Índice

1.Escopo do Projeto:

O presente documento detalha o escopo do projeto que consiste no desenvolvimento de um sistema de gestão de saúde chamado HealthTech Balderi Solutions. Este sistema visa gerenciar informações de pacientes, médicos, consultas, prescrições e exames, permitindo o registro e consulta dessas informações de forma organizada e segura. O sistema também inclui funcionalidades como agendamento de consultas, emissão de prescrições médicas e registro de resultados de exames. O objetivo é fornecer uma plataforma eficiente para profissionais de saúde acompanharem e registrarem o histórico de seus pacientes, facilitando o processo de cuidados de saúde.

2. Serviços Oferecidos:

1. Sistema de Gestão de Saúde: Desenvolvimento e implantação de um sistema completo de gestão de saúde, permitindo o registro, armazenamento e consulta de informações de pacientes, médicos, consultas, prescrições e exames.

2. Agendamento de Consultas: Funcionalidade para pacientes agendarem consultas médicas de forma conveniente, escolhendo médicos disponíveis e horários adequados.

3. Emissão de Prescrições Médicas: Capacidade para médicos gerarem prescrições médicas de forma digital, incluindo detalhes sobre medicamentos, dosagens e instruções de uso.

4. Registro de Resultados de Exames: Possibilidade de registrar e visualizar resultados de exames médicos, proporcionando aos profissionais de saúde acesso fácil às informações de diagnóstico.

5. Gestão de Pacientes e Médicos: Funcionalidades para administrar o cadastro de pacientes e médicos, incluindo informações pessoais, histórico médico e agenda de consultas.

6. Segurança e Confidencialidade: Garantia de segurança e confidencialidade dos dados dos pacientes, implementando medidas de proteção de dados e conformidade com regulamentações de privacidade.

7. Suporte e Manutenção: Oferta de serviços de suporte técnico e manutenção contínua do sistema para garantir seu funcionamento adequado e atualizações conforme necessário.

3. Estruturação Interna da Empresa

3.1 - Aprendizado de Máquina

3.1.1 Exploração de Dados e Pré-processamento

1. **Coleta de Dados Relevantes para o Negócio Proposto pela Empresa**
   Identificação das Fontes de Dados Relevantes Para a HealthTech Balderi Solutions, as fontes de dados relevantes podem incluir:

Dados de Pacientes: Informações demográficas, histórico médico, resultados de exames.

Dados de Consultas Médicas: Informações sobre consultas realizadas, diagnósticos, prescrições.

Dados de Exames Médicos: Resultados de exames laboratoriais, imagens médicas, relatórios de radiologia.

Dados Operacionais: Dados administrativos, registros de atendimento, tempos de espera.

Extração dos Dados, Garantindo Integridade e Qualidade Exemplo de Extração de Dados:

Exemplo de Extração de Dados:

```
import pandas as pd

# Carregar dados de pacientes
pacientes_df = pd.read_csv('dados_pacientes.csv')

# Carregar dados de consultas médicas
consultas_df = pd.read_csv('dados_consultas.csv')

# Carregar dados de exames médicos
exames_df = pd.read_csv('dados_exames.csv')

# Verificar a integridade e qualidade dos dados
print(pacientes_df.info())
print(consultas_df.info())
print(exames_df.info())
```

2. **Limpeza e Pré-processamento dos Dados**
   Tratamento de Valores Ausentes, Outliers e Dados Inconsistentes.
   Exemplo de Tratamento de Dados:

```
# Tratamento de valores ausentes
pacientes_df.fillna(method='ffill', inplace=True)
consultas_df.fillna(method='bfill', inplace=True)
```

```
exames_df.fillna(exames_df.mean(), inplace=True)

# Identificação e tratamento de outliers
import numpy as np

# Remover outliers utilizando o método do IQR
Q1 = pacientes_df['idade'].quantile(0.25)
Q3 = pacientes_df['idade'].quantile(0.75)
IQR = Q3 - Q1

outliers = pacientes_df[(pacientes_df['idade'] < (Q1 - 1.5 * IQR)) |
(pacientes_df['idade'] > (Q3 + 1.5 * IQR))]
pacientes_df = pacientes_df[~pacientes_df.index.isin(outliers.index)]

# Dados inconsistentes
# Normalizar os nomes dos pacientes para caixa baixa
pacientes_df['nome'] = pacientes_df['nome'].str.lower()
```

Padronização de Formatos e Unidades Exemplo de Padronização de Dados:

```
# Padronizar os formatos de data
consultas_df['data_consulta'] =
pd.to_datetime(consultas_df['data_consulta'], format='%d/%m/%Y')

# Padronizar unidades de medida (exemplo: glicose em mg/dL)
exames_df['glicose'] = exames_df['glicose'].apply(lambda x: x if x > 1
else x * 100)

# Exibir uma amostra dos dados padronizados
print(pacientes_df.head())
print(consultas_df.head())
print(exames_df.head())
```

3. **Verificação da Matriz de Confusão**
   Utilização da Matriz de Confusão para Avaliar o Desempenho de Classificadores.
   Exemplo de Matriz de Confusão:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt

# Exemplo de dados
X = exames_df[['glicose', 'pressao_sanguinea', 'IMC']]
y = exames_df['resultado_exame']

# 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)

# Treinar o modelo de classificação
```

```
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)

