

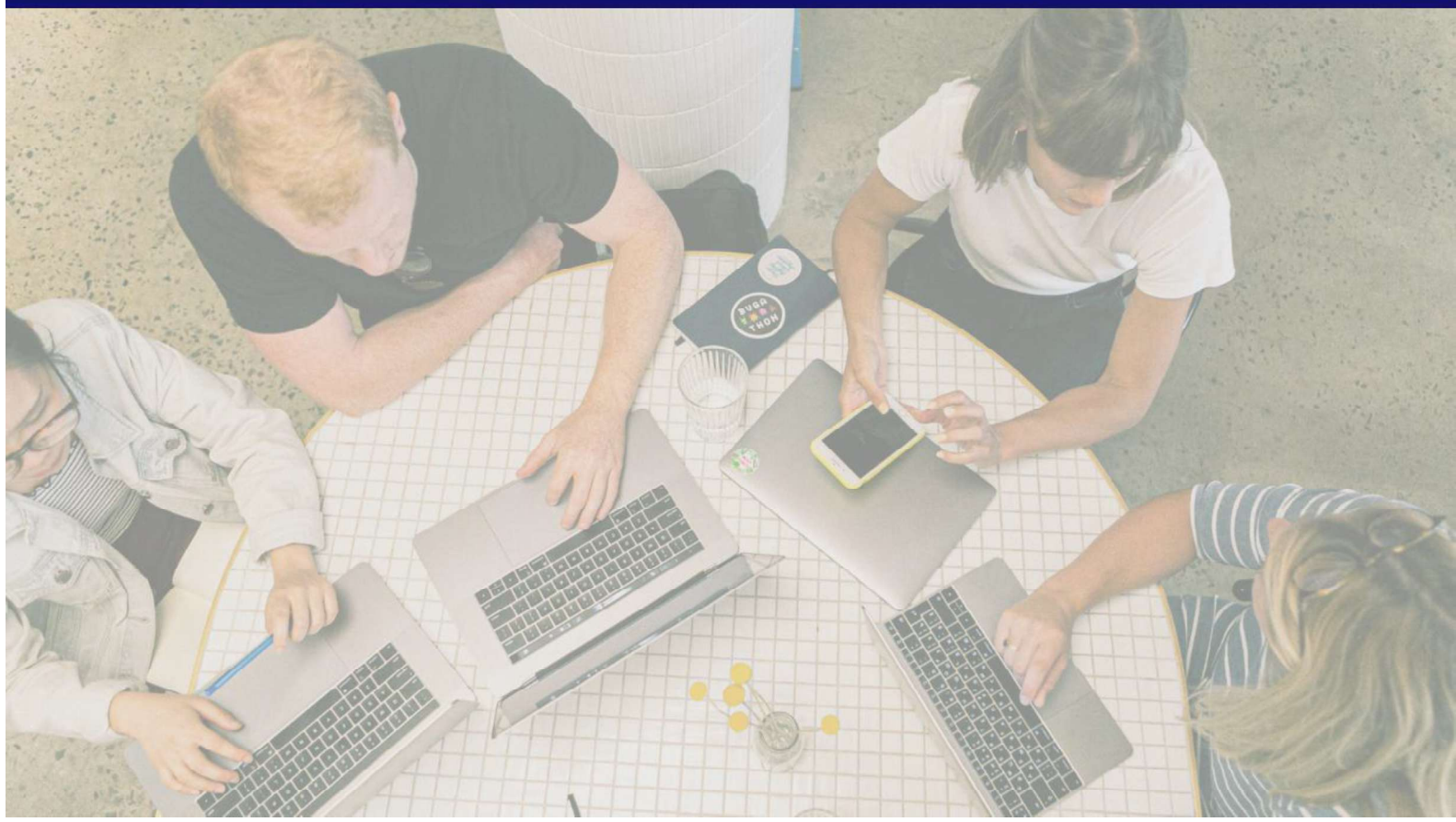


UNifeob
| ESCOLA DE NEGÓCIOS



2024

PROJETO INTEGRADO



UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS

ESCOLA DE NEGÓCIOS

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
CIÊNCIA DA COMPUTAÇÃO**

PROJETO INTEGRADO

**DESENVOLVIMENTO DE SOLUÇÕES CONSOLE
INTEGRADAS PARA EDUCAÇÃO,
SUSTENTABILIDADE, INCLUSÃO SOCIAL E
EMPREENDEDORISMO**

CASA DAS PORTAS E JANELAS

SÃO JOÃO DA BOA VISTA, SP

NOVEMBRO 2024

UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS

ESCOLA DE NEGÓCIOS

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
CIÊNCIA DA COMPUTAÇÃO**

PROJETO INTEGRADO

**DESENVOLVIMENTO DE SOLUÇÕES CONSOLE
INTEGRADAS PARA EDUCAÇÃO,
SUSTENTABILIDADE, INCLUSÃO SOCIAL E
EMPREENDEDORISMO**

CASA DAS PORTAS E JANELAS

MÓDULO MODELAGEM E DESENVOLVIMENTO DE SISTEMAS

Business Intelligence – Profª. Mariângela Martimbianco Santos

Programação Orientada a Objeto – Prof. Nivaldo de Andrade

Lógica de Programação – Prof. Marcelo Ciacco Almeida

Modelagem de Dados – Prof. Max Streicher Vallim

Projeto de Modelagem e Desenvolvimento de Sistemas – Profª. Mariângela M. Santos

Estudante: André Luis Bertelli Pena, RA 24000057

Estudante: Cleudio de Faria, RA 24001366

Estudante: Filipe Luís Poiatti, RA: 24001155

Estudante: Guilherme Ernando Ribeiro Alves, RA: 24002028

Estudante: Pedro Henrique de Paiva Fagoti, RA: 24001181

Estudante: Renan Augusto Cristi Avelar, RA: 24000603

SÃO JOÃO DA BOA VISTA, SP

NOVEMBRO 2024

SUMÁRIO

1. INTRODUÇÃO	5
3. PROJETO INTEGRADO	7
3.1 PROGRAMAÇÃO ORIENTADA A OBJETO	7
3.1.1 CLASSES E OBJETOS	8
CLASSES E OBJETOS	8
Classes	8
Objetos	8
3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO.	8
3.1.2.1 Atributos	8
3.1.2.2 Métodos	9
3.1.2.3 Encapsulamento	9
3.1.2.4 Herança	9
3.1.2.5 Polimorfismo	10
3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS	10
3.2 LÓGICA DE PROGRAMAÇÃO	10
3.2.1. Algoritmos	10
3.2.2. Variáveis	11
3.2.3. Tipos de Dados	11
3.2.4. Funções	11
3.2.5. Estruturas Condicionais	12
3.2.6. Operadores Lógicos	12
3.2.7. Operadores de Comparação	12
3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE	13
3.2.2 DESENVOLVIMENTO DE APLICAÇÕES	13
3.2.3 IMPLEMENTAÇÃO E VALIDAÇÃO	13
3.3 MODELAGEM DE DADOS	13
3.3.1 MODELO CONCEITUAL	14
3.3.2 MODELO LÓGICO E FÍSICO	15
3.3.3 SQL	17
3.4 BUSINESS INTELLIGENCE	18
3.4.1 ORGANIZAÇÃO E IDENTIFICAÇÃO DAS INFORMAÇÕES	19
3.4.2 MANIPULAÇÃO E ANÁLISE DE DADOS	19
3.4.3 CRIAÇÃO DE MODELOS DE ANÁLISE DE DADOS	19
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS	19
3.5.1 GERENCIANDO FINANÇAS	20
• Tópico 1: Introdução aos conceitos econômicos e financeiros básicos	20
• Tópico 2: Entendendo o ambiente: independência financeira, o valor da minha	

riqueza e o registro do dia a dia	20
● Tópico 3: Dívidas e juros compostos, opções de empréstimo e alternativas ao endividado	20
● Tópico 4: Estabelecer metas para a realização de seus sonhos e como envolver o grupo a que você pertence para atingir seus objetivos	21
3.5.2 ESTUDANTES NA PRÁTICA	22
4. CONCLUSÃO	23
REFERÊNCIAS	24

1. INTRODUÇÃO

A proposta do projeto, visa construir um software para fazer o gerenciamento de vendas dos produtos e cadastro dos clientes da loja parceira do projeto, visa criar programa para organizar e documentar de forma digital as vendas, pois a empresa utiliza uma maneira mais antiga de vendas,

