

Disciplina: Padrões de Projeto

Aluna: Láyza Ferreira Lopes

8º Período

### **Atividade sobre MVC**

#### **1. O que é MVC e qual seu objetivo?**

O MVC (Modelo, Vista e Controlador) é uma forma de organizar o desenvolvimento de software. Usá-lo deixa o código mais limpo e fácil de ajustar, porque cada parte tem seu papel bem definido. Desse jeito, as mudanças ou melhorias podem ser feitas sem bagunçar o restante da aplicação. É uma forma prática e eficiente de criar software.

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

- **Model:** Onde a lógica de negócios e os dados da aplicação são gerenciados.
- **View:** O que o usuário vê, a interface que apresenta os dados.
- **Controller:** A ponte entre o Modelo e a Vista, que lida com as entradas do usuário e atualiza a interface e os dados conforme necessário.

#### **3. Cite exemplos de frameworks que utilizam MVC.**

Ruby on Rails (Ruby), Django (Python) e Laravel (PHP)

#### **4. Como o MVC facilita a manutenção do sistema?**

O padrão MVC realmente facilita a manutenção de sistemas, tornando tudo mais claro e eficiente. Cada parte da aplicação do MVC tem um papel bem definido. Isso ajuda a evitar confusões e permite que as equipes de desenvolvimento colaborem de forma mais integrada, sem se atrapalharem mutuamente.

Outra grande vantagem do MVC é que o código se torna mais limpo e livre de repetições desnecessárias. Isso não só reduz a chance de erros, mas também torna o código mais fácil de entender e trabalhar. Quando surge a necessidade de testar ou corrigir problemas, o MVC simplifica esse processo, já que cada componente é independente.

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

Redes Sociais, Aplicativos, Plataformas de E-commerce, etc

## **6. Quais os desafios de implementar MVC?**

Implementar medidas de segurança adequadas pode ser um desafio, exigindo práticas de codificação segura e a utilização de recursos de segurança fornecidos pelo framework.

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

A falta de separação de camadas pode resultar em um código confuso e difícil de entender. Quando a lógica de negócios, a interface do usuário e a lógica de controle estão misturadas, é mais difícil localizar e corrigir erros, além de ser mais complicado adicionar novas funcionalidades.

## **8. MVC pode ser usado para APIs? Justifique.**

Sim, o padrão MVC é uma abordagem eficaz para o desenvolvimento de APIs, pois promove uma estrutura clara, facilita a manutenção e testes, e permite uma melhor organização e modularidade do código.

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

A comunicação entre Model, View e Controller no padrão MVC é o que mantém tudo organizado e funcionando bem. O Model cuida dos dados e das regras de negócio, sem falar diretamente com a View. Ele passa informações para o Controller quando algo muda. A View é responsável por mostrar os dados para o usuário e enviar as ações dele para o Controller. Já o Controller é o intermediário que processa as ações do usuário, atualiza o Model quando necessário e avisa a View para atualizar a interface.

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

O padrão MVC continua sendo usado em aplicações web modernas, mas adaptado às novas tecnologias. No backend, frameworks como Laravel organizam os dados e regras de negócio, enquanto no frontend, ferramentas como React aplicam

conceitos similares, com o estado como Model, a interface como View e funções atuando como Controller.

Em SPAs (Single Page Applications), o backend funciona como API e o frontend cuida da interface dinâmica. Essa abordagem deixa o desenvolvimento mais flexível, o código organizado e a manutenção mais simples.