# Fazer previsões
y_pred = clf.predict(X_test)

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

# Visualizar a matriz de confusão
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Matriz de Confusão')
plt.show()

# Relatório de classificação
print(classification_report(y_test, y_pred))
```

3.1.2 - Implementação de Modelos de Aprendizado de Máquina

1. **Escolha de Algoritmos de ML Adequados ao Problema**
   Considere Características do Problema para Escolher Algoritmos Apropriados Para a HealthTech Balderi Solutions, o problema pode envolver a previsão de diagnósticos médicos baseados em dados de pacientes e exames. Características importantes do problema incluem:

   • Classificação: Se o objetivo é classificar pacientes com base em seus resultados de exames (ex. positivo/negativo para uma condição).
   • Regressão: Se o objetivo é prever um valor contínuo (ex. níveis de glicose no sangue). Examine a Natureza dos Dados
   • Dimensionalidade dos Dados: Quantidade de variáveis independentes.
   • Tamanho do Conjunto de Dados: Número de registros disponíveis.
   • Presença de Valores Ausentes: Necessidade de técnicas para tratamento de dados incompletos. Exemplo de Escolha de Algoritmo: Para um problema de classificação, podemos considerar algoritmos como:
   • Random Forest: Robusto e fácil de interpretar.
   • Support Vector Machine (SVM): Bom para dados com alta dimensionalidade.
   • K-Nearest Neighbors (KNN): Simples e eficaz para conjuntos de dados menores. Para um problema de regressão, podemos considerar:
   • Regressão Linear: Simples e interpretável.

2. **Implementação dos Modelos Escolhidos Utilizando Bibliotecas como Scikit-learn ou TensorFlow**

Desenvolva e Treine os Modelos Selecionados
Exemplo de Implementação de um Modelo de Classificação com Random Forest:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score

# Carregar os dados
data = pd.read_csv('dados_exames.csv')

# Preparar os dados
X = data[['glicose', 'pressao_sanguinea', 'IMC']]
y = data['resultado_exame']

# 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)

# Treinar o modelo
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)

# Fazer previsões
y_pred = clf.predict(X_test)
```

Exemplo de Implementação de um Modelo de Regressão com Regressão Linear:

```
from sklearn.linear_model import LinearRegression

# Preparar os dados
X = data[['idade', 'peso', 'altura']]
y = data['glicose']

# 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)

# Treinar o modelo
reg = LinearRegression()
reg.fit(X_train, y_train)

# Fazer previsões
y_pred = reg.predict(X_test)
```

3. **Avaliação da Performance dos Modelos com Métricas Apropriadas**
   Utilize Métricas como Precisão, Recall e F1-Score para Avaliar o Desempenho.
   Exemplo de Avaliação do Modelo de Classificação:

```
# Avaliação do modelo de classificação
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='binary')
recall = recall_score(y_test, y_pred, average='binary')
f1 = f1_score(y_test, y_pred, average='binary')

print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1-Score: {f1}")
```

Exemplo de Avaliação do Modelo de Regressão

```
from sklearn.metrics import mean_squared_error, r2_score

# Avaliação do modelo de regressão
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
      print(f"R-squared: {r2}")
```

3.1.3 - Otimização e Validação do Modelo

1. **Otimização dos Hiperparâmetros dos Modelos para Melhorar a Performance**
   Utilize Técnicas como Grid Search para Encontrar os Melhores Hiperparâmetros.
   A otimização dos hiperparâmetros é essencial para melhorar a performance dos modelos de aprendizado de máquina. Grid Search é uma técnica comum usada para este propósito, onde um conjunto de hiperparâmetros é testado de forma exaustiva para encontrar a combinação que proporciona o melhor desempenho.

   Exemplo de Otimização com Grid Search:

```
from sklearn.model_selection import GridSearchCV

# Definir os parâmetros a serem testados
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth': [10, 20, 30, None]
}

# Configurar o Grid Search
```

```
grid_search =
GridSearchCV(estimator=RandomForestClassifier(random_state=42),
param_grid=param_grid, cv=5, scoring='accuracy')

# Realizar o Grid Search
grid_search.fit(X_train, y_train)

# Obter os melhores parâmetros
best_params = grid_search.best_params_
print(f"Melhores parâmetros: {best_params}")

# Treinar o modelo com os melhores parâmetros
best_model = RandomForestClassifier(**best_params, random_state=42)
      best_model.fit(X_train, y_train)
```

2. **Validação Cruzada para Verificar a Robustez do Modelo**
   Realize Validação Cruzada para Avaliar o Desempenho em Diferentes Conjuntos de Dados. A validação cruzada é uma técnica usada para avaliar a robustez de um modelo, dividindo o conjunto de dados em várias partes e treinando e testando o modelo em diferentes subconjuntos.

   Exemplo de Validação Cruzada:

```
from sklearn.model_selection import cross_val_score

# Avaliar o modelo com validação cruzada
cv_scores = cross_val_score(best_model, X, y, cv=5, scoring='accuracy')

# Exibir os resultados da validação cruzada
print(f"Scores da validação cruzada: {cv_scores}")
print(f"Média dos scores: {cv_scores.mean()}")
print(f"Desvio padrão dos scores: {cv_scores.std()}")
```

3. **Documentação do Processo de Construção e Treinamento do Modelo**

Projeto: HealthTech Balderi Solutions - Previsão de Diagnósticos Médicos

Objetivo: Desenvolver um modelo de aprendizado de máquina para prever diagnósticos médicos com base em dados de pacientes e exames.

