UNIVERSIDADE DE BRASÍLIA - UnB Gama
Relatório de Orientação a objeto

RELATÓRIO MVC JAVA

Por:
Lucas Alves Vilela - 211062141
Ana Carolina Costa César - 190101750
Pedro Sampaio Dias - 211043745
Paulo Renato Soares Paes - 211029512

Brasília
Brasília-DF, 21° de setembro 2022

Padrão MVC

Introdução:

O padrão MVC (Model-View-Controller) é um padrão de arquitetura de software muito usado e conhecido. Ele foi documentado pela primeira vez em 1978 e é utilizado para separar as camadas de responsabilidade de uma aplicação, desacoplando a interface do usuário (view), os dados (Model) e lógica e comunicação entre as duas camadas anteriores (Controller).

Objetivo:

Este relatório pretende apresentar o padrão MVC, detalhar o papel de cada uma de suas camadas e, em seguida, serão apresentados um exemplo em UML e um link para seu código implementado em JAVA.

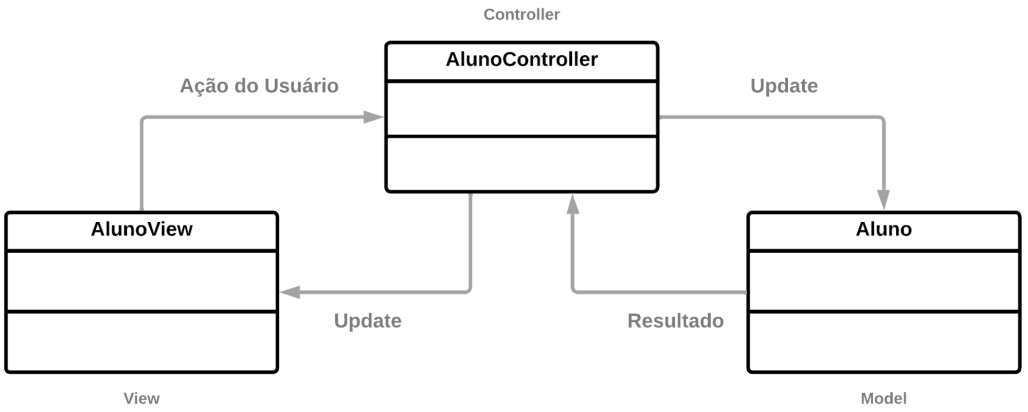

Imagem 1: Diagrama de funcionamento do MVC

O padrão MVC:

Neste padrão, o pedido do usuário chega ao Controller, o qual é responsável por interagir com a Model para obter os dados exigidos. Assim que isso for feito, o controller escolhe qual View deve ser mostrada para o usuário e compõe a tela com os dados retirados da Model.

Ao mudar uma camada, não é necessário que quem estiver programando tenha que refazer toda uma outra camada, como geralmente acontece em códigos apelidados de “espaguete”. Portanto o uso desse padrão diminui o retrabalho e a probabilidade de erro. Como o Model não depende das outras camadas, ele pode ser construído e testado sem precisar da representação visual.

Model:

Além de conter os dados e estados da aplicação, o Model contém todas as operações, lógica de negócio e implementações das lógicas dos estados.

View:

É a responsável por apresentar o conteúdo na interface. A View deve sempre refletir o estado do Model. Deve tratar apenas de recursos ligados à interface (panels, buttons, etc.), também deve ter o mínimo de lógica entre views.

Controller:

Controller, como o nome diz, controla a interação entre a interface com o usuário e a Model. É ele que lida e responde às requisições do usuário. Caso o Controller ficar muito complexo, é recomendado passe a lógica de negócio para a Model.

Vantagens do padrão MVC

- A desacoplação entre camadas criada pelo MVC permite a divisão de tarefas entre membros da equipe de forma com que cada um possa trabalhar com aquilo que estiver mais confortável. Vê se principalmente a divisão entre front-end(Views) e back-end(Model e Controller);
- Uma aplicação desenvolvida com MVC fica mais organizada, o que aumenta o fluxo de trabalho, facilita a identificação de erros e bugs, facilita o reaproveitamento de código para outros projetos e torna-a mais escalável;
- É possível trabalhar em cada camada sem afetar diretamente as outras, o que permite desenvolvimento paralelo e a aplicação de diversas metodologias ágeis;
- O Controller impede que dados incorretos sejam passados diretamente para o Model, preservando sua integridade.

