



Locadora de Veículos

Uni - Car

Colaboradores

Nomes:

Registro dos alunos:

Gustavo Souza Lopes -----

RA: 2224100154

Felipe C. R. Franceschetti -----

RA: 2224100180

João Pedro C. Silva -----

RA: 2224102613

Sabrina Pereira de Melo -----

RA: 2224107435

Guilherme Henrique R. Soares -----

RA: 2224104232

Yan de Souza Santos -----

RA: 2224107572

Kevin Rodrigues Dias -----

RA: 2224105088

João Vitor D. do N. Silva -----

RA: 2224104384

Victor Gabriel Souza Lopes -----

RA: 2224107866

Serviços Oferecidos pela Empresa

- ***Locação de Veículos***

Econômicos

Suv's

Vans

Carros de alto padrão

Locação por dia, semana e Mês: Flexibilidade para atender diferentes necessidades dos clientes.

- ***Reservas Online***

Sistemas de Reserva: Plataforma online para facilitar a reserva e pagamento.

Confirmação instantânea: Recebimento de confirmação imediata por e-mail ou sms

- ***Serviços Adicionais***

Entrega e coleta: Opção de entrega e coleta dos veículos em locais convenientes (ex: aeroportos, hotéis, endereços pessoais). Assentos para Crianças: Oferecimento de assentos para crianças, em conformidade com a legislação local.

- ***Cobertura de Seguro***

Planos de Seguro: Opções de seguro básico e abrangente para proteção do veículo e do condutor.

Assistência na Estrada: Serviço de assistência 24 horas em caso de emergências.

- ***Serviços de Manutenção***

Chaveiro e Mecânico: Serviços básicos de manutenção 24 horas como pane e problema com pneus

- ***Programa de Fidelidade***

Descontos e Benefícios: Programa de pontos ou descontos para clientes frequentes.

Ofertas Especiais: Promoções exclusivas para membros do programa de fidelidade.

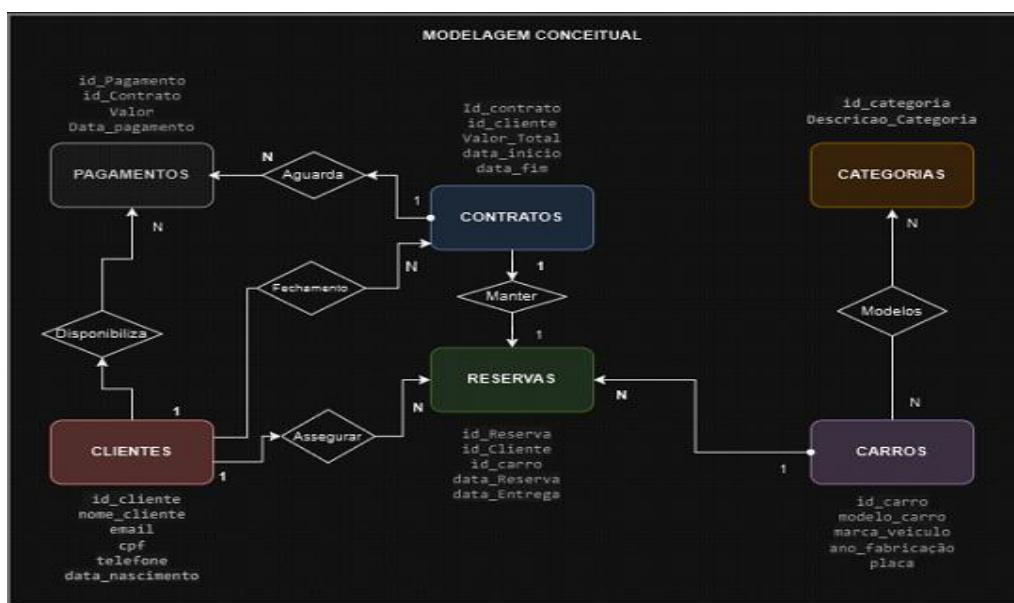
- **Serviço de Atendimento ao Cliente**

Suporte 24/7: Equipe disponível para suporte em caso de dúvidas ou problemas.

