



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:

Adryelle P. F. Araujo, RA 1012023100409

Isabella Donizetti Dardi, RA 1012023100203

Leonardo Vitorino Figueiredo, RA 1012023100153

Lucas Almeida Barbosa, RA 1012023100368

Pedro Ricardo Nunes Oliveira, RA 1012022200918

SÃO JOÃO DA BOA VISTA, SP
ABRIL, 2023

SUMÁRIO

1 INTRODUÇÃO	4
2 DESCRIÇÃO DA EMPRESA	5
3 PROJETO DE CONSULTORIA EMPRESARIAL	6
3.1 BANCO DE DADOS	6
3.1.1 MER - MODELAGEM ENTIDADE RELACIONAMENTO	6
3.1.2 DER - DIAGRAMA ENTIDADE RELACIONAMENTO	6
3.1.3 FÍSICO	7
3.2 PROGRAMAÇÃO ORIENTADA A OBJETOS	12
3.2.1 DIAGRAMA DE CLASSES	12
3.2.2 CÓDIGOS DO SISTEMA	12
3.2.3 IMAGENS DO SISTEMA	14
3.3 CONTEÚDO DA FORMAÇÃO PARA A VIDA: CRIANDO O NOVO	21
3.3.1 CRIANDO O NOVO	21
3.3.2 ESTUDANTES NA PRÁTICA	22
4 CONCLUSÃO	23
REFERÊNCIAS	24
ANEXOS	25

1 INTRODUÇÃO

O projeto tem como objetivo desenvolver um gerenciamento de estoque e vendas para microempreendedores e pequenas empresas.

Será utilizado o conceito de programação orientada a objetos em Java e modelagem de banco de dados relacional em MySQL para concretização do atual trabalho.

Onde a programação orientada a objetos (POO) é um paradigma de programação que traz consigo as denominações de classe e objeto, no qual a classe é uma estrutura que possui atributos (que são como variáveis) e métodos (que são como funções), servindo como modelo para a criação de um objeto, que nada mais é que uma instância da classe, sendo capaz de receber, processar e enviar dados de forma autônoma. Dentro desse paradigma possui alguns conceitos, tais como: encapsulamento, abstração, herança e polimorfismo (VIANNA, 2013).

O encapsulamento é uma forma de segurança na classe, onde ele restringe o acesso a atributos e métodos por meio de palavras chaves. A abstração em POO é o conceito onde se cria uma classe onde não será possível criar um objeto a partir dela e sim classes filhas (o próximo conceito de herança deixará mais claro), logo essa classe servirá como um molde para as próximas (CARVALHO *et al*, 2012).

A herança em POO é exatamente o que a palavra significa: a capacidade de herdar coisas, no caso do paradigma é herdar características de outras classes (como atributos e métodos), enquanto o polimorfismo permite que as classes herdeiras possam usar um método com a mesma assinatura (isso é, com o mesmo nome), porém com comportamentos/implementações diferentes (RICARTE, 2001).

Enquanto o conceito de banco de dados é um conjunto de dados relacionados, segundo Machado (2014) *“um banco de dados possui as seguintes propriedades: é uma coleção lógica coerente de dados com um significado inerente; ele é projetado, construído e preenchido com valores de dados para um propósito específico; ele representa algum aspecto do mundo real, o qual é chamado de minimundo”*. O banco de dados é algo amplamente utilizado hoje em dia, principalmente por empresas, já que se faz muito necessário o armazenamento de dados.

O banco de dados possui três tipos de modelos, o modelo conceitual que é um esquema gráfico com os dados que serão armazenados, o modelo lógico descreve como os dados deverão ser armazenados e seus tipos de relacionamentos e por fim o modelo físico que é quando é escolhido o SGBD (sistema gerenciador de banco de dados) e implementa o banco por meio da linguagem (MACHADO, 2014).

2 DESCRIÇÃO DA EMPRESA

A empresa escolhida para o atual projeto foi a Sandra Sueli dos Santos Damaglio LTDA, inscrita sob o número de CNPJ 189.782.46.0001-06, localizada no endereço Rua Mato Grosso, nº 451, Vila Fleming - São João da Boa Vista.

A empresa atua no mercado de produtos elétricos, comercializando cabos, lâmpadas, módulos e espelhos para tomadas e interruptores, ventiladores, chuveiros etc.

Seu atual controle de estoque e vendas se baseia em planilhas do excel, um sistema pouco automatizado para tal tarefa, podendo trazer falhas.

3 PROJETO DE CONSULTORIA EMPRESARIAL

Para o banco de dados foi utilizado o MySQL, que usa a linguagem SQL, e o ambiente de desenvolvimento foi o Workbench. Enquanto para as interfaces para cadastramento de dados e para lógica de programação, foi utilizada a linguagem de programação Java e a IDE Netbeans.

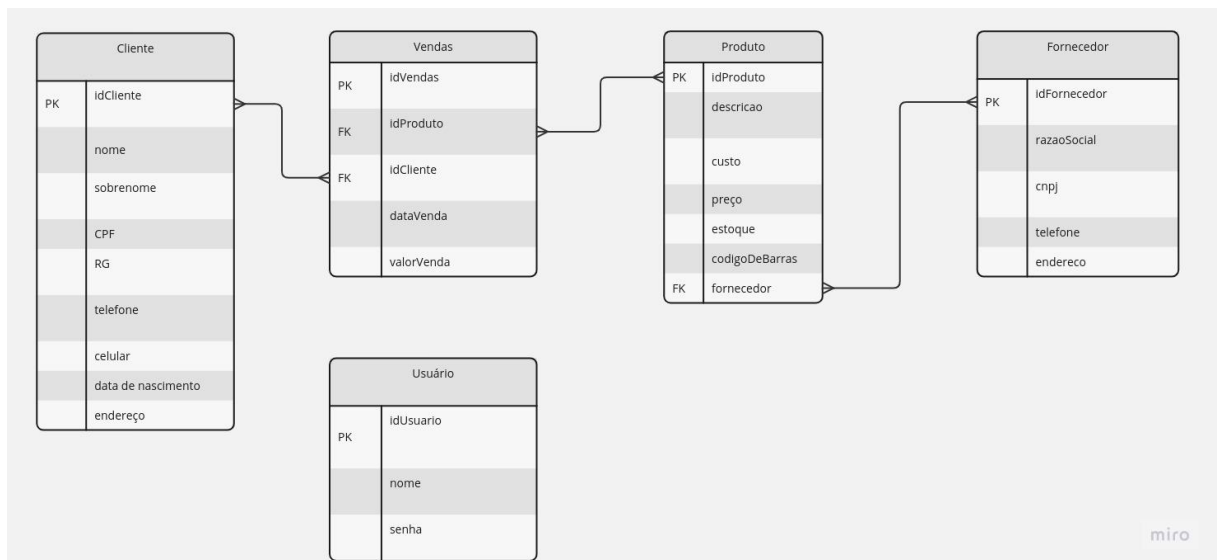
3.1 BANCO DE DADOS

Fazer a utilização de um banco de dados traz mais facilidade nos processos, além de ter mais segurança com os dados (SILVA *et al*, 2019), em vista disso para a atual empresa foi levantada a necessidade de uma tabela para: cliente, venda, produto, fornecedor e usuário.

3.1.1 MER - MODELAGEM ENTIDADE RELACIONAMENTO

Com a utilização do site MIRO, foi criado o modelo de entidade e relacionamento baseado conforme consta na FIGURA 1.

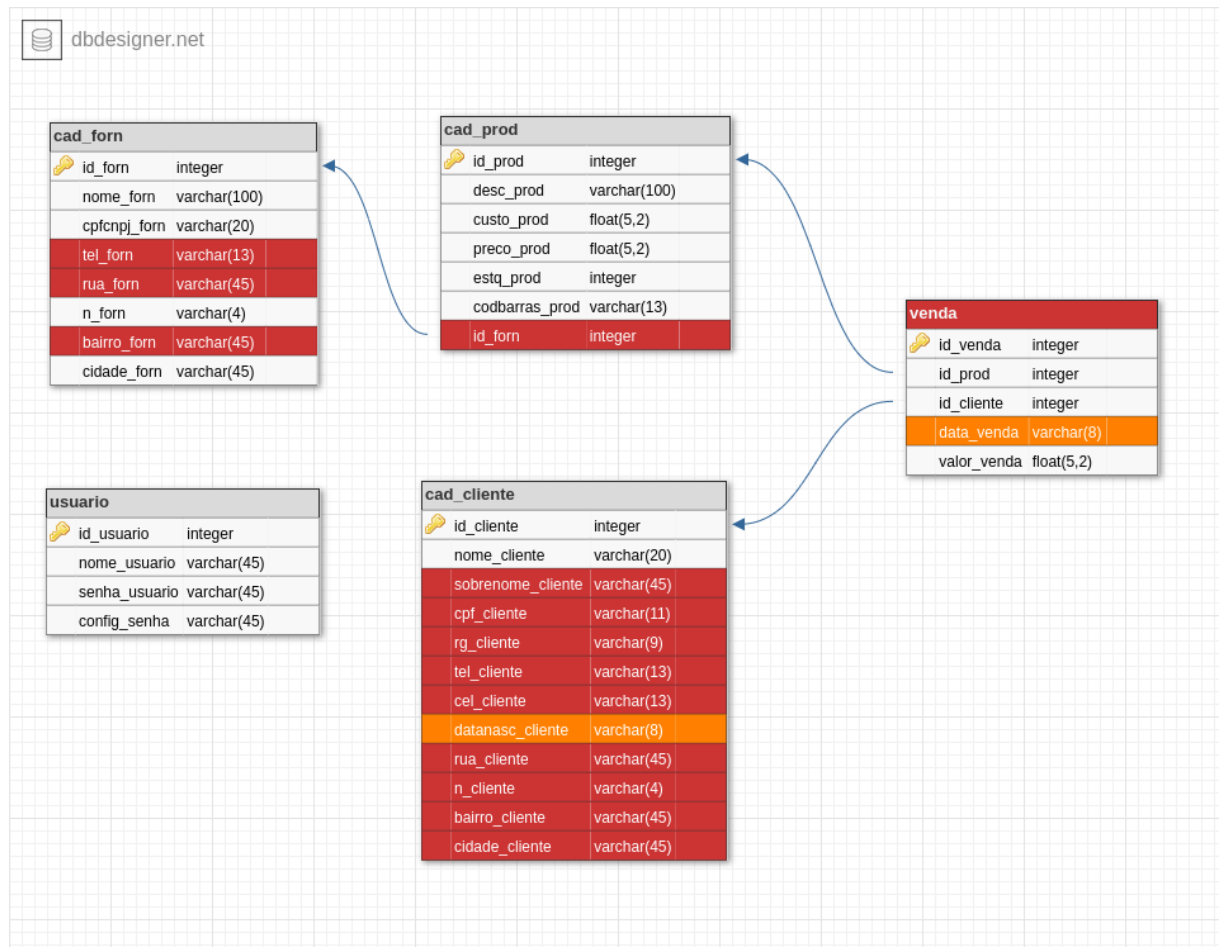
FIGURA 1 - Modelo MER: sistema empresarial



3.1.2 DER - DIAGRAMA ENTIDADE RELACIONAMENTO

Com a utilização do site DB Designer, foi criado o diagrama de entidade e relacionamento baseado conforme consta na FIGURA 2.

