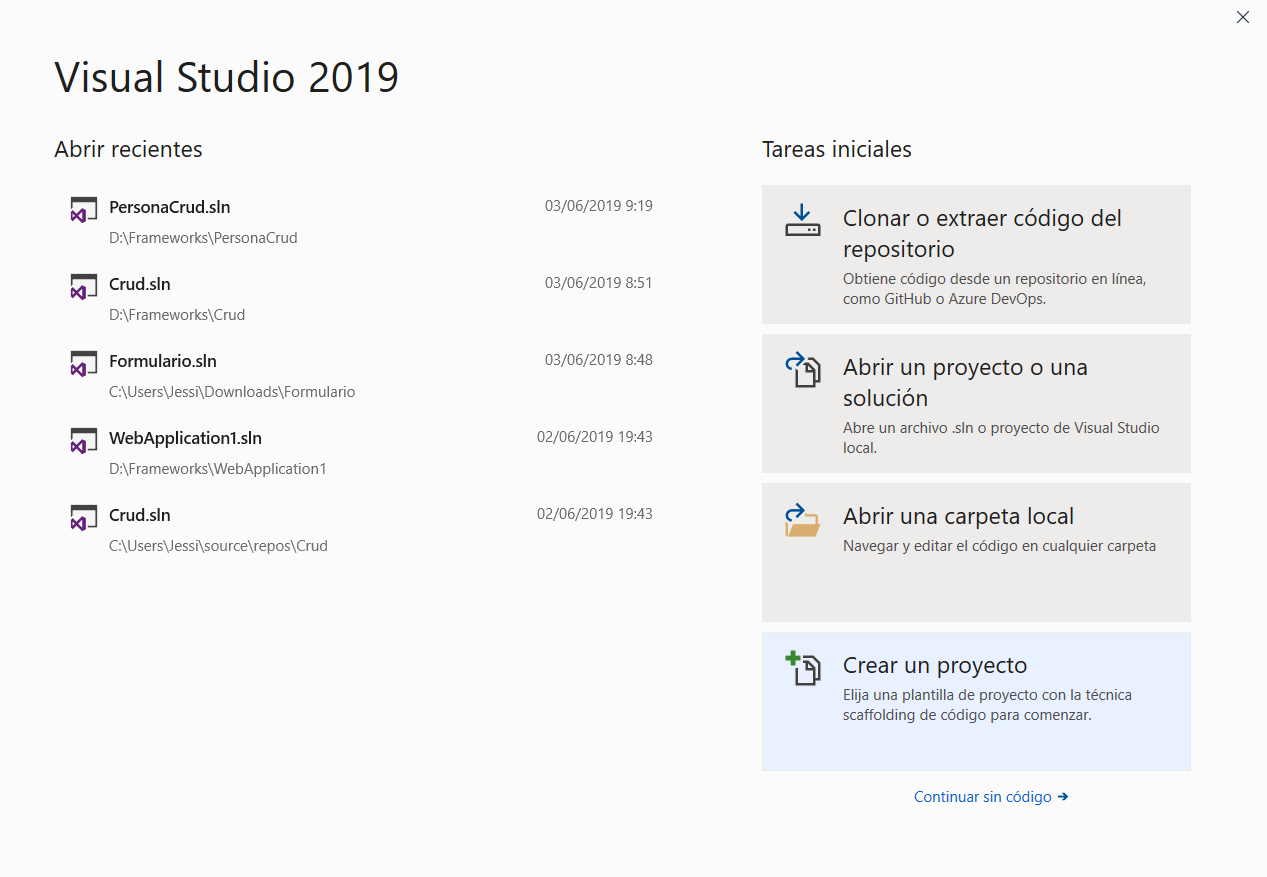
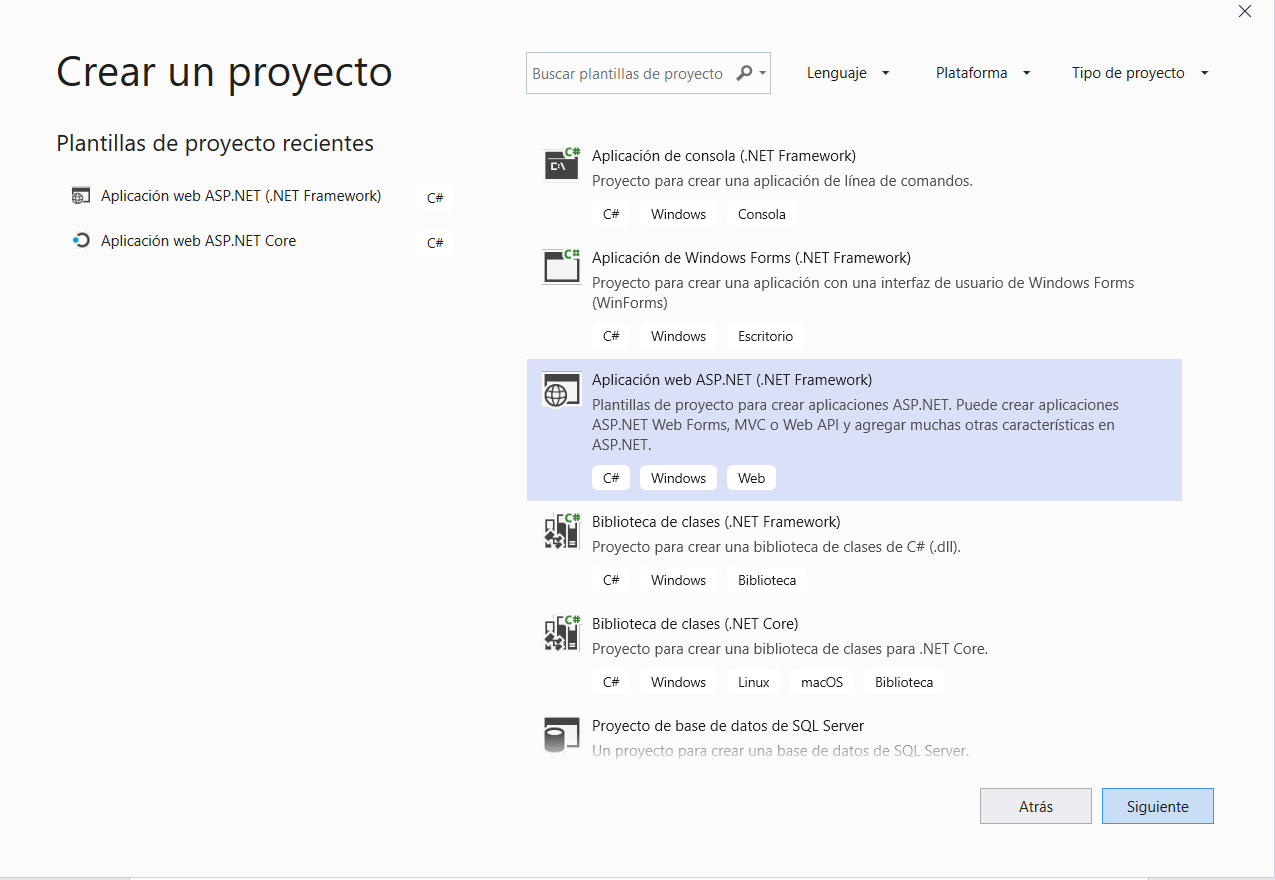
crud guide on asp.net

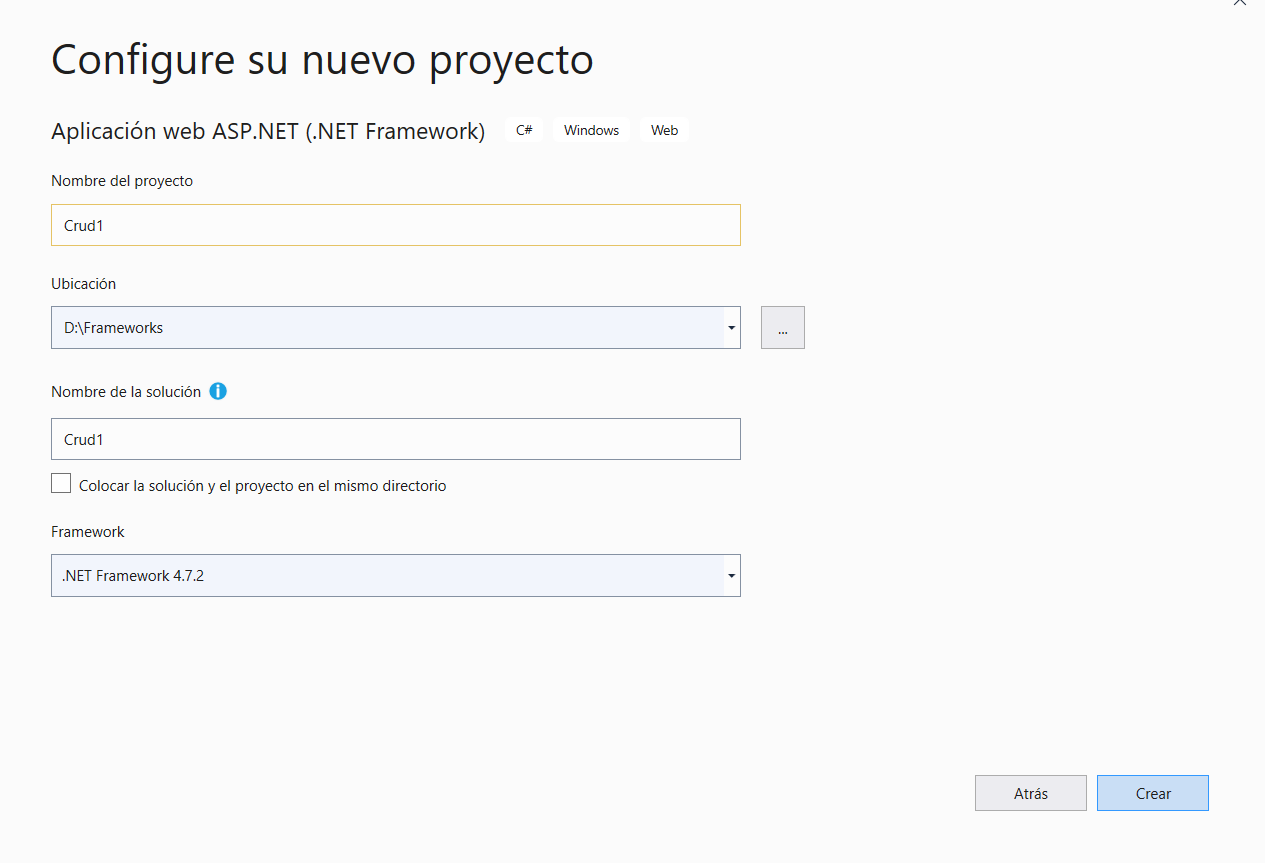
1.To begin, you create opens the Visual Studio IDE, then login with a Hotmail account and you subsequently select Create new project, so:

**

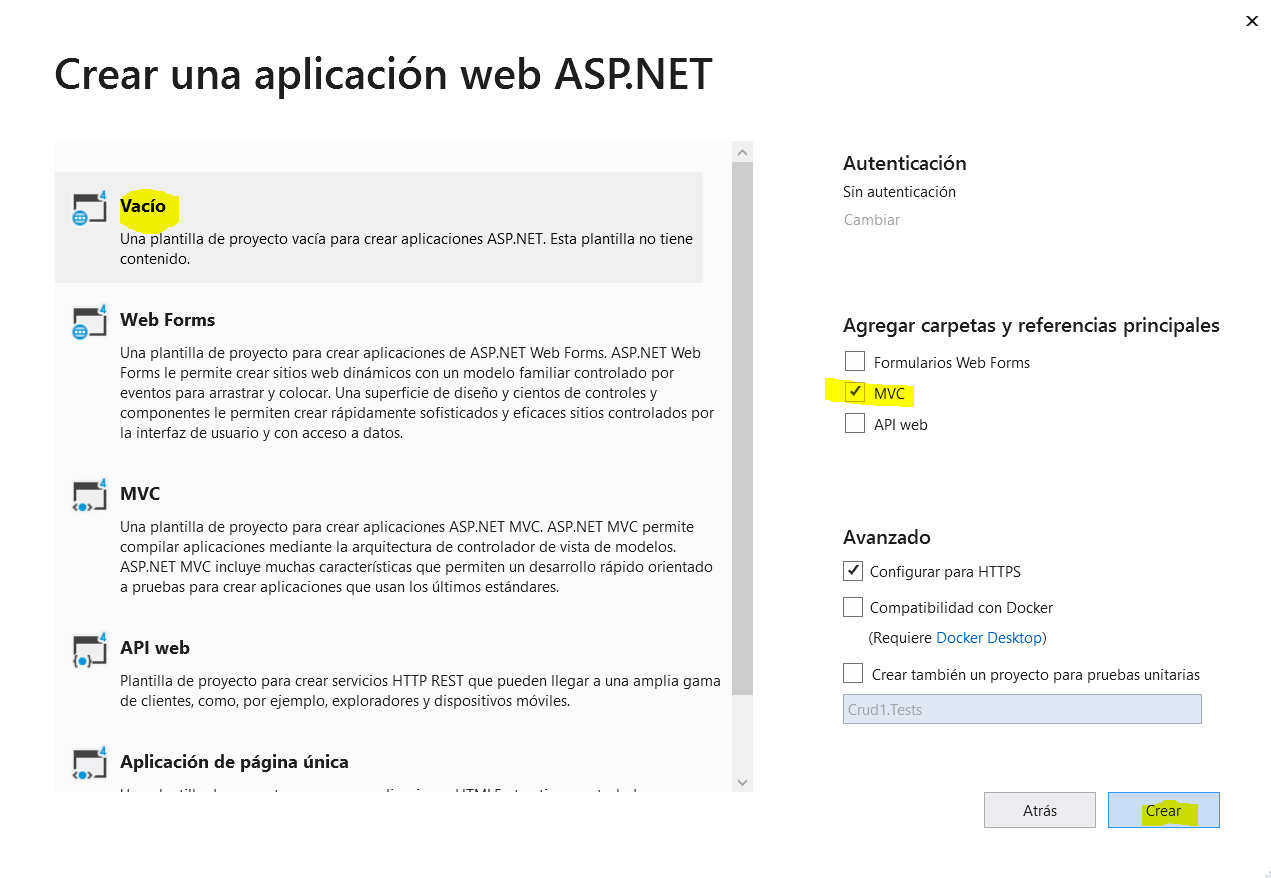
2. Select the item ASP.NET Web application (NET Framework) and click on next



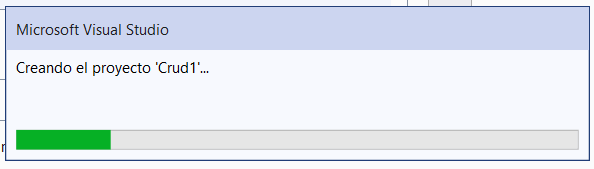
3. Writes the name of the project, its location, if you want to change the name of the solution or the default is left, and selects the version of the Framework that will work and click on create.



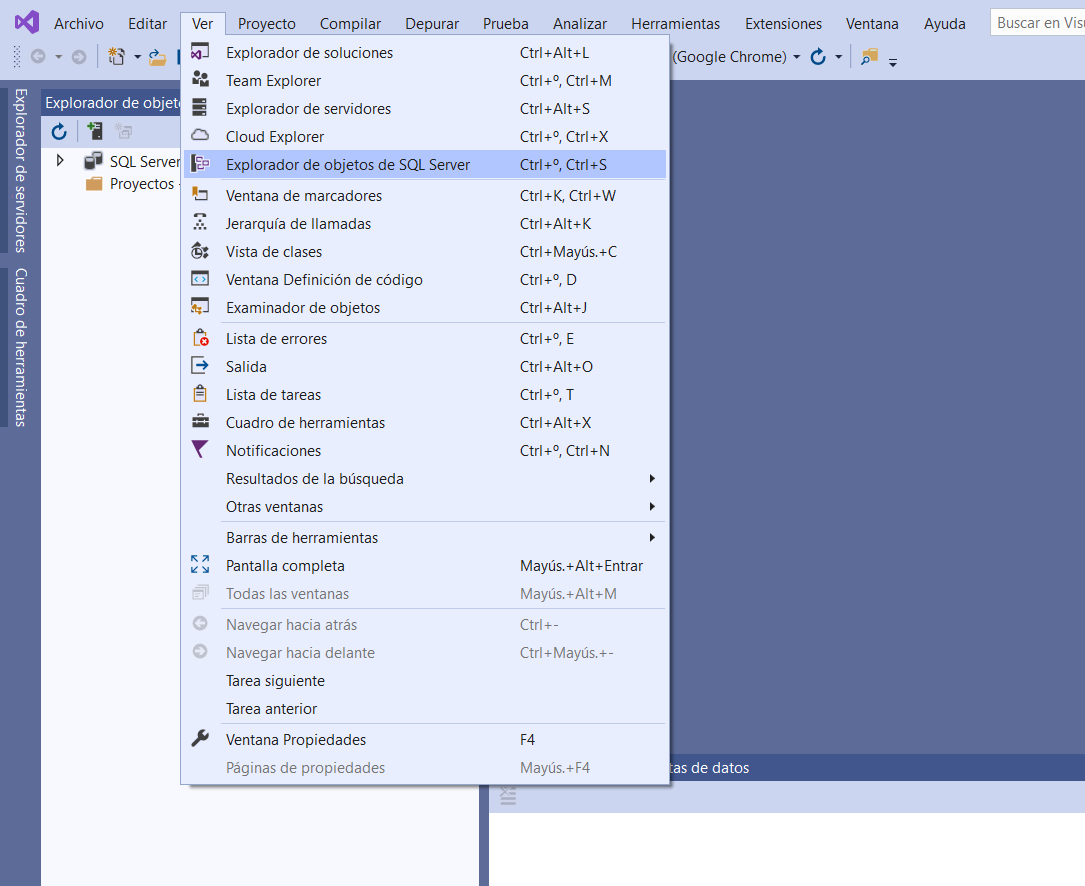
4.  Later, in creation window, you select the empty item, and enabling the MVC option, and click Create

.  

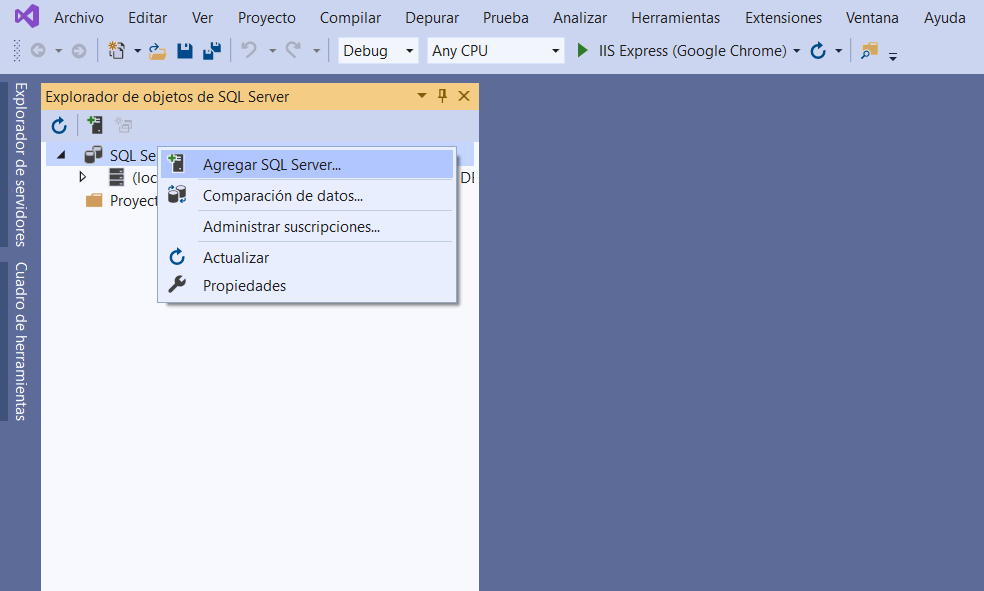
5. You wait a few minutes while the application finishes create.



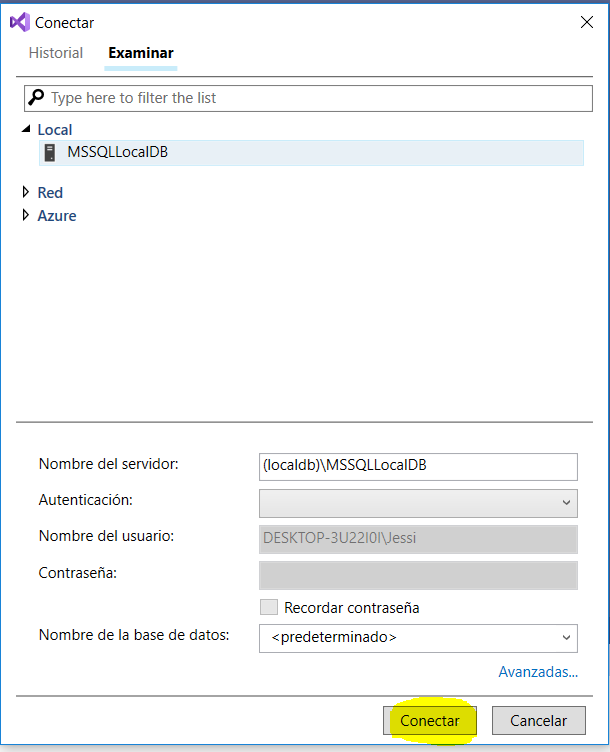
6. Once created, is to create and connect the database to the application. In the menu bar, the tab view, select Object Explorer of SQL Server.



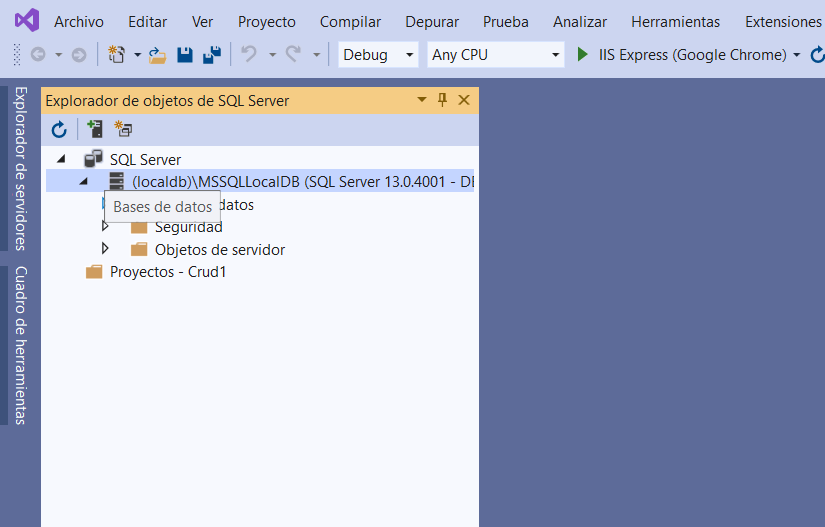
7. Right click on the SQL Server option, and click Add SQL Server



