

**UNIVERSIDADE NOVE DE JULHO  
DIRETORIA DOS CURSOS DE INFORMÁTICA**

ADRIANA VILANOVA DOS SANTOS - RA 2222106311

GABRIELLA SILVA PASCOTO - RA 2222101861

MATHEUS TENORIO DA SILVA - RA 2222101468

MAURO GABRIEL CESAR PEREIRA - RA 2222100490

**PLANEJAMENTO EM INFORMÁTICA:  
LOCCAR**

**SÃO PAULO  
2024**

ADRIANA VILANOVA DOS SANTOS - RA 2222106311

GABRIELLA SILVA PASCOTO - RA 2222101861

MATHEUS TENORIO DA SILVA - RA 2222101468

MAURO GABRIEL CESAR PEREIRA - RA 2222100490

## **PLANEJAMENTO EM INFORMÁTICA: LOCCAR**

Trabalho apresentado à Universidade Nove de Julho, UNINOVE, em cumprimento parcial às exigências da disciplina de Planejamento em informática, sob orientação do Prof. **Felipe Santos de Jesus**.

**SÃO PAULO  
2024**

## SUMÁRIO

|  |    |
|--|----|
| 1. Escopo .....  | 4  |
| 2. Serviços Oferecidos .....                                   | 5  |
| 3. Estrutura Interna da Empresa.....                           | 6  |
| 3.1. Aprendizado de Máquina.....                               | 6  |
| 3.1.1. Exploração de Dados e Pré Processamento.....            | 6  |
| 3.1.2. Implementação de Modelos de Aprendizado de Máquina..... | 7  |
| 3.1.3. Otimização e Validação do Modelo .....                  | 9  |
| 3.2. Ciência de Dados.....                                     | 11 |
| 3.2.1 Análise Descritiva dos Dados.....                        | 12 |
| 3.2.2 Modelagem Estatística.....                               | 13 |
| 3.3. Modelagem de Dados.....                                   | 21 |
| 3.3.1 Modelagem Conceitual.....                                | 21 |
| 3.3.2 Modelagem Lógica e Normalização.....                     | 22 |
| 3.3.3 Dicionário de Dados e Simulação de Cadastro.....         | 23 |
| 3.4. Redes de Computadores.....                                | 27 |
| 3.4.1 Planta Baixa de Rede da Empresa.....                     | 27 |
| 3.4.2 Configuração de IP de Todos os Equipamentos.....         | 29 |
| 3.5. Segurança da Informação.....                              | 31 |
| 3.5.1 Análise de Riscos.....                                   | 31 |
| 3.5.2 Implementação de Medidas de Segurança.....               | 32 |
| 4. Conclusão.....  | 38 |
| 5. Referências.....  | 39 |

## **1. ESCOPO**

A empresa Loccar foi criada para facilitar a maneira de locação de carros para os usuários. Tendo como objetivo desenvolver um sistema para automatizar e otimizar o processo de aluguel. Simplificando o processo do cliente e o nosso.

Buscamos como uma das melhorias da nossa empresa o Registro de locações, incluindo detalhes do horário de início e término da locação, tipo do veículo desejado e a geração de contratos.

## 2. SERVIÇOS OFERECIDOS:

A empresa Loccar oferecerá os seguintes serviços:

- **Locação de Veículos:** O principal serviço prestado pela Loccar é a locação de veículos. Os clientes poderão alugar veículos rapidamente por períodos longos ou curtos, baseando-se nos valores de taxação impostos pela empresa.
- **Variedades de Veículos:** A empresa possui uma ampla variedade de veículos disponíveis para serem alugados, para atender todas as diferentes necessidades e preferências de nossos clientes.
- **Reservas Agendadas:** Oferecemos o serviço de reservas antecipadas, para que nossos clientes possam garantir a disponibilidade de um veículo em uma data de sua preferência.
- **Seguro:** A Loccar oferece seguro para casos de furtos, colisões, e proteção contra danos de terceiros.
- **Entrega e Retirada:** Um dos serviços oferecidos pela corporação é a retirada e entrega dos veículos em residências e locais específicos. Não sendo necessário a visita na agência física.
- **Locação Corporativa:** Um dos diferenciais é oferecer serviços para empresas, como tarifas corporativas, descontos corporativos, serviços específicos para atender a necessidade do cliente.

### 3. ESTRUTURA INTERNA DA EMPRESA

#### 3.1. Aprendizado de máquina

##### 3.1.1. Exploração de Dados e Pré-processamento

###### 3.1.1.1. Coleta de Dados relevantes para o negócio proposto pela empresa

```
import pandas as pd
dados = pd.read_csv('/kaggle/input/projetinho/submission.csv')
dados = pd.read_csv('/kaggle/input/projetinho/sample_4E0BhPN.csv')
```

###### 3.1.1.2. Limpeza e Pré-processamento dos Dados

###### Ausentes

```
import pandas as pd
import numpy as np
```

```
data = {
'carro': [Uno, , hb20, np.nan,],
'km': [18, 1000, np.nan ]
}
```

```
df = pd.DataFrame(data)
```

```
# Preencher valores ausentes com a média
media_carro = df['carro'].mean()
df['idade'] = df['carro'].fillna(media_carro)
```

```
# Preencher valores ausentes com a mediana
mediana_carro = df['carro'].median()
df['carro'] = df['carro'].fillna(mediana_salario)
```

```
print("DataFrame após preenchimento de valores ausentes:")
print(df)
```

###### Outliers

```
from scipy import stats
```

```
z_scores_km = np.abs(stats.zscore(df['km']))
```

```

limite = 3
outliers_km = df['km'][z_scores_km > limite]

# Substituir outliers pela mediana
mediana_km = df['km'].median()
df['km'] = df['km'].mask(z_scores_km > limite, mediana_km)

print("\nDataFrame após tratamento de outliers no km:")
print(df)

```

### **Inconsistentes**

```

# Verificar dados inconsistentes no km
limite_inferior_km = 18
limite_superior_km = 100
df = df[(df['km'] >= limite_inferior_km) & (df['km'] <= limite_superior_km)]

print("\nDataFrame após remoção de dados inconsistentes no km:")
print(df)

```

#### **3.1.1.3. Verificar a Matriz Confusão**

```

from sklearn.metrics import confusion_matrix, classification_report
import numpy as np

# Dados de exemplo
y_true = np.array([0, 0, 1, 1, 0, 1, 0, 0, 0, 1])
y_pred = np.array([1, 0, 0, 1, 0, 1, 1, 1, 1, 1])

# Matriz de confusão
conf_matrix = confusion_matrix(y_true, y_pred)
print("Matriz de Confusão:")
print(conf_matrix)

# Relatório de classificação
print("\nRelatório de Classificação:")
print(classification_report(y_true, y_pred))

```

### **3.1.2. Implementação de Modelos de Aprendizado de Máquina**

#### **3.1.2.1. Escolha de algoritmos de ML Adequados ao Problema**

```

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

```

```

iris = load_iris()
X = iris.data
y = iris.target

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

clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

# Fazer previsões no conjunto de teste
y_pred = clf.predict(X_test)

# Avaliar a precisão do modelo
accuracy = accuracy_score(y_test, y_pred)
print("natureza_dos_dados:", accuracy)

```

### 3.1.2.2. Implementação dos Modelos Escolhidos Utilizando Bibliotecas como Scikit-learn ou TensorFlow

```

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Carregar o conjunto de dados Iris
iris = load_iris()
X = iris.data
y = iris.target

# Dividir os dados em conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Inicializar e treinar o modelo Random Forest
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)

# Fazer previsões no conjunto de teste
y_pred = clf.predict(X_test)

# Avaliar o desempenho do modelo
accuracy = accuracy_score(y_test, y_pred)
print("Acurácia do modelo:", accuracy)

# Exibir o relatório de classificação
print("\nRelatório de Classificação:")
print(classification_report(y_test, y_pred, target_names=iris.target_names))

```



### **3.1.2.3. Avaliação da performance dos modelos com métricas apropriadas**

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

# Carregar o conjunto de dados Iris
iris = load_iris()
X = iris.data
y = iris.target

# Dividir os dados em conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Inicializar e treinar o modelo Random Forest
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)