FIGURA 2 - Diagrama DER: sistema empresarial



3.1.3 FÍSICO

No ambiente de desenvolvimento Workbench com a utilização da linguagem SQL foi criado o modelo físico do banco de dados conforme figuras 3 até a 8.

FIGURA 3 - CÓDIGO EM SQL

```

1 CREATE TABLE `cad_forn` (
2   `id_forn` INT NOT NULL AUTO_INCREMENT,
3   `nome_forn` VARCHAR(100) NOT NULL,
4   `cpfcnpj_forn` VARCHAR(20) NOT NULL,
5   `tel_forn` VARCHAR(13) NOT NULL,
6   `rua_forn` VARCHAR(45) NOT NULL,
7   `n_forn` VARCHAR(4) NOT NULL,
8   `bairro_forn` VARCHAR(45) NOT NULL,
9   `cidade_forn` VARCHAR(45) NOT NULL,
10  PRIMARY KEY (`id_forn`)
11 );
12

```


FIGURA 4 - CÓDIGO EM SQL

```
1 • CREATE TABLE `cad_prod` (  
2     `id_prod` INT NOT NULL AUTO_INCREMENT,  
3     `desc_prod` varchar(100) NOT NULL,  
4     `custo_prod` FLOAT(5,2) NOT NULL,  
5     `preco_prod` FLOAT(5,2) NOT NULL,  
6     `estq_prod` INT NOT NULL,  
7     `codbarras_prod` varchar(13) NOT NULL,  
8     `id_forn` INT NOT NULL,  
9     PRIMARY KEY (`id_prod`)  
10 );  
11
```

FIGURA 5 - CÓDIGO EM SQL

```
1 • CREATE TABLE `venda` (  
2     `id_venda` INT NOT NULL AUTO_INCREMENT,  
3     `id_prod` INT NOT NULL,  
4     `id_cliente` INT NOT NULL,  
5     `data_venda` varchar(8) NOT NULL,  
6     `valor_venda` FLOAT(5,2) NOT NULL,  
7     PRIMARY KEY (`id_venda`)  
8 );  
9
```

FIGURA 6 - CÓDIGO EM SQL

```
1 • CREATE TABLE `usuario` (  
2     `id_usuario` INT NOT NULL AUTO_INCREMENT,  
3     `nome_usuario` varchar(45) NOT NULL,  
4     `senha_usuario` varchar(45) NOT NULL,  
5     `config_senha` varchar(45) NOT NULL,  
6     PRIMARY KEY (`id_usuario`)  
7 );  
8
```

FIGURA 7 - CÓDIGO EM SQL

```
1 • CREATE TABLE `cad_cliente` (  
2     `id_cliente` INT NOT NULL AUTO_INCREMENT,  
3     `nome_cliente` varchar(20) NOT NULL,  
4     `sobrenome_cliente` varchar(45) NOT NULL,  
5     `cpf_cliente` varchar(11) NOT NULL,  
6     `rg_cliente` varchar(9) NOT NULL,  
7     `tel_cliente` varchar(13) NOT NULL,  
8     `cel_cliente` varchar(13) NOT NULL,  
9     `datanasc_cliente` varchar(8) NOT NULL,  
10    `rua_cliente` varchar(45) NOT NULL,  
11    `n_cliente` varchar(4) NOT NULL,  
12    `bairro_cliente` varchar(45) NOT NULL,  
13    `cidade_cliente` varchar(45) NOT NULL,  
14    PRIMARY KEY (`id_cliente`)  
15 );  
16
```

FIGURA 8 - CÓDIGO EM SQL

```
1 • ALTER TABLE `cad_prod` ADD CONSTRAINT `cad_prod_fk0` FOREIGN KEY (`id_forn`) REFERENCES `cad_forn` (`id_f  
2  
3 • ALTER TABLE `venda` ADD CONSTRAINT `venda_fk0` FOREIGN KEY (`id_prod`) REFERENCES `cad_prod` (`id_prod`);  
4  
5 • ALTER TABLE `venda` ADD CONSTRAINT `venda_fk1` FOREIGN KEY (`id_cliente`) REFERENCES `cad_cliente` (`id_c  
6
```

A partir das figuras 9 até 13 é mostrado as tabelas criadas, a visualização ocorre dentro do ambiente de desenvolvimento workbench.

FIGURA 9 - TABELA cad_cliente

#	Field	Schema	Table	Type	Character Set	Display Size	Precision	Scale
1	id_cliente	banco_pi	cad_cliente	INT				
2	nome_cliente	banco_pi	cad_cliente	VARCHAR		11	0	0
3	sobrenome_cliente	banco_pi	cad_cliente	VARCHAR		20	0	0
4	cpf_cliente	banco_pi	cad_cliente	VARCHAR		45	0	0
5	rg_cliente	banco_pi	cad_cliente	VARCHAR		11	0	0
6	tel_cliente	banco_pi	cad_cliente	VARCHAR		9	0	0
7	cel_cliente	banco_pi	cad_cliente	VARCHAR		13	0	0
8	datanasc_cliente	banco_pi	cad_cliente	VARCHAR		13	0	0
9	rua_cliente	banco_pi	cad_cliente	VARCHAR		8	0	0
10	n_cliente	banco_pi	cad_cliente	VARCHAR		45	0	0
11	bairro_cliente	banco_pi	cad_cliente	VARCHAR		4	0	0
12	cidade_cliente	banco_pi	cad_cliente	VARCHAR		45	0	0

FIGURA 10 - TABELA cad_for

Administration Schemas

Query 1

```
1 • SELECT * FROM cad_for;
```

2

Field Types

#	Field	Schema	Table	Type	Character Set	Display Size	Precision	Scale
1	id_for	banco_pi	cad_for	INT		11	0	0
2	nome_for	banco_pi	cad_for	VARCHAR		100	0	0
3	cpfnpj_for	banco_pi	cad_for	VARCHAR		20	0	0
4	tel_for	banco_pi	cad_for	VARCHAR		13	0	0
5	rua_for	banco_pi	cad_for	VARCHAR		45	0	0
6	n_for	banco_pi	cad_for	VARCHAR		4	0	0
7	bairro_for	banco_pi	cad_for	VARCHAR		45	0	0
8	cidade_for	banco_pi	cad_for	VARCHAR		45	0	0

Object Info Session

Table: cad_cliente

Columns:

- id_cliente int AI PK
- nome_cliente varchar(20)
- sobrenome_cliente varchar(45)
- cpf_cliente varchar(11)
- rg_cliente varchar(9)
- tel_cliente varchar(13)
- cel_cliente varchar(13)
- datanasc_cliente varchar(8)
- rua_cliente varchar(45)

FIGURA 11 - TABELA cad_prod

Administration Schemas

Query 1

```
1 • SELECT * FROM cad_prod;
```

2

Field Types

#	Field	Schema	Table	Type	Character Set	Display Size	Precision	Scale
1	id_prod	banco_pi	cad_prod	INT		11	0	0
2	desc_prod	banco_pi	cad_prod	VARCHAR		100	0	0
3	custo_prod	banco_pi	cad_prod	DOUBLE		5	-2	2
4	preco_prod	banco_pi	cad_prod	DOUBLE		5	-2	2
5	estq_prod	banco_pi	cad_prod	INT		11	0	0
6	codbarras_prod	banco_pi	cad_prod	VARCHAR		13	0	0
7	id_for	banco_pi	cad_prod	INT		11	0	0

Object Info Session

Table: cad_cliente

Columns:

- id_cliente int AI PK
- nome_cliente varchar(20)
- sobrenome_cliente varchar(45)
- cpf_cliente varchar(11)
- rg_cliente varchar(9)
- tel_cliente varchar(13)
- cel_cliente varchar(13)
- datanasc_cliente varchar(8)
- rua_cliente varchar(45)

FIGURA 12 - TABELA usuario

Administration Schemas

SCHEMAS

Filter objects

aulas_bd

banco_pi

Tables

cad_cliente

cad_forn

cad_prod

usuario

venda

Object Info Session

Table: cad_cliente

Columns:

id_cliente int AI PK

nome_cliente varchar(20)

sobrenome_cliente varchar(45)

cpf_cliente varchar(11)

rg_cliente varchar(9)

tel_cliente varchar(13)

cel_cliente varchar(13)

datanasc_cliente varchar(8)

rua_cliente varchar(45)

Query 1

1 • SELECT * FROM usuario;

2

Field Types

#	Field	Schema	Table	Type	Character Set	Display Size	Precision	Scale
1	id_usuario	banco_pi	usuario	INT		11	0	0
2	nome_usuario	banco_pi	usuario	VARCHAR		45	0	0
3	senha_usuario	banco_pi	usuario	VARCHAR		45	0	0
4	config_senha	banco_pi	usuario	VARCHAR		45	0	0

FIGURA 13 - TABELA venda

Administration Schemas

SCHEMAS

Filter objects

aulas_bd

banco_pi

Tables

cad_cliente

cad_forn

cad_prod

usuario

venda

Object Info Session

Table: cad_cliente

Columns:

id_cliente int AI PK

nome_cliente varchar(20)

sobrenome_cliente varchar(45)

cpf_cliente varchar(11)

rg_cliente varchar(9)

tel_cliente varchar(13)

cel_cliente varchar(13)

datanasc_cliente varchar(8)

rua_cliente varchar(45)

Query 1

1 • SELECT * FROM venda;

2

Field Types

#	Field	Schema	Table	Type	Character Set	Display Size	Precision	Scale
1	id_venda	banco_pi	venda	INT		11	0	0
2	id_prod	banco_pi	venda	INT		11	0	0
3	id_cliente	banco_pi	venda	INT		11	0	0
4	data_venda	banco_pi	venda	VARCHAR		8	0	0
5	valor_venda	banco_pi	venda	DOUBLE		5	-2	2

