Resumo Arquitetura MVC (Model-View-Controller)

A arquitetura MVC (Model-View-Controller) é um padrão de design que divide uma aplicação em três partes distintas para melhor organização e escalabilidade. Vamos resumir cada parte do MVC:

- \*\*Model (Modelo)\*\*:

Lida com dados e regras de negócios. Gerencia a lógica do sistema, interagindo com bancos de dados ou outras fontes de dados. Em C#, pode ser uma classe representando entidades como "Usuário" ou "Produto".

- \*\*View (Visão)\*\*:

Responsável pela apresentação. Mostra dados ao usuário e recebe suas interações. Não deve conter lógica de negócios. Em C#, pode ser uma página HTML em ASP.NET Core, componentes WPF ou formulários Windows Forms.

- \*\*Controller (Controlador)\*\*:

Atua como intermediário entre a View e o Model. Recebe solicitações do usuário, interage com o Model e retorna a resposta correta (atualizando a View ou enviando dados). Em C#, os Controllers recebem solicitações HTTP e decidem quais operações executar.

### Como o MVC Funciona:

1. \*\*Interação do Usuário\*\*: O usuário interage com a View.

2. \*\*Processamento do Controller\*\*: O Controller recebe a interação, chama o Model para obter ou alterar dados e escolhe como responder.

3. \*\*Atualização da View\*\*: A View é atualizada com a resposta do Controller.

### Benefícios do MVC:

- \*\*Separação de Preocupações\*\*: Mantém a aplicação organizada, separando lógica de dados, apresentação e controle.

- \*\*Facilidade de Testes\*\*: Permite testar o Model e o Controller sem envolver a View.

- \*\*Flexibilidade\*\*: Facilita adicionar novas funcionalidades ou interfaces.

- \*\*Colaboração\*\*: Permite que equipes trabalhem em diferentes partes do sistema sem conflitos.

Em resumo, o MVC ajuda a manter a aplicação organizada, facilitando manutenção, testes e expansão de recursos. É amplamente utilizado para criar sistemas bem estruturados em C#.