

UNIVERSIDADE DE SÃO PAULO  
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES  
COMPUTAÇÃO ORIENTADA A OBJETOS -  
ACH2003

GABRIEL VENTORIM SILVA - 13672147  
GUSTAVO VENTORIM SILVA - 13672342  
LAILA MALAFAIA VIEIRA - 13672255

**RELATÓRIO: Refatoração de código**

## 1. ALTERAÇÕES

As alterações no código entregue foram realizadas pautando-se no padrão Strategy e Decorator, essas mudanças tem como objetivo facilitar possíveis mudanças no projeto e deixá-lo mais adaptável.

### 1.1 Interfaces criadas

Para a refatoração do programa que gera relatórios, foram utilizadas várias interfaces que são implementadas de acordo com as escolhas, *inputs* (entradas) , feitas para a manipulação dos produtos.

#### 1.1.1 Algoritmo

A interface Algoritmo é implementada por duas classes, *InsertionSort* e *QuickSort*, cujos modos de ordenação são, respectivamente, Insertion Sort e Quick Sort.

#### 1.1.2 Criterio

A interface Criterio foi adicionada para poder criar um tipo mais geral que englobe os diferentes critérios de ordenação dos produtos da tabela dada. Ela é implementada por seis classes criadas: *CriterioDescCresc*, *CriterioDescDecres*, *CriterioEstoqueCresc*, *CriterioEstoqueDecres*, *CriterioPrecoCresc* e *CriterioPrecoDecres*.

#### 1.1.3 Filtro

A interface Filtro foi adicionada para poder criar um tipo mais geral que englobe os diferentes critérios de filtragem dos produtos da tabela. Ela é implementada por cinco classes criadas: *FiltroCatIguar*, *FiltroIntervalo*, *FiltroEstMenorIguar*, *FiltroSubString* e *FiltroTodos*.

#### 1.1.4 Formato

A interface Formato é usada para diferentes formas de texto. Ela é implementada por uma classe abstrata usada como Decorator, a *FormatoDecorator*, sendo que esta é a superclasse de quatro subclasses: *FormatoCor*, *FormatoItalico*, *FormatoNegrito* e *FormatoPadrao*.

### 1.2 Classes criadas

Além das classes desenvolvidas para a implementação das interfaces criadas, houve também a adição da classe *ProdutoComFormatacao* que implementa *Produto*.

### **1.2.1 ProdutoComFormatacao**

Esta classe tem como objetivo criar objetos do tipo Produto instanciados pela classe ProdutoComFormatacao, em que diferentemente da classe ProdutoPadrao, coloca formatações na escrita dos produtos, como: cor, negrito e itálico.

## **1.3 Adaptações na classe GeradorDeRelatorios**

Em relação à classe GeradorDeRelatorios, foi-se feito alterações multifocais, seja devido às novas classes e interfaces criadas, ou pela melhor eficiência do código.

### **1.3.1 Mudança das variáveis de instância**

O construtor foi adaptado para armazenar as variáveis de acordo com as seguintes mudanças:

- O vetor Produto[] produtos foi alterado para uma lista de produtos, usando: List<Produto> produtos
- A String algoritmo foi alterada para o tipo de interface Algoritmo algoritmo, e instanciada por uma das classes que a implementam.
- A String criterio foi alterada para o tipo de interface Criterio criterio, e instanciada por uma das classes que a implementam.
- A String filtro foi alterada para o tipo da interface Filtro filtro, e instanciada por uma das classes que a implementam.
- A String argFiltro foi alterada para uma List<String> argFiltro
- Retirada de format\_flags

### **1.3.2 Mudanças em métodos**

- Método particiona() : retirado da classe. Este é responsabilidade da classe QuickSort e não da classe GeradorDeRelatorios.
- Método ordena() : modificado para realizar o método ordena(), da classe Algoritmo, implementado pelas classes que implementam Algoritmo, aplicando o padrão Strategy.
- Método geraRelatorio() : alterado para aceitar Decorator após o loop “for”, decorando a variável “formato” com as devidas formatações.
- Método carregaProdutos() : removido. Não está sendo utilizado no projeto.
- Método lerCSV() : criado para substituir carregaProdutos(). Agora a leitura está sendo feita diretamente do arquivo “produtos.csv” dado.
- Método main() : atualização nas impressões (System.out.println) das utilizações do código. Foram adicionadas novas funcionalidades do código, alterações de “opcao\_algoritmo”, “opcao\_criterio\_ord” e “opcao\_criterio\_filtro”, com mudanças no tipo de String para Algoritmo, Criterio e Filtro, respectivamente, além da alteração do nome da variável. Além dessas alterações, a “opcao\_parametro\_filtro” passou a

ser List<String> e houve a retirada do "opcao\_formatacao". Por fim, também foram adicionadas as instâncias de cada um dos tipos citados. Também no método main, foi-se alterado o número de parâmetros de acordo com o filtro, **intervalo** recebe dois, **todos** nenhum e **os demais** um único parâmetro.