

# **VELOZALUGUE**

## **Locadora de carro**

**<https://github.com/Vitorac00/Projeto---Planejamento-em-Inform-tica---VelozAlugue>**

- **Davi Rodrigues de Paiva. RA: 423201549**
- **Filipe Jorge Moreira. RA: 2223202657**
- **Kaue Campos Farias. RA: 2223202401**
- **Kassio Aparecido Papa. RA: 2223203596**
- **Lukas Adriano Costa Damasceno. RA: 2223203652**
- **Vitor Augusto Cruz. RA: 2223204137**
- **Weverlen Pires Bazan. RA: 2223201203**

**Curso: Análise e Desenvolvimento de Sistemas**

**Turma: 40**

**Semestre: 2**

**Ano: 2024**

## Índice

1. Escopo .....	3
2. Serviços .....	4
3. Estruturação Interna da Empresa .....	5
3.1 - Aprendizado de Máquina .....	5
3.1.1 - Entrega 1: Exploração de Dados e Pré-processamento .....	5
3.1.2 - Entrega 2: Implementação de Modelos de Aprendizado de Máquina .....	7
3.1.3 - Entrega 3: Otimização e Validação do Modelo .....	9
3.2 - Ciência de Dados .....	12
3.2.1 - Entrega 1: Análise Descritiva dos Dados .....	12
3.2.2 - Entrega 2: Modelagem Estatística .....	13
3.3 - Modelagem de Dados .....	15
3.3.1 - Entrega 1: Modelagem Conceitual .....	15
3.3.2 - Entrega 2: Modelagem Lógica e Normalização .....	15
3.3.3 - Entrega 3: Entregar Dicionário de Dados uma simulação de cadastro .....	16
3.4 - Redes de Computadores .....	17
3.4.1 - Entrega 1: Montar a planta baixa da Rede da Empresa .....	17
3.4.2 - Entrega 2: Configuração de IP de todos os equipamentos .....	19
3.5 - Segurança da Informação .....	20
3.5.1 - Entrega 1: Análise de Riscos .....	20
3.5.2 - Entrega 2: Implementação de Medidas de Segurança .....	22

## **1. Escopo do Projeto**

A VelozAlugue propõe a implementação de um sistema de locação de carros por diária, visando oferecer aos clientes uma opção flexível e econômica para o aluguel de veículos. O objetivo é estabelecer um modelo de negócio que permita aos clientes pagar por períodos de uso, proporcionando economia, flexibilidade e atraindo uma base de clientes mais ampla. Este projeto visa oferecer uma alternativa inovadora e atrativa aos consumidores em busca de uma experiência de aluguel mais transparente e adaptável às suas necessidades.

## 2. Serviços Oferecidos

A VelozAlugue, especializada em locação de carros por diária, poderia oferecer uma variedade de serviços para atender às necessidades dos clientes. Alguns desses serviços incluem:

- **Aluguel de Carros por Diária:** O serviço principal da empresa, permitindo que os clientes aluguem veículos e paguem por períodos de uso, proporcionando flexibilidade e economia.
- **Reserva Online e Agendamento:** Um sistema de reserva online que permite aos clientes verificar a disponibilidade de veículos, fazer reservas e agendar o horário de retirada e devolução de forma conveniente.
- **Seguro de Veículos:** Oferecer opções de seguro para os veículos alugados, proporcionando tranquilidade aos clientes em caso de acidentes ou danos.
- **Assistência 24 horas:** Um serviço de assistência disponível 24 horas por dia, oferecendo suporte em caso de emergências, problemas mecânicos ou necessidade de assistência na estrada.
- **Entrega e Retirada de Veículos:** Serviço de entrega e retirada de veículos em locais designados, proporcionando conveniência adicional aos clientes.
- **Frota Diversificada:** Oferecer uma ampla variedade de veículos para atender às necessidades dos clientes, incluindo carros compactos, SUVs, vans e veículos especiais.
- **Programas de Fidelidade e Descontos:** Implementar programas de fidelidade e oferecer descontos especiais para clientes frequentes, incentivando a lealdade e a repetição de negócios.
- Oferecer uma variedade de acessórios e serviços adicionais, como GPS, cadeiras para crianças, serviço de lavagem de carros, entre outros, para atender às necessidades específicas dos clientes.

