



**UNifeob**  
| ESCOLA DE NEGÓCIOS

**2023**

**PROJETO DE CONSULTORIA  
EMPRESARIAL**



UNIFEOB  
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO  
OCTÁVIO BASTOS  
ESCOLA DE NEGÓCIOS  
**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**  
**GESTÃO DE TECNOLOGIA DA INFORMAÇÃO**

**PROJETO INTEGRADO**  
SISTEMA EMPRESARIAL

SÃO JOÃO DA BOA VISTA, SP

ABRIL 2023

UNIFEOB  
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO  
OCTÁVIO BASTOS  
ESCOLA DE NEGÓCIOS  
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS  
GESTÃO DE TECNOLOGIA DA INFORMAÇÃO

**PROJETO INTEGRADO**  
**SISTEMA EMPRESARIAL**

MÓDULO DESENVOLVIMENTO DESKTOP

Banco de Dados – Prof. Sidney Gitcoff Telles

Programação Orientada a Objeto – Prof. Sidney Gitcoff Telles

Projeto de Desenvolvimento Desktop – Prof. Sidney Gitcoff Telles

Estudantes:

Bruno Domingos, RA 1012023100245

Daniel dos Reis Germinaro, RA 1012023100287

João Roberto Buffo Nogueira, RA 1012023100141

Mara Vicentina Pinto, RA 1012023100321

Murilo de Lima Santos, RA 1012023100352

Rodolfo de Lima Peres, RA 1012023100191

SÃO JOÃO DA BOA VISTA, SP  
ABRIL, 2023

# SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>4</b>
<b>2. DESCRIÇÃO DA EMPRESA</b>	<b>4</b>
<b>3. PROJETO DE CONSULTORIA EMPRESARIAL</b>	<b>5</b>
<b>3.1. BANCO DE DADOS</b>	<b>5</b>
3.1.1. MER - MODELAGEM ENTIDADE RELACIONAMENTO	5
3.1.2. DER - DIAGRAMA ENTIDADE RELACIONAMENTO	6
3.1.3. FÍSICO	7
<b>3.2. PROGRAMAÇÃO ORIENTADA A OBJETOS</b>	<b>12</b>
3.2.1. DIAGRAMA DE CLASSES	13
3.2.2. CÓDIGOS DO SISTEMA	14
3.2.3. IMAGENS DO SISTEMA	22
<b>4. CONTEÚDO DA FORMAÇÃO PARA A VIDA: CRIANDO O NOVO</b>	<b>28</b>
4.1. ESTUDANTES NA PRÁTICA – LINK DO VÍDEO	28
<b>5. CONCLUSÃO</b>	<b>29</b>
<b>6. REFERÊNCIAS</b>	<b>29</b>

# **1. INTRODUÇÃO**

Nos últimos anos, a evolução tecnológica tem impactado fortemente a forma como as pessoas trabalham e se comunicam. Com a alta popularidade dos dispositivos móveis nos últimos anos, é comum subestimar o papel fundamental que os sistemas desktop ainda desempenham em muitas empresas e organizações em todo o mundo.

Apesar da revolução dos aplicativos móveis, muitas atividades complexas ainda dependem de um ambiente de desktop completo, que possua grande capacidade de processamento e altos recursos de segurança.

O intuito deste projeto é desenvolver um sistema desktop para uma farmácia de pequeno porte, contemplando as atividades básicas de uma farmácia, como controle de produtos e clientes, entrada e saída, controle de vendas.

Com o sistema desktop desenvolvido para essa farmácia, será possível automatizar muitos processos manuais, melhorando a eficiência e reduzindo o tempo gasto em tarefas rotineiras. Além disso, o sistema oferecerá uma maior segurança no controle de estoque, garantindo a integridade dos dados da farmácia e dos seus clientes. Com isso, espera-se que a implementação do sistema proporcione uma gestão mais eficiente do negócio, aumentando a satisfação dos clientes e gerando melhores resultados para a farmácia.

# **2. DESCRIÇÃO DA EMPRESA**

A empresa contratada foi a drogaria com o nome social “Drogaria Caio Donnabella” CNPJ 46.843.482/0001-42 que está localizada na cidade de Caconde na rua Coronel Tito Leonel número 80 bairro Bela Vista

Em conversa com o cliente, foi relatado que houve um acidente meteorológico em sua farmácia no dia 03/10/2022 em que causou problemas com anotações não digitais, perdendo desta forma, 30% do seu estoque por falta de organização. Também fomos informados de que a forma em que o cliente estava realizando as suas anotações mensais estava sendo insuficiente para calcular com precisão os seus lucros mensais, o deixando o utente deslocado com seus dados no fim do mês.

Depois de estudarmos o seu caso, entramos em contato novamente com o cliente e oferecemos uma proposta de um futuro para a sua farmácia com um programa mais útil e mais organizado, no qual ele e seus funcionários terão acesso total a localização dos seus produtos, valores e quantidades em estoque de forma fácil e prática.

Partindo deste pressuposto, concluímos que o trabalho do nosso cliente se tornará, desta forma, mais prático e organizado, além de ter muito mais agilidade para cadastros, vendas, devolução de troco e cálculos lucrativos mensais.

### **3. PROJETO DE CONSULTORIA EMPRESARIAL**

Nesta etapa do PI serão apresentados os conteúdos que cada unidade de estudo utilizou para realizar o projeto, assim como a forma que serão aplicados na empresa escolhida para a realização do projeto.

#### **3.1. BANCO DE DADOS**

Um banco de dados é um sistema que armazena informações organizadas de maneira estruturada, permitindo o acesso rápido e eficiente a essas informações. A criação de um banco de dados envolve várias etapas, que podem variar dependendo do tipo de banco de dados e do tamanho do projeto. Algumas das principais etapas incluem a definição dos requisitos do sistema, a modelagem do banco de dados, o projeto do banco de dados, a implementação e testes e a operação e manutenção do sistema.

Para elaborar um banco de dados, é necessário planejamento cuidadoso e colaboração de uma equipe multidisciplinar, incluindo desenvolvedores, analistas de dados e especialistas em infraestrutura. No entanto, como estudante do primeiro período de Ciências da Computação, é provável que você ainda não esteja familiarizado com todos os termos técnicos e conceitos relacionados ao assunto. Portanto, é importante buscar mais informações sobre o assunto em livros, artigos e outros recursos disponíveis para aprimorar seus conhecimentos sobre bancos de dados.

##### **3.1.1. MER - MODELAGEM ENTIDADE RELACIONAMENTO**

O Modelo Entidade-Relacionamento (MER) é uma ferramenta utilizada para modelagem de dados, que tem como objetivo descrever as entidades envolvidas em um sistema, seus atributos e relacionamentos. Ele é utilizado para representar visualmente as informações que serão armazenadas em um banco de dados e é um dos modelos mais utilizados para a criação de bancos de dados relacionais.

A criação de um MER é uma etapa importante na elaboração de um banco de dados, pois ajuda a definir a estrutura e as funcionalidades necessárias para atender aos requisitos do sistema. Depois de criado o modelo, é possível transformá-lo em um modelo lógico e físico,

que será utilizado para criar o banco de dados propriamente dito, com as tabelas, campos e restrições necessárias para armazenar as informações de maneira organizada e eficiente.

Após reunião com o representante da farmácia, cliente alvo do nosso projeto, foi possível definir os objetivos do projeto, e para facilitar o entendimento do nosso cliente e da equipe desenvolvedora foi elaborado o MER que pode ser visualizado na figura 1.

