

RELATÓRIO

“TEMA: Modelo MVC”

Grupo8:

19/0133813 – Italo Bruno Avelino da Silva Melo
21/1029620 - Douglas Alves dos Santos
/ -

Brasília, 19 de Setembro de 2022

Orientação por Objetos

INTRODUÇÃO

O MVC é um modelo de arquitetura de software que faz a separação dos projetos em 3 etapas, **view**, **model** e **controller**. Sendo elas, a interação com o usuário, a manipulação e comunicação dos bancos de dados e o controle do sistema.

DESENVOLVIMENTO

O MVC ou Model-View-Controller formulado na década de 70 e criado na década de 80 para que o desenvolvimento e manutenção seja muito mais simples e também para que seja possível um reaproveitamento de classes e partes de um projeto em outros. Criando então três camadas muito bem definidas de projetos independentes, o *view*, *model* e o *controller*.

A utilização do MVC tem de benefício a possibilidade da existência de várias interfaces com o usuário que podem ser alteradas de maneiras sem alterar outras partes, deixando as classes muito mais flexíveis e dando mais oportunidades de reuso das classes.

- **View**

A *view* é a camada visual de interação com o usuário. É pela interface que haverá as entradas de dados inseridos pelo usuário e a exibição das saídas do sistema que serão mostradas ao usuário, sendo por formulários de entrada, tabelas, grids ou outras formas. A *view* não tem conexão com o *model*, as respostas do sistema são passadas pelo *controller*.

- **Model**

O *model*, é a camada que possui toda a lógica da aplicação. Ele é o responsável por toda a interação e manipulação com os dados armazenados. É no *model* que será realizado as operações de CRUD, é a camada responsável pelo motivo que a aplicação foi construída sendo o centro do projeto.

- **Controller**

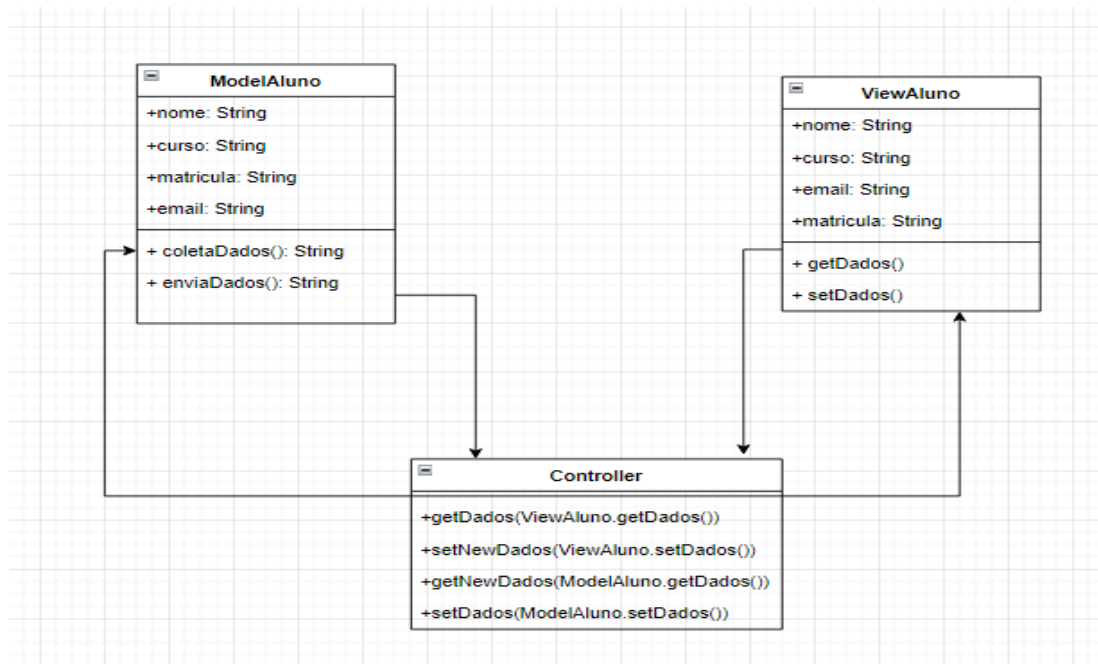
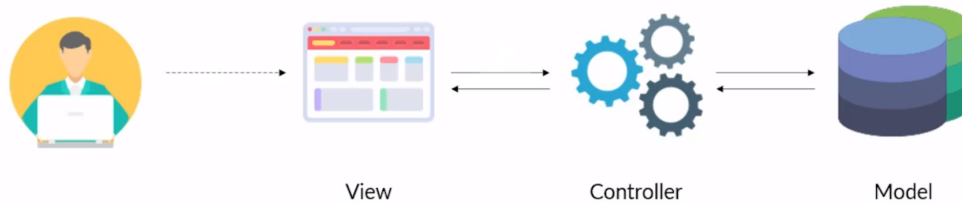
O *Controller* recebe as requisições da *view* e faz a intermediação com o *model* deste modo toda requisição do usuário passa pelo *controller* para assim então enviar ao *model* realizar as operações, após isso se receber uma resposta do *model* transmite ao *view*.

Devido ao *Controller* ter acesso às duas classes, e ser o responsável para a intermediação das informações, ele precisa pegar os gets e os sets tanto do *model* quanto do *view*, fazendo da seguinte maneira, pega os gets da *view*, envia para o *model*, então pega os gets do *model* e envia para os sets da *view*, e assim apresentar para o usuário.

Orientação por Objetos

Diagrama de interação do MVC

MVC Model-View-Controller



REFERENCIAL TEÓRICO

PADRÃO MVC Java Magazine. **Devmedia**. Disponível em: <<https://www.devmedia.com.br/padrao-mvc-java-magazine/21995>>. Acesso em: 19 de setembro de 2022.

BANAS, Derek. MVC Java Tutorial. YouTube, 21 de fevereiro de 2013. Disponível em: <<https://www.youtube.com/watch?v=dTVVa2gfht8>>. Acesso em: 19 de setembro de 2022.