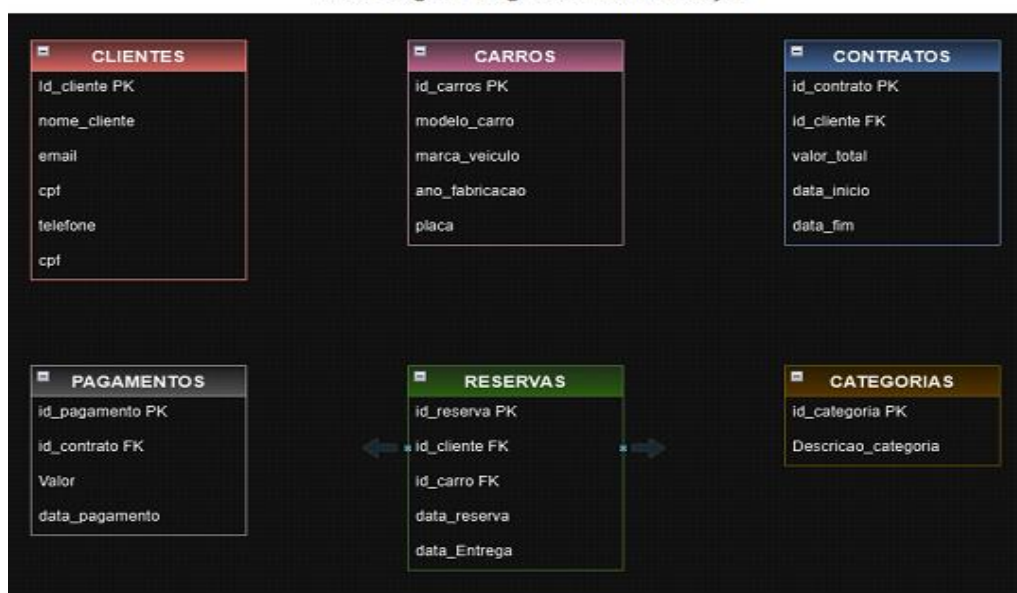
Atendimento Personalizado: Consultores treinados para ajudar os clientes a escolher o veículo ideal.

- **Política de Combustível**

Opções de Combustível: Política flexível de combustível (cheio a cheio ou pré-pago)



Modelagem Lógica e Normalização



ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
id_cliente	INT	Identificação do Cliente	PK
nome_cliente	VARCHAR (50)	Nome do Cliente	
email	VARCHAR (100)	Email do Cliente	
cpf	VARCHAR (11)	Numeração do CPF	
telefone	INT	Telefone para contato	
data_nasc	DATE	data de nascimento do cliente	

ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
Carros	INT	Identificação de cada Carro	PK
modelo_carro	VARCHAR (40)	Nome do modelo do veículo	
marca_veiculo	VARCHAR (30)	Marca do veículo	
ano_fabricacao	INT	Ano de Fabricação do veículo	
placa	VARCHAR (10)	Placa veículo	

ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
id_contrato	INT	Identificação do Contrato	PK
id_cliente	INT	Identificação do Cliente	FK
valor_total	DECIMAL	Valor total do Contrato	
data_inicio	DATE	Data do inicio do Contrato	
data_fim	DATE	Data Final do Contrato	

ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
id_pagamento	INT	Identificação do Pagamento do contrato	PK
id_contrato	INT	Identificação do Contrato	FK
Valor	DECIMAL	Valor do Contrato	
data_pagamento	DATE	Data do Pagamento	

ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
id_categoria	INT	Identificação numérica da categoria do veículo	PK
Descricao_categoria	VARCHAR (20)	Descrição da categoria de cada carro	
ATRIBUTO	TIPO	DESCRIÇÃO	CHAVE
id_Reserva	INT	Identificação da Reserva do veículo	PK
id_Cliente	INT	Identificação do Cliente	FK
id_Carro	INT	Identificação numérica de cada Carro	FK
Data_Reserva	DATE	Data da Reserva do Veículo	
Data_Entrega	DATE	Data de Entrega do Veículo	