Suas vendas são realizadas através de papeis, o cliente vai até a loja, realiza uma compra e o vendedor preenche um papel de recibo de 2 vias a mão e fornece para o cliente, a contra via do recibo é armazenada num suporte de espeto de papel, que contém os dados da venda, nome do cliente, se foi pago ou não a compra

Eles possuem 3 espetos de papel, um espeto para vendas já concluídas e pagas, outro espeto para as compras que serão entregues e outro espeto para as compras que já foram entregues.

Mediante dessa observação da maneira de como a empresa trabalha, foi ofertado para empresa o Projeto de Integração da Unifeob, foi explicado sobre o que era o projeto e qual a sua finalidade para ajudar a empresa parceira, o proprietário ficou um pouco desconfiado da proposta, mais foi acolhido e orientado pela equipe do projeto, foi explicado que se tratava de um projeto de fins acadêmicos dos alunos de Ciências da Computação e alunos de Análise e Desenvolvimento de Sistemas

O proprietário da empresa aceitou a proposta e assinou um documento oficial em parceria com alunos da Unifeob,

Foi proposto um projeto de software para automatizar via console a venda de produtos, controle de entregas de produtos, e cadastro dos clientes.

2. DESCRIÇÃO DA EMPRESA

A empresa a sofrer consultoria dos alunos do Projeto Integrado de Ciências da Computação e ADS no módulo MODELAGEM E DESENVOLVIMENTO DE SISTEMAS, é uma empresa de vendas de portas de janelas com nome e razão social Casa das Portas e Janelas com CNPJ 16.5913/1001 como proprietário e CEO da empresa, o senhor Marcos Edésio Todero.

A empresa possui sua sede em São João da Boa Vista, SP, endereço Rua 14 de julho, 1256 com filiais nas cidades de Espírito Santo do Pinhal, SP, Aguai, SP e Vargem Grande do Sul, SP.

A empresa já está no ramo de vendas de portas, janelas e telas para mosquitos há mais de 25 anos, fornecendo produtos para toda a região de São João da Boa Vista.

3. PROJETO INTEGRADO

3.1 PROGRAMAÇÃO ORIENTADA A OBJETO

No projeto sistema de Vendas e Cadastro de Clientes para uma pequena loja, com foco na Programação Orientada a Objetos (POO), os seguintes objetivos e metas devem ser alcançados:

1. Autenticação e Controle de Acesso:

- Foram aplicados os conceitos de encapsulamento e poliformismo na primeira tela do trabalho onde o vendedor ou gerente (administrador) tem acesso e privilégios diferentes no sistema, o usuário precisa ter um login e uma senha cadastrada no banco de dados para acessar o sistema de loja.

2. Manuseio Seguro de Senhas:

- Foram aplicados os conceitos de encapsulamento e abstração no projeto, na parte de validação da senha, onde usuário sabe que precisa ter uma senha, mais não sabe como funciona a criptografia

3. Gerenciamento de Produtos:

- No cadastro de produtos no sistema, foram criados herança da variável produto, exemplo: produto, janela, existem vários tipos de janelas,

4. Gerenciamento de Pedidos:

- No projeto foram criados status de entrega que visa saber se mercadoria foi entregue ou não, foram aplicados conceitos de abstração e herança.

