

MODEL VIEW CONTROLLER (MVC)

O que é MVC?

O MVC é uma sigla do termo em inglês Model (modelo) View (visão) e Controller (Controle) que facilita a troca de informações entre a interface do usuário aos dados no banco, fazendo com que as respostas sejam mais rápidas e dinâmicas.

- O MVC é um padrão de arquitetura de software.
- O MVC sugere uma maneira para você pensar na divisão de responsabilidades.

O princípio básico do MVC é a divisão da aplicação em três camadas:

- camada de interação do usuário (view).
- camada de manipulação dos dados (model).
- camada de controle (controller).

Com o MVC, é possível separar o código relativo à interface do usuário das regras de negócio, o que sem dúvida traz muitas vantagens.

Quais os papéis de cada camada?

Quando falamos sobre o MVC, cada uma das camadas apresenta geralmente as seguintes responsabilidades:

- **Model (Modelo)**

Essa classe também é conhecida como Business Object Model (objeto modelo de negócio). Sua responsabilidade é gerenciar e controlar a forma como os dados se comportam por meio das funções, lógica e regras de negócios estabelecidas.

Ele é o detentor dos dados que recebe as informações do Controller, válida se ela está correta ou não e envia a resposta mais adequada.

- **View (Visão)**

Essa camada é responsável por apresentar as informações de forma visual ao usuário. Em seu desenvolvimento devem ser aplicados apenas recursos ligados a aparência como mensagens, botões ou telas.

- **Controller (Controlador)**

A camada de controle é responsável por intermediar as requisições enviadas pelo View com as respostas fornecidas pelo Model, processando os dados que o usuário informou e repassando para outras camadas.

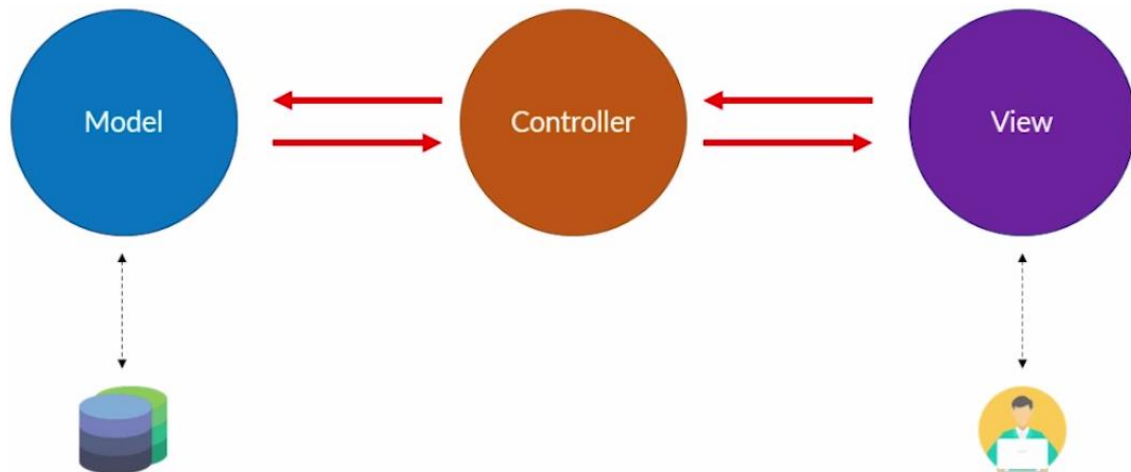
Como os componentes interagem?

Tudo começa com a interação do usuário na camada View. A partir daí o controlador pega essas informações e envia para o Model que fica responsável por avaliar aqueles dados e transmitir uma resposta.

O controlador recebe essas respostas e envia uma notificação de validação daquela informação para a camada visão, fazendo com a mesma apresente o resultado de maneira gráfica e visual.

Todo esse processo leva em consideração as regras de negócio aplicadas na construção de todo projeto.

Diagrama



Por que usar MVC?

- **Segurança:** O controller funciona como uma espécie de filtro capaz de impedir que qualquer dado incorreto chegue até a camada modelo.
- **Organização:** Esse método de programação permite que um novo desenvolvedor tenha muito mais facilidade em entender o que foi construído, assim como os erros se tornam mais fácil de serem encontrados e corrigidos.
- **Eficiência:** Como a arquitetura de software é dividida em 3 componentes, sua aplicação fica muito mais leve, permitindo que vários desenvolvedores trabalhem no projeto de forma independente.
- **Tempo:** Com a dinâmica facilitada pela colaboração entre os profissionais de desenvolvimento, o projeto pode ser concluído com muito mais rapidez, tornando o projeto escalável.
- **Transformação:** As mudanças que forem necessárias também são mais fluidas, já que não será essencial trabalhar nas regras de negócio e correção de bugs.
- **Testes:** Uma vantagem é que ele nos ajuda a deixar o código mais fácil de fazer manutenção, já que temos as responsabilidades devidamente separadas. Isso também traz uma facilidade na compreensão do código, além da sua reutilização. Além disso, você conseguirá realizar testes de uma maneira muito mais rápida e eficiente.

Além disso, um software precisa ter estabilidade no processo de comunicação entre seus elementos de maneira dinâmica para que a experiência do usuário não seja prejudicada.

Implementação do MVC

Além da possibilidade de implementação manual, existem diversos frameworks que implementam o padrão MVC e são muito utilizados em

diversos projetos. Existem diversos artigos e sites especializados que comparam as diferenças e vantagens entre esses diferentes frameworks. No entanto, o melhor é sempre verificar o que cada framework disponibiliza para os desenvolvedores e analisar se atende às expectativas.

Considerações finais

O MVC funciona como um padrão de arquitetura de software que melhora a conexão entre as camadas de dados, lógica de negócio e interação com usuário. Através da sua divisão em três componentes, o processo de programação se torna algo mais simples e dinâmico.

Por padrão existem a camada Model, Controller e View que deram origem a sigla dessa arquitetura de software mais utilizado entre os desenvolvedores.

Algumas empresas podem até cobrar o conhecimento de determinados frameworks para sua aplicação no dia-a-dia, por isso é interessante que o candidato participe de bootcamps que explorem o assunto de maneira adequada, a fim de melhorar a curva de aprendizado de quem quer explorar a área.

Referências

<https://www.treinaweb.com.br/blog/o-que-e-mvc>

<https://www.lewagon.com/pt-BR/blog/o-que-e-padrao-mvc>

<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>