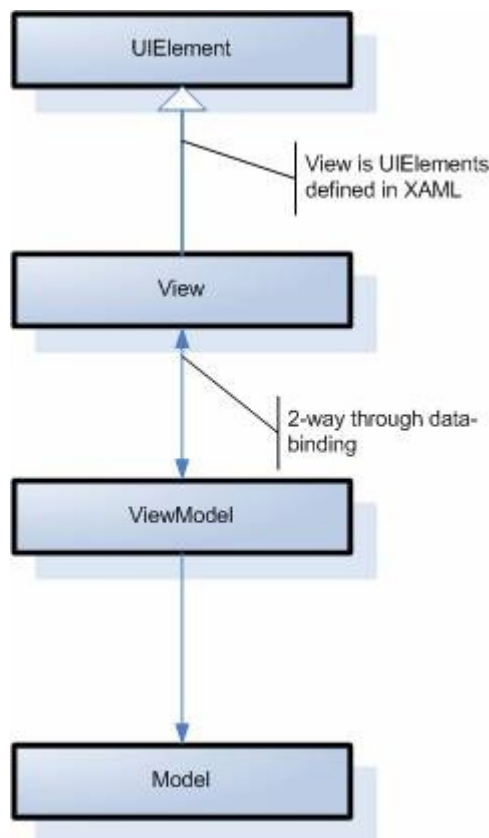


O MVVM é um pattern que foi criado em 2005, por John Gossman, um dos arquitetos do WPF e Silverlight na Microsoft. O MVVM, visa estabelecer uma clara separação de responsabilidades em uma aplicação WPF e Silverlight, mantendo uma espécie de fachada entre o Modelo de objetos (entenda classes de negócio, serviços externos e até mesmo acesso a banco de dados) e a View que é a interface, com a qual o usuário interage. Para entendermos melhor, como se dá esta separação e visualizar como os componentes interagem dentro deste cenário, observe a figura abaixo:



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.

O MVVM permite a você ter uma visão, da clara separação da Interface com o usuário(View), sua lógica de apresentação (ViewModel) e os seus Dados(Model). E trabalhando desta forma, temos separação de responsabilidades, desacoplamento e conseguimos evoluir e manter melhor as nossas aplicações.

View – A responsabilidade da View é definir a aparência ou estrutura que o usuário vê na tela. A View se liga ao ViewModel, através da propriedade **DataContext** que é setada para a classe ViewModel correspondente à aquela View.

ViewModel – A responsabilidade da ViewModel no contexto do MVVM, é disponibilizar para a View uma lógica de apresentação. A View Model não tem nenhum conhecimento específico sobre a view, ou como ela implementada, nem o seu tipo. A ViewModel implementa propriedades e comandos, para que a View possa preencher seus controles e notifica a mesma, caso haja alteração de estado; seja através de eventos ou notificação de alteração.

Model – o Model no MVVM, encapsula a lógica de negócios e os dados. O Modelo nada mais é do que o Modelo de domínio de uma aplicação, ou seja, as classes de negócio que serão utilizadas em uma determinada aplicação. O Modelo também contém os papéis e também a validação dos dados de acordo com o negócio, cuja aplicação em questão visa atender.

Referência: <https://www.devmedia.com.br/entendendo-o-pattern-model-view-viewmodel-mvvm/18411>

vídeo: <https://channel9.msdn.com/blogs/DX/Introduo-ao-MVVM>