# Fazer previsões no conjunto de teste
y_pred = clf.predict(X_test)

# Avaliar o desempenho do modelo
report = classification_report(y_test, y_pred, target_names=iris.target_names)

print("Relatório de Classificação:")
print(report)
```

### **3.1.3. Otimização e Validação do Modelo**

#### **3.1.3.1. Otimização dos Hiperparâmetros dos Modelos para Melhorar a Performance**

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

# Carregar o conjunto de dados Iris
iris = load_iris()
X = iris.data
y = iris.target

# Dividir os dados em conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

```

# Definir os hiperparâmetros para o Grid Search
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# Inicializar o classificador RandomForest
rf = RandomForestClassifier(random_state=42)

# Inicializar o Grid Search com validação cruzada
grid_search = GridSearchCV(rf, param_grid, cv=5, scoring='accuracy')

# Realizar o ajuste dos dados de treinamento
grid_search.fit(X_train, y_train)

# Melhores hiperparâmetros encontrados
best_params = grid_search.best_params_
print("Melhores hiperparâmetros encontrados:")
print(best_params)

```

### 3.1.3.2. Validação Cruzada para Verificar a Robustez do Modelo

```

from sklearn.datasets import load_iris
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestClassifier

# Carregar o conjunto de dados Iris
iris = load_iris()
X = iris.data
y = iris.target

# Inicializar o classificador RandomForest
rf = RandomForestClassifier(n_estimators=100, random_state=42)

# Realizar validação cruzada com k-fold (k=5)
cv_scores = cross_val_score(rf, X, y, cv=5)

# Exibir os scores de validação cruzada
print("Scores de validação cruzada:", cv_scores)

# Calcular a média dos scores de validação cruzada
mean_cv_score = cv_scores.mean()
print("Média dos scores de validação cruzada:", mean_cv_score)

```

### **3.1.3.3. Documentação do Processo de construção e Treinamento do Modelo**

Documentação do Processo de Construção e Treinamento do Modelo

Introdução:

Este documento fornece uma visão detalhada do processo de construção e treinamento do modelo de aprendizado de máquina para a tarefa específica. Descreve as etapas, parâmetros selecionados e os resultados obtidos durante o desenvolvimento do modelo.

Objetivo:

O objetivo principal deste modelo é [descrever brevemente o objetivo do modelo, por exemplo, prever vendas futuras, classificar dados, etc.].

Etapas do Processo:

1. Exploração de Dados e Pré-processamento:

Coleta de Dados:

Descreva as fontes de dados utilizadas.

Lista de variáveis/features incluídas.

Limpeza e Pré-processamento:

Identificação e tratamento de valores ausentes, outliers, etc.

Transformações aplicadas aos dados.

2. Implementação de Modelos de Aprendizado de Máquina:

Escolha de Algoritmos:

Justificativa para a escolha dos algoritmos utilizados.

Implementação:

Detalhes sobre como os modelos foram implementados.

Utilização de bibliotecas (por exemplo, Scikit-learn, TensorFlow).

3. Otimização e Validação do Modelo:

Otimização de Hiperparâmetros:

Descrição do processo de otimização.

Lista dos hiperparâmetros ajustados.

Validação Cruzada:

Detalhes sobre como a validação cruzada foi realizada.

Resultados obtidos.

Parâmetros do Modelo:

Lista completa de hiperparâmetros e seus valores finais após a otimização.

Outros parâmetros relevantes para o modelo.

Métricas de Avaliação:

Descrição das métricas utilizadas para avaliar o desempenho do modelo.

Resultados específicos obtidos para cada métrica.

## **3.2. Ciência de Dados**

### **3.2.1. Análise Descritiva dos Dados**

#### **3.2.1.1. Utilização de Técnicas Estatísticas Básicas para descrever os Dados**

```

import pandas as pd
import numpy as np

# Carregue o conjunto de dados
dados = pd.read_csv('caminho/para/seu/arquivo/car_rental_data.csv')

# Supondo que queremos calcular as estatísticas de uma coluna específica,
por exemplo, 'demand'
coluna_dados = dados['demand']

# Calculando a média
media = np.mean(coluna_dados)
print(f"Média: {media}")

# Calculando a mediana
mediana = np.median(coluna_dados)
print(f"Mediana: {mediana}")

# Calculando o desvio padrão
desvio_padrao = np.std(coluna_dados)
print(f"Desvio Padrão: {desvio_padrao}")

```

### 3.2.1.2. Visualização de Dados utilizando Bibliotecas como Matplotlib e Seaborn

```

import matplotlib.pyplot as plt

# Supondo que 'date' seja a coluna de datas e 'demand' a coluna de demanda
plt.figure(figsize=(10, 6))
plt.plot(dados['date'], dados['demand'])
plt.xlabel('Data')
plt.ylabel('Demanda')
plt.title('Demanda ao longo do tempo')
plt.xticks(rotation=45)
plt.show()

```

Gráfico de dispersão (Scatter plot):  
import seaborn as sns

```
# Supondo que 'temperature' seja uma coluna de temperatura que influencia a
demanda
plt.figure(figsize=(10, 6))
sns.scatterplot(x='temperature', y='demand', data=dados)
plt.xlabel('Temperatura')
plt.ylabel('Demanda')
plt.title('Demanda vs Temperatura')
plt.show()
```

### **3.2.1.3. Identificação de Padrões e tendências nos Dados**

#### **Identificações de Outliers:**

```
plt.figure(figsize=(10, 6))
sns.boxplot(y='demand', data=dados)
plt.title('Identificação de Outliers na Demanda')
plt.show()
```

#### **Análise de padrões sazonais:**

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='day_of_week', y='demand', data=dados)
plt.xlabel('Dia da Semana')
plt.ylabel('Demanda')
plt.title('Demanda por Dia da Semana')
plt.show()
```

### **3.2.2. Modelagem Estatística**

#### **3.2.2.1. Aplicação de técnicas estatísticas avançadas para modelagem dos dados**

```
import pandas as pd

import statsmodels.api as sm

from scipy import stats

# Carregar o conjunto de dados

dados = pd.read_csv('caminho/para/seu/arquivo/car_rental_data.csv')
```

```

# Regressão Linear

X = dados['temperature']

y = dados['demand']

X = sm.add_constant(X)

modelo = sm.OLS(y, X).fit()

print(modelo.summary())


grupos = [dados[dados['month'] == mes]['demand'] for mes in
dados['month'].unique()]

anova_resultado = stats.f_oneway(*grupos)

print('ANOVA F-statistic:', anova_resultado.statistic)

print('ANOVA p-value:', anova_resultado.pvalue)

```

### **3.2.2.2. Uso de ferramentas como regressão linear, classificação etc.**

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix


# Carregar o conjunto de dados
dados = pd.read_csv('caminho/para/seu/arquivo/car_rental_data.csv')


# Regressão Linear
X_lr = dados[['temperature']]

y_lr = dados['demand']


X_train_lr, X_test_lr, y_train_lr, y_test_lr = train_test_split(X_lr, y_lr,
test_size=0.2, random_state=42)

```

```

modelo_lr = LinearRegression()

modelo_lr.fit(X_train_lr, y_train_lr)

y_pred_lr = modelo_lr.predict(X_test_lr)

mse_lr = mean_squared_error(y_test_lr, y_pred_lr)

r2_lr = r2_score(y_test_lr, y_pred_lr)

print(f'Regressão Linear - MSE: {mse_lr}, R-squared: {r2_lr}')

# Classificação
dados['demand_category'] = pd.cut(dados['demand'], bins=[0, 100, 200, 300],
labels=['Baixa', 'Média', 'Alta'])

X_clf = dados[['temperature', 'humidity', 'windspeed']]

y_clf = dados['demand_category']

X_train_clf, X_test_clf, y_train_clf, y_test_clf = train_test_split(X_clf, y_clf,
test_size=0.2, random_state=42)

scaler = StandardScaler()

X_train_clf = scaler.fit_transform(X_train_clf)

X_test_clf = scaler.transform(X_test_clf)

modelo_rf = RandomForestClassifier(random_state=42)

modelo_rf.fit(X_train_clf, y_train_clf)

y_pred_clf = modelo_rf.predict(X_test_clf)

print(confusion_matrix(y_test_clf, y_pred_clf))

print(classification_report(y_test_clf, y_pred_clf))

```

### 3.2.2.3. Avaliação da adequação dos modelos estatísticos aos dados

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, confusion_matrix,
classification_report, roc_auc_score, roc_curve
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
import matplotlib.pyplot as plt

