

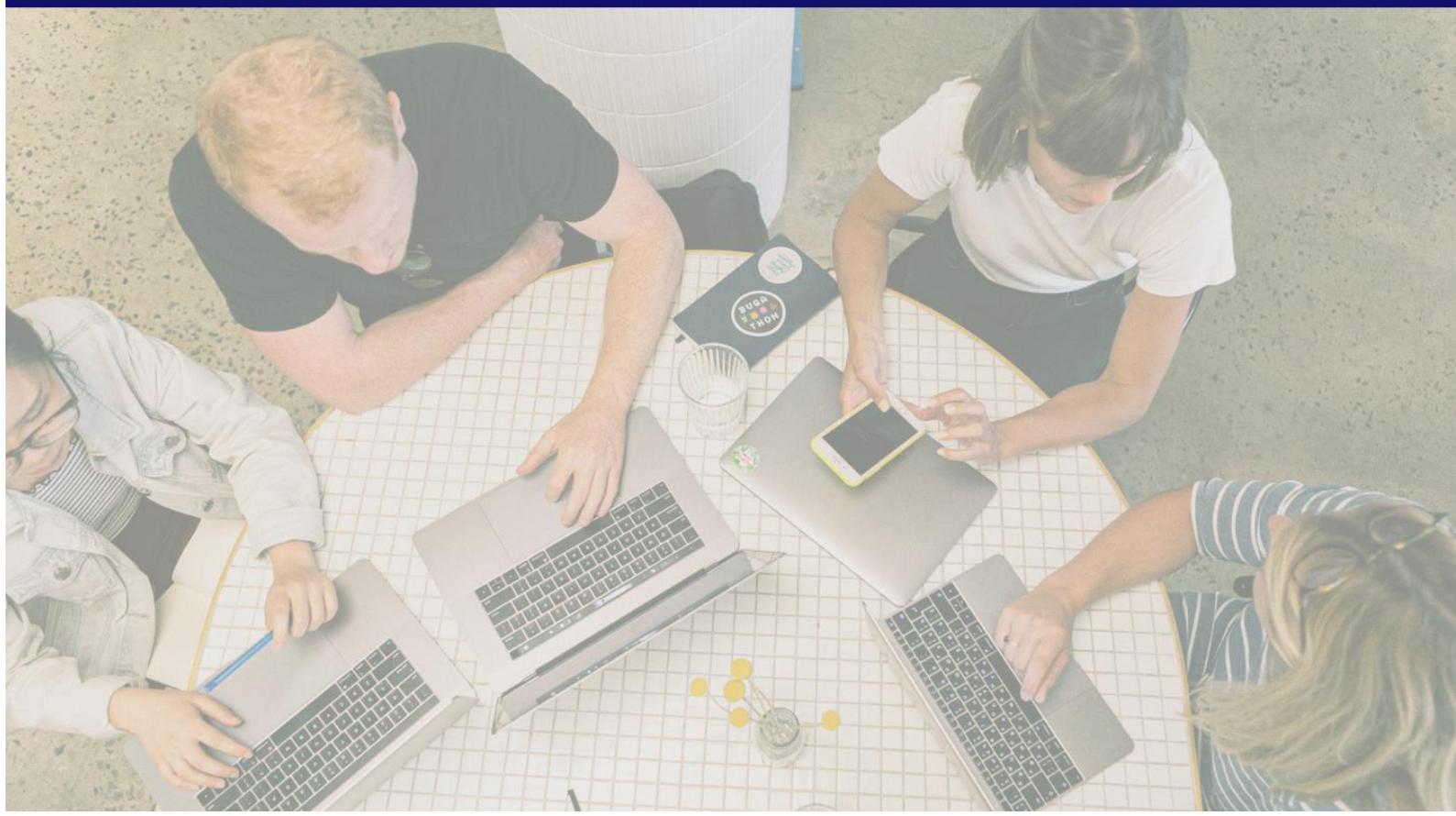


**UNifeob**  
| ESCOLA DE NEGÓCIOS



2024

# PROJETO INTEGRADO



**CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO OCTÁVIO BASTOS  
ESCOLA DE NEGÓCIOS**

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

**PROJETO INTEGRADO**

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

**<EMPRESA>**

**SÃO JOÃO DA BOA VISTA-SP**

**2024**

**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**

**<EMPRESA>**

**MÓDULO MODELAGEM E DESENVOLVIMENTO DE SISTEMAS**

Business Intelligence – Prof<sup>ª</sup>. 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 – Prof<sup>ª</sup>. Mariângela M. Santos

