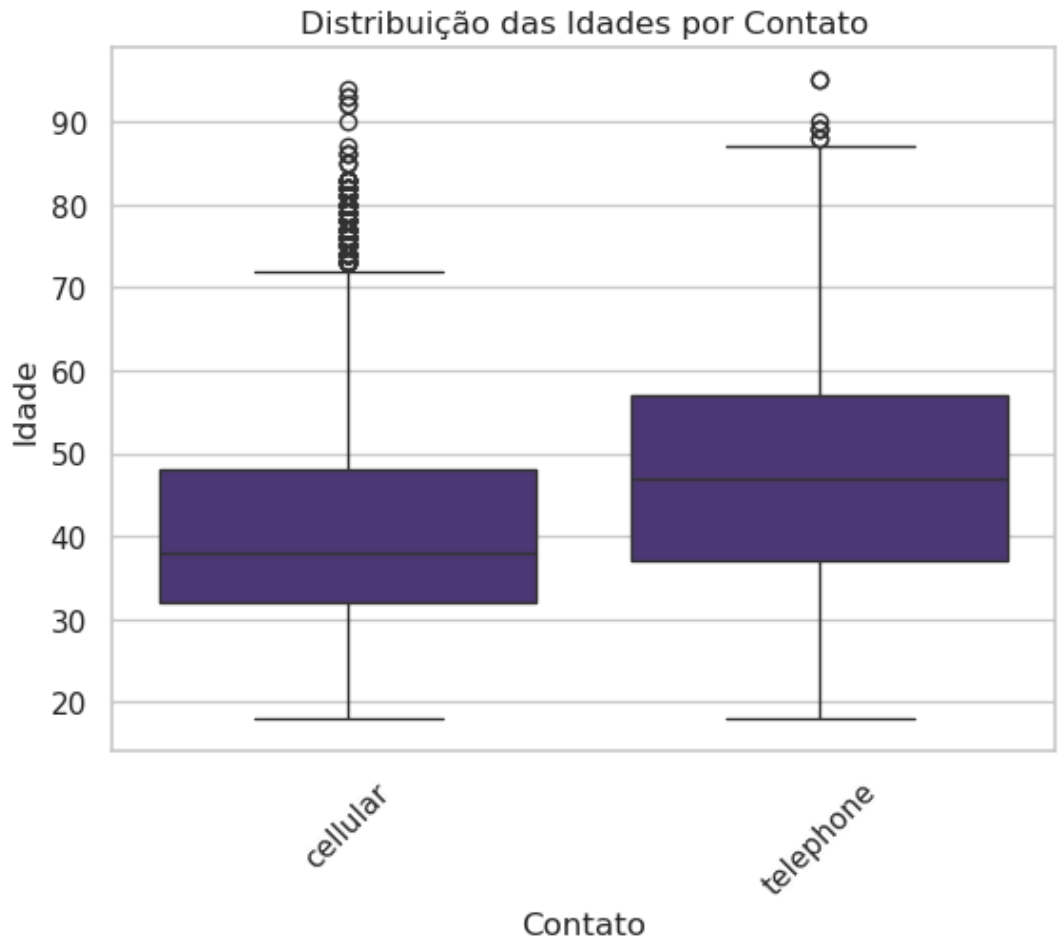


Na tabela e no gráfico acima, podemos ver que a tendência mencionada anteriormente aumenta um pouco, com 91,80% dos clientes que aceitam fazer o depósito a prazo preferindo o contato por celular.

Contato e Idade

A comparação entre os métodos de contato e a idade na nossa base de dados é importante para identificar qual tipo de pessoa geralmente utiliza celular ou telefone para comunicação.

In [ ]:
```
# Gráfico de contato por idade
sns.boxplot(x='contact', y='age', data=X)
plt.xticks(rotation=45)  # Rotaciona os rótulos do eixo X para melhor visualização
plt.title('Distribuição das Idades por Contato')
plt.xlabel('Contato')
plt.ylabel('Idade')
plt.show()
```

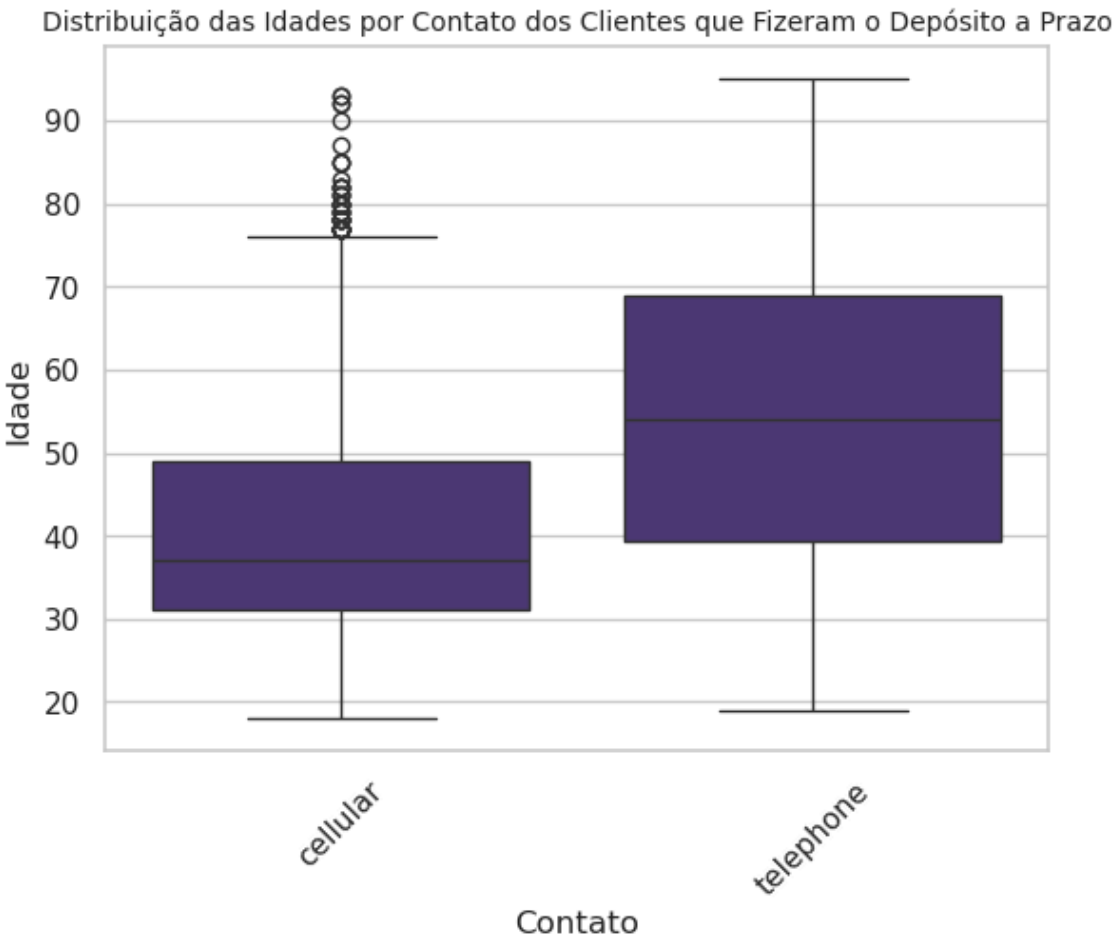


No boxplot acima, podemos observar a distribuição das idades por tipo de contato, seja por celular ou telefone fixo. É evidente que as pessoas que utilizam celular tendem a ser mais jovens em comparação com aquelas que utilizam telefone fixo. A mediana de idade para os usuários de celular é aproximadamente 38 anos, enquanto para os usuários de telefone fixo, a mediana é cerca de 58 anos.

Essa diferença significativa nas idades sugere que o celular é predominantemente utilizado por uma população mais jovem, possivelmente devido à sua maior familiaridade com tecnologias móveis e à necessidade de mobilidade constante. Por outro lado, o telefone fixo é mais comum entre uma população mais velha, que pode estar mais acostumada a métodos tradicionais de comunicação.

Além disso, os outliers presentes no gráfico indicam que, embora a maioria dos usuários de celular esteja na faixa etária mais jovem, ainda existem alguns usuários mais velhos que utilizam esse meio de contato. O mesmo se aplica ao telefone fixo, onde alguns usuários mais jovens preferem essa forma de comunicação, talvez por motivos de estabilidade ou preferência pessoal.

In [ ]:
```
# Gráfico de contato por idade dos clientes que fizeram o depósito a prazo
sns.boxplot(x='contact', y='age', data=X[y.values == 'yes'])
plt.xticks(rotation=45)  # Rotaciona os rótulos do eixo X para melhor visualização
plt.title('Distribuição das Idades por Contato dos Clientes que Fizeram o Depósito a Prazo', fontsize=10)
plt.xlabel('Contato')
plt.ylabel('Idade')
plt.show()
```



uando analisamos as observações dos clientes que aceitaram fazer o depósito a prazo, percebemos que os comentários anteriores ainda se mantêm válidos. No entanto, há uma pequena alteração na mediana de idade das pessoas que utilizam o telefone, que agora é de aproximadamente 58 anos.

Isso sugere que, mesmo entre os clientes que optaram pelo depósito a prazo, a preferência pelo tipo de contato segue o padrão observado na base de dados geral. Os usuários de celular continuam a ser mais jovens, enquanto os usuários de telefone fixo são, em média, mais velhos.

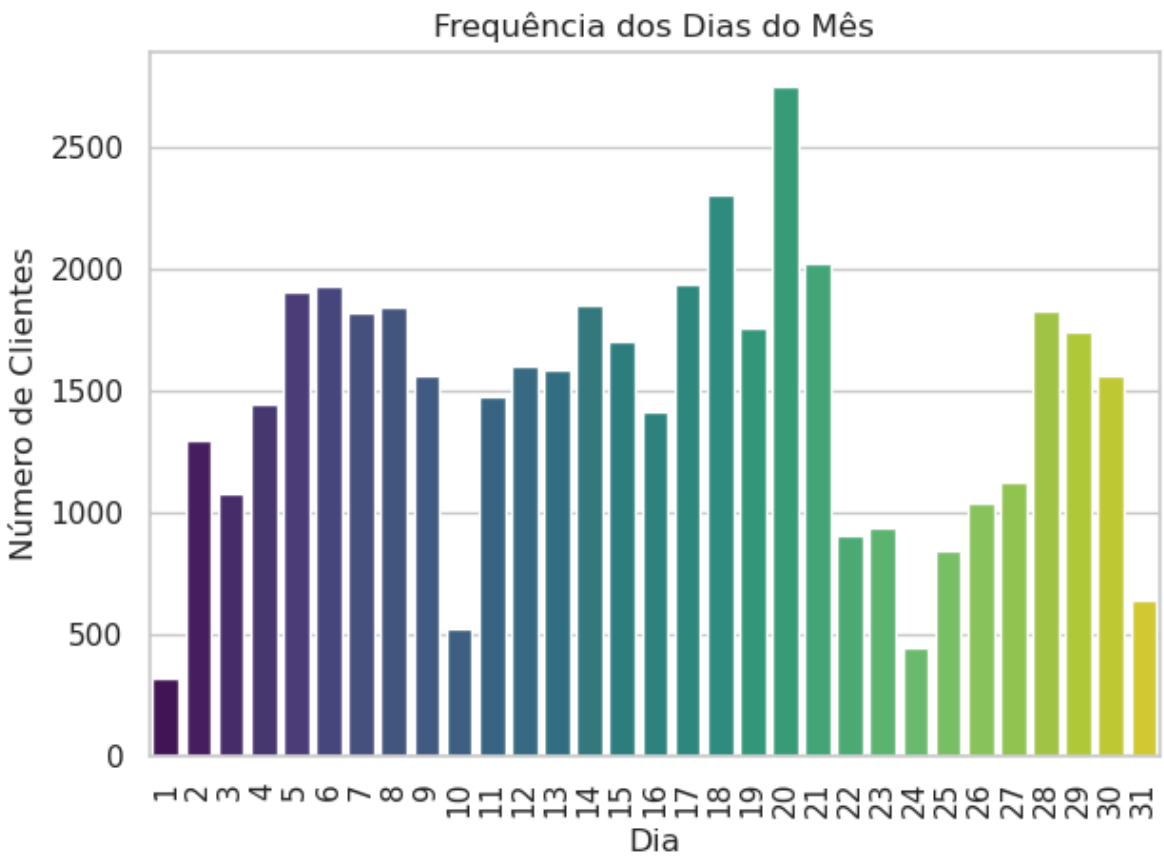
Dia da Semana

A variável *day_of_week* é uma variável numérica que representa o dia do mês em que o contato foi realizado. Os valores dessa variável variam de 1 a 31, correspondendo ao primeiro e ao último dia do mês, respectivamente.