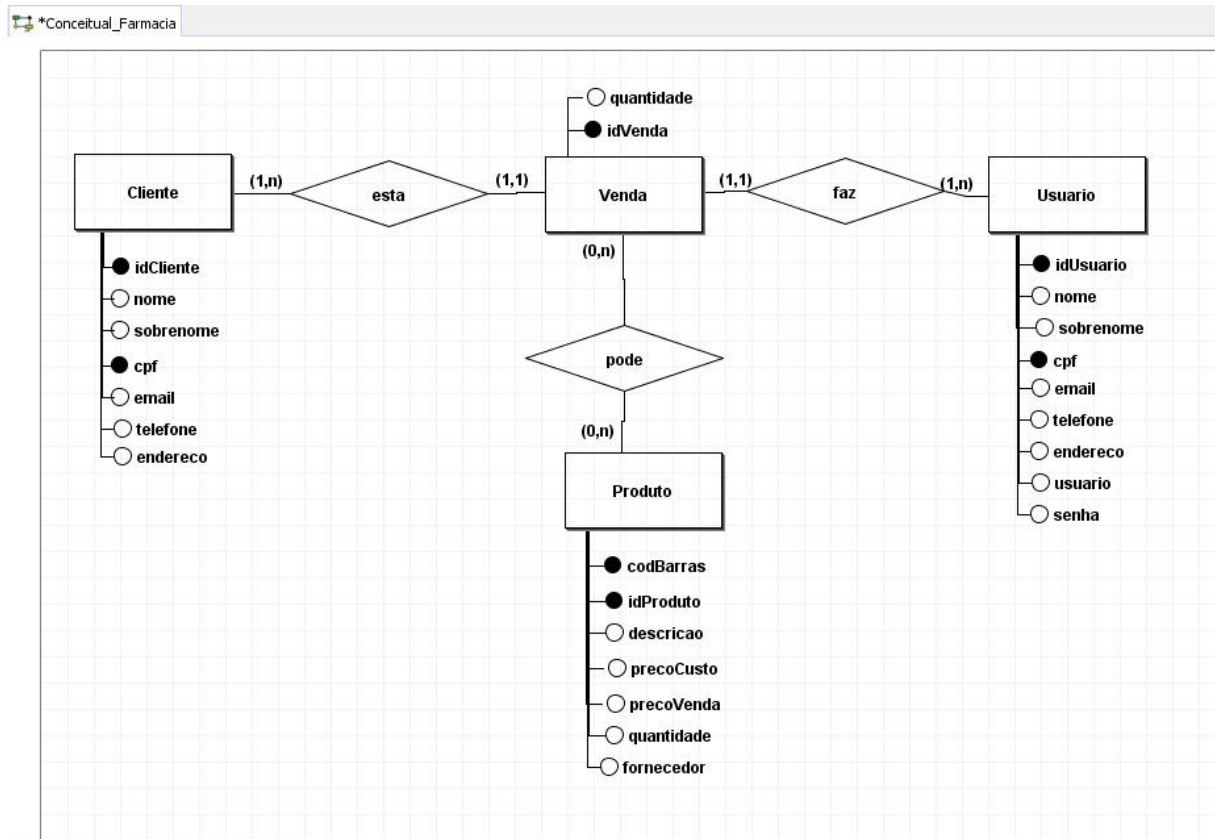


Figura 1. Modelo Entidade Relacionamento (MER) do projeto

Fonte: Autores (2023)

### 3.1.2. DER - DIAGRAMA ENTIDADE RELACIONAMENTO

O Diagrama Entidade-Relacionamento (DER) é uma técnica utilizada para modelagem de dados, que tem como objetivo representar graficamente as informações que serão armazenadas em um banco de dados. O DER é uma evolução do Modelo Entidade-Relacionamento (MER), sendo utilizado para representar visualmente as informações de forma mais detalhada e precisa, ajudando a identificar possíveis erros e melhorias na modelagem do banco de dados.

A elaboração do DER é uma etapa importante na criação de um banco de dados, pois permite visualizar de forma clara e precisa as informações que serão armazenadas e como elas se relacionam entre si. Além disso, o DER pode ser utilizado como uma ferramenta de



comunicação entre a equipe de desenvolvimento e os stakeholders, como clientes e usuários finais, para garantir que todos tenham a mesma compreensão sobre o sistema e suas funcionalidades.

O DER elaborado a partir do MER citado no tópico anterior pode ser visualizado nas figuras 2. Ele foi elaborado no programa MySQL.

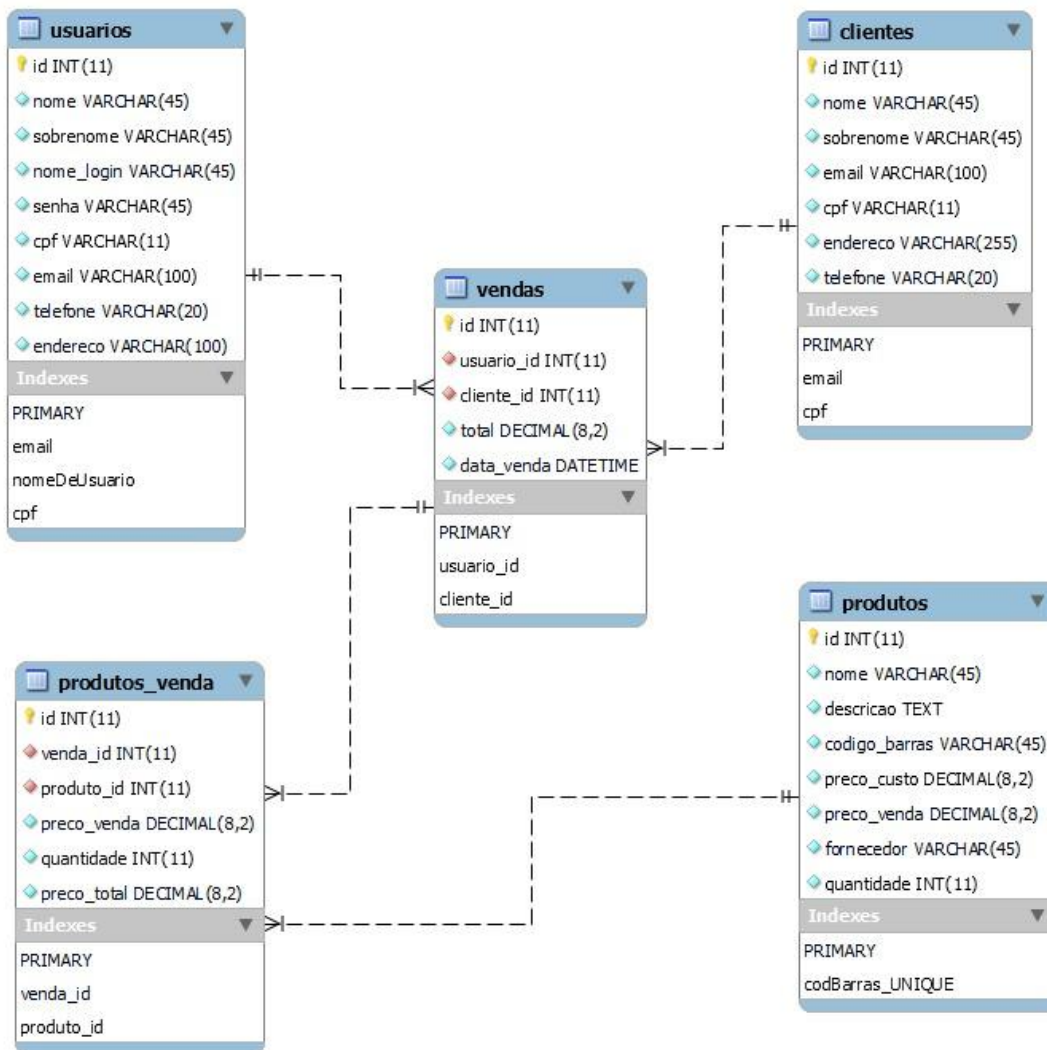


Figura 2. Diagrama Entidade Relacionamento - elaborado no software MySQL.

