



Campus Santo Amaro

Tecnologia em análise e desenvolvimento de sistemas

Turma 50



Loja de bicicleta – Click Flow

São Paulo

2024



Colaboradores:

Carlos Henrique Bezerra da Silva – 2224108274

Gabriel Leme Dantas - 2224101466

Geovana Lilian Gomes da Silva - 2224108323

Felipe Ito de Araújo - 2224106783

Karen Ribeiro Soares - 2224106637

Kevin Freitas Santos - 2224105975

Carolina Alves de Melo - 2224106174

Yasmin Felix Silva - 2224108583

Letícia Vasconcelos Bulhões - 2224108612

Amanda Raissa Hipólito Andrade - 2224109373



Sumário

Colaboradores:	2
Aprendizado de Máquina, Machine Learning e Ciência de Dados	4
Entrega 1: Exploração de Dados e Pré-processamento.....	4
Entrega 2: Implementação de Modelos de Aprendizado de Máquina de Regressão Linear	7
Entrega 3: Implementação de Modelos de Aprendizado de Máquina de Classificação	9
Modelagem de Dados	10
Entrega 1: Modelagem Conceitual	10
Entrega 2: Modelagem Lógica e Normalização.....	10
Entrega 3: Entregar Dicionário de Dados uma simulação de cadastro	10
Redes de Computadores	12
Entrega 1: Montar a planta baixa de Rede da Empresa	12
Entrega 2: Configuração de IP de todos os equipamentos	12
Segurança da Informação	14
Entrega 1: Implementação de Medidas de Segurança.....	14
Entrega 2: Matriz GUT	17

Aprendizado de Máquina, Machine Learning e Ciência de Dados

Entrega 1: Exploração de Dados e Pré-processamento

```
# Importação das bibliotecas necessárias
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix

# Carregar os dados
df = pd.read_csv('bike_buyers (2).csv')

# Remover as colunas 'ID', 'Education', 'Occupation', 'Home Owner'
df.drop(columns=['ID', 'Education', 'Occupation', 'Home Owner'], inplace=True)

# Verificar valores ausentes
print("\nVerificação de valores ausentes por coluna:")
print(df.isnull().sum())

# Preencher valores ausentes (se houver) com a mediana (para colunas numéricas)
df.fillna(df.median(numeric_only=True), inplace=True)

# Alterar tipo de dado da coluna 'Purchased Bike' para booleano
df['Purchased Bike'] = df['Purchased Bike'].apply(lambda x: True if x == 'Yes' else False)

# Alterar tipo de dado da coluna 'Annual Income' para float
df['Income'] = df['Income'].astype(float)

# Manter os valores originais para visualização
df_visual = df.copy()

# Normalizar as colunas numéricas com StandardScaler (apenas para modelagem)
scaler = StandardScaler()
df_scaled = pd.DataFrame(scaler.fit_transform(df[['Income', 'Children', 'Cars', 'Age']]),
                        columns=['Income', 'Children', 'Cars', 'Age'])

# Substituir as colunas normalizadas no DataFrame original (apenas para modelagem)
df[['Income', 'Children', 'Cars', 'Age']] = df_scaled

# Separar dados em treino e teste
X = df.drop(columns='Purchased Bike') # Features (ajustado para a coluna 'Purchased Bike')
y = df['Purchased Bike'] # Target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Exibir as dimensões dos conjuntos de treino e teste
print(f"Dimensões do conjunto de treino: {X_train.shape}")
print(f"Dimensões do conjunto de teste: {X_test.shape}")

# Exibir DataFrame original para visualização (com valores reais)
df_visual.head()
```

```
# =====

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt

# Carregar o arquivo CSV
df = pd.read_csv('bike_buyers (2).csv')

# Converter a coluna 'Purchased Bike' para valores binários (1 para "Yes" e 0 para "No")
df['Purchased Bike'] = df['Purchased Bike'].map({'Yes': 1, 'No': 0})

# Remover colunas irrelevantes (como ID e name) e lidar com valores ausentes
df = df.drop(['ID', 'name'], axis=1)
df = df.dropna()

# Codificar variáveis categóricas
df = pd.get_dummies(df, drop_first=True)

# Separar variáveis de entrada (X) e variável alvo (y)
X = df.drop('Purchased Bike', axis=1)
y = df['Purchased Bike']

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

# Treinar um classificador de Árvore de Decisão
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)

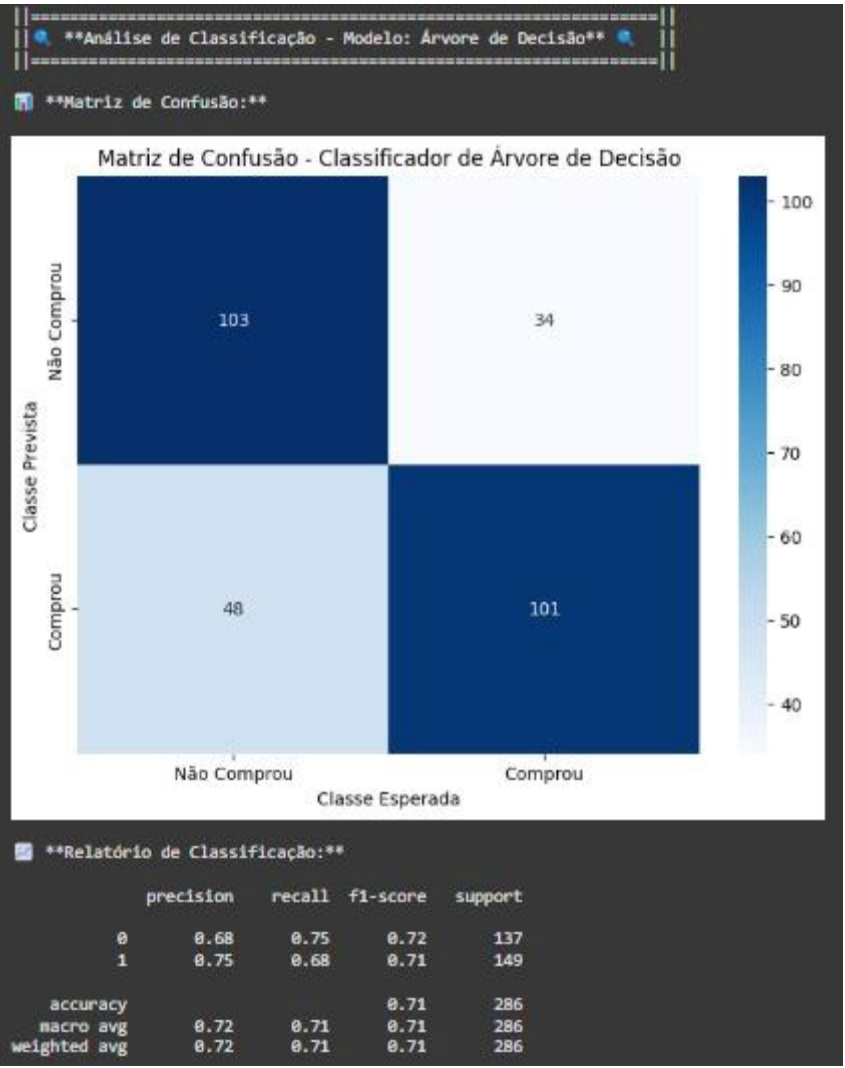
# Fazer previsões no conjunto de teste
y_pred = clf.predict(X_test)

# Gerar a matriz de confusão
conf_matrix = confusion_matrix(y_test, y_pred)

print("||=====||")

# Visualizar a matriz de confusão com mais estilo
print("|| 🐦 **Análise de Classificação - Modelo: Árvore de Decisão** 🐦 ||")
print("||=====||")
print("\n 🐦 **Matriz de Confusão:**\n")
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues",
            xticklabels=['Não Comprou', 'Comprou'],
            yticklabels=['Não Comprou', 'Comprou'])
plt.xlabel("Classe Esperada")
plt.ylabel("Classe Prevista")
plt.title("Matriz de Confusão - Classificador de Árvore de Decisão")
plt.show()

# Exibir o relatório de classificação
print("\n 🐦 **Relatório de Classificação:**\n")
print(classification_report(y_test, y_pred))
```



Entrega 2: Implementação de Modelos de Aprendizado de Máquina de Regressão Linear

REGRESSÃO LINEAR

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Carregar o arquivo CSV
df = pd.read_csv('bike_buyers (2).csv')

# Definir faixas etárias e rótulos
bins = [20, 30, 40, 50, 60, 70]
labels = ['20-30', '30-40', '40-50', '50-60', '60-70']
df['Age Range'] = pd.cut(df['Age'], bins=bins, labels=labels, right=False)

# Contar a quantidade de bicicletas compradas por faixa etária
bike_counts = df[df['Purchased Bike'] == 'Yes'].groupby('Age Range').size()

# Calcular os pontos médios das faixas etárias e os valores ajustados
x_trein = np.array([(bins[i] + bins[i+1]) / 2 for i in range(len(bins) - 1)])
y_trein = np.array(bike_counts.reindex(labels, fill_value=0))