```

```

# Carregar o conjunto de dados
dados = pd.read_csv('caminho/para/seu/arquivo/car_rental_data.csv')

# Regressão Linear
X_lr = dados[['temperature']]
y_lr = dados['demand']

X_train_lr, X_test_lr, y_train_lr, y_test_lr = train_test_split(X_lr, y_lr,
test_size=0.2, random_state=42)

modelo_lr = LinearRegression()

modelo_lr.fit(X_train_lr, y_train_lr)

y_pred_lr = modelo_lr.predict(X_test_lr)

mse_lr = mean_squared_error(y_test_lr, y_pred_lr)
r2_lr = r2_score(y_test_lr, y_pred_lr)

print(f'Regressão Linear - MSE: {mse_lr}, R-squared: {r2_lr}')

# Classificação
dados['demand_category'] = pd.cut(dados['demand'], bins=[0, 100, 200, 300],
labels=['Baixa', 'Média', 'Alta'])

X_clf = dados[['temperature', 'humidity', 'windspeed']]

y_clf = dados['demand_category']

X_train_clf, X_test_clf, y_train_clf, y_test_clf = train_test_split(X_clf, y_clf,
test_size=0.2, random_state=42)

scaler = StandardScaler()

X_train_clf = scaler.fit_transform(X_train_clf)

X_test_clf = scaler.transform(X_test_clf)

modelo_rf = RandomForestClassifier(random_state=42)
modelo_rf.fit(X_train_clf, y_train_clf)

y_pred_clf = modelo_rf.predict(X_test_clf)

print(confusion_matrix(y_test_clf, y_pred_clf))

print(classification_report(y_test_clf, y_pred_clf))

if len(y_test_clf.unique()) == 2:
    y_prob_clf = modelo_rf.predict_proba(X_test_clf)[: , 1]

```



```

auc_roc = roc_auc_score(y_test_clf, y_prob_clf)

print(f'AUC-ROC: {auc_roc}')

fpr, tpr, _ = roc_curve(y_test_clf, y_prob_clf)
plt.figure(figsize=(10, 6))

plt.plot(fpr, tpr, marker='.')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Curva ROC')

plt.show()

```

### 3.2.2.4. Implementação de modelos preditivos utilizando Python

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import mean_squared_error, r2_score, confusion_matrix,
classification_report
import seaborn as sns
import matplotlib.pyplot as plt

# Carregar os dados
dados = pd.read_csv('caminho/para/seu/arquivo/car_rental_data.csv')

# Verificar os primeiros registros
print(dados.head())

# Preparação básica (ajuste conforme necessário)
dados = dados.dropna()

# Análise de correlação
sns.pairplot(dados[['temperature', 'humidity', 'windspeed', 'demand']])
plt.show()

# Calcular a correlação
correlation_matrix = dados.corr()
print(correlation_matrix)

# Regressão Linear
X = dados[['temperature', 'humidity', 'windspeed']]
y = dados['demand']

```

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
modelo_lr = LinearRegression()
modelo_lr.fit(X_train, y_train)
y_pred_lr = modelo_lr.predict(X_test)
mse_lr = mean_squared_error(y_test, y_pred_lr)
r2_lr = r2_score(y_test, y_pred_lr)
print(f'MSE (Linear Regression): {mse_lr}')
print(f'R² (Linear Regression): {r2_lr}')

# Classificação
dados['demand_category'] = pd.cut(dados['demand'], bins=[0, 100, 200, 300],
labels=['Baixa', 'Média', 'Alta'])
y_clf = dados['demand_category']
X_train_clf, X_test_clf, y_train_clf, y_test_clf = train_test_split(X, y_clf, test_size=0.2,
random_state=42)
modelo_rf = RandomForestClassifier(random_state=42)
modelo_rf.fit(X_train_clf, y_train_clf)
y_pred_clf = modelo_rf.predict(X_test_clf)
print(confusion_matrix(y_test_clf, y_pred_clf))
print(classification_report(y_test_clf, y_pred_clf))

```

### 3.2.2.5. Avaliação da performance dos modelos preditivos

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import mean_squared_error, r2_score, confusion_matrix,
classification_report, roc_auc_score, roc_curve
import matplotlib.pyplot as plt

# Carregar os dados
dados = pd.read_csv('caminho/para/seu/arquivo/car_rental_data.csv')

# Preparação básica
dados = dados.dropna()

# Dividir os dados em conjuntos de treino e teste
X = dados[['temperature', 'humidity', 'windspeed']]
y = dados['demand']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Regressão Linear
modelo_lr = LinearRegression()
modelo_lr.fit(X_train, y_train)
y_pred_lr = modelo_lr.predict(X_test)
mse_lr = mean_squared_error(y_test, y_pred_lr)

```

```

r2_lr = r2_score(y_test, y_pred_lr)
print(f'Mean Squared Error (MSE): {mse_lr}')
print(f'R-squared (R²): {r2_lr}')

# Classificação
dados['demand_category'] = pd.cut(dados['demand'], bins=[0, 100, 200, 300],
labels=['Baixa', 'Média', 'Alta'])
y_clf = dados['demand_category']
X_train_clf, X_test_clf, y_train_clf, y_test_clf = train_test_split(X, y_clf, test_size=0.2,
random_state=42)
modelo_rf = RandomForestClassifier(random_state=42)
modelo_rf.fit(X_train_clf, y_train_clf)
y_pred_clf = modelo_rf.predict(X_test_clf)
print(confusion_matrix(y_test_clf, y_pred_clf))
print(classification_report(y_test_clf, y_pred_clf))

# AUC-ROC (para problemas binários)
if len(y_test_clf.unique()) == 2:
    y_prob_clf = modelo_rf.predict_proba(X_test_clf)[: , 1]
    auc_roc = roc_auc_score(y_test_clf, y_prob_clf)
    print(f'AUC-ROC: {auc_roc}')
    fpr, tpr, _ = roc_curve(y_test_clf, y_prob_clf)
    plt.figure(figsize=(10, 6))
    plt.plot(fpr, tpr, marker='.')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Curva ROC')
    plt.show()

```

### 3.2.2.6. Comparação entre diferentes abordagens de análise preditiva

#### Redes Neurais X Forest

```

from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier

# Exemplo de Rede Neural
mlp_model = MLPClassifier(hidden_layer_sizes=(100,), max_iter=1000)
mlp_model.fit(X_train, y_train)
mlp_predictions = mlp_model.predict(X_test)
mlp_accuracy = accuracy_score(y_test, mlp_predictions)

# Exemplo de Random Forest
rf_model = RandomForestClassifier(n_estimators=100)
rf_model.fit(X_train, y_train)
rf_predictions = rf_model.predict(X_test)
rf_accuracy = accuracy_score(y_test, rf_predictions)

```

```
print("Rede Neural Accuracy:", mlp_accuracy)
print("Random Forest Accuracy:", rf_accuracy)
```

### **Métodos Ensemble X Modelos Individuais**

```
from sklearn.ensemble import VotingClassifier
from sklearn.model_selection import cross_val_score