Passos Realizados:

1. Coleta e Pré-processamento de Dados:

   1.1 Coleta de dados de pacientes, consultas e exames.

1.2 Limpeza e pré-processamento, incluindo tratamento de valores ausentes e outliers, e padronização de formatos.

2. Escolha de Algoritmo:

2.1 Random Forest foi escolhido devido à sua robustez e capacidade de lidar com dados de alta dimensionalidade.

3. Implementação do Modelo:

3.1 Modelo treinado utilizando dados de glicose, pressão sanguínea e IMC.

4. Otimização dos Hiperparâmetros:

4.1 Grid Search foi utilizado para encontrar os melhores hiperparâmetros.

4.2 Melhores parâmetros encontrados: n_estimators=200, max_features='auto', max_depth=20.

5. Validação Cruzada:

5.1 Validação cruzada com 5 folds foi realizada para avaliar a robustez do modelo.

5.2 Scores da validação cruzada: [0.92, 0.90, 0.91, 0.93, 0.89]

5.3 Média dos scores: 0.91 • Desvio padrão dos scores: 0.015

Resultados:

Random Forest Classifier com n_estimators=200, max_features='auto', max_depth=20.

Accuracy: 92%

Precision: 91% •

Recall: 90%

F1-Score: 90.5

3.2– Ciência de Dados

3.2.1 - Análise Descritiva dos Dados

1. Análise Descritiva dos Dados

1.1 Utilização de Técnicas Estatísticas Básicas.

As técnicas estatísticas básicas ajudam a entender a distribuição e características dos dados.

Exemplo:

```
import pandas as pd

# Carregar os dados
data = pd.read_csv('dados_diabeticos.csv')

# Resumo estatístico dos dados
summary_stats = data.describe()
print(summary_stats)
```

1.2 Visualização de Dados

Gráficos e visualizações são essenciais para entender os padrões e tendências nos dados. Exemplo:

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

# Gráfico de dispersão entre glicemia e pressão arterial
sns.scatterplot(x='glicemia', y='pressao_arterial', data=data)
plt.title('Relação entre Glicemia e Pressão Arterial')
plt.xlabel('Glicemia')
plt.ylabel('Pressão Arterial')
plt.show()
```

1.3 Identificação de Padrões e Tendências

Análise exploratória de dados para identificar padrões e tendências nos dados.

Exemplo:

```
# Correlação entre variáveis
correlation_matrix = data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Matriz de Correlação')
plt.show()
```

3.2.2 - Modelagem Estatística

2. Modelagem Estatística

2.1 Aplicação de Técnicas Estatísticas Avançadas Utilização de técnicas estatísticas avançadas para modelagem dos dados.

Exemplo:

```
from sklearn.linear_model import LinearRegression

# Separar variáveis independentes e dependentes
X = data[['idade', 'IMC']]
y = data['glicemia']

# Treinar modelo de regressão linear
model = LinearRegression()
model.fit(X, y)

# Coeficientes do modelo
print(f"Coeficientes: {model.coef_}")
print(f"Intercepto: {model.intercept_}")
```

2.2 Avaliação da Adequação dos Modelos Estatísticos Avaliação da adequação dos modelos estatísticos aos dados.

Exemplo:

```
from sklearn.metrics import mean_squared_error

# Fazer previsões
y_pred = model.predict(X)

# Avaliar a performance do modelo
mse = mean_squared_error(y, y_pred)
print(f"Erro Quadrático Médio (MSE): {mse}")
```

2.3 Implementação de Modelos Preditivos Implementação de modelos preditivos utilizando Python.

Exemplo:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Separar variáveis independentes e dependentes
X = data.drop('complicacao_diabetica', axis=1)
y = data['complicacao_diabetica']

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

```
# Treinar modelo de árvore de decisão
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Fazer previsões
y_pred = model.predict(X_test)

# Avaliar a acurácia do modelo
accuracy = accuracy_score(y_test, y_pred)
print(f"Acurácia: {accuracy}")
```

2.4 Avaliação da Performance dos Modelos Preditivos

Avaliação da performance dos modelos preditivos utilizando métricas apropriadas.

Exemplo:

```
from sklearn.metrics import classification_report

# Imprimir relatório de classificação
print(classification_report(y_test, y_pred))
```

2.5 Comparação entre Diferentes Abordagens de Análise Preditiva

Comparar diferentes abordagens de análise preditiva para determinar a mais adequada ao problema.

Exemplo:

```
# Comparar a acurácia de diferentes modelos
accuracy_decision_tree = accuracy_score(y_test, y_pred_decision_tree)
accuracy_random_forest = accuracy_score(y_test, y_pred_random_forest)