Fonte: Autores (2023)

### 3.1.3. FÍSICO

A parte física do projeto de banco de dados é responsável por definir como os dados serão armazenados no hardware. Uma das principais preocupações na parte física do projeto de banco de dados é garantir o desempenho e a confiabilidade do sistema.



O código que gera as tabelas do banco de dados foi criado no MySQL com base nos MER e DER dos tópicos anteriores, tanto o código quanto as tabelas geradas podem ser visualizadas nas figuras 3, 4, 5, 6 e 7.

```
1 -- MySQL dump 10.13 Distrib 8.0.19, for Win64 (x86_64)
2 --
3 -- Host: localhost Database: bdfarmacia
4 -----
5 -- Server version 5.5.5-10.4.27-MariaDB
6
7 ● /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8 ● /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9 ● /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 ● /*!50503 SET NAMES utf8mb4 */;
11 ● /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 ● /*!40103 SET TIME_ZONE='+00:00' */;
13 ● /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 ● /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 ● /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 ● /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17
18 --
19 -- Table structure for table `clientes`
20 --
21
22 ● DROP TABLE IF EXISTS `clientes`;
23 ● /*!40101 SET @saved_cs_client = @@character_set_client */;
24 ● /*!50503 SET character_set_client = utf8mb4 */;
25 ● CREATE TABLE `clientes` (
26   `id` int(11) NOT NULL AUTO_INCREMENT,
27   `nome` varchar(45) NOT NULL,
28   `sobrenome` varchar(45) NOT NULL,
29   `email` varchar(100) NOT NULL,
30   `cpf` varchar(11) NOT NULL,
31   `endereço` varchar(255) NOT NULL,
32   `telefone` varchar(20) NOT NULL,
33   PRIMARY KEY (`id`),
34   UNIQUE KEY `email` (`email`),
35   UNIQUE KEY `cpf` (`cpf`)
36 ) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
37 ● /*!40101 SET character_set_client = @saved_cs_client */;
38
39 --
40 -- Dumping data for table `clientes`
```

Figura 3. Códigos do BD - MySQL

Fonte: Autores (2023)

```

40 -- Dumping data for table `clientes`
41 --
42
43 ● LOCK TABLES `clientes` WRITE;
44 ● /*!40000 ALTER TABLE `clientes` DISABLE KEYS */;
45 ● INSERT INTO `clientes` VALUES (2,'João','Silva','teste@teste.com.br','12345689895','Rua Teste, 500, Centro','1234568954'),(7,'Márcia','C
46 ● /*!40000 ALTER TABLE `clientes` ENABLE KEYS */;
47 ● UNLOCK TABLES;
48
49 --
50 -- Table structure for table `produtos`
51 --
52
53 ● DROP TABLE IF EXISTS `produtos`;
54 ● /*!40101 SET @saved_cs_client = @@character_set_client */;
55 ● /*!50503 SET character_set_client = utf8mb4 */;
56 ● CREATE TABLE `produtos` (
57   `id` int(11) NOT NULL AUTO_INCREMENT,
58   `nome` varchar(45) NOT NULL,
59   `descricao` text NOT NULL,
60   `codigo_barras` varchar(45) NOT NULL,
61   `preco_custo` decimal(8,2) NOT NULL,
62   `preco_venda` decimal(8,2) NOT NULL,
63   `fornecedor` varchar(45) NOT NULL,
64   `quantidade` int(11) NOT NULL,
65   PRIMARY KEY (`id`),
66   UNIQUE KEY `codBarras_UNIQUE` (`codigo_barras`)
67 ) ENGINE=InnoDB AUTO_INCREMENT=19 DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
68 ● /*!40101 SET character_set_client = @saved_cs_client */;
69
70 --
71 -- Dumping data for table `produtos`
72 --
73
74 ● LOCK TABLES `produtos` WRITE;
75 ● /*!40000 ALTER TABLE `produtos` DISABLE KEYS */;
76 ● INSERT INTO `produtos` VALUES (14,'Ibuprofeno','Anti-inflamatório','12345',5.00,10.00,'Fornecedor 1',42),(15,'Nimesulida','Anti-inflamatór
77 ● /*!40000 ALTER TABLE `produtos` ENABLE KEYS */;
78 ● UNLOCK TABLES;

```

Figura 4. Códigos do BD - MySQL

Fonte: Autores (2023)

```

79
80 --
81 -- Table structure for table `produtos_venda`
82 --
83
84 ● DROP TABLE IF EXISTS `produtos_venda`;
85 ● /*!40101 SET @saved_cs_client = @@character_set_client */;
86 ● /*!50503 SET character_set_client = utf8mb4 */;
87 ● CREATE TABLE `produtos_venda` (
88     `id` int(11) NOT NULL AUTO_INCREMENT,
89     `venda_id` int(11) NOT NULL,
90     `produto_id` int(11) NOT NULL,
91     `preco_venda` decimal(8,2) NOT NULL,
92     `quantidade` int(11) NOT NULL,
93     `preco_total` decimal(8,2) NOT NULL,
94     PRIMARY KEY (`id`),
95     KEY `venda_id` (`venda_id`),
96     KEY `produto_id` (`produto_id`),
97     CONSTRAINT `produtos_venda_ibfk_1` FOREIGN KEY (`venda_id`) REFERENCES `vendas` (`id`),
98     CONSTRAINT `produtos_venda_ibfk_2` FOREIGN KEY (`produto_id`) REFERENCES `produtos` (`id`)
99 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
100 ● /*!40101 SET character_set_client = @saved_cs_client */;
101
102 --
103 -- Dumping data for table `produtos_venda`
104 --
105
106 ● LOCK TABLES `produtos_venda` WRITE;
107 ● /*!40000 ALTER TABLE `produtos_venda` DISABLE KEYS */;
108 ● /*!40000 ALTER TABLE `produtos_venda` ENABLE KEYS */;
109 ● UNLOCK TABLES;
110
111 --
112 -- Table structure for table `usuarios`
113 --
114
115 ● DROP TABLE IF EXISTS `usuarios`;
116 ● /*!40101 SET @saved_cs_client = @@character_set_client */;
117 ● /*!50503 SET character_set_client = utf8mb4 */;
118 ● CREATE TABLE `usuarios` (