# Regressão linear
N = len(x_trein)
sum_x = np.sum(x_trein)
sum_y = np.sum(y_trein)
sum_xy = np.sum(x_trein * y_trein)
sum_x2 = np.sum(x_trein ** 2)

# Cálculo do peso (w) e viés (b)
w = (N * sum_xy - sum_x * sum_y) / (N * sum_x2 - sum_x ** 2)
b = (sum_y - w * sum_x) / N

# Previsão da linha de regressão
f_wb = w * x_trein + b

# Plotando o gráfico de regressão linear com os dados ajustados
plt.figure(figsize=(10, 6))

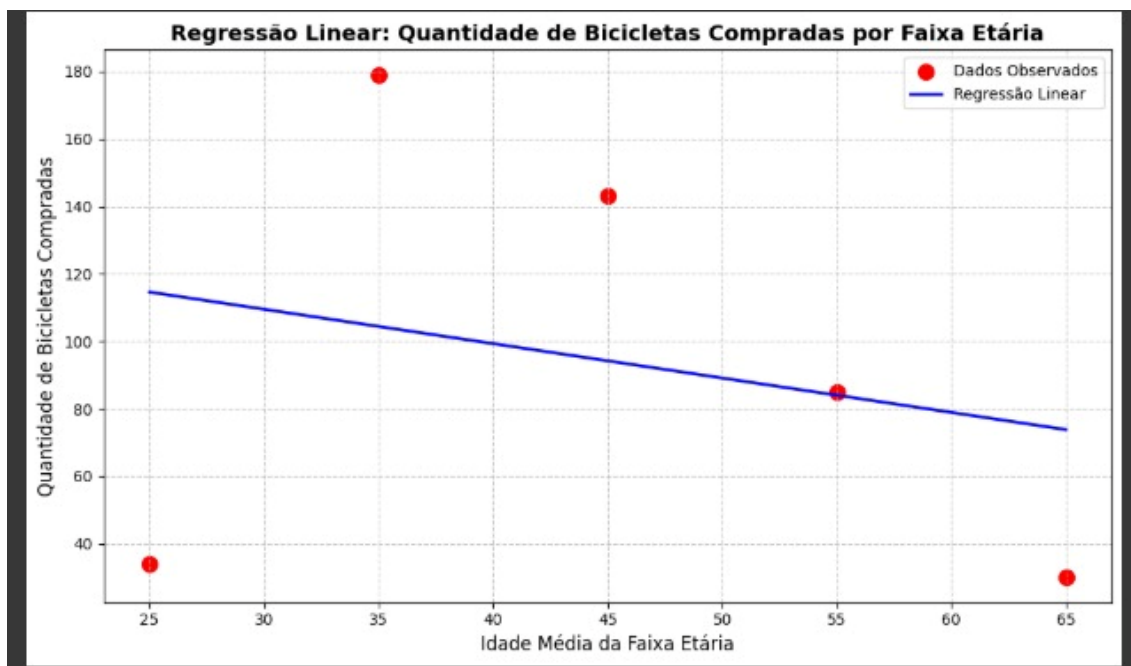
# Plotando os dados observados (pontos vermelhos)
plt.scatter(x_trein, y_trein, color='red', marker='o', s=100, label='Dados Observados')

# Plotando a linha de regressão
plt.plot(x_trein, f_wb, color='blue', label='Regressão Linear', linewidth=2)

# Ajustando rótulos e título
plt.ylabel('Quantidade de Bicicletas Compradas', fontsize=12)
plt.xlabel('Idade Média da Faixa Etária', fontsize=12)
plt.title('Regressão Linear: Quantidade de Bicicletas Compradas por Faixa Etária', fontsize=14, fontweight='bold')
plt.legend()

# Melhorando a grade do gráfico
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout()

# Exibir o gráfico
plt.show()
```



Entrega 3: Implementação de Modelos de Aprendizado de Máquina de Classificação

CLASSIFICAÇÃO POR MATPLOTT

```
# MATPLOTT DAS COMPRAS DAS BICICLETAS
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns

# Converte 'Purchased Bike' para valores numéricos, se necessário
df['Purchased Bike'] = df['Purchased Bike'].map({'Yes': 1, 'No': 0}).fillna(df['Purchased Bike'])

# Calcula a quantidade de compradores e não compradores
purchased_count = df['Purchased Bike'].sum()
not_purchased_count = len(df) - purchased_count