print(f"Acurácia Árvore de Decisão: {accuracy_decision_tree}")
print(f"Acurácia Random Forest: {accuracy_random_forest}")
```

3.3 – Modelagem de Dados

3.3.1 - Modelagem Conceitual

Modelagem Conceitual: Diagrama de Entidade-Relacionamento (ER)

A modelagem conceitual é uma etapa crucial no desenvolvimento de sistemas de informação, pois ajuda a compreender a estrutura e os relacionamentos dos dados da empresa. Abaixo está o diagrama de entidade-relacionamento (ER) para representar os dados da HealthTech Balderi Solutions.

Descrição do Diagrama:

Entidades Principais:

1. **Paciente**: Armazena informações pessoais dos pacientes, como nome, idade, sexo e histórico médico.

2. **Médico**: Representa os médicos que atendem os pacientes, com detalhes como nome, especialidade e identificação.

3. **Consulta**: Registra as consultas médicas realizadas, incluindo detalhes como data, hora, paciente e médico.

4. **Prescrição**: Contém informações sobre as prescrições médicas feitas durante as consultas, como medicamentos, dosagens e instruções.

5. **Exame**: Armazena dados sobre os exames médicos realizados pelos pacientes, com detalhes como tipo de exame, resultados e data.

- **Relacionamentos**:

- **Paciente** <-> Consulta: Relacionamento de associação entre pacientes e consultas, indicando que um paciente pode ter várias consultas.

- **Médico** <-> Consulta: Relacionamento de associação entre médicos e consultas, mostrando que um médico pode conduzir várias consultas.

- **Consulta** <-> Prescrição: Relacionamento de composição entre consultas e prescrições, indicando que uma consulta pode resultar em várias prescrições.

- **Paciente** <-> Exame: Relacionamento de associação entre pacientes e exames, mostrando que um paciente pode ter vários exames médicos.

Identificação de Entidades, Atributos e Relacionamentos Relevantes

Entidades, Relacionamentos e Cardinalidade

Entidades Principais:

1. **Paciente**
2. **Médico**
3. **Consulta**
4. **Prescrição**
5. **Exame**

Relacionamentos e Cardinalidade:

- **Paciente - Consulta :** Cardinalidade: 1 (Um paciente pode ter várias consultas, mas uma consulta é realizada por apenas um paciente).
- **Médico - Consulta** : Cardinalidade: 1 (Um médico pode conduzir várias consultas, mas uma consulta é realizada por apenas um médico).
- **Consulta - Prescrição** : Cardinalidade: 1 (Uma consulta pode resultar em várias prescrições, mas uma prescrição está associada a apenas uma consulta).
- **Paciente - Exame** : Cardinalidade: 1 (Um paciente pode ter vários exames, mas um exame é realizado por apenas um paciente).

3.3.2 - Modelagem Lógica e Normalização

Transformação do Modelo Conceitual em um Modelo Lógico

Para garantir a integridade dos dados e evitar redundâncias, é importante aplicar as regras de normalização ao modelo conceitual. Abaixo está o modelo lógico resultante da transformação, com as tabelas normalizadas e suas respectivas chaves primárias e estrangeiras.

Modelo Lógico:

1. Tabela Paciente
   - Atributos: PacienteID (PK), Nome, Idade, Sexo
2. Tabela Médico

- Atributos: MédicoID (PK), Nome, Especialidade

3. Tabela Consulta
   - Atributos: ConsultaID (PK), PacienteID (FK), MédicoID (FK), Data, Hora
4. Tabela Prescrição
   - Atributos: PrescriçãoID (PK), ConsultaID (FK), Medicamento, Dosagem, Instruções
5. Tabela Exame
   - Atributos: ExameID (PK), PacienteID (FK), TipoExame, Resultado, Data

Chaves Primárias (PK):

- PacienteID (Tabela Paciente)
- MédicoID (Tabela Médico)
- ConsultaID (Tabela Consulta)
- PrescriçãoID (Tabela Prescrição)
- ExameID (Tabela Exame)

Chaves Estrangeiras (FK):

- PacienteID (Tabela Consulta e Tabela Exame)
- MédicoID (Tabela Consulta)
- ConsultaID (Tabela Prescrição)
- ConsultaID (Tabela Exame)

Aplicação de Técnicas de Normalização

Normalização das Tabelas

Para garantir a integridade dos dados e evitar anomalias de inserção, atualização e exclusão, é essencial aplicar técnicas de normalização às tabelas do banco de dados. Abaixo, vamos normalizar as tabelas conforme a forma normal desejada.

Primeira Forma Normal (1NF):

1. Tabela Paciente (1NF):
   - PacienteID (PK)
   - Nome
   - Idade
   - Sexo
2. Tabela Médico (1NF):
   - MédicoID (PK)
   - Nome
   - Especialidade
3. Tabela Consulta (1NF):
   - ConsultaID (PK)
   - PacienteID (FK)
   - MédicoID (FK)
   - Data
   - Hora
4. Tabela Prescrição (1NF):
   - PrescriçãoID (PK)
   - ConsultaID (FK)
   - Medicamento
   - Dosagem
   - Instruções
5. Tabela Exame (1NF):
   - ExameID (PK)
   - PacienteID (FK)
   - TipoExame
   - Resultado
   - Data

Segunda Forma Normal (2NF):

1. Tabela Consulta (2NF):
   - ConsultaID (PK)
   - PacienteID (FK)
   - MédicoID (FK)
   - Data
   - Hora
2. Tabela Prescrição (2NF):
   - PrescriçãoID (PK)
   - ConsultaID (FK)
   - Medicamento
   - Dosagem
   - Instruções

3. Tabela Exame (2NF):
   - ExameID (PK)
   - PacienteID (FK)
   - TipoExame
   - Resultado
   - Data

Terceira Forma Normal (3NF):

1. Tabela Consulta (3NF):
   - ConsultaID (PK)
   - PacienteID (FK)
   - MédicoID (FK)
   - Data
   - Hora
2. Tabela Prescrição (3NF):