Exploração de dados

```

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_absolute_error, mean_squared_error

teste_csv = pd.read_csv('car_rental_data.csv')

teste_df = pd.DataFrame(teste_csv)

remov_dupli = teste_df.drop_duplicates(keep="last")

remov_vlr_null = teste_df.dropna(thresh=2)

duplicatas = teste_csv.duplicated()

X_simple = remov_vlr_null[['purchase_value']]

y = remov_vlr_null['selling_value']

```

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

```
model_regression = LinearRegression()
```

```
model_regression.fit(X_train, y_train)
```

```
y_pred = model_regression.predict(X_test)
```

```
mae = mean_absolute_error(y_test, y_pred)
```

```
mse = mean_squared_error(y_test, y_pred)
```

```
mae_percentage = (mae / y_test.mean()) * 100
```

```
mse_percentage = (mse / y_test.mean()) * 100
```

```
print("Erro Médio Absoluto (MAE):", mae)
```

```
print("Erro Quadrático Médio (MSE):", mse)
```

```
print("Porcentagem de Erro MAE:", mae_percentage, "%")
```

```
print("Porcentagem de Erro MSE:", mse_percentage, "%")
```

```
plt.figure(figsize=(10, 6))
```

```
plt.scatter(X_test, y_test, color='blue', label='Valor Real', alpha=0.5)
```

```
plt.plot(X_test, y_pred, color='red', label='Valor Previsto', alpha=0.5)
```

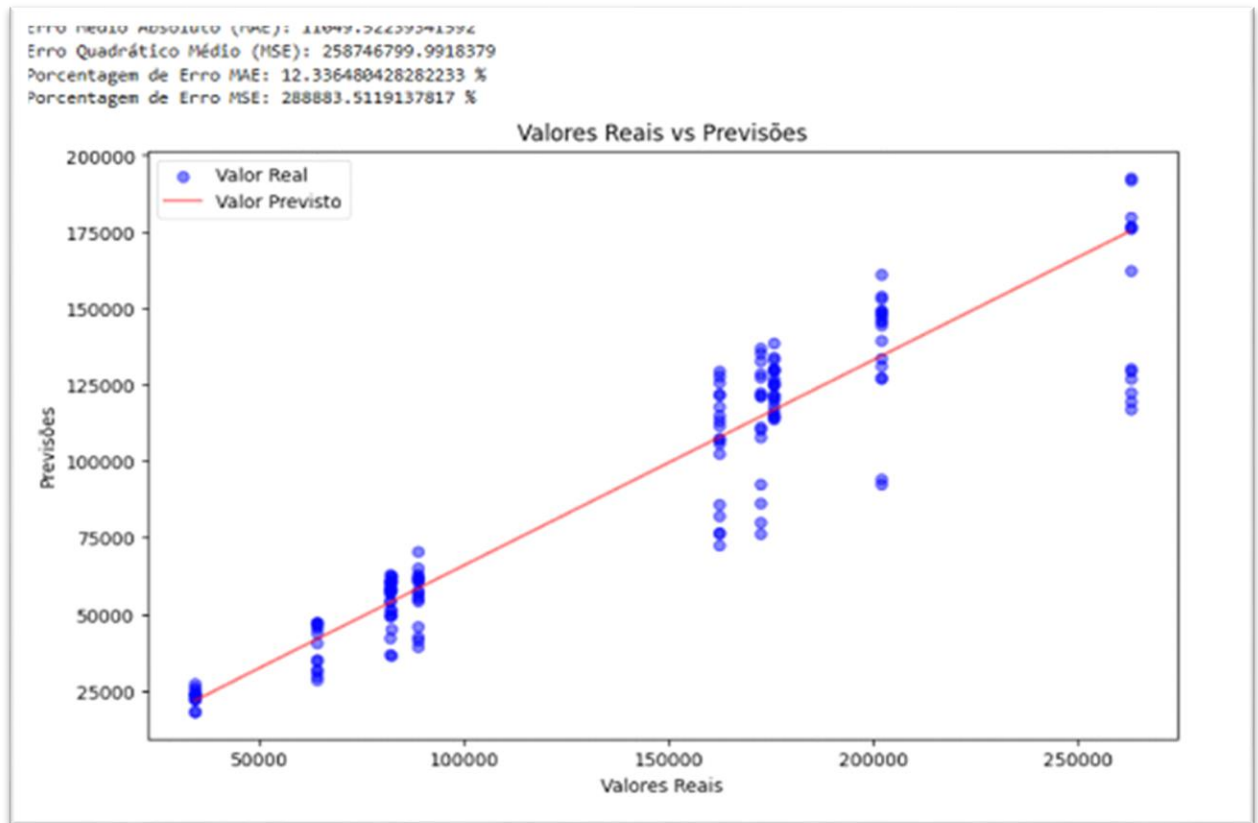
```
plt.xlabel('Valores Reais')
```

```
plt.ylabel('Previsões')
```

```
plt.title('Valores Reais vs Previsões')
```

```
plt.legend()
```

```
plt.show()
```