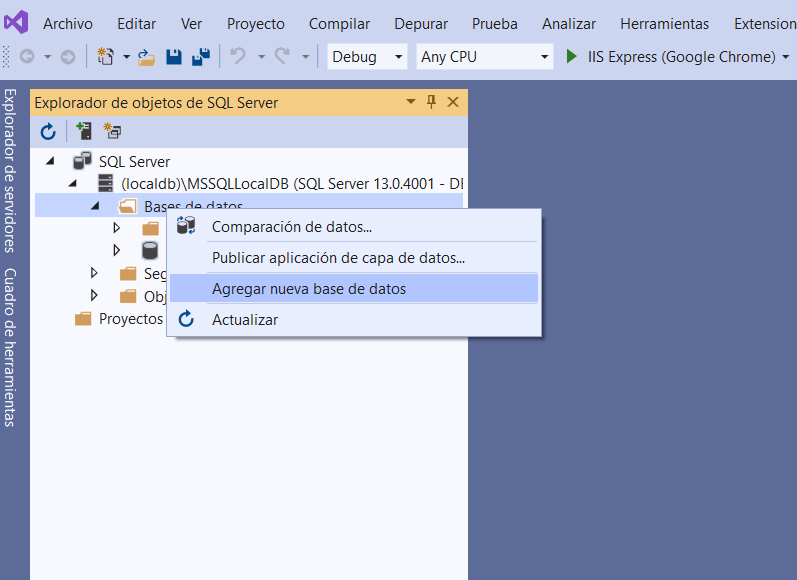
8. Appears a new window where there are three options, click the Local option, you select MSSQLLocalDB and click on connect



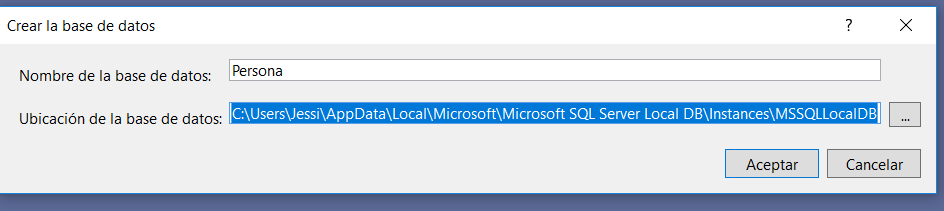
9. Click on the arrow on deployment of LocalDB



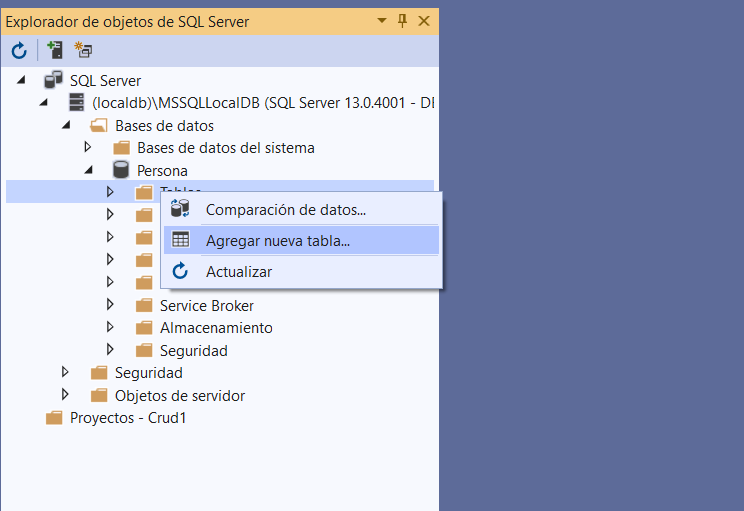
10. Right click databases, and click Add new database



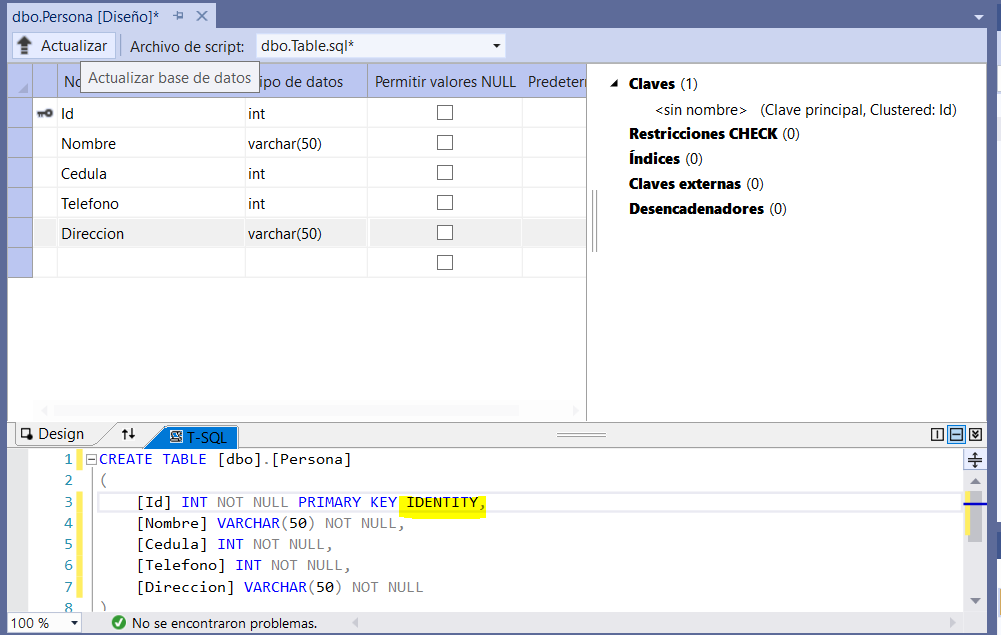
11. Writes the name of the database, in the text box the address appears where you will save the database, click accept



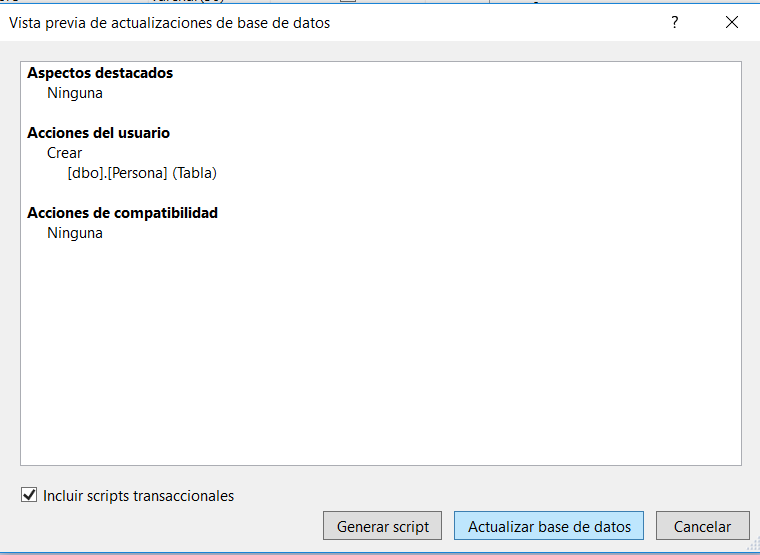
12.The new database, appears in SQL Server Object Explorer, click the arrow on deployment of this and right click in the option tables, select the option add new table.



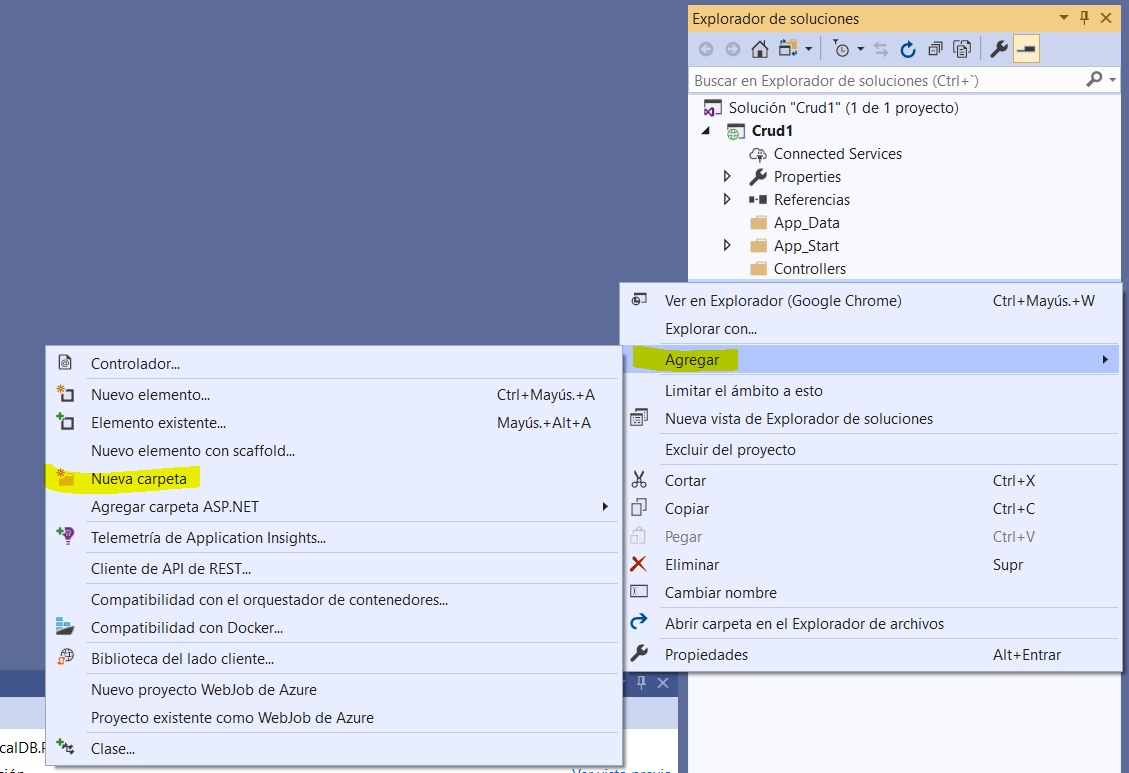
13.Creates the desired tables, with required fields, it should not be forgotten that the primary key must have the auto\_increment “IDENTITY”. And click on update



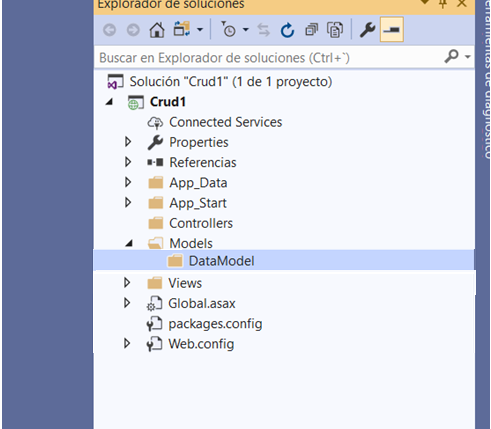
14. A verification window appear, click on update database