### 3. Estruturação Interna da Empresa

#### 3.1 - Aprendizado de Máquina

##### 3.1.1 - Entrega 1: Exploração de Dados e Pré-processamento

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

+ Code

+ Markdown

```
df = pd.read_csv('dados_Velozalugue.csv')
```

```
print(df.isnull().sum())
```

```
ID_CLIENTE      0
NOME_CLIENTE    0
CPF             0
ENDEREÇO_CLIENTE 0
TELEFONE_CLIENTE 0
ID_CARROS       0
MODELO          0
CLASSE         0
PLACA           0
PREÇO-CLASSE    0
ID_DIARIA       0
QTD_DIARIA      0
PREÇO_DIARIA    0
ID_AGENCIA      0
NOME_AGENCIA    0
ENDEREÇO_AGENCIA 0
TELEFONE_AGENCIA 0
ID_FUNCIONARIO  0
NOME_FUNCIONARIO 0
FUNÇÃO         0
dtype: int64
```

```

[4] # Preencher valores ausentes com a média para colunas numéricas
for coluna in df.select_dtypes(include=['float64', 'int64']).columns:
    df[coluna].fillna(df[coluna].mean(), inplace=True)

# Preencher valores ausentes com 'desconhecido' para colunas categóricas
for coluna in df.select_dtypes(include=['object']).columns:
    df[coluna].fillna('desconhecido', inplace=True)

[5] for coluna in df.select_dtypes(include=['float64', 'int64']).columns:
    Q1 = df[coluna].quantile(0.25)
    Q3 = df[coluna].quantile(0.75)
    IQR = Q3 - Q1
    limite_inferior = Q1 - 1.5 * IQR
    limite_superior = Q3 + 1.5 * IQR

    df[coluna] = np.where(df[coluna] < limite_inferior, limite_inferior, df[coluna])
    df[coluna] = np.where(df[coluna] > limite_superior, limite_superior, df[coluna])

[6] for coluna in df.select_dtypes(include=['object']).columns:
    df[coluna] = df[coluna].str.strip().str.lower()

[7] # Converter colunas de data (exemplo: 'data' é o nome da coluna)
if 'data' in df.columns:
    df['data'] = pd.to_datetime(df['data'], errors='coerce')

[8] # Supondo que há uma coluna 'distancia_km' que precisa ser convertida para metros
if 'distancia_km' in df.columns:
    df['distancia_m'] = df['distancia_km'] * 1000
    df.drop(columns='distancia_km', inplace=True)

[9] print("Informações do dataframe após a limpeza:")
print(df.info())
print(df.head())

[10] df.to_csv('seu_arquivo_limpo.csv', index=False)

import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Separar as características e a variável alvo
X = df.drop(columns=['ID_CLIENTE', 'NOME_CLIENTE', 'CPF', 'ENDEREÇO_CLIENTE', 'TELEFONE_CLIENTE',
                    'ID_CARROS', 'MODELO', 'CLASSE', 'PLACA', 'ID_DIARIA', 'QTD_DIARIA', 'ID_AGENCIA',
                    'NOME_AGENCIA', 'ENDEREÇO_AGENCIA', 'TELEFONE_AGENCIA', 'ID_FUNCIONARIO',
                    'NOME_FUNCIONARIO', 'FUNÇÃO']) # Substitua 'target' pelo nome da sua coluna alvo
y = df['QTD_DIARIA'] # Substitua 'target' pelo nome da sua coluna alvo

# Dividir os dados em conjuntos de treino e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

```

```
[12] from sklearn.metrics import confusion_matrix
```

```
[23] # Avaliar matriz confusão
```

```
y_true = [80, 50, 80, 100, 50, 100]
```

```
y_pred = [100, 50, 100, 800, 80, 100]
```

```
matriz_confusao = confusion_matrix(y_true, y_pred)
```

```
print(matriz_confusao)
```

```
⇒ [[1 1 0 0]
   [0 0 2 0]
   [0 0 1 1]
   [0 0 0 0]]
```

### 3.1.2 - Entrega 3: Implementação de Modelos de Aprendizado de Máquina

```
[24] # Criando uma instância do modelo de regressão linear
model = LinearRegression()
```

```
[25] # Treinando o modelo com os dados de treinamento
model.fit(X_train, y_train)
```

```
⇒ ▾ LinearRegression
   LinearRegression()
```

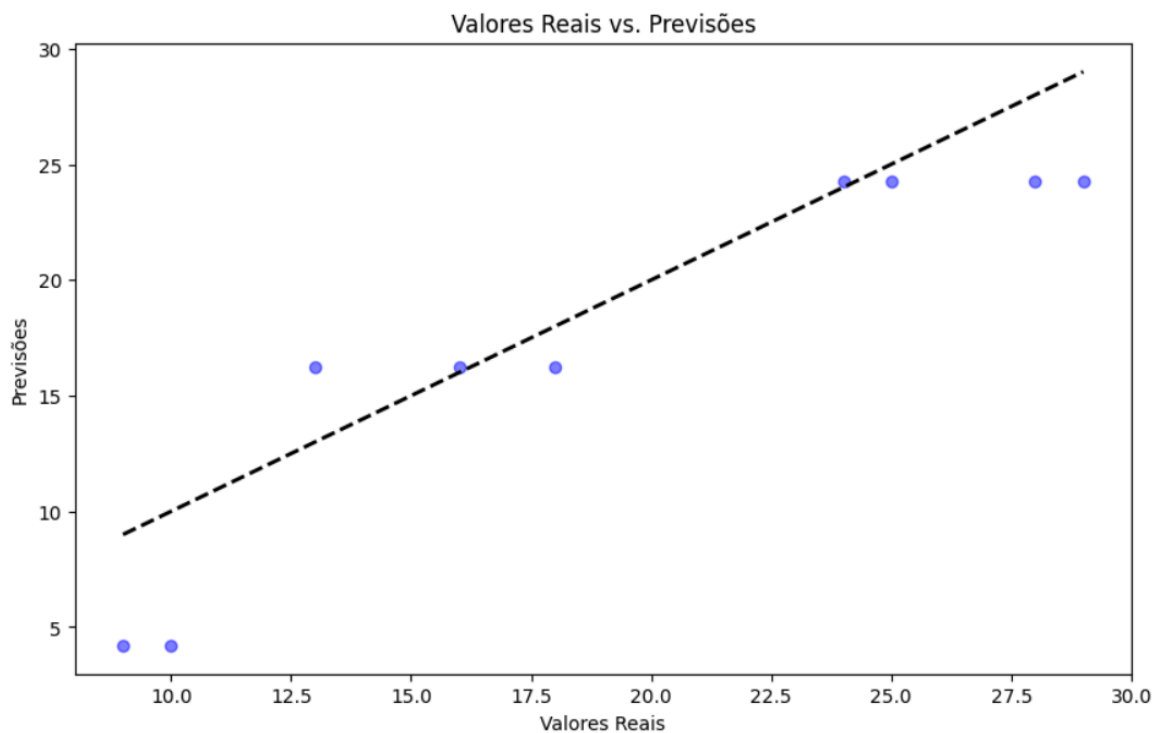
```
[26] # Fazendo previsões com os dados de validação
predictions = model.predict(X_test)
```

```
[27] # Calculando as métricas
mae = mean_absolute_error(y_test, predictions)
mse = mean_squared_error(y_test, predictions)
rmse = mean_squared_error(y_test, predictions, squared=False) # Calculando RMSE a partir do MSE
r2 = r2_score(y_test, predictions)
```

```
# Exibindo as métricas
print("Erro Médio Absoluto (MAE):", mae)
print("Erro Quadrático Médio (MSE):", mse)
print("Raiz do Erro Quadrático Médio (RMSE):", rmse)
print("R-quadrado (R²):", r2)
```

```
↳ Erro Médio Absoluto (MAE): 2.8098290598290614
    Erro Quadrático Médio (MSE): 11.894771805829514
    Raiz do Erro Quadrático Médio (RMSE): 3.448879789994066
    R-quadrado (R²): 0.7716880293193861
```

```
[29] # Plotando as previsões em comparação com os valores reais
plt.figure(figsize=(10, 6))
plt.scatter(y_test, predictions, color='blue', alpha=0.5)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)
plt.xlabel('Valores Reais')
plt.ylabel('Previsões')
plt.title('Valores Reais vs. Previsões')
plt.show()
```