Implementação e aprendizado

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.linear_model import LinearRegression
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,  
confusion_matrix
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
teste_csv = pd.read_csv('car_rental_data.csv')
```

```
teste_df = pd.DataFrame(teste_csv)
```

```
remov_dupli = teste_df.drop_duplicates(keep="last")
```



```
remov_vlr_null = teste_df.dropna(thresh=2)
```

```
duplicatas = teste_csv.duplicated()
```

```
y_classification = pd.cut(remov_vlr_null['purchase_value'], bins=3, labels=['baixo',  
'médio', 'alto'])
```

```
X_simple = remov_vlr_null [['selling_value']]
```

```
X_train, X_test, y_train, y_test = train_test_split(X_simple, y_classification, test_size=0.2,  
random_state=42)
```

```
model_knn = KNeighborsClassifier(n_neighbors=5)
```

```
model_knn.fit(X_train, y_train)
```

```
y_pred = model_knn.predict(X_test)
```

```
accuracy = accuracy_score(y_test, y_pred)
```

```
precision = precision_score(y_test, y_pred, average='weighted')
```

```
recall = recall_score(y_test, y_pred, average='weighted')
```

```
f1 = f1_score(y_test, y_pred, average='weighted')
```

```
print("Acurácia:", accuracy)
```

```
print("Precisão:", precision)
```

```
print("Recall:", recall)
```

```
print("F1-score:", f1)
```

```
cm = confusion_matrix(y_test, y_pred)
```

```
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False)
```

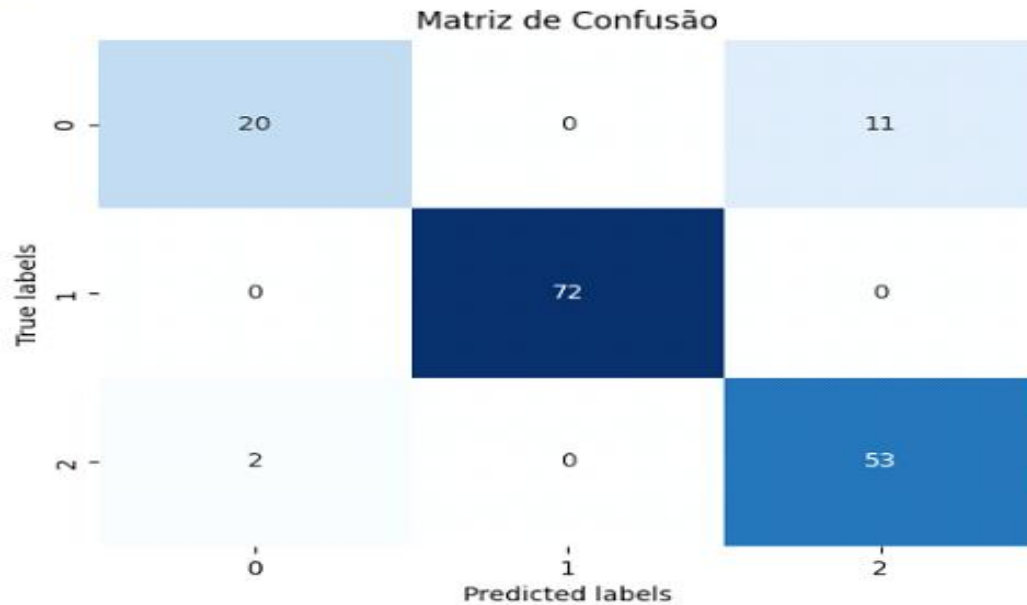
```
plt.xlabel('Predicted labels')
```

