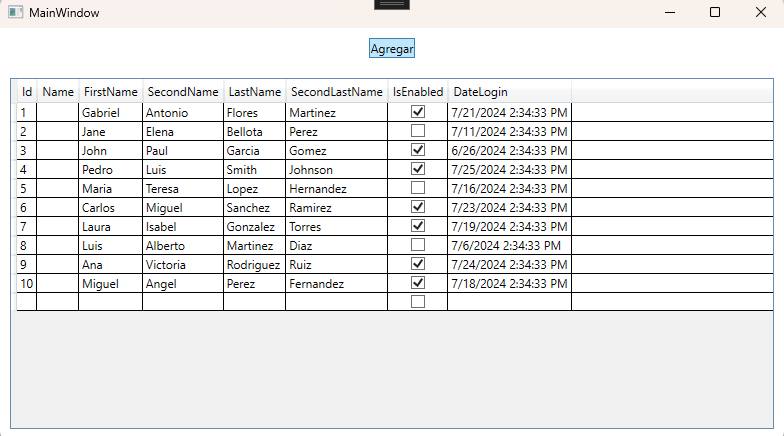
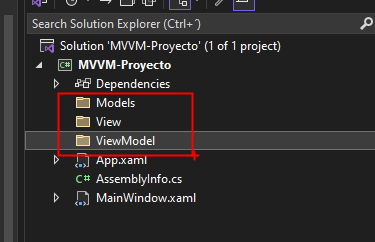
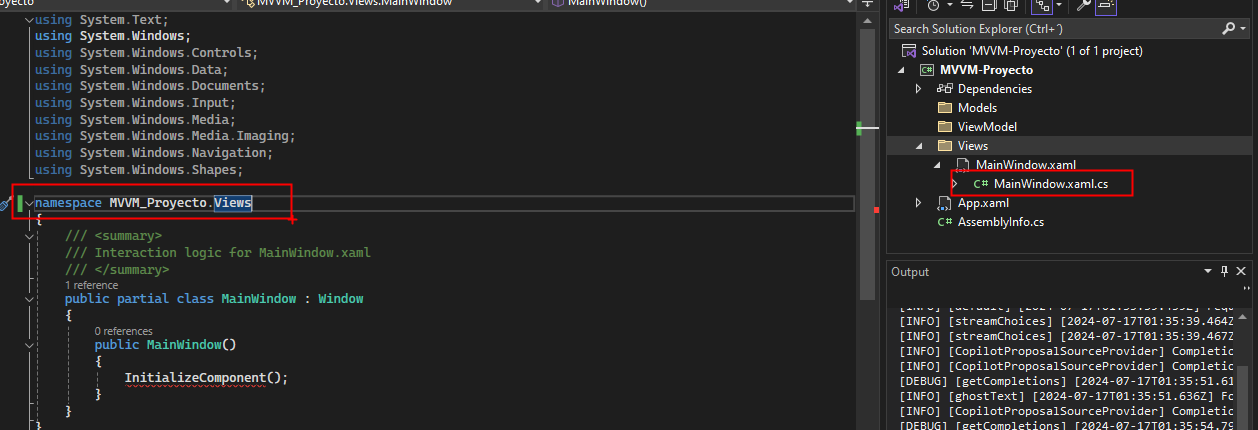
**Proyecto con MVVM en WPF**

****

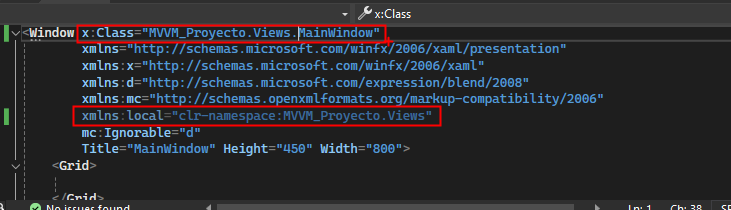
1. Creamos un Nuevo proyecto en WPF
2. Creamos las carpetas: **View**, **Model** y **ViewModel**



