

Projeto apresentado ao Centro Universitário Brasileiro **UNIBRA**;

**Curso**: Analise e Desenvolvimento de Sistemas;

**Disciplina**: Engenharia de Requisitos.

**Professor**: Ivson Estevão Soares

**artefato: PAPELARIA CANERNOS E MIMOS**

**LOGO:**



**equipe STAKEHOLDERS**

(Funções e Participantes)

**GESTÃO DE PROJETO:**

BRUNA IZABEL DE SOUZA BELLO QUEIROZ

**ANALISTA DE REQUISITOS:**

RODRIGO ANDRÉ SILVA PRIMO

DANIEL ALVES DO NASCIMENTO

VINICIUS GUERRA BACELAR FERREIRA

MATHEUS FELIPE OLIVEIRA CIRNE DE AZEVEDO

**DESENVOLVEDORES:**

FELIPE JOSÉ DE MELO

ERICKE BARBOSA DOS SANTOS

LEONARDO BARBOSA DA SILVA

LEONARDO CORREIA DE AGUIAR

JORGE LUCAS SANTOS DE OLIVEIRA

**SUMÁRIO**

**1.** Introdução………………………………………………………………………….....

**2.** REFERENCIAL TEORICO................................................................................

**2.1**. Método de levantamento de requisitos …………… .......................................

**2.2**. Entrevista………………………………………………………………………......

**3**. DESENVOLVIMENTO DO BANCO DE DADOS…………………………….......

**3.1.** Modelo Logico……………………………………………………………………..

**3.2.** Modelo ER ....................................................................................................

**3.3.** Modelo Físico……………………………………………………………………...

4. PROGRAMAÇÃO ORIENTADA A OBJETOS…………………….....................

**5**. TERMOS EM INGLÊS……………………………………………………………...

**6**. CRONOGRAMA ...............................................................................................

**7**. TECNOLOGIAS UTILIZADAS .........................................................................

**8**.REFERÊNCIAS..................................................................................................

**1. INTRODUÇÃO**

O processo de análise de requisitos envolve a coleta de informações, a identificação e definição de requisitos funcionais e não funcionais, a validação dos requisitos com o cliente, a priorização e o gerenciamento dos requisitos ao longo do ciclo de vida do software.

Uma análise de requisitos eficaz pode contribuir para a entrega de um software de alta qualidade, que atenda às expectativas do cliente e dos usuários, além de reduzir custos e prazos no desenvolvimento do projeto. A análise de requisitos é um processo contínuo e iterativo, que pode ser realizado em conjunto com outras atividades do processo de desenvolvimento de software, como a prototipação, o design, a implementação e os testes.

**2. REFERENCIAL TEÓRICO**

**2.1 Método de levantamento de requisitos**

Decidimos utilizar o método de vídeo/entrevista. Essa abordagem é comum em entrevistas de emprego, entrevistas de pesquisa ou em qualquer situação em que os participantes estejam localizados em diferentes lugares geográficos.

A entrevista é uma técnica de análise de requisitos muito utilizada em projetos de desenvolvimento de software. Nessa técnica, o analista de requisitos conduz entrevistas com os usuários e outros stakeholders do sistema, com o objetivo de coletar informações sobre suas necessidades, expectativas e requisitos em relação ao software a ser desenvolvido.

**2.2. Entrevista**

Este é o resultado Vídeo/entrevista que foi realizada na empresa **PAPELARIA CADERNOS E MIMOS**,a entrevista foi feita com a proprietária.