```

Figura 5. Códigos do BD - MySQL

Fonte: Autores (2023)

```

118 ● CREATE TABLE `usuarios` (
119     `id` int(11) NOT NULL AUTO_INCREMENT,
120     `nome` varchar(45) NOT NULL,
121     `sobrenome` varchar(45) NOT NULL,
122     `nome_login` varchar(45) NOT NULL,
123     `senha` varchar(45) NOT NULL,
124     `cpf` varchar(11) NOT NULL,
125     `email` varchar(100) NOT NULL,
126     `telefone` varchar(20) NOT NULL,
127     `endereco` varchar(100) NOT NULL,
128     PRIMARY KEY (`id`),
129     UNIQUE KEY `email` (`email`),
130     UNIQUE KEY `nomeDeUsuario` (`nome_login`),
131     UNIQUE KEY `cpf` (`cpf`)
132 ) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
133 ● /*!40101 SET character_set_client = @saved_cs_client */;
134
135 --
136 -- Dumping data for table `usuarios`
137 --
138
139 ● LOCK TABLES `usuarios` WRITE;
140 ● /*!40000 ALTER TABLE `usuarios` DISABLE KEYS */;
141 ● INSERT INTO `usuarios` VALUES (1,'Mara','Vicentina','mara','123456','12356985654','marav@gmail.com','19856325241','Rua Joaquim dos Santos
142 ● /*!40000 ALTER TABLE `usuarios` ENABLE KEYS */;
143 ● UNLOCK TABLES;
144
145 --
146 -- Table structure for table `vendas`
147 --
148
149 ● DROP TABLE IF EXISTS `vendas`;
150 ● /*!40101 SET @saved_cs_client = @@character_set_client */;
151 ● /*!50503 SET character_set_client = utf8mb4 */;
152 ● CREATE TABLE `vendas` (
153     `id` int(11) NOT NULL AUTO_INCREMENT,
154     `usuario_id` int(11) NOT NULL,
155     `cliente_id` int(11) NOT NULL,
156     `total` decimal(8,2) NOT NULL DEFAULT 0.00,
157     `data_venda` datetime NOT NULL,

```

Figura 6. Códigos do BD - MySQL

Fonte: Autores (2023)

```

157     `data_venda` datetime NOT NULL,
158     PRIMARY KEY (`id`),
159     KEY `usuario_id` (`usuario_id`),
160     KEY `cliente_id` (`cliente_id`),
161     CONSTRAINT `vendas_ibfk_1` FOREIGN KEY (`usuario_id`) REFERENCES `usuarios` (`id`),
162     CONSTRAINT `vendas_ibfk_2` FOREIGN KEY (`cliente_id`) REFERENCES `clientes` (`id`)
163 } ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
164 ● /*!40101 SET character_set_client = @saved_cs_client */;
165
166 --
167 -- Dumping data for table `vendas`
168 --
169
170 ● LOCK TABLES `vendas` WRITE;
171 ● /*!40000 ALTER TABLE `vendas` DISABLE KEYS */;
172 ● INSERT INTO `vendas` VALUES (1,1,7,120.00,'2023-04-11 19:47:07'),(2,1,9,80.00,'2023-04-11 19:57:27'),(3,1,9,80.00,'2023-04-11 19:59:23');
173 ● /*!40000 ALTER TABLE `vendas` ENABLE KEYS */;
174 ● UNLOCK TABLES;
175
176 --
177 -- Dumping routines for database `bdfarmacia`
178 --
179 ● /*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
180
181 ● /*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
182 ● /*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
183 ● /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
184 ● /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
185 ● /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
186 ● /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
187 ● /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
188
189 -- Dump completed on 2023-04-12 11:54:28
190

```

Figura 7. Códigos do BD - MySQL

Fonte: Autores (2023)

### 3.2. PROGRAMAÇÃO ORIENTADA A OBJETOS

A programação orientada a objetos é uma área da programação que se concentra na criação de objetos que podem interagir entre si para resolver um problema ou realizar uma tarefa. Para realização deste projeto utilizamos a linguagem Java, que é voltada para a programação orientada a objetos, projetada para ser portátil, ou seja, pode ser executada em diferentes plataformas, como Windows, Linux e Mac OS. Ela foi criada na década de 1990 pela Sun Microsystems e hoje em dia é mantida pela Oracle Corporation.

Para tal utilizamos o NetBeans que é um ambiente de desenvolvimento integrado (IDE - Integrated Development Environment) de código aberto que permite criar aplicativos em várias linguagens de programação, incluindo Java, PHP, C ++, HTML5 e outros.

### 3.2.1. DIAGRAMA DE CLASSES

O Diagrama de Classes é uma representação gráfica da estrutura e das relações entre as classes de um sistema orientado a objetos. Ele é uma das principais ferramentas utilizadas na modelagem de sistemas, sendo amplamente utilizado em engenharia de software.

O diagrama de classes é útil para entender a estrutura e o comportamento de um sistema orientado a objetos, e pode ser usado para identificar classes, propriedades e métodos que são necessários para a implementação do sistema. Ele também pode ser usado para documentar um sistema existente e para comunicar a estrutura do sistema para outros membros da equipe de desenvolvimento.

O diagrama de classes utilizado em nosso projeto pode ser visualizado na figura 8.

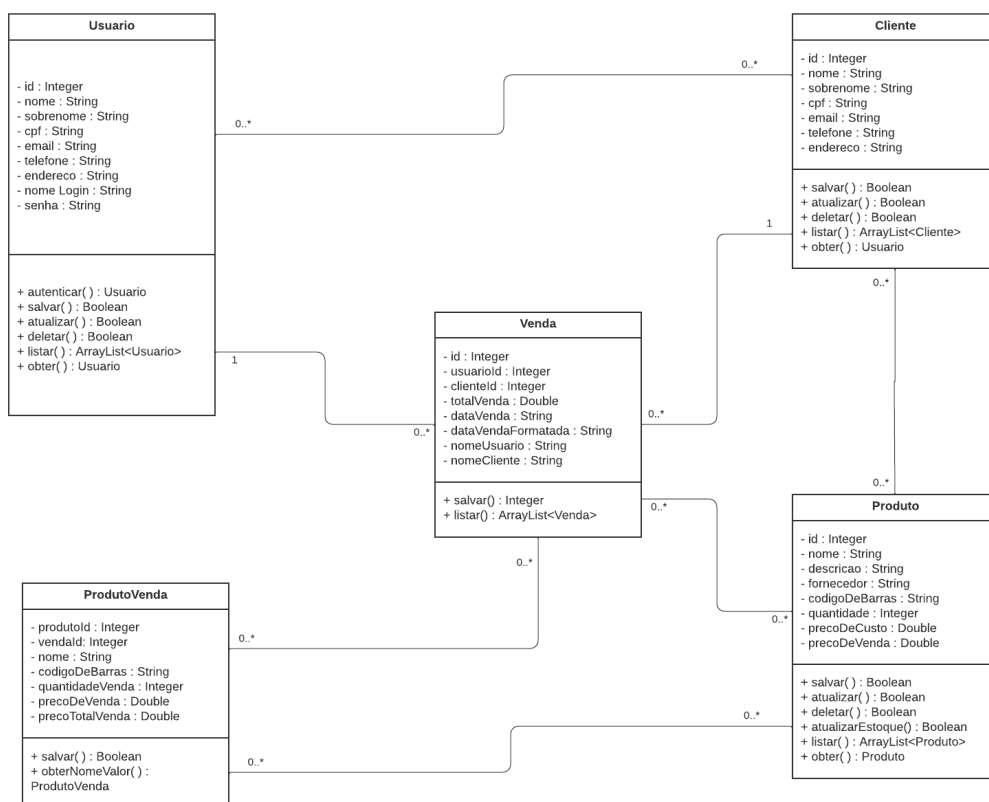


Figura 8. Diagrama de Classes do projeto.