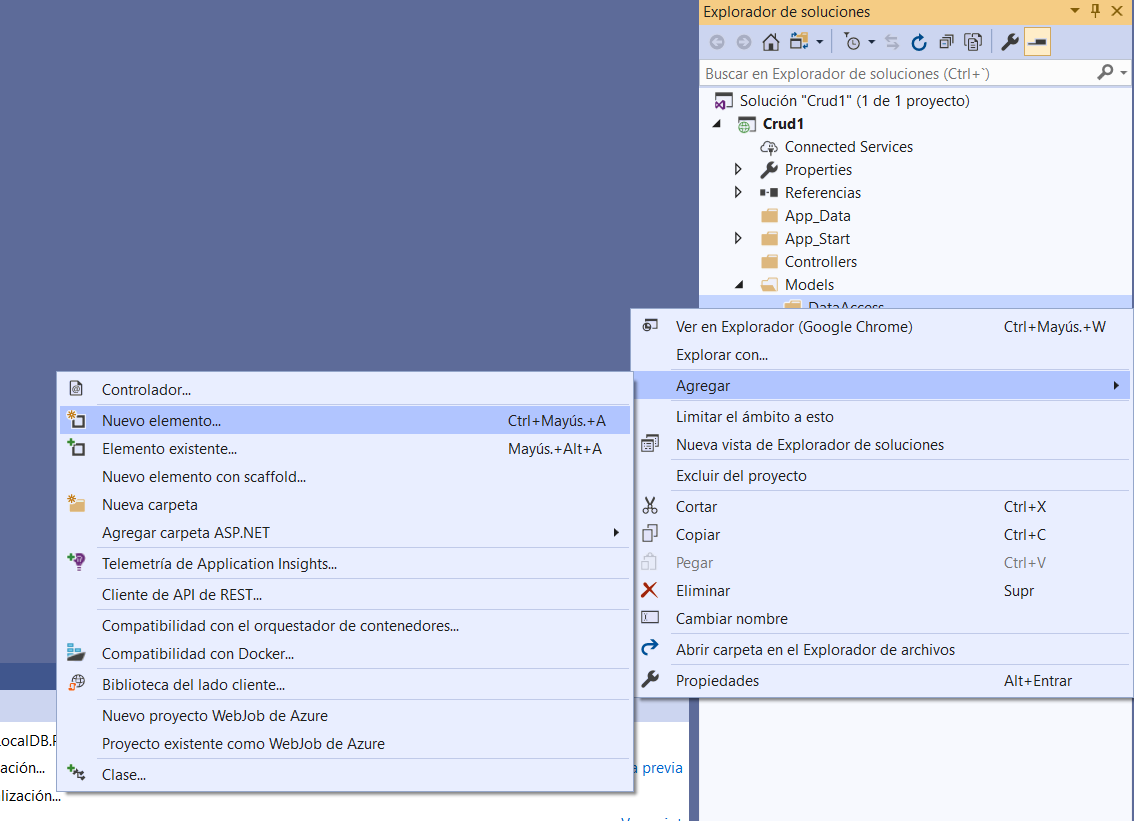
15. Once the database is created, it must be connected with the application, so you should go to the folder Models, right click on it and click Add and then click new folder



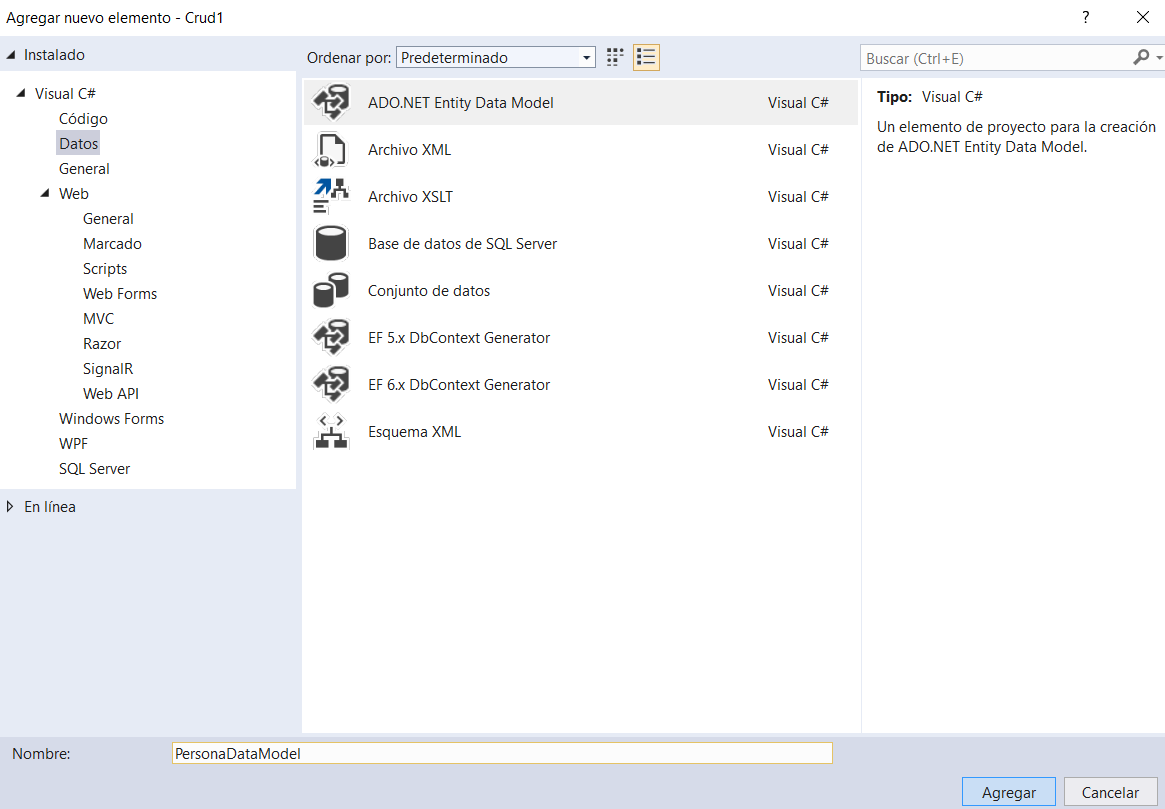
16. Change the name to the folder by DataModel, where the databases will be lodge



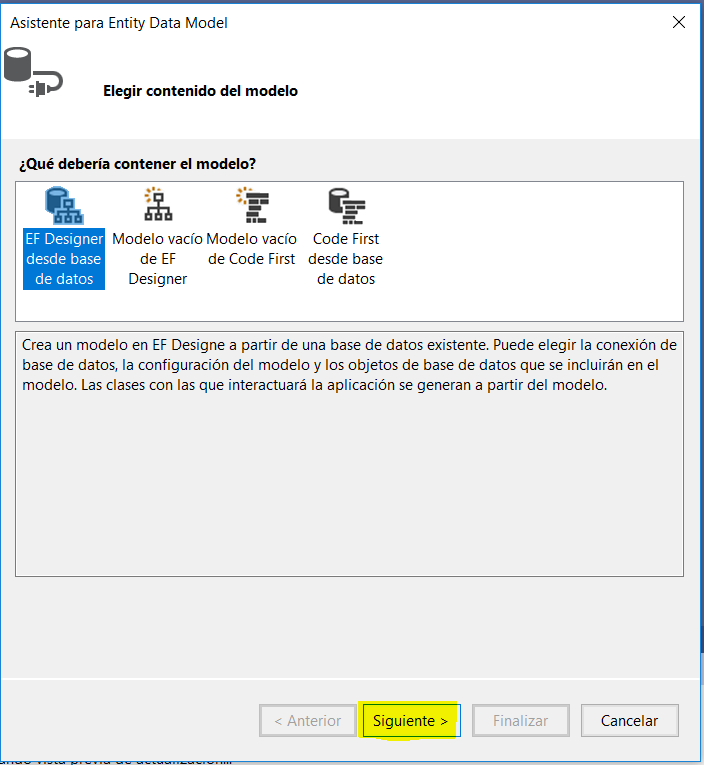
17. Right click on the folder DataModel, click Add, and click new item



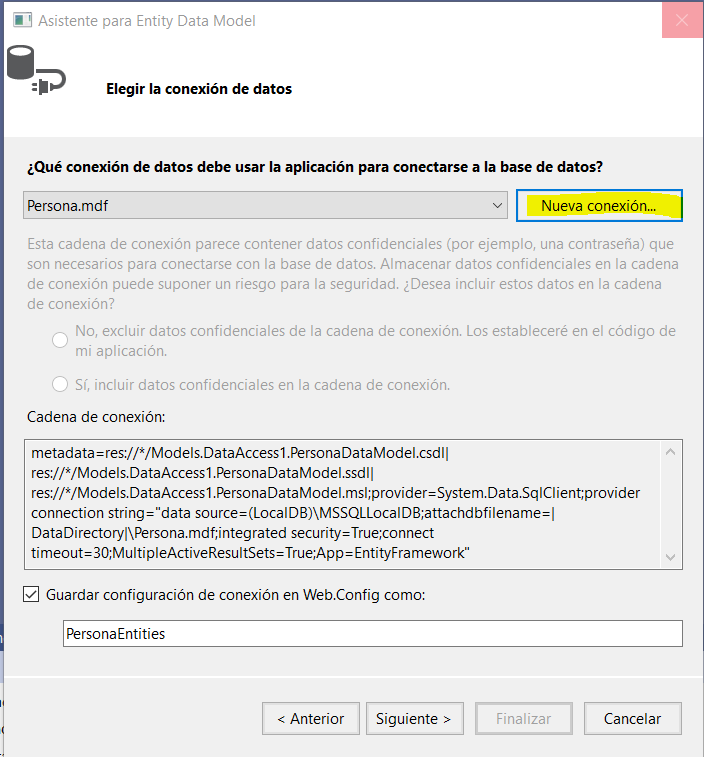
18. In the next window select the option data, and then option ADO.NET Entity Data Model, at the lower part is written the name that you want to give and click Add



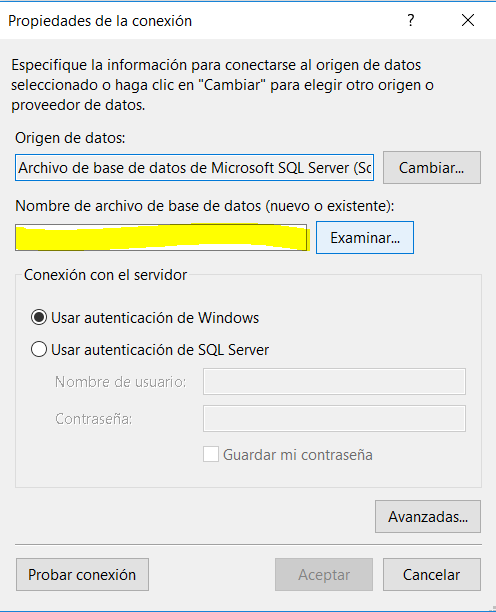
19. The option is left by default, and click on next



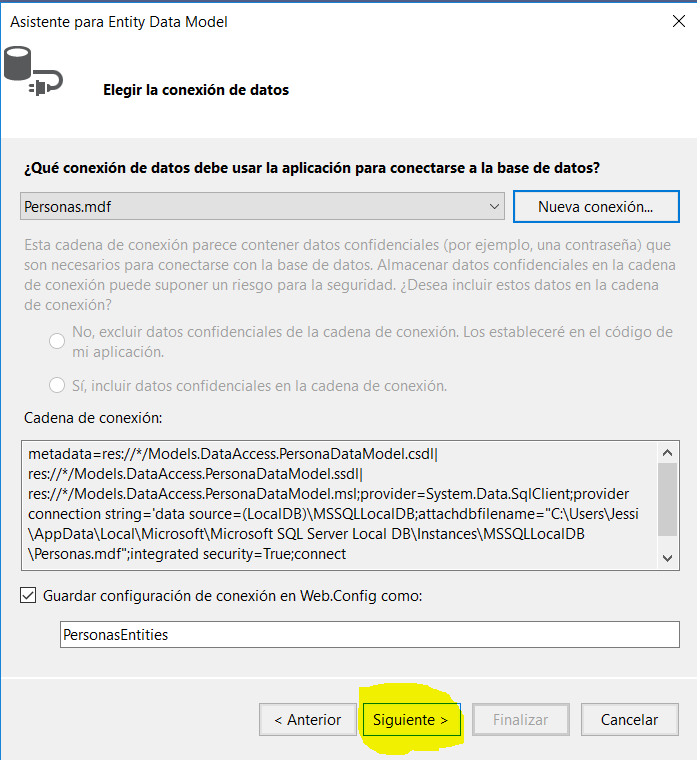
20.Click on new connection



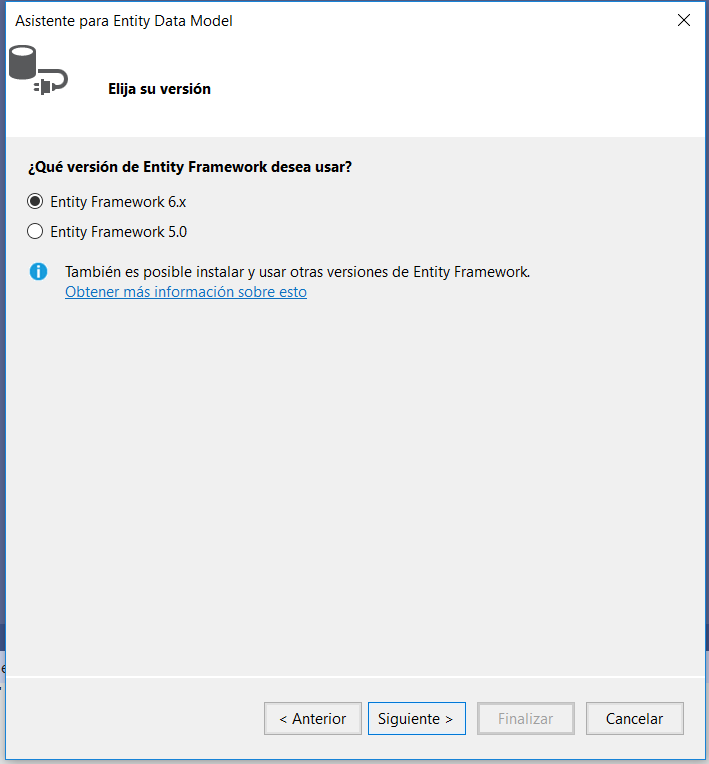
21.Click on browse, here is search the path where it is hosted on our computer database. And click Accept