**Nome**: Papelaria Cadernos e Mimos

**Quantidade de funcionários**: 1 funcionário

**Localização**: Av. Leopoldo Lins, N.1080 – Bairro: Centro - Tamandaré-PE

**Missão**: Sempre me qualificar para trazer novidades e qualidade nos produtos e serviços.

**Visão**: Fazer a loja online alcançar clientes de todos os nichos e em todo Brasil.

**Valores**: Inovação, Diversidade, Qualidade, Personalizados e Exclusivos.

**Pergunta: Qual o seu público alvo?**

**R**: Eu não tenho um público alvo específico.

**Pergunta: Que tipos de produtos você vende?**

**R**: Cadernos, Agendas, Planner’s, Cadernetas de Vacinação, entre outros.

**Pergunta: Quais dos seus produtos são os mais vendidos?**

**R**: Agendas e Cadernetas de Vacinação.

**Pergunta: Você oferece mais algum outro serviço?**

**R**: Sim, Serviços de Encadernação, Plastificação e diversos tipos de Impressão.

**Pergunta: Quanto tempo leva desde o início da criação até a entrega do produto ao cliente?**

**R**: Dependendo do produto e a quantidade, entre 5 horas a 10 horas. Depois da data acordada no agendamento.

**Pergunta: É preciso mostrar imagens de outros produtos a novos clientes?**

**R**: Geralmente sim. Vídeos e fotos de pedidos finalizados que estão nas redes sociais.

**Pergunta: Quais suas principais formas de entrega?**

**R**: A mais comum é retirada no local, mas envio por transportadoras também.

**Pergunta: Quais formas de pagamento você disponibiliza?**

**R**:Valor a vista. Pix. Cartão de Credito.

**Pergunta: Qual tipo de sistema te atenderia melhor?**

**R**: Um aplicativo (Android/iOS).

**Pergunta**: **Qual forma você se comunica com os clientes?**

**R:** WhatsApp.

**Pergunta: Qual seria o objetivo de um novo sistema?**

**R**: Alcançar novos clientes, facilitar compras, controle de estoque.

**Relatório da Vídeo/Entrevista**

**Resultados obtidos da Vídeo/entrevista resultaram nos seguintes problemas:**

**Atendimento:**

Não há problema específico a ser resolvido, mas há oportunidades de melhoria em relação ao sistema, que pode ajudar a alcançar novos clientes e aprimorar o controle de estoque.

**Tendo em vista a situação observada:**

**Sobre o atendimento:**

Com base nos requisitos mencionados na entrevista, sugiro o desenvolvimento de um aplicativo de compras que permita aos clientes visualizar os produtos, fazer pedidos, escolher as opções de entrega e pagamento e acompanhar o status do pedido em tempo real.

Além disso, o aplicativo pode incluir recursos adicionais, como notificações push para informar os clientes sobre promoções ou atualizações de status do pedido, avaliações de produtos e recursos de compartilhamento em redes sociais para ajudar a divulgar a loja.

**3. DESENVOLVIMENTO DO BANCO DE DADOS**

* **Minimundo**

A empresa Papelaria Cadernos de Mimos necessita ampliar o seu atendimento online. Mas, devido ao baixo número de funcionários, a solução escolhida foi um aplicativo, pare que ela possa expandir seu marketing digital além do seu site e atendimento via WhatsApp.