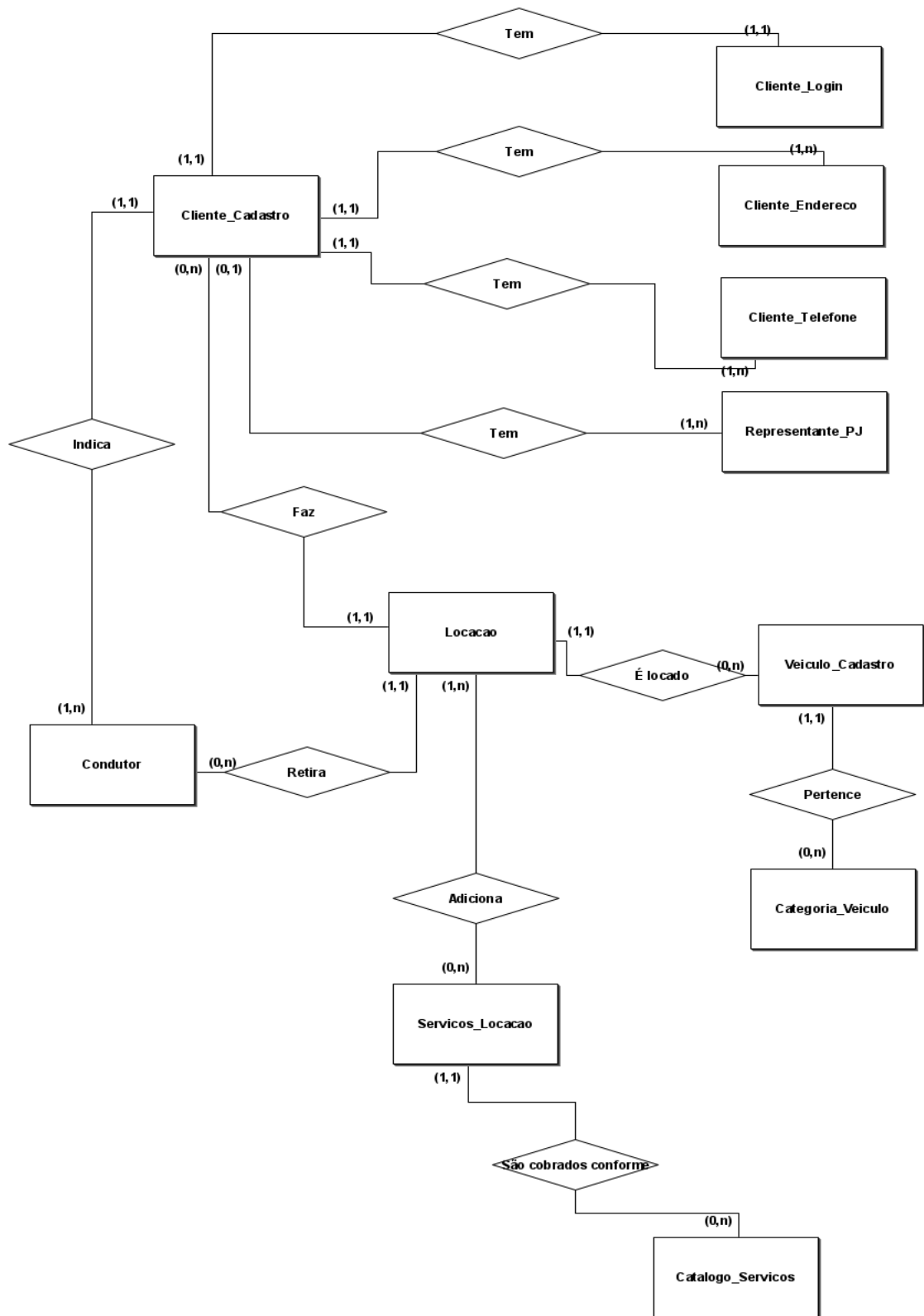
# Exemplo de Modelo Ensemble (Votação)
voting_model = VotingClassifier(estimators=[('lr', lr_model), ('dt', dt_model), ('svm',
svm_model)], voting='hard')
ensemble_scores = cross_val_score(voting_model, X_train, y_train, cv=5,
scoring='accuracy')

# Exemplo de Modelo Individual
individual_scores = cross_val_score(svm_model, X_train, y_train, cv=5,
scoring='accuracy')