Estudantes:

Eduardo Peres Filho , RA 24001915

João V R Avello Correia, RA 24001368

Júlio Braidó Dutra, RA 24001364

Luis O Delapieri Meneghin, RA 24000053

**SÃO JOÃO DA BOA VISTA-SP**

**2024**

## SUMÁRIO

|   |    |
|---|----|
| 1. INTRODUÇÃO   | 4  |
| 2. DESCRIÇÃO DA EMPRESA   | 7  |
| 3. PROJETO INTEGRADO  | 8  |
| 3.1 PROGRAMAÇÃO ORIENTADA A OBJETO                                | 8  |
| 3.1.1 CLASSES E OBJETOS   | 8  |
| 3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO. | 9  |
| 3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS                      | 10 |
| 3.2 LÓGICA DE PROGRAMAÇÃO   | 10 |
| 3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE       | 11 |
| 3.2.2 DESENVOLVIMENTO DE APLICAÇÕES                               | 11 |
| 3.2.3 IMPLEMENTAÇÃO E VALIDAÇÃO                                   | 11 |
| 3.3 MODELAGEM DE DADOS  | 11 |
| 3.3.1 MODELO CONCEITUAL   | 12 |
| 3.3.2 MODELO LÓGICO E FÍSICO                                      | 12 |
| 3.3.3 SQL   | 12 |
| 3.4 BUSINESS INTELLIGENCE   | 12 |
| 3.4.1 ORGANIZAÇÃO E IDENTIFICAÇÃO DAS INFORMAÇÕES                 | 12 |
| 3.4.2 MANIPULAÇÃO E ANÁLISE DE DADOS                              | 12 |
| 3.4.3 CRIAÇÃO DE MODELOS DE ANÁLISE DE DADOS                      | 13 |
| 3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS        | 13 |
| 3.5.1 GERENCIANDO FINANÇAS  | 13 |
| 3.5.2 ESTUDANTES NA PRÁTICA                                       | 14 |
| 4. CONCLUSÃO  | 16 |
| REFERÊNCIAS   | 17 |
| ANEXOS  | 18 |

## **1. INTRODUÇÃO**

Este projeto visa o desenvolvimento de uma aplicação de controle de estoque de gás, implementada em Python e estruturada com foco em Programação Orientada a Objetos (POO) (ALMEIDA, 2023). A aplicação permite gerenciar o inventário de diferentes tipos de gás, oferecendo funcionalidades como autenticação de usuários, adição e remoção de itens do estoque e consulta das quantidades disponíveis. Utilizando princípios de POO como encapsulamento, herança e polimorfismo, o sistema foi projetado para ser modular e eficiente, promovendo a manutenção e expansão do código (ALMEIDA, 2023).

Com o uso de banco de dados MySQL, a aplicação garante a persistência dos dados e facilita a integridade e acessibilidade das informações (ORACLE, 2024). Além disso, o projeto incorpora a integração com o Power BI para análise e visualização dos dados de estoque. Isso permite que os usuários possam criar relatórios visuais, analisar padrões de consumo e tomar decisões baseadas em dados, impulsionando a eficiência operacional e oferecendo uma visão estratégica sobre o uso e a reposição dos recursos (GLAUCO, 2017). Assim, o projeto combina lógica de programação e visualização de dados para fornecer uma solução prática e sustentável.

## **2. DESCRIÇÃO DA EMPRESA**

A empresa Goncalves & Goncalves - Comercio E Distribuidora De Gas Ltda que tem como razão social Goncalves & Goncalves - Comercio E Distribuidora De Gas Ltda foi fundada em 13/11/1985 e está cadastrada na Solutudo no segmento de Gás com o CNPJ 55.160.824/0001-10. No mercado, a empresa está localizada na Rua General Osorio, Nº 272 no bairro Centro em Aguaí - SP, CEP 13860-000. A empresa Goncalves & Goncalves - Comercio E Distribuidora De Gas Ltda está cadastrada na Receita Federal sob o CNAE 4784-9/00 com atividade fim de Comércio Varejista De Gás Liquefeito De Petróleo (Glp).