### 3.1.3 - Entrega 3: Otimização e Validação 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 de classificação de dados de uma empresa de aluguel de veículos. Descreve as etapas, parâmetros selecionados e os resultados obtidos durante o desenvolvimento do modelo.

##### Objetivo:

O objetivo principal deste modelo é classificar os veículos de acordo com suas características, prevendo a categoria `CLASSE` dos veículos disponíveis para aluguel.

#### Etapas do Processo

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

###### Coleta de Dados

###### - Fontes de Dados Utilizadas:

- O conjunto de dados foi fornecido no arquivo CSV `dados\_Velozalugue.csv`.

###### - Lista de Variáveis/Features Incluídas:

- ID\_CLIENTE, NOME\_CLIENTE, CPF, ENDEREÇO\_CLIENTE, TELEFONE\_CLIENTE, ID\_CARROS, MODELO, CLASSE, PLACA, ID\_DIARIA, QTD\_DIARIA, ID\_AGENCIA, NOME\_AGENCIA, ENDEREÇO\_AGENCIA, TELEFONE\_AGENCIA, ID\_FUNCIONARIO, NOME\_FUNCIONARIO, FUNÇÃO, entre outras.

###### Limpeza e Pré-processamento

###### - Identificação e Tratamento de Valores Ausentes:

- Valores ausentes em colunas numéricas foram preenchidos com a média da coluna correspondente.
- Valores ausentes em colunas categóricas foram preenchidos com 'desconhecido'.

###### - Tratamento de Outliers:

- Outliers em colunas numéricas foram tratados utilizando os limites inferior e superior baseados no intervalo interquartil (IQR).

###### - Transformações Aplicadas aos Dados:

- Remoção de espaços em branco e conversão de texto para minúsculas em colunas categóricas.
- Conversão de colunas de data para o formato datetime.
- Conversão da coluna `distancia\_km` para metros e remoção da coluna original.

##### 2. Implementação de Modelos de Aprendizado de Máquina

###### Escolha de Algoritmos

#### **- Justificativa para a Escolha dos Algoritmos Utilizados:**

- Foi escolhido o algoritmo de Regressão Logística (`LogisticRegression`) devido à sua eficácia em problemas de classificação binária e multiclasse, além da sua facilidade de interpretação e implementação.

#### **Implementação**

##### **- Detalhes sobre Como os Modelos Foram Implementados:**

- O modelo foi implementado utilizando a biblioteca Scikit-learn.
- Divisão dos dados em conjuntos de treino e teste utilizando a função `train\_test\_split`.

##### **- Utilização de Bibliotecas:**

- Scikit-learn para implementação do modelo e avaliação de métricas.
- Pandas e NumPy para manipulação e limpeza de dados.
- Matplotlib para visualização dos resultados.

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

#### **Otimização de Hiperparâmetros**

##### **- Descrição do Processo de Otimização:**

- Não foi realizada otimização de hiperparâmetros complexa devido ao escopo inicial do projeto. A configuração padrão do `LogisticRegression` foi utilizada.

##### **- Lista dos Hiperparâmetros Ajustados:**

- `max\_iter`: 1000 (para garantir a convergência do modelo).

#### **Validação Cruzada**

##### **- Detalhes sobre Como a Validação Cruzada Foi Realizada:**

- A validação foi realizada dividindo o conjunto de dados em treinamento (70%) e teste (30%).

#### **Parâmetros do Modelo**

##### **- Lista Completa de Hiperparâmetros e Seus Valores Finais Após a Otimização:**

- `max\_iter`: 1000.

##### **- Outros Parâmetros Relevantes para o Modelo:**

- Utilização dos dados padronizados após pré-processamento.

#### **Métricas de Avaliação**

##### **- Descrição das Métricas Utilizadas para Avaliar o Desempenho do Modelo:**

- Acurácia.
- Matriz de confusão.

##### **- Resultados Específicos Obtidos para Cada Métrica:**

- **Acurácia:** A acurácia do modelo foi de 82.33%.