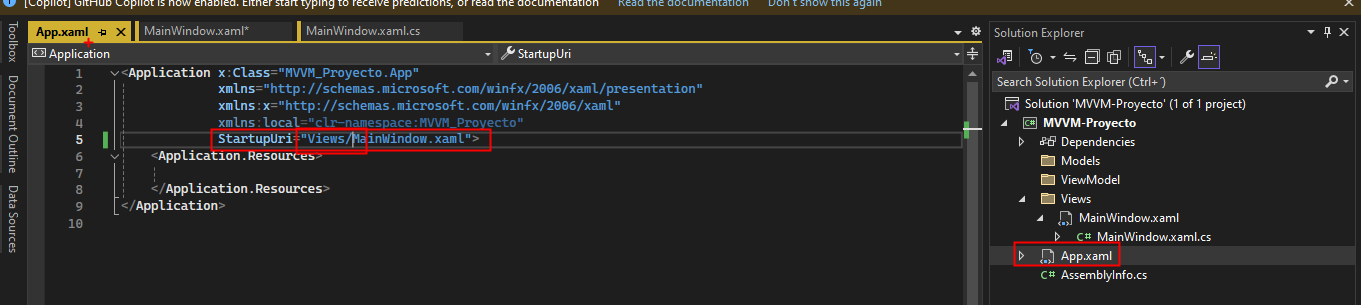
1. Pasamos nuestro archivo **MainWindow.xaml** que representa mi vista a la carpeta **View y** reajustamos el **nameSpace** de nuestro archivo **MainWindow.xaml.cs**



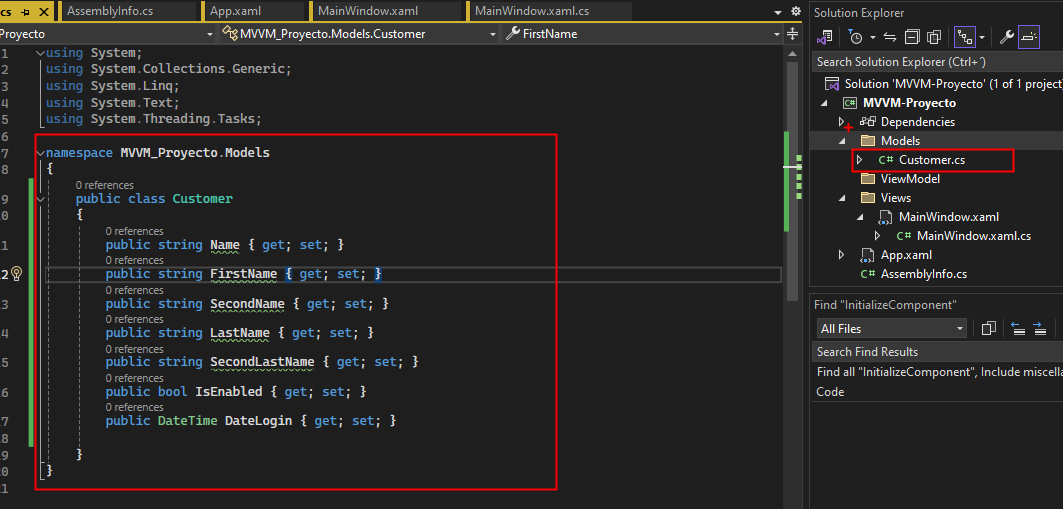
1. Luego tengo que modificar mi diseñador **Xaml:**