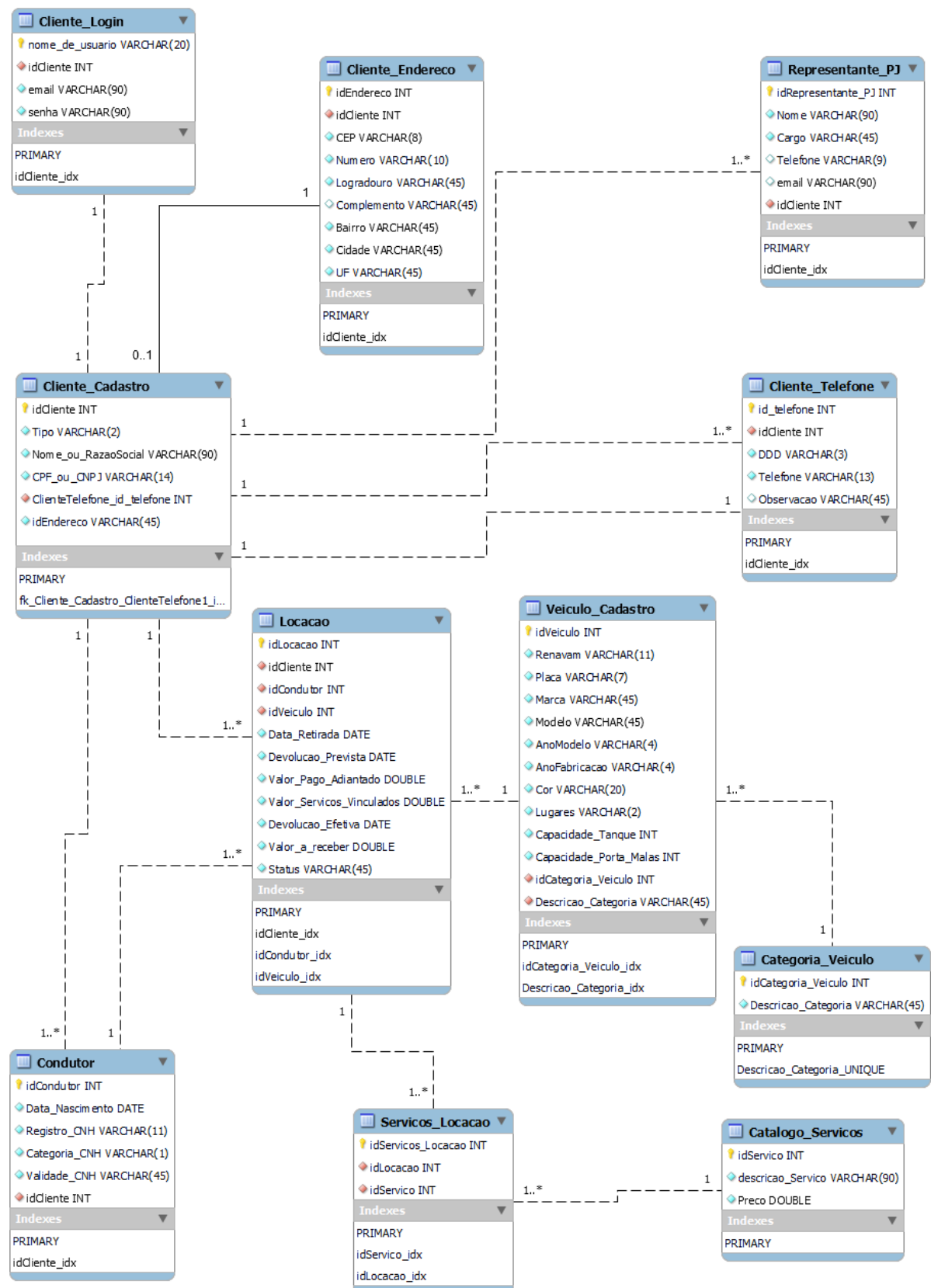
print("Ensemble Mean Accuracy:", ensemble_scores.mean())
print("Individual SVM Mean Accuracy:", individual_scores.mean())
```

### 3.3. Modelagem de Dados

#### 3.3.1. Modelagem Conceitual



### 3.3.2. Modelagem Lógica e Normalização



### 3.3.3. Entregar um dicionário de Dados

| Tabela Cliente_Cadastro |             |  |             |
|-------------------------|-------------|--|-------------|
| Atributo                | Tipo        | Descrição  | Chave       |
| idCliente               | int         | Identificador único de cliente                                 | Primária    |
| Tipo                    | varchar(90) | Detalha se cliente é pessoa física(PF) ou pessoa jurídica (PJ) | -           |
| Nome_ou_RazaoSocial     | varchar(90) | Nome do cliente  | -           |
| CPF_ou_CNPJ             | varchar(14) | Armazena CPF ou CNPJ do cliente                                | -           |
| id_telefone             | int         | Chave estrangeira para id_telefone                             | Estrangeira |
| idEndereco              | int         | Chave estrangeira para idEndereco                              | Estrangeira |

| Tabela Cliente_Login |             |  |             |
|----------------------|-------------|--|-------------|
| Atributo             | Tipo        | Descrição                                  | Chave       |
| nome_de_Usuario      | varchar(20) | Identificador único do login de um cliente | Primária    |
| idCliente            | int         | Chave estrangeira para idCliente           | Estrangeira |
| email                | varchar(90) | armazena o email do cliente                | -           |
| senha                | varchar(90) | armazena a senha do cliente                | -           |

| Tabela Cliente_Endereco |             |   |             |
|-------------------------|-------------|---|-------------|
| Atributo                | Tipo        | Descrição   | Chave       |
| idEndereco              | int         | Identificador único do endereço de um cliente     | Primária    |
| idCliente               | int         | Chave estrangeira para idCliente                  | Estrangeira |
| CEP                     | varchar(8)  | Armazena o cep do endereço do cliente             | -           |
| Numero                  | varchar(10) | Armazena o número do endereço do cliente          | -           |
| Logradouro              | varchar(45) | Armazena o logradouro do endereço do cliente      | -           |
| Complemento             | varchar(45) | Armazena o complemento do endereço do cliente     | -           |
| Bairro                  | varchar(45) | Armazena o bairro do endereço do cliente          | -           |
| Cidade                  | varchar(45) | Armazena a cidade do endereço do cliente          | -           |
| UF                      | varchar(2)  | Armazena a sigla do estado do endereço do cliente | -           |

| Tabela Representante_PJ |             |  |             |
|-------------------------|-------------|--|-------------|
| Atributo                | Tipo        | Descrição  | Chave       |
| idRepresentante         | int         | Identificador único de um representante de cliente pessoa jurídica | Primária    |
| Nome                    | varchar(90) | Armazena o nome do representante do cliente, caso seja PJ          | -           |
| Cargo                   | varchar(45) | Armazena o cargo do representante do cliente, caso seja PJ         | -           |
| Telefone                | varchar(9)  | Armazena o telefone pessoal do representante do cliente            | -           |
| email                   | varchar(45) | Armazena o email pessoal do representante do cliente               | -           |
| idCliente               | int         | Chave estrangeira para idCliente                                   | Estrangeira |

| Tabela Catalogo_Servicos |             |  |          |
|--------------------------|-------------|--|----------|
| Atributo                 | Tipo        | Descrição                                | Chave    |
| idServico                | int         | Identificador único de serviço oferecido | Primária |
| descricao_Servico        | varchar(90) | Descreve o serviço oferecido             | -        |
| Preco                    | double      | Preço cobrado pelo serviço               | -        |

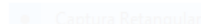
| Tabela Servicos_Locacao |      |  |             |
|-------------------------|------|--|-------------|
| Atributo                | Tipo | Descrição  | Chave       |
| idServicos_Locacao      | int  | Identificador único dos serviços usados em uma locação | Primária    |
| idLocacao               | int  | Chave estrangeira para idLocacao                       | Estrangeira |
| idServico               | int  | Chave estrangeira para idServico                       | Estrangeira |

| Tabela Condutor |             |  |             |
|-----------------|-------------|--|-------------|
| Atributo        | Tipo        | Descrição                                    | Chave       |
| idCondutor      | int         | Identificador único de condutor              | Primária    |
| Data_Nascimento | date        | Data de nascimento do condutor               | -           |
| Registro_CNH    | varchar(11) | Número de registro da CNH do condutor        | -           |
| Categoria_CNH   | varchar(1)  | Indica a categoria de CNH                    | -           |
| Validade_CNH    | date        | Indica a data de validade da CNH do condutor | -           |
| idCliente       | int         | Chave estrangeira para idCliente             | Estrangeira |



## Registro nas tabelas do banco de dados:

| Tabela Cliente_Cadastro |      |                             |                |                 |                |
|-------------------------|------|-----------------------------|----------------|-----------------|----------------|
| idCliente(PK)           | Tipo | Nome_ou_RazaoSocial         | CPF_ou_CNPJ    | id_telefone(FK) | idEndereco(FK) |
| 1                       | PF   | Adriana Vilanova dos Santos | 12312312323    | 1               | 1              |
| 2                       | PF   | Gabriella Silva Pascotto    | 34534534545    | 2               | 2              |
| 3                       | PF   | Matheus Tenorio da Silva    | 56756756756    | 3               | 3              |
| 4                       | PF   | Mauro Gabriel Cesar Pereira | 67867867878    | 4               | 4              |