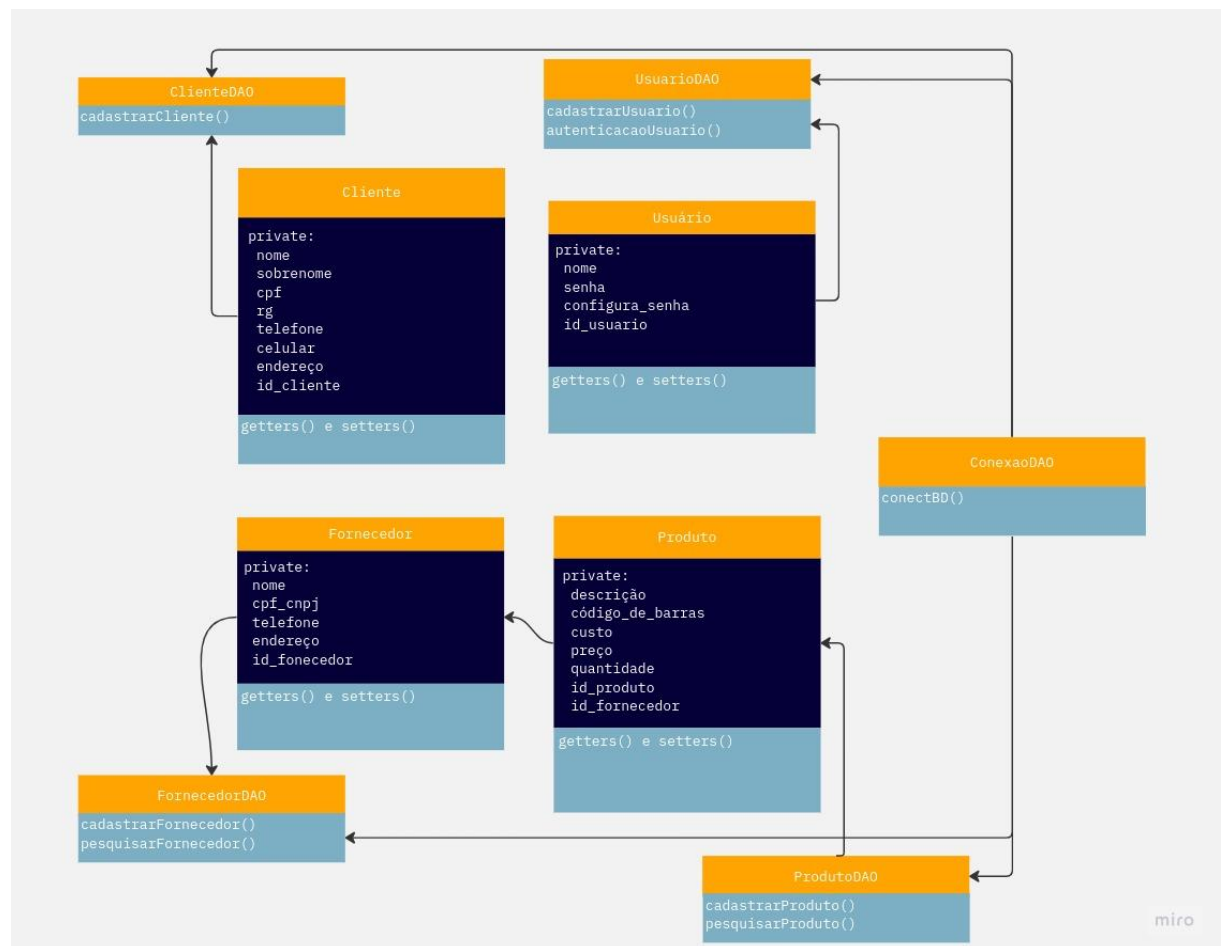
3.2 PROGRAMAÇÃO ORIENTADA A OBJETOS

Com a utilização da linguagem JAVA e a IDE NetBeans foi construído um sistema para cadastramento de cliente, produtos e fornecedores, além de cadastrar as vendas realizadas, tendo manipulação do banco de dados inserindo os novos dados.

3.2.1 DIAGRAMA DE CLASSES

O seguinte diagrama de classes, figura 14, foi elaborado no site web MIRO.

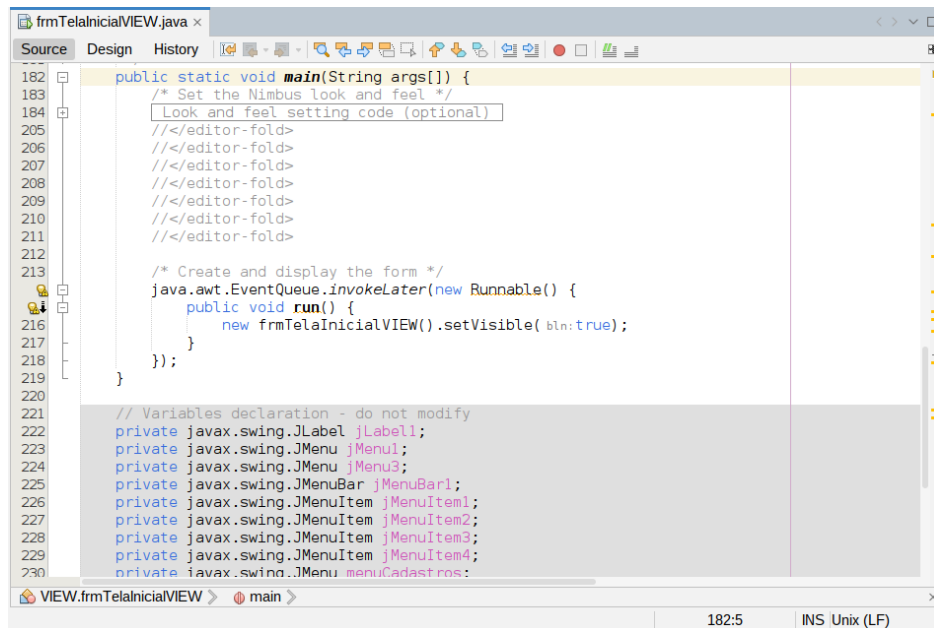
FIGURA 14 - DIAGRAMA DE CLASSES



3.2.2 CÓDIGOS DO SISTEMA

O arquivo principal .main está localizado na parte da tela inicial do sistema, como mostrado na figura 15.

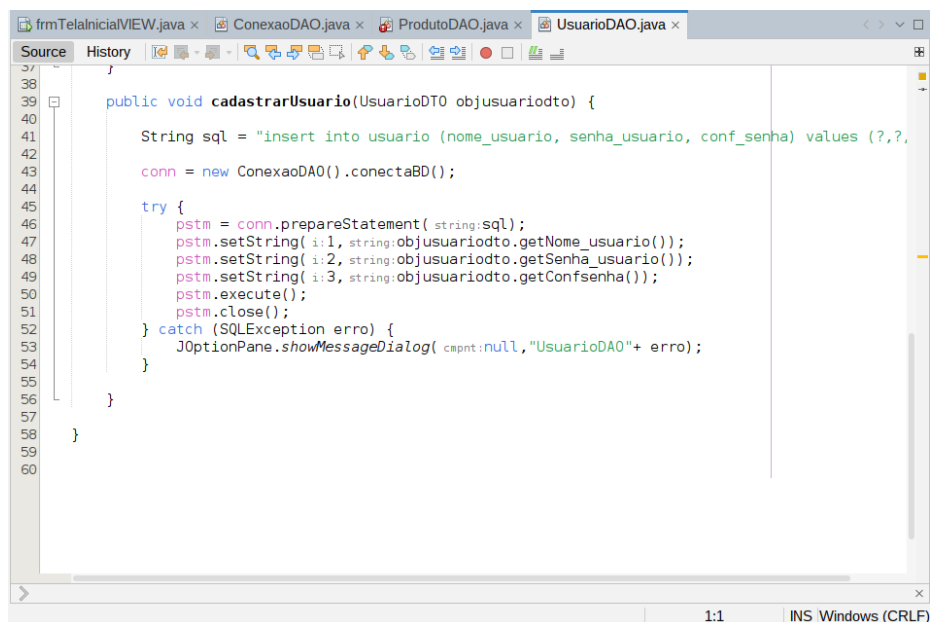
FIGURA 15 - CÓDIGO JAVA



```
182 public static void main(String args[]) {
183     /* Set the Nimbus look and feel */
184     Look and feel setting code (optional)
185     //</editor-fold>
186     //</editor-fold>
187     //</editor-fold>
188     //</editor-fold>
189     //</editor-fold>
190     //</editor-fold>
191     //</editor-fold>
192     //</editor-fold>
193
194     /* Create and display the form */
195     java.awt.EventQueue.invokeLater(new Runnable() {
196         public void run() {
197             new frmTelaInicialVIEW().setVisible(true);
198         }
199     });
200 }
201
202 // Variables declaration - do not modify
203 private javax.swing.JLabel jLabel1;
204 private javax.swing.JMenu jMenu1;
205 private javax.swing.JMenu jMenu3;
206 private javax.swing.JMenuBar jMenuBar1;
207 private javax.swing.JMenuItem jMenuItem1;
208 private javax.swing.JMenuItem jMenuItem2;
209 private javax.swing.JMenuItem jMenuItem3;
210 private javax.swing.JMenuItem jMenuItem4;
211 private javax.swing.JMenu menuCadast ros;
```

O model com as instruções SQL de cadastro de usuário se encontra na figura 16.

FIGURA 16 - INSTRUÇÕES SQL DENTRO DO JAVA



```
37 }
38
39 public void cadastrarUsuario(UsuarioDTO objusuario) {
40
41     String sql = "insert into usuario (nome_usuario, senha_usuario, conf_senha) values (?, ?, ?)";
42
43     conn = new ConexaoDAO().conectaBD();
44
45     try {
46         pstmt = conn.prepareStatement(sql);
47         pstmt.setString(1, objusuario.getNome_usuario());
48         pstmt.setString(2, objusuario.getSenha_usuario());
49         pstmt.setString(3, objusuario.getConfSenha());
50         pstmt.execute();
51         pstmt.close();
52     } catch (SQLException erro) {
53         JOptionPane.showMessageDialog(null, "UsuarioDAO"+ erro);
54     }
55 }
56
57 }
58
59 }
60
```

O controler que chama esse model está mostrado na figura 17.