```
plt.ylabel('True labels')
```

```
plt.title('Matriz de Confusão')
```

```
plt.show()
```

Acurácia: 0.9177215189873418
 Precisão: 0.922333501150748
 Recall: 0.9177215189873418
 F1-score: 0.9138469813528468



```
import pandas as pd
from sklearn.model_selection import cross_val_score, KFold
from sklearn.cluster import KMeans
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score

teste_csv = pd.read_csv('car_rental_data.csv')
teste_df = pd.DataFrame(teste_csv)
remov_dupli = teste_df.drop_duplicates(keep="last")
remov_vlr_null = remov_dupli.dropna(thresh=2)

cols_to_drop = ['purchase_value', 'selling_value']
cols_to_drop = [col for col in cols_to_drop if col in remov_vlr_null.columns]
x = remov_vlr_null.drop(cols_to_drop, axis=1)

x = pd.get_dummies(x, drop_first=True)

y = remov_vlr_null['purchase_value']
X = x

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

knn = KNeighborsClassifier(n_neighbors=5)
cv = KFold(n_splits=5, shuffle=True, random_state=42)
knn_cv_scores = cross_val_score(knn, X_scaled, y, cv=cv, scoring='accuracy')
print("Acurácia média KNeighborsClassifier:", knn_cv_scores.mean())

kmeans = KMeans(n_clusters=3, random_state=42)

kmeans.fit(X_scaled)
silhouette_avg = silhouette_score(X_scaled, kmeans.labels_)
print("Silhueta média KMeans:", silhouette_avg)

Acurácia média KNeighborsClassifier: 0.21500443441102957
Silhueta média KMeans: 0.00805593920848253
```

Otimização

```
import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

import numpy as np


teste_csv = pd.read_csv('car_rental_data.csv')

teste_df = pd.DataFrame(teste_csv)

remov_dupli = teste_df.drop_duplicates(keep="last")

remov_vlr_null = remov_dupli.dropna(thresh=2)


print("Colunas disponíveis:", remov_vlr_null.columns)


cols_to_drop = ['purchase_value', 'selling_value']

cols_to_drop = [col for col in cols_to_drop if col in remov_vlr_null.columns]

x = remov_vlr_null.drop(cols_to_drop, axis=1)


x = pd.get_dummies(x, drop_first=True)


scaler = StandardScaler()

x_scaled = scaler.fit_transform(x)


n_cluster = 3


kmeans = KMeans(n_clusters=n_cluster, random_state=42)

kmeans.fit(x_scaled)

remov_vlr_null['cluster'] = kmeans.labels_
```

```

print('Centro dos Clusters')

print(scaler.inverse_transform(kmeans.cluster_centers_))

plt.figure(figsize=(10,6))

for cluster in range (n_cluster):

    cluster_data = remov_vlr_null[remov_vlr_null['cluster'] == cluster]

    plt.scatter(cluster_data['purchase_value'],cluster_data['segment'],label=f'cluster
{cluster}')

plt.title ('Clusters de dados de segmento')