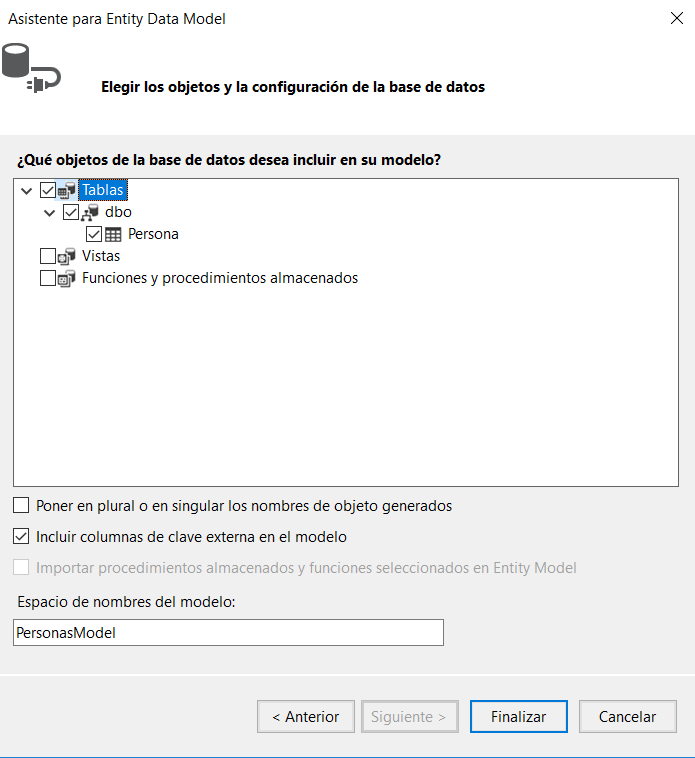
22. Once loaded the database, it will appear in the lower part the name of our migration, in this case PersonasEntities, and click on next.



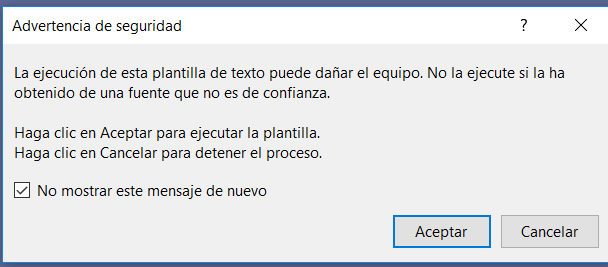
23. Choose the version of the Entity Framework and click on next



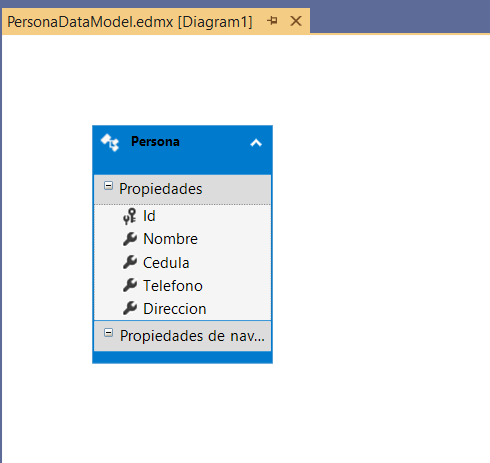
24. Finally, click on the arrow display of the tables option, and enable him or tables that are needed. Click on finish



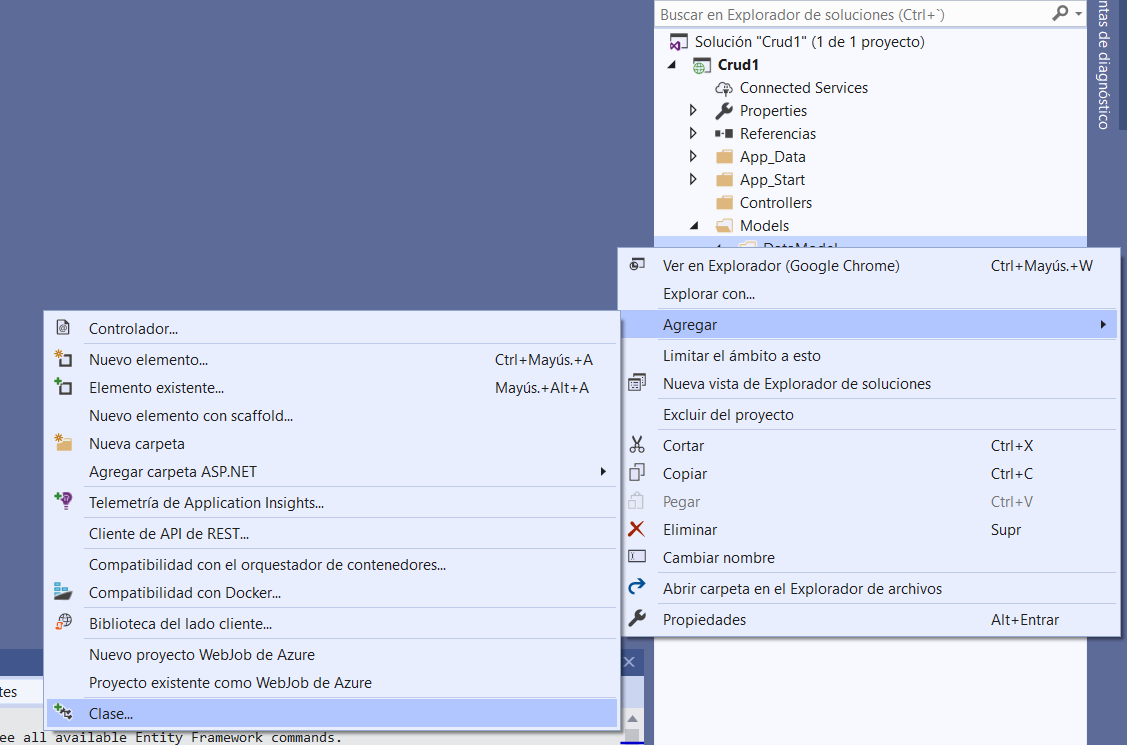
25. In the security warning, enable the box don't show this message again, and click Accept



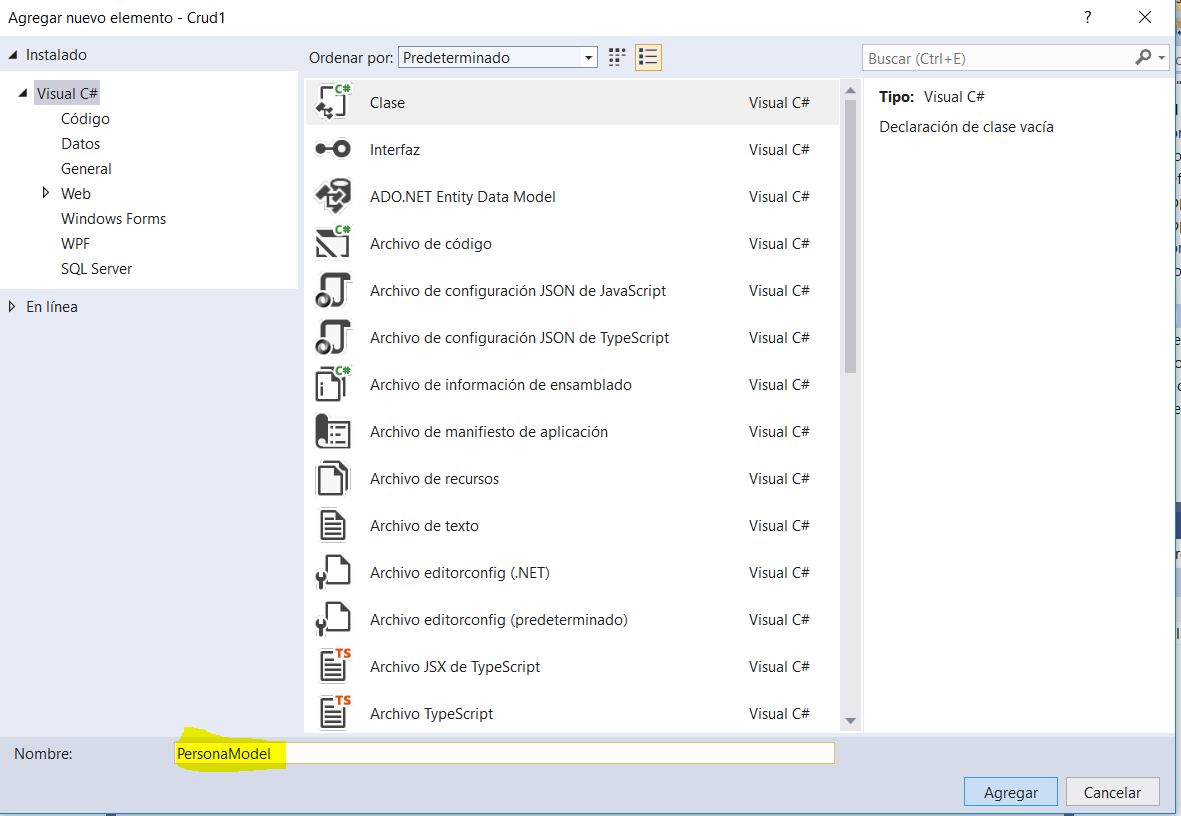
26. Migrated table you can see.



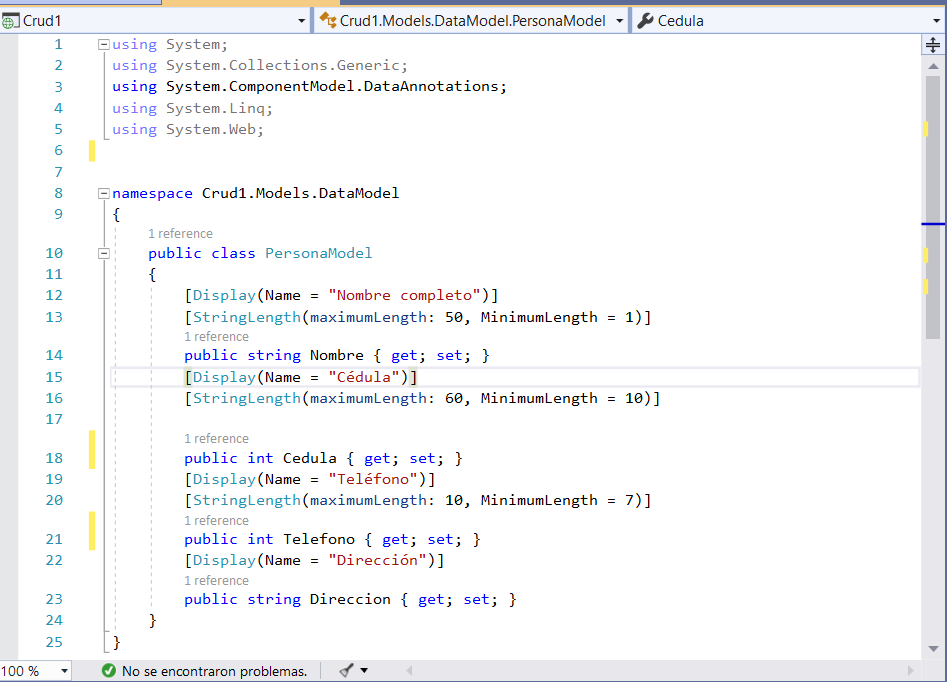
27. To create the model click on the option Models, click Add, and click on class.