UML de exemplo:

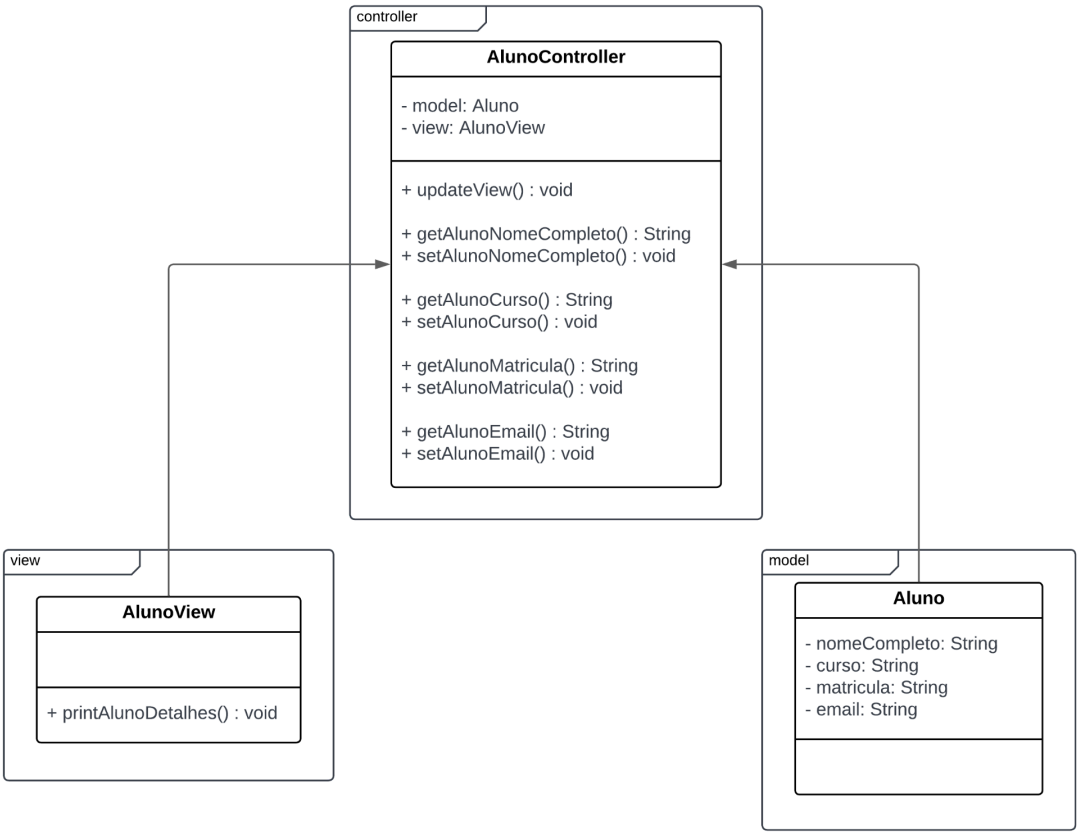

Imagem 2: Diagrama de Classes implementado pelo Grupo. Omitindo os getters e setters na classe “Aluno” e a relação de dependência entre View e Model.

Link para o repositório no github com o código de exemplo implementado:

https://github.com/Lucas-AV/TP1-OO

Referências:

Documentação Microsoft, acessado em 20 de setembro de 2022. Disponível em : https://learn.microsoft.com/pt-br/aspnet/core/mvc/overview?WT.mc_id=dotnet-35129-website&view=aspnetcore-6.0

le wagon, acessado em 20 de setembro de 2022. Disponível em : https://www.lewagon.com/pt-BR/blog/o-que-e-padrao-mvc

Macoratti.net, acessado em 20 de setembro de 2022. Disponível em : https://www.macoratti.net/14/05/net_mvc.htm

DEVMEDIA, acessado em 20 de setembro de 2022. Disponível em : https://www.devmedia.com.br/padrao-mvc-java-magazine/21995#6