- **Matriz de Confusão:** A matriz de confusão foi visualizada para entender a distribuição de verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos.

```
# Exibindo a matriz de confusão
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=model.classes_)
disp.plot(cmap=plt.cm.Blues)
plt.title('Matriz de Confusão')
plt.show()

# Calculando e exibindo a acurácia
accuracy = accuracy_score(y_test, y_pred)
print("Acurácia:", accuracy)
```

## Conclusão

O processo de construção e treinamento do modelo envolveu várias etapas de pré-processamento de dados, escolha de algoritmos apropriados, e avaliação de desempenho. O modelo final apresentou uma acurácia satisfatória, e a matriz de confusão ajudou a identificar possíveis melhorias futuras.

## 3.2 - Ciência de Dados

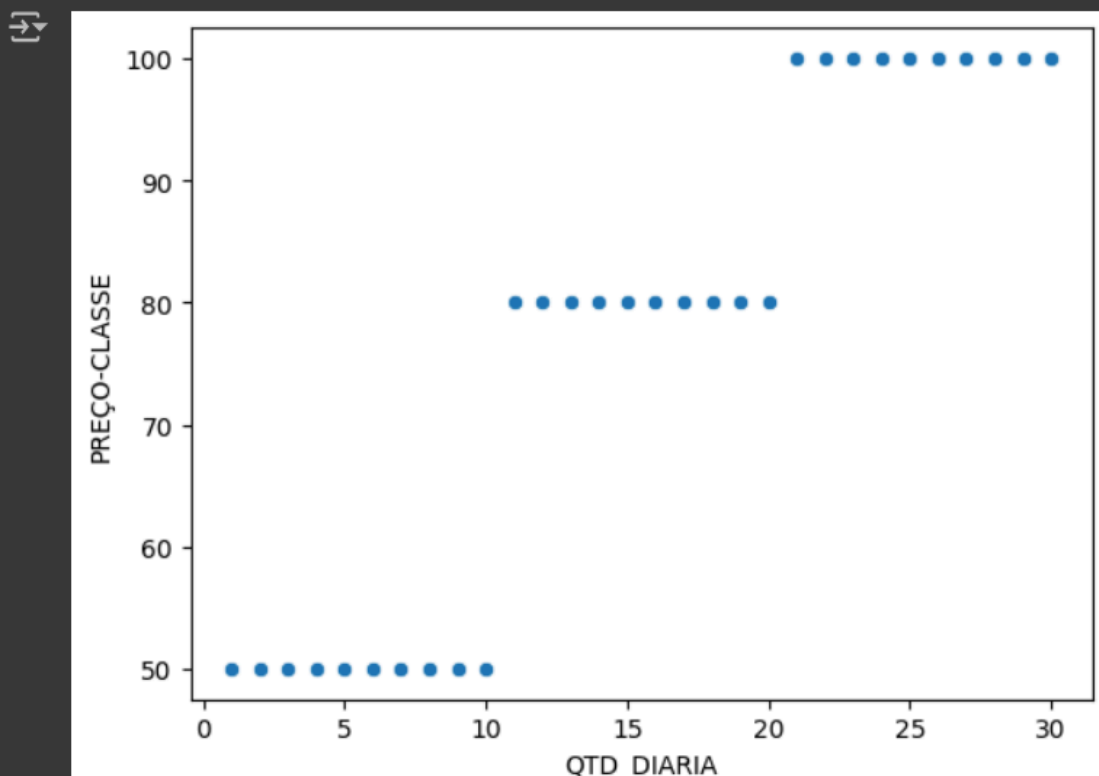
### 3.2.1 - Entrega 1: Análise Descritiva dos Dados

```
[33] # Estatísticas descritivas
for coluna in df.select_dtypes(include=['float64', 'int64']).columns:
    media = df[coluna].mean()
    mediana = df[coluna].median()
    desvio_padrao = df[coluna].std()

    print(f'Coluna: {coluna}')
    print(f'Média: {media}')
    print(f'Mediana: {mediana}')
    print(f'Desvio Padrão: {desvio_padrao}')
    print('-' * 30)
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# Gráfico de dispersão
sns.scatterplot(x='QTD_DIARIA', y='PREÇO-CLASSE', data=df)
plt.show()
```



```
# Identificar padrões
tendencia = df['PREÇO-CLASSE'].value_counts()
print(tendencia)
```

```
PREÇO-CLASSE
50.0      10
80.0      10
100.0     10
Name: count, dtype: int64
```