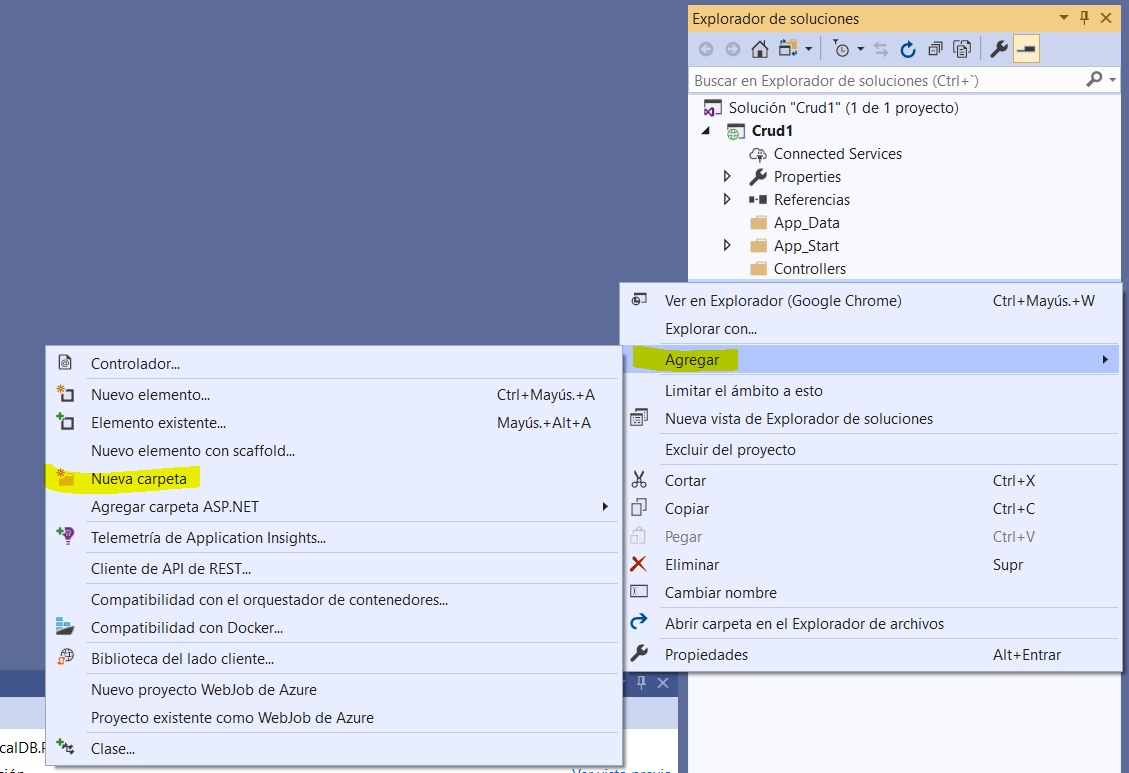
28. Class is selected and the name of it is given at the bottom of the current window.

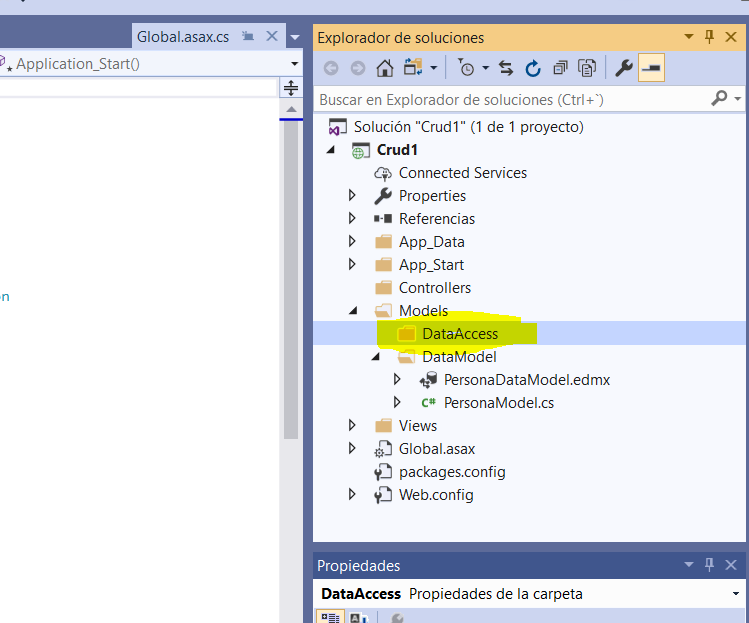


29. Creates Constructors of the same fields that you have in the database table, here some validations required

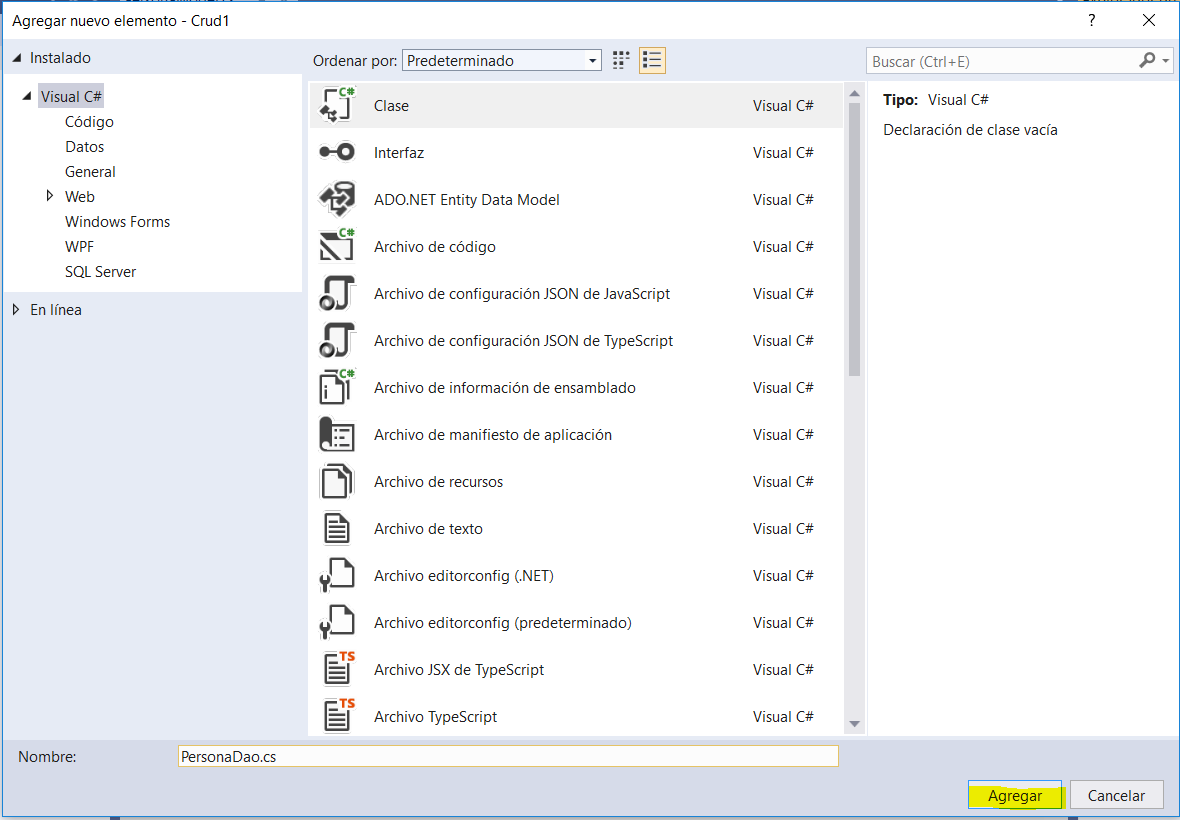


30. Creates a new folder called DataAccess, where will be the Dao’s, here are the methods that will be called the PersonaModel model.

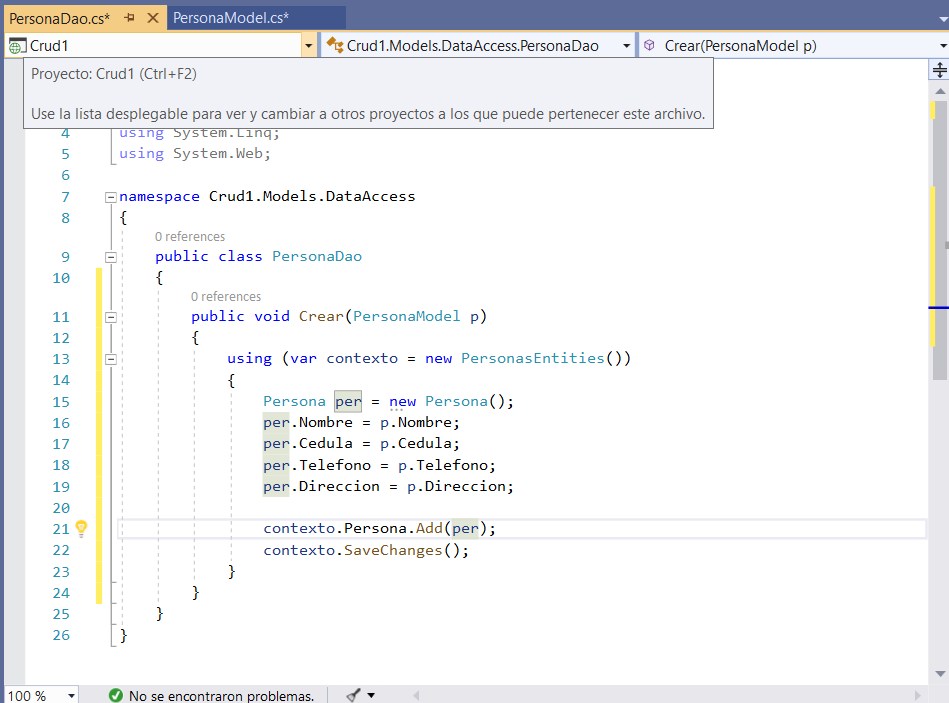




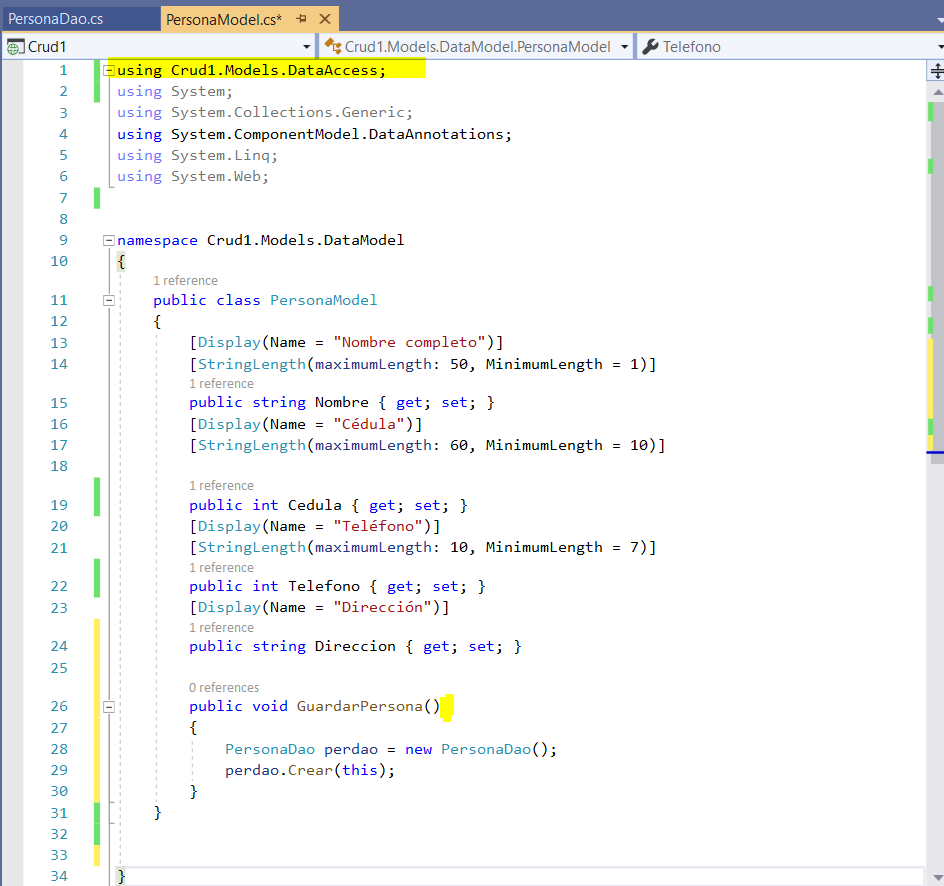
31. Inside the DataAccess folder, creates the PersonaDao class