1. En el archivo **App.xaml** tenemos que modificar:

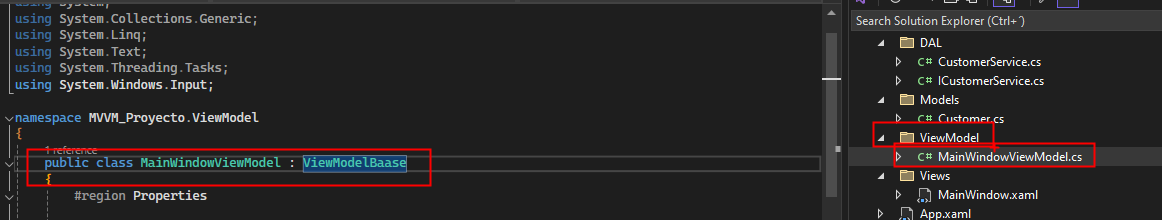


1. Ahora procedemos a crear nuestra clase **Customer** dentro de la carpeta **Model:**

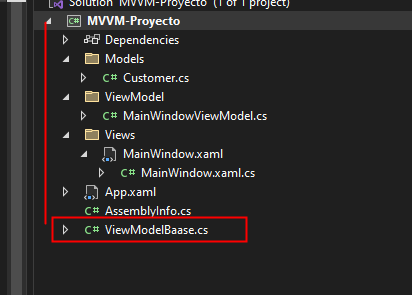
****

1. Ahora creamos otra clase dentro de la carpeta **ViewModel** la cual tendrá por nombre **MainWindowViewModel** y tendrá la función de tener toda la lógica de la aplicación.

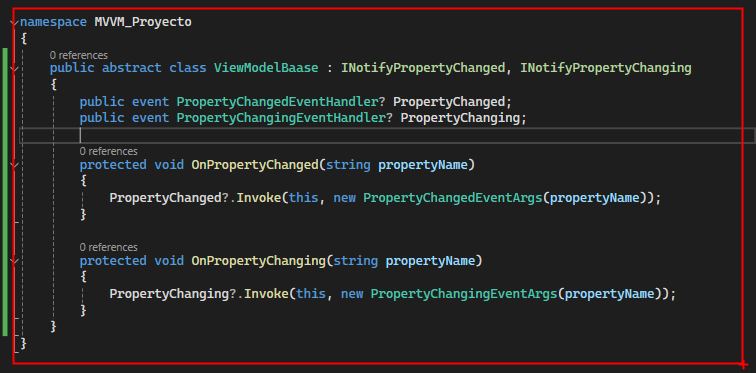
Esta clase **MainWindowViewModel** heredara de una clase **ViewModelBase** que también crearemos ahora:



1. Entonces creamos una clase **abstracta** de nombre **ViewModelBase** por fuera de las carpetas creadas a esta altura:



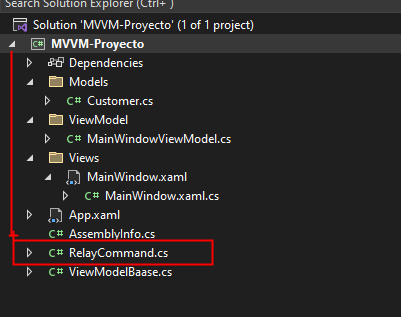
Y hacemos la implementación:



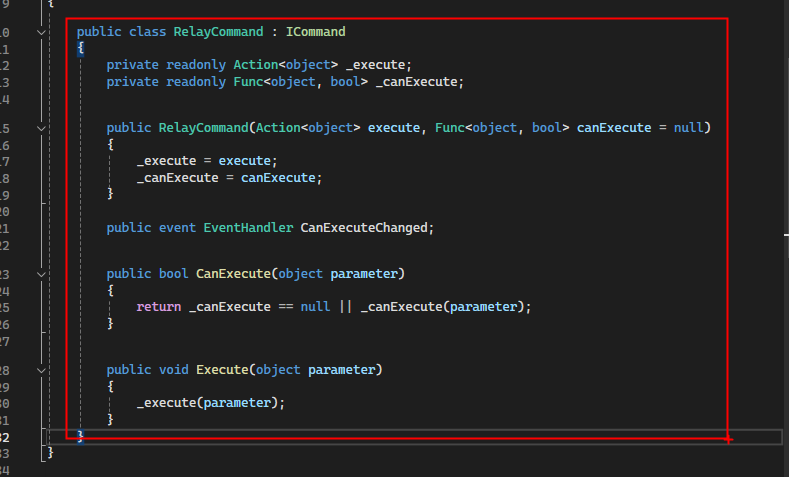
Lo cual es necesario para que puede usarse en mi clase **MainWindowViewModel**

Esta clase **ViewModelBase** no es más que declarar una clase base que va a heredar de dos interfaces muy importantes a la hora de cambiar una propiedad dentro de una vista que es **INotifyPropertyChanged** y **INotifyPropertyChanging** ya que notificamos a la vista que ocurrió un cambio.