| 5                       | PJ   | Salao Mais Bonita           | 80114323000191 | 5               | 5              |
| 6                       | PJ   | Maria Maria Advogada        | 31515828000163 | 5               | 5              |



| Tabela Cliente_Login |               |  |        |
|----------------------|---------------|--|--------|
| nomeUsuario(PK)      | idCliente(FK) | email  | senha  |
| driDrica             | 1             | <a href="mailto:dridrica@gmail.com">dridrica@gmail.com</a>                 | querty |
| gabiPascotto         | 2             | <a href="mailto:gabipascotto@hotmail.com">gabipascotto@hotmail.com</a>     | asdf   |
| matheusTenorio       | 3             | <a href="mailto:matheustenorio@yahoo.com">matheustenorio@yahoo.com</a>     | hijklç |
| mauroGabriel         | 4             | <a href="mailto:maurogabriel@gmail.com">maurogabriel@gmail.com</a>         | yuiop  |
| maisBonita           | 5             | <a href="mailto:salaomaisbonita@uol.com.br">salaomaisbonita@uol.com.br</a> | tyghbn |
| mariaMaria           | 6             | <a href="mailto:mariamariaadv@oab.sp.br">mariamariaadv@oab.sp.br</a>       | qazwsx |

| Tabela Cliente_Endereco |               |          |        |            |                |                 |           |    |
|-------------------------|---------------|----------|--------|------------|----------------|-----------------|-----------|----|
| idEndereco(PK)          | idCliente(FK) | CEP      | Numero | Logradouro | Complemento    | Bairro          | Cidade    | UF |
| 1                       | 1             | 04846530 | 52     | Rua A      | casa 1         | Jardins         | São Paulo | SP |
| 2                       | 2             | 04578963 | 23     | Rua B      | ap 23B         | Cerqueira César | São Paulo | SP |
| 3                       | 3             | 02245678 | 1489   | Rua C      |                | Ibirapuera      | São Paulo | SP |
| 4                       | 4             | 01453578 | 36     | Rua D      |                | Cidade Jardim   | São Paulo | SP |
| 5                       | 5             | 01545961 | 459    | Rua E      | torre 2 3andar | Centro          | Guarulhos | SP |
| 6                       | 6             | 02746385 | 18     | Rua F      |                | Morumbi         | São Paulo | SP |

| Tabela Representante_PJ |         |              |           |  |               |
|-------------------------|---------|--------------|-----------|--|---------------|
| idRepresentante(PK)     | Nome    | Cargo        | Telefone  | email  | idCliente(FK) |
| 1                       | Roberto | Gerente      | 956231741 | <a href="mailto:roberto@maisbonita.br">roberto@maisbonita.br</a> | 5             |
| 2                       | Maria   | Proprietária | 976301044 | <a href="mailto:mari@mariaadv.br">mari@mariaadv.br</a>           | 6             |

| Tabela Cliente_Telefone |               |     |             |                      |
|-------------------------|---------------|-----|-------------|----------------------|
| idTelefone(PK)          | idCliente(FK) | DDD | Telefone    | Observação           |
| 1                       | 1             | 11  | 999999999   |                      |
| 2                       | 2             | 11  | 988888888   |                      |
| 3                       | 3             | 11  | 977777777   |                      |
| 4                       | 4             | 11  | 966666666   |                      |
| 5                       | 5             | 11  | 50502310    |                      |
| 6                       | 6             | 11  | 08007452310 | ligar das 10h às 18h |

| Tabela Veiculo_Cadastro |              |         |         |            |        |        |       |         |        |            |           |                    |
|-------------------------|--------------|---------|---------|------------|--------|--------|-------|---------|--------|------------|-----------|--------------------|
| idVeic(PK)              | Renavam      | Placa   | Marca   | Modelo     | AnoMod | AnoFab | Cor   | Lugares | Tanque | PortaMalas | idCat(FK) | DescrCategoria(FK) |
| 1                       | 1234567890   | CQ02310 | Fiat    | Mobi       | 2024   | 2024   | Preto | 5       | 45     | pequeno    | 1         | Promoção           |
| 2                       | 99887766147  | GTY1289 | Hyundai | HB20       | 2024   | 2023   | Cinza | 5       | 45     | pequeno    | 2         | Tradicional        |
| 3                       | 887744556611 | HJK1236 | Honda   | Civic      | 2023   | 2023   | Preto | 5       | 80     | grande     | 3         | Superior           |
| 4                       | 23238596321  | QAZ9678 | Porshe  | 718Boxster | 2024   | 2024   | Azul  | 2       | 80     | pequeno    | 4         | Luxo               |

| Tabela Categoria_Veiculo |                     |
|--------------------------|---------------------|
| idCategoria_Veiculo(PK)  | Descricao_Categoria |
| 1                        | Promoção            |
| 2                        | Tradicional         |
| 3                        | Superior            |
| 4                        | Luxo                |

| Tabela Locacao |               |                |               |            |             |          |            |          |            |  |
|----------------|---------------|----------------|---------------|------------|-------------|----------|------------|----------|------------|--|
| idLoc(PK)      | idCliente(FK) | idCondutor(FK) | idVeiculo(FK) | Retirada   | DevPrevista | VIAdiant | VIServicos | AReceber | Status     |  |
| 1              | 5             | 5              | 1             | 01/05/2024 | 02/05/2024  | 100      | 50         | 50       | Finalizada |  |
| 2              | 6             | 6              | 2             | 18/05/2024 | 20/05/2024  | 400      | 140        | 140      | Finalizada |  |
| 3              | 2             | 2              | 3             | 29/05/2024 | 03/06/2024  | 1800     |            |          | Aberta     |  |

| Tabela Catalogo_Servicos |                    |       |
|--------------------------|--------------------|-------|
| idServico(PK)            | descricao_Servico  | Preco |
| 1                        | diária promoção    | 100   |
| 2                        | diária tradicional | 200   |
| 3                        | diária superior    | 300   |
| 4                        | diária             | 4900  |