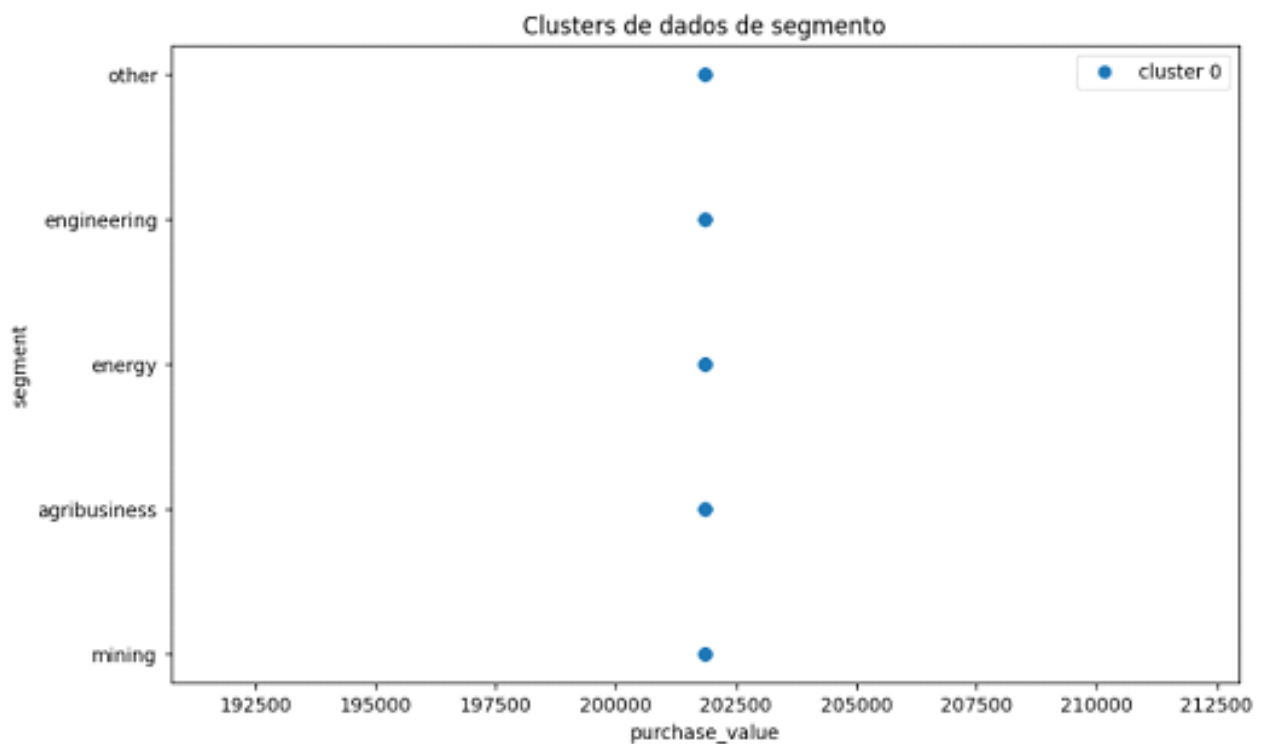
plt.xlabel('purchase_value')