In [ ]:
```
# Criando uma tabela de frequência
dow_counts = X['day_of_week'].value_counts().sort_index()
dow_percentages= (dow_counts * 100) / sum(dow_counts)
print(dow_percentages)
```

```
day_of_week
1    0.7122
2    2.8599
3    2.3866
4    3.1961
5    4.2246
6    4.2733
7    4.0189
8    4.0742
9    3.4527
10   1.1590
11   3.2713
12   3.5456
13   3.5058
14   4.0875
15   3.7668
16   3.1298
17   4.2888
18   5.1050
19   3.8862
20   6.0870
21   4.4812
22   2.0017
23   2.0769
24   0.9887
25   1.8580
26   2.2893
27   2.4795
28   4.0477
29   3.8597
30   3.4638
31   1.4222
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia
sns.barplot(x=dow_counts.index, y=dow_counts.values, palette='viridis')
plt.title('Frequência dos Dias do Mês')
plt.xlabel('Dia')
plt.ylabel('Número de Clientes')
plt.xticks(rotation=90)
plt.tight_layout()
```

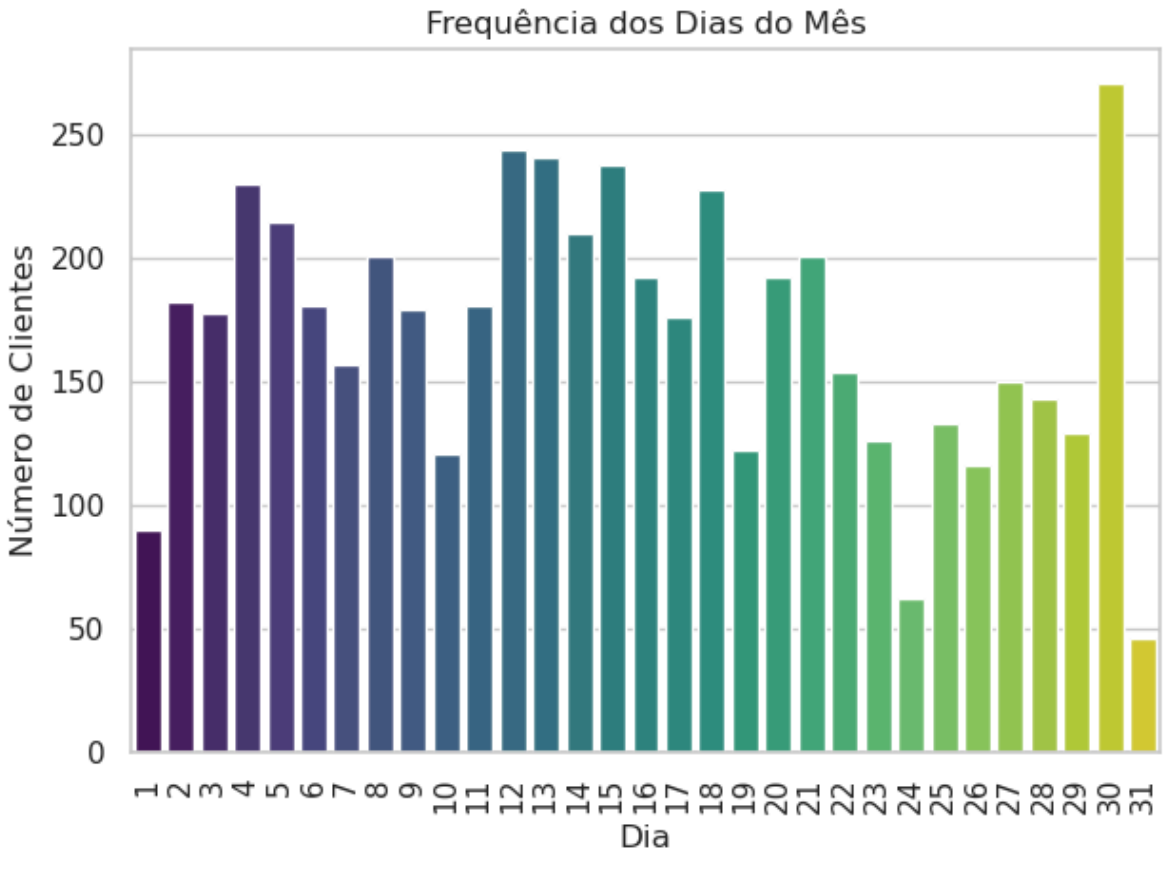


No gráfico acima, podemos observar que a maioria dos clientes foi contatada pelo banco no dia 21, representando 6,087% do total de observações. Por outro lado, o dia 1 tem o menor número de observações, com aproximadamente 0,7122% do total. A análise do gráfico e da tabela revela que a maior concentração de contatos entre o banco e os clientes ocorreu entre os dias 11 e 21, cobrindo esse intervalo de 11 dias. Portanto, para melhorar a eficácia das ações de telemarketing, é essencial identificar o dia da semana que apresenta a maior taxa de contato, especialmente entre os clientes que estão mais propensos a aceitar o depósito a prazo.

In [ ]:
```
# Criando uma tabela de frequência dos clientes que fizeram o depósito a prazo
dow_y_counts = X[y.values == 'yes']['day_of_week'].value_counts().sort_index()
dow_y_percentages= (dow_y_counts * 100) / sum(dow_y_counts)
print(dow_y_percentages)
```

```
day_of_week
1    1.7016
2    3.4411
3    3.3655
4    4.3486
5    4.0650
6    3.4222
7    2.9684
8    3.8003
9    3.3844
10   2.2878
11   3.4222
12   4.6133
13   4.5566
14   3.9705
15   4.4999
16   3.6302
17   3.3277
18   4.3108
19   2.3067
20   3.6302
21   3.8003
22   2.9117
23   2.3823
24   1.1722
25   2.5147
26   2.1932
27   2.8361
28   2.7037
29   2.4390
30   5.1238
31   0.8697
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia dos clientes que fizeram o depósito a prazo
sns.barplot(x=dow_y_counts.index, y=dow_y_counts.values, palette='viridis')
plt.title('Frequência dos Dias do Mês')
plt.xlabel('Dia')
plt.ylabel('Número de Clientes')
plt.xticks(rotation=90)
plt.tight_layout()
```



Ao analisar a frequência dos dias em que os clientes aceitaram fazer o depósito, observamos que o dia 30 apresenta a maior quantidade de observações, representando aproximadamente 5,12% do total de clientes. Por outro lado, os dias com menor número de observações são o dia 31 e o dia 24. No entanto, ainda há uma concentração significativa de contatos no meio do mês.

Mês

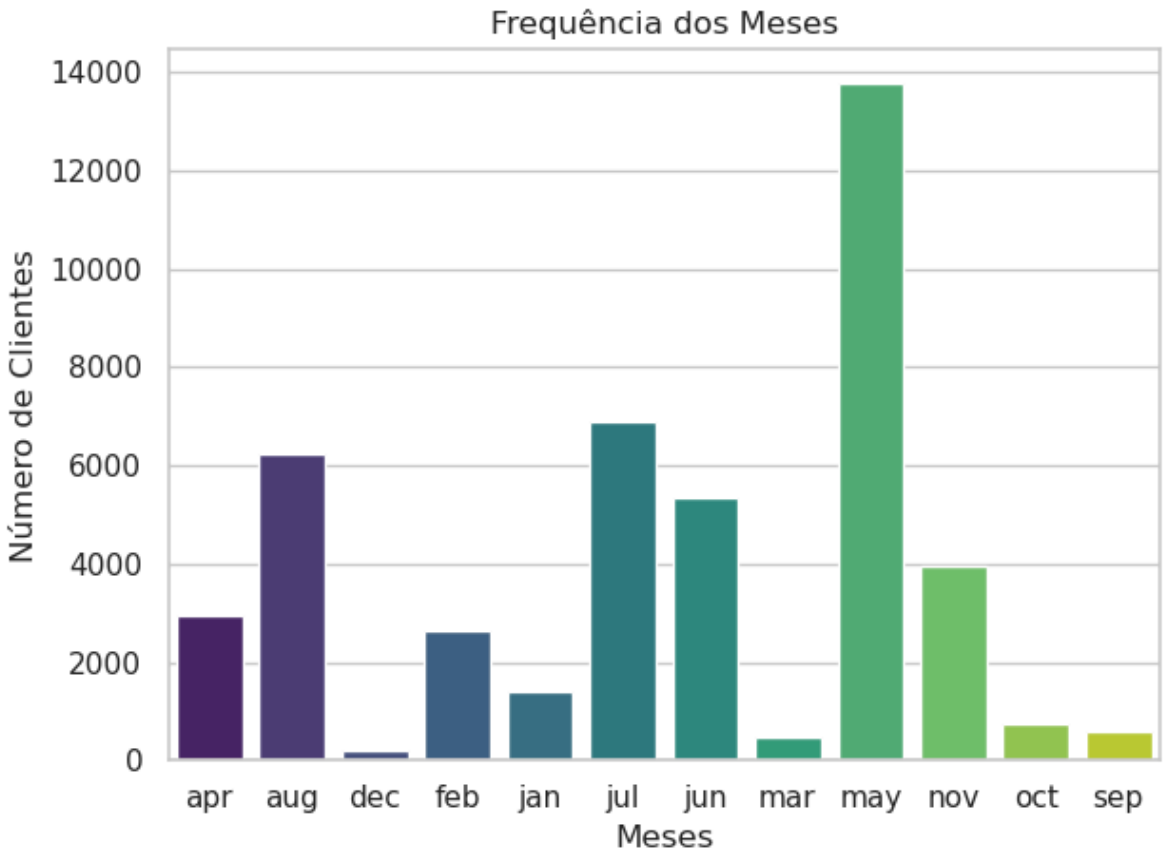
A variável *month* é uma variável categórica que representa os nomes dos meses em que o contato foi realizado. As categorias dessa variável são:

```
Jan: Janeiro
Feb: Fevereiro
Mar: Março
Apr: Abril
May: Maio
Jun: Junho
Jul: Julho
Aug: Agosto
Sep: Setembro
Oct: Outubro
Nov: Novembro
Dec: Dezembro
```

In [ ]:
```python
# Criando uma tabela de frequência
month_counts = X['month'].value_counts().sort_index()
month_percentages= (month_counts * 100) / sum(month_counts)
print(month_percentages)
```

```
month
apr    6.4851
aug   13.8174
dec    0.4733
feb    5.8592
jan    3.1032
jul   15.2507
jun   11.8135
mar    1.0551
may   30.4483
nov    8.7810
oct    1.6323
sep    1.2807
Name: count, dtype: float64
```