### 3.2.2 - Entrega 2: Modelagem Estatística

```
[24] # Criando uma instância do modelo de regressão linear
model = LinearRegression()
```

```
[25] # Treinando o modelo com os dados de treinamento
model.fit(X_train, y_train)
```

```
LinearRegression()
```

```
[26] # Fazendo previsões com os dados de validação
predictions = model.predict(X_test)
```

```
[27] # Calculando as métricas
mae = mean_absolute_error(y_test, predictions)
mse = mean_squared_error(y_test, predictions)
rmse = mean_squared_error(y_test, predictions, squared=False) # Calculando RMSE a partir do MSE
r2 = r2_score(y_test, predictions)
```

```
# Exibindo as métricas
print("Erro Médio Absoluto (MAE):", mae)
print("Erro Quadrático Médio (MSE):", mse)
print("Raiz do Erro Quadrático Médio (RMSE):", rmse)
print("R-quadrado (R²):", r2)
```

```
[26] # Fazendo previsões com os dados de validação
      predictions = model.predict(X_test)

[27] # Calculando as métricas
      mae = mean_absolute_error(y_test, predictions)
      mse = mean_squared_error(y_test, predictions)
      rmse = mean_squared_error(y_test, predictions, squared=False) # Calculando RMSE a partir do MSE
      r2 = r2_score(y_test, predictions)
```


```
[47] # Função para calcular estatísticas descritivas das colunas numéricas
      def calcular_estatisticas_descritivas(df):
          estatisticas = {}
          for coluna in df.select_dtypes(include=['float64', 'int64']).columns:
              media = df[coluna].mean()
              mediana = df[coluna].median()
              desvio_padrao = df[coluna].std()
              estatisticas[coluna] = {'Média': media, 'Mediana': mediana, 'Desvio Padrão': desvio_padrao}
          return estatisticas

      def avaliar_modelo1(df):
          # Vamos usar as estatísticas descritivas das colunas numéricas como pontuação do modelo
          estatisticas = calcular_estatisticas_descritivas(df)
          pontuacao_modelo = sum(estatisticas[coluna]['Média'] for coluna in estatisticas)
          return pontuacao_modelo

      def avaliar_modelo2(df):
          # Vamos usar as estatísticas descritivas das colunas numéricas como pontuação do modelo
          estatisticas = calcular_estatisticas_descritivas(df)
          pontuacao_modelo = sum(estatisticas[coluna]['Desvio Padrão'] for coluna in estatisticas)
          return pontuacao_modelo

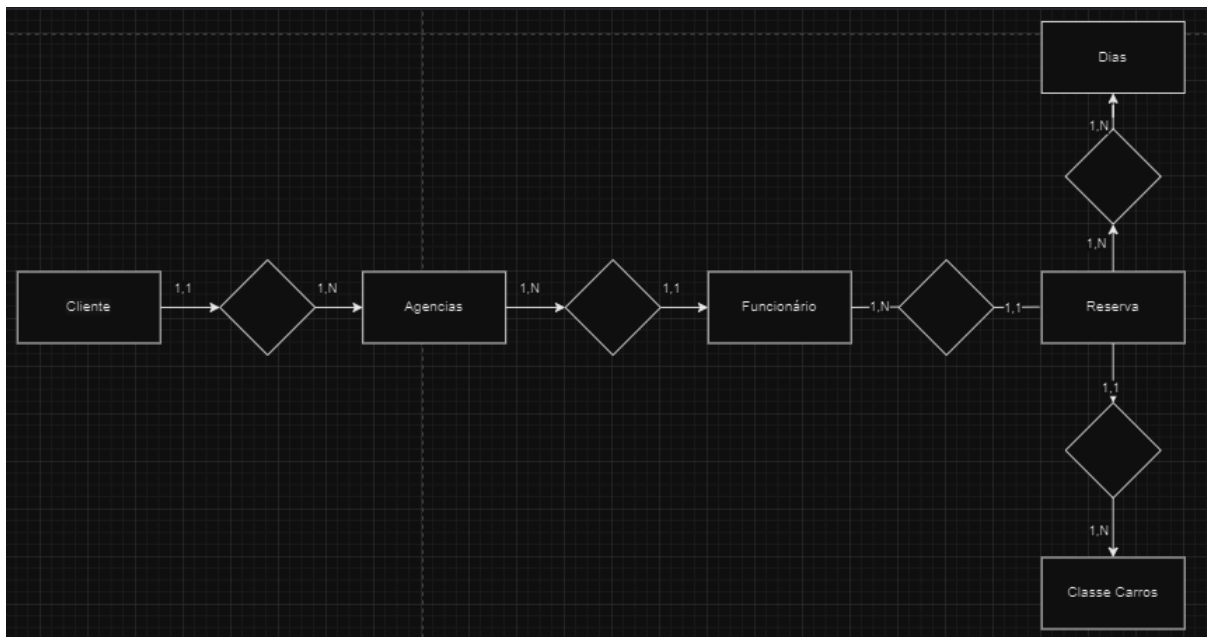
      # Comparar modelos
      resultado1 = avaliar_modelo1(df)
      resultado2 = avaliar_modelo2(df)

      if resultado1 > resultado2:
          print('Modelo 1 é melhor.')
      else:
          print('Modelo 2 é melhor.')
```

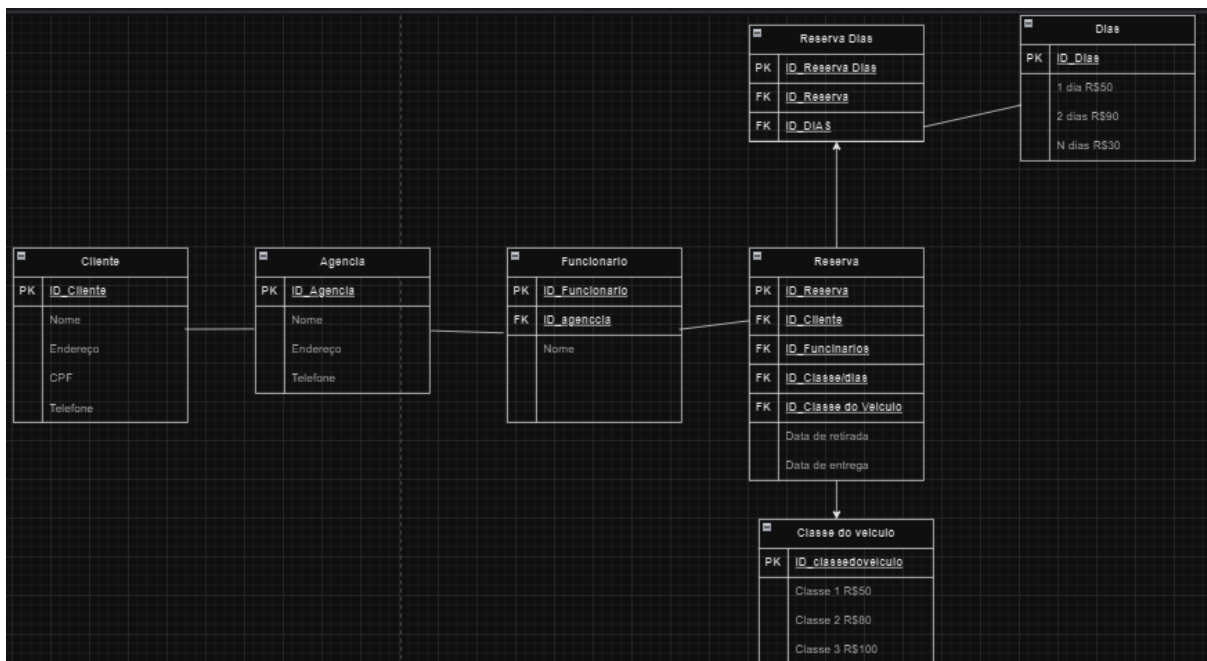
 Modelo 2 é melhor.