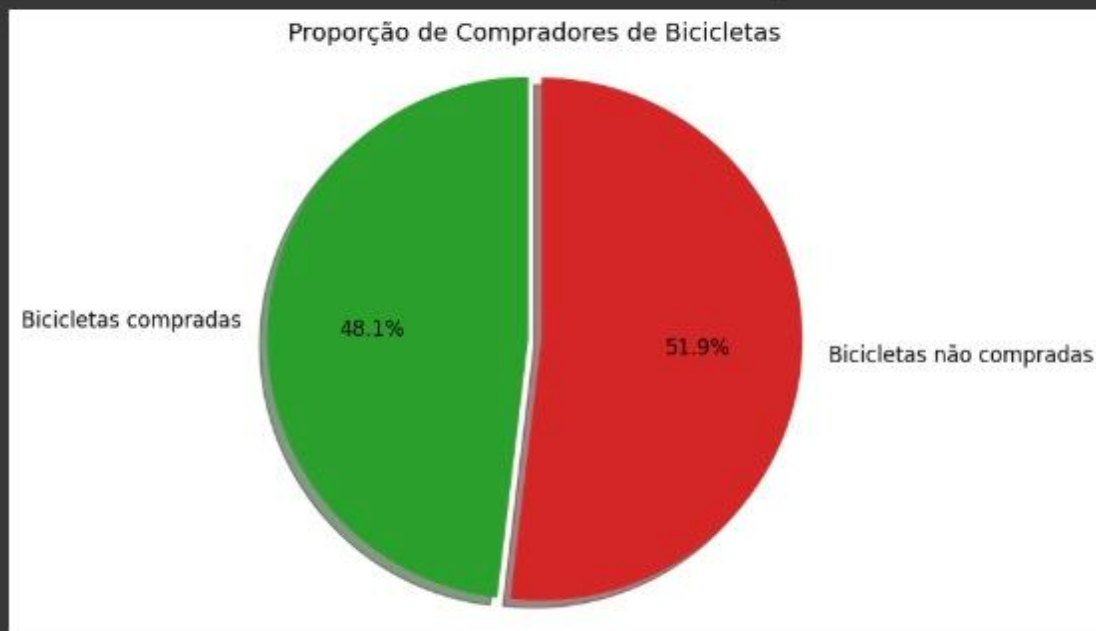
# Define os dados para o gráfico de pizza
numbers = [purchased_count, not_purchased_count]
labels = ["Bicicletas compradas", "Bicicletas não compradas"]
colors = ["#2ca02c", "#d62728"] # Usando cores mais agradáveis
explode = (0.05, 0) # Destacando a fatia "Purchased Bike"

# Cria o gráfico de pizza com configurações aprimoradas
plt.figure(figsize=(6, 6)) # Ajustando o tamanho da figura
plt.pie(numbers, labels=labels, autopct='%1.1f%%', colors=colors,
        explode=explode, shadow=True, startangle=90, textprops={'fontsize': 12})

# Adiciona um título informativo
plt.title("Proporção de Compradores de Bicicletas", fontsize=14)

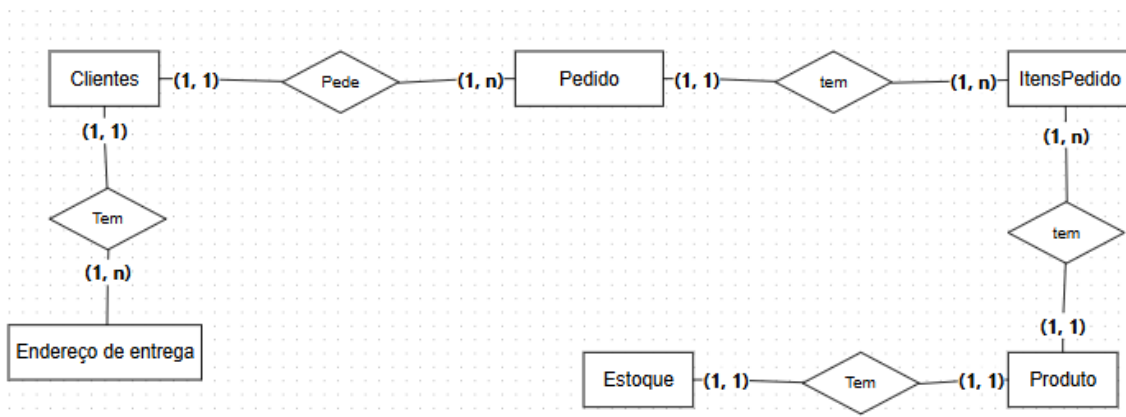
# Centraliza o gráfico
plt.axis('equal')

