Nome da Empresa: Evonin

Ramo da Empresa: Varejo

Link Github: [clique aqui](https://github.com/SRJeanAguiar/planejamento-Informatica-projeto)

Equipe Responsável:

Jean Lucas Oliveira de Aguiar RA: 2222101720  
Vinicius Meneses da Silva RA: 2222103940  
Eric Zonato lima RA: 2222103734  
Leonardo Nicolas Barbosa RA: 2222104964  
Anderson Fernandes Filho RA: 2222102272

Curso: Análise e desenvolvimento de sistemas

Turma: 40

Semestre: 5

Ano: 2024

Sumário

[1. Escopo do projeto: 2](#_Toc167644214)

[2. Serviços Oferecidos: 3](#_Toc167644215)

[3. Estruturação Interna da Empresa 4](#_Toc167644216)

[3.1 Aprendizado de máquina 4](#_Toc167644217)

[3.1.1 - Entrega 1: Exploração de Dados e Pré-processamento 4](#_Toc167644218)

[3.1.2 - Entrega 2: Implementação de Modelos de Aprendizado de Máquina 5](#_Toc167644219)

[3.1.3 - Entrega 3: Otimização e Validação do Modelo 5](#_Toc167644220)

[3.2 Ciência de Dados (Python e Estatística): 8](#_Toc167644221)

[3.2.1 - Entrega 1: Análise Descritiva dos Dados 8](#_Toc167644222)

[3.2.2 - Entrega 2: Modelagem Estatística 8](#_Toc167644223)

[3.3 Modelagem de dados 12](#_Toc167644224)

[3.3.1 - Entrega 1: Modelagem Conceitual 12](#_Toc167644225)

[3.3.2 - Entrega 2: Modelagem Lógica e Normalização 12](#_Toc167644226)

[3.3.3 - Entrega 3: Entregar Dicionário de Dados uma simulação de cadastro: 14](#_Toc167644227)

[3.4 Rede de computadores 19](#_Toc167644228)

[3.4.1 - Entrega 1: Montar a planta baixa de Rede da Empresa 19](#_Toc167644229)

[3.4.2 - Entrega 2: Configuração de IP de todos os equipamentos 20](#_Toc167644230)

[3.5 Segurança da Informação 21](#_Toc167644231)

[3.5.1 - Entrega 1: Análise de Riscos 21](#_Toc167644232)

[3.5.2 - Entrega 2: Implementação de Medidas de Segurança 25](#_Toc167644233)

# Escopo do projeto:

O presente documento detalha o escopo do projeto proposto pela empresa Evonin.

A Evnonin é uma empresa de varejo especializada na venda de produtos tecnológicos através da internet. a Evnonin destaca-se pela sua interface de usuário intuitiva e navegação facilitada. Nosso propósito é simplificar o processo de aquisição de produtos tecnológicos, permitindo que os clientes encontrem e comprem os itens desejados de maneira eficiente e sem complicações.

# Serviços Oferecidos:

* Loja Online Intuitiva
* Entrega Rápida e Segura
* Suporte ao Cliente
* Programas de Fidelidade e Promoções
* Serviços de Pagamento Flexíveis
* Consultoria e Recomendações Personalizadas
* Política de Devolução e Troca Fácil

# Estruturação Interna da Empresa

# Aprendizado de máquina

Para acessar o repositório do GitHub [clique aqui](https://github.com/SRJeanAguiar/planejamento-Informatica-projeto/tree/ea915d680967745739da0ee7e392c522c068b7fe/Aprendizado%20de%20M%C3%A1quina)

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

Coleta de dados relevantes para o negócio proposto pela empresa:

import pandas as pd  
from sklearn.metrics import confusion\_matrix, accuracy\_score, recall\_score  
from sklearn.model\_selection import train\_test\_split, GridSearchCV, cross\_val\_score  
from sklearn.ensemble import RandomForestClassifier  
  
# Carregar dados de cada planilha em um DataFrame  
produtos\_df = pd.read\_excel('produtos.xlsx')  
clientes\_df = pd.read\_excel('clientes.xlsx')  
pedidos\_df = pd.read\_excel('pedidos.xlsx')  
itens\_pedido\_df = pd.read\_excel('itens\_pedido.xlsx')  
estoque\_df = pd.read\_excel('estoque.xlsx')  
categorias\_df = pd.read\_excel('categorias.xlsx')  
fornecedores\_df = pd.read\_excel('fornecedores.xlsx')