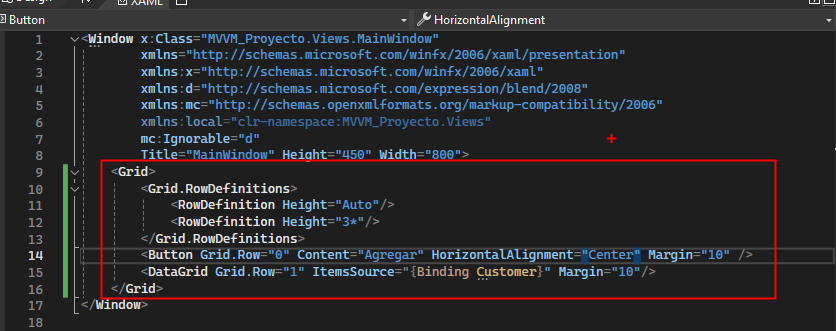
1. Seguidamente tenemos que crear nuestra clase **RelayCommand** que será necesario de igual forma para mi **MainWindowViewModel**:



Y ponemos el siguiente código:

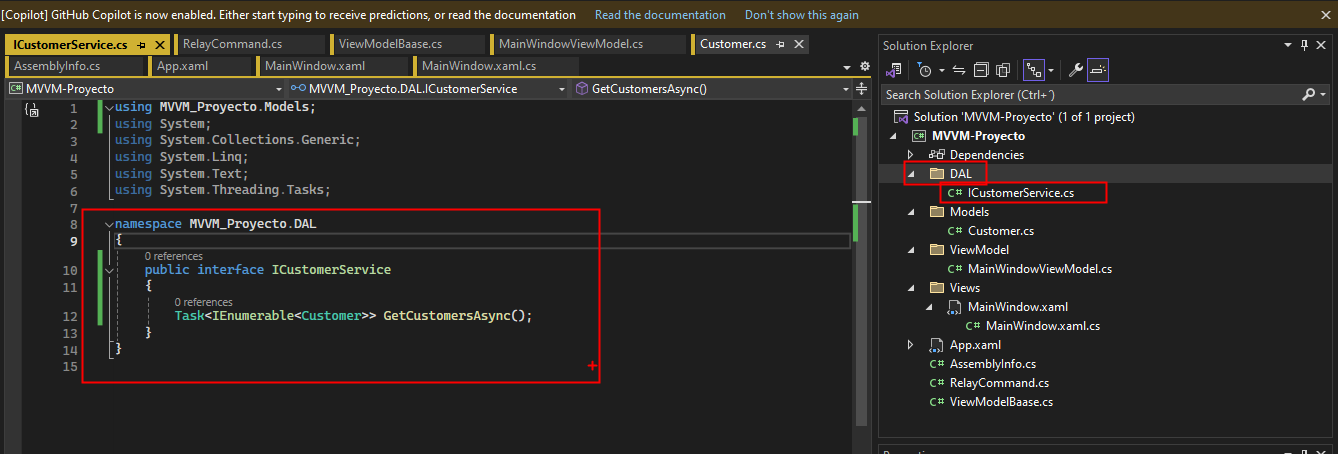


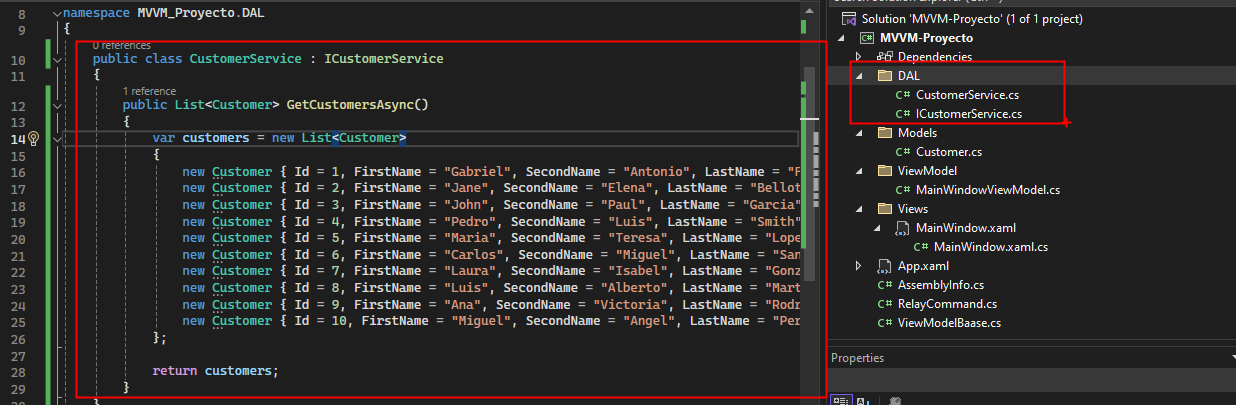
1. Ahora implementamos la vista:



1. Seguidamente vamos a crear una clase que simulara nuestra base de datos, que se llamara **ICustomerService**:

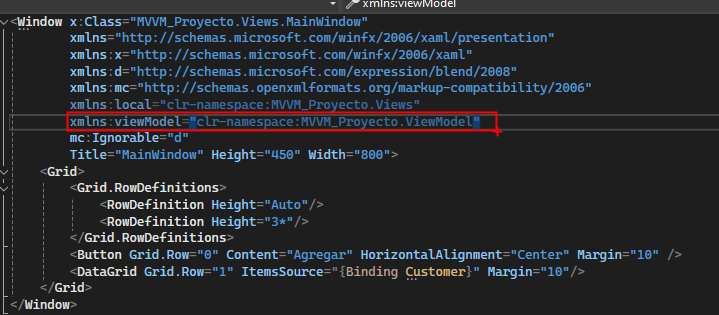
Para ello primeramente crearemos una carpeta de nombre **DAL** la cual contendrá la interfaz y su implementación, esta carpeta estará por fuera de las demás carpetas creadas:





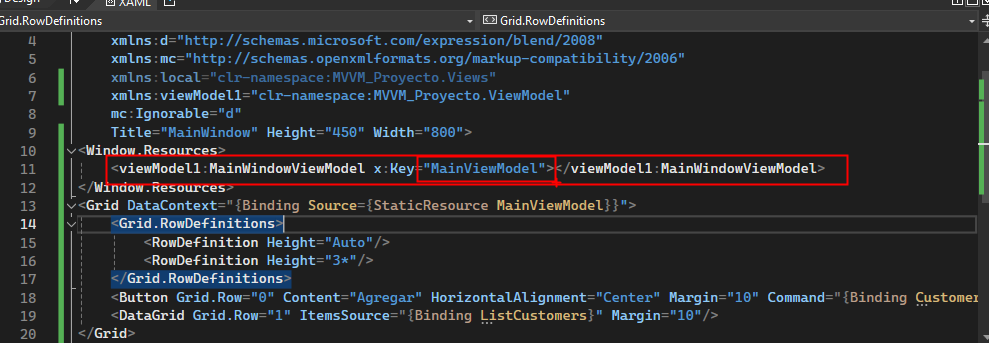
1. Ahora si pasamos a desarrollar la implementación como tal de **MVVM:**

Primeramente, tenemos que traer nuestra clase **MainWindowViewModel** que se encuentra dentro de la carpeta **ViewModel:**

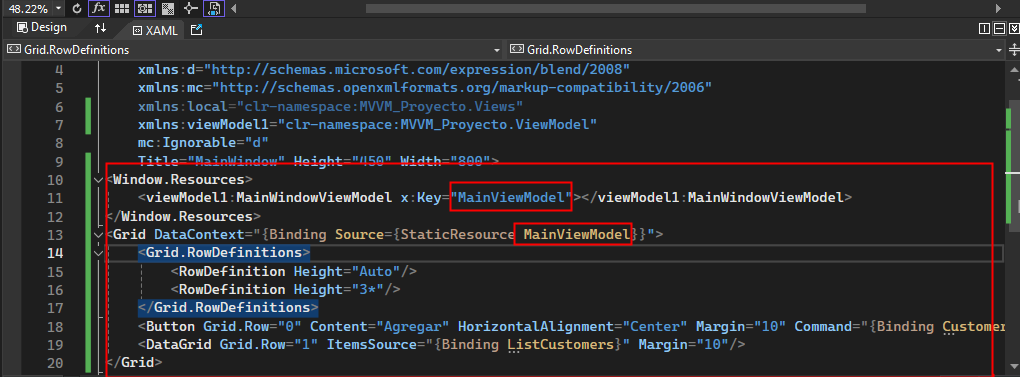
****

1. Ahora seguidamente para usar nuestra clase **MainWindowViewModel** y poder enlasarla con nuestra vista, tenemos que hacer uso de **Window.Resource:**