○

5. Sistema de Pagamento:

- No projeto foram criadas classes respeitando o tipo de pagamento, a vista em dinheiro terá desconto de 10% que poderá ser alterado e no cartão divide em 10 vezes sem juros, respeitando os conceitos de polimorfismo e abstração

No projeto desenvolvido foi respeitado todos os tópicos acima mencionados, foram criado cadastro de login separado cadastro de vendedor e cadastro de gerente que seria o administrador, que tem mais privilégios de acesso no sistema,

3.1.1 CLASSES E OBJETOS

CLASSES E OBJETOS

Classes

1. **CriarConta** (em `criar_conta.py`)
 - Representa o sistema de criação de contas.
2. **SistemaLogin** (em `login.py`)
 - Classe abstrata para o sistema de login, define a estrutura básica para autenticação.
3. **Login** (em `login.py`)
 - Implementar a autenticação de usuário, carregamento de dados e recuperação de senha, herda de **SistemaLogin**.

Objetos

- `criar_conta = CriarConta()`
 - Instância da classe **CriarConta**, responsável pela criação de contas de usuários.
- `login = Login()`
 - Instância da classe **Login**, utilizada para realizar a autenticação e recuperação de senha.

3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO.

3.1.2.1 Atributos

1. **CriarConta:**
 - `self.max_tentativas_senha`: Número máximo de tentativas para definir uma senha.
 - `self.tentativas`: Conta o número de tentativas realizadas.
2. **Login:**
 - `self.max_tentativas`: Número máximo de tentativas de login.
 - `self.tentativas`: Conta o número de tentativas de login.

3.2.1.2 Métodos

1. **CriarConta:**

- `__init__()`: Construtor da classe que inicializa as variáveis.
- `validar_senha(senha)`: Valida a senha de acordo com regras específicas (mínimo de 8 caracteres, uma letra maiúscula e um caractere especial).
- `gerar_senha_aleatoria()`: Gera uma senha aleatória de 8 caracteres.
- `perguntar_gerar_senha()`: Pergunta ao usuário se deseja gerar uma senha aleatória.
- `criar_nova_conta()`: Responsável pela criação de uma nova conta, incluindo a senha e perguntas de segurança.
- `salvar_usuario(conta)`: Salva os dados da nova conta em um arquivo JSON.

2. **SistemaLogin** (Classe abstrata):

- `autenticar()`: Método abstracto que será implementado por subclasses para realizar a autenticação.

3. **Login**:

- `__init__()`: Construtor que inicializa o limite de tentativas.
- `carregar_usuarios()`: Carrega os usuários armazenados no arquivo JSON.
- `autenticar()`: Responsável pela autenticação do usuário, solicitando login e senha, além de validação das tentativas.
- `recuperar_senha(usuario)`: Realiza a recuperação de senha utilizando as perguntas de segurança.
- `redefinir_senha(usuario)`: Permite que o usuário defina uma nova senha após a recuperação.
- `salvar_usuario(usuario_atualizado)`: Atualiza os dados de um usuário no arquivo JSON.

3.1.2.3 Encapsulamento

- **Atributos privados**: Não são explicitamente indicados como privados no código (usando `__` ou `_`), mas os atributos de classes como `tentativas`, `max_tentativas_senha`, e `max_tentativas` representam um nível básico de controle sobre os dados internos.
- **Métodos**: O controle sobre como os dados são manipulados é feito através de métodos como `validar_senha()`, `gerar_senha_aleatoria()`, e `salvar_usuario()`.

3.1.2.4 Herança

- **Login** herda de **SistemaLogin**, implementando o método abstrato `autenticar()`.

3.1.2.5 Polimorfismo

- No código, o polimorfismo não é explicitamente explorado além do uso da herança e da sobrescrita de métodos (como `autenticar()`) na classe **Login**, que implementa de forma concreta a autenticação estipulada pela classe abstrata **SistemaLogin**.
 - No código, o polimorfismo não é explicitamente explorado além do uso da herança e da sobrescrita de métodos (como `autenticar()`) na classe **Login**, que implementa de forma concreta a autenticação estipulada pela classe abstrata **SistemaLogin**.

No desenvolvimento do projeto "Desenvolvimento de Soluções Console Integradas para Educação, Sustentabilidade, Inclusão Social e Empreendedorismo", você utilizará conceitos fundamentais da Programação Orientada a Objetos, incluindo Atributos, Métodos, Encapsulamento, Herança e Polimorfismo.

3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS

Foram utilizados métodos estáticos, públicos e privados no projeto respeitando a hierarquia no projeto, no sistema de login, na recuperação de senha, são aplicados todos esses conceitos.

3.2 LÓGICA DE PROGRAMAÇÃO

3.2.1. Algoritmos

Segundo os escritores (José Augusto N. G. Manzano; Jayr Figueiredo de no livro Algoritmos - Lógica para Desenvolvimento de Programação de Computadores no capítulo 2) o termo algoritmo, do ponto de vista computacional, pode ser entendido como regras formais, sequenciais e bem definidas a partir do entendimento lógico de um problema a ser resolvido por um programador com o objetivo de transformá-lo em um programa que seja possível de ser tratado e executado por um computador, em que dados de entrada são transformados em dados de saída.

Um **algoritmo** é uma sequência lógica de instruções para resolver um problema. Nos arquivos de código fornecidos, vemos algoritmos aplicados para:

- **Criar uma nova conta** no sistema.
- **Validar uma senha** com base em critérios definidos (comprimento, caracteres especiais, maiúsculas).
- **Autenticar um usuário** comparando as informações fornecidas (nome de usuário e senha) com as armazenadas no sistema.