# Exibe o gráfico
plt.show()
```

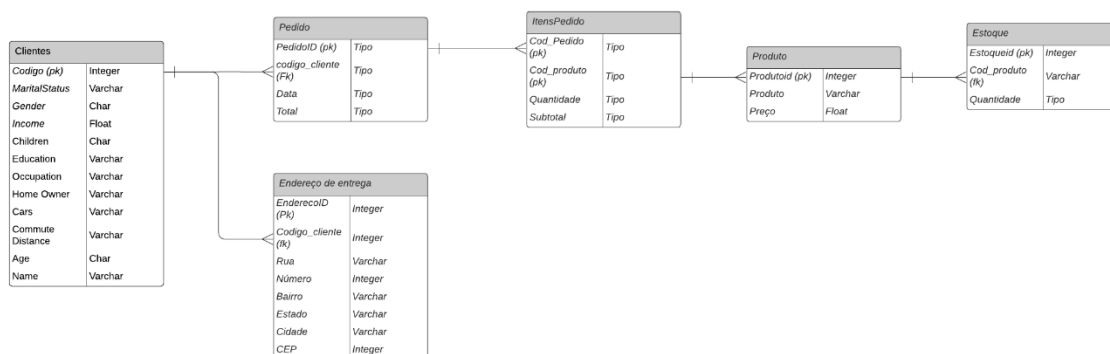


Modelagem de Dados

Entrega 1: Modelagem Conceitual



Entrega 2: Modelagem Lógica e Normalização



Entrega 3: Entregar Dicionário de Dados uma simulação de cadastro

Clientes			
Campo	Tipo	Descrição	Chave
Codigo	Integer	Código único do cliente	PK
MaritalStatus	Varchar	Estado civil do cliente	
Gender	Char	Gênero do cliente	
Income	Float	Renda anual do cliente	
Children	Integer	Número de filhos do cliente	
Education	Varchar	Nível de educação do cliente	
Occupation	Varchar	Profissão do cliente	
Home Owner	Varchar	Indica se o cliente é proprietário de imóvel	
Cars	Varchar	Número de carros do cliente	
Commute Distance	Varchar	Distância média de deslocamento do cliente	
Age	Char	Idade do cliente	
Name	Varchar	Nome do cliente	



Pedido			
Campo	Tipo	Descrição	Chave
PedidoID (PK)	Integer	Identificador único do pedido.	PK
Codigo_Cliente (FK)	Integer	Código do cliente que fez o pedido.	FK
Data	Tipo	Data em que o pedido foi realizado.	
Total	Tipo	Valor total do pedido.	

ItensPedido			
Campo	Tipo	Descrição	Chave
Cod_Pedido (FK)	Tipo	Código do pedido associado.	FK
Cod_Produto (FK)	Tipo	Código do produto incluído no pedido.	FK
Quantidade	Tipo	Quantidade do produto no pedido.	
Subtotal	Tipo	Valor subtotal dos itens no pedido.	

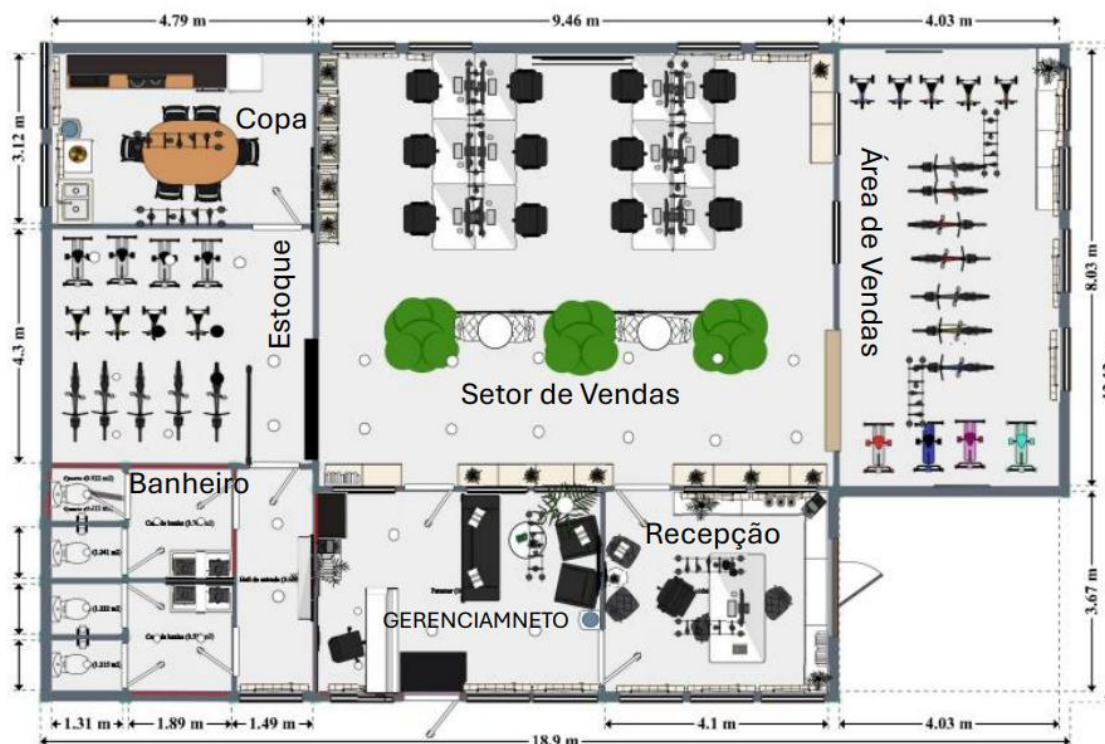
Produto			
Campo	Tipo	Descrição	Chave
ProdutoID (PK)	Integer	Identificador único do produto.	PK
Produto	Varchar	Nome ou descrição do produto.	
Preço	Float	Preço unitário do produto.	

Estoque			
Campo	Tipo	Descrição	Chave
EstoqueID	Integer	Identificador único do estoque.	PK
Cod_Produto (FK)	Varchar	Código do produto disponível no estoque.	FK
Quantidade	Tipo	Quantidade do produto disponível no estoque.	

Endereço de entrega			
Campo	Tipo	Descrição	Chave
EnderecoID (PK)	Integer	Identificador único do endereço.	PK
Codigo_Cliente (FK)	Integer	Código do cliente associado ao endereço.	FK
Rua	Varchar	Nome da rua do endereço.	
Numero	Varchar	Número do imóvel no endereço.	
Bairro	Varchar	Bairro do endereço.	
Estado	Varchar	Estado onde o endereço está localizado.	
Cidade	Varchar	Cidade do endereço.	
CEP	Integer	Código postal do endereço.	

Redes de Computadores

Entrega 1: Montar a planta baixa de Rede da Empresa



RECEPÇÃO

01 PC

01 Impressora

SALA DE GESTÃO

01 Notebook

01 Impressora

SETOR DE VENDAS ONLINE

01 TV

12 Notebook

Entrega 2: Configuração de IP de todos os equipamentos

Faixas de IPs e Máscara de Sub-rede:

- Faixa de IP principal: 192.168.0.0/24
- Gateway (Roteador): 192.168.0.1



- **DNS:** 8.8.8.8 (primário), 8.8.4.4 (secundário)

Setor de Vendas (12 Notebooks + TV)

12 Notebooks do Setor de Vendas:

- **Faixa de IPs para DHCP:** 192.168.0.10 a 192.168.0.21 (total de 12 endereços)
- **Máscara de Sub-rede:** 255.255.255.0
- **Gateway:** 192.168.0.1
- **DNS:** 8.8.8.8 (primário), 8.8.4.4 (secundário)

TV de Setor de Vendas

- **IP Estático:** 192.168.0.30
- **Máscara de Sub-rede:** 255.255.255.0
- **Gateway:** 192.168.0.1
- **DNS:** 8.8.8.8 (primário), 8.8.4.4 (secundário)

Sala de Gestão (1 Notebook + 1 Impressora)

- **Notebook de Gestão:**
- **IP Estático:** 192.168.0.30
- **Máscara de Sub-rede:** 255.255.255.0
- **Gateway:** 192.168.0.1
- **DNS:** 8.8.8.8 (primário), 8.8.4.4 (secundário)

