

# Resumo dos Padrões de Projeto Mobile

## 1. MVC (Model-View-Controller)

### Definição:

- **Model:** Representa os dados e a lógica de negócios.
- **View:** A interface do usuário que exibe os dados do Model.
- **Controller:** Gerencia as entradas do usuário e atualiza o Model e a View conforme necessário.

### Vantagens:

- Separação clara entre a interface do usuário e a lógica de negócios.
- Facilita o teste de unidades (Unit Testing).

### Desvantagens:

- Pode se tornar difícil de manter em projetos maiores devido à complexidade crescente das interações entre o Model, View e Controller.

## 2. MVP (Model-View-Presenter)

### Definição:

- **Model:** Contém a lógica de negócios e os dados.
- **View:** Define a interface de usuário, que é implementada pela atividade ou fragmento.
- **Presenter:** Atua como intermediário entre a View e o Model, manipulando a lógica de apresentação.

### Vantagens:

- Melhor separação de responsabilidades do que MVC.
- O Presenter não tem referência direta à View, facilitando o teste e a manutenção.

### Desvantagens:

- Pode adicionar complexidade ao código, especialmente em projetos menores.

## 3. MVVM (Model-View-ViewModel)

### Definição:

- **Model:** Gerencia os dados e a lógica de negócios.
- **View:** Representa a interface do usuário.

- **ViewModel:** Manipula a lógica de apresentação e se comunica com o Model. Atua como um link entre o Model e a View.

**Vantagens:**

- Facilita a vinculação de dados (Data Binding), o que simplifica o código da View.
- Melhorar a testabilidade da lógica de apresentação.

**Desvantagens:**

- Pode ser complexo para configurar, especialmente em projetos pequenos ou para desenvolvedores menos experientes.