O código executa esses passos em sequência, de forma organizada, o que define a lógica do programa.

3.2.2. Variáveis

As variáveis são fundamentais para armazenar e manipular dados no programa. Elas podem ser temporárias ou persistentes (como as armazenadas em arquivos).

Foram criadas variáveis globais e locais para projeto, respeitando a hierarquia, foi criado variáveis do tipo strings para armazenamentos de dados

Exemplos no código:

- usuário, email, senha: armazenam temporariamente as informações fornecidas pelo usuário.
- tentativas, max_tentativas_senha: controlam o número de tentativas disponíveis.
- usuarios: uma lista que armazena os dados carregados do arquivo JSON.

3.2.3. Tipos de Dados

Em Python, os **tipos de dados** podem variar de acordo com o uso. No código, temos:

- **Strings , Inteiros ,Listas:**
 - foram usados todos esses tipos de dados no projeto,

3.2.4. Funções

As **funções** são blocos de código reutilizáveis, que executam uma tarefa específica. No código, temos várias funções que encapsulam diferentes funcionalidades, como:

- `validar_senha(senha)` , `gerar_senha_aleatoria()` , `autenticar()` , `recuperar_senha()`

- Foram usados todos esses exemplos de funções citadas acima, pois na tela login precisa de recuperação de senha, precisa de cadastro de senha, precisa validar senha.
- Cada função é responsável por uma tarefa específica, facilitando a manutenção e a legibilidade do código.

3.2.5. Estruturas Condicionais

As **estruturas condicionais** são usadas para executar diferentes ações dependendo de uma condição lógica. No código, exemplos de condicionais incluem:

- `if self.perguntar_gerar_senha():` — Se o usuário optar por gerar uma senha aleatória.
- `if len(senha) < 8:` — Se a senha não atender ao comprimento mínimo, o programa exibe uma mensagem de erro.
- `if senha == usuario_encontrado['senha']:` — Se a senha fornecida pelo usuário estiver correta.
- Foram usadas estruturas condicionais no projeto, para decisões, exemplo, na tela de cadastro de login, se o vendedor ou gerente não tiver um login de conta, a estrutura de decisão entra para o usuário poder criar um login
- Essas estruturas ajudam o programa a tomar decisões e executar diferentes fluxos de código.

3.2.6. Operadores Lógicos

Qualquer código que se vai programar, precisa usar operadores lógicos, não existe código sem operadores lógicos. Foram usados operadores lógicos no projeto

3.2.7. Operadores de Comparação

Os **operadores de comparação** permitem comparar valores para verificar igualdade, desigualdade, ou outras relações. No código, temos operadores como:

- `==`: Usado para verificar igualdade (ex: `if senha == usuario_encontrado['senha']:`).
- `<`: Usado para verificar se um valor é menor que outro (ex: `if len(senha) < 8:`).

Esses operadores são essenciais para controle de fluxo no código.

Qualquer código que se vai programar, precisa usar operadores comparação, não existe código sem operadores comparação. Foram usados operadores lógicos no projeto

A lógica da programação é uma parte fundamental no início do desenvolvimento de sistemas. Nesse tópico, os estudantes aprenderão como funciona a lógica por trás dos computadores e como aplicá-la na prática. Serão abordados conceitos fundamentais, como algoritmos, variáveis, tipos de dados, funções, estruturas condicionais, operadores lógicos e operadores de comparação, voltados para a linguagem de programação Python.

3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE

Foram aplicados todos os conceitos sólidos de desenvolvimento de software nesse projeto, todos os ensinamentos aprendidos em aula, foram aplicados.

3.2.2 DESENVOLVIMENTO DE APLICAÇÕES

Foi realizada a regra de negócios após o levantamento de requisitos do projeto, realizado reuniões de equipe e definindo as regras de negócio.

3.2.3 IMPLEMENTAÇÃO E VALIDAÇÃO

Foi realizado teste no software desenvolvido, cadastrando contas aleatoriamente para validar o sistema, no cadastro, se o usuário não preencher todos os dados do input, aparece uma mensagem obrigatória para que o usuário digite os dados solicitados no sistema.

Foi testado a recuperação de senha , onde o usuário consegue cadastrar uma nova senha.

3.3 MODELAGEM DE DADOS

O banco de dados desenvolvido para esse projeto consiste em tabelas de cadastro de clientes, com endereço completo, uma tabela de data da venda do produto, data da entrega do produto, tabela de cadastro de produtos, uma tabela de status de entrega para saber se o produto já foi entregue ou vai ser entregue ao cliente ainda. A ideia é conectar banco de dados ao sistema, terá uma tabela de cadastro de login no sistema , onde o vendedor ou administrador da empresa terão cadastros e login diferenciados no sistema.