Impressora de Gestão:

- **IP Estático:** 192.168.0.31
- **Máscara de Sub-rede:** 255.255.255.0
- **Gateway:** 192.168.0.1
- **DNS:** 8.8.8.8 (primário), 8.8.4.4 (secundário)

Recepção (1 PC + 1 Impressora)

Para a recepção, configuramos **IP estático** para o PC e a impressora.

- **PC de Recepção:**
- **IP Estático:** 192.168.0.40
- **Máscara de Sub-rede:** 255.255.255.0
- **Gateway:** 192.168.0.1
- **DNS:** 8.8.8.8 (primário), 8.8.4.4 (secundário)

Impressora de Recepção:

- **IP Estático:** 192.168.0.41
- **Máscara de Sub-rede:** 255.255.255.0



- **Gateway:** 192.168.0.1
- **DNS:** 8.8.8.8 (primário), 8.8.4.4 (secundário)

Segurança da Informação

Entrega 1: Implementação de Medidas de Segurança

1. Política de Senhas Fortes

- Todos os usuários devem criar senhas fortes que incluem uma combinação de letras maiúsculas, minúsculas, números e caracteres especiais. A senha deve ter, no mínimo, 12 caracteres e ser alterada a cada 90 dias.

2. Política de Autenticação de Dois Fatores (2FA)

- A autenticação de dois fatores deve ser habilitada para o acesso ao sistema, especialmente para contas de administradores e gerentes, para garantir uma camada extra de segurança.

3. Política de Acesso Baseado em Função (Role-Based Access Control - RBAC)

- O acesso aos sistemas e dados deve ser limitado com base nas funções dos usuários. Apenas o mínimo de permissões necessárias para executar as atividades deve ser concedido.

4. Política de Princípio de Menor Privilégio

- Todos os usuários devem ter acesso apenas aos dados e sistemas necessários para o desempenho de suas tarefas, restringindo privilégios administrativos ao mínimo.

5. Política de Revalidação Periódica de Acesso

- Revisões de acesso devem ser realizadas a cada seis meses para garantir que as permissões de acesso de cada usuário estejam atualizadas com base em suas funções atuais.

6. Política de Log de Acesso

- Todos os acessos aos sistemas e dados devem ser registrados em logs, incluindo informações como horário, usuário, recurso acessado e ações realizadas, para fins de auditoria e monitoramento.

7. Política de Controle de Acesso Físico

- O acesso físico ao local onde os sistemas e servidores são mantidos deve ser restrito a pessoal autorizado, com uso de controles de entrada, como cartões de acesso.

8. Política de Bloqueio Automático de Sessões Inativas

- As sessões de usuários inativas por mais de 15 minutos devem ser automaticamente bloqueadas, exigindo que o usuário faça login novamente para continuar.

9. Política de Acesso Remoto Seguro



- O acesso remoto ao sistema só deve ser permitido por meio de redes seguras e VPNs (Redes Privadas Virtuais), com autenticação 2FA para aumentar a segurança.

10. Política de Acesso Temporário

- Qualquer acesso temporário para realização de tarefas específicas deve ter uma data de expiração, após a qual o acesso é automaticamente revogado.

11. Política de Controle de Acesso a Dados Sensíveis

- O acesso a dados sensíveis, como informações de clientes e de pagamento, deve ser restrito a pessoal autorizado, e esses dados devem estar criptografados.

12. Política de Treinamento de Segurança de Acesso

- Todos os funcionários devem passar por treinamentos periódicos sobre práticas de segurança e políticas de controle de acesso, para conscientização e prevenção de incidentes.