Fonte: Autores (2023)

### 3.2.2. CÓDIGOS DO SISTEMA

Os códigos desenvolvidos no main.java, na model e na controller estão exibidos nas figuras de 9 a 24. Para mais informações sobre os códigos desenvolvidos está disponível no repositório que pode ser acessado através do link abaixo:

[https://github.com/mara-vicentina/sistema\\_farma\\_facil.git](https://github.com/mara-vicentina/sistema_farma_facil.git)

O sql necessário para geração do banco de dados com as tabelas desenvolvidas foi inserido dentro do arquivo README.md

A conexão com o banco de dados se encontra no pacote com.mycompany.projeto\_pi\_farmacia.utils. Para realizar o primeiro login no sistema, já deixamos um usuário cadastrado no banco de dados com o usuário sendo admin e a senha 1234.

```
import com.mycompany.projeto_pi_farmacia.controller.UsuarioController;
import com.mycompany.projeto_pi_farmacia.model.Usuario;
import com.sun.glass.events.KeyEvent;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;

public class TelaLogin extends javax.swing.JFrame {
    public TelaInicial telaInicial = new TelaInicial();
    public static Usuario usuarioLogado = null;

    public TelaLogin() {
        initComponents();
    }
}
```

Figura 9. Códigos do main.java

Fonte: Autores (2023)

```
private void formKeyPressed(java.awt.event.KeyEvent evt) {
    switch (evt.getKeyCode()) {
        case com.sun.glass.events.KeyEvent.VK_ESCAPE:
            this.setVisible(b: false);
            break;
    }
}

private void botaoEntrarActionPerformed(java.awt.event.ActionEvent evt) {
    String nomeLogin = campoUsuario.getText();
    String senha = campoSenha.getText();

    Usuario usuario = new UsuarioController().autenticarUsuario(nomeLogin, senha);

    if (usuario != null) {
        this.usuarioLogado = usuario;
        this.telaInicial.setVisible(b: true);
        this.setVisible(b: false);
    } else {
        JOptionPane.showMessageDialog(parentComponent: this, message: "Usuário ou senha incorreto(s).");
    }
}
```



Figura 10. Códigos do main.java

Fonte: Autores (2023)

```
public static void main(String args[]) {  
    java.awt.EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            try {  
                UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());  
                new TelaLogin().setVisible(true);  
            } catch (ClassNotFoundException ex) {  
                Logger.getLogger(name: TelaLogin.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);  
            } catch (InstantiationException ex) {  
                Logger.getLogger(name: TelaLogin.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);  
            } catch (IllegalAccessException ex) {  
                Logger.getLogger(name: TelaLogin.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);  
            } catch (UnsupportedLookAndFeelException ex) {  
                Logger.getLogger(name: TelaLogin.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);  
            }  
        }  
    });  
}
```

Figura 11. Códigos do main.java

Fonte: Autores (2023)

```
package com.mycompany.projeto_pi_farmacia.model;  
  
import com.mycompany.projeto_pi_farmacia.utils.DataBase;  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.ArrayList;  
  
public class Usuario {  
    private Integer id;  
    private String nome;  
    private String sobrenome;  
    private String cpf;  
    private String email;  
    private String telefone;  
    private String endereco;  
    private String nomeLogin;  
    private String senha;  
}
```

Figura 12. Códigos model usuário

Fonte: Autores (2023)

```

public Usuario(
    Integer id,
    String nome,
    String sobrenome,
    String cpf,
    String email,
    String telefone,
    String endereco,
    String nomeLogin,
    String senha) {
    this.id = id;
    this.nome = nome;
    this.sobrenome = sobrenome;
    this.cpf = cpf;
    this.email = email;
    this.telefone = telefone;
    this.endereco = endereco;
    this.nomeLogin = nomeLogin;
    this.senha = senha;
}

public Integer getId() {
    return id;
}

```

Figura 13. Códigos model usuário

Fonte: Autores (2023)

```

public void setId(Integer id) {
    this.id = id;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public String getSobrenome() {
    return sobrenome;
}

public void setSobrenome(String sobrenome) {
    this.sobrenome = sobrenome;
}

public String getCpf() {
    return cpf;
}

```

Figura 14. Códigos model usuário

Fonte: Autores (2023)

```

public void setCpf(String cpf) {
    this.cpf = cpf;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getTelefone() {
    return telefone;
}

public void setTelefone(String telefone) {
    this.telefone = telefone;
}

public String getEndereco() {
    return endereco;
}

```

Figura 15. Códigos model usuário  
 Fonte: Autores (2023)

```

public void setEndereco(String endereco) {
    this.endereco = endereco;
}

public String getNomeLogin() {
    return nomeLogin;
}

public void setNomeLogin(String nomeLogin) {
    this.nomeLogin = nomeLogin;
}

public String getSenha() {
    return senha;
}

public void setSenha(String senha) {
    this.senha = senha;
}

```

Figura 16. Códigos model usuário  
 Fonte: Autores (2023)

```

public static Usuario autenticar(String nomeLogin, String senha) {
    String query = String.format(
        format:"SELECT * FROM usuarios WHERE nome_login = '%s' AND senha = '%s';",
        args: nomeLogin, args: senha);
    Connection connection = new DataBase().abrirConexao();

    try {
        PreparedStatement ps = connection.prepareStatement(sql:query);
        ResultSet rs = ps.executeQuery();
        Boolean retornoLogin = rs.next();
        Usuario usuarioAutenticado = new Usuario(
            id: rs.getInt(columnLabel: "id"),
            nome: rs.getString(columnLabel: "nome"),
            sobrenome: rs.getString(columnLabel: "sobrenome"),
            cpf: rs.getString(columnLabel: "cpf"),
            email: rs.getString(columnLabel: "email"),
            telefone: rs.getString(columnLabel: "telefone"),
            endereco: rs.getString(columnLabel: "endereco"),
            nomeLogin: rs.getString(columnLabel: "nome_login"),
            senha: rs.getString(columnLabel: "senha"));
        connection.close();

        return usuarioAutenticado;
    } catch (SQLException ex) {
        System.err.println(x: ex.getMessage());
        System.err.println(x: "Erro ao fazer Login.");
    }
    return null;
}

```

Figura 17. Códigos model usuário

Fonte: Autores (2023)

```

public Boolean deletar() {
    String query = String.format(
        format:"DELETE FROM usuarios WHERE id = %s;", args: this.getId());
    System.out.println(x: query);
    Connection connection = new DataBase().abrirConexao();

    try {
        PreparedStatement ps = connection.prepareStatement(sql:query);
        ps.execute();
        connection.close();
        return true;
    } catch (SQLException ex) {
        System.err.println(x: "Erro ao deletar usuário.");
    }

    return false;
}

```

Figura 18. Códigos model usuário

Fonte: Autores (2023)

```

public static ArrayList<Usuario> listar() {
    ArrayList<Usuario> usuarios = new ArrayList<>();

    String query = String.format("SELECT * FROM usuarios;");
    System.out.println(x: query);
    Connection connection = new DataBase().abrirConexao();
    try {
        PreparedStatement ps = connection.prepareStatement(sql: query);
        ResultSet rs = ps.executeQuery();
        while(rs.next()) {
            Usuario usuario = new Usuario(
                id: rs.getInt(columnLabel: "id"),
                nome: rs.getString(columnLabel: "nome"),
                sobrenome: rs.getString(columnLabel: "sobrenome"),
                cpf: rs.getString(columnLabel: "cpf"),
                email: rs.getString(columnLabel: "email"),
                telefone: rs.getString(columnLabel: "telefone"),
                endereco: rs.getString(columnLabel: "endereco"),
                nomeLogin: rs.getString(columnLabel: "nome_login"),
                senha: rs.getString(columnLabel: "senha")
            );
            usuarios.add(e: usuario);
        }
        connection.close();
    } catch (SQLException ex) {
        System.err.println(x: "Erro ao listar usuarios.");
    }
    return usuarios;
}