In [ ]:
```python
# Criando um gráfico de frêquencia dos clientes que aceitaram o produto
sns.barplot(x=month_counts.index, y=month_counts.values, palette='viridis')
plt.title('Frequência dos Meses')
plt.xlabel('Meses')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```

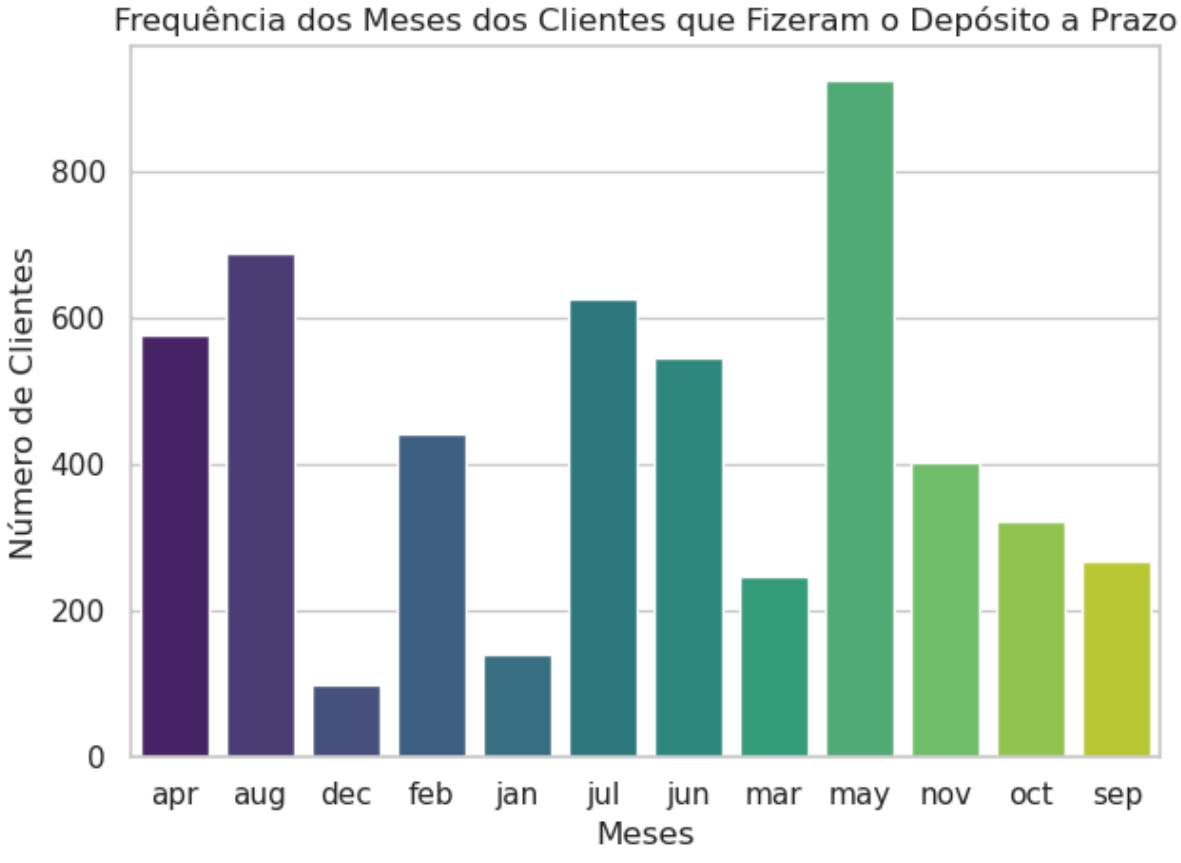


No gráfico acima, podemos observar que o mês de maio apresenta quase 14.000 observações, representando 30,44% do total. Em seguida, temos o mês de julho com 15,25%, agosto com 13,82% e junho com 11,81%. É importante notar que esses meses correspondem ao meio do ano. Portanto, podemos inferir que os clientes têm uma maior facilidade de serem contatados nessa época, que é um período com menos demanda e menos eventos festivos.

In [ ]:
```
# Criando uma tabela de frequência dos clientes que aceitaram o produto
month_y_counts = X[y.values == 'yes']['month'].value_counts().sort_index()
month_y_percentages= (month_y_counts * 100) / sum(month_y_counts)
print(month_y_percentages)
```

```
month
apr   10.9094
aug   13.0081
dec    1.8907
feb    8.3381
jan    2.6848
jul   11.8548
jun   10.3233
mar    4.6890
may   17.4891
nov    7.6196
oct    6.1070
sep    5.0860
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia dos clientes que aceitaram o produto
sns.barplot(x=month_y_counts.index, y=month_y_counts.values, palette='viridis')
plt.title('Frequência dos Meses dos Clientes que Fizeram o Depósito a Prazo')
plt.xlabel('Meses')
plt.ylabel('Número de Clientes')
#plt.xticks(rotation=90)
plt.tight_layout()
```



Acima, podemos verificar que essa tendência se mantém consistente ao analisar os clientes que tiveram conversão com o marketing direto. Assim, o banco pode seguir o padrão e realizar campanhas de telemarketing mais intensas durante o meio do ano.

Duração

A variável *duration* é uma variável inteira que representa a duração do último contato, em segundos. Essa informação só é disponível ao final da conversa. Portanto, é importante lembrar que a variável *duration* deve ser utilizada apenas para fins de benchmarking, para avaliar a eficácia geral das campanhas de telemarketing. Para o modelo preditivo proposto neste projeto, essa variável não será considerada, pois não está disponível no momento da previsão.

In [ ]:
```
# Resumo estatístico
X["duration"].describe()
```

Out[ ]:
```
count   45211.0000
mean      258.1631
std       257.5278
min         0.0000
25%       103.0000
50%       180.0000
75%       319.0000
max      4918.0000
Name: duration, dtype: float64
```

In [ ]:
```
# Visualizando a moda
X["duration"].mode()
```

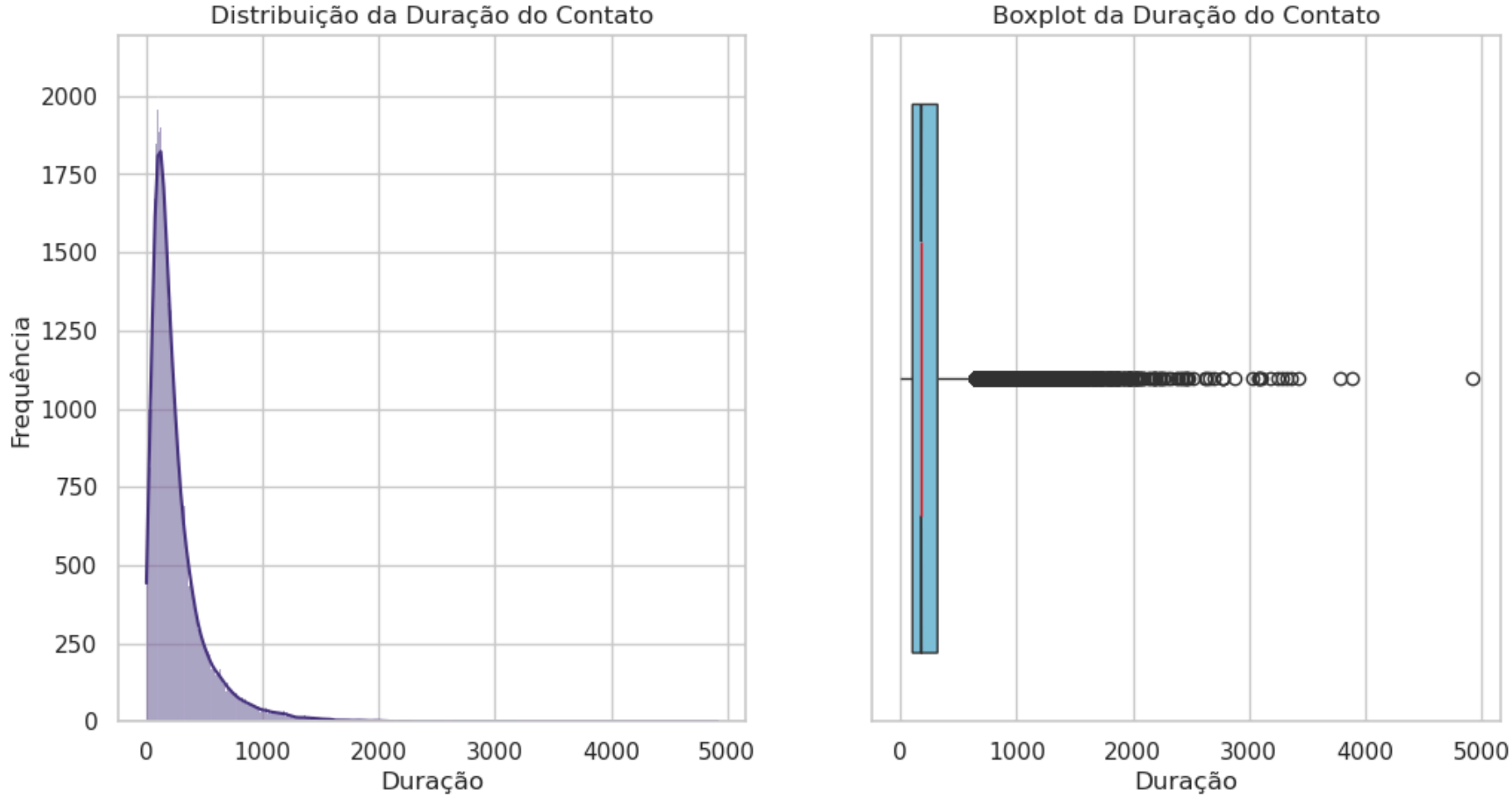
Out[ ]:
```
0    124
Name: duration, dtype: int64
```

Ao analisar o resumo da variável *duration*, que representa a duração da última conversa entre o cliente e o banco, observamos que a média é de aproximadamente 258,16 segundos, com um desvio padrão de 257,53 segundos. Isso indica que, em média, as durações estão concentradas no intervalo entre 0,63 e 515,69 segundos.

Além disso, o valor mínimo é 0 segundos, o que sugere que esses clientes provavelmente não realizaram um depósito a prazo, pois não houve contato com o banco. O valor máximo registrado é de 4.918 segundos, o que pode ser um erro de digitação. Para confirmar isso, é importante verificar se há outros outliers próximos a esse valor extremo.

Com uma mediana de 180 segundos e uma média de 258,16 segundos, podemos inferir que a variável *duration* apresenta uma distribuição assimétrica positiva dos dados, com uma cauda mais longa à direita.

In [ ]:
```
# Criação do grafico histograma e boxplot
plt.figure(figsize=(12,6))
plt.subplot(1, 2, 1)
sns.histplot(X["duration"], kde=True)
plt.title("Distribuição da Duração do Contato")
plt.xlabel("Duração")
plt.ylabel("Frequência")
plt.subplot(1, 2, 2)
sns.boxplot(X["duration"], orient='h', notch=True, showcaps=False,
            boxprops={"facecolor": (0, .5, .7, .5)},
            medianprops={"color": "r", "linewidth": 1})
plt.title("Boxplot da Duração do Contato")
plt.xlabel("Duração")
plt.show()
```



Como mencionado anteriormente, a variável duration apresenta uma assimetria positiva nos dados. Isso indica que muitos clientes passam apenas alguns segundos em contato com o banco, enquanto algumas observações têm uma duração de contato significativamente maior. Clientes que passam mais tempo na conversa podem estar buscando mais informações sobre o produto ofertado e, portanto, podem estar mais propensos a aceitar a oferta após uma conversa mais longa.

No entanto, é importante ressaltar que a variável duration só pode ser medida após o contato ser realizado. Portanto, para fins de previsão, ela não pode ser utilizada, pois não está disponível antes da realização da chamada.

Campanha

A variáevel *capmpaign* é uma variável inteira que representa o número de contatos realizados durante a campanha de telemarketing.

In [ ]:
```
# Criando uma tabela de frequência dos clientes que fizeram o depósito a prazo
campaign_counts = X['campaign'].value_counts().sort_index()
campaign_percentages= (campaign_counts * 100) / sum(campaign_counts)
print(campaign_percentages)
```