## **3. PROJETO INTEGRADO**

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

### **3.1 PROGRAMAÇÃO ORIENTADA A OBJETO**

A Programação Orientada a Objetos (POO) possibilita uma organização estruturada do código por meio de conceitos como classes, objetos, herança e polimorfismo, facilitando o desenvolvimento de sistemas modulares e escaláveis (ALMEIDA, 2023). Utilizando esses princípios, implementamos um sistema de controle de estoque de gás, onde cada funcionalidade e entidade do sistema foi modelada em uma classe específica. Esse sistema exemplifica como a POO pode ser aplicada para gerenciar o estoque de maneira eficiente, mantendo o código organizado e de fácil manutenção.

#### **3.1.1 CLASSES E OBJETOS**

A utilização de classes e objetos no sistema tem como principal objetivo organizar e estruturar o código de maneira eficiente. Cada classe representa uma entidade específica e permite modularizar o sistema, facilitando a manutenção e a escalabilidade.

##### **Classe Usuário**

A classe Usuário é responsável pela autenticação no sistema, gerenciando os dados de login do usuário (nome de usuário e senha). Seus métodos incluem a verificação e realização do login.

Impacto: Centraliza a lógica de autenticação, facilitando futuras implementações relacionadas à segurança, como validação de senhas ou recuperação de contas.

##### **Classe Gas**

A classe Gas representa os diferentes tipos de gás disponíveis no sistema. Possui métodos para adicionar e remover quantidades de gás conforme necessário.

Impacto: Organiza o controle de gás e permite manipulações de forma segura, sem impactar outras funcionalidades. Facilita a expansão do sistema para incluir novos tipos de gás, caso necessário.

### Classe Estoque

A classe Estoque gerencia a quantidade de gás disponível, permitindo adicionar, remover e listar itens no estoque.

**Impacto:** Centraliza a lógica de controle do estoque, evitando erros de gestão e tornando o processo mais eficiente. Facilita a implementação de novas funcionalidades, como alertas de baixo estoque.

### Facilidades Proporcionadas

**Modularidade:** Cada classe possui uma responsabilidade específica, tornando o código mais organizado e fácil de entender.

**Reusabilidade:** As classes podem ser reutilizadas em outras partes do sistema ou em novos projetos.

**Facilidade de Manutenção:** Alterações em uma classe não afetam as demais, tornando a manutenção mais simples e rápida.

**Escalabilidade:** Novas funcionalidades podem ser adicionadas de forma simples, sem a necessidade de grandes modificações no código existente.

## **3.1.2 ATRIBUTOS, MÉTODOS, ENCAPSULAMENTO, HERANÇA E POLIMORFISMO.**

No desenvolvimento do projeto, utilizamos conceitos de Atributos, Métodos, Encapsulamento, Herança e Polimorfismo para garantir a organização e funcionalidade do sistema (HENRIQUE, 2023).

**Atributos:** As classes Usuario, Gas e Estoque possuem atributos que armazenam informações específicas. Por exemplo, Usuario tem atributos para user e senha, enquanto Gas tem tipo e quantidade.

**Métodos:** Cada classe possui métodos que definem as ações que podem ser realizadas. Por exemplo, Usuario possui o método login(), Gas possui adicionar\_quantidade() e remover\_quantidade(), e Estoque possui adicionar\_item(), remover\_item() e listar\_itens().

**Encapsulamento:** Os atributos da classe Usuario são privados, garantindo que os detalhes internos sejam ocultos e acessíveis apenas através de métodos públicos, como login().

**Herança:** Embora não tenhamos utilizado herança explicitamente no código fornecido, você poderia criar subclasses de Gas para tipos específicos de gás, herdando comportamentos comuns.

**Polimorfismo:** Você pode implementar polimorfismo definindo métodos com o mesmo nome em subclasses diferentes, adaptando comportamentos conforme necessário.