```

Figura 19. Códigos model usuário

Fonte: Autores (2023)

```

public static Usuario obter(Integer id) {
    Usuario usuario = null;
    String query = String.format(
        format: "SELECT * FROM usuarios WHERE id = %s;", args: id);
    System.out.println(x: query);
    Connection connection = new DataBase().abrirConexao();

    try {
        PreparedStatement ps = connection.prepareStatement(sql: query);
        ResultSet rs = ps.executeQuery();
        rs.next();
        usuario = new Usuario(
            id: rs.getInt(columnLabel: "id"),
            nome: rs.getString(columnLabel: "nome"),
            sobrenome: rs.getString(columnLabel: "sobrenome"),
            cpf: rs.getString(columnLabel: "cpf"),
            email: rs.getString(columnLabel: "email"),
            telefone: rs.getString(columnLabel: "telefone"),
            endereco: rs.getString(columnLabel: "endereco"),
            nomeLogin: rs.getString(columnLabel: "nome_login"),
            senha: rs.getString(columnLabel: "senha")
        );
        connection.close();
        return usuario;
    } catch (SQLException ex) {
        System.err.println(x: ex.getMessage());
        System.err.println(x: "Erro ao obter usuário.");
    }
    return null;
}

```

Figura 20. Códigos model usuário

Fonte: Autores (2023)

```

package com.mycompany.projeto_pi_farmacia.controller;

import com.mycompany.projeto_pi_farmacia.model.Usuario;
import java.util.ArrayList;

public class UsuarioController {

    public Usuario autenticarUsuario(String nomeLogin, String senha) {
        try {
            return Usuario.autenticar(nomeLogin, senha);
        } catch (Exception ex) {
            return null;
        }
    }
}

```

Figura 21. Códigos controller usuário

Fonte: Autores (2023)

```

public Boolean inserirUsuario(
    String nome,
    String sobrenome,
    String cpf,
    String email,
    String telefone,
    String endereco,
    String nomeLogin,
    String senha) {
    try {
        Usuario usuario = new Usuario(
            id: null, nome,
            sobrenome,
            cpf,
            email,
            telefone,
            endereco,
            nomeLogin,
            senha);
        return usuario.salvar();
    } catch (Exception ex) {
        return false;
    }
}

```

Figura 22. Códigos controller usuário

Fonte: Autores (2023)

```

public Boolean atualizarUsuario(
    Integer id,
    String nome,
    String sobrenome,
    String cpf,
    String email,
    String telefone,
    String endereco,
    String nomeLogin,
    String senha) {
    try {
        Usuario usuario = Usuario.obter(id);
        usuario.setNome(nome);
        usuario.setSobrenome(sobrenome);
        usuario.setCpf(cpf);
        usuario.setEmail(email);
        usuario.setTelefone(telefone);
        usuario.setEndereco(endereco);
        usuario.setNomeLogin(nomeLogin);
        usuario.setSenha(senha);
        return usuario.atualizar();
    } catch (Exception ex) {
        return false;
    }
}

```

Figura 23. Códigos controller usuario

Fonte: Autores (2023)

```

public Boolean removerUsuario(Integer id) {
    try {
        Usuario usuario = Usuario.obter(id);
        return usuario.deletar();
    } catch (Exception ex) {
        return false;
    }
}

public ArrayList<Usuario> listarUsuarios() {
    try {
        return Usuario.listar();
    } catch (Exception ex) {
        return new ArrayList<>();
    }
}

```

Figura 24. Códigos controller usuario

Fonte: Autores (2023)



### 3.2.3 IMAGENS DO SISTEMA

A figura 25 representa a tela de login na qual realiza-se a autenticação do usuário.



Figura 25 - Tela de login do sistema.

Fonte: Autores (2023)

A partir do login do usuário temos uma tela, figura 26, na qual apresenta-se as opções do sistema relacionadas ao usuário, clientes, produtos e vendas.

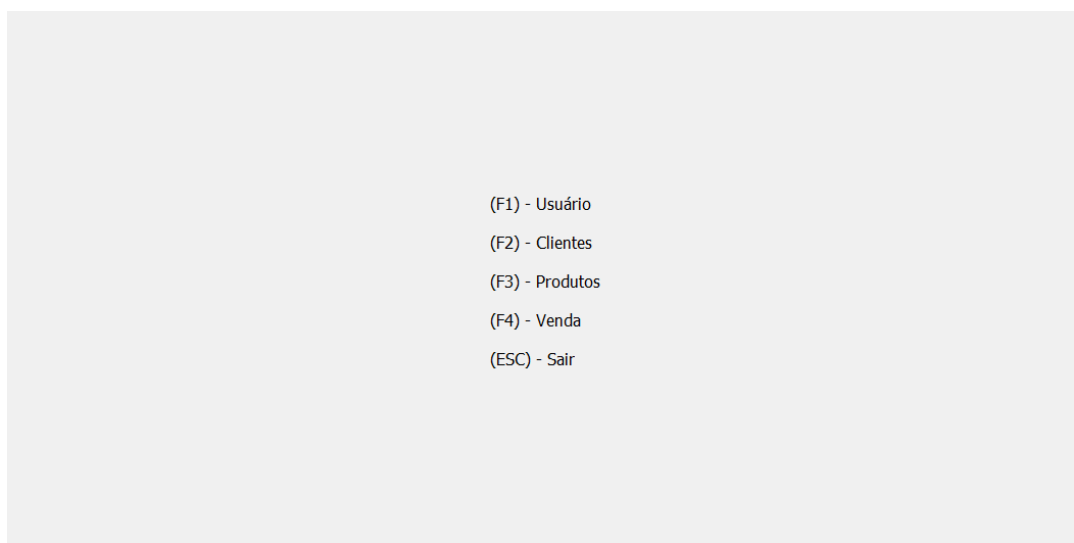


Figura 26-. Tela de opções do sistema.

Fonte: Autores (2023)

Quando o usuário aperta a tecla F1 referente ao usuário exibe-se a tela de usuários, figura 27, na qual apresenta-se uma listagem dos usuários cadastrados no sistema, com opções para cadastrar um novo usuário, editar um usuário já existente e excluir um usuário já existente.

(ESC) - Sair

**Tela de Usuários**

ID	Nome	Sobrenome	CPF	E-mail	Telefone	Endereço	Nome Login	Senha
1	Mara	Vicentina	12356985654	marav@gmail.c...	19856325241	Rua Joaquim d...	mara	123456
2	Joana	Silva	123458975	joana@hotmail...	15263879452	Rua Matielo, 62...	joana	1234
3	Silvia	Trindade	52639857462	silvia@hotmail...	15263885236	Rua Matielo, 68...	silvia	321
4	Gilberto	Trindade	52639875421	gilberto@hotma...	15263836258	Rua Matielo, 68...	gilberto	123456
5	Alberto	Filho	52639873652	alberto@hotmai...	15263835263	Rua Matielo, 99...	alberto	12345

Figura 27. Tela de usuários do sistema.

Fonte: Autores (2023)

Na figura 28 apresenta-se a tela de cadastro do usuário com campos para preenchimento de nome, sobrenome, CPF, e-mail, telefone, endereço, nome de usuário e senha. A partir desse preenchimento salva-se os dados do usuário e retorna para a listagem anterior, figura 27.