FIGURA 17 - CÓDIGO EM JAVA

```
...va ConexaoDAO.java x ProdutoDAO.java x UsuarioDAO.java x frmCadUsuarioVIEW.java x
Source Design History
141 frmTelaInicialVIEW mTelaInicial = new frmTelaInicialVIEW();
142     this.setVisible( bIn: false);
143     // mInicial.setExtendedState(JFrame.MAXIMIZED_BOTH);
144     mTelaInicial.setVisible( bIn: true);
145 }
146
147
148 private void btnCadastrarActionPerformed(java.awt.event.ActionEvent evt) {
149     // TODO add your handling code here:
150     String usuario, senha, confsenha;
151
152     usuario = txtCadUsuario.getText();
153     senha = txtSenha.getText();
154     confsenha = txtConfSenha.getText();
155
156     if (txtSenha.getText().equals( o:txtConfSenha.getText())){
157         UsuarioDTO objusuariodto = new UsuarioDTO();
158         objusuariodto.setNome_usuario( nome_usuario:usuario);
159         objusuariodto.setSenha_usuario( senha_usuario:senha);
160         objusuariodto.setConfsenha(confsenha);
161
162         UsuarioDAO objusuariodao = new UsuarioDAO();
163         objusuariodao.cadastrarUsuario(objusuariodto);
164     } else{
165         JOptionPane.showMessageDialog( cmpnt:null, o:"As senhas não são iguais" );}
166
167 }
168
169
VIEW.frmCadUsuarioVIEW > frmCadUsuarioVIEW > 18:5 INS Unix (LF)
```

3.2.3 IMAGENS DO SISTEMA

A seguir estarão representando todas as telas criadas em Java para o gerenciamento de vendas e estoque.

FIGURA 18 - LOGIN

The image shows a login window titled "Login Gerenciador de Vendas". It contains two text input fields: "Usuario:" and "Senha:". Below these fields is a button labeled "Login". The window has a light gray background and a blue border.

FIGURA 19 - TELA INICIAL

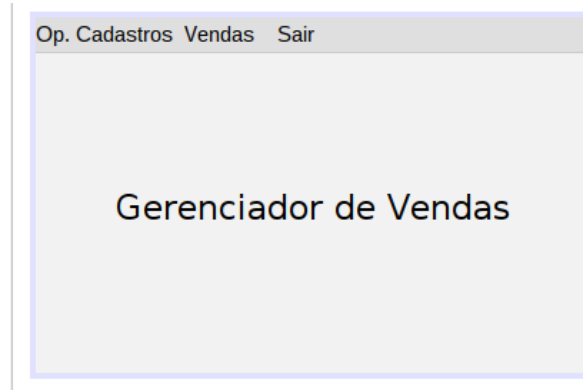


FIGURA 20 - TELA INICIAL NA OPÇÃO CADASTROS

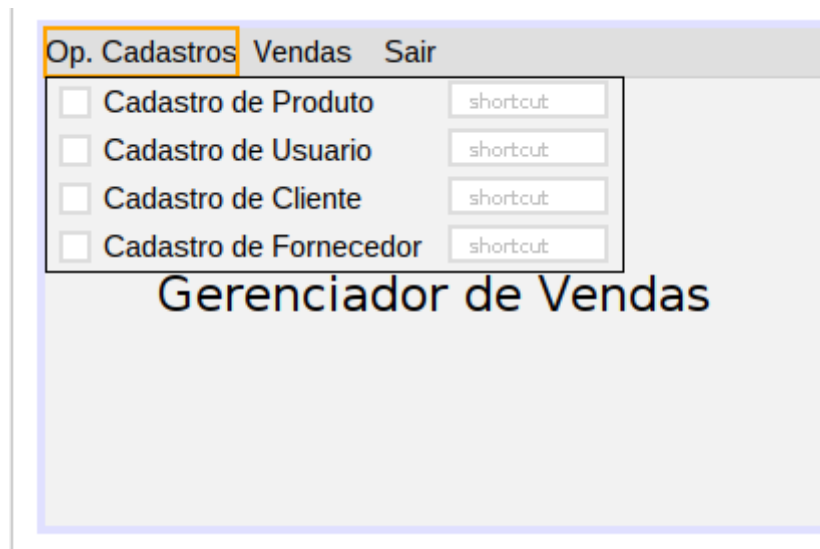


FIGURA 21 - TELA INICIAL NA OPÇÃO VENDAS

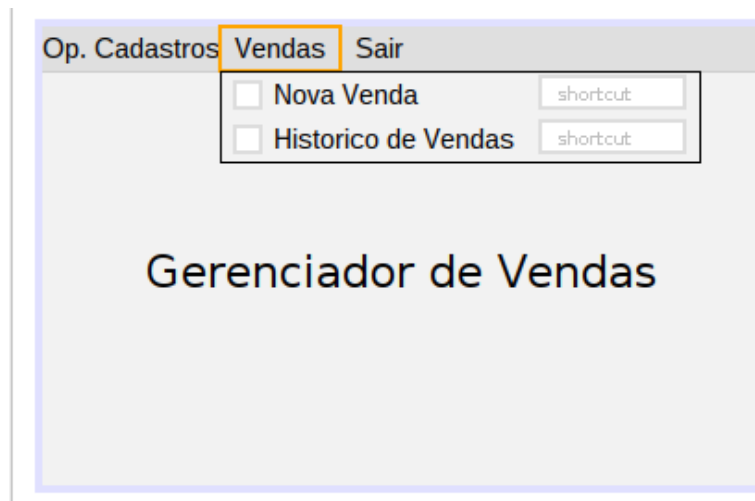


FIGURA 22 - TELA INICIAL NA OPÇÃO SAIR

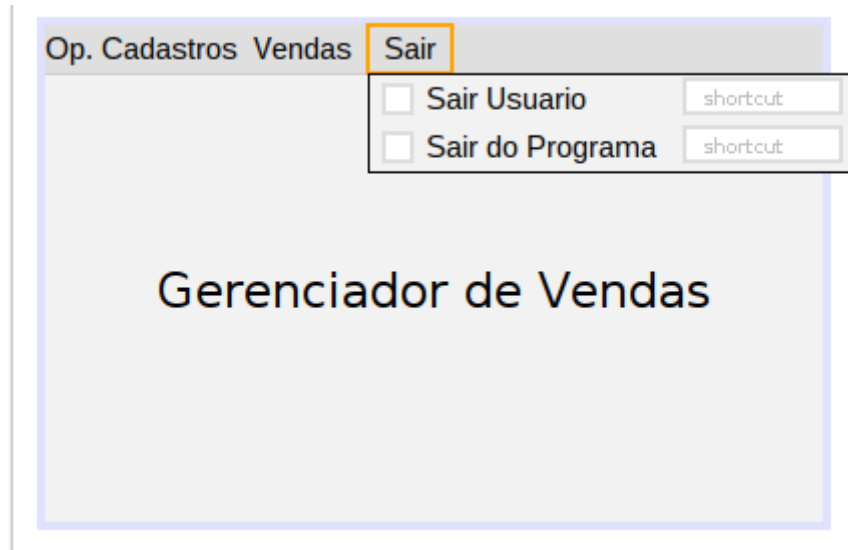


FIGURA 23 - TELA CADASTRO DE PRODUTO

The image shows a form titled 'Cadastro de Produto'. The form contains several input fields and buttons. The fields are: 'Código interno:' followed by a text input; 'Código do fornecedor:' followed by a text input; 'Descrição do produto:' followed by a wide text input; 'Custo:' followed by a text input; 'Preço de Venda:' followed by a text input; 'Qtd. Estoque:' followed by a text input; and 'Código de Barras:' followed by a text input. Below the input fields, there are three buttons: 'Cadastrar', 'Menu Principal', and 'Rel. Produtos Cadastrados'.

FIGURA 24 - TELA DE RELATÓRIO DE PRODUTOS CADASTRADOS

Relatorio de Cadastro de Produtos					
ID do Produto	Descrição do ...	Custo Produto ...	Preço Produto ...	Esstoque do P...	Código de Barr...

Pesquisa Produtos Menu Principal

FIGURA 25 - TELA DE CADASTRO DE USUÁRIOS

Cadastro de Usuario

Usuario:

Senha:

Confirme a senha:

Cadastrar Limpar

Menu Principal

FIGURA 26 - TELA DE CADASTRO DE CLIENTE

Cadastro de Clientes

Relatorio Cad. Clientes Codigo Cliente:

Nome: Sobrenome:

CPF: RG:

Telefone: Celular:

Data de Nascimento: Ex: 22-10-2001

Endereço:

Rua: Nº:

Bairro: Cidade:

FIGURA 27 - TELA DE RELATÓRIO DE CADASTRO DE CLIENTES

Cód Clie...	Nome	Sobreno...	CPF	RG	Telefone	Celular	Data Na...	Rua	Numero	Bairro	Cidade

FIGURA 28 - TELA DE CADASTRO DE FORNECEDOR

Cadastro de Fornecedor

Codigo do Fornecedor:

Nome Fornecedor:

CPF/CNPJ:

Telefone:

Endereço: Rua:

Numero: Bairro:

Cidade:

FIGURA 29 - TELA DE RELATÓRIO DE CADASTRO DE FORNECEDOR

Relatorio de fornecedores cadastrados

Cod. For...	Nome F...	CPF/CN...	Telefone	Rua	Numero	Bairro	Cidade

FIGURA 30 - TELA DE VENDA

Venda

Código do Produto:

Descrição de Produto:

Preço: Estoque Disponível:

3.3 CONTEÚDO DA FORMAÇÃO PARA A VIDA: CRIANDO O NOVO

3.3.1 CRIANDO O NOVO

Síntese dos seguintes conteúdos baseado na apostila disponível na plataforma de aprendizagem.

- **Tópico 1:** Design Thinking nos estudos e na profissão

O que seria isso? É a forma de pensar que promove o bem-estar na vida das pessoas, sendo empático. Uma linha de pensamento que segue em identificar problemas e gerar soluções que de para serem executadas e esse tipo de pensamento ajuda na construção de um profissional T, ajudando a desenvolver um pensamento crítico porém ainda continuar sendo criativo.

Em uma visão profissional, na prática, o design thinking é centrado no ser humano, então ao invés de desenvolver o atual projeto apenas pensando no software, decidimos seguir com uma visão mais ampla: como o usuário vai interagir com nosso produto? Qual a melhor forma de fazer para que ele conclua as ações? Como ele vai se sentir cadastrando um produto?

Foi com essa visão que desenhamos e redesenhamos nosso sistema, para ser o mais agradável possível para o usuário em seu dia-a dia.

- **Tópico 2:** Há mil maneiras de pensar

A ideia de viver em um mundo exponencial parece assustadora, tudo mudando em questões de segundo? Mas tudo depende da forma como isso é encarado, afinal são as pessoas que geram o capital, a mudança de tudo.

O desenvolvimento responsável é um grande aliado para todo esse crescimento, trazendo consigo o engajamento entre o planeta e nós mesmo, visando as necessidades, além de tomar decisões diretas com visões inovadoras.

A interação uns com os outros traz consigo inúmeros desafios, dentro deles a crença limitante, que acontece no meio em que crescemos. Mas, na prática, como superar isso? Com o diálogo! Uma coisa simples como conversar é uma importante ferramenta com quem sabe o que dizer, ter uma comunicação assertiva tanto de forma intrapessoal quanto de forma interpessoal faz parte do processo de quebrar as crenças limitantes.