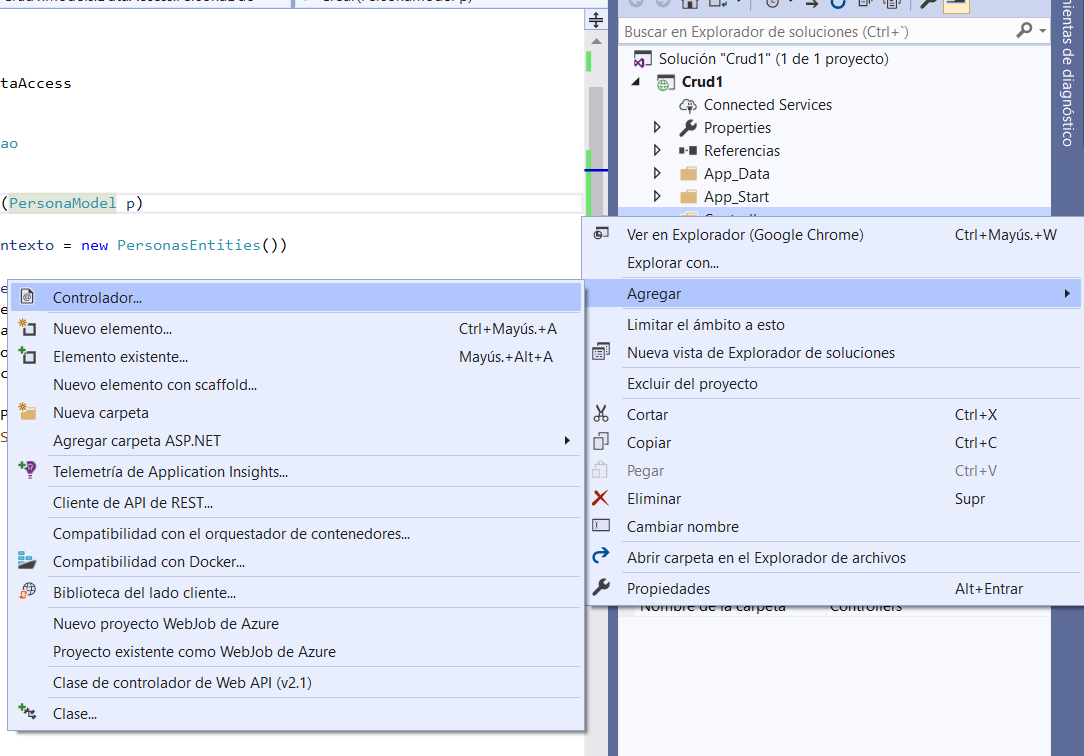
32. PersonaDao, is the method create, which starts bringing the data of the Entity in a variable untyped (Var) context, then will instance a variable type person to be able to bring data for data. These data are assigned to the variable context, and finally changes are saved.

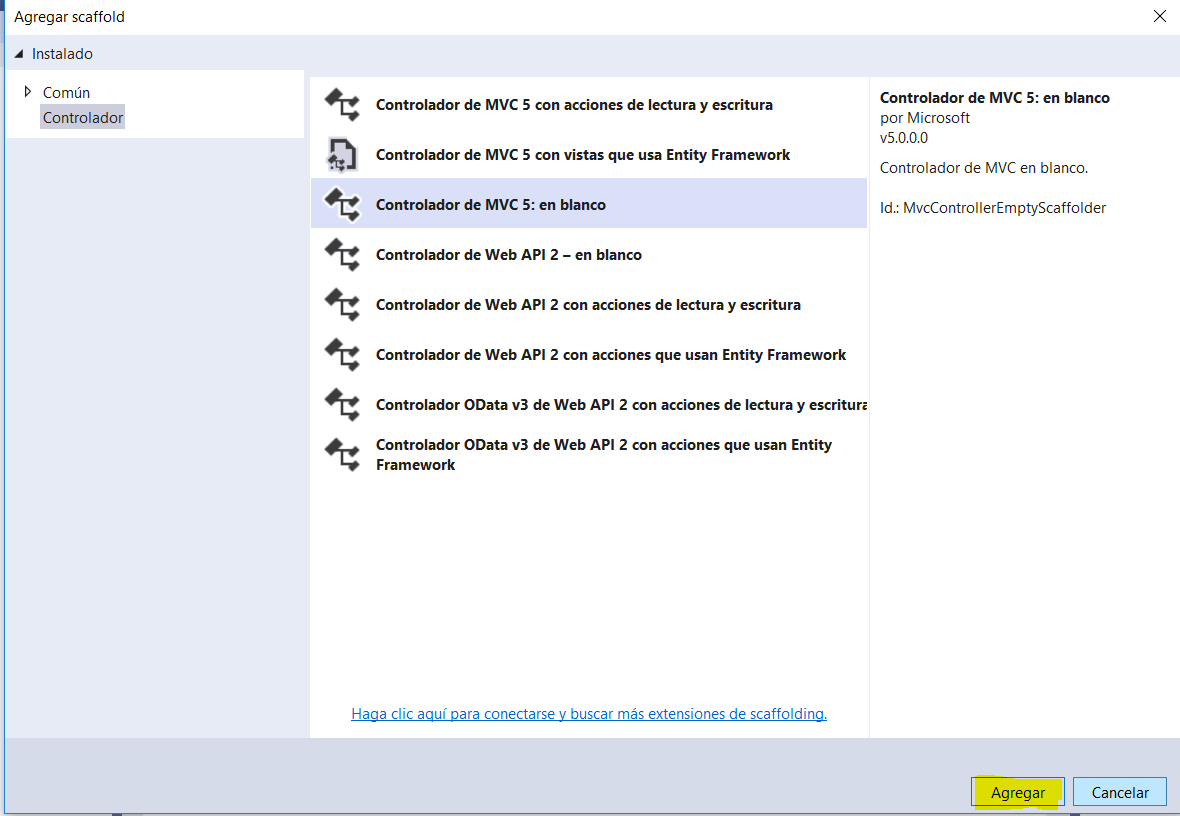


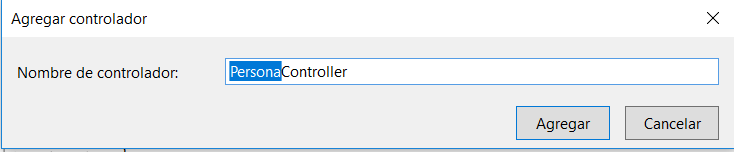
33. The PersonaModel model, creates the method that will be called when creating Dao



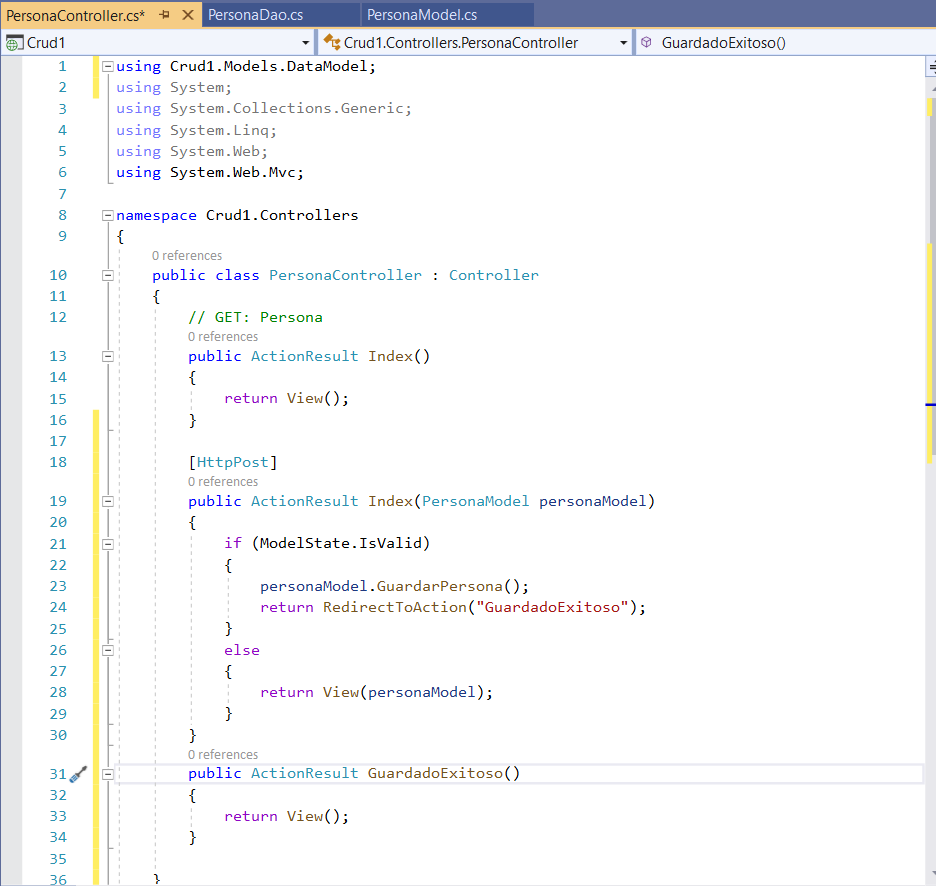
34. Now, the Controller, which should be MVC5 controller in white, with the name of PersonaController in this case is created.