### 3.3 - Modelagem de Dados

#### 3.3.1 - Modelagem Conceitual



#### 3.3.2 - Entrega 2: Modelagem Lógica e Normalização



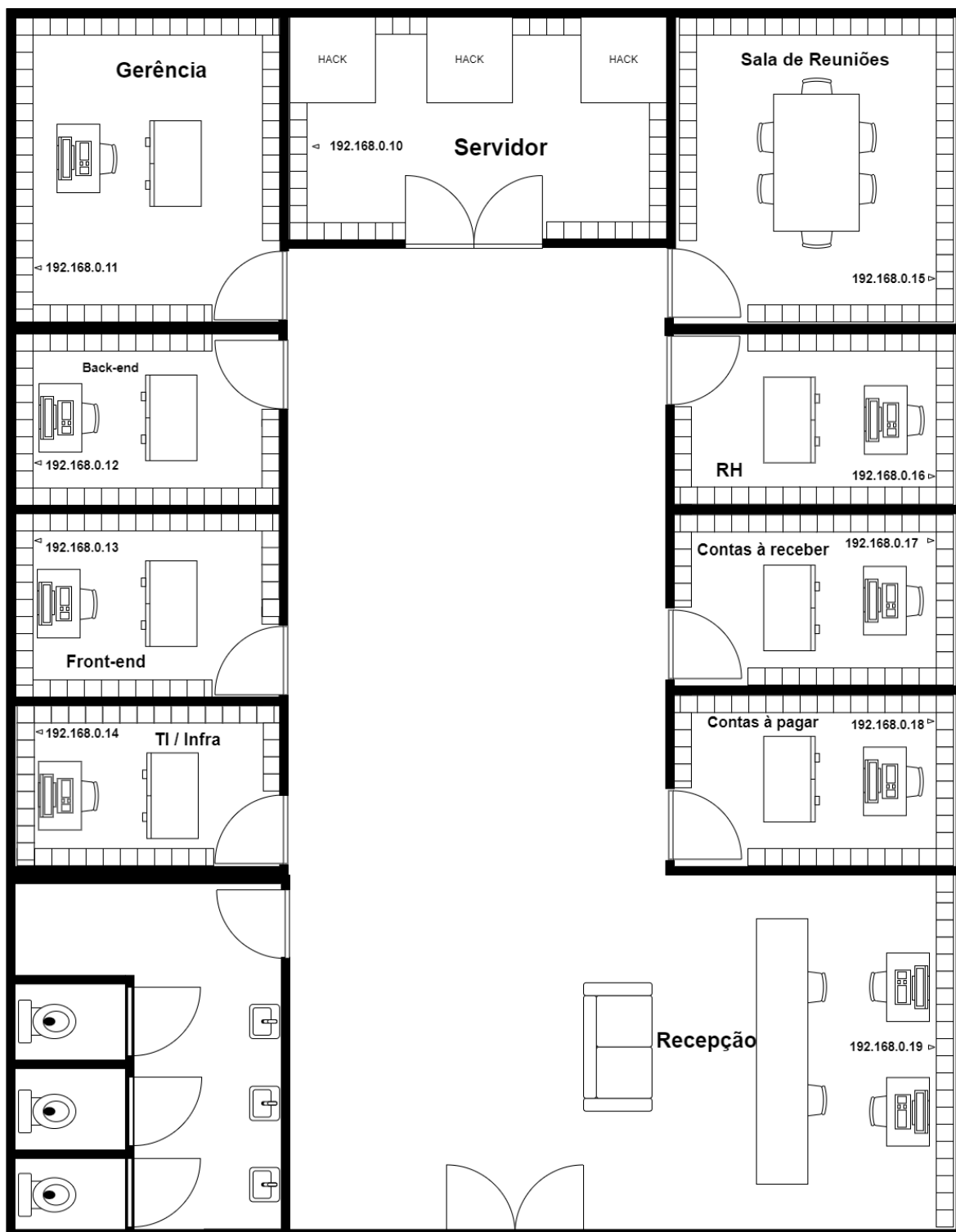
### 3.3.3 - Entrega 3: Entregar Dicionário de Dados uma simulação de cadastro

CLIENTE				
ID_CLIENTE	NOME	CPF	ENDEREÇO	TELEFONE
1	KLEBER	353115899-20	RUA 15 A	95050-4958
2	MARIA	450789215-10	RUA16 B	92323-7829
3	VITOR	265063547-01	RUA 17 C	96548-3216
AGÊNCIA				
ID_AGÊNCIA	NOME	ENDEREÇO_AGENCIA	TELEFONE	
101	Jardim das Flores	Praça das Magnólias, 279	95050 4958	
FUNCIONARIO				
ID_FUNCIONARIO	NOME	FUNÇÃO		
1010	SOFIA	ATENDENTE		
1011	ANTONIO	ATENDENTE		
RESERVA DIAS				
ID_RESERVA DIAS	ID DIAS	ID_RESERVAS		
1010111	101011	11		
1010112	101012	12		
1010113	101013	13		
RESERVA				
ID_RESERVA	ID_CLIENTE	ID_FUNCIONARIO	ID_CLASSE DO VEICULO	
11	1	1010	10101	
12	2	1011	10102	
13	3	1010	10103	
CLASSE DO VEICULO				
ID_CLASSE DO VEICULO	CLASSES	PREÇO		
10101	CLASSE 1	50		
10102	CLASSE 2	80		
10103	CLASSE 3	100		
DIAS				
ID DIAS	QTD DIAS			
101011	1			
101012	2			
101013	3 ou mais			