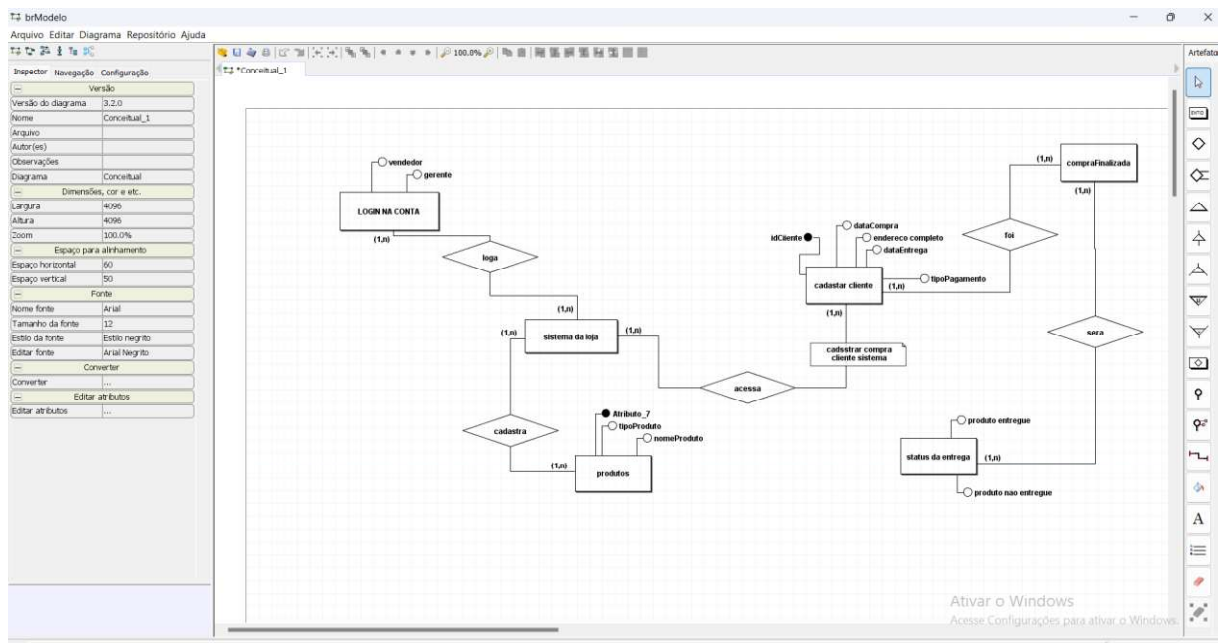
Segundo diz o autor Luís Damas no livro SQL - Structured Query Language, 6ª edição

É o sistema de gerência de banco de dados que vai fornecer o conjunto completo de serviços de acesso aos dados. Para isso, necessita de uma linguagem que permita ao usuário (pessoa ou aplicação) consultar ou operar sobre o banco de dados. É aqui que aparece a linguagem SQL (Structured Query Language).

3.3.1 MODELO CONCEITUAL

Segundo o autor Abraham Silberschatz do livro Sistema de Banco de Dados do capítulo 2 “ Um banco de dados relacional consiste em uma coleção de tabelas, cada qual recebendo um nome exclusivo. Por exemplo, considere a tabela instrutor da Figura 2.1, que armazena informações sobre instrutores. A tabela tem quatro cabeçalhos de coluna: ID, nome, nome_dept e salário (anual). Cada linha dessa tabela registra informações sobre um instrutor, consistindo na ID (identificação) do instrutor, nome, nome_dept e salário. De modo semelhante, a tabela curso da Figura 2.2 armazena informações sobre cursos, consistindo em id_curso, título, nome_dept e créditos, para cada curso ”

A equipe depois de realizar levantamento de requisitos junto com a empresa parceira do projeto, realizou um diagrama de entidade e relacionamento desenvolvido no BrModelo , foi aplicado os conhecimentos adquiridos em sala de aula. SEGUE O DIAGRAMA:

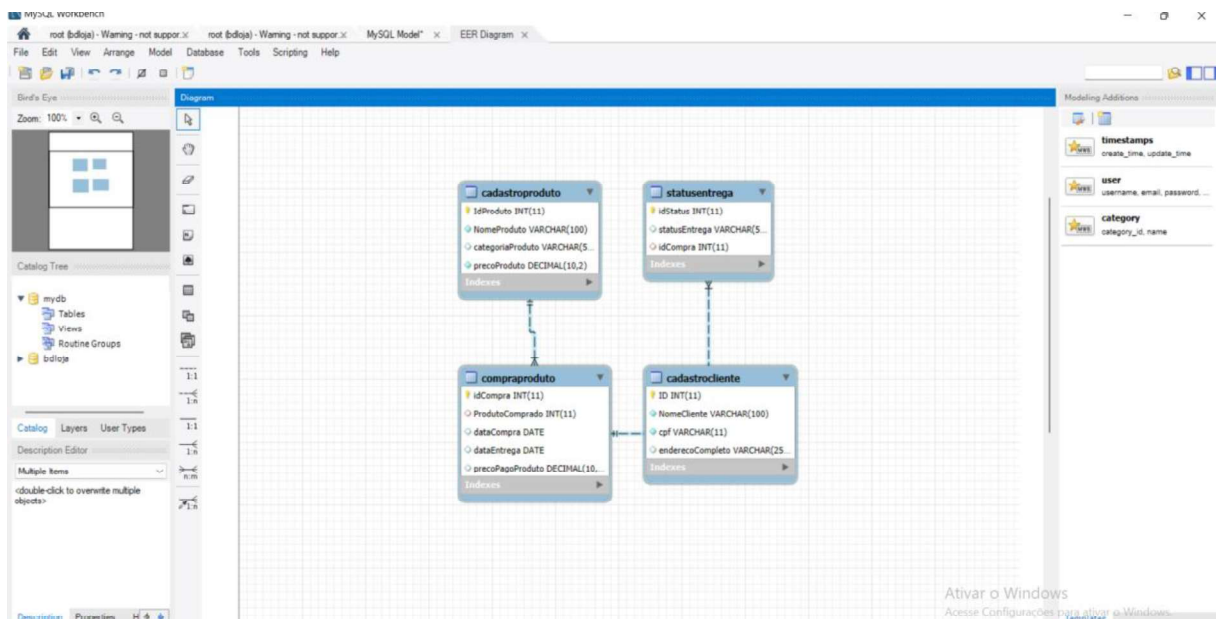


EXPLICAÇÃO DOS PRINCIPAIS RELACIONAMENTOS

O vendedor ou o gerente da loja loga no sistema , logado no sistema eles tem a opção de cadastrar produtos, ou cadastrar clientes e suas compras, inserindo dados de endereço

completo do cliente, data da compra e data da entrega, conseguem conferir o status de entrega da mercadoria, se esta entregue ou não.