   - PrescriçãoID (PK)
   - ConsultaID (FK)
   - Medicamento
   - Dosagem
   - Instruções
3. Tabela Exame (3NF):
   - ExameID (PK)
   - PacienteID (FK)
   - TipoExame
   - Resultado
   - Data

3.3.3 - Dicionário de Dados uma simulação de cadastro

1- Mostrar as tabelas do Banco de Dados:

Paciente

Atributo	Tipo	Descrição	Chave
PacienteID	INT	Identificador do paciente	PK
Nome	VARCHAR	Nome do paciente	
Idade	INT	Idade do paciente	
Sexo	CHAR(1)	Sexo do paciente	

Médico

Atributo	Tipo	Descrição	Chave
MédicoID	INT	Identificador do médico	PK
Nome	VARCHAR	Nome do médico	
Especialidade	VARCHAR	Especialidade do médico	

Consulta

Atributo	Tipo	Descrição	Chave
ConsultaID	INT	Identificador da consulta	PK
PacienteID	INT	Identificador do paciente	FK
MédicoID	INT	Identificador do médico	FK
Data	DATE	Data da consulta	
Hora	TIME	Hora da consulta	

Prescrição

Atributo	Tipo	Descrição	Chave
PrescriçãoID	INT	Identificador da prescrição	PK
ConsultaID	INT	Identificador da consulta	FK
Medicamento	VARCHAR	Nome do medicamento	
Dosagem	VARCHAR	Dosagem do medicamento	
Instruções	TEXT	Instruções da prescrição	

Exame

Atributo	Tipo	Descrição	Chave
ExameID	INT	Identificador do exame	PK
PacienteID	INT	Identificador do paciente	FK
TipoExame	VARCHAR	Tipo de exame	
Resultado	VARCHAR	Resultado do exame	
Data	DATE	Data do exame	

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

Paciente

PacienteID	Nome	Idade	Sexo
1	João	35	M
2	Maria	28	F
3	Carlos	45	M

Médico

MédicoID	Nome	Especialidade
101	Dr. Silva	Cardiologia
102	Dra. Lima	Pediatria
103	Dr. Souza	Ortopedia

Consulta

ConsultaID	PacienteID	MédicoID	Data	Hora
501	1	101	2024-05-30	10:00:00
502	2	102	2024-06-01	14:30:00
503	3	103	2024-06-02	09:45:00

Prescrição

PrescriçãoID	ConsultaID	Medicamento	Dosagem	Instruções
201	501	AAS 100mg	1 compr.	Tomar após a refeição
202	502	Amoxicilina	500mg	1 comprimido 3x ao dia
203	503	Dipirona	500mg	Tomar em caso de dor

Exame

ExameID	PacienteID	TipoExame	Resultado	Data
301	1	Hemograma	Normal	2024-06-10
302	2	Ultrassom	Normal	2024-06-12
303	3	Raio-X	Fratura	2024-06-15

3.4 – Redes de Computadores

3.4.1 - Montar a planta baixa de Rede da Empresa

Definição dos Departamentos A empresa HealthTech Balderi Solutions possui os seguintes departamentos:

1. Administração: Responsável pela gestão administrativa e financeira da empresa.

2. Desenvolvimento de Software: Equipe responsável pelo desenvolvimento de software personalizado e consultoria em TI.

3. Consultoria em Saúde: Equipe responsável pela consultoria em transformação digital e análise de dados em saúde.

4. Suporte Técnico: Equipe responsável pelo suporte técnico e manutenção da infraestrutura de TI. Definição dos Equipamentos por Departamento

- Administração:
- Computadores Desktop para funcionários administrativos.
- Impressoras de uso geral.
- Roteador para conexão com a internet.
- Desenvolvimento de Software:
- Estações de trabalho com alto desempenho para desenvolvedores.
- Servidor de desenvolvimento para hospedar aplicações em ambiente local.
- Consultoria em Saúde:

- Estações de trabalho com foco em análise de dados e modelagem estatística.
- Servidor de banco de dados para armazenamento de dados sensíveis.
- Suporte Técnico:
- Laptops para técnicos de suporte em deslocamento.
- Ferramentas de monitoramento de rede e sistemas.

3.4.2 - Configuração de IP de Todos os Equipamentos

Definição da Classe de Rede

1. Administração: 192.168.0.0/24

1.1 Intervalo de IPs: 192.168.0.1 - 192.168.0.50

1.2 Gateway padrão: 192.168.0.1

1.3 Máscara de sub-rede: 255.255.255.0

2. Desenvolvimento de Software: 192.168.1.0/24

2.1 Intervalo de IPs: 192.168.1.1 - 192.168.1.50

2.2 Gateway padrão: 192.168.1.1

2.3 Máscara de sub-rede: 255.255.255.0

3. Consultoria em Saúde: 192.168.2.0/24

3.1 Intervalo de IPs: 192.168.2.1 - 192.168.2.50

3.2 Gateway padrão: 192.168.2.1

3.3 Máscara de sub-rede: 255.255.255.0

4. Suporte Técnico: 192.168.3.0/24

4.1 Intervalo de IPs: 192.168.3.1 - 192.168.3.50

4.2 Gateway padrão: 192.168.3.1

Máscara de sub-rede: 255.255.255.0 Essas configurações garantem que cada departamento tenha sua própria faixa de endereços IP exclusiva e que todos os dispositivos em um mesmo departamento possam se comunicar entre si na mesma rede local.

3.5 - Segurança da Informação

3.5.1 - Análise de Riscos

1. **Identificação e Avaliação dos Riscos de Segurança para a Empresa:** Para garantir a segurança da informação na HealthTech Balderi Solutions, é crucial identificar e avaliar as ameaças e vulnerabilidades que podem comprometer os sistemas e dados da empresa. Abaixo está uma lista de 20 possíveis ameaças e/ou vulnerabilidades

1- Vazamento de Dados Confidenciais : Comprometimento de informações pessoais e médicas dos pacientes.
2- Ataques de Malware: Infecção por vírus, ransomware ou spyware nos sistemas da empresa.
3- Acesso Não Autorizado: Acesso indevido aos sistemas e dados por funcionários ou terceiros não autorizados.
4- Falhas de Segurança na Rede: Vulnerabilidades na rede que podem ser exploradas por hackers.