### 3.4 - Redes de Computadores

#### 3.4.1 - Entrega 1: Montar a planta baixa de Rede da Empresa



## 1. Servidor-

- Rack com sistema de segurança física
- Calha de tomadas 8 tomadas 10A
- Painel cego 1U - 19"
- Organizadores de cabos 1U - 19"
- Switch 36 portas
- Bandeja deslizante 19"
- Sistema de ventilação
- Unidade de armazenamento
- Fonte de alimentação
- Baterias estacionárias
- No-break
- Plataforma de gerenciamento remoto
- Modem
- Ponto de distribuição de fibra óptica

2. Gerência - Computador, Impressora, Ramal e Ponto de rede.

3. Back-end - Computador, Ponto de rede, Ramal.

4. Front-end - Computador, Ponto de rede, Ramal.

5. TI / infra - Computador, ponto de rede, Ramal.

6. Sala de Reuniões - Ponto de rede, Ramal

7. Rh- Computador, ponto de rede, Ramal.

8. Contas à Receber - Computador, Ponto de rede, Ramal, Impressora.

9. Conta à Pagar - Computador, Ponto de Rede, Ramal.

10. Recepção - Computador, Ponto de rede, Ramal, Impressora.

### **3.4.2 - Entrega 2: Configuração de IP de todos os equipamentos**

Classe de Rede: Classe C (192.168.0.0)

Padrão de Rede:

1. Servidor - 192.168.0.10
2. Gerência - 192.168.0.11
3. Back-end - 192.168.0.12
4. Front-end - 192.168.0.13
5. TI / Infra - 192.168.0.14
10. Sala de Reuniões - 192.168.0.15
6. RH - 192.168.0.16
7. Contas à receber - 192.168.0.17
8. Contas à pagar - 192.168.0.18
9. Recepção - 192.168.0.19

## **3.5 - Segurança da Informação**

### **3.5.1 - Entrega 1: Análise de Riscos**

- **Ataques de malware**
  - Impacto: Alto
  - Probabilidade: Média
  - Os ataques de malware podem resultar em perda de dados, interrupção das operações e danos à reputação da empresa.
- **Phishing**
  - Impacto: Médio
  - Probabilidade: Alta
  - O phishing pode resultar na divulgação de informações confidenciais dos clientes e funcionários, levando ao comprometimento da segurança da informação.
- **Roubo de dados de cartão de crédito**
  - Impacto: Alto
  - Probabilidade: Média
  - O roubo de dados de cartão de crédito pode resultar em responsabilidade legal, multas e danos à reputação da empresa.
- **Ataques de ransomware**
  - Impacto: Alto
  - Probabilidade: Média
  - Os ataques de ransomware podem resultar em interrupção das operações, perda de dados críticos e custos de recuperação significativos.
- **Acesso não autorizado aos sistemas**
  - Impacto: Médio
  - Probabilidade: Média
  - O acesso não autorizado pode levar à divulgação de informações confidenciais, roubo de dados e comprometimento da segurança dos sistemas.
- **Vazamento de informações de clientes**
  - Impacto: Alto
  - Probabilidade: Média
  - O vazamento de informações de clientes pode resultar em danos à reputação da empresa, perda de confiança dos clientes e consequências legais.
- **Ataques de negação de serviço (DDoS)**
  - Impacto: Alto
  - Probabilidade: Baixa
  - Os ataques DDoS podem resultar em interrupção das operações, perda de receita e danos à reputação da empresa.

- **Fraude de identidade**
  - Impacto: Médio
  - Probabilidade: Média
  - A fraude de identidade pode resultar em perda financeira, danos à reputação da empresa e impacto nas operações comerciais.
- **Falta de atualizações de segurança**
  - Impacto: Alto
  - Probabilidade: Média
  - A falta de atualizações de segurança pode deixar os sistemas vulneráveis a ataques e explorações de vulnerabilidades conhecidas.
- **Violação de conformidade regulatória**
  - Impacto: Alto
  - Probabilidade: Média
  - Violações regulatórias podem resultar em multas significativas, perda de licenças comerciais e danos à reputação da empresa.
- **Ameaças internas**
  - Impacto: Médio
  - Probabilidade: Média
  - As ameaças internas, como funcionários descontentes ou mal-intencionados, podem resultar em roubo de dados, sabotagem e violações de segurança.
- **Perda ou roubo de dispositivos móveis**
  - Impacto: Médio
  - Probabilidade: Média
  - A perda ou roubo de dispositivos móveis pode resultar na divulgação de informações confidenciais dos clientes e funcionários.
- **Fraude de cartão de crédito**
  - Impacto: Alto
  - Probabilidade: Média
  - A fraude de cartão de crédito pode resultar em perda financeira, disputas de clientes e danos à reputação da empresa.
- **Vulnerabilidades de software não corrigidas**
  - Impacto: Alto
  - Probabilidade: Média
  - As vulnerabilidades de software não corrigidas podem ser exploradas por invasores para comprometer a segurança dos sistemas.
- **Falta de conscientização em segurança da informação**
  - Impacto: Médio
  - Probabilidade: Alta
  - A falta de conscientização em segurança da informação pode resultar em práticas inadequadas de segurança, aumentando o risco de incidentes de segurança.
- **Roubo de propriedade intelectual**
  - Impacto: Alto