Y con nuestro **nameSpace** creado de nombre **viewModel1** llamamos a la clase que vamos a usar que es **MainWindowViewModel**, y posteriormente le indicamos la llave con la cual nos vamos a referir a ella, en este caso**: MainViewModel:**

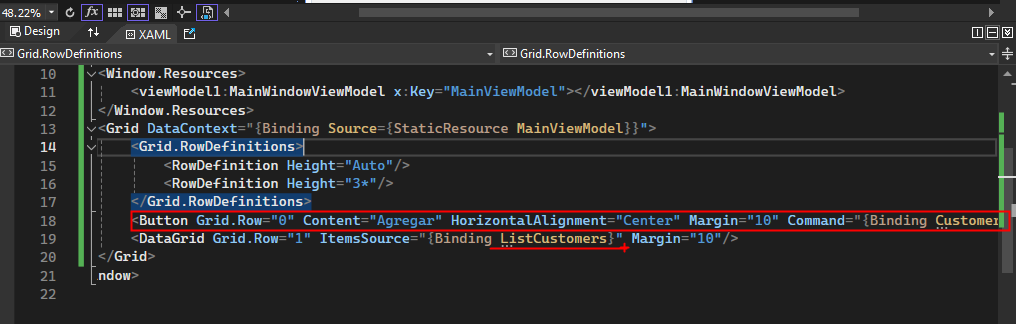
****

Ahora en nuestro **Grid** usaremos la propiedad **DataContext** para que con el **binding** se pueda enlazar con dicha clave **MainViewModel:**

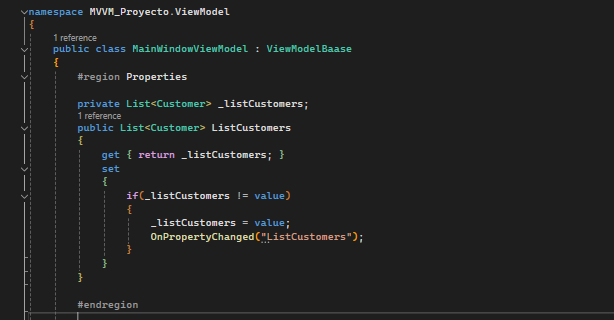
****

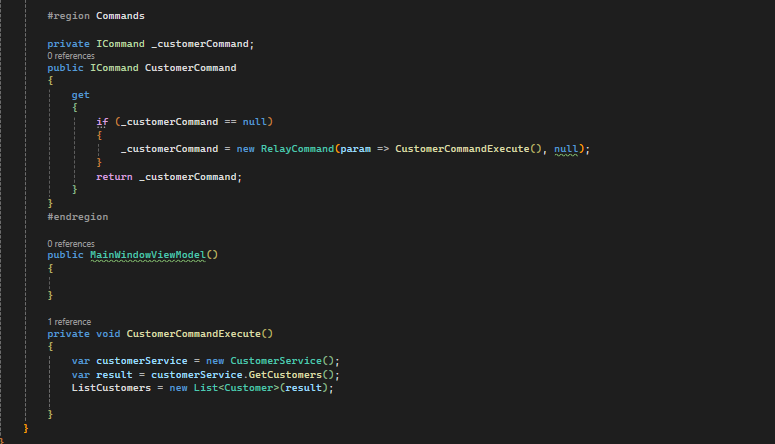
Con esto ya hicimos el enlace de la vista **MainWindow.xaml** a **ViewModel** y el **ViewModel** va a llamar cosas del **Model**

