

Padrões de Projeto

Denis W. Oliveira – CIU 74384

Introdução

Durante toda a criação e documentação dos projetos de software, sempre teve-se o costume de seguir um padrão próprio da equipe, ou em alguns casos, padrão nenhum, trazendo uma quantidade enorme de problemas para quem fosse dar manutenção ao sistema.

Entre esses problemas, temos o problema de legibilidade do código, classes com responsabilidades diferentes das planejadas, alto acoplamento e baixa coesão, bad smells e etc., que impediam uma boa manutenção e dificultava a entrega do sistema ou crescimento da aplicação conforme a demanda do mercado fosse aumentando.

Pensando na resolução desse problema, foram criados os Padrões de Projeto, que são métodos de se planejar o sistema a ser criado de forma que desacople suas funcionalidades e o torne modular e independente, facilitando o reaproveitamento de código, a criação de módulos diferentes e independentes entre si e facilitando a manutenção e o crescimento da aplicação de acordo com a demanda.

Implementação

Para implementação desse trabalho, utilizei primeiramente como padrão primário, a estrutura MVC, separando o *model*, *controller* e *view* da aplicação.

Segue abaixo, a árvore explicativa dos pacotes e classes implementados até o momento:

- **Controller:**
 - **Main:** Classe principal da aplicação, responsável por chamar as outras classes que serão utilizadas.
- **View:**
 - **Menu:** Classe responsável pela interação com o usuário
 - **Pacote Pagina:**
 - **Cabecalho :** Classe responsável por alocar as tags escolhidas pelo usuário no cabeçalho da página. Implementa o conceito de Singleton, sendo criada apenas uma vez.
 - **Corpo:** Classe responsável por alocar as tags escolhidas pelo usuário no corpo da página. Implementa o conceito de Singleton, sendo criada apenas uma vez.
 - **Rodapé:** Classe responsável por alocar as tags escolhidas pelo usuário no rodapé da página. Implementa o conceito de Singleton, sendo criada apenas uma vez.
 - **Página:** Classe responsável por alocar os conteúdos das classes Cabecalho, Corpo e Rodapé, e também alocar as tags externas, mas que fazem parte da página. Implementa o conceito de Singleton, sendo criada apenas uma vez.
- **Model:**
 - **Fabrica:**

- **Fabrica Abstrata:** Classe responsável pelo gerenciamento de todas as fábricas utilizadas no projeto . Utilizando o conceito de Abstract Factory, ela faz a intermediação entre as solicitações do usuário e a criação de elementos HTML de forma organizada, construindo apenas as fábricas que criam esses respectivos elementos.
- **Fabrica Estrutura:** Classe responsável pelo gerenciamento das tags de estrutura do HTML, como *html*, *head*, *body* e *div*. Utilizando o conceito de Factory Method, essa classe intermedia a criação dos elementos de estrutura e retorna ao usuário apenas o que ele solicitou, sem ocorrer a manipulação direta dos elementos em qualquer outra classe.
- **Fabrica Imagens:** Classe responsável pela criação dos elementos de imagem, como a tag *img* e seus parâmetros utilizados. Utiliza o conceito de Factory Method, intermediando a solicitação do usuário e criação efetiva do elemento, impedindo que o mesmo seja instanciado em outros lugares.
- **Fabrica Listas:** Classe responsável pelo gerenciamento das tags de lista do HTML, como *ul*, *ol*, e *li*. Utilizando o conceito de Factory Method, essa classe intermedia a criação dos elementos de lista e retorna ao usuário apenas o que ele solicitou, sem ocorrer a manipulação direta dos elementos em qualquer outra classe.
- **Fabrica Tabelas:** Classe responsável pelo gerenciamento das tags de tabela do HTML, como *table*, *tr*, *td* e *th*. Utiliza o conceito de Factory Method, intermediando a criação de tabelas de acordo com a solicitação do usuário, impedindo que os elementos sejam criados em outras classes.
- **Fabrica Tag Texto:** Classe responsável pelo gerenciamento das tags de texto do HTML, como *h1*, *h2*, *a*, *p*, etc.. Utiliza o conceito de Factory Method, intermediando a criação das tags de acordo com a solicitação do usuário e ao mesmo tempo impedindo o instanciamento dessas classes em outras classes fora da fábrica.
- **Tags:**
 - **Elemento Padrão:**
 - **Elemento:** Classe abstrata que define a estrutura básica de um elemento HTML. É a classe “pai” de todas as classes de tags HTML utilizadas no projeto.
 - **Tag Estrutura:**
 - **Html:** Classe responsável pela tag *html*.
 - **Head:** Classe responsável pela tag *head*.
 - **Body:** Classe responsável pela tag *body*.
 - **Div:** Classe responsável pela tag *div*.
 - **Tag Imagem:**
 - **Img:** Classe responsável pela tag *img*.
 - **Tag Lista:**
 - **Lista:** Classe responsável pela tag *ul*.
 - **ItemLista:** Classe responsável pela tag *li*.
 - **Tag Tabela:**
 - **Tabela:** Classe responsável pela tag *table*.
 - **TabelaLinha:** Classe responsável pela tag *tr*.

- **TabelaCabeçalho:** Classe responsável pela tag *th*.
- **TabelaDetalhe:** Classe responsável pela tag *td*
- **Tag Texto:**
 - **P:** Classe responsável pela tag *p*.
 - **H1:** Classe responsável pela tag *h1*.