Aplique no seu dia-a-dia: Pare, escute a si mesmo e ao outro, converse de forma clara, a ajudar a solucionar os conflitos com criatividade e por um meio saudável.

- **Tópico 3:** Criando asas

O famoso “pensar fora da caixa”. Porém o que realmente é fazer isso?

Pensar fora da caixa implica em criar novas caixas, novas linhas de pensamento criativos. Em suma, o ser humano possui sete padrões cerebrais

ligados à criatividade: conexão, razão, visão, absorção, transformação, avaliação e fluxo.

Normalmente, seguimos de três a quatro padrões, logo para “sair da caixa” é necessário identificar os padrões no qual está submetido e criar rotinas em que se mude esses padrões.

O que fazer na prática? Redesenhar a rotina, alterando os padrões em que está preso, adicionando novas coisas ou apenas mudando a ordem do que já acontece. Afinal, sua forma de pensar impacta diretamente no modo que age e no modo em que a criatividade funciona.

- **Tópico 4:** Com vocês: O duplo diamante!

O duplo diamante é o diagrama para descrever o processo do design thinking: descoberta > definição > desenvolvimento > entrega.

Onde cada etapa tem seu objetivo, na descoberta é o momento de observar, pesquisar e analisar. Na definição é o momento de interpretar tudo da fase anterior, alinhando com as necessidades e objetivos (nessa fase geralmente gera insights). No desenvolvimento utiliza-se os insights para desenvolver as soluções de forma sistemática, testando as soluções para chegar na melhor de todas. Por fim, mas não menos importante, a entrega é o momento final, em que o produto está finalizado e pronto para ser lançado.

Muito bonito, mas como aplicar isso no dia-a-dia? No atual projeto foi necessário seguir o passo-a-passo para concluirmos, onde na descoberta foi o momento de observar os conteúdos e pesquisar, na definição foi o momento de realmente entender como tudo funcionava, no desenvolvimento foi a hora de colocar a mão na massa e fazer, na entrega foi finalizado o relatório e finalmente o programa estava pronto. É possível aplicar o duplo diamante em qualquer projeto, sendo capaz de pensar de forma mais inovadora e criativa.

3.3.2 ESTUDANTES NA PRÁTICA

No desenvolvimento desse tópico foi optado por fazer um banner falando mais sobre o método do duplo diamante para a realização de projetos/trabalhos. O grupo colaborou de forma onde ocorresse a melhor passagem de conteúdo sobre o tema.

Link de acesso ao banner:

https://drive.google.com/drive/folders/1LdLz2qaNI_H1KQNgndYiUi4csGUNRw6J?usp=sharing

4 CONCLUSÃO

Por meio do atual projeto foi possível conhecer os conceitos de banco de dados, além dos modelos de entidade relacionamento e do diagrama de entidade relacionamento, e aplicá-los na prática com a criação dos modelos relacional, lógico e físico. Sendo capaz, também, a compreensão do funcionamento da programação orientada a objetos, tendo esclarecimento sobre encapsulamento, abstração, herança e polimorfismo, de fato aplicando esses conceitos no projeto prático.

Em suma, a maior dificuldade foi a relação do banco de dados MySQL com a linguagem de programação Java, a forma de como fazer a ligação das duas coisas que, no início, era um pouco turva, mas ao decorrer do projeto ficou clara e foi totalmente capaz de ser executada, resultado em um bom sistema de gerenciamento de vendas e estoque.

REFERÊNCIAS

CARVALHO, Victorio Albani de; TEIXEIRA, Giovany Frossard. **Programação orientada a objetos**. Espírito Santo, 2012. Disponível em <http://redeetec.mec.gov.br/images/stories/pdf/eixo_infor_comun/tec_inf/081112_progr_obj.pdf> . Acessado em 09/04/2023.

MACHADO, F. N. R. Banco de dados: projeto e implementação. São Paulo: Érica, 2014.

RICARTE, Ivan Luiz Marques. Programação orientada a objetos: uma abordagem com Java. Campinas, 2001. Disponível em <<https://www.dca.fee.unicamp.br/cursos/PooJava/Aulas/poojava.pdf>> Acessado em 09/04/2023.

SILVA, Adriel Thainan Izidoro da; BRONZATTO, Marcello *et al.* A Importância do Banco de Dados para o Controle em uma Indústria. São Paulo, 2019.

VIANNA, João Tito Almeida. Introdução a orientação a objetos. Juiz de Fora, 2013. Disponível em <<https://www.ufjf.br/peteletrica/files/2012/10/Curso-Orienta%C3%A7%C3%A3o-a-objetos.pdf>> Acessado em 09/04/2023.

ANEXOS

FIGURA 1 - Modelo MER: sistema empresarial

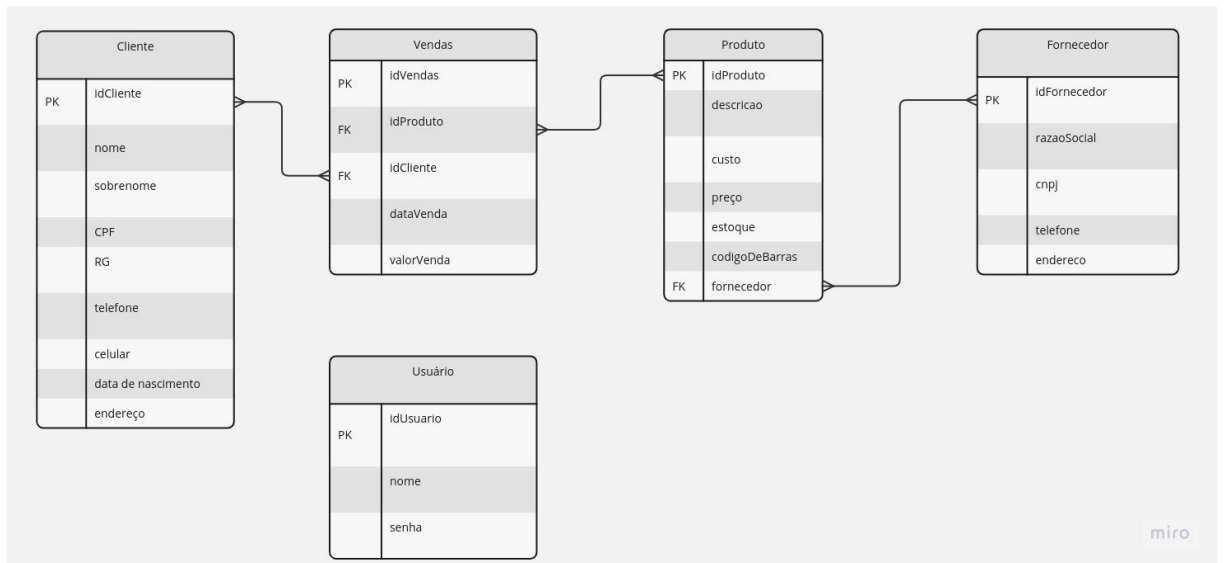


FIGURA 2 - Diagrama DER: sistema empresarial

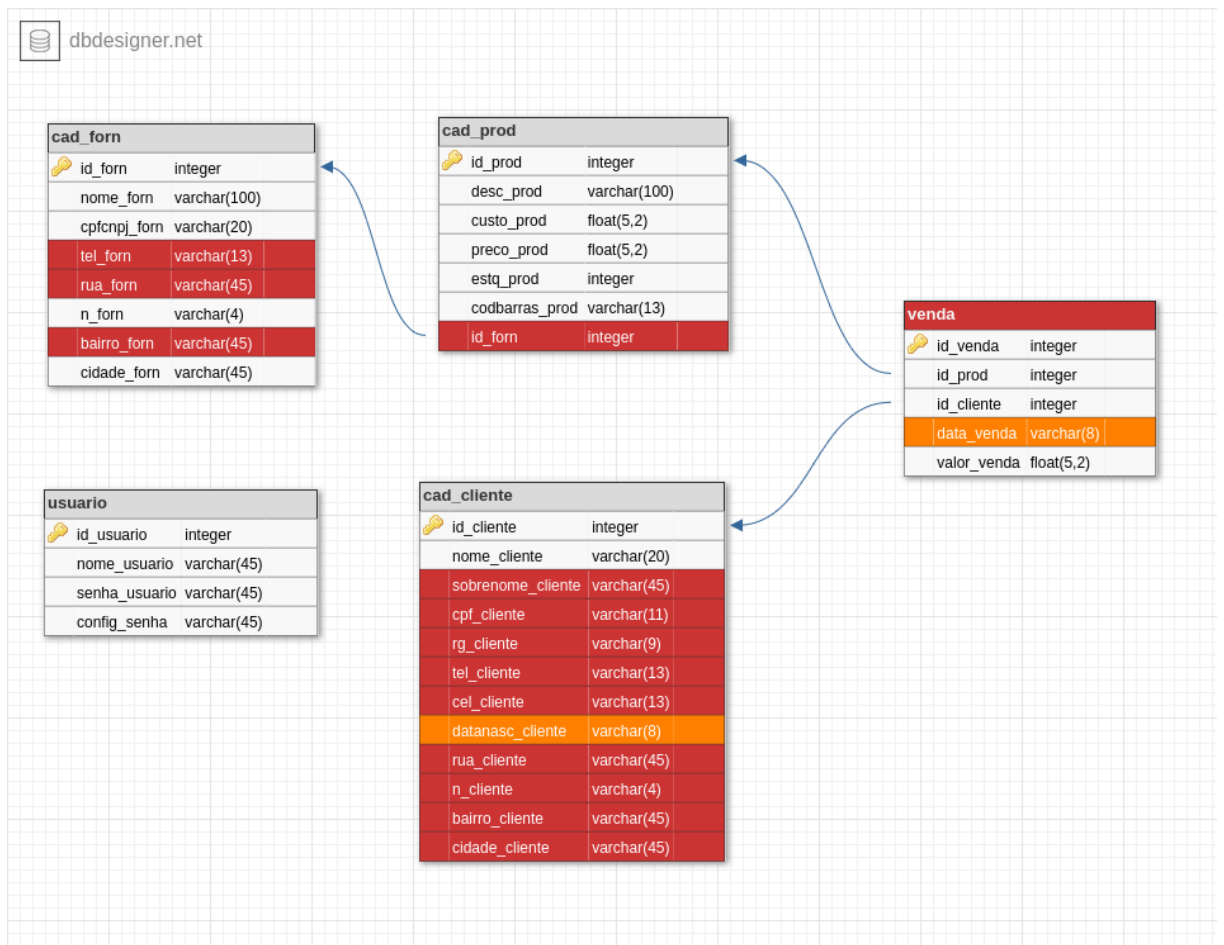


FIGURA 3 - CÓDIGO EM SQL

```

1 • CREATE TABLE `cad_forn` (
2     `id_forn` INT NOT NULL AUTO_INCREMENT,
3     `nome_forn` varchar(100) NOT NULL,
4     `cpfcnpj_forn` varchar(20) NOT NULL,
5     `tel_forn` varchar(13) NOT NULL,
6     `rua_forn` varchar(45) NOT NULL,
7     `n_forn` varchar(4) NOT NULL,
8     `bairro_forn` varchar(45) NOT NULL,
9     `cidade_forn` varchar(45) NOT NULL,
10    PRIMARY KEY (`id_forn`)
11 );
12