```
campaign
1    38.8047
2    27.6592
3    12.2116
4     7.7901
5     3.9017
6     2.8555
7     1.6257
8     1.1944
9     0.7233
10    0.5884
11    0.4446
12    0.3428
13    0.2942
14    0.2057
15    0.1858
16    0.1747
17    0.1526
18    0.1128
19    0.0973
20    0.0951
21    0.0774
22    0.0509
23    0.0487
24    0.0442
25    0.0487
26    0.0288
27    0.0221
28    0.0354
29    0.0354
30    0.0177
31    0.0265
32    0.0199
33    0.0133
34    0.0111
35    0.0088
36    0.0088
37    0.0044
38    0.0066
39    0.0022
41    0.0044
43    0.0066
44    0.0022
46    0.0022
50    0.0044
51    0.0022
55    0.0022
58    0.0022
63    0.0022
Name: count, dtype: float64
```

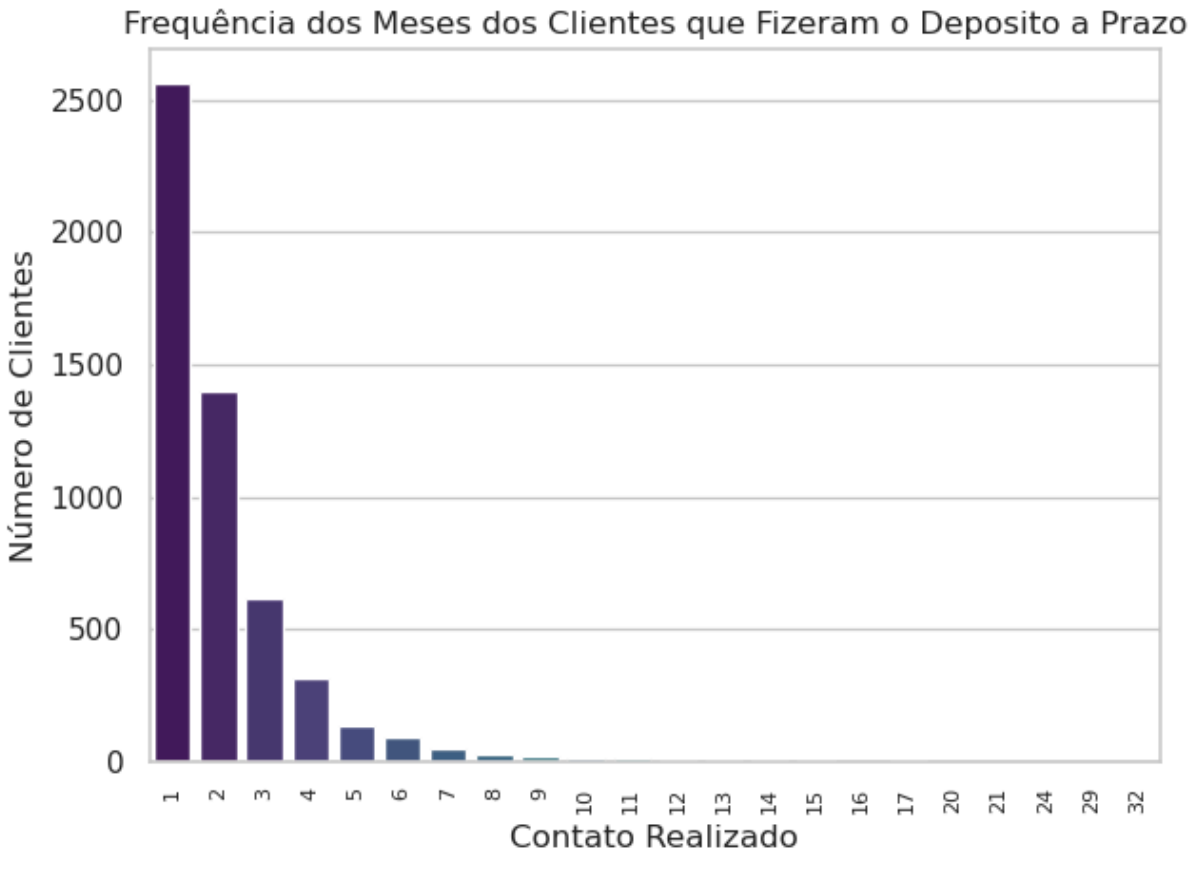
In [ ]:
```
# Criando um gráfico de frêquencia dos clientes que aceitaram o produto
sns.barplot(x=campaign_counts.index, y=campaign_counts.values, palette='viridis')
plt.title('Frequência de Contato Realizado ')
plt.xlabel('Contato Realizado')
plt.ylabel('Número de Clientes')
plt.xticks(rotation=90, fontsize=8)
plt.tight_layout()
```

Na tabela acima, podemos observar que clientes foram contatados entre 1 e 63 vezes pela mesma campanha de telemarketing. O número máximo de contatos é excessivo, mas a partir do oitavo contato, a proporção cai para menos de 1%. O número de clientes que recebeu apenas um contato foi de 38,80%, enquanto aqueles que receberam duas ligações totalizaram 27,66%, três ligações representaram 12,21% e quatro ligações foram 7,79%, somando aproximadamente 86,46% do total das observações. É fundamental identificar quais clientes realizaram mais investimentos com base na quantidade de ligações recebidas.

In [ ]:
```
# Criando uma tabela de frequência dos clientes que fizeram o depósito a prazo
campaign_y_counts = X[y.values == 'yes']['campaign'].value_counts().sort_index()
campaign_y_percentages= (campaign_y_counts * 100) / sum(campaign_y_counts)
print(campaign_y_percentages)
```

```
campaign
1    48.4213
2    26.4889
3    11.6846
4     5.9936
5     2.6281
6     1.7395
7     0.8886
8     0.6050
9     0.3971
10    0.2647
11    0.3025
12    0.0756
13    0.1134
14    0.0756
15    0.0756
16    0.0378
17    0.1134
20    0.0189
21    0.0189
24    0.0189
29    0.0189
32    0.0189
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia dos clientes que aceitaram o produto
sns.barplot(x=campaign_y_counts.index, y=campaign_y_counts.values, palette='viridis')
plt.title('Frequência dos Meses dos Clientes que Fizeram o Deposito a Prazo')
plt.xlabel('Contato Realizado')
plt.ylabel('Número de Clientes')
plt.xticks(rotation=90, fontsize=8)
plt.tight_layout()
```



Podemos observar que a quantidade de contatos realizados para clientes que aceitaram o produto foi significativamente menor, com o valor máximo agora limitado a 32 contatos. A distribuição é a seguinte: 48,42% dos clientes receberam apenas um contato, 26,49% receberam dois, 11,68% receberam três e 5,99% receberam quatro contatos. Portanto, é importante que o banco adote uma abordagem diferente no telemarketing para evitar perturbar os clientes e melhorar a taxa de retorno.

Pdays

A variável *pdays* é do tipo inteiro e pode assumir valores de -1 até +infinito. Ela representa o número de dias que se passaram desde o último contato com o cliente, sendo que o valor -1 indica que o cliente nunca foi contatado anteriormente.

In [ ]:
```
# Criando uma tabela de frequência
X['pdays'].describe()
```

Out[ ]:
```
count   45211.0000
mean       40.1978
std       100.1287
min        -1.0000
25%        -1.0000
50%        -1.0000
75%        -1.0000
max       871.0000
Name: pdays, dtype: float64
```

Ao analisar o resumo da variável *pdays*, podemos notar que pelo menos 75% dos dados são -1, indicando que 75% das observações correspondem a clientes que não foram contatados anteriormente. Além disso, podemos observar que o cliente que teve a maior demora desde o último contato esperou 871 dias para receber um retorno do banco.

Devido a esse desequilíbrio e forte assimetria positiva dos dados, na próxima etapa do projeto será proposta alguma transformação para melhorar a qualidade dos dados.

Anterior

A variável *previous* também é do tipo inteiro, assumindo apenas valores inteiros e positivos. Ela indica o número de contatos realizados antes da campanha de telemarketing.

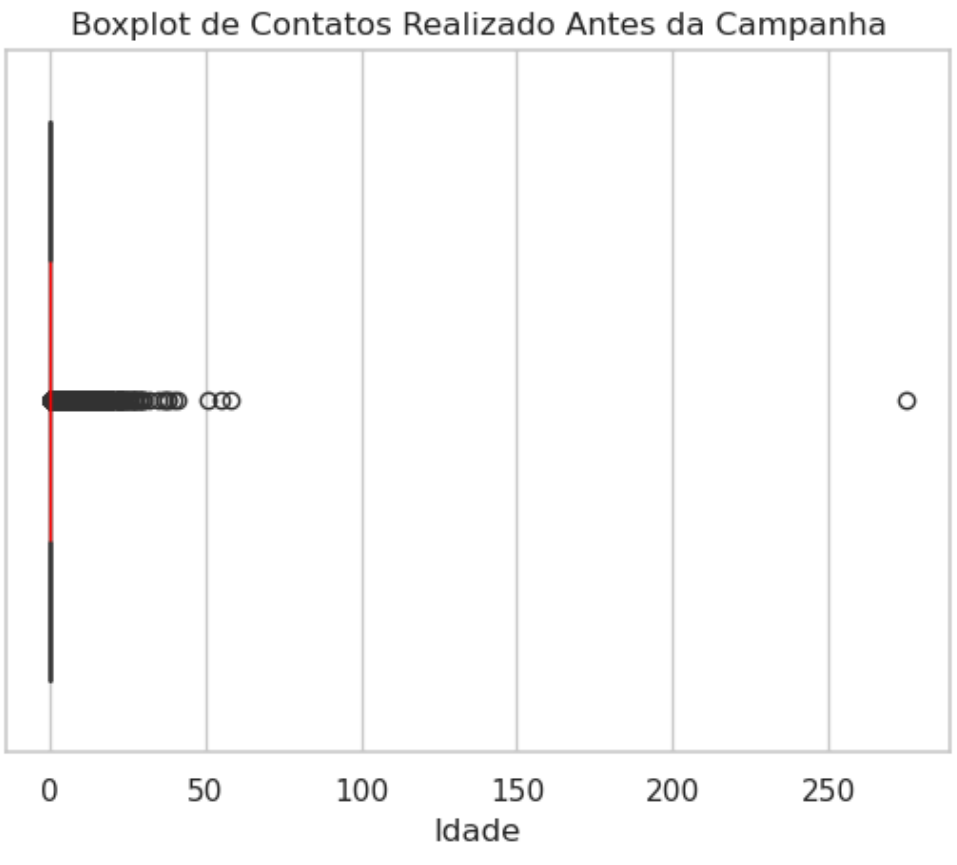
In [ ]:
```
X['previous'].describe()
```

Out[ ]:
```
count   45211.0000
mean        0.5803
std         2.3034
min         0.0000
25%         0.0000
50%         0.0000
75%         0.0000
max       275.0000
Name: previous, dtype: float64
```

Nas estatísticas da variável previous, podemos ver que também há um desequilíbrio entre as observações, com pelo menos 75% dos clientes nunca tendo sido contatados antes da campanha. No entanto, existem alguns possíveis clientes antigos e muito ativos que tiveram até 275 contatos anteriores com o banco.

In [ ]:
```
# Criação do grafico histograma e boxplot
sns.boxplot(X["previous"], orient='h', notch=True, showcaps=False,
            boxprops={"facecolor": (0, .5, .7, .5)},
            medianprops={"color": "r", "linewidth": 1})
plt.title("Boxplot de Contatos Realizado Antes da Campanha")
```