13. Política de Gestão de Senhas de Sistemas Críticos

- Senhas de sistemas críticos devem ser armazenadas em gerenciadores de senha seguros e devem ser trocadas imediatamente após qualquer alteração de pessoal.

14. Política de Requisição de Acesso

- Novos acessos devem ser solicitados e justificados por meio de um formulário formal, com aprovação de um supervisor.

15. Política de Revogação de Acesso

- O acesso aos sistemas e dados de funcionários deve ser imediatamente revogado no caso de desligamento, mudança de função ou transferência.

16. Política de Controle de Dispositivos Pessoais

- O uso de dispositivos pessoais para acesso aos sistemas da empresa deve ser evitado ou monitorado. Caso seja permitido, os dispositivos devem seguir padrões de segurança da empresa.

17. Política de Controle de Acesso de Fornecedores e Terceiros

- Fornecedores e terceiros que necessitem de acesso aos sistemas devem ter acessos restritos e monitorados, e apenas durante o tempo necessário para realizar suas atividades.

18. Política de Monitoramento e Auditoria de Acesso

- Os registros de acesso devem ser revisados periodicamente para identificar e corrigir quaisquer acessos indevidos ou comportamentos suspeitos.

19. Política de Proteção Contra Engenharia Social

- Orientar os funcionários sobre os riscos de engenharia social e phishing, enfatizando a importância de não divulgar informações de acesso ou detalhes de segurança a pessoas não autorizadas.



20. Política de Reporte de Acessos Suspeitos

- Os funcionários devem ter um canal de comunicação para relatar qualquer tentativa de acesso suspeita ou comportamento anômalo nos sistemas, incentivando o reporte imediato.

Medidas de Detecção e Prevenção de Ataques

- **Instalação de um Sistema IDS/IPS**
 - Configure um sistema IDS/IPS como o **Snort**, **Suricata** ou **Zeek** para monitorar a rede em tempo real, detectando e prevenindo ataques com base em assinaturas e comportamentos suspeitos.
- **Firewall de Próxima Geração (NGFW)**
 - Utilize um firewall de próxima geração que combine funções de firewall tradicionais com capacidades de inspeção profunda de pacotes e detecção de ameaças, oferecendo maior proteção contra ataques complexos.
- **Monitoramento e Bloqueio de IPs Suspeitos**
 - Configure listas de bloqueio para impedir o acesso de endereços IP conhecidos por atividades maliciosas. Ferramentas como IDS/IPS e firewalls podem ser programadas para bloquear automaticamente esses IPs.
- **Análise de Comportamento de Rede (NBA)**
 - Implante sistemas de análise de comportamento de rede que detectem anomalias de tráfego. Essa tecnologia permite identificar comportamentos atípicos, como grandes transferências de dados fora do horário comercial, o que pode indicar um possível ataque.
- **Segmentação de Rede**
 - Divida a rede em segmentos (zonas) para limitar o acesso a dados sensíveis. A segmentação de rede impede que, caso um invasor acesse uma área, ele comprometa o restante da rede.
- **Sistema de Monitoramento de Logs (SIEM)**
 - Use um sistema de monitoramento de eventos e informações de segurança (SIEM), como **Splunk** ou **ELK Stack**, que centraliza e analisa logs para detectar padrões de ataque em eventos passados e emitir alertas.