```

FIGURA 4 - CÓDIGO EM SQL

```

1 • CREATE TABLE `cad_prod` (
2     `id_prod` INT NOT NULL AUTO_INCREMENT,
3     `desc_prod` varchar(100) NOT NULL,
4     `custo_prod` FLOAT(5,2) NOT NULL,
5     `preco_prod` FLOAT(5,2) NOT NULL,
6     `estq_prod` INT NOT NULL,
7     `codbarras_prod` varchar(13) NOT NULL,
8     `id_forn` INT NOT NULL,
9     PRIMARY KEY (`id_prod`)
10 );
11

```

FIGURA 5 - CÓDIGO EM SQL

```

1 • CREATE TABLE `venda` (
2     `id_venda` INT NOT NULL AUTO_INCREMENT,
3     `id_prod` INT NOT NULL,
4     `id_cliente` INT NOT NULL,
5     `data_venda` varchar(8) NOT NULL,
6     `valor_venda` FLOAT(5,2) NOT NULL,
7     PRIMARY KEY (`id_venda`)
8 );
9

```

FIGURA 6 - CÓDIGO EM SQL

```

1 • CREATE TABLE `usuario` (
2     `id_usuario` INT NOT NULL AUTO_INCREMENT,
3     `nome_usuario` varchar(45) NOT NULL,
4     `senha_usuario` varchar(45) NOT NULL,
5     `config_senha` varchar(45) NOT NULL,
6     PRIMARY KEY (`id_usuario`)
7 );
8

```

FIGURA 7 - CÓDIGO EM SQL

```
1 • CREATE TABLE `cad_cliente` (  
2     `id_cliente` INT NOT NULL AUTO_INCREMENT,  
3     `nome_cliente` varchar(20) NOT NULL,  
4     `sobrenome_cliente` varchar(45) NOT NULL,  
5     `cpf_cliente` varchar(11) NOT NULL,  
6     `rg_cliente` varchar(9) NOT NULL,  
7     `tel_cliente` varchar(13) NOT NULL,  
8     `cel_cliente` varchar(13) NOT NULL,  
9     `datanasc_cliente` varchar(8) NOT NULL,  
10    `rua_cliente` varchar(45) NOT NULL,  
11    `n_cliente` varchar(4) NOT NULL,  
12    `bairro_cliente` varchar(45) NOT NULL,  
13    `cidade_cliente` varchar(45) NOT NULL,  
14    PRIMARY KEY (`id_cliente`)  
15 );  
16
```

FIGURA 8 - CÓDIGO EM SQL

```
1 • ALTER TABLE `cad_prod` ADD CONSTRAINT `cad_prod_fk0` FOREIGN KEY (`id_forn`) REFERENCES `cad_forn` (`id_f  
2  
3 • ALTER TABLE `venda` ADD CONSTRAINT `venda_fk0` FOREIGN KEY (`id_prod`) REFERENCES `cad_prod` (`id_prod`);  
4  
5 • ALTER TABLE `venda` ADD CONSTRAINT `venda_fk1` FOREIGN KEY (`id_cliente`) REFERENCES `cad_cliente` (`id_c  
6
```

A partir das figuras 9 até 13 é mostrado as tabelas criadas, a visualização ocorre dentro do ambiente de desenvolvimento, workbench.

FIGURA 9 - TABELA cad_cliente

Administration Schemas

Query 1

```
1 • SELECT * FROM cad_cliente;
```

2

#	Field	Schema	Table	Type	Character Set	Display Size	Precision	Scale
1	id_cliente	banco_pi	cad_cliente	INT		11	0	0
2	nome_cliente	banco_pi	cad_cliente	VARCHAR		20	0	0
3	sobrenome_cliente	banco_pi	cad_cliente	VARCHAR		45	0	0
4	cpf_cliente	banco_pi	cad_cliente	VARCHAR		11	0	0
5	rg_cliente	banco_pi	cad_cliente	VARCHAR		9	0	0
6	tel_cliente	banco_pi	cad_cliente	VARCHAR		13	0	0
7	cel_cliente	banco_pi	cad_cliente	VARCHAR		13	0	0
8	datanasc_cliente	banco_pi	cad_cliente	VARCHAR		8	0	0
9	rua_cliente	banco_pi	cad_cliente	VARCHAR		45	0	0
10	n_cliente	banco_pi	cad_cliente	VARCHAR		4	0	0
11	bairro_cliente	banco_pi	cad_cliente	VARCHAR		45	0	0
12	cidade_cliente	banco_pi	cad_cliente	VARCHAR		45	0	0

Object Info Session

Table: cad_cliente

Columns:

- id_cliente int AI PK
- nome_cliente varchar(20)
- sobrenome_cliente varchar(45)
- cpf_cliente varchar(11)
- rg_cliente varchar(9)
- tel_cliente varchar(13)
- cel_cliente varchar(13)
- datanasc_cliente varchar(8)
- rua_cliente varchar(45)

FIGURA 10 - TABELA cad_for

Administration Schemas

Query 1

```
1 • SELECT * FROM cad_for;
```

2

#	Field	Schema	Table	Type	Character Set	Display Size	Precision	Scale
1	id_for	banco_pi	cad_for	INT		11	0	0
2	nome_for	banco_pi	cad_for	VARCHAR		100	0	0
3	cpfnpj_for	banco_pi	cad_for	VARCHAR		20	0	0
4	tel_for	banco_pi	cad_for	VARCHAR		13	0	0
5	rua_for	banco_pi	cad_for	VARCHAR		45	0	0
6	n_for	banco_pi	cad_for	VARCHAR		4	0	0
7	bairro_for	banco_pi	cad_for	VARCHAR		45	0	0
8	cidade_for	banco_pi	cad_for	VARCHAR		45	0	0

Object Info Session

Table: cad_cliente

Columns:

- id_cliente int AI PK
- nome_cliente varchar(20)
- sobrenome_cliente varchar(45)
- cpf_cliente varchar(11)
- rg_cliente varchar(9)
- tel_cliente varchar(13)
- cel_cliente varchar(13)
- datanasc_cliente varchar(8)
- rua_cliente varchar(45)

FIGURA 11 - TABELA cad_prod

Administration Schemas

Query 1

```
1 • SELECT * FROM cad_prod;
```

2

#	Field	Schema	Table	Type	Character Set	Display Size	Precision	Scale
1	id_prod	banco_pi	cad_prod	INT		11	0	0
2	desc_prod	banco_pi	cad_prod	VARCHAR		100	0	0
3	custo_prod	banco_pi	cad_prod	DOUBLE		5	-2	2
4	preco_prod	banco_pi	cad_prod	DOUBLE		5	-2	2
5	estq_prod	banco_pi	cad_prod	INT		11	0	0
6	codbarras_prod	banco_pi	cad_prod	VARCHAR		13	0	0
7	id_for	banco_pi	cad_prod	INT		11	0	0

Object Info Session

Table: cad_cliente

Columns:

- id_cliente int AI PK
- nome_cliente varchar(20)
- sobrenome_cliente varchar(45)
- cpf_cliente varchar(11)
- rg_cliente varchar(9)
- tel_cliente varchar(13)
- cel_cliente varchar(13)
- datanasc_cliente varchar(8)
- rua_cliente varchar(45)

FIGURA 12 - TABELA usuario

The screenshot displays a database management interface. On the left, a 'SCHEMAS' tree shows the 'banco_pi' schema expanded to show tables: 'cad_cliente', 'cad_forn', 'cad_prod', 'usuario', and 'venda'. The 'Object Info' panel for 'cad_cliente' lists its columns: id_cliente (int AI PK), nome_cliente (varchar(20)), sobrenome_cliente (varchar(45)), cpf_cliente (varchar(11)), rg_cliente (varchar(9)), tel_cliente (varchar(13)), cel_cliente (varchar(13)), datanasc_cliente (varchar(8)), and rua_cliente (varchar(45)). The main area shows a query editor with 'Query 1' containing 'SELECT * FROM usuario;'. Below the query, a 'Field Types' table lists the columns of the 'usuario' table:

#	Field	Schema	Table	Type	Character Set	Display Size	Precision	Scale
1	id_usuario	banco_pi	usuario	INT		11	0	0
2	nome_usuario	banco_pi	usuario	VARCHAR		45	0	0
3	senha_usuario	banco_pi	usuario	VARCHAR		45	0	0
4	config_senha	banco_pi	usuario	VARCHAR		45	0	0

FIGURA 13 - TABELA venda

The screenshot displays a database management interface. On the left, the 'SCHEMAS' tree shows the 'banco_pi' schema expanded to show tables: 'cad_cliente', 'cad_forn', 'cad_prod', 'usuario', and 'venda'. The 'Object Info' panel for 'cad_cliente' lists its columns: id_cliente (int AI PK), nome_cliente (varchar(20)), sobrenome_cliente (varchar(45)), cpf_cliente (varchar(11)), rg_cliente (varchar(9)), tel_cliente (varchar(13)), cel_cliente (varchar(13)), datanasc_cliente (varchar(8)), and rua_cliente (varchar(45)). The main area shows a query editor with 'Query 1' containing 'SELECT * FROM venda;'. Below the query, a 'Field Types' table lists the columns of the 'venda' table:

#	Field	Schema	Table	Type	Character Set	Display Size	Precision	Scale
1	id_venda	banco_pi	venda	INT		11	0	0
2	id_prod	banco_pi	venda	INT		11	0	0
3	id_cliente	banco_pi	venda	INT		11	0	0
4	data_venda	banco_pi	venda	VARCHAR		8	0	0
5	valor_venda	banco_pi	venda	DOUBLE		5	-2	2

