PRÁCTICA 06 – EF

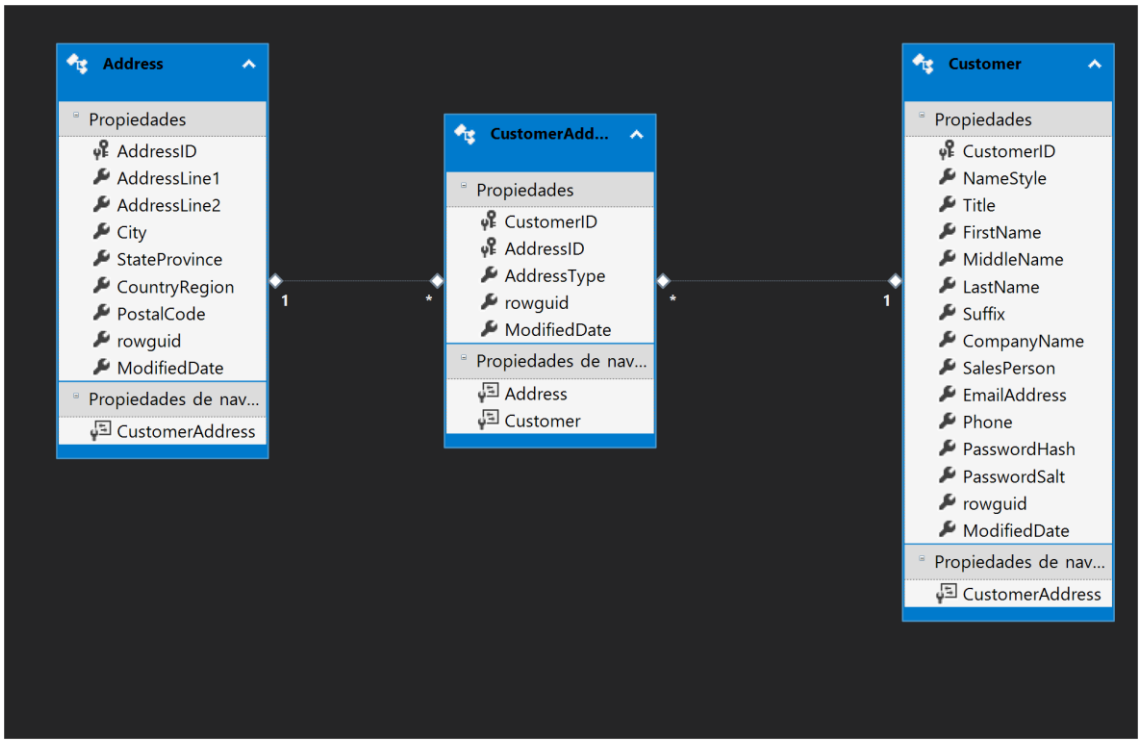
EF con MVVM

DESARROLLO RÁPIDO DE APLICACIONES

Por Juan José Camacho Hidalgo
UNIVERSIDAD DE ALMERÍA

1. EFMVVMWpfApp

Mediante Entity Framework, haciendo uso de database-first. Tenemos una base de datos AdventureWorksLT, y en base a esta generamos un modelo en el proyecto almacenándose en EFMVVMClassLibrary, con el siguiente diagrama de datos:

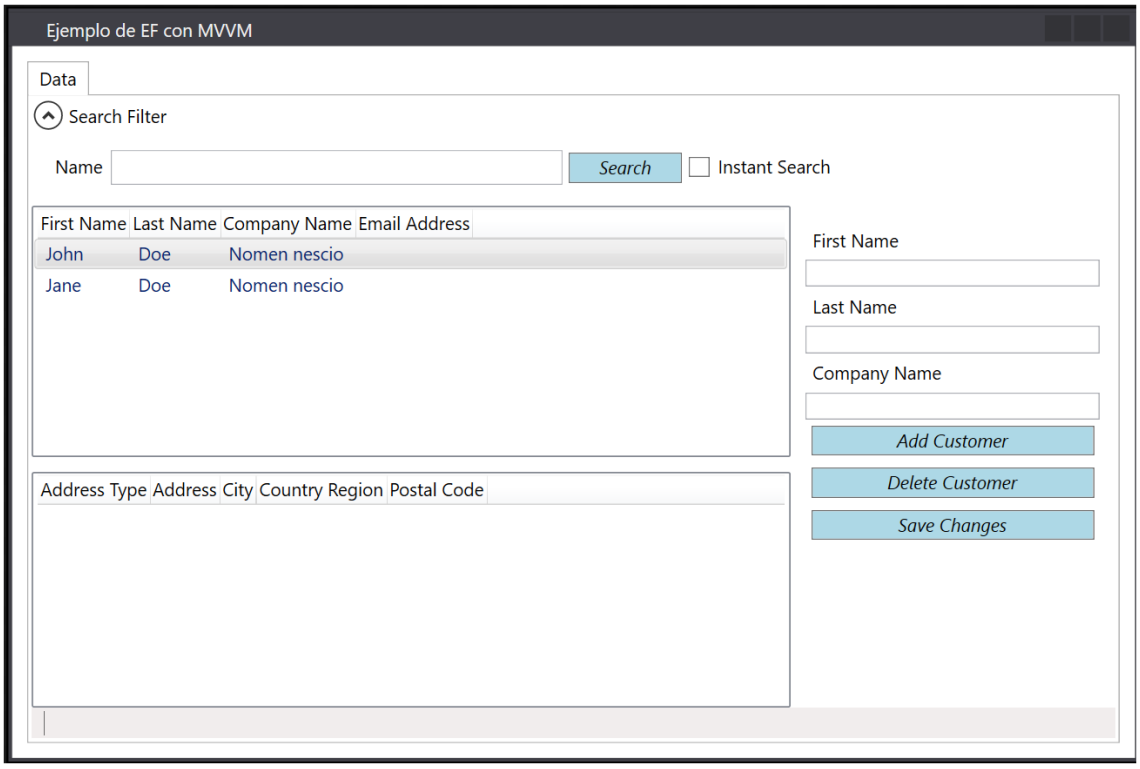


En este modelo encontramos los modelos de Address, Customer y CustomerAddress. Por tanto ya disponemos de ellos.

Para la conexión con la base de datos, es muy importante mantener correctamente App.config, donde en connectionStrings, tenemos todos los datos de acceso personales, pero

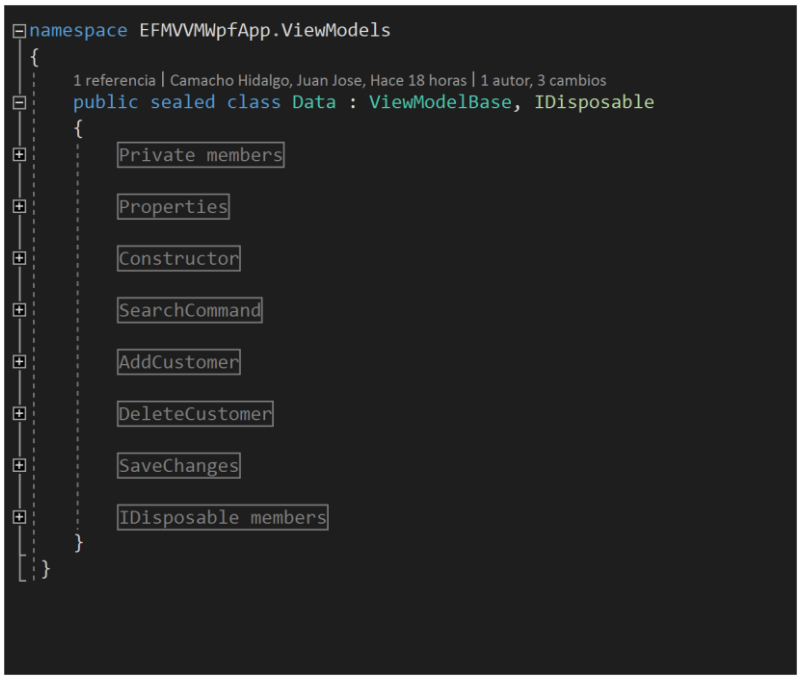
creando y asociando las bases de datos desde explorador de servidores y orígenes de datos, es modificado automáticamente por Visual Studio , si así lo deseamos, como opción.

La vista, siguiendo el patrón MVVM, se trataría de Data.xaml, que es la captura que muestro a continuación, y MainWindow, que sería la principal.



En la carpeta Services, disponemos de ServiceLocator, un patrón para encapsular los procesos, y MsgBoxService, para los mensajes que se muestran por pantalla, implementando la interfaz IMsgBoxService (método Show).

