

# Relatório de Comparação entre Versões do Gerador de Relatórios em Java

Enzo Zanetti Camargo Penteado 15574558, Denilson Santos de Jesus 15648230

## 1. Introdução

Este relatório apresenta uma análise comparativa entre as duas versões do programa **GeradorDeRelatorios** em Java: a versão original com implementação monolítica e a versão refatorada, que aplica princípios de design orientado a objetos e padrões de projeto como *Strategy* e *Decorator*.

## 2. Diferenças Estruturais e de Projeto

### 2.1. Design e Extensibilidade

A versão original concentra toda a lógica na classe **GeradorDeRelatorios**, tornando o código rígido e difícil de estender. A refatoração introduz modularidade com a criação de interfaces para critérios de ordenação (**CriterioOrdenacao**), algoritmos de ordenação (**AlgoritmoOrdenacao**) e critérios de filtragem (**CriterioFiltro**).

Essa separação favorece a extensibilidade, pois novos comportamentos podem ser adicionados sem alterar o código existente, respeitando o princípio do aberto-fechado.

### 2.2. Padrão Strategy

A lógica de ordenação e filtragem foi extraída da classe principal e encapsulada em estratégias independentes. Por exemplo:

- Algoritmos de ordenação: **InsertionSort** e **QuickSort**
- Critérios de ordenação: **CriterioDescricao**, **CriterioPreco**, **CriterioEstoque**
- Filtros: **FiltroTodos**, **FiltroPorCategoria**, **FiltroPorEstoqueMenorIgual**, **FiltroPorIntervaloDePreco**, **FiltroPorDescricaoContendo**.

### 2.3. Padrão Decorator

Na versão original, o estilo (negrito, itálico) é aplicado globalmente a todos os produtos usando flags binárias. Já na versão refatorada, cada produto pode ter um estilo individual aplicado por meio de decoradores como **ProdutoNegrito**, **ProdutoItalico** e **ProdutoColorido**, sem alterar a lógica de geração do relatório.

### 3. Leitura de Dados

A versão original utiliza uma lista fixa de produtos codificada no próprio código-fonte. A versão refatorada lê os dados de um arquivo CSV, permitindo entrada de dados dinâmica e adaptável a diferentes contextos.

### 4. Critérios de Ordenação e Filtragem

A versão original permite apenas ordenação crescente por três critérios fixos: descrição, preço e estoque. A versão refatorada permite ordenação tanto crescente quanto decrescente, além de oferecer novos critérios de filtragem como:

- `preco_intervalo`
- `descricao_contem`

### 5. Geração do Relatório HTML

Ambas as versões geram um relatório HTML. No entanto, a versão original mistura a lógica de formatação e geração do HTML. Na refatorada, a responsabilidade pela formatação é delegada ao próprio objeto `Produto`, que se autoformata via `formataParaImpressao()`.

### 6. Uso de Coleções e Streams

A versão original usa arrays (`Produto[]`); a nova versão usa `List<Produto>` com `Streams`, permitindo filtragem e manipulação mais modernas, eficientes e concisas.

### 7. Interface de Linha de Comando

A versão original possui uma interface limitada para argumentos. A versão refatorada permite mais flexibilidade, inclusive passando o caminho para um CSV, além de suportar novos critérios e parâmetros.

### 8. Conclusão

A refatoração resulta em um sistema muito mais modular, extensível e aderente aos princípios da programação orientada a objetos. As mudanças promovem clareza, reaproveitamento de código, facilidade de manutenção e escalabilidade. O uso dos padrões *Strategy* e *Decorator*, além do suporte à entrada de dados via CSV e novos critérios de ordenação e filtragem, tornam o sistema mais robusto e pronto para uso em cenários reais.