(ESC) - Sair

**Cadastro de Usuário**

Nome  Sobrenome   
 CPF  E-mail   
 Telefone  Endereço   
 Nome de Usuário  Senha

Figura 28. Tela de cadastro de usuário.

Fonte: Autores (2023)

Quando o usuário já foi exibido na listagem, quando seleciona-se uma linha da tabela há a opção de editar os dados referentes a ele. Na imagem 29 observa-se que os dados são obtidos para cada campo de edição, conforme o cadastro feito anteriormente, a partir da edição dos campos, salva-se os dados editados e atualiza-se a tabela referente a listagem dos usuários.

(ESC) - Sair

### Edição de Usuário

Nome	Sobrenome
<input type="text" value="Joana"/>	<input type="text" value="Silva"/>
CPF	E-mail
<input type="text" value="123458975"/>	<input type="text" value="joana@hotmail.com"/>
Telefone	Endereço
<input type="text" value="15263879452"/>	<input type="text" value="Rua Matielo, 620, Centro"/>
Nome de Usuário	Senha
<input type="text" value="joana"/>	<input type="text" value="1234"/>

Figura 29. Tela de edição do usuário.

Fonte: Autores (2023)

Quando o usuário aperta a tecla F2 referente aos clientes exibe-se a tela de clientes, figura 30, na qual apresenta-se uma listagem dos clientes cadastrados no sistema, com opções para cadastrar um novo cliente, editar um cliente já existente e excluir um cliente já existente.

(ESC) - Sair

### Tela de Clientes

ID	Nome	Sobrenome	CPF	E-mail	Telefone	Endereço
2	João	Silva	12345689895	teste@teste.com.br	1234568954	Rua Teste, 500, Centro
7	Marília	Carvalho	42281844800	marilia.carvalho@hotmail.com	19991331100	Rua Cesar, 215, Centro
9	Mara	Vicentina	46985632145	mara@teste.com.br	789654258	rua teste, teste
10	Maria	Beraldo	36529856832	teste.teste@teste.com	13265874521	rua João Castro, 500, Centro

Figura 30. Tela de clientes.

Fonte: Autores (2023)

Na imagem 31 apresenta-se a tela de cadastro do cliente com campos para preenchimento de nome, sobrenome, CPF, e-mail, telefone e endereço. A partir desse preenchimento salva-se os dados do cliente e é exibido na listagem na tela anterior, figura 30, supracitada.



(ESC) - Sair

**Cadastro de Cliente**

Nome	Sobrenome
<input type="text"/>	<input type="text"/>
CPF	Email
<input type="text"/>	<input type="text"/>
Telefone	Endereço
<input type="text"/>	<input type="text"/>

Figura 31. Tela de cadastro do cliente.

Fonte: Autores (2023)

Quando o cliente já foi exibido na listagem, quando seleciona-se uma linha da tabela há a opção de editar os dados referentes a ele. Na figura 32 observa-se que os dados são obtidos para cada campo de edição, conforme o cadastro feito anteriormente, a partir da edição dos campos, salva-se os dados editados e atualiza-se a tabela referente a listagem dos clientes.



(ESC) - Sair

**Edição de Cliente**

Nome	Sobrenome
<input type="text" value="João"/>	<input type="text" value="Silva"/>
CPF	Email
<input type="text" value="12345689895"/>	<input type="text" value="teste@teste.com.br"/>
Telefone	Endereço
<input type="text" value="1234568954"/>	<input type="text" value="Rua Teste, 500, Cent"/>

Figura 32. Tela de edição do cliente.

Fonte: Autores (2023)

Quando o usuário aperta a tecla F3 referente aos produtos exibe-se a tela de produtos, figura 33, na qual apresenta-se uma listagem dos produtos cadastrados no sistema, com opções para cadastrar um novo produto, editar um produto já existente e excluir um produto já existente.



ID	Nome	Descrição	Código de Barras	Quantidade	Fornecedor	Preço de Custo	Preço de Venda
14	Ibuprofeno	Anti-inflamatório	12345	50	Fornecedor 1	5.0	10.0
15	Nimesulida	Anti-inflamatório	123456	60	Fornecedor 2	7.0	12.0
16	Diclofenaco	Anti-inflamatório	1234567	65	Fornecedor 3	8.0	15.0
17	Naproxeno	Anti-inflamatório	12345678	45	Fornecedor 4	4.0	16.0
18	Amoxicilina teste	Antibiótico teste	1234567898	52	Fornecedor 4	31.0	51.0

Figura 33. Tela de produtos.

Fonte: Autores (2023)

Na figura 34 apresenta-se a tela de cadastro dos produtos com campos para preenchimento de nome, código de barras, descrição, quantidade, fornecedor, preço de custo e preço de venda. A partir desse preenchimento salva-se os dados do produto e é exibido na listagem na tela anterior referente a figura 33.



(ESC) - Sair

### Cadastro de Produto

Nome  Código de Barras

Descrição

Quantidade  Fornecedor

Preço de Custo  Preço de Venda

Figura 34-. Tela de cadastro do produto.

Fonte: Autores (2023)

Quando o produto já foi exibido na listagem, quando seleciona-se uma linha da tabela há a opção de editar os dados referentes a ele. Na figura 35 observa-se que os dados são obtidos para cada campo de edição, conforme o cadastro feito anteriormente, a partir da edição dos campos, salva-se os dados editados e atualiza-se a tabela referente a listagem dos produtos.

(ESC) - Sair

### Edição de Produto

Nome: Nimesulida      Código de Barras: 123456

Descrição: Anti-inflamatório

Quantidade: 60      Fornecedor: Fornecedor 2

Preço de Custo: 7.0      Preço de Venda: 12.0

Salvar      Cancelar

Figura 35. Tela de edição do produto.

Fonte: Autores (2023)

Quando o usuário aperta a tecla F4 referente às vendas exibe-se a tela de Vendas, figura 36, na qual apresenta-se uma listagem das vendas cadastradas no sistema, com opção para realizar uma venda.

(ESC) - Sair

### Tela de Vendas

Realizar Venda

ID	Nome do Cliente	Venda Realizada pelo Usuario	Total da Venda	Data de Venda
14	João Silva	Mara Vicentina	128.0	13/04/2023 00:09:26
13	João Silva	Mara Vicentina	80.0	13/04/2023 00:06:26
12	João Silva	Mara Vicentina	108.0	12/04/2023 23:37:12
9	João Silva	Mara Vicentina	24.0	12/04/2023 21:51:48
4	João Silva	Mara Vicentina	96.0	12/04/2023 15:09:58
3	Mara Vicentina	Mara Vicentina	80.0	11/04/2023 19:59:23
2	Mara Vicentina	Mara Vicentina	80.0	11/04/2023 19:57:27
1	Marilia Carvalho	Mara Vicentina	120.0	11/04/2023 19:47:07

Figura 36. Tela de venda.

Fonte: Autores (2023)

A última tela do sistema é a tela de venda de produtos na qual insere-se o código de barras e a quantidade de venda do produto na tabela, seleciona-se o cliente e a venda é finalizada. Nessa tela também já consta o valor total da venda de acordo com os produtos adicionados na tabela.

Código de Barras	Nome do Produto	Quantidade	Valor Unitário	Valor Total
123456	Nimesulida	5	12.0	60.0

Figura 37. Tela de venda dos produtos.