5- Ataques de Engenharia Social: Tentativas de manipulação de funcionários para obter acesso não autorizado às informações.
6- Phishing: Emails fraudulentos tentando obter informações sensíveis dos funcionários.
7- Falhas de Software: Bugs e vulnerabilidades em softwares utilizados pela empresa.
8- Falta de Atualizações de Segurança: Não aplicar patches de segurança e atualizações de software.
9- Roubo ou Perda de Dispositivos: Dispositivos contendo dados sensíveis que são perdidos ou roubados.
10- Senhas Fracas ou Comprometidas: Uso de senhas fracas ou reutilização de senhas em múltiplos sistemas.
11- Ameaças Internas: Funcionários descontentes ou mal-intencionados que podem comprometer a segurança.
12- DDoS (Ataques de Negação de Serviço Distribuídos): Ataques que visam sobrecarregar os sistemas e torná-los indisponíveis.
13- Interceptação de Comunicação: Captura de dados sensíveis durante a transmissão pela rede.
14- Falta de Criptografia: Dados armazenados ou transmitidos sem criptografia adequada.
15- Configurações de Segurança Padrão: Uso de configurações padrão que não foram personalizadas para segurança.
16- Desastres Naturais: Eventos como incêndios, inundações ou terremotos que podem danificar infraestruturas.
17- Erro Humano: Ações não intencionais dos funcionários que podem comprometer a segurança.

18- Backdoors em Software: Acesso não autorizado através de backdoors em aplicações de software.
19- Exploração de Zero-Day: Exploração de vulnerabilidades desconhecidas pelos desenvolvedores.
20- Falta de Monitoramento Contínuo: Não monitorar continuamente sistemas e redes para identificar atividades suspeitas.

Exemplo de Avaliação de Riscos: Para cada ameaça ou vulnerabilidade, é importante avaliar o impacto potencial e a probabilidade de ocorrência. Isso pode ser feito utilizando uma matriz de risco.

Matriz de Risco

Ameaça/Vulnerabilidade	Impacto	Probabilidade	Nível de Risco
Vazamento de Dados Confidenciais	Alto	Médio	Alto
Ataques de Malware	Alto	Alto	Alto
Acesso Não Autorizado	Alto	Médio	Alto
Falhas de Segurança na Rede	Alto	Médio	Alto
Ataques de Engenharia Social	Médio	Alto	Alto
Phishing	Médio	Alto	Alto
Falhas de Software	Médio	Médio	Médio
Falta de Atualizações de Segurança	Alto	Médio	Alto
Roubo ou Perda de Dispositivos	Médio	Médio	Médio
Senhas Fracas ou Comprometidas	Alto	Médio	Alto
Ameaças Internas	Alto	Médio	Alto
DDoS	Alto	Médio	Alto
Interceptação de Comunicação	Alto	Médio	Alto
Falta de Criptografia	Alto	Médio	Alto
Configurações de Segurança Padrão	Médio	Médio	Médio
Desastres Naturais	Alto	Baixo	Médio
Erro Humano	Médio	Alto	Alto
Backdoors em Software	Alto	Médio	Alto
Exploração de Zero-Day	Alto	Médio	Alto
Falta de Monitoramento Contínuo	Alto	Médio	Alto

2. **Análise de Vulnerabilidades e Ameaças Potenciais**
   Avaliação do Impacto e da Probabilidade de Ocorrência de Cada Risco. A avaliação de riscos deve considerar dois principais fatores: o impacto potencial (a gravidade das consequências caso o risco se concretize) e a probabilidade de ocorrência (a chance de o risco acontecer). Abaixo está a avaliação detalhada dos riscos identificados para a HealthTech Balderi Solutions.

   Critérios de Avaliação:

• Impacto:

• Alto: Consequências severas que podem afetar significativamente a operação da empresa.

• Médio: Consequências moderadas que podem causar interrupções, mas são gerenciáveis.

• Baixo: Consequências menores com impacto mínimo na operação.

• Probabilidade:

• Alta: Muito provável que ocorra no curto prazo.

• Média: Possível de ocorrer, mas não frequentemente.

• Baixa: Improvável de ocorrer, mas ainda possível. Avaliação de Riscos:

1.Vazamento de Dados Confidenciais

1.1 Impacto: Alto

1.2 Probabilidade: Médio

1.3 Avaliação: O vazamento de dados de pacientes pode levar a sérios problemas de privacidade e regulamentação, além de danificar a reputação da empresa. Probabilidade média devido a controles de segurança existentes.

2. Ataques de Malware

2.1 Impacto: Alto

2.2 Probabilidade: Alto

2.3 Avaliação: Malware pode causar paralisação dos sistemas e perda de dados. Alta probabilidade devido à prevalência de malware.

3. Acesso Não Autorizado

3.1 Impacto: Alto

3.2 Probabilidade: Médio

3.3 Avaliação: Pode resultar em comprometimento de dados sensíveis. Probabilidade média se medidas de segurança apropriadas forem implementadas.

4. Falhas de Segurança na Rede

4.1 Impacto: Alto

4.2 Probabilidade: Médio

4.3 Avaliação: Vulnerabilidades de rede podem ser exploradas para acesso não autorizado. Probabilidade média com boa gestão de segurança de rede.

5. Ataques de Engenharia Social

5.1 Impacto: Médio

5.2 Probabilidade: Alto

5.3 Avaliação: Engenharia social pode enganar funcionários e comprometer dados. Alta probabilidade devido à dependência do fator humano.

6. Phishing

6.1 Impacto: Médio

6.2 Probabilidade: Alto

6.3 Avaliação: Ataques de phishing são comuns e podem resultar em acesso a dados confidenciais. Alta probabilidade devido à prevalência.

7. Falhas de Software

7.1 Impacto: Médio

7.2 Probabilidade: Médio