- **Autenticação de Dois Fatores (2FA)**
 - Configure autenticação de dois fatores para acesso aos sistemas críticos e à rede, exigindo que funcionários usem um segundo fator de autenticação além da senha, dificultando invasões com credenciais roubadas.
- **Teste de Penetração Regular**
 - Realize testes de penetração periódicos para identificar vulnerabilidades na rede e nos sistemas, corrigindo falhas que possam ser exploradas por invasores.

- **Configuração de Honeypots**
 - Instale honeypots em áreas não críticas da rede para atrair e monitorar possíveis invasores. Essas armadilhas permitem identificar e analisar táticas de ataque usadas para melhorar as defesas.
- **Atualizações Automáticas e Gerenciamento de Patches**
 - Configure atualizações automáticas e um processo de gerenciamento de patches para garantir que todos os sistemas estejam protegidos contra as últimas ameaças conhecidas e vulnerabilidades.

Entrega 2: Matriz GUT

Problema/Ação	Gravidade (G)	Urgência (U)	Tendência (T)	G x U x T	Justificativa
1. Falta de controle de inventário	5	4	4	80	Pode causar falta de produtos para venda, perda de vendas e aumento de custos.
2. Atendimento ao cliente insatisfatório	4	5	4	80	Clientes insatisfeitos podem migrar para concorrentes, impactando diretamente as vendas e a imagem da loja.
3. Problemas de segurança nos dados dos clientes	5	5	5	125	Exposição de dados pode levar a prejuízos financeiros e danos à reputação.
4. Ausência de política de devolução	3	3	3	27	Pode gerar insatisfação entre clientes, mas tem impacto menor que outros problemas.
5. Falhas na manutenção preventiva dos equipamentos	4	4	3	48	Equipamentos sem manutenção podem gerar custos adicionais e comprometer a qualidade dos serviços.
6. Marketing insuficiente para alcançar novos clientes	3	4	4	48	A falta de marketing reduz o alcance e a atração de novos clientes, afetando o crescimento.
7. Falta de treinamento para equipe de vendas	4	4	4	64	Sem treinamento adequado, a equipe pode ter baixo desempenho e impactar as vendas e satisfação do cliente.

8. Sistema de controle de estoque ineficaz	4	3	4	48	A falta de controle pode levar a compras desnecessárias e falta de itens importantes.
9. Atraso no atendimento de pedidos online	4	5	5	100	Os atrasos afetam a experiência do cliente e a confiabilidade do serviço de e-commerce.
10. Dificuldade em obter feedback dos clientes	3	3	3	27	Limita a capacidade de melhorar o serviço, mas é menos impactante do que outros problemas.

Explicação dos Principais Problemas

- **Problemas de segurança nos dados dos clientes (125 pontos):** Este é o problema mais crítico. A segurança dos dados é extremamente importante para evitar possíveis multas e danos à imagem da empresa.
- **Atraso no atendimento de pedidos online (100 pontos):** Afeta diretamente a satisfação e confiança dos clientes, podendo prejudicar a estratégia de vendas online.
- **Falta de controle de inventário e atendimento ao cliente insatisfatório (80 pontos cada):** A falta de controle no inventário leva a perdas financeiras, e o atendimento ruim impacta diretamente a fidelidade e satisfação dos clientes

Github

<https://github.com/lemebiel/Click-Flow>

Youtube

<https://youtu.be/-7a19gsZOTM?feature=shared>