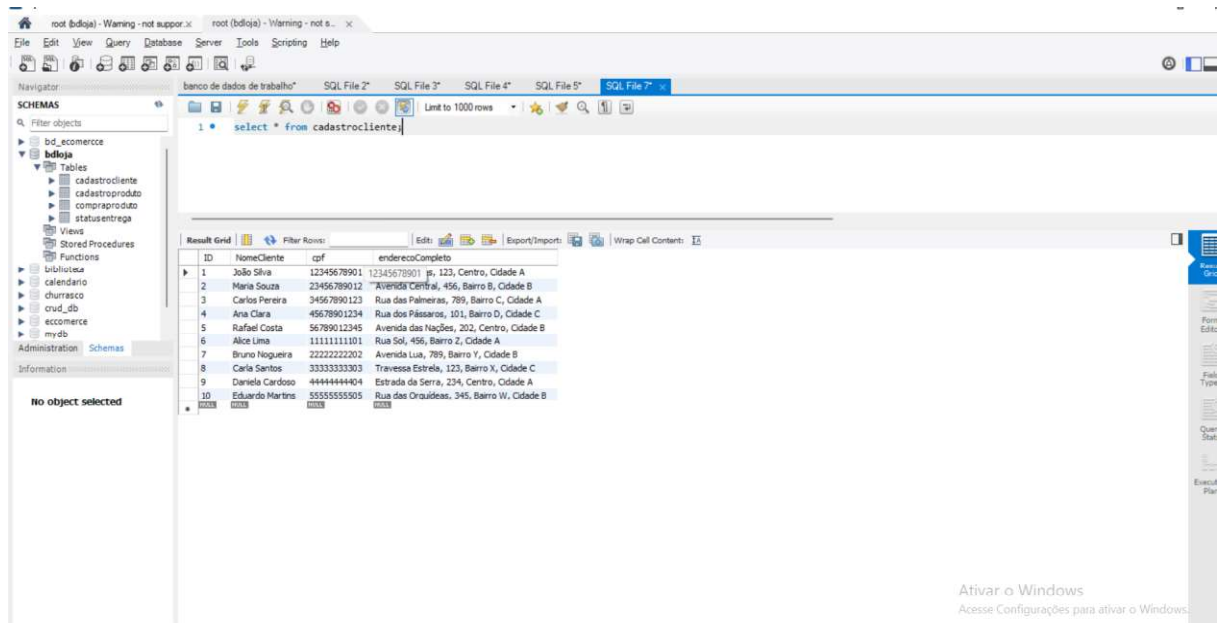
3.3.2 MODELO LÓGICO E FÍSICO



```

mysql-workbench
root (bdloja) - Warning - not suppor... root (bdloja) - Warning - not s... x
File Edit View Query Database Server Tools Scripting Help
banco de dados de trabalho SQL File 2* SQL File 3* SQL File 4* SQL File 5* SQL File 7
4 * USE bdloja;
5
6 -- Tabela CadastroCliente
7 * CREATE TABLE CadastroCliente (
8     ID INT AUTO_INCREMENT PRIMARY KEY,
9     NomeCliente VARCHAR(100) NOT NULL,
10    cpf VARCHAR(11) NOT NULL UNIQUE,
11    enderecoCompleto VARCHAR(255)
12 )
13
14 -- Tabela CadastroProduto
15 * CREATE TABLE CadastroProduto (
16     IdProduto INT AUTO_INCREMENT PRIMARY KEY,
17     NomeProduto VARCHAR(100) NOT NULL,
18     categoriaProduto VARCHAR(50),
19     precoProduto DECIMAL(10, 2) NOT NULL
20 )
21
22 -- Tabela CompraProduto
23 * CREATE TABLE CompraProduto (
24     idCompra INT AUTO_INCREMENT PRIMARY KEY,
25     ProdutoComprado INT,
26     dataCompra DATE,
27     dataEntrega DATE,
28     precoPagoProduto DECIMAL(10, 2),
29     FOREIGN KEY (ProdutoComprado) REFERENCES CadastroProduto(IdProduto)
30 )
31
32 -- Tabela StatusEntrega
33 * CREATE TABLE StatusEntrega (

```

A tabela login tem ligação com tabela sistema, após vendedor ou gerente estiver logado, ele acessa a tabela produtos ou a tabela cadastro cliente onde consegue cadastrar a compra do cliente, data da venda, data de entrega e status de entrega

Através do software de modelagem BrModelo e das orientações e aprendizado em aula do professor MAX, foi criado o modelo lógico e o modelo físico do projeto da empresa parceira. O banco de dados desenvolvido tinha entidades e seus atributos, com seus relacionamentos e cardinalidades, atributos seriam Cliente, Loja, Produto, dataVenda, Data Entrega, status Entrega.

O Modelo Físico foi gerado usando a ferramenta MYSQL WORKBENCH junto com os conhecimentos adquiridos em aula.

3.3.3 SQL

Através dos comandos básicos de SQL (Insert, Update, Delete e Select) foram utilizados para realização de inserção de dados nas tabelas no banco de dados desenvolvido para empresa parceira do projeto, foram usado comandos do SQL como SELECT * FROM (NOME DA TABELA), através desse comando, é possível visualizar a tabela criada,, comandos como CREATE TABLE (NOME DA TABELA), para criar tabelas, comando como INSERT INTO NOME DA TABELA (COLUNAS DE TABELAS) VALUES (DADOS A SEREM INSERIDOS), através dos comandos UPDATE E DELETE é possível inserir e deletar arquivos da tabela criada.

ALGUNS EXEMPLOS DE COMANDOS.... INSERT, DELETE E UPDATE.