7.3 Avaliação: Bugs de software podem causar interrupções e problemas de segurança. Probabilidade média dependendo da qualidade do software.

8. Falta de Atualizações de Segurança

8.1 Impacto: Alto

8.2 Probabilidade: Médio

8.3 Avaliação: Não aplicar atualizações pode deixar o sistema vulnerável. Probabilidade média se as atualizações forem gerenciadas adequadamente.

9. Roubo ou Perda de Dispositivos

9.1 Impacto: Médio

9.2 Probabilidade: Médio

9.3 Avaliação: Dispositivos perdidos ou roubados podem conter dados sensíveis. Probabilidade média com políticas adequadas de gestão de dispositivos.

10. Senhas Fracas ou Comprometidas

10.1 Impacto: Alto

10.2 Probabilidade: Médio

10.3Avaliação: Senhas fracas podem ser facilmente exploradas. Probabilidade média com políticas de senha fortes.

11. Ameaças Internas

11.1 Impacto: Alto

11.2 Probabilidade: Médio

11.3 Avaliação: Funcionários descontentes podem comprometer a segurança. Probabilidade média com gestão eficaz de pessoal.

12. DDoS (Ataques de Negação de Serviço Distribuídos)

12.1 Impacto: Alto

12.2 Probabilidade: Médio

12.3 Avaliação: Ataques DDoS podem tornar os sistemas indisponíveis. Probabilidade média com mitigação de DDoS em vigor.

13. Interceptação de Comunicação

13.1 Impacto: Alto

13.2 Probabilidade: Médio

13.3 Avaliação: Dados sensíveis podem ser capturados durante a transmissão. Probabilidade média com uso de criptografia.

14. Falta de Criptografia

14.1 Impacto: Alto

14.2 Probabilidade: Médio

14.3 Avaliação: Dados não criptografados são vulneráveis. Probabilidade média com criptografia adequada implementada.

15. Configurações de Segurança Padrão

15.1 Impacto: Médio

15.2 Probabilidade: Médio

15.3 Avaliação: Configurações padrão podem ser exploradas. Probabilidade média se não modificadas adequadamente.

16. Desastres Naturais

16.1 Impacto: Alto

16.2 Probabilidade: Baixo

16.3 Avaliação: Desastres naturais podem causar danos físicos aos sistemas. Baixa probabilidade, mas planejamento de contingência é essencial.

17. Erro Humano

17.1 Impacto: Médio

17.2 Probabilidade: Alto

17.3 Avaliação: Erros humanos são comuns e podem comprometer a segurança. Alta probabilidade devido à natureza humana.

18. Backdoors em Software

18.1 Impacto: Alto

18.2 Probabilidade: Médio

18.3 Avaliação: Backdoors podem ser usados para acesso não autorizado. Probabilidade média com controle de qualidade de software.

19. Exploração de Zero-Day

19.1 Impacto: Alto

19.2 Probabilidade: Médio

19.3 Avaliação: Vulnerabilidades desconhecidas podem ser exploradas. Probabilidade média com resposta rápida a ameaças emergentes.

20. Falta de Monitoramento Contínuo

20.1 Impacto: Alto

20.2 Probabilidade: Médio

20.3 Avaliação: Sem monitoramento, atividades suspeitas podem passar despercebidas. Probabilidade média com monitoramento implementado.

3.5.2 - Implementação de Medidas de Segurança

1. **Implementação de Políticas de Controle de Acesso aos Sistemas e Dados**
   Para mitigar os riscos identificados e garantir a segurança da informação, é fundamental implementar políticas robustas de controle de acesso. Estas políticas devem ser desenvolvidas com base nos riscos identificados e na estrutura organizacional da HealthTech Balderi Solutions.

   Políticas de Controle de Acesso:
   1. Autenticação Multifator (MFA) : Todos os usuários devem usar autenticação multifator para acessar sistemas críticos e dados sensíveis.
   2. Princípio do Menor Privilégio : Os usuários só devem ter acesso aos dados e sistemas necessários para realizar suas funções.
   3. Gestão de Identidade e Acesso (IAM) : Implementar um sistema IAM para gerenciar identidades de usuários e controlar o acesso a recursos com base em funções e responsabilidades.
   4. Revisão Regular de Acessos : Realizar revisões periódicas das permissões de acesso dos usuários para garantir que estejam atualizadas e adequadas.
   5. Senhas Fortes e Política de Senhas : Exigir que todas as senhas sejam fortes e trocadas regularmente. Políticas devem incluir requisitos mínimos de comprimento, complexidade e periodicidade de troca.
   6. Acesso Baseado em Funções (RBAC) : Implementar controles de acesso baseados em funções, onde as permissões são atribuídas com base na função do usuário dentro da organização.
   7. Controle de Acesso Físico : Limitar o acesso físico a servidores, centros de dados e áreas sensíveis a pessoal autorizado mediante identificação e autenticação.
   8. Logs de Acesso e Auditoria : Manter registros detalhados de todas as tentativas de acesso e atividades dos usuários, com auditorias regulares para detectar e responder a atividades suspeitas.
   9. Treinamento e Conscientização de Segurança :Treinar regularmente os funcionários sobre práticas seguras de acesso e conscientização sobre phishing, engenharia social e outras ameaças.
   10. Política de Desligamento Automático : Configurar sistemas para desligar automaticamente sessões inativas após um período determinado para prevenir acessos não autorizados.

Exemplo de Política de Controle de Acesso:
Título: Política de Acesso Baseado em Funções (RBAC)