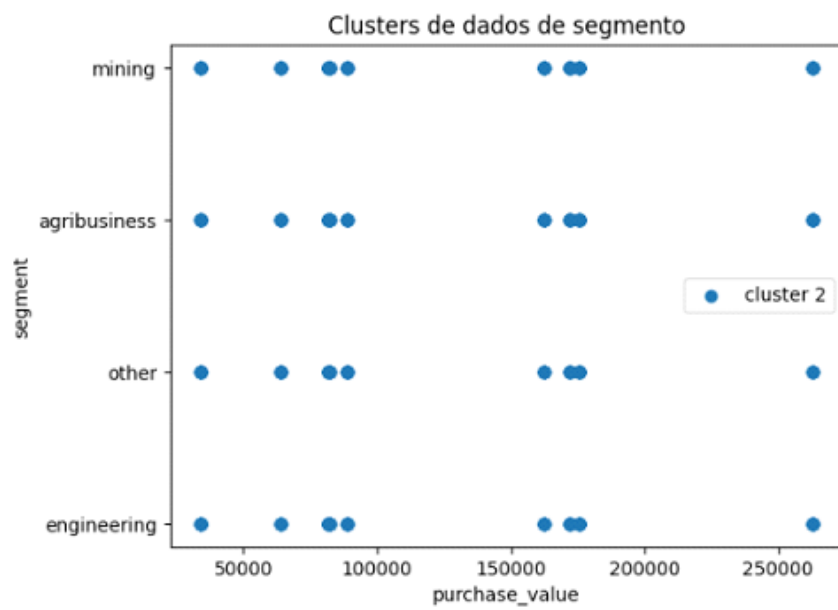
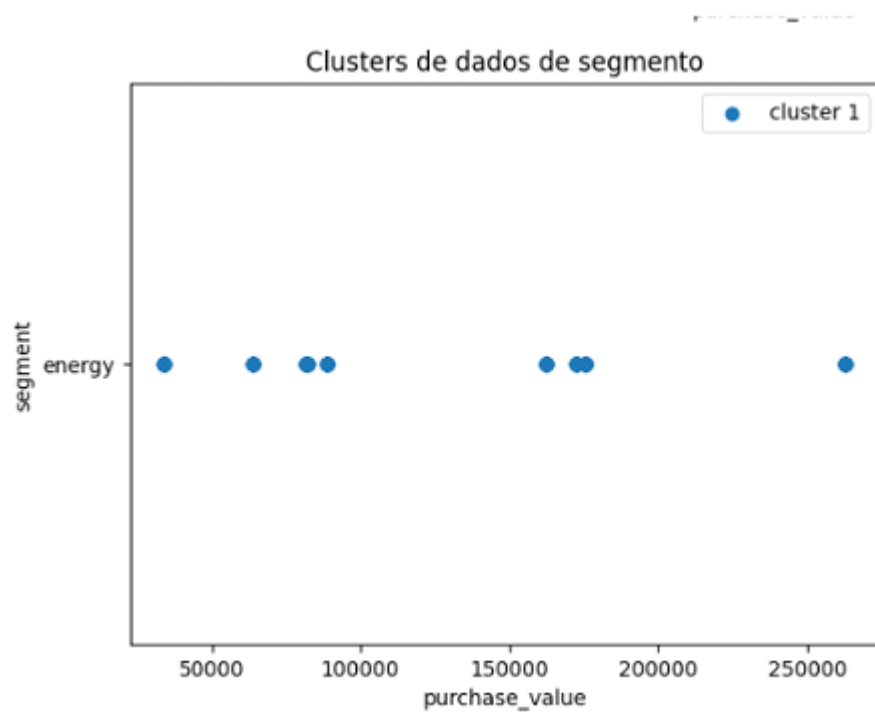
plt.ylabel('segment')

plt.legend()

plt.show()

```





Gateway e Dispositivos de Rede

- o Gateway/Roteador: 192.168.10.1**
- o Central de Comutação: 192.168.10.2**

Sala de exposição (192.168.10.10 - 192.168.10.19)

- o Computador de Vendas 1: 192.168.10.10**
- o Computador de Vendas 2: 192.168.10.11**
- o Impressora: 192.168.10.12**
- o Interruptor: 192.168.10.13**

Administração (192.168.10.20 - 192.168.10.29)

- o Computador 1: 192.168.10.20**
- o Computador 2: 192.168.10.21**
- o Computador 3: 192.168.10.22**
- o Impressora de Contratos: 192.168.10.23**
- o Interruptor: 192.168.10.24**

Atendimento ao Cliente (192.168.10.30 - 192.168.10.39)

- o Computador 1: 192.168.10.30**
- o Computador 2: 192.168.10.31**
- o Computador 3: 192.168.10.32**
- o Impressora: 192.168.10.33**
- o Interruptor: 192.168.10.34**

Oficina/Mecânica (192.168.10.40 - 192.168.10.49)

- o Computador de Serviço: 192.168.10.40**
- o Tablet de Diagnóstico: 192.168.10.41**
- o Interruptor: 192.168.10.42**

Estacionamento (192.168.10.50 - 192.168.10.59)

o Câmera de Segurança 1: 192.168.10.50

o Câmera de Segurança 2: 192.168.10.51

o Tablet para Inspeção: 192.168.10.52

o Interruptor: 192.168.10.53