### 3.1.3 MÉTODOS ESTÁTICOS, PÚBLICOS E PRIVADOS

Os métodos implementados no sistema são organizados em três categorias:

**Métodos Estáticos:** No código atual, não utilizamos métodos estáticos. Contudo, você pode adicionar métodos que não dependem de uma instância específica da classe, como um método que calcula o total de gás em estoque.

**Métodos Públicos:** Métodos como adicionar\_item(), remover\_item() e listar\_itens() na classe Estoque são públicos, permitindo que outras partes do código interajam com o sistema de controle de estoque.

**Métodos Privados:** Na classe Usuario, o método login() utiliza atributos privados \_\_user e \_\_senha, que não são acessíveis diretamente fora da classe. Isso ajuda a manter a integridade dos dados.

## 3.2 LÓGICA DE PROGRAMAÇÃO

A lógica de programação é essencial para a construção de qualquer sistema, pois define a sequência lógica das operações e o fluxo de decisões que o programa seguirá. Nesta etapa, exploramos conceitos fundamentais como algoritmos, variáveis, tipos de dados, funções, estruturas condicionais, operadores lógicos e operadores de comparação, aplicando-os na linguagem escolhida para o desenvolvimento (PAIVA et al., 2020). Esses elementos básicos formam a base para a criação de funcionalidades robustas e permitem que o sistema funcione de forma lógica e estruturada.

### 3.2.1 CONCEITOS FUNDAMENTAIS DO DESENVOLVIMENTO DE SOFTWARE

#### Algoritmos

Os algoritmos são sequências de instruções que descrevem como realizar uma tarefa específica. O código inclui várias funções que representam esses algoritmos, como:

### 3.2.2 DESENVOLVIMENTO DE APLICAÇÕES

Os estudantes devem apresentar as regras de negócios definidas para a aplicação a ser desenvolvida, implementando a lógica detalhada nos algoritmos e definindo as funcionalidades que compõem a aplicação final.

#### **Regras** **de** **Negócio**

As regras de negócio do sistema de controle de estoque de gás incluem:

- Um usuário deve se autenticar antes de acessar o sistema;
- A quantidade de gás em estoque deve ser atualizada após cada adição ou remoção;
- O sistema deve permitir listar todos os itens disponíveis no estoque para consulta rápida e controle de inventário.

#### **Funcionalidades** **e** **Módulos**

As funcionalidades principais do sistema incluem:

- **Autenticação de Usuário:** Implementada na classe Usuário, permitindo que o usuário acesse o sistema mediante validação de login e senha;
- **Gerenciamento de Estoque:** A classe **Estoque** permite o gerenciamento do estoque de gás, com funções para adicionar, remover e listar itens de maneira estruturada.

Essa estrutura modular possibilita a reutilização de código, facilitando a manutenção e a expansão do sistema no futuro.

### 3.2.3 IMPLEMENTAÇÃO E VALIDAÇÃO

A etapa de implementação e validação foca na junção de todos os componentes desenvolvidos, garantindo que a aplicação funcione como um todo coeso e eficiente.

## Implementação dos Módulos

Os módulos implementados incluem:

- Usuário: Gerencia a autenticação do usuário;
- Estoque: Gerencia a quantidade de gás em estoque, com métodos para adicionar e remover.

```
class Estoque:
    def __init__(self):
        self.estoque = {}

    def adicionar_item(self, gas, quantidade):
        self.estoque[gas.tipo] = self.estoque.get(gas.tipo, 0) + quantidade

    def remover_item(self, gas, quantidade):
        if gas.tipo in self.estoque and self.estoque[gas.tipo] >= quantidade:
            self.estoque[gas.tipo] -= quantidade
```

## Testes e Validação

Os testes devem ser realizados para assegurar que:

- O login do usuário funcione corretamente;
- As operações de adição e remoção de gás funcionem como esperado;
- As listagens reflitam o estado atual do estoque.

```
# Exemplo de teste para a função de adição
usuario = Usuario("admin", "admin")
if usuario.login():
    estoque = Estoque()
    gas1 = Gas("Gas Natural")
    estoque.adicionar_item(gas1, 50)
    estoque.listar_itens()
```