```
plt.xlabel("Idade")
plt.show()
```



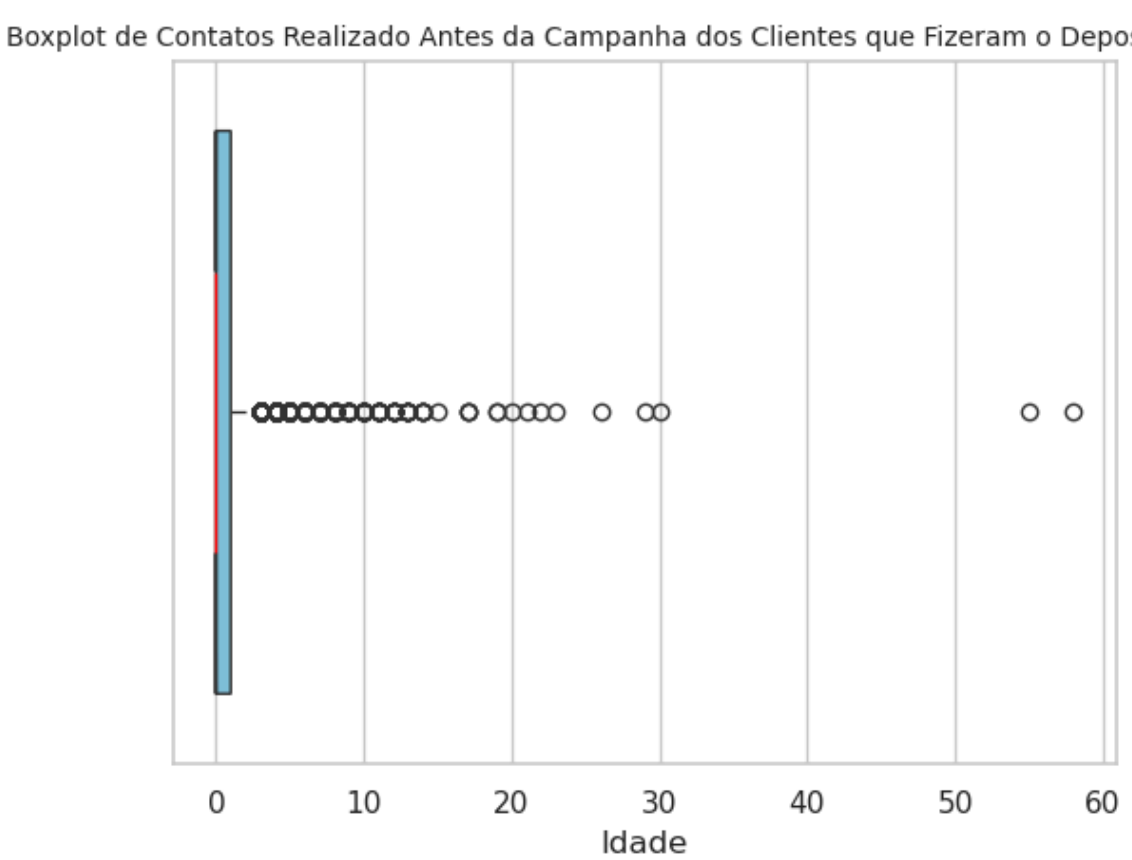
No boxplot acima, podemos ver que a observação de 275 está bem distante das demais, podendo até mesmo ser um erro de digitação. No entanto, devido ao grande número de observações, essa única observação não tem um peso tão significativo. Além disso, podemos afirmar que 99,99% das observações tiveram 50 ou menos contatos com o banco antes da campanha de marketing direto. Para entender quais clientes mais aceitaram a nova oferta do banco, precisamos verificar as estatísticas apenas dos clientes que fizeram o depósito.

In [ ]:
```
X[y.values == 'yes']['previous'].describe()
```

Out[ ]:
```
count   5289.0000
mean       1.1704
std        2.5533
min        0.0000
25%        0.0000
50%        0.0000
75%        1.0000
max       58.0000
Name: previous, dtype: float64
```

Quando analisamos apenas as estatísticas dos clientes que fizeram o investimento com o banco, observamos um aumento na média de contatos. A proporção de clientes que nunca tiveram contato anterior com o banco cai para menos de 75%, com os clientes que tiveram um contato pertencendo ao terceiro quartil. Além disso, notamos que o cliente com o maior número de contatos teve 58 ligações.

In [ ]:
```
# Criação do grafico boxplot dos clientes que aceitaram o produto
sns.boxplot(X[y.values == 'yes']['previous'], orient='h', notch=True, showcaps=False,
            boxprops={"facecolor": (0, .5, .7, .5)},
            medianprops={"color": "r", "linewidth": 1})
plt.title("Boxplot de Contatos Realizado Antes da Campanha dos Clientes que Fizeram o Deposito a Prazo", fontsize=10)
plt.xlabel("Idade")
plt.show()
```



Agora, ao observar o boxplot, notamos uma redução dos outliers. Além disso, podemos afirmar que 99,96% dos clientes que fizeram o depósito tiveram 30 ou menos contatos com o banco antes da última campanha de telemarketing. Isso também reforça uma observação feita anteriormente:

atualmente, as pessoas estão cada vez mais preferindo empresas que não realizam tantas ligações oferecendo produtos, devido ao baixo retorno desse tipo de marketing.

Presultado

Esta é a última variável independente do dataset, representando o resultado da campanha de marketing anterior. Ela é do tipo categórica e assume os seguintes valores:

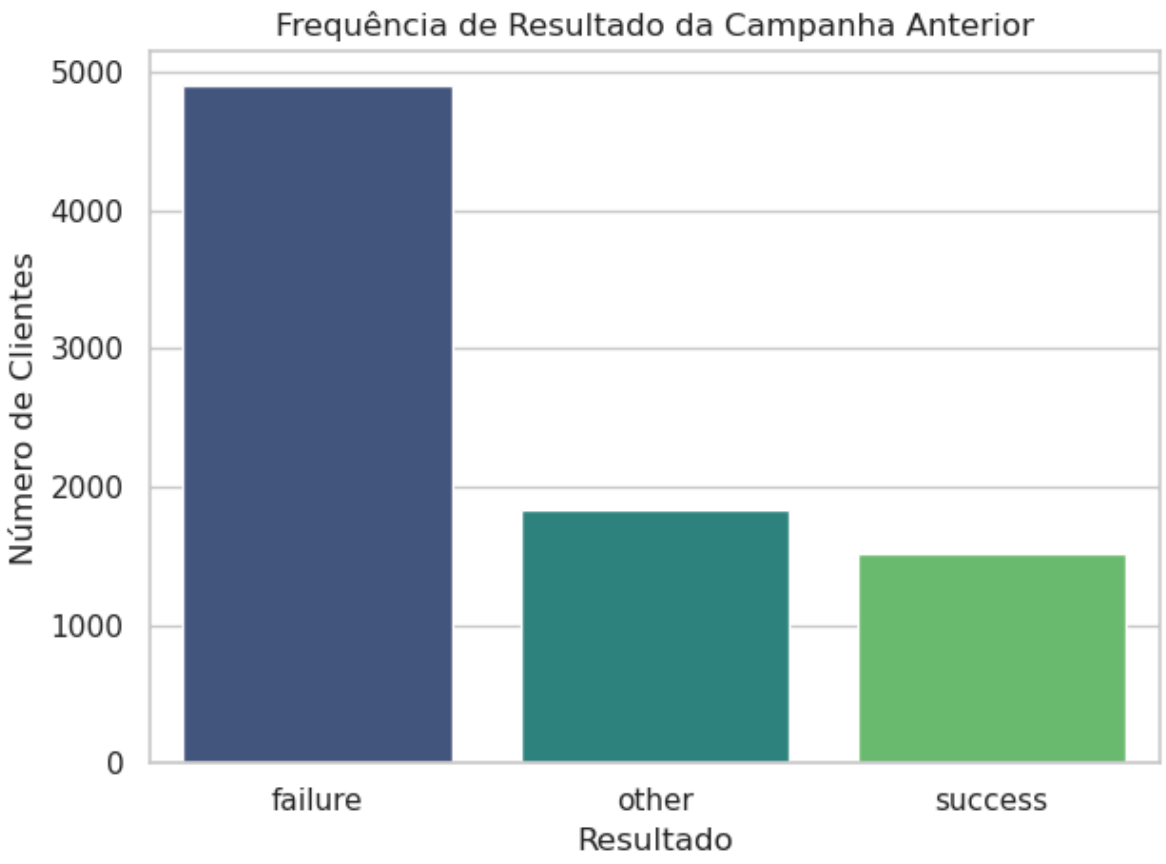
```
Failure: Fracasso
Success: Sucesso
Other: Inexistente
```

Vale lembrar que essa variável é uma das variaveis que mais possui dados faltantes, tendo 36.959 dados ausentes

In [ ]:
```
# Criando uma tabela de frequência dos clientes que fizeram o depósito a prazo
poutcome_counts = X['poutcome'].value_counts().sort_index()
poutcome_percentages= (poutcome_counts * 100) / sum(poutcome_counts)
print(poutcome_percentages)
```

```
poutcome
failure   59.3917
other     22.2976
success   18.3107
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia dos clientes que aceitaram o produto
sns.barplot(x=poutcome_counts.index, y=poutcome_counts.values, palette='viridis')
plt.title('Frequência de Resultado da Campanha Anterior')
plt.xlabel('Resultado')
plt.ylabel('Número de Clientes')
plt.tight_layout()
```

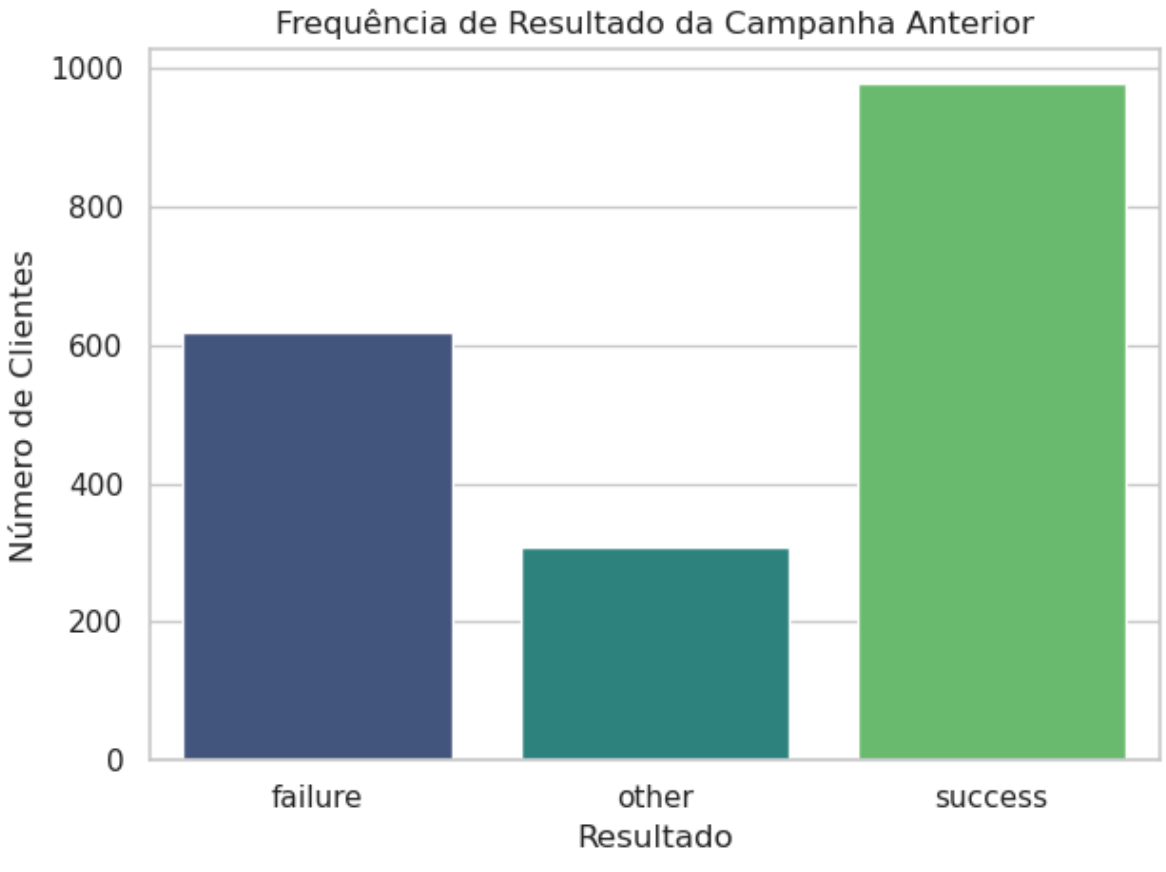


Tanto na tabela quanto no gráfico de barras, podemos visualizar que aproximadamente 59,39% das 8.252 observações têm o valor "Failure" para esta variável. Isso indica que, embora não tenha havido uma conversão anterior, o banco ainda continua realizando campanhas para esses clientes. O sucesso foi registrado em apenas 18,31% das observações.

In [ ]:
```
# Criando uma tabela de frequência dos clientes que fizeram o depósito a prazo
poutcome_y_counts = X[y.values == 'yes']['poutcome'].value_counts().sort_index()
poutcome_y_percentages= (poutcome_y_counts * 100) / sum(poutcome_y_counts)
print(poutcome_y_percentages)
```