FIGURA 14 - DIAGRAMA DE CLASSES

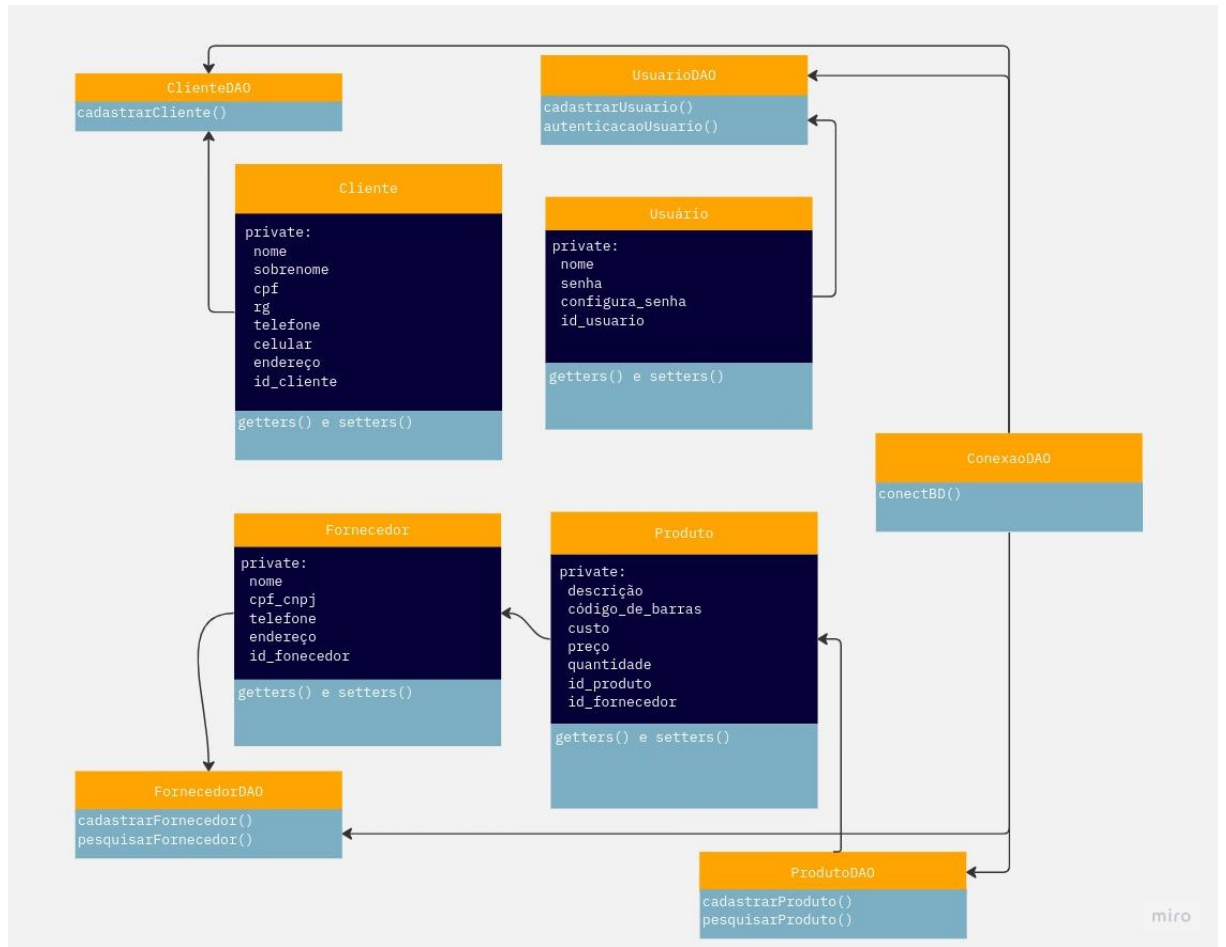
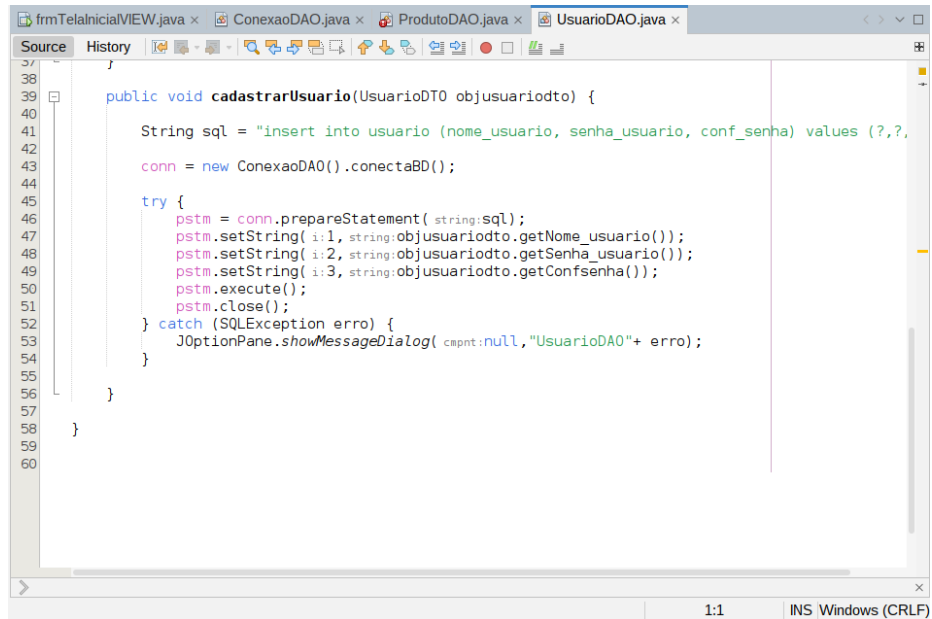


FIGURA 15 - CÓDIGO JAVA

```

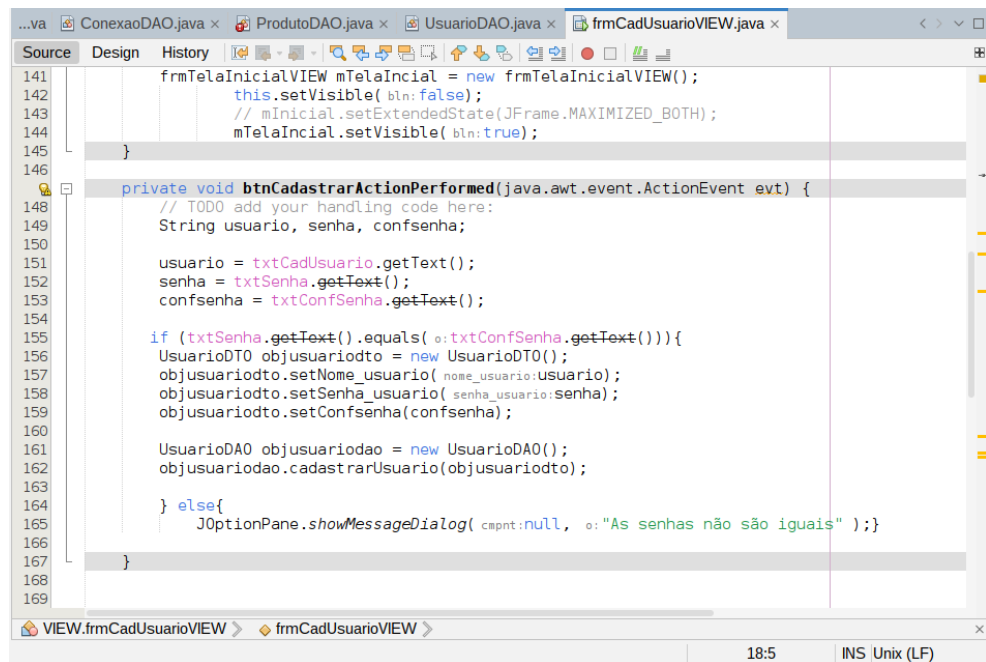
frmTelaInicialVIEW.java x
Source Design History
182 public static void main(String args[]) {
183     /* Set the Nimbus look and feel */
184     Look and feel setting code (optional)
185     //</editor-fold>
186     //</editor-fold>
187     //</editor-fold>
188     //</editor-fold>
189     //</editor-fold>
190     //</editor-fold>
191     //</editor-fold>
192     /* Create and display the form */
193     java.awt.EventQueue.invokeLater(new Runnable() {
194         public void run() {
195             new frmTelaInicialVIEW().setVisible(true);
196         }
197     });
198 }
199
200 // Variables declaration - do not modify
201 private javax.swing.JLabel jLabel1;
202 private javax.swing.JMenu jMenu1;
203 private javax.swing.JMenu jMenu3;
204 private javax.swing.JMenuBar jMenuBar1;
205 private javax.swing.JMenuItem jMenuItem1;
206 private javax.swing.JMenuItem jMenuItem2;
207 private javax.swing.JMenuItem jMenuItem3;
208 private javax.swing.JMenuItem jMenuItem4;
209 private javax.swing.JMenu menuCadastro;
210
211 VIEW.frmTelaInicialVIEW > main >
182:5 INS Unix (LF)
    
```

FIGURA 16 - INSTRUÇÕES SQL DENTRO DO JAVA



```
37  
38  
39  
40 public void cadastrarUsuario(UsuarioDTO objusuario) {  
41     String sql = "insert into usuario (nome_usuario, senha_usuario, conf_senha) values (?,?,  
42  
43     conn = new ConexaoDAO().conectaBD();  
44  
45     try {  
46         pstmt = conn.prepareStatement( string:sql);  
47         pstmt.setString( i:1, string:objusuario.getNome_usuario());  
48         pstmt.setString( i:2, string:objusuario.getSenha_usuario());  
49         pstmt.setString( i:3, string:objusuario.getConfsenha());  
50         pstmt.executeUpdate();  
51     } catch (SQLException erro) {  
52         JOptionPane.showMessageDialog( cmpnt:null, "UsuarioDAO"+ erro);  
53     }  
54  
55 }  
56  
57 }  
58  
59  
60
```

FIGURA 17 - CÓDIGO EM JAVA



```
141 frmTelaInicialVIEW mTelaInicial = new frmTelaInicialVIEW();  
142 this.setVisible( bln:false);  
143 // mInicial.setExtendedState(JFrame.MAXIMIZED_BOTH);  
144 mTelaInicial.setVisible( bln:true);  
145 }  
146  
147 private void btnCadastrarActionPerformed(java.awt.event.ActionEvent evt) {  
148     // TODO add your handling code here:  
149     String usuario, senha, confsenha;  
150  
151     usuario = txtCadUsuario.getText();  
152     senha = txtSenha.getText();  
153     confsenha = txtConfSenha.getText();  
154  
155     if (txtSenha.getText().equals( o:txtConfSenha.getText())){  
156         UsuarioDTO objusuario = new UsuarioDTO();  
157         objusuario.setNome_usuario( nome_usuario:usuario);  
158         objusuario.setSenha_usuario( senha_usuario:senha);  
159         objusuario.setConfsenha(confsenha);  
160  
161         UsuarioDAO objusuariodao = new UsuarioDAO();  
162         objusuariodao.cadastrarUsuario(objusuario);  
163     } else{  
164         JOptionPane.showMessageDialog( cmpnt:null, o:"As senhas não são iguais");  
165     }  
166 }  
167  
168  
169
```