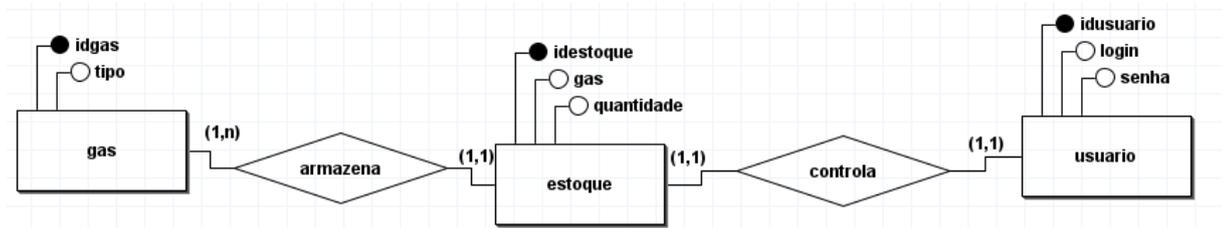
## 3.3 MODELAGEM DE DADOS

Nessa etapa, desenvolvemos o banco de dados do sistema de controle de estoque de gás, começando pelo modelo lógico e avançando até o modelo físico. Definimos tabelas e as relações entre entidades, como usuários e tipos de gás, e implementamos a estrutura no MySQL para assegurar a integridade e consistência dos dados. Essa modelagem facilita o armazenamento e o acesso seguro e eficiente das informações, essenciais para as operações do sistema.(ORACLE BRASIL, 2024)

### 3.3.1 MODELO CONCEITUAL

O conceito do banco de dados foi criado a partir da ideia de controlar o estoque de uma fornecedora de gás, pensando nos tipos que tem de gás no estoque e com a necessidade de apenas um usuário, com as devidas cardinalidades.(IBM, 2024)

Em seguida, o conceito do banco de dados:



### 3.3.2 MODELO LÓGICO E FÍSICO

- Modelo Lógico - Para a criação do Modelo Lógico foi utilizado a função CREATE TABLE porquê acreditamos que seja mais prático(W3School, 2024), em seguida o modelo:

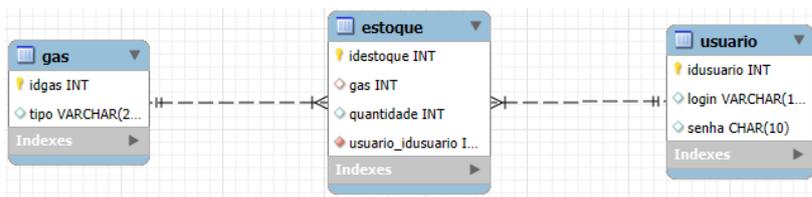
```
CREATE TABLE gas (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    tipo VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE estoque (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    id_gas INT,  
    quantidade INT NOT NULL,
```

```
FOREIGN KEY (id_gas) REFERENCES gas(id) ON DELETE CASCADE
);
```

```
CREATE TABLE usuario (
  id INT AUTO_INCREMENT PRIMARY KEY,
  login VARCHAR(50) NOT NULL UNIQUE,
  senha VARCHAR(255) NOT NULL
);
```

- Modelo Físico - Criado através do Modelo Lógico utilizando o método Reverse Engineer do MySQL Workbench(MySQL, 2024), em seguida o modelo físico:



### 3.3.3 SQL

Os códigos usados são feitos pensando no controle de estoque, de uma maneira que seja prático e fácil o controle dele

Os códigos gerenciam o banco de dados no MySQL e realizam operações no estoque. O código SELECT junto com o JOIN mostra o estoque e retorna o ID, tipo de gás e a quantidade de cada item juntando a tabela elementos da estoque com a da tabela gás. É usado o código SELECT na quantidade do estoque para saber qual unidade iremos mexer, em seguida se usa o UPDATE para alterar os dados no Data Base, o mesmo servindo tanto para somar ou consumir o estoque.