- Probabilidade: Média
  - O roubo de propriedade intelectual pode resultar em perda de vantagem competitiva, danos financeiros e disputas legais.
- **Ataques de engenharia social**
  - Impacto: Médio
  - Probabilidade: Média
  - Os ataques de engenharia social podem levar à divulgação de informações confidenciais ou à instalação de malware nos sistemas da empresa.
- **Falta de backups adequados**
  - Impacto: Alto
  - Probabilidade: Média
  - A falta de backups adequados pode resultar na perda permanente de dados críticos em caso de falha do sistema ou ataque de ransomware.
- **Violação de privacidade dos clientes**
  - Impacto: Alto
  - Probabilidade: Média
  - Violações de privacidade dos clientes podem resultar em danos à reputação da empresa, disputas legais e perda de confiança dos clientes.
- **Falta de política de segurança da informação**
  - Impacto: Alto
  - Probabilidade: Média
  - A falta de uma política de segurança da informação pode deixar a empresa vulnerável a ataques, violações de dados e consequências legais.

### **3.5.2 - Entrega 2: Implementação de Medidas de Segurança**

1. Implementação de políticas de controle de acesso aos sistemas e dados:

**Gerência:**

Acesso completo a todos os dados e sistemas da empresa.

Permissão para visualizar e modificar dados financeiros, mas com monitoramento regular para garantir conformidade e segurança.

**Back-end:**

Acesso restrito aos sistemas e bancos de dados necessários para o desenvolvimento e manutenção das aplicações.

Não ter acesso direto aos dados financeiros, a menos que especificamente autorizado para fins de resolução de problemas.

**Front-end:**

Acesso aos sistemas de interface do usuário e a elementos visuais das aplicações.

Sem acesso direto aos dados financeiros, exceto para visualização de dados não confidenciais necessários para o desenvolvimento de interfaces.

**TI / Infraestrutura:**

Acesso privilegiado aos sistemas e infraestrutura de rede da empresa.

Acesso restrito aos dados financeiros, limitado àqueles necessários para manutenção e segurança da infraestrutura.

**Sala de Reuniões:**

Acesso físico restrito a membros autorizados da equipe para reuniões e apresentações.

Nenhum acesso aos sistemas ou dados financeiros, a menos que autorizado individualmente.

**RH:**

Acesso a dados pessoais dos funcionários para gestão de recursos humanos.

Sem acesso direto aos dados financeiros, exceto para fins de processamento de folha de pagamento e reembolsos.

**Contas a Receber:**

Acesso restrito aos sistemas e dados relacionados às contas a receber da empresa.

Permissão para visualizar e modificar dados financeiros específicos às contas a receber, mas sem acesso aos dados financeiros gerais.

**Contas a Pagar:**

Acesso restrito aos sistemas e dados relacionados às contas a pagar da empresa.

Permissão para visualizar e modificar dados financeiros específicos às contas a pagar, mas sem acesso aos dados financeiros gerais.

**Recepção:**

Acesso limitado aos sistemas de entrada e registro de visitantes.

Nenhum acesso aos sistemas ou dados financeiros, exceto para fins de registro de despesas relacionadas à recepção.

**2. Configurações de sistemas de detecção de intrusão e prevenção de ataques:**

- **Sistema de Detecção de Intrusão (IDS/IPS):**
  - Configurar alertas em caso de atividades suspeitas na rede, como tentativas de acesso não autorizado, tráfego malicioso ou padrões incomuns de comportamento.
- **Firewalls de Próxima Geração:**

- Implementar firewalls capazes de inspecionar o tráfego de rede em nível de aplicação, identificando e bloqueando ameaças conhecidas e desconhecidas.
- **Soluções de Antivírus/Antimalware Avançadas:**
  - Utilizar soluções de antivírus e antimalware que empregam inteligência artificial e aprendizado de máquina para identificar e neutralizar ameaças em tempo real.
- **Monitoramento de Log de Segurança:**
  - Realizar monitoramento contínuo dos logs de segurança para identificar padrões suspeitos de atividade, como tentativas de login falhas, acessos a arquivos sensíveis e alterações de configuração não autorizadas.
- **Análise Comportamental de Usuário e Entidade (UEBA):**
  - Implementar soluções de UEBA para detectar comportamentos anômalos de usuários e entidades dentro da rede, ajudando a identificar possíveis ameaças internas.
- **Autenticação Multifatorial (MFA):**
  - Exigir autenticação multifatorial para acessar sistemas e aplicativos críticos, adicionando uma camada extra de segurança contra acesso não autorizado.
- **Criptografia de Dados:**
  - Criptografar dados sensíveis em repouso e em trânsito para protegê-los contra acesso não autorizado, mesmo se forem comprometidos.
- **Atualizações e Patch Management:**
  - Implementar um processo rigoroso de gerenciamento de patches para garantir que sistemas e aplicativos sejam regularmente atualizados com as últimas correções de segurança.
- **Testes de Penetração e Vulnerabilidade:**
  - Realizar testes de penetração regulares para identificar e corrigir vulnerabilidades de segurança antes que sejam exploradas por invasores.
- **Treinamento e Conscientização em Segurança:**
  - Fornecer treinamento regular em segurança cibernética para funcionários, educando-os sobre práticas seguras de computação, reconhecimento de ameaças e procedimentos de resposta a incidentes.