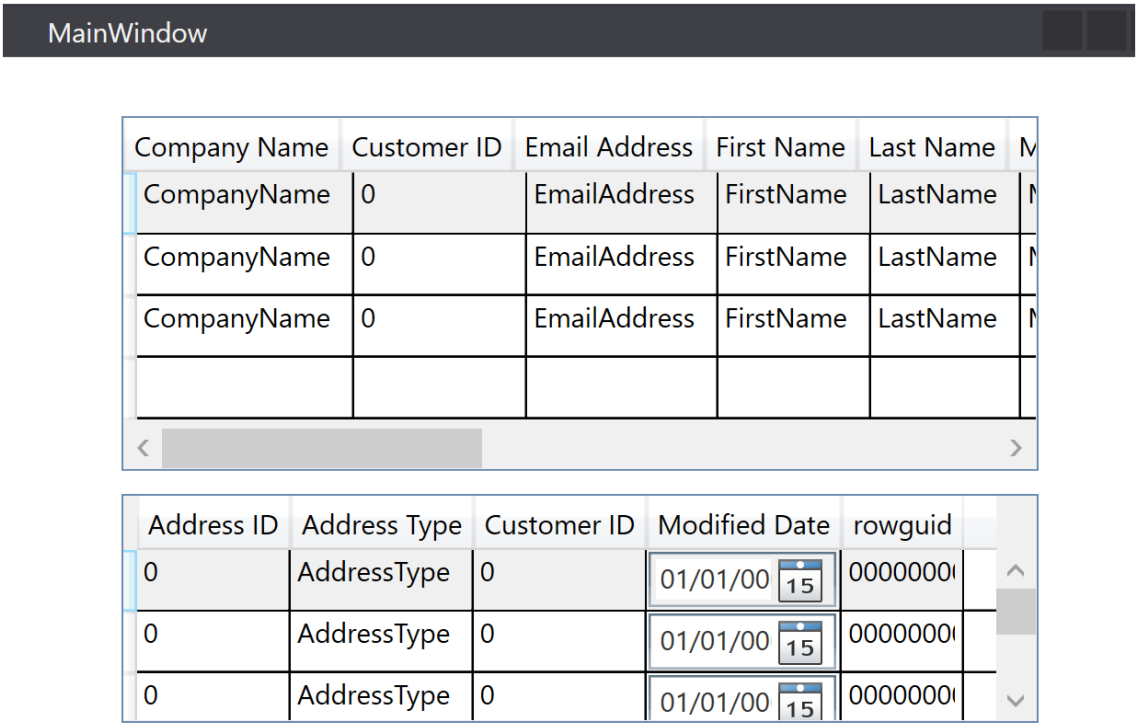
La clase Data conlleva todos los métodos del modelo de la vista, que hacen uso de la base de datos, y los que no. Define el comportamiento de la vista.



Además, los métodos están definidos mediante la interfaz ICommand, de forma que la interacción se realiza mediante comandos.

Como podemos observar, también disponemos de las clases ViewModelBase que implementa INotifyPropertyChanged, de cara a informar sobre propiedades que cambien en el servicio, y RelayCommand, para comprobar si puede ejecutarse un comando.

Por último, el proyecto EFWpfApp tiene como objetivo comprobar el buen funcionamiento de este, para ello, hace uso de esta vista:



Comprueba la carga correcta de los datos de la base de datos.

Y dentro se encarga la clase MainWindow de incorporar los eventos de apertura y cierre de la ventana:

```
3 referencias | Camacho Hidalgo, Juan Jose, Hace 18 horas | 1 autor, 3 cambios
public partial class MainWindow : Window
{
    private AdventureWorksLT2008Entities _context = new AdventureWorksLT2008Entities()
    0 referencias | Camacho Hidalgo, Juan Jose, Hace 18 horas | 1 autor, 3 cambios
    public MainWindow()
    {
        InitializeComponent();
    }

    1 referencia | Camacho Hidalgo, Juan Jose, Hace 18 horas | 1 autor, 3 cambios
    private void Window_Loaded(object sender, RoutedEventArgs e)
    {

        System.Windows.Data.CollectionViewSource customerViewSource = ((System.Windows
        // Cargar datos estableciendo la propiedad CollectionViewSource.Source:
        // customerViewSource.Source = [origen de datos genérico]
        _context.Customer.Load();
        customerViewSource.Source = _context.Customer.Local;
    }

    0 referencias | Camacho Hidalgo, Juan Jose, Hace 18 horas | 1 autor, 3 cambios
    protected override void OnClosing(System.ComponentModel.CancelEventArgs e)
    {
        base.OnClosing(e);
        this._context.Dispose();
    }
}
```