* **Requisitos do Sistema**

Sistema Operacional: Android 5.0 ou superior, iOS 10 ou superior

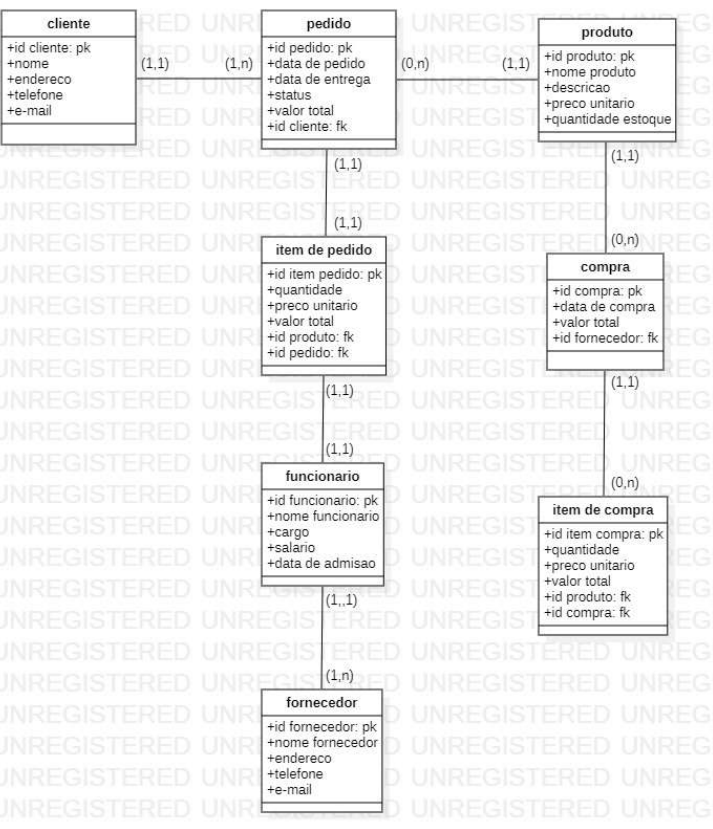
Memória RAM: 2 GB ou superior

Armazenamento: 100 MB de espaço livre

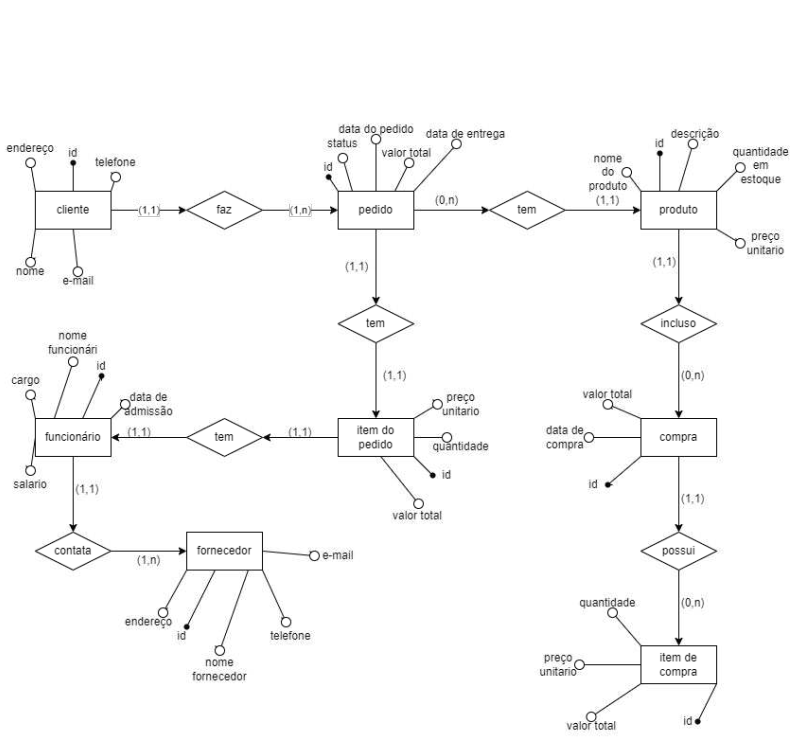
Processador: mínimo de 1,4 GHz ou superior

Conexão à internet: conexão de dados móveis (3G/4G) ou Wi-Fi

**3.1. Modelo Lógico**



**3.2. Modelo ER**



**3.3. Modelo Físico**

**CRIAÇÃO DO BANCO DE DADOS**

CREATE DATABASE grafica\_loja;

**SELECIONA O BANCO DE DADOS PARA USO**

USE grafica\_loja;

-- Criação da tabela Cliente

CREATE TABLE Cliente (

ID INT NOT NULL AUTO\_INCREMENT,

nome VARCHAR(50) NOT NULL,

endereco VARCHAR(100) NOT NULL,

telefone VARCHAR(20) NOT NULL,

email VARCHAR(50) NOT NULL,

PRIMARY KEY (ID)

);

**CRIAÇÃO DA TABELA PEDIDO**

CREATE TABLE Pedido (

ID INT NOT NULL AUTO\_INCREMENT,

data\_pedido DATE NOT NULL,

data\_entrega DATE,

status VARCHAR(20) NOT NULL,

valor\_total DECIMAL(10,2) NOT NULL,

cliente\_id INT NOT NULL,

PRIMARY KEY (ID),

FOREIGN KEY (cliente\_id) REFERENCES Cliente(ID)

);

