

Discente: Luiz Gustavo Alves Alencar.

Disciplina: Padrões de projeto, 8º período de Sistemas de informação.

Atividade avaliativa

1. O que é MVC e qual seu objetivo?

O MVC é um padrão de arquitetura de software. O objetivo do MVC é a divisão da aplicação em três camadas: a camada de interação do usuário (view), a camada de manipulação dos dados (model) e a camada de controle (controller).

2. Explique as funções de Model, View e Controller.

O Model é responsável pelo acesso e manipulação dos dados na sua aplicação, recebe as requisições do Controller e gerando as respostas.

A View é responsável pela interface que será apresentada, front-end, mostrando as informações do model para o usuário.

O Controller é a camada de controle, responsável por ligar o model e a view, fazendo com que os models possam ser repassados para as views e vice-versa.

3. Cite exemplos de frameworks que utilizam MVC.

Laravel: um framework PHP.

Django: um framework Python.

Spring MVC: um framework Java.

4. Como o MVC facilita a manutenção do sistema?

Ele facilita a manutenção porque separa as responsabilidades da aplicação em camadas distintas fazendo com que os desenvolvedores possam trabalhar em diferentes partes da aplicação simultaneamente, sem causar conflitos.

5. Dê exemplos de projetos onde o MVC pode ser usado.

Projetos Web, desktop e mobile.

6. Quais os desafios de implementar MVC?

Curva de aprendizado: para desenvolvedores iniciantes ou equipes sem familiaridade com o padrão, entender e implementar a separação entre Model, View e Controller pode ser uma tarefa complexa.

Divisão de responsabilidades: identificar claramente o que pertence a cada camada pode ser difícil.

7. O que ocorre se não houver separação de camadas?

A ausência da separação de camadas em uma aplicação pode levar a diversos problemas e comprometendo a manutenção, escalabilidade e qualidade do software. Entre os problemas envolvidos destaca-se: Acoplamento excessivo, dificuldade de manutenção e dificuldade de teste e evolução.

8. MVC pode ser usado para APIs? Justifique.

Sim, para manter a boa organização do código e por se tratar de API se encontra mais as camadas do Model e do Controller.

9. Explique a comunicação entre Model, View e Controller.

View é a camada de interação com o usuário, responsável por exibir os dados fornecidos pelo Controller.

O Controller é responsável por gerenciar a lógica de fluxos entre o usuário e o sistema.

Model é a camada que lida com os dados e a lógica do sistema. O Model recebe as requisições do Controller e gera respostas. Ele é o intermediário entre View e Controller.

10. Como o MVC se aplica em aplicações web modernas?

O padrão MVC se aplica para se obter a organização lógica de interação do usuário, melhorar a conexão entre as camadas de dados e a lógica de negócio.