

O padrão de projeto Model-View-ViewModel (MVVM) foi originalmente criado para aplicativos Windows Presentation Foundation (WPF) usando XAML para separar a interface do usuário (UI) da lógica de negócios e aproveitando ao máximo o data binding (a vinculação de dados).

Aplicações arquitetadas desta forma têm uma camada ViewModel distinta que não possui dependências de sua interface de usuário. Esta arquitetura em si é otimizada para testes de unidade, bem como para o desenvolvimento multiplataforma.

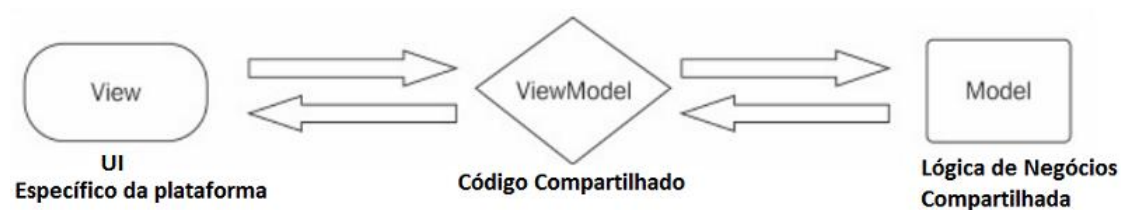
Como as classes ViewModel de um aplicativo não têm dependências sobre a camada de interface do usuário, você pode facilmente trocar uma interface de usuário iOS por uma interface Android e escrever testes contra a camada ViewModel.

O padrão MVVM é composto basicamente dos seguintes elementos:

Model: A camada de modelo é a lógica de negócios que impulsiona a aplicação e quaisquer objetos de negócios;

View: Esta camada é a interface do usuário. No caso do desenvolvimento cross platform, ela inclui qualquer código específico da plataforma para conduzir a interface do usuário da aplicação.

ViewModel: Esta camada age como a cola em aplicações MVVM. As camadas ViewModel coordenam as operações entre a view e as camadas model. Uma camada ViewModel irá conter propriedades que a View vai obter ou definir, e funções para cada operação que pode ser feita pelo usuário em cada view. A camada ViewModel também evocará operações sobre a camada Model, se necessário.



Referência: http://www.macoratti.net/16/09/net_mvvm1.htm

Vídeo: <https://www.youtube.com/watch?v=ijXjCtCXcN4>