

Inversão de controle, desacoplamento, injeção de dependência, são formas de manter um código limpo, fácil de dar manutenção e que seja organizado, nos fazendo assim entender e amadurecer no processo de desenvolvimento de software.

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.

### Responsabilidades e características

**View** – A responsabilidade da View é definir a aparência ou estrutura que o usuário vê na tela. O ideal é que o codebehind da view, contenha apenas a chamada ao método `InitializeComponent` dentro do construtor, ou em alguns casos, código que manipule os controles visuais, ou crie animações; algo que é mais difícil de fazer em XAML. A View se liga ao ViewModel, através da propriedade `DataContext` que é setada para a classe ViewModel correspondente à aquela View.

Exemplo imagético do padrão MVVM:

