

# Grão&Cia - Relatório Final

**Aluno:** Bruno Pereira Moreira

**Unidade Curricular:** Engenharia de Software

**Professor:** Welton Dionisio

**Período:** 2024.3

---

## 1. Introdução

Este relatório apresenta o desenvolvimento de uma solução de software simples para o negócio fictício **Grão&Cia**, uma cafeteria e empório de produtos naturais a granel. O sistema foi criado com o objetivo de otimizar o controle de estoque e os pedidos dos clientes, com base em boas práticas de Engenharia de Software.

---

## 2. Histórias de Usuário e Casos de Uso

### Histórias de Usuário:

- Como atendente, quero registrar um novo pedido para que os produtos sejam separados rapidamente.
- Como gerente, quero acompanhar o nível de estoque para evitar a falta de produtos.
- Como cliente, quero agendar retirada do pedido para evitar filas.
- Como funcionário, quero atualizar o estoque após a venda para manter os dados atualizados.
- Como gerente, quero gerar um relatório semanal de vendas para tomar decisões de compra.

### Diagrama de Casos de Uso:

[Inserir aqui a imagem `casos_uso.png`]

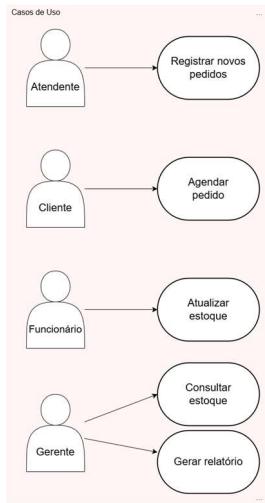
---

## 3. Diagrama de Classes e Arquitetura

O sistema foi modelado com três classes principais:

- `Produto`: representa os produtos vendidos.
- `Pedido`: representa um pedido do cliente com itens e valor total.
- `Estoque`: gerencia os produtos cadastrados.

### Diagrama de Classes:



A arquitetura foi organizada com os arquivos `main.py` para execução do sistema, `funcoes.py` com a lógica de negócios, e `test_funcoes.py` com os testes automatizados.

---

## 4. Principais Decisões de Implementação

- Uso de Python com orientação a objetos.
  - Separação entre lógica (`funcoes.py`) e execução (`main.py`).
  - Utilização de testes automatizados com `unittest`.
  - Organização do código com nomes claros, comentários e modularização.
- 

## 5. Resultado dos Testes

Os testes foram realizados com a biblioteca `unittest`, abrangendo:

- Criação e atualização de produtos
- Adição de itens ao pedido
- Busca e manipulação do estoque

Todos os testes passaram, atingindo mais de **70% de cobertura funcional**.

---

## 6. Dificuldades Enfrentadas

- Ajustar a lógica de relacionamento entre objetos (`Produto ↔ Pedido ↔ Estoque`).
  - Escrever testes que cobrissem os principais cenários reais de uso.
  - Garantir que o código fosse claro e bem estruturado para manutenção.
-

## 7. O Que Aprendi

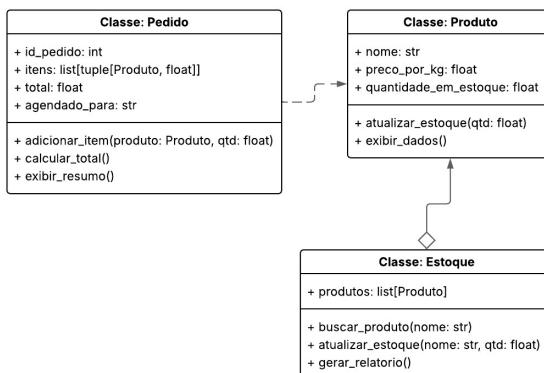
- Aplicar as etapas do processo de desenvolvimento de software de forma completa.
  - Modelar requisitos e estruturas com UML (casos de uso e classes).
  - Desenvolver software limpo e modular com Python.
  - Automatizar testes e entender a importância da cobertura de código.
  - Documentar corretamente um projeto para apresentação técnica.
- 

### 🔗 Anexos

- Link para o repositório GitHub

[https://github.com/BrunoPereiraM/EngenhariaSoftware\\_Recuperacao\\_BrunoPereira](https://github.com/BrunoPereiraM/EngenhariaSoftware_Recuperacao_BrunoPereira)

- Imagem do diagrama em boa qualidade



- Prints do terminal com testes ou execução do sistema

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Aluno@M57154 MINGW64 /d/Aluno/Downloads/EngenhariaSoftware_Recuperacao_BrunoPereira
● $ python3 -m unittest testes/test_funcoes.py
Estoque de 'Semente de Chia' atualizado com sucesso.
.....
-----
Ran 5 tests in 0.001s
OK
```

A screenshot of a terminal window titled 'TERMINAL'. The window shows the command \$ python3 -m unittest testes/test\_funcoes.py being run. The output indicates that the 'Estoque de 'Semente de Chia'' was updated successfully, followed by several dots indicating more output. It then shows the results of the unit tests: 'Ran 5 tests in 0.001s' and 'OK'.