```
poutcome
failure   32.4750
other     16.1324
success   51.3925
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia dos clientes que aceitaram o produto
sns.barplot(x=poutcome_y_counts.index, y=poutcome_y_counts.values, palette='viridis')
plt.title('Frequência de Resultado da Campanha Anterior')
plt.xlabel('Resultado')
plt.ylabel('Número de Clientes')
plt.tight_layout()
```



Como podemos ver, o tipo de cliente que mais aceita fazer o depósito a prazo é aquele que já teve uma conversão anterior. Portanto, o banco deve ajustar sua abordagem e focar mais em clientes que já demonstraram interesse anteriormente. No entanto, devido à quantidade significativa de valores ausentes nesta variável, pode ser difícil identificar quais clientes tiveram uma maior conversão.

Variável Resposta

Vamos realizar uma análise exploratória da variáveis a ser classificada no modelo do projeto. Abaixo esta estatística descritiva e visualização da variável.

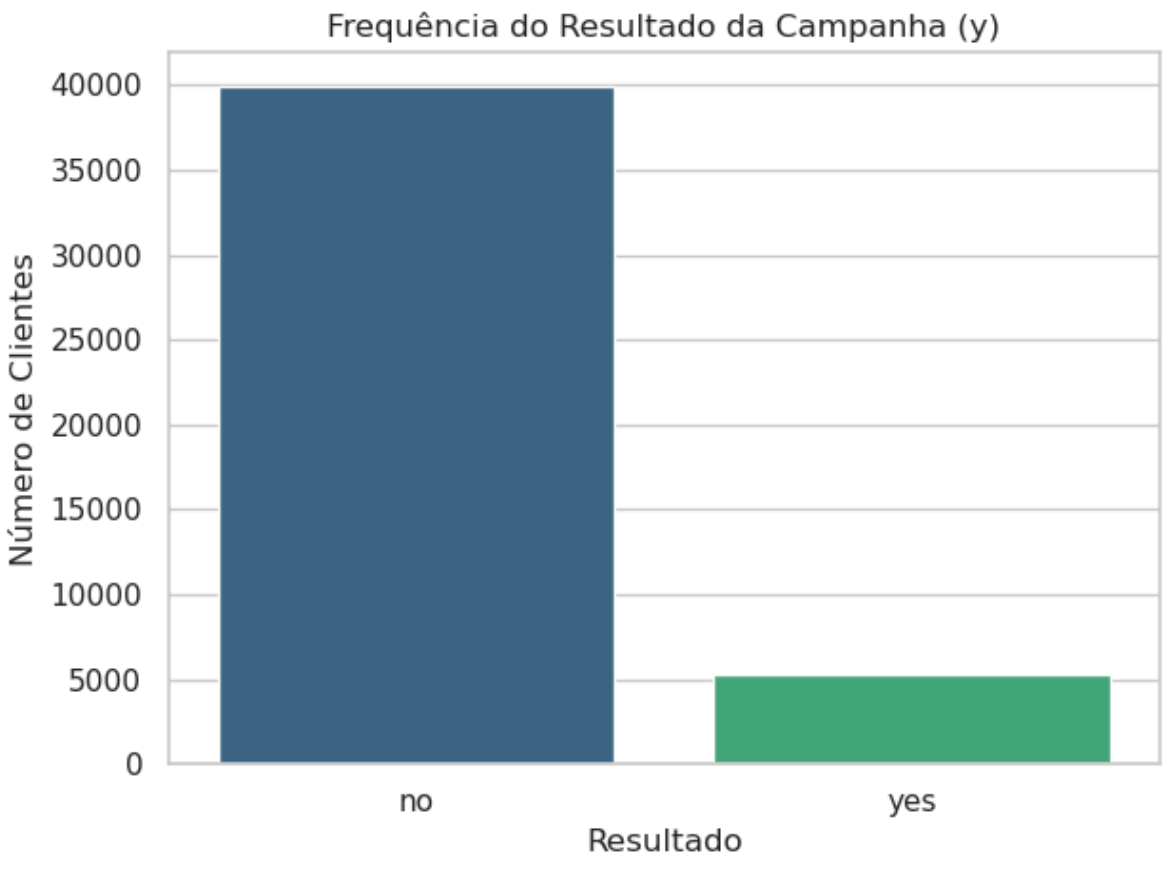
Y

A variável *y* do nosso banco de dados é do tipo binário, indicando se o cliente subscreveu o depósito a prazo (1) ou não (0).

In [ ]:
```
# Criando uma tabela de frequência dos clientes que fizeram o depósito a prazo
y_counts = y.value_counts().sort_index()
y_percentages= (y_counts * 100) / sum(y_counts)
print(y_percentages)
```

```
y
no    88.3015
yes   11.6985
Name: count, dtype: float64
```

In [ ]:
```
# Criando um gráfico de frêquencia dos clientes que aceitaram o produto
sns.barplot(x=[y_counts.index[0][0], y_counts.index[1][0]], y=y_counts.values, palette='viridis')
plt.title('Frequência do Resultado da Campanha (y)')
plt.xlabel('Resultado')
plt.ylabel('Número de Clientes')
plt.tight_layout()
```



Tanto na tabela quanto no gráfico, podemos visualizar que a campanha de telemarketing do banco teve um desempenho insatisfatório, conseguindo captar depósitos de aproximadamente 11,70% dos clientes que participaram da campanha. Portanto, este projeto é crucial para identificar os tipos de

clientes mais propensos a fazer a subscrição do depósito, a fim de melhorar a eficácia das próximas campanhas de marketing.

Devido ao desequilíbrio entre as classes, é de extrema importância aplicar métodos de machine learning para equilibrar as classes e obter melhores resultados nas análises e previsões.

Verificação da Qualidade dos Dados

A verificação da qualidade dos dados envolve a avaliação da completude, consistência e integridade dos dados. Identificamos possíveis problemas, como dados duplicados e inconsistentes, e discutimos propostas de correção e limpeza dos dados. A completude dos dados é analisada verificando a presença de valores ausentes, enquanto a consistência é avaliada através da coerência entre variáveis relacionadas.

Completude dos Dados

Valores Ausentes

A presença de valores ausentes em 'job', 'education', 'contact' e 'poutcome' precisa ser tratada, sendo necessário propostas de métodos de imputação, como substituição pela moda ou uso de técnicas de machine learning.

In [ ]:
```
# verificando valores ausentes no dataset
print(X.isnull().sum())
```

```
age              0
job            288
marital          0
education     1857
default          0
balance          0
housing          0
loan             0
contact      13020
day_of_week      0
month            0
duration         0
campaign         0
pdays            0
previous         0
poutcome     36959
dtype: int64
```

In [ ]:
```
y.value_counts()
```

Out[ ]:
```
y
no     39922
yes     5289
Name: count, dtype: int64
```

Podemos observar na tabela acima o número de observações para cada característica e a quantidade de dados faltantes. O banco de dados possui um total de 45.211 observações. As variáveis job (trabalho), education (educação), contact (contato) e poutcome (resultado da campanha anterior) apresentam 288, 1.857, 13.020 e 36.959 dados faltantes, respectivamente.

Consistência e Integridade

Dados Duplicados

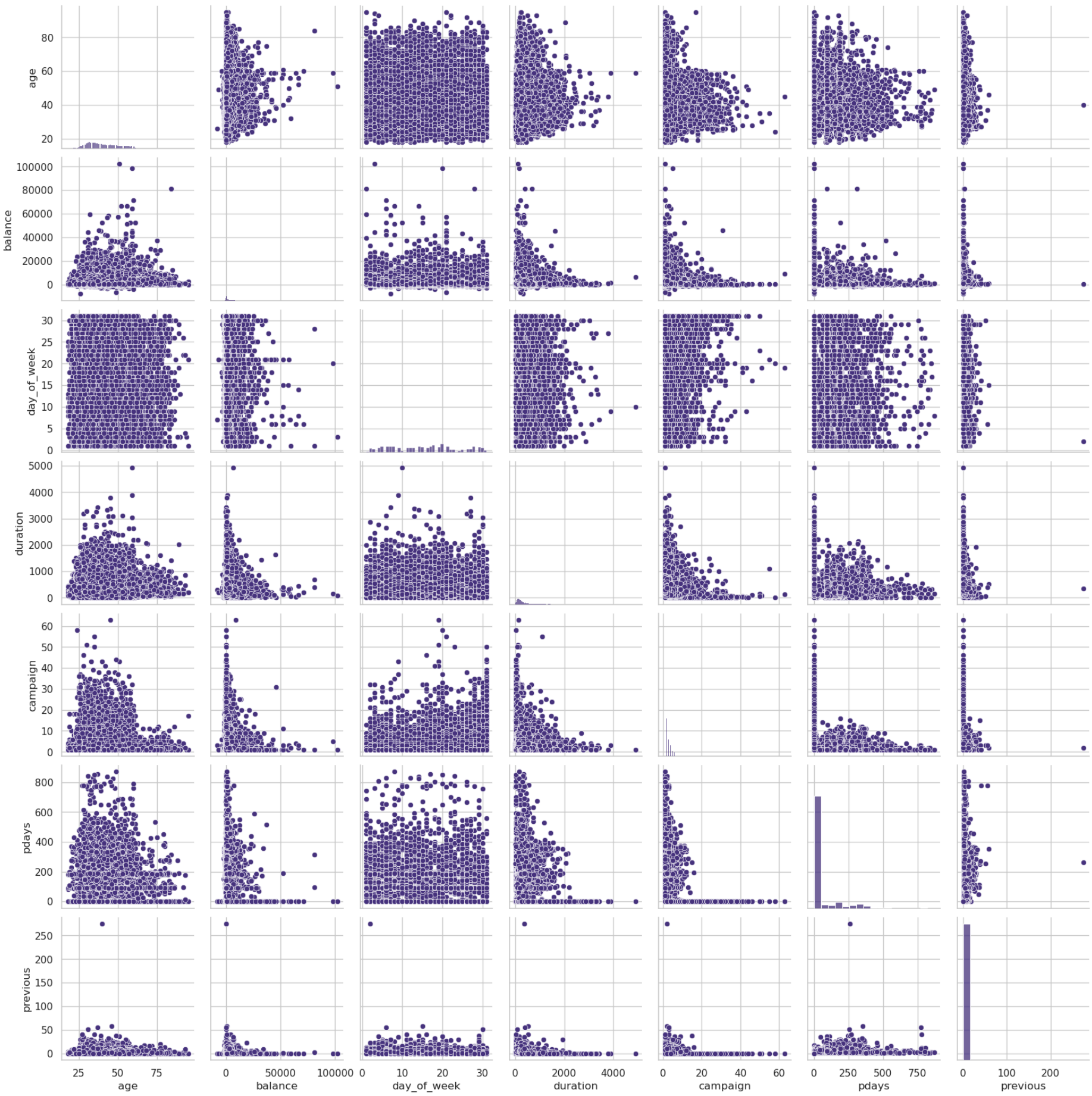
In [ ]:
```
# Verificação de registros duplicados
X.duplicated().sum()
```

Out[ ]:
```
0
```

Inconsistências

In [ ]:
```
# Analisando a correlação dos dados
sns.pairplot(X)
```

Out[ ]:
```
<seaborn.axisgrid.PairGrid at 0x7fe24c8a2dd0>
```



In [ ]:
```
# Convertendo variáveis categóricas para numéricas usando codificação one-hot
df_encoded = pd.get_dummies(X, columns=['job','marital',
                                        'education','default',
                                        'housing','loan',
                                        'contact','month',
                                        'poutcome'], drop_first=True)