1. Ahora procederemos a disparar un **command** en el **button**:



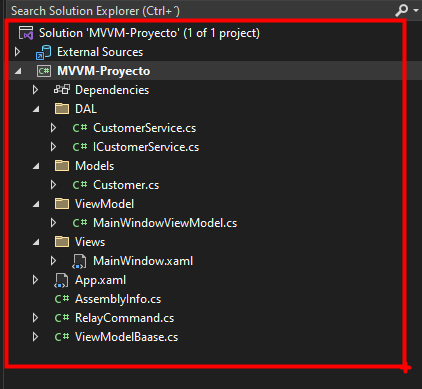
1. Posteriormente nos dirigimos a nuestra clase que representa nuestro **ViewModel** con el nombre de **MainWindowViewModel** y marcamos dos regiones la región de **properties** y la región de **comands**, seguidamente más abajo pondremos el constructor y los métodos necesarios, en ese orden:



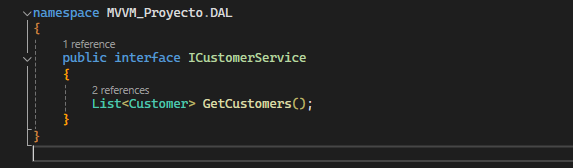


1. Probamos nuestra aplicación

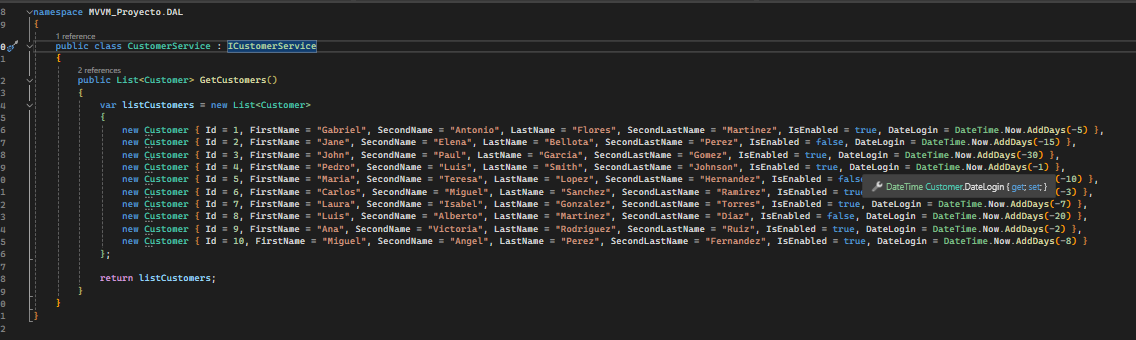
**Al final debemos tener esta estructura:**