-Código para ver o estoque(Select e Join):

```
SELECT e.idestoque, g.tipo, e.quantidade FROM estoque e INNER JOIN gas g ON e.gas = g.idgas;
```

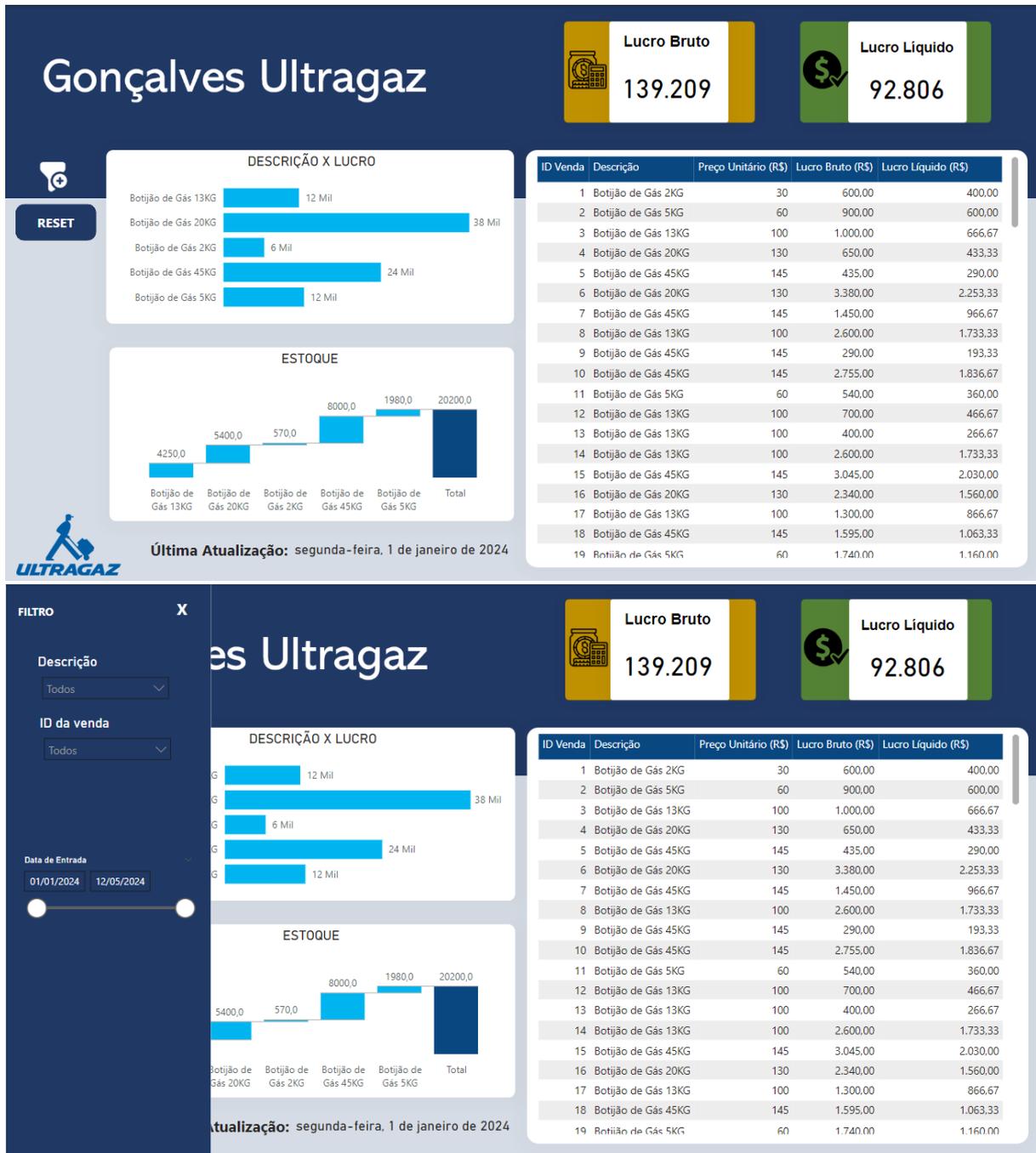
-Código para adicionar ou tirar (Select):

```
SELECT quantidade FROM estoque WHERE idestoque = %s
```

-Código para atualizar(Update):

UPDATE estoque SET quantidade = %s WHERE idestoque = %s

### 3.4 BUSINESS INTELLIGENCE



Nesta parte do projeto, foi desenvolvido um dashboard interativo utilizando Power BI para analisar e mensurar informações sobre o controle de estoque, vendas e lucros de uma loja

de botijões de gás (GLAUCO, 2017). O objetivo principal do dashboard é proporcionar insights claros e práticos para a gestão, facilitando o entendimento das operações diárias e auxiliando na tomada de decisões estratégicas.

