

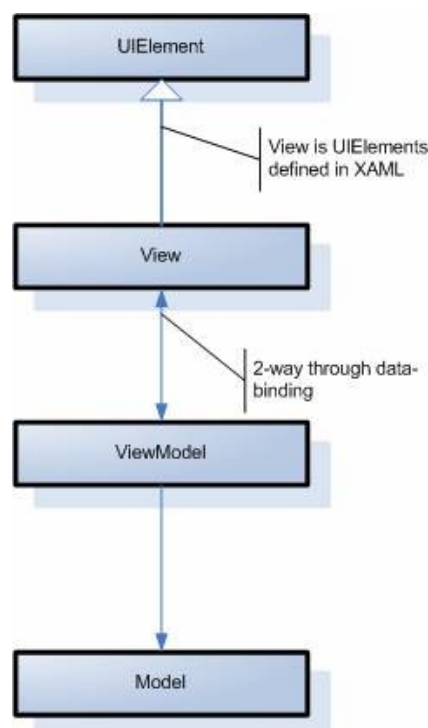
O MVVM é um pattern que foi criado em 2005, por John Gossman, um dos arquitetos do WPF e Silverlight na Microsoft. O MVVM, visa estabelecer uma clara separação de responsabilidades em uma aplicação WPF e Silverlight, mantendo uma espécie de fachada entre o Modelo de objetos e a View que é a interface, com a qual o usuário interage.

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.



Como ilustra a figura acima, há uma separação das camadas. A camada Model (Modelo) não conhece a View (Camada de apresentação) e vice-versa, na verdade a View conhece a ViewModel e se comunica com ela através do mecanismo de binding. E são os avançados mecanismos de binding, eventos roteados e comandos roteados, que fazem do MVVM um pattern poderoso para construção de aplicações WPF e Silverlight.

Link de um vídeo explicativo: <https://www.youtube.com/watch?v=EEc18BUrI7o>