**CRIAÇÃO DA TABELA PRODUTO**

CREATE TABLE Produto (

ID INT NOT NULL AUTO\_INCREMENT,

nome VARCHAR(50) NOT NULL,

descricao VARCHAR(200),

preco\_unitario DECIMAL(10,2) NOT NULL,

qtd\_estoque INT NOT NULL,

PRIMARY KEY (ID)

);

**CRIAÇÃO DA TABELA ITEM DE PEDIDO**

CREATE TABLE ItemPedido (

ID INT NOT NULL AUTO\_INCREMENT,

quantidade INT NOT NULL,

preco\_unitario DECIMAL(10,2) NOT NULL,

valor\_total DECIMAL(10,2) NOT NULL,

produto\_id INT NOT NULL,

pedido\_id INT NOT NULL,

PRIMARY KEY (ID),

FOREIGN KEY (produto\_id) REFERENCES Produto(ID),

FOREIGN KEY (pedido\_id) REFERENCES Pedido(ID)

);

**CRIAÇÃO DA TABELA FUNCIONARIO**

CREATE TABLE Funcionario (

ID INT NOT NULL AUTO\_INCREMENT,

nome VARCHAR(50) NOT NULL,

cargo VARCHAR(50) NOT NULL,

salario DECIMAL(10,2) NOT NULL,

data\_admissao DATE NOT NULL,

PRIMARY KEY (ID)

);

**CRIAÇÃO DA TABELA FORNECEDOR**

CREATE TABLE Fornecedor (

ID INT NOT NULL AUTO\_INCREMENT,

nome VARCHAR(50) NOT NULL,

endereco VARCHAR(100) NOT NULL,

telefone VARCHAR(20) NOT NULL,

email VARCHAR(50) NOT NULL,

PRIMARY KEY (ID)

);

**CRIAÇÃO DA TABELA COMPRA**

CREATE TABLE Compra (

ID INT NOT NULL AUTO\_INCREMENT,

data\_compra DATE NOT NULL,

valor\_total DECIMAL(10,2) NOT NULL,

fornecedor\_id INT NOT NULL,

PRIMARY KEY (ID),

FOREIGN KEY (fornecedor\_id) REFERENCES Fornecedor(ID)

);

**CRIAÇÃO DA TABELA ITEM DE COMPRA**

CREATE TABLE ItemCompra (

ID INT NOT NULL AUTO\_INCREMENT,

quantidade INT NOT NULL,

preco\_unitario DECIMAL(10,2) NOT NULL,

valor\_total DECIMAL(10,2) NOT NULL,

produto\_id INT NOT NULL,

compra\_id INT NOT NULL,

PRIMARY KEY (ID),

FOREIGN KEY (produto\_id) REFERENCES Produto(ID),

FOREIGN KEY (compra\_id) REFERENCES Compra(ID)

);

**4. PROGRAMAÇÃO ORIENTADA A OBJETOS: DIAGRAMA / CLASSES**

**//CLASSE CLIENTE**

**public class Cliente {**

private static int idCliente;

private static String nome;

private static String endereco;

private static String telefone;

private static String email;

**// GETTERS E SETTERS**

public int getId()

{return idCliente;}

public void setId(int idCliente)

{Cliente.idCliente = idCliente;}

public static String getNome()

{return nome};

public static void setNome(String nome)

{Cliente.nome = nome;}

public static String getEndereco()

{return endereco;}

public static void setEndereco(String endereco)

{Cliente.endereco = endereco;}

public static String getTelefone()

{return telefone;}

public static void setTelefone(String telefone)

{Cliente.telefone = telefone;}

public static String getEmail()

{return email;}

public static void setEmail(String email)

{Cliente.email = email;}

};

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**//CLASSE COMPRA**

**public class Compra {**

private static int idCompra;

private static String dataCompra;

private static double valorTotal;

private static int idFornecedor;

**// GETTERS E SETTERS**

public static int getIdCompra()

{return idCompra;}

public static void setIdCompra(int idCompra)

{Compra.idCompra = idCompra;}

public static String getDataCompra()

{return dataCompra;}

public static void setDataCompra(String dataCompra)