Fonte: Autores (2023)

## 4. CONTEÚDO DA FORMAÇÃO PARA A VIDA: CRIANDO O NOVO

Formação para a vida é um processo contínuo de aprendizagem e desenvolvimento que capacita indivíduos para lidar com os desafios e oportunidades da vida de maneira eficaz e saudável. Essa formação inclui habilidades cognitivas, sociais e emocionais, bem como conhecimentos práticos e técnicos. Algumas das habilidades essenciais incluem resolução de problemas, comunicação, trabalho em equipe, pensamento crítico, criatividade, autoconhecimento e gestão de emoções. Além disso, é importante ter conhecimento sobre questões sociais, ambientais e culturais relevantes para a vida em sociedade. A formação para a vida é uma responsabilidade compartilhada entre família, escola, comunidade e indivíduo e pode ser promovida através de diversas atividades educacionais, como cursos, programas extracurriculares, experiências práticas e oportunidades de voluntariado.

### 4.1. ESTUDANTES NA PRÁTICA – LINK DO VÍDEO

Nesta etapa foi produzido um vídeo no formato de pitch de vendas, onde o grupo apresenta o projeto a ser utilizado pelo cliente alvo. Segue o link do vídeo:

PROJETO INTEGRADO - ADS - 2023 - FARMAFÁCIL - <https://youtu.be/1nAaqC6umxI>



## 5. CONCLUSÃO

Este trabalho mostrou a importância contínua dos sistemas desktop na era digital e como eles ainda são fundamentais para muitas empresas e em especial as de pequeno porte.

Além disso, o trabalho apresentou a aplicação prática de um sistema desktop para uma farmácia de pequeno porte, demonstrando como a tecnologia pode ajudar a otimizar processos e aumentar a eficiência em negócios. A implementação desse sistema proporciona uma gestão mais eficiente do negócio, o que pode resultar em melhores resultados financeiros e satisfação dos clientes.

Em resumo, a evolução tecnológica nos últimos anos tem trazido diversas mudanças para o mundo dos negócios, e a importância dos sistemas desktop ainda é inegável. A tecnologia continua a avançar, e é fundamental que as empresas se mantenham atualizadas para aproveitar ao máximo as possibilidades que a tecnologia oferece.

## 6. REFERÊNCIAS

- Beaulieu, A. (2014). Aprendendo SQL (2ª ed.). O'Reilly Media.
- Date, C. J. (2015). Introdução a sistemas de bancos de dados (8ª ed.). Campus.
- Deitel, P. J., & Deitel, H. M. (2016). Java: como programar (10ª ed.). Pearson.
- Elmasri, R., & Navathe, S. B. (2016). Sistemas de banco de dados (6ª ed.). Pearson.
- Nunes, M. A. (2017). Desenvolvimento de Aplicações Web com JSF, Spring Framework e Hibernate. Brasport.
- Oracle. (2021). Tecnologias Java SE. Recuperado em 04 de abril de 2023, de <https://www.oracle.com/br/java/technologies/>
- Queiroz, G. A. (2019). Programação Java para a Web. Editora Érica.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2012). Sistema de Banco de Dados (6ª ed.). Campus.
- Siouto, É. M. (2016). Programação Orientada a Objetos com Java: uma introdução prática utilizando BlueJ. Novatec Editora.
- Souza, V. L. (2016). Modelagem conceitual de banco de dados. Editora Ciência Moderna.

## RELATÓRIO FINAL DAS ATIVIDADES DE EXTENSÃO

### 1. IDENTIDADE DA ATIVIDADE

**RELATÓRIO:** Relatório do Projeto Integrado

**CURSO:** Análise e Desenvolvimento de Sistemas e Gestão de Tecnologia da Informação

**MÓDULO:** Desenvolvimento Desktop

**PROFESSOR RESPONSÁVEL:** Sidney Gitcoff Telles

**ESTUDANTE:** Se encontra no tópico 3 do relatório

**PERÍODO DE REALIZAÇÃO:** 02/2023 a 04/2023

### 2. DESENVOLVIMENTO

#### Contextualização

Nesse trimestre foi proposto para o projeto integrado a realização de um sistema desktop, a partir disso iniciamos a procura de um possível cliente, sendo escolhida uma farmácia. O cliente relatou um acidente meteorológico que causou problemas com as anotações não digitais e resultou na perda de 30% do estoque. Foi oferecida uma proposta de um programa mais útil e organizado para melhorar a gestão da farmácia.

#### Desafio

O desafio nesse trimestre foi desenvolver um sistema, que incluísse atividades básicas de uma empresa, como controle de produtos, entrada e saída, controle de vendas, no caso em questão de uma farmácia.

#### Cronograma das Ações

- Iniciamos com uma reunião para decidir qual o público alvo do sistema e levar a ideia ao possível cliente
- Montagem das telas e dos dados necessários a ela
- Criação do banco de dados
- Conexão entre o sistema e o banco
- Finalização do projeto para entrega

#### Síntese das Ações

A criação de um sistema para farmácias envolve várias etapas, como a definição do público-alvo, o levantamento dos requisitos e a montagem das telas e dados. Em seguida, é preciso criar um banco de dados e fazer a conexão entre o sistema e o banco. Por fim, o projeto é finalizado e entregue ao cliente. Essas ações são essenciais para garantir que o sistema atenda às necessidades do negócio e funcione de forma eficiente.

#### a. Aspectos positivos

Conforme fomos realizando o estudo dos conteúdos e realizando as aulas ao vivo com o professor conseguimos dar andamento no projeto.

#### b. Dificuldades encontradas

Encontramos mais dificuldades em relação aos horários de disponibilidade para reuniões em grupo, todas fizemos online por ter integrantes de outras cidades.

Em relação ao projeto não houve muitas dificuldades pois fomos organizando as etapas do projeto.

c. Resultados atingidos  
Conseguimos realizar e concluir o desenvolvimento de um sistema Desktop voltado para farmácias. Portanto colocou-se em prática os conteúdos de banco de dados e programação orientada a objetos estudados na plataforma durante o trimestre.

d. Sugestões / Outras observações  
Poderia haver mais orientações em relação ao preenchimento de alguns dados do documento. Com algum vídeo explicativo geral, ajudaria a esclarecer alguns pontos do documento.

### 3. EQUIPE DOS ESTUDANTES NO PROJETO

RA 1012023100245	NOME Bruno Domingos
RA 1012023100287	NOME Daniel dos Reis Germinaro
RA 1012023100141	NOME João Roberto Buffo Nogue
RA 1012023100321	NOME Mara Vicentina Pinto
RA 1012023100352 RA 1012023100191	NOME Murilo de Lima Santos NOME Rodolfo de Lima Peres

**Curso de Análise e Desenvolvimento de Sistemas e Gestão de Tecnologia da Informação**

**Módulo Desenvolvimento Desktop**

**Cronograma de Validação - Projeto Integrado**

<b>Unidade Estudo</b>	<b>Participação no Projeto</b>	<b>Data da Validação</b>
Banco de Dados	Construção do banco de dados com MER, DER E Físico.	29/03
Programação Orientada a Objetos	Desenvolvimento as telas e da parte lógica do sistema que conectará com o banco de dados	30/03
<b><u>Descrição do Projeto:</u></b> criar um sistema, um módulo reduzido, que seja utilizado em qualquer estabelecimento comercial ou empresarial. Esse sistema deverá contemplar atividades básicas da empresa, como controle de produtos, entrada e saída, controle de vendas, módulos menores que possam ser criados e executados neste trimestre.		