Objetivo: Assegurar que os usuários tenham apenas os acessos necessários para desempenhar suas funções, minimizando os riscos de acesso não autorizado.
Escopo: Todos os funcionários, contratados e terceiros que necessitam acessar os sistemas e dados da HealthTech Balderi Solutions.

Política:
1. Definição de Funções: Todas as funções dentro da organização serão definidas e categorizadas com base nas necessidades de acesso a sistemas e dados.
2. Atribuição de Permissões: Permissões serão atribuídas a funções, e não a usuários individuais, para facilitar a gestão e garantir a conformidade.
3. Revisão de Acessos: As permissões de acesso serão revisadas trimestralmente para assegurar que os usuários mantenham apenas os acessos necessários.
4. Modificação de Acessos: Quaisquer mudanças nas funções dos funcionários (promoções, transferências, demissões) devem resultar em revisão e ajuste imediato das permissões de acesso.
5. Acesso Temporário: Acessos temporários, se necessários, devem ser concedidos com base em justificativas documentadas e com uma data de expiração claramente definida.

Procedimentos:
1.1 Solicitação de Acesso: Todas as solicitações de acesso devem ser submetidas por meio do sistema de gestão de identidades (IAM) e aprovadas pelo supervisor imediato do funcionário.
1.2 Auditoria e Monitoramento: Auditorias regulares serão conduzidas para revisar logs de acesso e detectar anomalias. Responsabilidades:
1.3 Gestores de TI: Implementar e manter o sistema IAM, assegurar a conformidade com a política de RBAC.
1.4 Supervisores: Aprovar solicitações de acesso e garantir que as permissões estejam alinhadas com as responsabilidades dos funcionários.
1.5 Funcionários: Seguir as políticas de acesso e reportar quaisquer incidentes ou suspeitas de acesso não autorizado imediatamente.

Configuração de Sistemas de Detecção de Intrusão e Prevenção de Ataques

Configuração de Sistemas de Detecção de Intrusão e Prevenção de Ataques

Para fortalecer a segurança dos sistemas da HealthTech Balderi Solutions, é essencial implantar e configurar sistemas eficazes de detecção de intrusão e prevenção de ataques. Abaixo estão 10 medidas de detecção e prevenção de ataques recomendadas para proteger os ativos de informação da empresa.

Medidas de Detecção e Prevenção de Ataques:

1. Firewalls de Rede : Configurar firewalls de rede para filtrar e monitorar o tráfego de rede, bloqueando tráfego malicioso e permitindo apenas comunicações autorizadas.
2. Sistemas de Detecção de Intrusão (IDS) : Implementar IDS para monitorar o tráfego de rede em busca de atividades suspeitas ou padrões de comportamento maliciosos que possam indicar uma intrusão.
3. Sistemas de Prevenção de Intrusão (IPS) : Complementar os IDS com sistemas de IPS, que podem responder automaticamente a atividades maliciosas, bloqueando pacotes de rede suspeitos ou aplicando políticas de segurança.
4. Monitoramento de Logs : Configurar sistemas de monitoramento de logs para registrar e analisar atividades de sistemas e rede, facilitando a detecção de comportamentos anômalos e eventos de segurança.
5. Análise de Tráfego SSL/TLS •:Realizar inspeção SSL/TLS para detectar e bloquear tráfego malicioso criptografado, garantindo que todo o tráfego seja visível e passível de inspeção.
6. Sistemas de Detecção de Malware : Implementar sistemas de detecção de malware para identificar e bloquear arquivos maliciosos antes que possam causar danos aos sistemas da empresa.
7. Controle de Acesso Baseado em Comportamento (BAC) : Utilizar sistemas de BAC para monitorar o comportamento dos usuários e detectar atividades anômalas que possam indicar comprometimento de contas ou credenciais.
8. Atualizações de Segurança Automáticas : Configurar sistemas para receber e aplicar automaticamente atualizações de segurança, garantindo que os sistemas estejam protegidos contra vulnerabilidades conhecidas.

9. Bloqueio de Portas Não Utilizadas : Desativar e bloquear portas de rede não utilizadas nos sistemas da empresa para reduzir a superfície de ataque e limitar pontos de entrada para potenciais invasores.
10. Autenticação Forte : Implementar autenticação forte para acessar sistemas críticos, como autenticação de dois fatores (2FA) ou autenticação baseada em certificados.

Exemplo de Configuração:
Medida: Configuração de IDS/IPS
Descrição: Implementar um sistema de IDS/IPS para monitorar o tráfego de rede em tempo real, detectando e bloqueando atividades maliciosas.
Configuração:

1.1 Instalar e configurar um IDS/IPS em um ponto estratégico da rede, como na borda da rede.

1.2 Definir políticas de detecção para identificar padrões de tráfego malicioso, como ataques de negação de serviço (DDoS), explorações de vulnerabilidades conhecidas e tentativas de acesso não autorizado.

1.3 Configurar ações de resposta automáticas para atividades suspeitas, como bloquear endereços IP, desativar portas ou enviar alertas para a equipe de segurança.

1.4 Realizar regularmente atualizações de assinaturas e regras de detecção para garantir a eficácia contínua do IDS/IPS contra ameaças emergentes