- Comando para inserir valores na Tabela CadastroCliente.
 - INSERT INTO CadastroCliente (NomeCliente, cpf, enderecoCompleto) VALUES ('João Silva', '12345678901', 'Rua das Flores, 123, Centro, Sao Joao Boa Vista'),
- Criando o banco de dados da Loja
 - CREATE DATABASE IF NOT EXISTS bdLoja;
- Deletando tabelas
 - DELETE FROM CadastroCliente;
 - DELETE FROM CadastroCliente WHERE ID=3;
- Criando update na tabela cadastroCliente.
 - UPDATE cadastroCliente SET statusEntrega = 4 WHERE idCliente = 22;

ALGUMAS CONSULTAS AO BANCO DE DADOS CRIADO .

- SELECT NomeProduto ,categoriaProduto, precoProduto FROM bdloja.cadastroproduto;
- SELECT * FROM bdloja.statusentrega WHERE statusEntrega = 'Entregue';
- SELECT cp.idCompra, cp.dataCompra, cp.dataEntrega, cp.precoPagoProduto, p.NomeProduto FROM bdloja.compraproduto cp

3.4 BUSINESS INTELLIGENCE

Aplicando os fundamentos aprendidos em aula, trabalhando com software Power BI e banco de dados, foi realizada uma análise da empresa parceira do projeto.

É muito importante escolher a ferramenta correta para executar o trabalho, pois assim consegue apresentar um rendimento na execução dos dados a serem analisados; segundo diz o autor (JACK HYMAN no livro Microsoft Power BI Para Leigos, no capítulo 3) *Escolher a versão correta do Power BI é como visitar a maior loja de doces do mundo: você pode escolher entre muitas alternativas com nuances sutis. A escolha se resume a desejos, necessidades,*

escala e, é claro, dinheiro. Algumas versões são gratuitas (bem, mais ou menos), e outras são caras. E, é claro, a diferença mais óbvia é que algumas versões são baseadas em desktop ou servidor, enquanto outras oferecem recursos somente online.

Obter dados de maneira segura, é muito importante para uma análise de dados segura e confiável,

Segundo o que diz (JACK HYMAN no livro Microsoft Power BI Para Leigos, no capítulo 6) *Os dados podem ser complicados às vezes. É certo que carregar um único arquivo contendo algumas planilhas ou talvez um feed com um único fluxo de dados para transformar é brincadeira de criança. O que acontece, no entanto, quando você tem um conjunto de dados alojado em um aplicativo corporativo que tem transações continuamente escritas nele? Esse cenário é bem diferente. E as corporações devem se preocupar (por uma boa razão) com a integração e os resultados de business intelligence (BI). Com o Power BI, as organizações não precisam se preocupar com manipulações técnicas complexas quando se trata de seus sistemas de dados ou de suas comunicações com feeds de dados de terceiros. Como verá neste capítulo, a integração é fluida*

Os dados analisados são dados fictícios, que foram gerados com ajuda de inteligência artificial, porque a empresa não fornece dados para análise, porém estão relacionados diretamente com o banco de dados desenvolvido para o projeto da empresa parceira.

Através do software Power BI e conhecimentos adquiridos em aulas práticas, foi feita análise de quantidade de clientes cadastrados no banco de dados da empresa, foi feita uma análise de vendas de produtos da empresa, quais produtos foram vendidos no trimestre, qual vendedor vendeu mais durante o mês, quais lojas venderam mais durante o trimestre, análise de quais cidades compram mais produtos, com essa análise dados, a equipe do projeto visa orientar a empresa parceira num crescimento no aumento de vendas de 10% ao ano.

3.4.1 ORGANIZAÇÃO E IDENTIFICAÇÃO DAS INFORMAÇÕES

Através de entrevistas, conversas diretas com gerente e vendedores da loja e levantamento de requisitos, foram levantadas informações sobre clientes, vendas e produtos.

3.4.2 MANIPULAÇÃO E ANÁLISE DE DADOS

Foram usadas as informações das tabelas do Banco de Dados criado para o projeto e com ajuda da Inteligência Artificial, gerados dados fictícios com 100 dados, como nome do cliente, endereço completo do cliente, data da compra, data da venda do produto, valor do produto na loja, através desses dados, foram realizadas as análises da empresa parceira do projeto.

3.4.3 CRIAÇÃO DE MODELOS DE ANÁLISE DE DADOS

Foram criados dashboards dinâmicos usando a ferramenta Power BI através da análise de dados criados no banco de dados do projeto da empresa parceira.

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS

A Formação para a Vida é um dos eixos do Projeto Pedagógico de Formação por Competências da UNIFEOD.

Esta parte do projeto está diretamente relacionada com a extensão universitária, ou seja, o objetivo é que seja aplicável e que tenha real utilidade para a sociedade, de um modo geral.

3.5.1 GERENCIANDO FINANÇAS

- **Tópico 1: Introdução aos conceitos econômicos e financeiros básicos**

Este tópico apresenta os princípios fundamentais da economia e finanças, como renda, despesas, poupança e investimentos. Entender esses conceitos ajuda a equilibrar o orçamento pessoal e familiar.

Exemplo prático: Ao final do mês, uma pessoa faz um balanço de suas receitas (salário, vendas de itens, etc.) e despesas (contas de luz, água, supermercado). Com base nisso, decide economizar 10% de sua renda para criar uma reserva de emergência.

- **Tópico 2: Entendendo o ambiente: independência financeira, o valor da minha riqueza e o registro do dia a dia**

Este tópico explora a importância de controlar o fluxo financeiro para alcançar a independência financeira. Além disso, incentiva a prática de registrar receitas e despesas diariamente, para que se tenha clareza sobre o valor real da sua riqueza.

Exemplo prático: Manter um aplicativo ou planilha de controle financeiro onde são registradas todas as entradas e saídas de dinheiro, como gastos com alimentação, transporte, lazer. Isso ajuda a visualizar quanto sobra no fim do mês e se é possível investir ou poupar mais.