| 5                        | leva e traz        | 50    |
| 6                        | taxa de limpeza    | 40    |

| Tabela Servicos_Locacao |               |               |
|-------------------------|---------------|---------------|
| idServicos_Locacao(PK)  | idLocacao(FK) | idServico(FK) |
| 1                       | 1             | 1             |
| 2                       | 1             | 2             |
| 3                       | 2             | 1             |
| 4                       | 2             | 6             |
| 5                       | 3             | 3             |

| Tabela Condutor |                 |              |               |              |               |
|-----------------|-----------------|--------------|---------------|--------------|---------------|
| idCondutor(PK)  | Data_Nascimento | Registro_CNH | Categoria_CNH | Validade_CNH | idCliente(FK) |
| 1               | 25/05/2001      | 12312312389  | B             | 31/12/2024   | 1             |
| 2               | 18/04/1993      | 45645645641  | B             | 05/06/2028   | 2             |
| 3               | 27/08/1980      | 36936985222  | AB            | 01/08/2024   | 3             |

### 3.4. Redes de Computadores

#### 3.4.1. Montar a planta baixa de Rede da Empresa

##### 3.4.1.1. Definir os Departamentos

Nossos departamentos foram divididos em 5, sendo eles **Atendimento**, **Administração**, **Financeiro**, **Recursos Humanos** e **T.I.**

Abaixo podemos ver na planta como os departamentos estão distribuídos

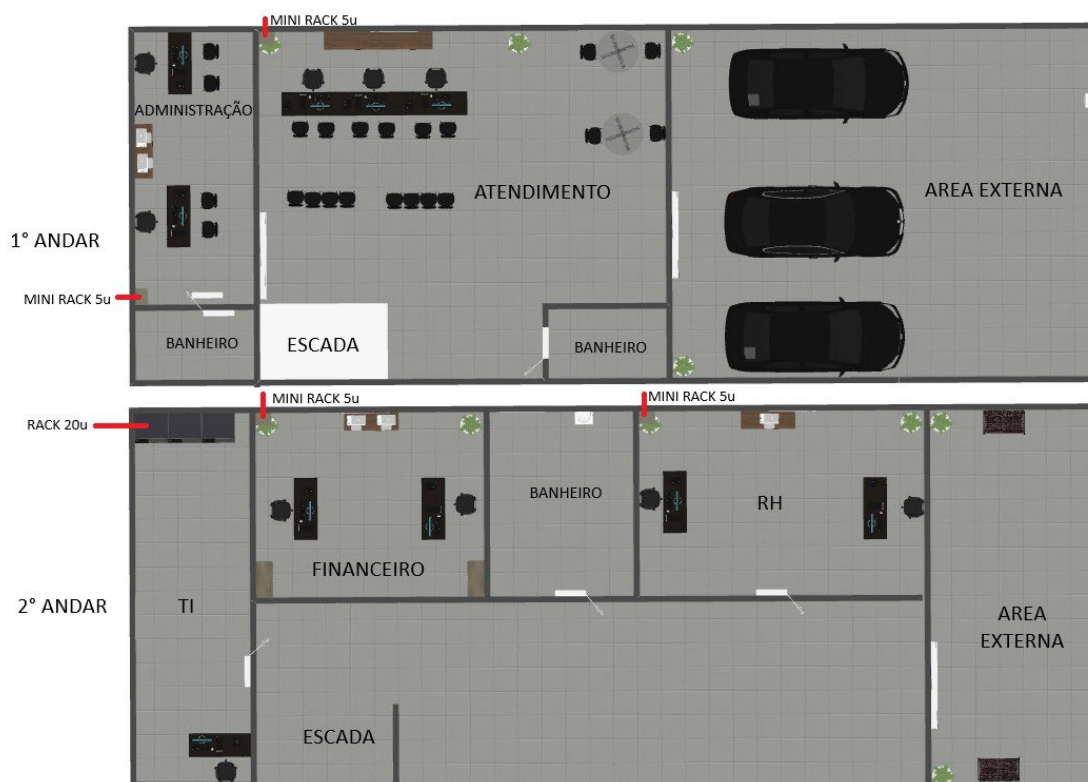
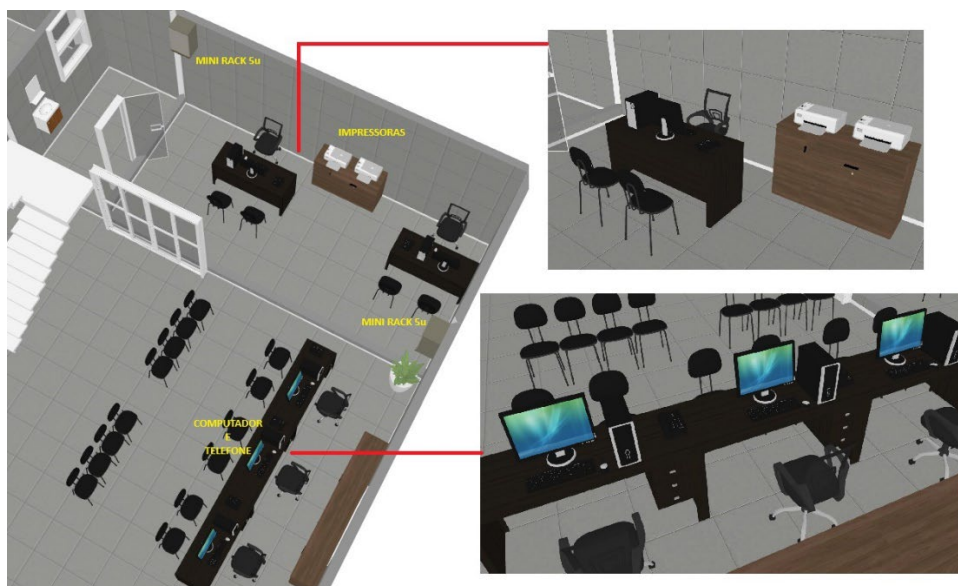


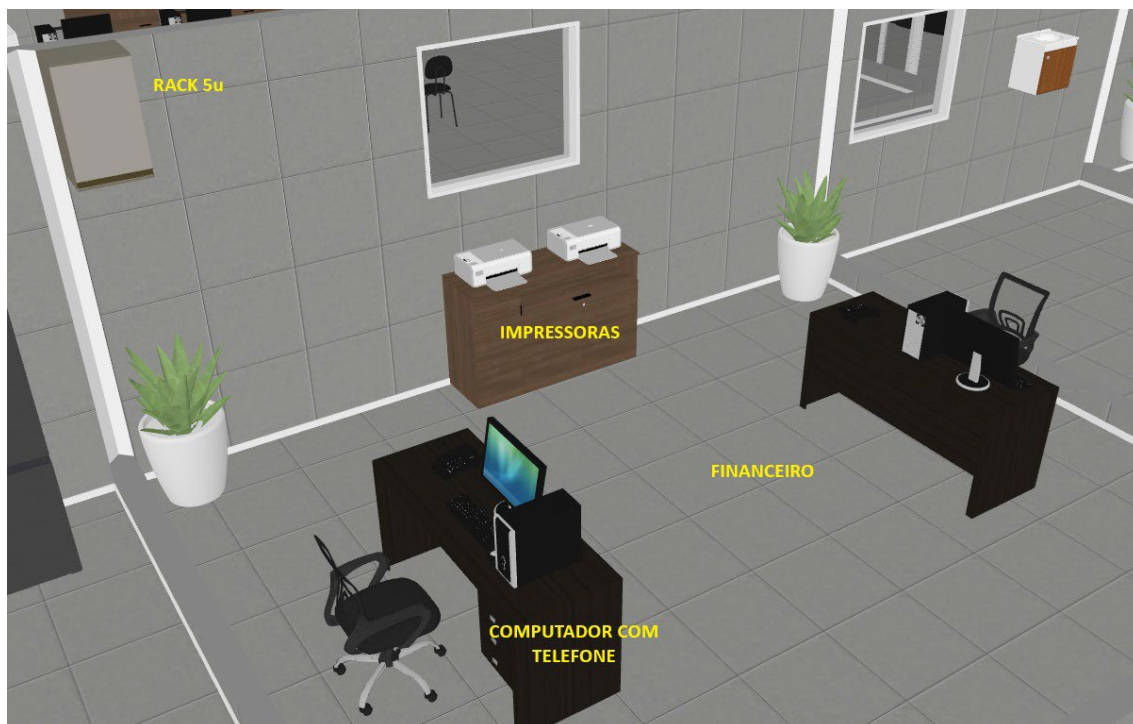
Imagem da planta da empresa.



Imagens mais detalhadas da planta identificando os dispositivos.



Imagem detalhada do setor do RH.



Imagens da parte interna do setor financeiro.

### 3.4.1.2. Definir os Equipamentos que serão Utilizados em cada Departamento

Dentro de cada departamento foram definidos os equipamentos utilizados:

- **TI:** Servidor, 1 Computador, 1 Impressora, 1 telefone.
- **Administração:** Servidor, 2 Computadores, 2 Impressoras, 2 telefones.
- **Recursos Humanos:** Servidor, 2 Computadores, 2 Impressoras, 2 telefones.
- **Financeiro:** Servidor, 2 Computadores, 2 Impressoras, 2 telefones.
- **Atendimento:** Servidor, 3 Computadores, 1 Impressora, 3 telefones.

### 3.4.2. Configuração de IP de todos os equipamentos

#### 3.4.2.1. Definir a Classe de Rede

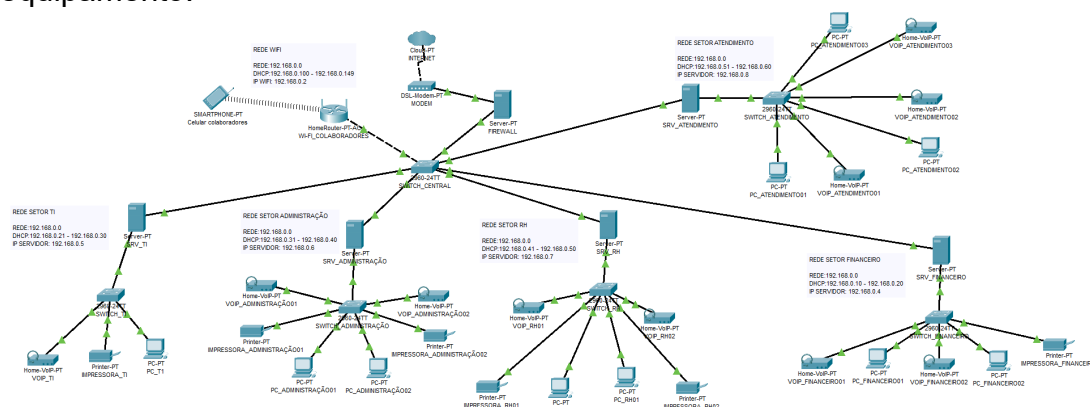
Todos os departamentos tem acesso direto a internet cabeada e Wireless. Para configurar os equipamentos, foi utilizado a classe C para definição de classe de Rede. Classe de Rede C: **192.168.0.0**.

### **3.4.2.2. Definir o Padrão de Rede de cada Departamento**

- **TI**
  - Rede: 192.168.0. - 10
  - Máscara: 255.255.255.0
  - DHCP: 192.168.0.21 - 192.168.0.30
  - IP Servidor: 192.168.0.5
  
- **ADMINISTRAÇÃO**
  - Rede: 192.168.0.11 - 20
  - Máscara: 255.255.255.0
  - DHCP: 192.168.0.31 - 192.168.0.40
  - IP Servidor: 192.168.0.6
  
- **RECURSOS HUMANOS**
  - Rede: 192.168.0.21 - 30
  - Máscara: 255.255.255.0
  - DHCP: 192.168.0.41 - 192.168.0.50
  - IP Servidor: 192.168.0.7
  
- **FINANCEIRO**
  - Rede: 192.168.0.31 - 39
  - Máscara: 255.255.255.0
  - DHCP: 192.168.0.10 - 192.168.0.20
  - IP Servidor: 192.168.0.4
  
- **ATENDIMENTO**
  - Rede: 192.168.0.40 - 47
  - Máscara: 255.255.255.0
  - DHCP: 192.168.0.51 - 192.168.0.60
  - IP Servidor: 192.168.0.8
  
- **REDE WIFI**
  - Rede: 192.168.0.0
  - Máscara: 255.255.255.0
  - DHCP: 192.168.0.100 - 192.168.0.149
  - IP WiFi: 192.168.0.2



Na imagem abaixo, podemos observar como foi feita a distribuição dos IPs de cada equipamento:



Para melhor visualização da imagem, disponibilizamos um arquivo em nuvem:

[https://drive.google.com/drive/folders/1b9Xx8Xr\\_JwlzmgnHxUymkR83fe9V6nK?usp=drive\\_link](https://drive.google.com/drive/folders/1b9Xx8Xr_JwlzmgnHxUymkR83fe9V6nK?usp=drive_link)

## 3.5. Segurança da Informação

### 3.5.1. Análise de Riscos

#### 3.5.1.1. Identificação e Avaliação dos Riscos de Segurança para a Empresa

**Inventário** → Ativos de informação da organização, incluindo hardware, software, dados e pessoas.

**Valor** → Valor de cada ativo, considerando o impacto potencial da sua perda ou comprometimento.

**Riscos** → Priorizar os riscos com base em sua gravidade. Avaliando a probabilidade de ocorrência de cada ameaça, (alta, média, baixa).

### **3.5.1.2. Análise de Vulnerabilidades e Ameaças Potenciais**

- Ameaças → Erros humanos, falhas de sistema.
- Tipos de Ameaças → Hackers, concorrentes, Malware
- Fraquezas → Identificar fraquezas nos sistemas e processos que podem ser exploradas por ameaças.
- Efetuar simulações de ataques para identificar vulnerabilidades não descobertas.

### **3.5.2. Implementação de Medidas de Segurança**

#### **3.5.2.1. Implementação de Políticas de Controle de Acesso aos Sistemas e Dados**

Permissões → Administrador, gerente, usuário, diretoria.

Controle de Acesso → Utilizar sistemas de Gestão de Identidade e Acesso ( I. A )

Identificar → classificar os sistemas com, aplicativos e dados.

Plano de Emergência → Definir procedimentos para responder a acessos não autorizados e estabelecer equipes de resposta.

#### **Configuração de sistemas de detecção de intrusão e prevenção de ataques:**

Instalação de Sistemas → **Configuração do sistema, incluindo definição de parâmetros básicos e integração com a infraestrutura existente.**

IDS/IPS → **Decida entre um IDS/IPS baseado em rede (NIDS/NIPS) e um baseado em host (HIDS/HIPS)**

Infraestrutura → Infraestrutura de rede para determinar onde colocar os sensores

Segurança → Políticas claras sobre o que constitui atividade suspeita e quais ações devem ser tomadas em resposta

Logs → logs para identificar padrões e investigar incidentes.



Manutenção e Atualizações → Mantenha as assinaturas e regras do IDS/IPS atualizadas para se proteger contra novas ameaças, configurações do sistema para adaptá-lo a mudanças na infraestrutura e nas ameaças

Revisões Semanal → Realize revisões do sistema e das políticas para garantir que continuam eficazes e atualizadas.

### **Riscos Identificados:**

- Ameaça: Vulnerabilidade de Softwares
  - Impacto: Alto
  - Probabilidade: Alto
- Ameaça: Malware
  - Impacto: Alto
  - Probabilidade: Médio
- Ameaça: Ataque DDOS
  - Impacto: Médio
  - Probabilidade: Alto
- Ameaça: Ameaças Internas
  - Impacto: Médio
  - Probabilidade: Baixo