{Compra.dataCompra = dataCompra;}

public static double getValorTotal()

{return valorTotal;}

public static void setValorTotal(double valorTotal)

{Compra.valorTotal = valorTotal;}

public static int getIdFornecedor()

{return idFornecedor;}

public static void setIdFornecedor(int idFornecedor)

{Compra.idFornecedor = idFornecedor;}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Classe conexão = Banco de Dados Mysq**

package conexao;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

**//CLASSE**

**public class ConexaoBanco** {

public static Connection getConexao() throws ClassNotFoundException, SQLException {

String ENDERECO\_BASE = "127.0.0.1";

String NOME\_BASE = "Grafica\_Loja";

String CLIENTE = "Cliente";

String SENHA = "Sigma";

Class.forName("com.mysql.cj.jdbc.Driver");

Connection conexao = DriverManager.getConnection("jdbc:mysql://" + ENDERECO\_BASE + "/" + NOME\_BASE, CLIENTE, SENHA);

return conexao;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Classe = Método Inserir e Consultar CLIENTE**

package sigma;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.List;

import conexao.ConexaoBanco;

import pooProjeto.Cliente;

**public class RepositorioPessoa {**

public Cliente inserirPessoa(Cliente pessoa) throws ClassNotFoundException, SQLException {

Connection conexao = ConexaoBanco.getConexao();

String query = "INSERT INTO CLIENTE (IDCLIENTE,NOME,TELEONE, EMAIL, ENDERECO) VALUES (?, ?, ?)";

PreparedStatement stm = conexao.prepareStatement(query);

int i = 1;

stm.setString(i++, pessoa.getIdCliente());

stm.setString(i++, pessoa.getNome());

stm.setString(i++, pessoa.getTeleone());

stm.setString(i++, pessoa.getEmail());

stm.setString(i++, pessoa.getEndereco());

stm.executeUpdate();

conexao.close();

return pessoa;

}

public List<Cliente> consultarPessoas() throws ClassNotFoundException, SQLException {

Connection conexao = ConexaoBanco.getConexao();

String query = "SELECT \* FROM CLIENTE";

Statement stm = conexao.createStatement();

ResultSet rs = stm.executeQuery(query);

List<Cliente> pessoas = new ArrayList<Cliente>();

while(rs.next()) {

Cliente pessoa = new Cliente();

pessoa.setIdCliente(rs.getInt("IDCLIENTE"));

pessoa.setNome(rs.getString("NOME"));

pessoa.setTelefonel(rs.getString(“TELEEFONE”));

pessoa.setEmail(rs.getString("EMAIL"));

pessoa.setEndereco(rs.getString("ENDERECO"));

pessoas.add(pessoa);

}

conexao.close();

return pessoas;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5. TERMOS EM INGLÊS**

* **Termos em inglês utilizados no artefato do projeto**

**Chat:** Um chat é um programa de computador que usa inteligência artificial para conversar com as pessoas por meio de uma interface de bate papo.

**Not Null**: Not Null é uma expressão usada em um banco de dados para indicar que um campo ou coluna não pode ser deixado vazio ou em branco. Em outras palavras, é uma restrição que que impede dados em branco ou nulos sejam inseridos em um determinado campo.