# Matriz de correlação
print(df_encoded.corr())
```

	age	balance	day_of_week	duration	campaign	pdays	\
age	1.0000	0.0978	-0.0091	-0.0046	0.0048	-0.0238	
balance	0.0978	1.0000	0.0045	0.0216	-0.0146	0.0034	
day_of_week	-0.0091	0.0045	1.0000	-0.0302	0.1625	-0.0930	
duration	-0.0046	0.0216	-0.0302	1.0000	-0.0846	-0.0016	
campaign	0.0048	-0.0146	0.1625	-0.0846	1.0000	-0.0886	
pdays	-0.0238	0.0034	-0.0930	-0.0016	-0.0886	1.0000	
previous	0.0013	0.0167	-0.0517	0.0012	-0.0329	0.4548	
job_blue-collar	-0.0440	-0.0488	-0.0229	0.0096	0.0090	0.0201	
job_entrepreneur	0.0218	0.0096	-0.0023	-0.0013	0.0021	-0.0142	
job_housemaid	0.0867	0.0017	0.0040	-0.0080	0.0031	-0.0313	
job_management	-0.0236	0.0678	0.0190	-0.0083	0.0167	-0.0079	
job_retired	0.4474	0.0469	-0.0101	0.0260	-0.0309	-0.0063	
job_self-employed	-0.0081	0.0179	0.0051	0.0074	0.0055	-0.0104	
job_services	-0.0658	-0.0382	-0.0065	0.0014	-0.0047	0.0057	
job_student	-0.1973	0.0012	-0.0159	-0.0065	-0.0218	0.0245	
job_technician	-0.0686	-0.0162	0.0325	-0.0092	0.0207	-0.0135	
job_unemployed	0.0004	0.0090	-0.0064	0.0203	-0.0184	-0.0104	
marital_married	0.2863	0.0257	0.0071	-0.0227	0.0314	-0.0276	
marital_single	-0.4278	-0.0125	-0.0074	0.0203	-0.0231	0.0279	
education_secondary	-0.0940	-0.0699	-0.0058	0.0021	-0.0209	0.0221	
education_tertiary	-0.0816	0.0840	0.0217	0.0009	0.0129	-0.0076	
default_yes	-0.0179	-0.0667	0.0094	-0.0100	0.0168	-0.0300	
housing_yes	-0.1855	-0.0688	-0.0280	0.0051	-0.0236	0.1242	
loan_yes	-0.0157	-0.0844	0.0114	-0.0124	0.0100	-0.0228	
contact_telephone	0.1703	0.0380	0.0237	-0.0232	0.0539	0.0160	
month_aug	0.0738	0.0086	0.0301	-0.0401	0.1504	-0.1074	
month_dec	0.0229	0.0216	-0.0114	0.0191	-0.0126	0.0472	
month_feb	-0.0012	-0.0035	-0.2833	-0.0096	-0.0307	0.0710	
month_jan	-0.0075	-0.0244	0.2505	0.0070	-0.0631	0.0495	
month_jul	0.0029	-0.0644	0.1472	0.0162	0.1041	-0.1363	
month_jun	0.0518	0.0296	-0.1938	-0.0214	0.0439	-0.1135	
month_mar	0.0195	0.0232	-0.0207	-0.0055	-0.0186	0.0320	
month_may	-0.1274	-0.0711	-0.0251	0.0071	-0.0676	0.0790	
month_nov	0.0328	0.1173	0.0961	-0.0060	-0.0847	0.0079	
month_oct	0.0601	0.0402	0.0305	0.0151	-0.0510	0.0568	
month_sep	0.0324	0.0219	-0.0539	0.0151	-0.0367	0.0844	
poutcome_other	-0.0230	0.0085	-0.0330	-0.0020	-0.0201	0.3898	
poutcome_success	0.0355	0.0352	-0.0303	0.0424	-0.0575	0.2285	

	previous	job_blue-collar	job_entrepreneur	\
age	0.0013	-0.0440	0.0218	
balance	0.0167	-0.0488	0.0096	
day_of_week	-0.0517	-0.0229	-0.0023	
duration	0.0012	0.0096	-0.0013	
campaign	-0.0329	0.0090	0.0021	
pdays	0.4548	0.0201	-0.0142	
previous	1.0000	-0.0171	-0.0082	
job_blue-collar	-0.0171	1.0000	-0.0966	
job_entrepreneur	-0.0082	-0.0966	1.0000	
job_housemaid	-0.0152	-0.0880	-0.0310	
job_management	0.0196	-0.2694	-0.0949	
job_retired	0.0058	-0.1203	-0.0423	
job_self-employed	-0.0024	-0.0996	-0.0351	
job_services	-0.0109	-0.1666	-0.0587	
job_student	0.0236	-0.0762	-0.0268	
job_technician	-0.0011	-0.2354	-0.0829	
job_unemployed	-0.0085	-0.0902	-0.0318	
marital_married	-0.0127	0.1220	0.0443	
marital_single	0.0170	-0.0883	-0.0503	
education_secondary	-0.0056	0.0405	-0.0549	
education_tertiary	0.0229	-0.3205	0.0676	
default_yes	-0.0183	0.0103	0.0263	
housing_yes	0.0371	0.1775	0.0106	
loan_yes	-0.0110	0.0183	0.0398	
contact_telephone	0.0281	-0.0032	-0.0043	
month_aug	-0.0525	-0.1147	-0.0444	
month_dec	0.0366	-0.0267	-0.0073	
month_feb	0.0652	-0.0383	-0.0001	
month_jan	0.0470	-0.0363	-0.0058	
month_jul	-0.0829	-0.0132	0.0259	
month_jun	-0.0608	0.0217	0.0155	
month_mar	0.0273	-0.0414	-0.0166	
month_may	0.0013	0.1654	-0.0099	
month_nov	0.0379	-0.0480	0.0510	
month_oct	0.0539	-0.0424	-0.0120	
month_sep	0.0650	-0.0448	-0.0078	
poutcome_other	0.3066	0.0013	-0.0135	
poutcome_success	0.2014	-0.0531	-0.0191	

	job_housemaid	...	month_jan	month_jul	month_jun	\
age	0.0867	...	-0.0075	0.0029	0.0518	
balance	0.0017	...	-0.0244	-0.0644	0.0296	
day_of_week	0.0040	...	0.2505	0.1472	-0.1938	
duration	-0.0080	...	0.0070	0.0162	-0.0214	
campaign	0.0031	...	-0.0631	0.1041	0.0439	
pdays	-0.0313	...	0.0495	-0.1363	-0.1135	
previous	-0.0152	...	0.0470	-0.0829	-0.0608	
job_blue-collar	-0.0880	...	-0.0363	-0.0132	0.0217	
job_entrepreneur	-0.0310	...	-0.0058	0.0259	0.0155	
job_housemaid	1.0000	...	-0.0051	0.0350	0.0527	
job_management	-0.0864	...	-0.0011	-0.0138	-0.0324	

job_retired	-0.0386	...	0.0110	-0.0018	0.0112	
job_self-employed	-0.0319	...	0.0028	0.0021	0.0099	
job_services	-0.0534	...	0.0027	0.0303	0.0062	
job_student	-0.0244	...	0.0089	-0.0307	-0.0124	
job_technician	-0.0755	...	0.0045	-0.0177	-0.0378	
job_unemployed	-0.0289	...	0.0454	-0.0080	0.0021	
marital_married	0.0458	...	-0.0379	0.0247	0.0154	
marital_single	-0.0622	...	0.0397	-0.0383	-0.0251	
education_secondary	-0.0654	...	-0.0036	0.0177	-0.0200	
education_tertiary	-0.0570	...	0.0138	-0.0233	-0.0381	
default_yes	-0.0004	...	-0.0070	0.0443	0.0076	
housing_yes	-0.0794	...	-0.0664	-0.0612	-0.1023	
loan_yes	-0.0172	...	-0.0045	0.1678	-0.0227	
contact_telephone	0.0394	...	0.0202	0.1026	-0.0736	
month_aug	0.0352	...	-0.0717	-0.1699	-0.1466	
month_dec	0.0003	...	-0.0123	-0.0293	-0.0252	
month_feb	-0.0125	...	-0.0446	-0.1058	-0.0913	
month_jan	-0.0051	...	1.0000	-0.0759	-0.0655	
month_jul	0.0350	...	-0.0759	1.0000	-0.1553	
month_jun	0.0527	...	-0.0655	-0.1553	1.0000	
month_mar	-0.0001	...	-0.0185	-0.0438	-0.0378	
month_may	-0.0667	...	-0.1184	-0.2807	-0.2422	
month_nov	-0.0133	...	-0.0555	-0.1316	-0.1136	
month_oct	0.0062	...	-0.0231	-0.0546	-0.0471	
month_sep	-0.0023	...	-0.0204	-0.0483	-0.0417	
poutcome_other	-0.0168	...	0.0567	-0.0731	-0.0511	
poutcome_success	-0.0094	...	0.0121	-0.0450	-0.0231	

	month_mar	month_may	month_nov	month_oct	month_sep	\
age	0.0195	-0.1274	0.0328	0.0601	0.0324	
balance	0.0232	-0.0711	0.1173	0.0402	0.0219	
day_of_week	-0.0207	-0.0251	0.0961	0.0305	-0.0539	
duration	-0.0055	0.0071	-0.0060	0.0151	0.0151	
campaign	-0.0186	-0.0676	-0.0847	-0.0510	-0.0367	
pdays	0.0320	0.0790	0.0079	0.0568	0.0844	
previous	0.0273	0.0013	0.0379	0.0539	0.0650	
job_blue-collar	-0.0414	0.1654	-0.0480	-0.0424	-0.0448	
job_entrepreneur	-0.0166	-0.0099	0.0510	-0.0120	-0.0078	
job_housemaid	-0.0001	-0.0667	-0.0133	0.0062	-0.0023	
job_management	0.0235	-0.0841	0.0510	0.0097	0.0232	
job_retired	0.0418	-0.0739	-0.0221	0.0777	0.0613	
job_self-employed	-0.0008	-0.0300	0.0410	0.0002	-0.0077	
job_services	-0.0179	0.0541	-0.0167	-0.0289	-0.0219	
job_student	0.0442	-0.0110	-0.0166	0.0290	0.0524	
job_technician	-0.0134	-0.0351	-0.0069	-0.0084	-0.0180	
job_unemployed	0.0068	-0.0370	0.0152	0.0018	0.0062	
marital_married	-0.0182	-0.0367	0.0224	-0.0097	-0.0123	
marital_single	0.0231	0.0340	-0.0309	0.0129	0.0189	
education_secondary	-0.0259	0.0814	-0.0201	-0.0233	-0.0268	
education_tertiary	0.0374	-0.1164	0.0549	0.0268	0.0279	
default_yes	-0.0140	-0.0029	0.0061	-0.0175	-0.0140	
housing_yes	-0.0663	0.4280	0.0012	-0.0854	-0.0763	
loan_yes	-0.0298	-0.0279	0.0192	-0.0301	-0.0337	
contact_telephone	0.0197	-0.0833	0.0395	0.0595	0.0231	
month_aug	-0.0413	-0.2649	-0.1242	-0.0516	-0.0456	
month_dec	-0.0071	-0.0456	-0.0214	-0.0089	-0.0079	
month_feb	-0.0258	-0.1651	-0.0774	-0.0321	-0.0284	
month_jan	-0.0185	-0.1184	-0.0555	-0.0231	-0.0204	
month_jul	-0.0438	-0.2807	-0.1316	-0.0546	-0.0483	
month_jun	-0.0378	-0.2422	-0.1136	-0.0471	-0.0417	
month_mar	1.0000	-0.0683	-0.0320	-0.0133	-0.0118	
month_may	-0.0683	1.0000	-0.2053	-0.0852	-0.0754	
month_nov	-0.0320	-0.2053	1.0000	-0.0400	-0.0353	
month_oct	-0.0133	-0.0852	-0.0400	1.0000	-0.0147	
month_sep	-0.0118	-0.0754	-0.0353	-0.0147	1.0000	
poutcome_other	0.0204	0.0075	0.0199	0.0274	0.0373	
poutcome_success	0.0531	-0.0599	0.0036	0.1033	0.1233	

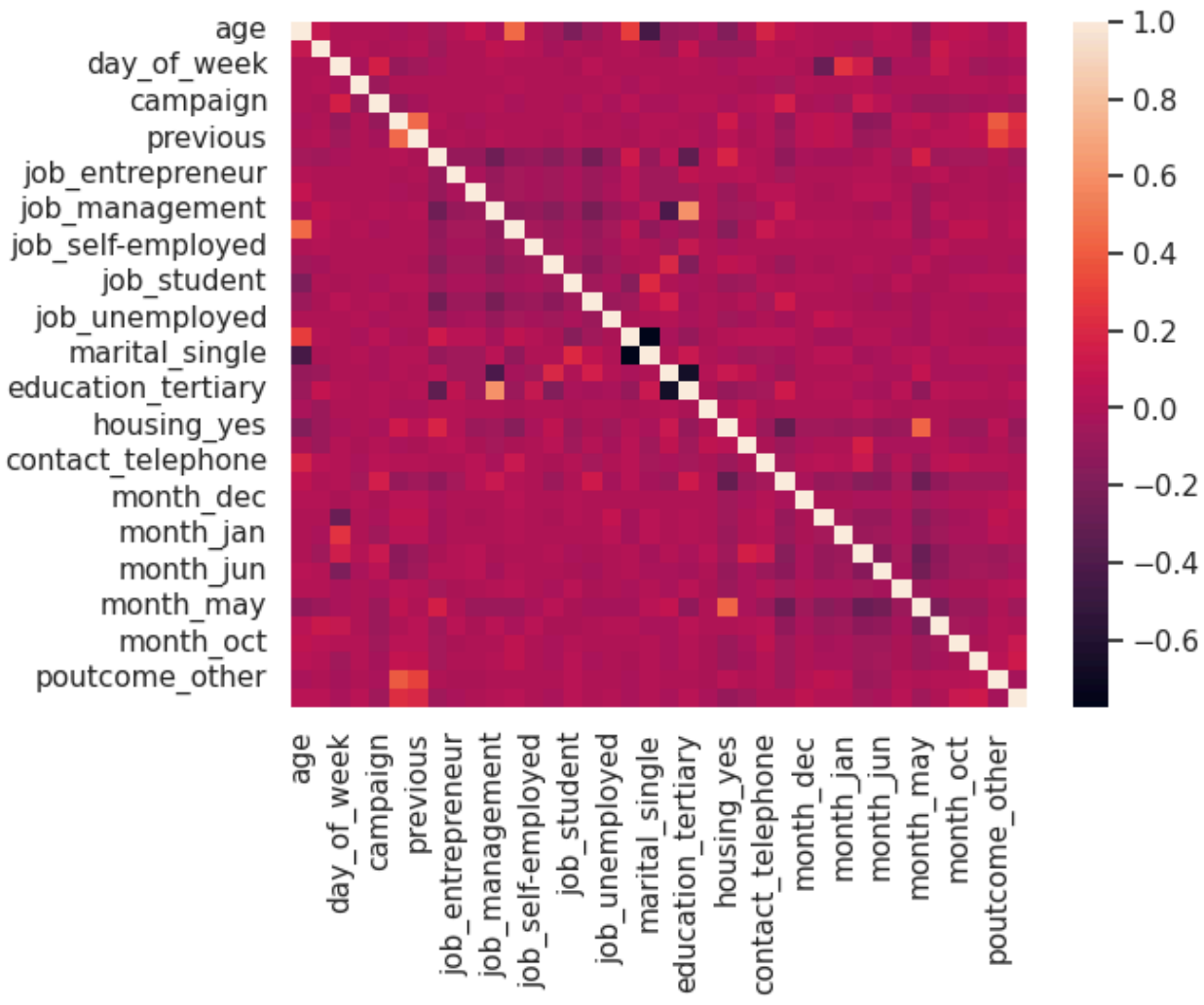
	poutcome_other	poutcome_success
age	-0.0230	0.0355
balance	0.0085	0.0352
day_of_week	-0.0330	-0.0303
duration	-0.0020	0.0424
campaign	-0.0201	-0.0575
pdays	0.3898	0.2285
previous	0.3066	0.2014
job_blue-collar	0.0013	-0.0531
job_entrepreneur	-0.0135	-0.0191
job_housemaid	-0.0168	-0.0094
job_management	0.0022	0.0215
job_retired	-0.0047	0.0555
job_self-employed	-0.0020	0.0015
job_services	0.0031	-0.0229
job_student	0.0336	0.0480
job_technician	-0.0030	-0.0029
job_unemployed	-0.0107	0.0150
marital_married	-0.0276	-0.0185
marital_single	0.0272	0.0261
education_secondary	0.0096	-0.0247
education_tertiary	0.0007	0.0479
default_yes	-0.0144	-0.0233
housing_yes	0.0397	-0.0914