### **3.4.1 ORGANIZAÇÃO E IDENTIFICAÇÃO DAS INFORMAÇÕES**

Foram organizadas e identificadas as seguintes informações importantes para o monitoramento do estoque e desempenho de vendas da loja:

- Descrição do Produto: Identificação dos diferentes tipos de botijões de gás (2KG, 5KG, 13KG, 20KG, 45KG);
- Preço Unitário: Valor de venda de cada botijão de gás;
- Lucro Bruto: Diferença entre o preço de venda e o custo de aquisição;
- Lucro Líquido: Lucro bruto subtraído de eventuais custos adicionais;
- Estoque Atual: Quantidade de cada tipo de botijão disponível em estoque;
- Data de Entrada: Data de reposição do estoque;
- Vendas no Mês: Quantidade de botijões vendidos em um período determinado.

Essas informações foram priorizadas por serem diretamente relacionadas ao controle de vendas e gestão do estoque, que são fundamentais para o funcionamento da loja.

### **3.4.2 MANIPULAÇÃO E ANÁLISE DE DADOS**

Para a criação do dashboard, foi realizada a manipulação da base de dados utilizando a tabela desenvolvida em Excel com as informações de botijões de gás. O processo incluiu:

- Limpeza de Dados: Ajustes em inconsistências nos dados, como formatações incorretas e valores duplicados;
- Consolidação: Agrupamento das informações em categorias relevantes para facilitar a visualização, como capacidade do botijão (em KG), fornecedor, e datas de venda;
- Criação de Indicadores: Foram desenvolvidos indicadores de lucro bruto, lucro líquido, vendas mensais e controle de estoque. Esses indicadores permitem uma análise aprofundada do desempenho de cada produto e sua lucratividade.

### **3.4.3 CRIAÇÃO DE MODELOS DE ANÁLISE DE DADOS**

O dashboard criado no Power BI inclui uma série de visualizações dinâmicas que permitem aos usuários monitorar os dados de forma eficiente:

- Tabela de Produtos: Exibe a descrição do botijão, preço unitário, lucro bruto e lucro líquido. Isso permite identificar facilmente quais produtos são mais rentáveis;
- Gráfico de Controle de Estoque: Um gráfico de barras foi utilizado para mostrar o nível de estoque atual de cada produto. Com isso, os gestores podem identificar rapidamente quais itens estão com baixo estoque e precisam ser repostos;
- Filtros Dinâmicos: Um botão de filtros foi implementado para que o usuário possa filtrar os dados por intervalo de datas ou pela descrição do produto, proporcionando flexibilidade na análise e visualização.

Esses modelos permitem que a gestão visualize tendências de venda, acompanhe o estoque em tempo real e ajuste estratégias de venda com base nos dados disponíveis.

### **3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: GERENCIANDO FINANÇAS**

Este projeto visa aplicar conceitos de finanças pessoais e empresariais, contribuindo para a extensão universitária e fornecendo uma base prática para a gestão financeira no contexto empresarial. A análise de vendas e controle de estoque da loja de botijões de gás oferece um cenário prático onde é possível aplicar esses conceitos e promover a utilidade social do aprendizado.

#### **3.5.1 GERENCIANDO FINANÇAS**

Tópico 1: Conceitos Fundamentais de Economia e Finanças

Os conceitos de oferta e demanda influenciam diretamente as vendas e os preços dos produtos. No caso da loja de botijões, entender o comportamento da demanda por diferentes tipos de botijão permite ajustar preços e manter o equilíbrio de estoque. Por exemplo, uma baixa demanda por botijões de 45kg pode resultar na redução de preço para estimular as vendas.

Tópico 2: Independência Financeira

No contexto da loja, a criação de um orçamento eficiente é crucial. O controle de receitas (vendas) e despesas (custo de aquisição, manutenção do estoque) ajuda a identificar oportunidades de poupança e reinvestimento, garantindo a sustentabilidade do negócio sem a necessidade de crédito externo.