- Ameaça: Senhas
  - Impacto: Alto
  - Probabilidade: Alto
- Ameaça: Dados Vasados
  - Impacto: Alto
  - Probabilidade: Alto
- Ameaça: SQL
  - Impacto: Alto
  - Probabilidade: Médio
- Ameaça: Ataques
  - Impacto: Médio
  - Probabilidade: Médio
- Ameaça: Infiltração de Dados
  - Impacto: Alto
  - Probabilidade: Alto
- Ameaça: Cavalo de Troia
  - Impacto: Médio
  - Probabilidade: Médio

- Ameaça: Ataques via Bluetooth
  - Impacto: Baixo
  - Probabilidade: Baixo
- Ameaça: Ataques via Wi-fi
  - Impacto: Baixo
  - Probabilidade: Médio
- Ameaça: Serviços Nuvem
  - Impacto: Alto
  - Probabilidade: Alto
- Ameaça: Patches Incompletos
  - Impacto: Baixo
  - Probabilidade: Baixo
- Ameaça: Segurança Fraca
  - Impacto: Alto
  - Probabilidade: Alto
- Ameaça: Protocolos de Rede
  - Impacto: Baixo
  - Probabilidade: Baixo
- Ameaça: Script
  - Impacto: Baixo
  - Probabilidade: Baixo
- Ameaça: Ataques de Redirecionamento
  - Impacto: Médio
  - Probabilidade: Médio
- Ameaça: Configurações de Seguranças incompletas e incorretas
  - Impacto: Alto
  - Probabilidade: Alto
- Ameaça: Falsificação de Identidade
  - Impacto: Alto
  - Probabilidade: Alto

## Políticas de Acesso

- **Controle de acesso a sistemas:**
  - O acesso a sistemas de controle como sistemas de automação, para proteger infraestruturas críticas contra ameaças cibernéticas.
- **Revisão de acesso regular:**
  - Revisar e atualizar regularmente as permissões de acesso dos usuários
- **Dispositivos Móveis:**
  - Acesso seguro a recursos da organização por meio de dispositivos móveis, como Celulares e tablets.
- **Rede sem Fio:**
  - Seguranças específicas para redes sem fio, como autenticação WPA2/WPA3, controle de acesso e de rede
- **Privilégios:**
  - Conceder acesso mínimo para que os usuários realizem suas funções
- **Senhas:**
  - Estabelecer caracteres para senhas, incluindo comprimento mínimo, alteração regular de senhas
- **Tentativas de Acessos:**
  - Bloquear temporariamente após um número específico de tentativas de login.
- **Acesso somente horário de trabalho ou autorizado fora do expediente:**
  - Limitar o acesso a determinados recursos ou sistemas com base no horário do dia ou na carga horária de trabalho dos usuários.
- **Terceiros tentando acessar login e senha de outro funcionário:**
  - Estabelecer protocolos para conceder e revogar o acesso de fornecedores, parceiros e outras externas.
- **Documentos e Arquivos:**
  - Permitir acessos específicos sobre documentos e arquivos, incluindo compartilhamento seguro e edição.

### **3.5.2.2. Configuração de Sistemas de Detecção de Intrusão e Prevenção de Ataques**

#### **Instale e Configure sistemas para detectar e prevenir intrusões**

- Ajustes e aprimoramento
- Manutenção e Supervisão
- Treinamento.
- Testes e validações
- Configuração do sistema.
- Instalação do sistema.
- Escolha o sistema.

#### **Medidas de detecção e prevenção de ataques na empresa**

- **Análise de comportamento do usuário:**
  - Análise comportamental de usuários para identificar desvios de padrões normais de comportamento que possam indicar atividade irregular
- **Atualizações de segurança:**
  - Sistemas e softwares atualizados com os patches de segurança mais recentes para corrigir novos e velhos erros
- **Treinamento de segurança:**
  - Treinamento regular em segurança para funcionários, sobre práticas seguras de computação e como identificar possíveis ameaças.
- **Acesso a páginas Web:**
  - Filtros de conteúdo da web para bloquear o acesso a sites maliciosos conhecidos e downloads de arquivos perigosos.
- **Antivirus:**
  - Antivírus atualizados em todos os dispositivos da empresa para detectar e remover software
- **Rede:**
  - Divida a rede em segmentos menores e restrinja o tráfego entre eles, limitando assim o alcance de ataques indevidos
- **Acesso de usuarios:**
  - Monitorando as atividades dos usuários com acessos elevados para detectar qualquer comportamento diferente da politica da empresa.
- **Dados:**
  - Utilizando o sistema de criptografia para proteger dados confidenciais armazenados e transmitidos.

- **Trafego de dados:**
  - Criptografia para proteger a integridade das comunicações, especificamente em redes Wi-Fi, redes públicas ou não confiáveis.
- **Backup:**
  - Realizar diariamente backups de dados e testes, testando a capacidade de restauração para garantir contra-ataques de hackers e perda de dados.
- **Email:**
  - Criar filtro para os e-mails da empresa, assim podendo identificar e bloquear e-mails de malware e spam, gerando alerta para o T.I antes que alcancem as caixas de entrada dos usuários.

## 4. CONCLUSÃO

Com base nas nossas pesquisas, podemos afirmar que a nossa empresa possui uma grande variedades de se consolidar com as técnicas atribuídas durante a criação do projeto. Todas as matérias foram interligadas para a construção da Loccar.

O que podemos afirmar é que, extraímos o máximo de informações possíveis para o escopo e conclusão do que foi proposto a todos. Podemos afirmar que a Loccar possui tudo o que uma empresa deve ter para se firmar no grande mercado de locadoras de carros, batendo de frente com a concorrência.

O projeto em si nos deixou um grande aprendizado, pois ele nos ensinou a como integrar novas linguagens, novos temas dentro da área de T.I, e também a como extrair diversas informações com a mais grande quantidade de códigos variados. Agradecemos todos os professores pelos ensinamentos a qual nos foi passado.

Esperamos que com a finalização do trabalho, possamos amadurecer como profissionais dentro da área, e claro, como pessoas também. Agradecemos a Uninove pela oportunidade.

## 5. REFERÊNCIAS

COELHO, Beatriz. **Entenda como fazer a conclusão do TCC corretamente.** 2017  
Disponível em:

< <https://rockcontent.com/br/blog/referencia-de-site-abnt/> > acesso em: 13 de Mai.2024.

COELHO, Taysa. **7 sites para fazer planta baixa online grátis e fáceis de usar.**  
Disponível em:

< <https://www.appgeek.com.br/planta-baixa-online/> > acesso em: 09 de Abril.2024.

PASSARIN, Leonardo. **Resumo da ISO 27001 – Sistema de Gestão de Segurança da Informação.** 2021.

Disponível em:

< <https://www.estrategiaconcursos.com.br/blog/resumo-da-iso-27001/> > acesso em: 29 de Mar. 2024.

GuiaViajarMelhor. **Como funciona o aluguel de carros: 15 dúvidas comuns na hora de locar um veículo.** 2020.

Disponível em:

< <https://guiaviajarmelhor.com.br/como-funciona-o-aluguel-de-carros-15-duvidas-comuns-na-hora-de-locar-um-veiculo/> > acesso em: 27 de Fev. 2024.