Limpeza e pré-processamento dos dados:

# Tratar valores ausentes, outliers e dados inconsistentes  
produtos\_df = produtos\_df.dropna() # Remover linhas com valores ausentes  
  
# Confirma a remoção de linhas com valores ausentes  
print("Produtos após remover valores ausentes:")  
print(produtos\_df)  
  
# Padroniza formatos e unidades  
produtos\_df['preço'] = produtos\_df['preço'].astype(float) #

# Confirma a conversão de formato  
print("\nFormato dos preços após conversão:")  
print(produtos\_df['preço'].dtype)

Verificar a matriz confusão:

# Classificação dos produtos com base no preço  
produtos\_df['classificação'] = produtos\_df['preço'].apply(lambda x: 1 if x > 50 else 0)  
  
# Valores verdadeiros (preços reais) e valores previstos (classificação baseada no preço)  
y\_true = produtos\_df['classificação']  
y\_pred = produtos\_df['classificação']  
  
# Calcula a matriz de confusão  
matriz\_confusao = confusion\_matrix(y\_true, y\_pred)  
  
# Exibe a matriz de confusão  
print("Matriz de Confusão:")  
print(matriz\_confusao)

## 3.1.2 - Entrega 2: Implementação de Modelos de Aprendizado de Máquina

Escolha de algoritmos de ML adequados ao problema:

# Dividir os dados em conjunto de treinamento e teste  
X = produtos\_df[['preço']] # Features  
y = produtos\_df['classificação'] # Target variable  
  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
# Escolher algoritmo de classificação  
modelo = RandomForestClassifier()

Desenvolva e treine os modelos selecionados.

# Treinar o modelo  
modelo.fit(X\_train, y\_train)

Avaliação da performance dos modelos com métricas apropriadas:

# Avaliar a precisão do modelo  
precisao = accuracy\_score(y\_test, y\_pred)  
print("Precisão do modelo:", precisao)  
  
# Identifica corretamente todos os exemplos positivos.  
recall = recall\_score(y\_test, y\_pred)  
print("Recall do modelo:", recall)

## 3.1.3 - Entrega 3: Otimização e Validação do Modelo

Otimização dos hiperparâmetros dos modelos para melhorar a performance:

# Definir os parâmetros para otimização  
parametros = {'n\_estimators': [50, 100, 150], 'max\_depth': [None, 10, 20]}  
  
# Criar o objeto GridSearchCV  
grid\_search = GridSearchCV(RandomForestClassifier(), parametros, cv=3)  
  
# Executar a busca em grade  
grid\_search.fit(X\_train, y\_train)  
  
# Melhores hiperparâmetros encontrados  
print("Melhores hiperparâmetros encontrados:")  
print(grid\_search.best\_params\_)  
  
# Avaliar o modelo com os melhores hiperparâmetros  
melhor\_modelo = grid\_search.best\_estimator\_  
y\_pred = melhor\_modelo.predict(X\_test)

Validação cruzada para verificar a robustez do modelo:

# Realizar validação cruzada  
pontuacoes = cross\_val\_score(melhor\_modelo, X, y, cv=3)  
  
# Exibir as pontuações de validação cruzada  
print("Pontuações de validação cruzada:", pontuacoes)  
  
# Calcular a média das pontuações  
media\_pontuacoes = pontuacoes.mean()  
print("Média das pontuações de validação cruzada:", media\_pontuacoes)

Elabore documentação detalhada sobre o processo, parâmetros e resultados.

Etapas do Processo:

1. Exploração de Dados e Pré-processamento:

Coleta de Dados: Os dados foram coletados de várias fontes, incluindo planilhas de produtos, clientes, pedidos, itens de pedido, estoque, categorias e fornecedores.

Limpeza e Pré-processamento: Foram identificados e removidos valores ausentes nas planilhas de produtos.

A coluna de preço foi convertida para o tipo float para padronizar o formato dos dados.