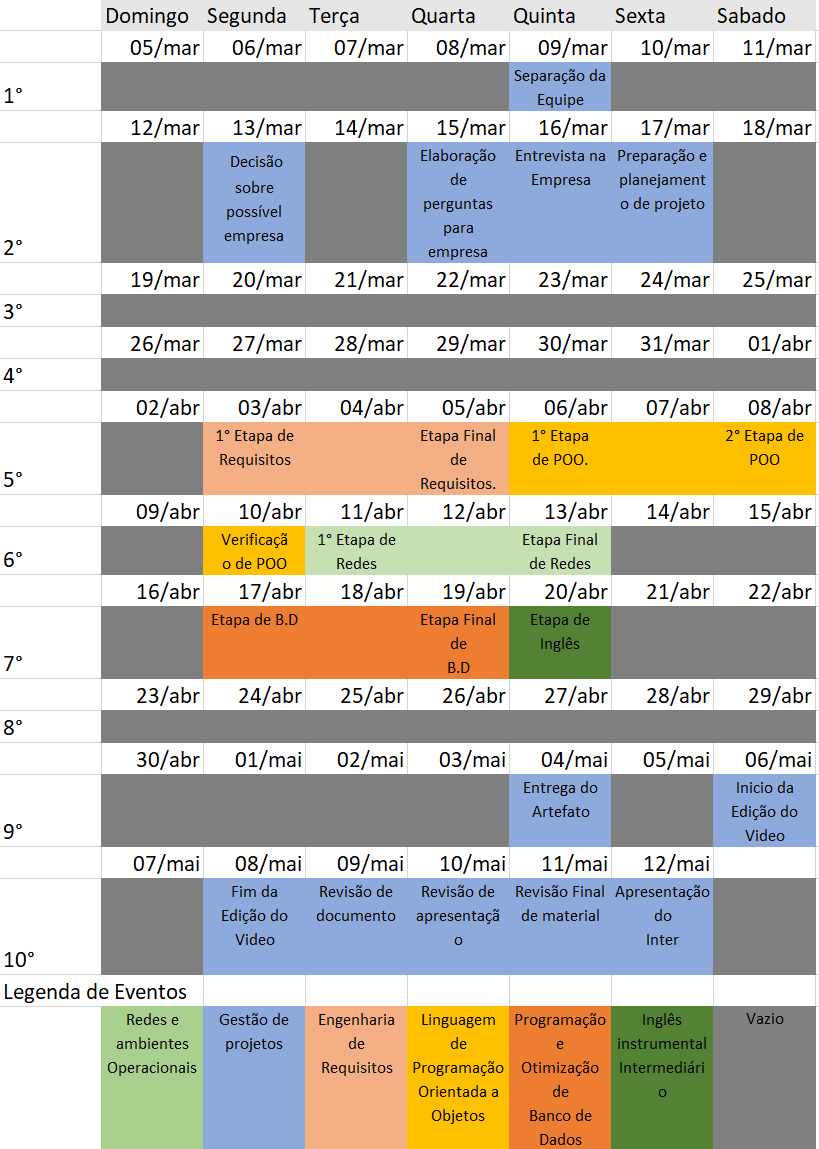
**Primary Key**: Primary Key significa "chave principal" em português. Este é um termo usado para identificar exclusivamente cada registro em uma tabela em um banco de dados. Uma chave primária consiste em um ou mais campos não repetidos que garantem a integridade dos dados.

**Foreign**: Foreign significa "estrangeiro" em português. Em um banco de dados, esse é o termo usado para denotar uma chave de referência, que é um campo em uma tabela relacionado à chave primária de outra tabela.

**Create:** Create significa "criar" em português. É um verbo muito utilizado em contextos tecnológicos e de programação para criar novos arquivos, pastas, bancos de dados, sites, entre outros.

**Table:** Table significa "mesa" em português. É usado em bancos de dados para descrever uma estrutura de dados que contém informações organizadas em linhas e colunas.

**6. CRONOGRAMA**

****

**7. TECNOLOGIAS UTILIZADAS**

* **Br Modelo**
* **MySQL**
* **Sketchware Pro**
* **Discord**
* **Eclipse**
* **GitHub**
* **Canva**
* **Vegas**

**8. REFERÊNCIAS**

PRESSMAN, Roger S. - **Engenharia de Software**: Uma Abordagem Profissional, 7ª edição, McGraw Hill, 2011.

WIEGERS, Karl E., - **Software Requirement’s**, 3ª edição, Microsoft Press,2013.

SOMMERVILLE, Ian. **Engenharia de Software** - 9ª edição, Pearson - CW Companion Website, 2011.

FONSECA, George. **Fundamentos de Banco de Dados**. Disponível em: <http://professor.ufop.br/sites/default/files/george/files/2020-2\_apostila\_cdd003.pdf>. Acesso em: 20/04/2023.

PUGA, Sandra; RISSETTI, Gerson. **Lógica de programação e estrutura de dados com aplicações em Java**. São Paulo: Pearson Education do Brasil, 2004.