- **Tópico 3: Dívidas e juros compostos, opções de empréstimo e alternativas ao endividado**

Esse tópico aborda como lidar com dívidas, o impacto dos juros compostos, e alternativas para quem está endividado, como a renegociação de dívidas e a escolha consciente de empréstimos.

Exemplo prático: Uma pessoa com uma dívida no cartão de crédito, que acumula juros compostos, decide renegociar sua dívida com o banco, conseguindo reduzir a taxa de juros ou alongar o prazo de pagamento para tornar as parcelas mais acessíveis.

- **Tópico 4: Estabelecer metas para a realização de seus sonhos e como envolver o grupo a que você pertence para atingir seus objetivos**

Aqui, o foco está em definir metas financeiras claras e criar planos para alcançá-las. Envolver o grupo familiar ou a comunidade é essencial para o sucesso dessas metas.

Exemplo prático: Uma família deseja comprar uma casa própria em cinco anos. Eles definem metas de economia e todos os membros da família se comprometem a poupar uma parte da renda mensal para atingir esse objetivo.

Segundo (Rui Barbosa *Trecho retirado do livro "Obras Completas, Vol. XVI, Tomo VII. Rio de Janeiro: Ministério da Educação e Saúde, 1948.*) *O equilíbrio monetário, para ser estável, depende essencialmente de condições profundas, ligadas ao desenvolvimento econômico do país.*

Juntando os 4 tópicos da unidade de formação para Vida, chegamos à conclusão que precisamos economizar, saber administrar o nosso dinheiro, saber gastar com sabedoria, ter um orçamento organizado, anotar tudo o que compramos, e saber comprar, não adianta comprar um tênis de marca cara que custou 1/5 do seu salário e depois ficar devendo a conta de água e energia da sua casa. Não adianta parcelar no cartão de crédito em 12 vezes e ficar devendo para o banco com juros compostos, pagar somente o mínimo da fatura do cartão de crédito, é o maior erro que uma pessoa comete, pois entrara numa dívida que talvez não irá conseguir pagar, as pessoas precisam aprender gestão financeiras, gestão de gastos, precisam ter aulas sobre educação financeira, para saber administrar o dinheiro da forma correta. Se você aprender a gerir seus gastos, com certeza realizará seus sonhos de ter uma vida melhor e confortável.

Segundo(Lima Barreto *Trecho retirado do livro "Lima Barreto Toda Crônica". Vol II. Rio de Janeiro: Agir, 2004.*) “*A economia é a base da prosperidade, sentença que está nos vinténs e, por estar em tão reles moeda, ninguém a cumpre.*”

3.5.2 ESTUDANTES NA PRÁTICA

Através do estudo Gerenciando suas Finanças, a equipe optou por realizar um banner informativo sobre gastos, com o tema 5 passos para o sucesso financeiro.

1. Economizar
 - a. Organize seus ganhos e gastos
2. Investir
 - a. Invista parte do seu dinheiro
3. Controle as Dívidas
 - a. Tenha cuidado com as compras no crédito
4. Controle o Orçamento
 - a. Controle seu orçamento, saiba gastar corretamente
5. Equilibrar Finanças

- a. Controle o que você ganha com o que você gasta

Link público do banner desenvolvido

https://www.canva.com/design/DAGR5ZgPu84/KWIYP6H-1Zg6zpaKHSLbDA/view?utm_content=DAGR5ZgPu84&utm_campaign=designshare&utm_medium=link&utm_source=editor

4. CONCLUSÃO

Depois de algumas reuniões com a empresa parceira do projeto, foi mostrado um escopo do projeto, a empresa ficou um pouco desconfiada, mas disse que ano que vem de 2025 vai tentar implementar o projeto na empresa. Disseram que precisam organizar algumas coisas, mas que gostaram da ideia e do projeto, agradeceram a faculdade Unifeob e os alunos envolvidos no projeto. Diante dessa afirmativa tão positiva da empresa, concluímos que o projeto foi bem aceito e a equipe ficou muito feliz com a afirmativa positiva.

REFERÊNCIAS

BARBOSA, Rui *Trecho retirado do livro "Obras Completas, Vol. XVI, Tomo VII. Rio de Janeiro: Ministério da Educação e Saúde, 1948. disponível em <https://www.migalhas.com.br/literarias/frase/rui-barbosa-volume-i/o-meio-intelectual-modifica-se-com-inaudita-presteza> Acesso em 08/11/2024*

BARRETO, Lima. Lima Barreto. São Paulo: Autêntica Editora, 2017. E-book. p.Capa. ISBN 9788551302705. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788551302705/>. Acesso em: 08 nov. 2024.

HYMAN, Jack. Microsoft Power BI Para Leigos. Rio de Janeiro: Editora Alta Books, 2023. E-book. p.41. ISBN 9788550820309. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788550820309/>. Acesso em: 08 nov. 2024.

DAMAS, Luís. SQL - Structured Query Language. 6th ed. Rio de Janeiro: LTC, 2007. E-book. p.29. ISBN 9788521632450. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788521632450/>. Acesso em: 08 nov. 2024.

SILBERSCHATZ, Abraham. Sistema de Banco de Dados. 7th ed. Rio de Janeiro: GEN LTC, 2020. E-book. p.24. ISBN 9788595157552. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788595157552/>. Acesso em: 08 nov. 2024.

MANZANO, José Augusto Navarro G.; OLIVEIRA, Jayr Figueiredo de. Algoritmos - Lógica para Desenvolvimento de Programação de Computadores. 29th ed. Rio de Janeiro: Érica, 2019. E-book. p.31. ISBN 9788536531472. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788536531472/>. Acesso em: 08 nov. 2024.