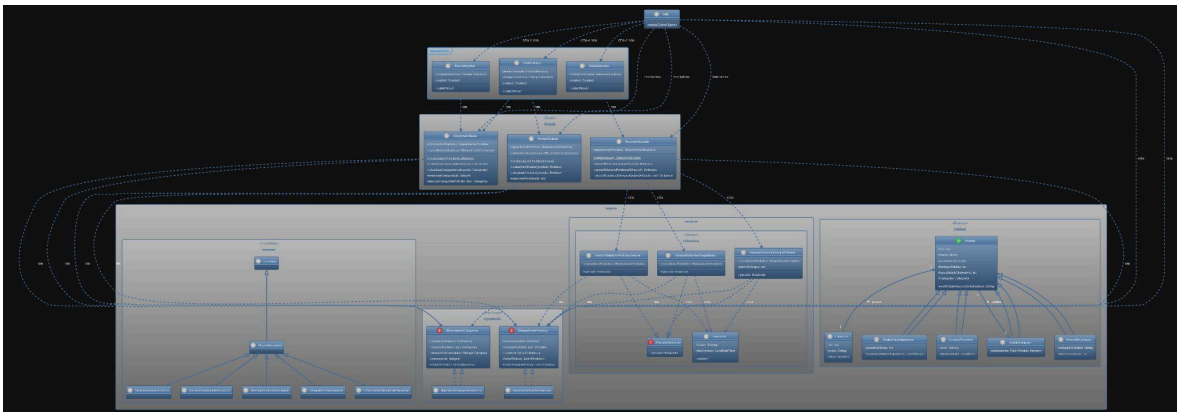
Universidade Federal do Agreste de Pernambuco
Curso: Bacharelado em Ciência da Computação.
Disciplina: Programação Orientada à Objetos.
Professora: Thaís Alves Burity Rocha.
Aluno: Thayson Guedes de Medeiros.
Semestre: 2025.1

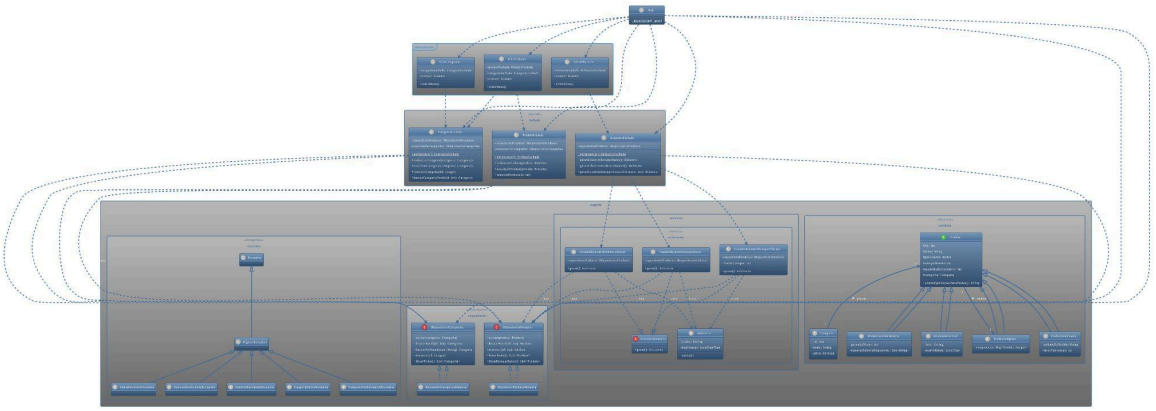
3ª Entrega do Projeto

1. Arquitetura do Sistema

1.1. Diagrama de Classes UML

- As imagens abaixo representam o diagrama de classes UML do sistema de Controle de Estoque, gerado a partir do código-fonte final. Ele ilustra as principais classes, seus atributos, métodos e os relacionamentos entre elas, além do fluxo do sistema como um todo.