2. Implementação de Modelos de Aprendizado de Máquina:

Escolha de Algoritmos:

Optou-se por utilizar o algoritmo RandomForestClassifier devido à sua capacidade de lidar com problemas de classificação e sua robustez em relação a overfitting.

Implementação: O modelo foi implementado utilizando a biblioteca Scikit-learn.

Os dados de treinamento foram alimentados no modelo RandomForestClassifier.

3. Otimização e Validação do Modelo:

Otimização de Hiperparâmetros: Foi realizada uma busca em grade utilizando GridSearchCV para encontrar os melhores hiperparâmetros para o modelo.

Os hiperparâmetros ajustados incluem 'n\_estimators' (número de árvores na floresta) e 'max\_depth' (profundidade máxima das árvores).

Validação Cruzada: A validação cruzada foi realizada utilizando a função cross\_val\_score com 3 divisões.

Os resultados obtidos indicam a precisão do modelo em diferentes conjuntos de dados.

Parâmetros do Modelo:

Os melhores hiperparâmetros encontrados foram 'n\_estimators': 50 e 'max\_depth': None.

Outros parâmetros relevantes para o modelo incluem o critério de divisão, o número máximo de features consideradas em cada divisão, etc.

Métricas de Avaliação: As métricas utilizadas para avaliar o desempenho do modelo incluem precisão e recall.

A precisão mede a proporção de verdadeiros positivos em relação a todos os exemplos classificados como positivos.

O recall mede a proporção de verdadeiros positivos em relação a todos os exemplos positivos reais.

Resultados:

O modelo apresentou uma precisão de 1.0 e recall de 1.0, indicando um desempenho perfeito nos dados de teste.

As pontuações de validação cruzada também mostraram uma precisão consistente de 1.0 em diferentes conjuntos de dados, indicando a robustez do modelo.

Esta documentação fornece uma visão abrangente do processo de construção e treinamento do modelo, bem como dos resultados obtidos, permitindo uma compreensão clara do modelo desenvolvido e sua capacidade de classificar produtos com base no preço.

# Ciência de Dados (Python e Estatística):

Para acessar o repositório do GitHub [clique aqui](https://github.com/SRJeanAguiar/planejamento-Informatica-projeto/tree/2374e665f2eee0c72d4bad175c41b51044ffd0ea/Ci%C3%AAncia%20de%20Dados)

## 3.2.1 - Entrega 1: Análise Descritiva dos Dados

Utilização de técnicas estatísticas básicas para descrever os dados:

#Estatísticas descritivas  
media = produtos\_df['preço'].mean()  
mediana = produtos\_df['preço'].median()  
desvio\_padrao = produtos\_df['preço'].std()  
  
print(f'Média: {media}, Mediana: {mediana}, Desvio Padrão: {desvio\_padrao}')

Visualização de dados utilizando bibliotecas como Matplotlib e Seaborn:

# Gráfico de dispersão entre preço e quantidade de itens pedidos  
plt.figure(figsize=(10, 6))  
sns.scatterplot(x='preço\_unitário', y='quantidade', data=itens\_pedido\_df)  
plt.title('Relação entre Preço e Quantidade de Itens Pedidos')  
plt.xlabel('Preço Unitario')  
plt.ylabel('Quantidade')  
plt.show()

Identificação de padrões e tendências nos dados:

# Identificação de padrões nos dados  
# Exemplo: Contagem de produtos por categoria  
tendencia\_categoria = produtos\_df['nome'].value\_counts()  
print('Contagem de Produtos por Categoria:')  
print(tendencia\_categoria)

## 3.2.2 - Entrega 2: Modelagem Estatística

Aplicação de técnicas estatísticas avançadas para modelagem dos dados:

# Calculando o preço total do pedido (variável dependente)  
itens\_pedido\_df['preco\_total'] = itens\_pedido\_df['quantidade'] \* itens\_pedido\_df['preço\_unitário']  
  
# Separando as variáveis independentes (X) e a variável dependente (y)  
X = itens\_pedido\_df[['quantidade', 'preço\_unitário']] # Variáveis independentes  
y = itens\_pedido\_df['preco\_total'] # Variável dependente  
  
