

# Documentação Completa - Pipeline Pattern

## Introdução

Este documento apresenta a documentação completa do **Pipeline Pattern**, um padrão de projeto aplicado para modularização e sequenciamento de processos em software. A solução foi desenvolvida como parte da disciplina **Reuso de Software**, explorando conceitos de separação de responsabilidades, extensibilidade e reaproveitamento de código.

---

## O Problema

Em diversos sistemas complexos, há a necessidade de processar informações de forma sequencial, garantindo que cada passo ocorra na ordem correta. No entanto, a implementação direta dessas sequências pode resultar em **código acoplado, difícil de manter e pouco flexível** para futuras mudanças.





Problemas identificados:

- **Baixa reutilização** de componentes;
  - **Dificuldade de escalabilidade** quando novos estágios precisam ser adicionados;
  - **Código monolítico** que dificulta a modularização e os testes individuais;
  - **Falta de separação de responsabilidades** entre diferentes partes do processamento.
- 

## A Solução - Pipeline Pattern

O **Pipeline Pattern** resolve esses problemas ao estruturar o processamento em **estágios independentes**, onde cada estágio realiza uma transformação ou ação específica sobre os dados antes de passá-los ao próximo.

Benefícios desta abordagem:

-  **Modularidade:** Cada etapa do pipeline é encapsulada em uma classe separada;
  -  **Reutilização de Código:** Múltiplos pipelines podem compartilhar estágios comuns;
  -  **Extensibilidade:** Facilmente adaptável a novos requisitos sem impactar o código existente;
  -  **Facilidade de Testes:** Cada estágio pode ser testado de forma isolada.
-

## Estrutura do Projeto

A organização do código reflete a separação entre lógica de negócio, estágios do pipeline e exemplos de aplicação.

```
/pipeline_pattern
├── /examples          # Exemplos de uso do pipeline
│   ├── authentication_pipeline.py # Pipeline de Autenticação
│   ├── data_processing_pipeline.py # Pipeline de Processamento de Dados (ETL)
│   └── ecommerce_pipeline.py      # Pipeline de Checkout de E-commerce
├── /src              # Implementação do Pipeline Pattern
│   ├── pipeline.py    # Classe principal do pipeline
│   ├── /stages        # Estágios do pipeline
│   │   ├── __init__.py # Indica que esta pasta é um pacote
│   │   ├── auth_stage.py # Estágio de Autenticação
│   │   ├── processing_stage.py # Estágio de Processamento
│   │   ├── notification_stage.py # Estágio de Notificação
│   │   └── stage_base.py # Classe Base para os Estágios
├── /tests            # Testes automatizados
│   └── test_pipeline.py # Testes do pipeline
├── README.md         # Documentação do projeto
├── requirements.txt   # Dependências do projeto
└── LICENSE           # Licença de uso
```

---

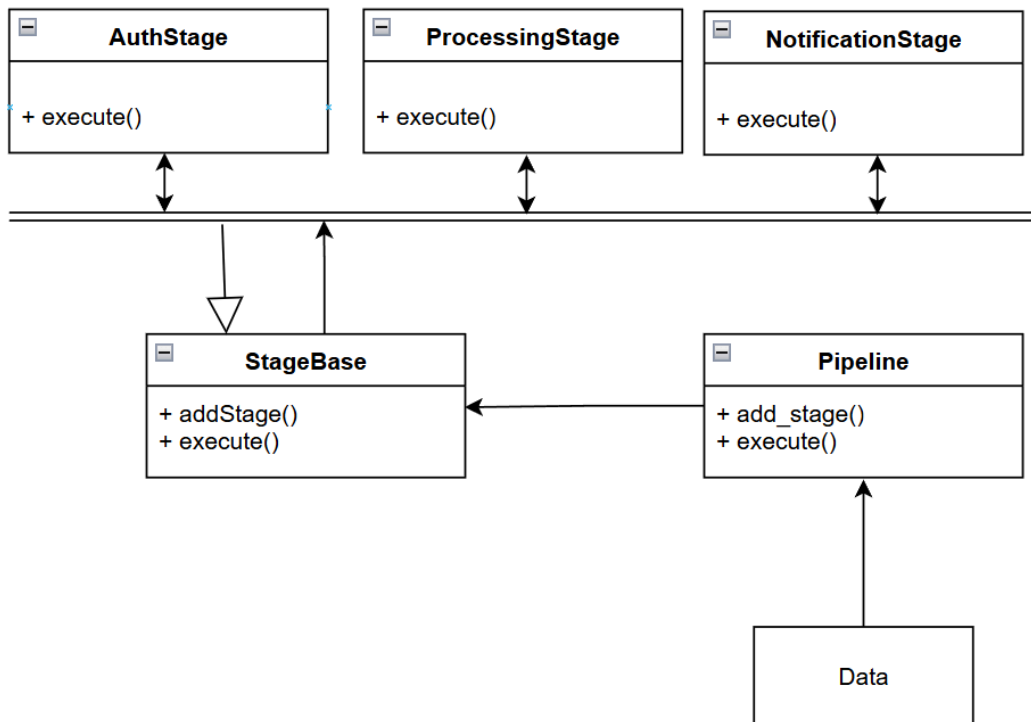
## Tecnologias Utilizadas

O projeto foi desenvolvido utilizando:

- **Python 3.13** - Linguagem de programação principal;
  - **pytest** - Framework de testes automatizados;
  - **Git/GitHub** - Controle de versão e colaboração;
  - **VS Code** - Ambiente de desenvolvimento principal.
-

## Diagrama de Classes

Abaixo, um **diagrama de classes** representando a estrutura do projeto e o fluxo entre os componentes do **Pipeline Pattern**:




O **Pipeline** gerencia a execução de múltiplos estágios. Todos os estágios herdam de **StageBase**, garantindo uma estrutura padronizada para a implementação dos processos.

---

## Contribuição da Equipe

A equipe de desenvolvimento trabalhou em conjunto para garantir a modularidade e funcionalidade do sistema. Abaixo está a descrição da contribuição de cada integrante:

- **André Silva Cavalcanti de Albuquerque:** Responsável pela **arquitetura do projeto**, criação do **repositório no GitHub**, implementação dos **testes automatizados** e integração final.
- **Antonio Elliton Dias Gomes:** Desenvolveu o **estágio de autenticação** e ajudou na **documentação técnica**.
- **Antônio Kaio Elias Portela:** Responsável pelo **estágio de processamento** e revisão do código.
- **Wilhelm de Sousa Steins:** Implementou o **estágio de notificação** e contribuiu com os exemplos de uso.
- **Mateus Gonçalves Loiola:** Atuou na organização da documentação e revisão do código.

 **Observação:** O repositório no GitHub foi mantido por **André Silva Cavalcanti de Albuquerque** para garantir **padronização e qualidade no versionamento do código**. Todos os integrantes participaram ativamente da implementação e testes, mas manter um único responsável pelo repositório evitou conflitos e facilitou a gestão do código-fonte.

---

## Licença

Este projeto está licenciado sob a **MIT License** – veja o arquivo [LICENSE](#) para mais detalhes.

---

## Dúvidas ou Sugestões?

Caso tenha dúvidas, entre em contato com um dos integrantes listados na documentação ou abra uma **issue** no repositório oficial:

 [GitHub Repository](#)