```
loan_yes                -0.0091          -0.0537
contact_telephone        0.0264           0.0085
month_aug               -0.0546          -0.0006
month_dec                0.0282           0.0786
month_feb                0.0687           0.0285
month_jan                0.0567           0.0121
month_jul               -0.0731          -0.0450
month_jun               -0.0511          -0.0231
month_mar                0.0204           0.0531
month_may                0.0075          -0.0599
month_nov                0.0199           0.0036
month_oct                0.0274           0.1033
month_sep                0.0373           0.1233
poutcome_other           1.0000          -0.0383
poutcome_success        -0.0383           1.0000

[38 rows x 38 columns]
```

In [ ]:
```
# Gráfico de calor para visualizar a correlação
sns.heatmap(df_encoded.corr())
```

Out[ ]: <Axes: >



Análise Crítica das Descobertas

Durante a exploração dos dados do conjunto Bank Marketing, identificamos várias descobertas significativas que possuem implicações diretas para o contexto do problema de negócio. Uma das principais descobertas foi a influência das variáveis relacionadas ao contato anterior, como "poutcome" (resultado da campanha de marketing anterior) e "duration" (duração do último contato), no sucesso da campanha atual. Essas variáveis mostraram-se fortes preditores de resposta positiva à campanha, indicando que o histórico de interações com clientes é crucial para a segmentação e personalização de campanhas futuras.

Outra descoberta importante foi a relevância das características socioeconômicas, como "education" (educação) e "job" (profissão), que também mostraram uma correlação significativa com a aceitação da oferta de depósito a prazo. Estas descobertas sugerem que as campanhas de marketing direcionadas, levando em conta o perfil socioeconômico dos clientes, podem aumentar a eficácia das ações.

De acordo com a verificação da qualidade dos dados, a partir do gráfico de pares, da matriz de correlação e do mapa de calor, observamos que os dados são não lineares e assimétricos. Isso implica que a seleção de recursos não dependerá exclusivamente do fator de correlação. Além disso, nenhum recurso individual está completamente correlacionado com a classe alvo, o que indica a necessidade de combinar múltiplos recursos para melhorar a capacidade preditiva dos modelos.

No entanto, encontramos algumas limitações nos dados que representam desafios. Por exemplo, a presença de valores ausentes e discrepantes pode distorcer as análises e modelos preditivos. Além disso, a variável "duration" pode introduzir vieses, pois a duração de uma chamada só é conhecida após a chamada ter sido feita, tornando-a um preditor que não pode ser utilizado para previsões em tempo real.

Propostas de Alterações para a Preparação dos Dados

Para melhorar a qualidade e a adequação dos dados para modelagem, propomos várias alterações. Primeiramente, a limpeza de dados deve ser realizada para tratar os valores ausentes. A imputação de valores ausentes pode ser feita utilizando a média para variáveis numéricas e a moda para variáveis categóricas, ou através de métodos mais avançados como algoritmos de aprendizado de máquina.

Além disso, sugerimos a remoção de outliers que podem distorcer os resultados das análises. Os outliers podem ser identificados utilizando técnicas estatísticas, como o método do IQR (Intervalo Interquartil), e removidos ou transformados conforme apropriado.

Para a transformação dos dados, recomendamos a normalização das variáveis numéricas para garantir que todas as variáveis estejam na mesma escala, o que é particularmente importante para algoritmos sensíveis à escala, como a SVM. A codificação de variáveis categóricas deve ser feita utilizando técnicas como one-hot encoding para evitar a introdução de ordens artificiais entre categorias.

Justificamos essas propostas com base na necessidade de garantir que os dados sejam limpos, consistentes e adequados para os modelos preditivos que serão utilizados. Essas alterações são essenciais para melhorar a precisão e a robustez dos modelos de aprendizado de máquina.

Conclusão

Em resumo, a análise dos dados do conjunto Bank Marketing revelou insights valiosos sobre os fatores que influenciam a aceitação de ofertas de depósito a prazo. As principais descobertas destacam a importância do histórico de interações com clientes e das características socioeconômicas no sucesso das campanhas de marketing.

Propomos várias alterações para a preparação dos dados, incluindo a imputação de valores ausentes, a remoção de outliers, a normalização de variáveis numéricas e a codificação de variáveis categóricas, visando melhorar a qualidade e a adequação dos dados para a modelagem.

Os próximos passos na fase de preparação dos dados incluem a implementação dessas propostas e a validação das melhorias na qualidade dos dados resultantes. A compreensão aprofundada dos dados é fundamental para o sucesso do projeto, pois permite a construção de modelos preditivos mais precisos e eficientes, resultando em campanhas de marketing mais eficazes e direcionadas