

## 1. Introdução

- Definição de MVVM
- Importância da arquitetura na criação de aplicações modernas
- Objetivo do trabalho: Implementar uma aplicação simples utilizando MVVM

## 2. Fundamentos do MVVM

- Descrição dos componentes principais:
  - **Model**: Como a camada de dados é gerenciada
  - **View**: A interface do usuário e sua interação
  - **ViewModel**: O papel de mediação entre Model e View
- Diferenças entre MVVM e outras arquiteturas (MVC, MVP)

## 3. Princípios do MVVM

- Data Binding: Como a ligação de dados facilita a interação
- Comandos: A abordagem MVVM para gerenciamento de ações do usuário
- Separação de preocupações: Vantagens de manter componentes desacoplados

## 4. Exemplo Prático: Aplicação de Lista de Tarefas

- **Objetivo da Aplicação**: Criar uma aplicação simples para gerenciar uma lista de tarefas (To-Do List).
- **Componentes da Aplicação**:
  - **Model**: Classe `Task` que contém propriedades como `Title`, `IsCompleted`, etc.
  - **ViewModel**: Classe `TaskViewModel` que gerencia a lista de tarefas, adiciona novas tarefas e atualiza o status das existentes.
  - **View**: Interface do usuário que exibe a lista de tarefas e permite adicionar ou marcar tarefas como concluídas.
- **Implementação**:
  - Utilizar uma linguagem/framework (ex: C# com WPF ou JavaScript com Angular).
  - Demonstrar a ligação de dados entre a View e o ViewModel.
  - Implementar comandos para adicionar e remover tarefas.

## 5. Casos de Uso e Benefícios

- Cenários em que o MVVM é mais eficaz na aplicação de lista de tarefas.
- Benefícios em termos de manutenibilidade, testabilidade e escalabilidade.

## 6. Desafios e Limitações

- Potenciais desafios na implementação do MVVM na aplicação de lista de tarefas.
- Limitações da arquitetura e quando considerar alternativas.

## 7. Conclusão

- Recapitulação dos pontos principais.
- Considerações finais sobre o uso de MVVM em projetos de software.

## **8. Referências**

- Livros, artigos e recursos online utilizados para pesquisa.

## **Avaliação**

- Clareza e organização do conteúdo.
- Compreensão dos conceitos de MVVM.
- Implementação prática da aplicação de lista de tarefas.
- Qualidade da apresentação (formatação, gramática, etc.).

Esse escopo fornece uma estrutura sólida para o trabalho, permitindo que os alunos não apenas compreendam a teoria por trás do MVVM, mas também a apliquem em um projeto prático. Se precisar de mais detalhes ou alterações, é só avisar!