1.2. Texto Explicativo da Arquitetura

O sistema foi projetado utilizando uma arquitetura em camadas para garantir a separação de responsabilidades, promover alta coesão e baixo acoplamento, e facilitar a manutenção do código. A estrutura é dividida da seguinte forma:

- Camada de Apresentação (**apresentação**): Responsável pela interação com o usuário via console. Classes como **TelaProdutos**, **TelaCategorias** e **TelaRelatorios** gerenciam a exibição de menus e a captura de dados de entrada. Esta camada não contém regras de negócio e apenas delega as chamadas para a camada de **Fachada**.

- Camada de Fachada (**fachada**): Implementa o padrão de projeto Facade, que serve como um ponto de entrada único e simplificado para as funcionalidades do sistema. As classes **ProdutoFachada, CategoriaFachada e RelatorioFachada** ocultam a complexidade das interações entre as classes de negócio e os repositórios, fornecendo uma interface coesa para a camada de apresentação.

- Camada de Negócio (**negocio**): É o núcleo do sistema e contém toda a lógica de negócio. Está subdividida em:
    - Entidades (**negocio.entidade**): Representam os objetos do domínio. A classe **Produto** é abstrata e serve como base para especializações como **ProdutoPerecivel, ProdutoComNumeroSerie, ProdutoFracionado e ProdutoComposto**, demonstrando o uso de Herança e Polimorfismo. No mesmo pacote de **negocio.entidade** também se encontram as classe de Produtos e Categorias em pacotes separados, como, (**negocio.entidade.produto e negocio.entidade.categoria**). Toda classe de negócio possui Javadoc com autor e descrição.

    - Repositórios (**negocio.repositorio**): Utilizada para abstrair a lógica de acesso a dados. As Interfaces **IRepositorioProdutos** e **IRepositorioCategorias** definem os contratos de persistência, enquanto as classes **RepositorioProdutosMemoria** e **RepositorioCategoriasMemoria** fornecem a implementação concreta (neste caso, em memória).

    - Serviços (**negocio.servicos**): Contém lógicas de negócio que não pertencem a uma única entidade, como a geração de relatórios. A interface **IGeradorRelatorio** e suas implementações (**GeradorRelatorioEstoqueBaixo**, etc.) utilizam polimorfismo para gerar diferentes tipos de relatórios.

    - Exceções (**negocio.excecoes**): O projeto define uma hierarquia de Exceções customizadas, como **DadosInvalidosException** e **CategoriaEmUsoException**, que herdam de **NegocioException**. Isso permite um tratamento de erros robusto e específico para as regras de negócio do sistema.

- Camada de Infraestrutura (infra): Fornece funcionalidades de suporte, como a classe **DadosPreDefinidos**, responsável por popular o sistema com dados iniciais para facilitar testes e demonstrações.

- Os relacionamentos entre as classes, como **Associação**, **Generalização/Herança**, **Realização** (implementação de interface) e **Dependência**, estão claramente definidos e podem ser observados no diagrama, seguindo as melhores práticas de Orientação a Objetos.

2. **Contribuições individuais**: Uma tabela que resume as estórias de usuário que foram implementadas e o responsável por cada uma delas, inclusive a especificação da estória.

Casos de Uso	Funcionalidade	Atribuição
UC001	**Gerenciar Produtos**	**Thayson Guedes**
UC003	**Gerenciar Categorias**	**Thayson Guedes**
UC016	**Relatórios de Estoque Baixo e Perto de Vencimento**	**Thayson Guedes**

Caso de uso	Estória de Usuário
UC003	Eu, como gerente, quero cadastrar, ler, atualizar e remover (CRUD) categorias.
UC001	Eu, como gerente, quero cadastrar diferentes tipos de produtos (padrão, perecível, com nº de série, etc.).
UC001	Eu, como gerente, quero poder listar todos os produtos e ver suas informações detalhadas.
UC001	Eu, como gerente, quero poder atualizar e remover produtos existentes.
UC016	Eu, como gerente, quero gerar um relatório de produtos com estoque baixo para saber o que comprar.
UC016	Eu, como gerente, quero gerar um relatório de produtos próximos ao vencimento para planejar promoções.
UC016	Eu, como gerente, quero gerar um relatório de produtos com estoque abaixo de um limite que eu definir.