FIGURA 18 - LOGIN

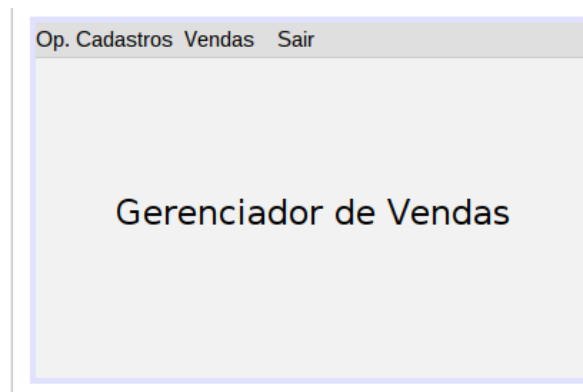


Login Gerenciador de Vendas

Usuario:

Senha:

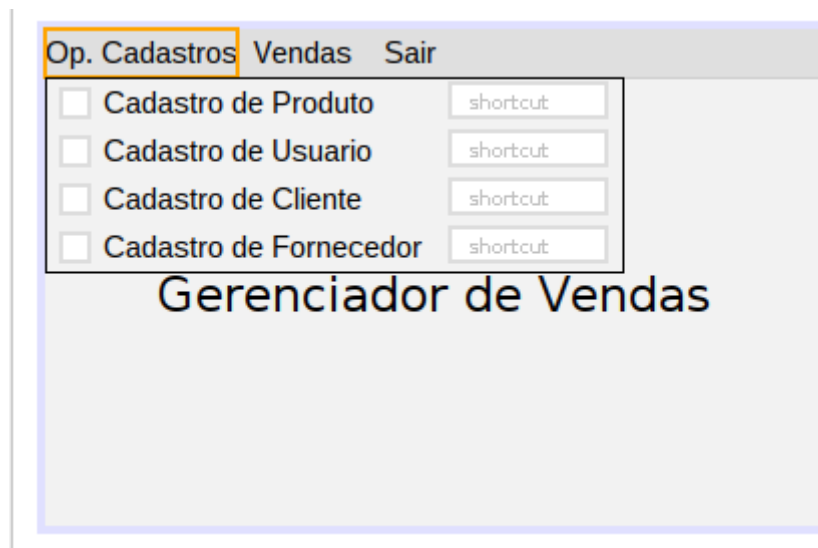
FIGURA 19 - TELA INICIAL



Op. Cadastros Vendas Sair

Gerenciador de Vendas

FIGURA 20 - TELA INICIAL NA OPÇÃO CADASTROS



Op. Cadastros Vendas Sair

<input type="checkbox"/>	Cadastro de Produto	<input type="text" value="shortcut"/>
<input type="checkbox"/>	Cadastro de Usuario	<input type="text" value="shortcut"/>
<input type="checkbox"/>	Cadastro de Cliente	<input type="text" value="shortcut"/>
<input type="checkbox"/>	Cadastro de Fornecedor	<input type="text" value="shortcut"/>

Gerenciador de Vendas

FIGURA 21 - TELA INICIAL NA OPÇÃO VENDAS

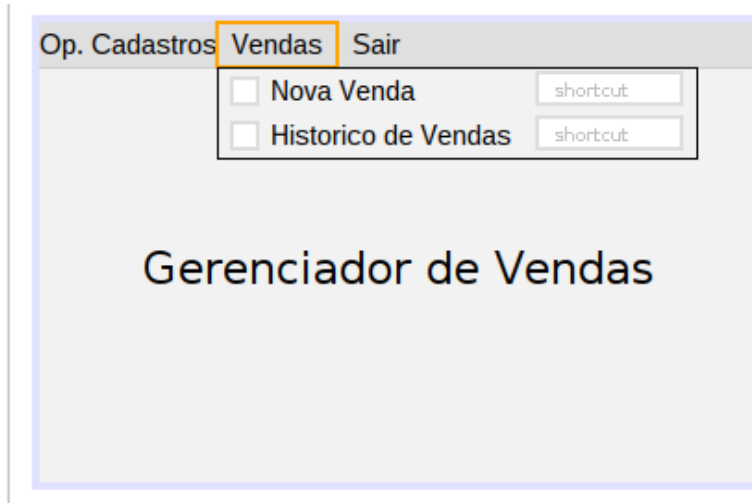


FIGURA 22 - TELA INICIAL NA OPÇÃO SAIR

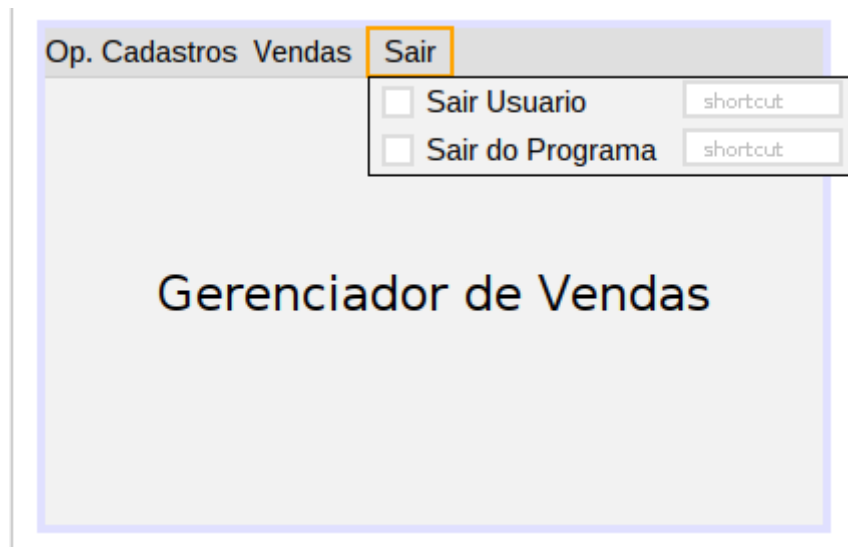


FIGURA 23 - TELA CADASTRO DE PRODUTO

Cadastro de Produto

Código interno: Código do fornecedor:

Descrição do produto:

Custo: Preço de Venda: Qtd. Estoque:

Código de Barras:

FIGURA 24 - TELA DE RELATÓRIO DE PRODUTOS CADASTRADOS

Relatorio de Cadastro de Produtos

ID do Produto	Descrição do ...	Custo Produto ...	Preço Produto ...	Esstoque do P...	Código de Barr...

FIGURA 25 - TELA DE CADASTRO DE USUÁRIOS

Cadastro de Usuario

Usuario:

Senha:

Confirme a senha:

FIGURA 26 - TELA DE CADASTRO DE CLIENTE

Cadastro de Clientes

Codigo Cliente:

Nome: Sobrenome:

CPF: RG:

Telefone: Celular:

Data de Nascimento: Ex: 22-10-2001

Endereço:

Rua: Nº:

Bairro: Cidade:

FIGURA 27 - TELA DE RELATÓRIO DE CADASTRO DE CLIENTES

Cód Clie...	Nome	Sobreno...	CPF	RG	Telefone	Celular	Data Na...	Rua	Numero	Bairro	Cidade

FIGURA 28 - TELA DE CADASTRO DE FORNECEDOR

Cadastro de Fornecedor

Codigo do Fornecedor:

Nome Fornecedor:

CPF/CNPJ:

Telefone:

Endereço: Rua:

Numero: Bairro:

Cidade:

FIGURA 29 - TELA DE RELATÓRIO DE CADASTRO DE FORNECEDOR

Relatorio de fornecedores cadastrados

Cod. For...	Nome F...	CPF/CN...	Telefone	Rua	Numero	Bairro	Cidade

FIGURA 30 - TELA DE VENDA

Venda

Código do Produto:

Descrição de Produto:

Preço: Estoque Disponível:



RELATÓRIO FINAL DAS ATIVIDADES DE EXTENSÃO

1. IDENTIDADE DA ATIVIDADE
RELATÓRIO: Sistema Empresarial
CURSO: Análise e Desenvolvimento de Sistemas e Gestão de Tecnologia da Informação
MÓDULO: Desenvolvimento Desktop
PROFESSOR RESPONSÁVEL: Sidney Gitcoff Telles
ESTUDANTE: Adryelle, Isabella, Leonardo V., Lucas A., Pedro R.
PERÍODO DE REALIZAÇÃO: 02/2023 a 04/2023

2. DESENVOLVIMENTO
Contextualização: No primeiro semestre do curso de ADS/GIT foi proposto um trabalho final (chamado projeto integrado) para estar sintetizando e aplicando o conteúdo do curso.
Desafio: Criar um sistema empresarial de controle de estoque e vendas para uma empresa local utilizando o paradigma de orientação a objetos e o modelo de banco de dados relacional, MySQL.
Cronograma das Ações: Durante o mês 02/2023 houve conversa entre o grupo para estar estabelecendo qual a empresa iria ser a modelo para o projeto e também com qual parte parte do projeto cada aluno ficaria responsável. No mês 03/2023 já estava estabelecido a responsabilidade de cada integrante, então foi o mês utilizado para desenvolvimento. No mês 04/2023 foi utilizado para finalizar detalhes e fazer envio do trabalho final.
Síntese das Ações: Adryelle e a Isabela ficaram responsáveis pela criação do banco de dados e atualização do relatório, enquanto o Leonardo, Lucas e Pedro ficaram com a parte das interfaces e os códigos em java.
a. Aspectos positivos: Aplicar o conteúdo teórico em um projeto real.
b. Dificuldades encontradas: Integração do banco de dados com java; como vender mais de um item por vez; tempo curto para aprender as linguagens (java e mysql); conseguir desenvolver um código funcional.
c. Resultados atingidos: Foi um resultado satisfatório, afinal o trabalho foi concluído com êxito onde obtivemos um programa funcional para gerenciamento de estoque de pequenos comercio, disponibilizado a opção de cadastrar os produtos, os clientes e fornecedores e registrar a vendas.

d. Sugestões / Outras observações:

3. EQUIPE DOS ESTUDANTES NO PROJETO

RA 1012023100203	NOME Isabella Donizetti Dardi
RA 1012023100153	NOME Leonardo Vitorino Figueiredo
RA 1012023100368	NOME Lucas Almeida Barbosa
RA 1012022200918	NOME Pedro Ricardo Nunes Oliveira
RA	NOME

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

Módulo Desenvolvimento Desktop

Cronograma de Validação - Projeto Integrado

Unidade Estudo	Participação no Projeto	Data da Validação
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
<u>Descrição do Projeto:</u> 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.		