# Adicionando uma constante (intercepto) ao modelo  
X = sm.add\_constant(X)  
  
# Ajustando o modelo de regressão linear  
modelo = sm.OLS(y, X)  
resultados = modelo.fit()  
  
# Exibindo um resumo dos resultados da regressão  
print(resultados.summary())

Uso de ferramentas como regressão linear, classificação etc.:

# Calculando o preço total do pedido (variável dependente)  
itens\_pedido\_df['preco\_total'] = itens\_pedido\_df['quantidade'] \* itens\_pedido\_df['preço\_unitário']  
  
# Separando as variáveis independentes (X) e a variável dependente (y)  
X = itens\_pedido\_df[['quantidade', 'preço\_unitário']] # Variáveis independentes  
y = itens\_pedido\_df['preco\_total'] # Variável dependente  
  
# Adicionando uma constante (intercepto) ao modelo  
X = sm.add\_constant(X)  
  
# Ajustando o modelo de regressão linear  
modelo = sm.OLS(y, X)  
resultados = modelo.fit()  
  
# Exibindo um resumo dos resultados da regressão  
print(resultados.summary())  
  
# Inicializar o modelo de regressão linear  
modelo = LinearRegression()

Avaliação da adequação dos modelos estatísticos aos dados:

# Calculando o preço total do pedido (variável dependente)  
itens\_pedido\_df['preco\_total'] = itens\_pedido\_df['quantidade'] \* itens\_pedido\_df['preço\_unitário']  
  
# Separando as variáveis independentes (X) e a variável dependente (y)  
X = itens\_pedido\_df[['quantidade', 'preço\_unitário']] # Variáveis independentes  
y = itens\_pedido\_df['preco\_total'] # Variável dependente  
  
# Adicionando uma constante (intercepto) ao modelo  
X = sm.add\_constant(X)  
  
# Ajustando o modelo de regressão linear  
modelo = sm.OLS(y, X)  
resultados = modelo.fit()  
  
# Exibindo um resumo dos resultados da regressão  
print(resultados.summary())  
  
# Inicializar o modelo de regressão linear  
modelo = LinearRegression()  
  
# Avaliar modelo  
R2 = resultados.rsquared  
print(f'R-squared: {R2}')

Implementação de modelos preditivos utilizando Python:

# Definir as características dos novos itens do pedido  
X\_treino = itens\_pedido\_df[['quantidade', 'preço\_unitário']]  
y\_treino = itens\_pedido\_df['id\_produto'] # Suponha que 'id\_produto' seja o alvo do modelo  
  
# Criar e treinar o modelo de regressão linear  
modelo = LinearRegression()  
modelo.fit(X\_treino, y\_treino)  
  
# Supondo que você já tenha treinado o modelo com os dados originais e agora quer fazer previsões para novos itens do pedido  
# Definir as características dos novos itens do pedido  
X\_novos\_dados = itens\_pedido\_df[['quantidade', 'preço\_unitário']]  
  
# Fazer previsões para os novos dados  
previsoes = modelo.predict(X\_novos\_dados)  
  
# Exibir as previsões  
print("Previsões para os novos itens do pedido:")  
print(previsoes)

Avaliação da performance dos modelos preditivos:

# Avaliar desempenho  
y\_verdadeiro = itens\_pedido\_df['id\_produto'] # Suponha que 'id\_produto' seja o valor verdadeiro  
  
# Calcular o erro médio quadrático  
erro\_medio\_quadratico = mean\_squared\_error(y\_verdadeiro, previsoes)  
print(f'Erro Médio Quadrático: {erro\_medio\_quadratico}')

Comparação entre diferentes abordagens de análise preditiva:

# Preparar os dados para os modelos  
X1 = itens\_pedido\_df[['quantidade', 'preço\_unitário']]  
X2 = itens\_pedido\_df[['quantidade', 'id\_produto']]  
y = itens\_pedido\_df['id']  
  
# Instanciar os modelos de regressão linear  
modelo1 = LinearRegression()  
modelo2 = LinearRegression()  
  
# Ajustar os modelos aos dados  
modelo1.fit(X1, y)  
modelo2.fit(X2, y)  
  
# Fazer previsões  
previsoes1 = modelo1.predict(X1)  
previsoes2 = modelo2.predict(X2)  
  
# Avaliar os modelos  
erro\_medio\_quadratico1 = mean\_squared\_error(y, previsoes1)  
erro\_medio\_quadratico2 = mean\_squared\_error(y, previsoes2)  
  
# Comparar modelos  
if erro\_medio\_quadratico1 < erro\_medio\_quadratico2:  
 print('Modelo 1 é melhor.')  
else:  
 print('Modelo 2 é melhor.')