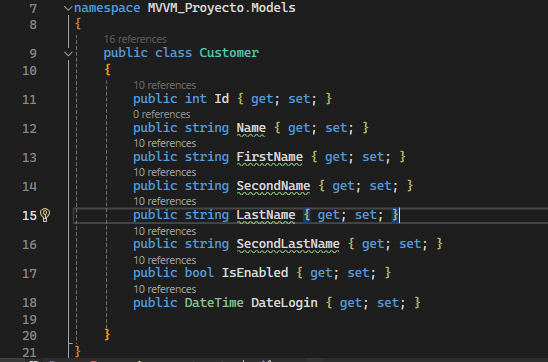
**ICustomerService:**



**CustomerService:**

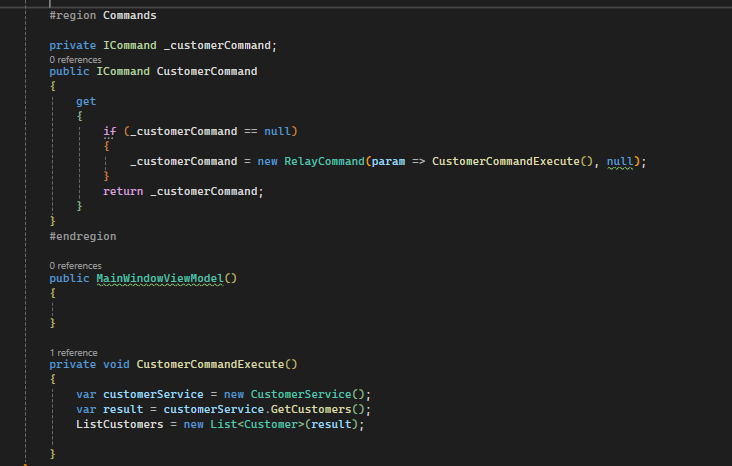


**Customer:**

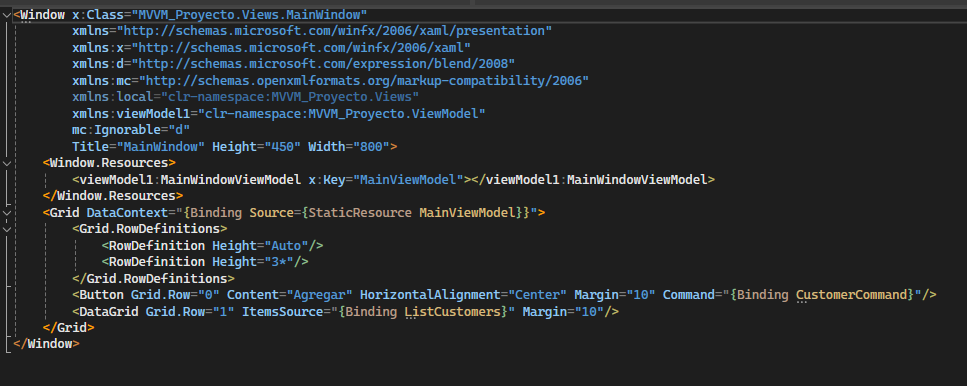


**MainWindowViewModel:**

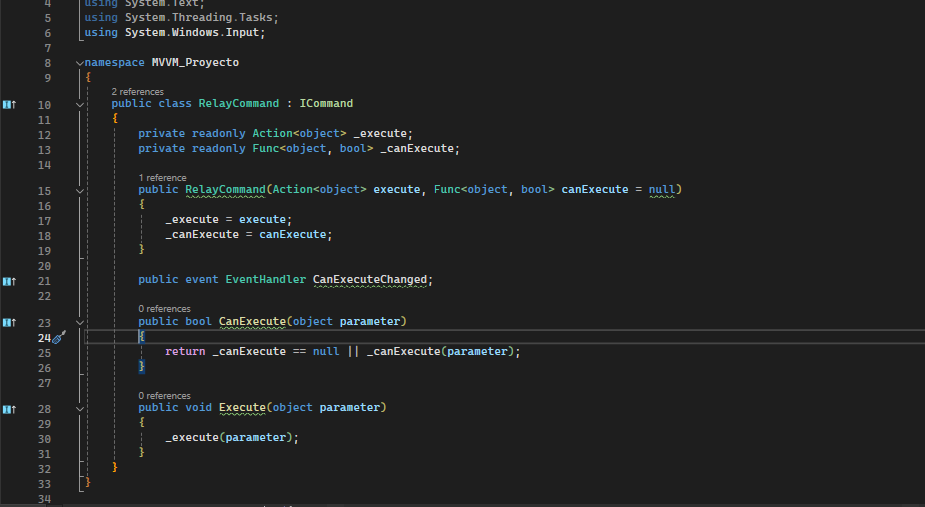




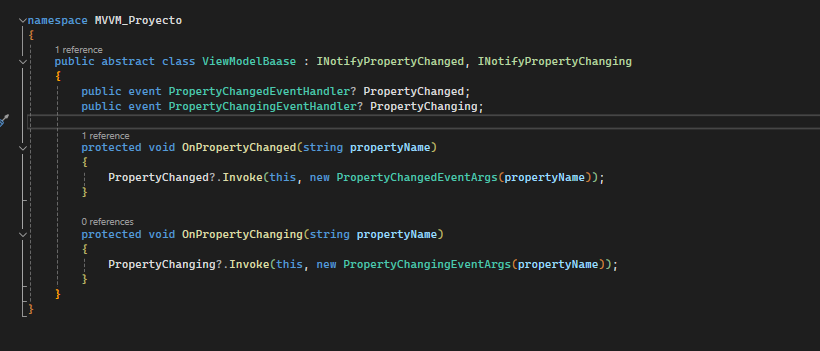
**MainWindow.xaml**



**RelayCommand**

****

**ViewModelBase:**

****