Tópico 3: Dívidas e Juros Compostos

Entender o impacto de juros compostos em financiamentos pode ajudar a loja a tomar decisões mais informadas na hora de negociar crédito com fornecedores. Evitar dívidas com juros altos ou negociar prazos de pagamento mais favoráveis pode melhorar a saúde financeira da empresa a longo prazo. Aumentar as vendas de botijões de 5kg em 10% nos próximos três meses pode ser alcançado através de campanhas promocionais ou melhorias no atendimento ao cliente.

#### Tópico 4: Metas SMART

Definir metas específicas para o aumento de vendas ou redução de custos pode ser uma estratégia eficaz. Por exemplo, estabelecer uma meta de aumentar as vendas de botijões de 5kg em 10% nos próximos três meses pode ser alcançado através de campanhas promocionais ou melhorias no atendimento ao cliente.

### 3.5.2 ESTUDANTES NA PRÁTICA



## 4. CONCLUSÃO

Este projeto de controle de estoque de gás, desenvolvido com Python e estruturado em Programação Orientada a Objetos (POO), atendeu às necessidades da empresa Gonçalves & Gonçalves - Comércio e Distribuidora de Gás, Ltda, centralizando e organizando o gerenciamento de estoque de forma prática e eficiente. A implementação de classes como

Usuario, Gas, e Estoque permitiu modularidade e escalabilidade, proporcionando uma estrutura de fácil manutenção e expansão futura.

A integração com o banco de dados MySQL assegurou a persistência dos dados, garantindo a consistência e integridade das informações essenciais para o controle de inventário. Com o apoio do Power BI, o projeto incluiu ainda um dashboard visual, oferecendo uma análise detalhada do estoque e das vendas, essencial para decisões estratégicas. Esses recursos somados à lógica de programação aplicada reforçaram a praticidade e a eficiência da aplicação, permitindo à empresa um controle mais preciso do estoque e uma visão estratégica do negócio.

## REFERÊNCIAS

ALMEIDA, Marcos. **Python: utilizando POO na Engenharia de Dados** - Alura. Disponível em: < [Python: utilizando POO na Engenharia de Dados | Alura](#) >. Acesso em: 08/10/2024

HENRIQUE, João. **POO: O que é a programação orientada a objetos?**. Disponível em: < [POO: o que é programação orientada a objetos? | Alura](#) >. Acesso em: 08/10/2024

**O que é o MYSQL** - Oracle Brasil. Disponível em: < [www.oracle.com/br/mysql/what-is-mysql/](http://www.oracle.com/br/mysql/what-is-mysql/) >. Acesso em: 08/10/2024

**Como conectar ao banco de dados MySQL usando Python** - DebugEverything. Disponível em: < [Como conectar ao banco de dados MySQL usando Python](#) >. Acesso em: 11/10/2024

Glauco, **Microsoft Power BI: Primeiros passos** - DevMedia. Disponível em: < [Microsoft Power BI: primeiros passos](#) > Acesso em: 13/09/2024

**Lógica de Programação com Python**. Disponível em: < [EBOOK - INTRODUÇÃO A PYTHON \(EDITORA IFRN\).pdf](#) > Acesso em

**Cardinalidade e como Cognos Analytics usa** - IBM. Disponível em: < [www.ibm.com/docs/pt-br/cognos-analytics/11.2.0?topic=ambiguity-cardinality-how-cognos-analytics-uses-it](http://www.ibm.com/docs/pt-br/cognos-analytics/11.2.0?topic=ambiguity-cardinality-how-cognos-analytics-uses-it) > Acesso em: 10/10/2024

**SQL CREATE TABLE Statement** - W3 School. Disponível em < [https://www.w3schools.com/sql/sql\\_create\\_table.asp](https://www.w3schools.com/sql/sql_create_table.asp) > Acesso em 10/10/2024

**Reverse Engineering Using a Create Script** - MySQL Workbench Manual. Disponível em < <https://dev.mysql.com/doc/workbench/en/wb-reverse-engineer-create-script.html> > Acesso em 10/10/2024

# ANEXOS

3.2.1 Code : < <https://pastebin.com/taSWV68S> >