**Atividade**

Revisão sobre MVC

**Karina Casola**

**Cássio Henrique Machado**

**Everton Hildebrando**

**João Marimon**

**Leonardo kutianski**

**Rafael de Freitas**

**Descrição**

Nesta atividade, você deverá responder a um questionário composto por 10 perguntas que abordam o padrão de arquitetura MVC (Model-View-Controller). Cada resposta deverá ser elaborada em formato argumentativo, contendo aproximadamente um parágrafo, com o objetivo de incentivar a reflexão crítica sobre o funcionamento e a aplicação desse padrão no desenvolvimento de software.

As perguntas abrangem desde a definição e os conceitos fundamentais do MVC até a análise de seus benefícios, desafios e sua relevância em arquiteturas modernas. Espera-se que, ao final, você demonstre uma compreensão do tema, explorando como a separação de responsabilidades entre Model, View e Controller impacta positivamente o desenvolvimento de aplicações robustas, escaláveis e de fácil manutenção.

**Desafio**

1. Responda cada pergunta com um parágrafo argumentativo.
2. Use exemplos práticos para ilustrar suas respostas quando necessário.
3. Desenvolver a atividade em dupla.
4. Responder às questões aleatoriamente.

**Perguntas**

1. O que é o padrão MVC e qual sua importância no desenvolvimento de software?

R: O MVC é um padrão de projeto para o desenvolvimento de software, que facilita o desenvolvimento de aplicações, de forma que todas as partes do projeto sejam muito bem definidas. Trazendo uma facilidade para a manutenção das aplicações.

1. Como o padrão MVC facilita o desenvolvimento colaborativo em equipes?

R: Aumento de produtividade;

* Uniformidade na estrutura do software;
* Redução de complexidade no código;
* Facilita a documentação;
* Estabelece um vocabulário comum de projeto entre desenvolvedores;
* Reduz o tempo de desenvolvimento de um projeto.

1. Quais são os principais benefícios da separação de responsabilidades proporcionada pelo MVC?

R: Os principais benefícios que o MVC proporciona na separação de responsabilidades é a facilidade de desenvolvimento de aplicações na distribuição de tarefas, assim aumentando a produtividade da equipe responsável e uniformizando toda a estrutura do código.

1. Como o padrão MVC contribui para a manutenção e evolução de sistemas ao longo do tempo?

R: Um dos maiores benefícios do Padrão MVC dentro de um projeto é a sua facilidade na manutenção devido ao fato de que, se for necessário a alteração no código, não será necessário mudar sua lógica, por conta de que quando se altera uma parte do código essa alteração é individual e não universal. E isso também contribui para a evolução do sistema como um todo, por sempre ser possível atualizar o código sem quebrar todo o projeto.

1. Qual a função do "Model" no padrão MVC e por que ele é crucial para o funcionamento da aplicação?

R: O Model dentro do MVC é a camada no qual intermedia todas as funções da API. Nesta camada onde se encontra a interação entre a entrada e a visualização dos dados para o usuário. E ao possuir os dados o Model também pode agir como alimentador da base do banco de dados.

1. Como o "View" no padrão MVC garante uma melhor experiência para o usuário?

R: A View é camada responsável por proporcionar a visualização dos dados do usuário

1. Qual a responsabilidade do "Controller" no padrão MVC e como ele coordena as interações entre Model e View?

R:Ele tem a responsabilidade de interpretar as entradas do mouse e teclado enviadas ao sistema pelo usuário mapeando essas ações em comandos que são enviados para o Model e/ou para a janela de visualização (View) para efetuar a alteração apropriada

1. Como o MVC melhora o processo de depuração (debugging) e teste de aplicações?

R:Por o MVC ter a natureza de isolar as responsabilidades de cada camada, conforme cada teste de aplicação, ele facilita o debugging e correção do projeto.

1. Quais são os principais desafios na implementação do padrão MVC em projetos grandes?

R: Os principais desafios em projetos grandes é a constância na padronização no código entre as equipes, e separação de responsabilidades entre os membros da equipe. Além da organização das pastas da aplicação para lidar com a escalabilidade e manutenção.

1. O padrão MVC ainda é relevante em arquiteturas modernas, como as baseadas em micro serviços?

R: Sim, a interação do “controller” ainda é uma excelente prática para a organização e escalabilidade do código. Embora o MVC foque na separação de responsabilidades dentro de uma aplicação, ele não se limita apenas a elas, sendo útil mesmo em sistemas distribuídos para manter uma estrutura organizada.

**Desenvolvimento**

**Questão 1**

…

**Questão 2**

…

**Questão 3**

…