# 3.3 Modelagem de dados

Para acessar o repositório do GitHub [clique aqui](https://github.com/SRJeanAguiar/planejamento-Informatica-projeto/tree/fde82a196e9fb7daf04e6a8f7230335a09ee709d/Modelagem%20de%20Dados)

## 3.3.1 - Entrega 1: Modelagem Conceitual

Criação de um diagrama de entidade-relacionamento (ER) para representar os dados da empresa:

Uma imagem contendo Diagrama

Descrição gerada automaticamente

Identificação das entidades, atributos e relacionamentos relevantes:

Categorias

Clientes

Estoque

Fornecedores

Produtos

Pedidos

Itens\_Pedido

## 3.3.2 - Entrega 2: Modelagem Lógica e Normalização

Transformação do modelo conceitual em um modelo lógico:

Categorias:

id (chave primária)

nome

Clientes:

id (chave primária)

nome

email

telefone

endereço

Estoque:

id (chave primária, FK para Produtos)

id\_produto

quantidade

Fornecedores:

id (chave primária)

nome

contato

Produtos:

id (chave primária)

nome

descriçao

preço

id\_categoria (chave estrangeira para Categorias)

Pedidos:

id (chave primária)

id\_cliente (chave estrangeira para Clientes)

data\_pedido

total

Itens\_pedido:

id (chave primária)

id\_pedido (chave estrangeira para Pedidos)

id\_produto (chave estrangeira para Produtos)

quantidade

preço\_unitario

Aplicação de técnicas de normalização para garantir a integridade dos dados:

Categorias, Clientes, Fornecedores e Produtos estão na Primeira Forma Normal

Estoque, Pedidos e Itens\_Pedido estão na Segunda Forma Normal

Itens\_pedido está na Terceira Forma Formal

Interface gráfica do usuário, Aplicativo, Tabela, Excel

Descrição gerada automaticamente

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

Mostrar as tabelas do Banco de Dados:

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Interface gráfica do usuário, Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Fazer o registro nas tabelas do Banco de Dados (Simulação de Registro do Banco):

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

Tabela

Descrição gerada automaticamente

# 3.4 Rede de computadores

Para acessar o repositório do GitHub [clique aqui](https://github.com/SRJeanAguiar/planejamento-Informatica-projeto/tree/20926e2f29be3c61378f95bef733fd40f541f7b5/Redes%20de%20Computadores)

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

Definir os departamentos:

Administração, Recursos Humanos, Desenvolvimento, Operações, Marketing

Definir os equipamentos que serão utilizados em cada departamento:

Administração:

* Servidor para armazenamento seguro de dados
* Computadores para analistas e contadores
* Impressoras para documentos e relatórios
* Calculadoras financeiras para operações contábeis
* Software de contabilidade e gestão financeira

Recursos Humanos:

* Servidor para armazenamento seguro de dados
* Computadores para gerentes e especialistas de RH
* Impressoras para contratos e documentos
* Scanner para digitalização de documentos de RH
* Software de gestão de recursos humanos

Desenvolvimento:

* Servidores para hospedagem de aplicações e armazenamento de dados
* Computadores para desenvolvedores e técnicos de suporte
* Software de gestão de TI e segurança (ex.: ferramentas de monitoramento, antivírus, sistemas de backup)

Operações:

* Computadores para representantes de vendas e gerentes
* Smartphones e tablets para mobilidade e acesso a CRM
* Impressoras para contratos e materiais de marketing
* Software de gestão de relacionamento com clientes (CRM)

Marketing:

* Computadores
* Software de Design Gráfico
* Câmeras e Equipamentos de Vídeo
* Software de Análise de Dados
* Impressoras

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

Definir a classe de Rede:

Classe de Rede: Classe C (192.168.0.0)

Definir o padrão de rede de cada departamento:

Padrão de Rede:

Administração: 192.168.0.1 - 192.168.0.50

Recursos Humanos: 192.168.0.51 - 192.168.0.100

Desenvolvimento: 192.168.0.101 - 192.168.0.150

Operações: 192.168.0.102 - 192.168.0.151

Marketing: 192.168.0.103 - 192.168.0.152

# 3.5 Segurança da Informação

Para acessar o repositório do GitHub [clique aqui](https://github.com/SRJeanAguiar/planejamento-Informatica-projeto/tree/20926e2f29be3c61378f95bef733fd40f541f7b5/Seguran%C3%A7a%20da%20Informa%C3%A7%C3%A3o)

## 3.5.1 - Entrega 1: Análise de Riscos

Identificação e avaliação dos riscos de segurança para a empresa:

Com base nos departamentos listados segue a lista de 20 possíveis ameaças e vulnerabilidades:

1.Phishing

Vulnerabilidade: Falta de treinamento em segurança para reconhecer emails fraudulentos.

2. Malware

Vulnerabilidade: Uso de software desatualizado ou sem antivírus adequado.

3. Acesso Não Autorizado

Vulnerabilidade: Controle inadequado de acesso e senhas fracas.

4. Ransomware

Vulnerabilidade: Falta de backups regulares e sistemas desatualizados.

5. Erro Humano

Vulnerabilidade: Ausência de políticas e procedimentos claros e treinamento insuficiente.

6. Sabotagem

Vulnerabilidade: Falta de monitoramento de funcionários e controle de acesso físico.

7. Violação de Dados

Vulnerabilidade: Proteção insuficiente de dados sensíveis e falta de criptografia.

8. Falhas de Backup

Vulnerabilidade: Processos de backup inadequados e testes de recuperação inexistentes.

9. Vulnerabilidades de Software

Vulnerabilidade: Falta de atualização regular e patches de segurança.

10. DDoS (Negação de Serviço)

Vulnerabilidade: Infraestrutura de rede sem defesas adequadas contra ataques de volume.

11. Exploração de Zero-Day

Vulnerabilidade: Dependência de software sem patches ou atualizações para vulnerabilidades recém-descobertas.

12. Man-in-the-Middle (MitM)

Vulnerabilidade: Comunicações não criptografadas e redes inseguras.

13. Falhas de Criptografia

Vulnerabilidade: Implementação inadequada ou falta de criptografia em dados sensíveis.

14. Roubo de Equipamentos

Vulnerabilidade: Falta de medidas de segurança física e monitoramento.

15. Configurações Incorretas

Vulnerabilidade: Falta de revisão e auditoria regular das configurações de sistemas e redes.

16. Insider Threats

Vulnerabilidade: Monitoramento inadequado e falta de controle sobre as atividades dos funcionários.

17. Exploração de Redes Sociais

Vulnerabilidade: Falta de políticas de uso de redes sociais e controle sobre contas oficiais.

18. Violação de Direitos Autorais

Vulnerabilidade: Uso não autorizado de materiais protegidos por direitos autorais sem verificação.

19. Roubo de Propriedade Intelectual

Vulnerabilidade: Proteção insuficiente de documentos e projetos sensíveis.

20. Interrupção de Fornecimento

Vulnerabilidade: Dependência excessiva de fornecedores únicos sem planos de contingência.

Análise de vulnerabilidades e ameaças potenciais:

1. Phishing

Impacto: Médio

Probabilidade: Alta

2. Malware

Impacto: Alto

Probabilidade: Alta

3. Acesso Não Autorizado

Impacto: Alto

Probabilidade: Média

4. Ransomware

Impacto: Alto

Probabilidade: Média

5. Erro Humano

Impacto: Médio

Probabilidade: Alta

6. Sabotagem

Impacto: Alto

Probabilidade: Baixa

7. Violação de Dados

Impacto: Alto

Probabilidade: Média

8. Falhas de Backup

Impacto: Alto

Probabilidade: Média

9. Vulnerabilidades de Software

Impacto: Médio

Probabilidade: Alta

10. DDoS (Negação de Serviço)

Impacto: Médio

Probabilidade: Média

11. Exploração de Zero-Day

Impacto: Alto

Probabilidade: Baixa

12. Man-in-the-Middle (MitM)

Impacto: Alto

Probabilidade: Baixa

13. Falhas de Criptografia

Impacto: Alto

Probabilidade: Média

14. Roubo de Equipamentos

Impacto: Médio

Probabilidade: Média

15. Configurações Incorretas

Impacto: Médio

Probabilidade: Alta

16. Insider Threats

Impacto: Alto

Probabilidade: Baixa

17. Exploração de Redes Sociais

Impacto: Médio

Probabilidade: Média

18. Violação de Direitos Autorais

Impacto: Médio

Probabilidade: Baixa

19. Roubo de Propriedade Intelectual

Impacto: Alto

Probabilidade: Baixa

20. Interrupção de Fornecimento

Impacto: Alto

Probabilidade: Baixa

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

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

Política de Acesso a Dados Financeiros:

Acesso aos dados financeiros restrito a membros do departamento financeiro. Somente funcionários autorizados com funções específicas têm permissão para visualizar ou modificar esses dados.

Política de Acesso Baseado em Funções (RBAC):

Permissões de acesso são concedidas com base na função ou cargo do funcionário. Apenas as funções necessárias para o desempenho do trabalho têm acesso aos dados e sistemas relevantes.

Política de Senhas Fortes:

Exige que todas as senhas tenham no mínimo 12 caracteres, incluindo letras maiúsculas, minúsculas, números e símbolos. Senhas devem ser alteradas a cada 90 dias.

Política de Autenticação Multifator (MFA):

Implementação de autenticação multifator para todas as contas de usuário, especialmente para acesso a sistemas críticos e dados sensíveis.

Política de Revisão de Acesso:

Revisão regular dos direitos de acesso dos funcionários. Acessos desnecessários ou não utilizados são removidos periodicamente.

Política de Acesso Remoto:

Acesso remoto aos sistemas da empresa só é permitido através de VPN segura e autenticada. Acesso remoto deve ser monitorado e registrado.

Política de Controle de Acesso Físico:

Acesso físico a áreas sensíveis, como salas de servidores e arquivos confidenciais, é restrito a pessoal autorizado e monitorado por sistemas de segurança.

Política de Acesso a Dados de RH:

Dados de recursos humanos, como informações pessoais dos funcionários, são acessíveis apenas por membros do departamento de RH e outros funcionários autorizados.

Política de Controle de Acesso Temporário:

Acesso temporário concedido a contratados e visitantes deve ser limitado no tempo e escopo, com supervisão direta e expiração automática.

Política de Registro e Monitoramento de Acessos:

Todos os acessos a sistemas críticos e dados sensíveis devem ser registrados e monitorados. Logs de acesso devem ser revisados regularmente para detectar atividades suspeitas.

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

Sistema de Detecção de Intrusão (IDS):

Configuração de Alertas: Configurar alertas em caso de atividades suspeitas na rede, como tentativas de acesso não autorizado ou varreduras de porta.

Sistema de Prevenção de Intrusão (IPS):

Bloqueio Automático: Implementar um IPS que bloqueie automaticamente tráfego malicioso identificado, prevenindo ataques antes que eles possam causar danos.

Análise de Tráfego em Tempo Real: Monitorar o tráfego de rede em tempo real para identificar padrões de comportamento anômalo que possam indicar um ataque.

Regras de Firewall: Configurar regras de firewall para bloquear tráfego não autorizado e permitir apenas comunicações essenciais e seguras.

Divisão de Rede: Dividir a rede em segmentos menores para limitar a propagação de ataques e isolar sistemas críticos de acesso geral.

Permissões Restritas: Implementar controle de acesso baseado em funções para garantir que usuários tenham apenas as permissões necessárias para suas funções.

Revisão Regular: Analisar logs de sistemas e redes regularmente para identificar e responder rapidamente a atividades suspeitas.

Verificação Adicional: Implementar autenticação multifator para adicionar uma camada extra de segurança no acesso a sistemas críticos.

Manutenção Regular: Garantir que todos os sistemas e softwares estejam atualizados com os patches de segurança mais recentes para proteger contra vulnerabilidades conhecidas.

Testes Periódicos: Realizar testes de penetração e avaliações de vulnerabilidade regularmente para identificar e corrigir pontos fracos na segurança.