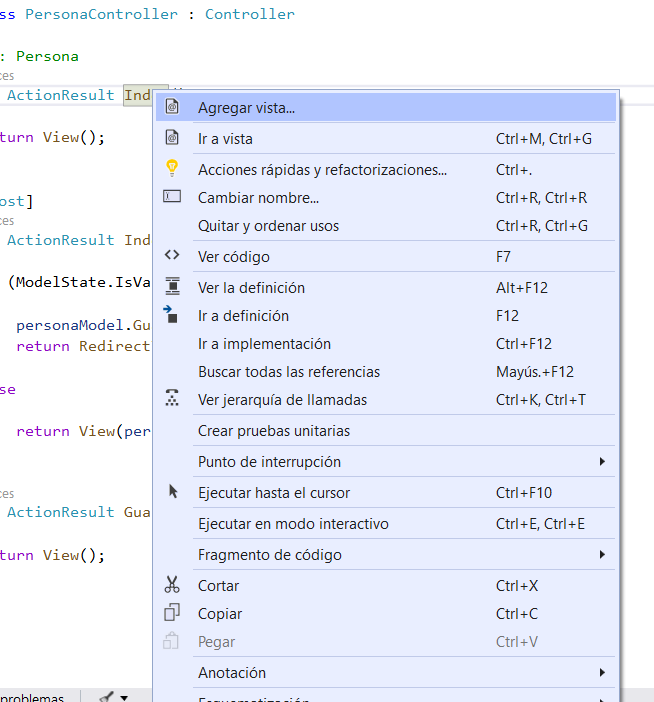


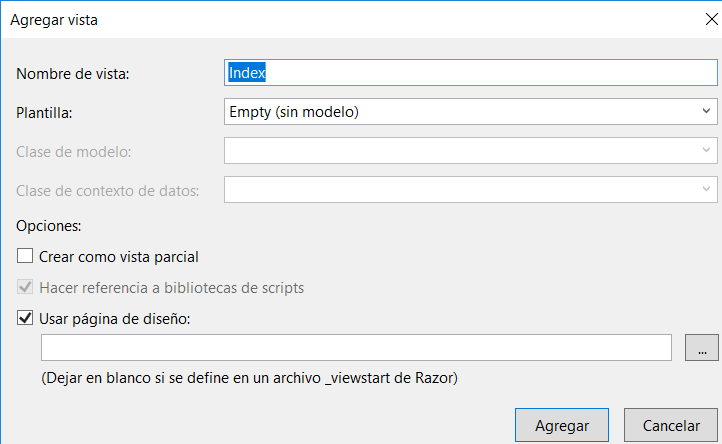


35. In the Controller, first is the section of code that shows the main view Index, this part comes by default, a validation of the model, is done if it is valid, the method GuardarPersona is called and is redirected to another view

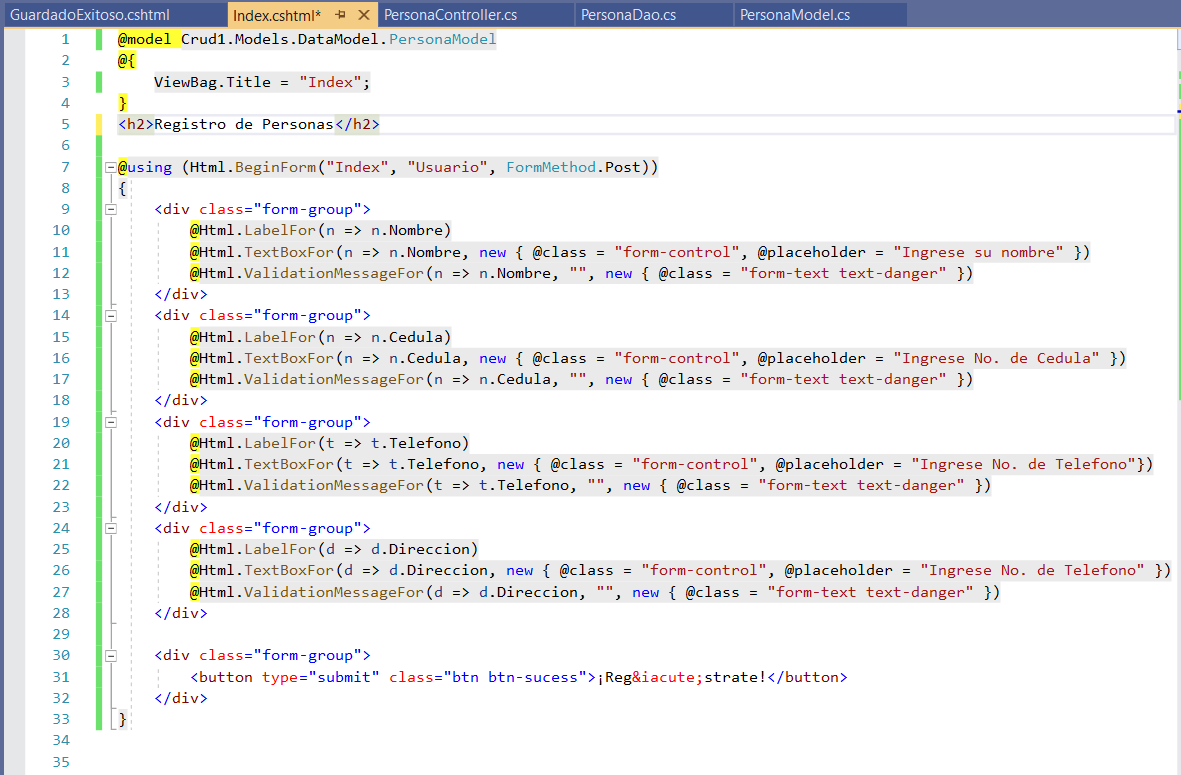


36. To add the views, right-click is given in the name of the Index in the driver section, left by default the name the view in this case will be called Index and click on add

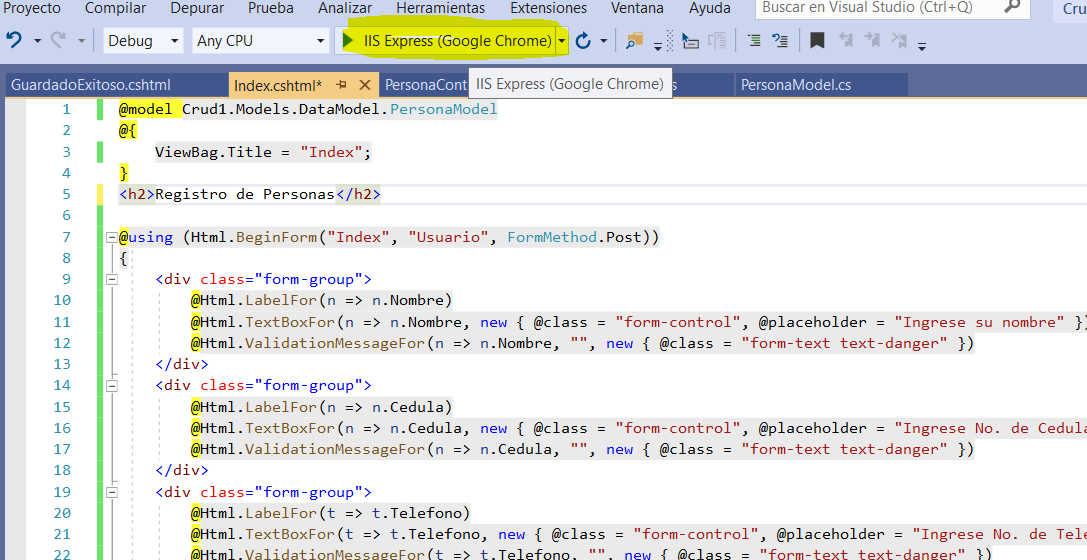


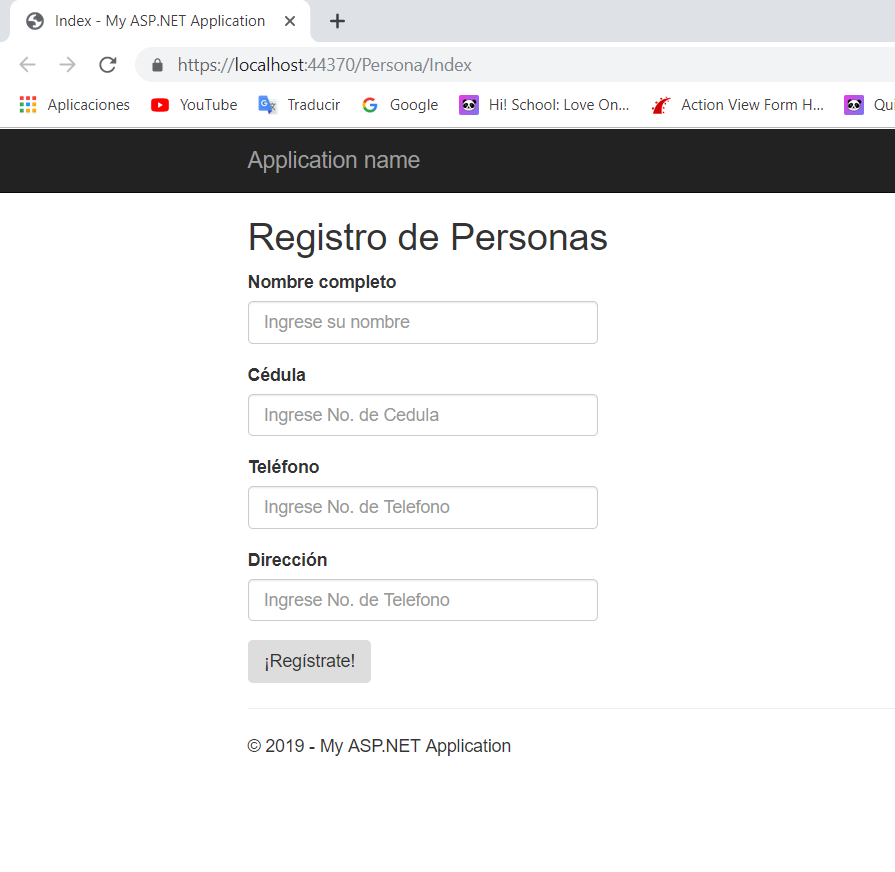


37. In Index view, a form is created using Razor. With the same fields that have the database, except for the Id.

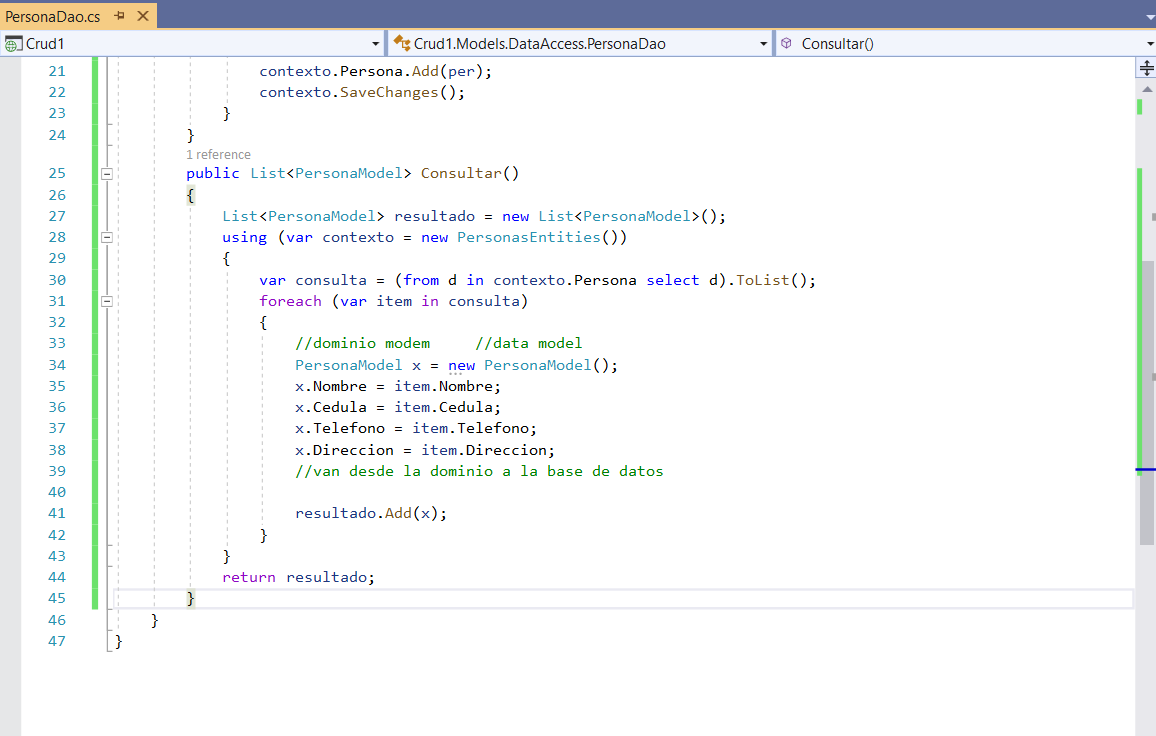


38. When compiling, effectively displayed the form that was created.

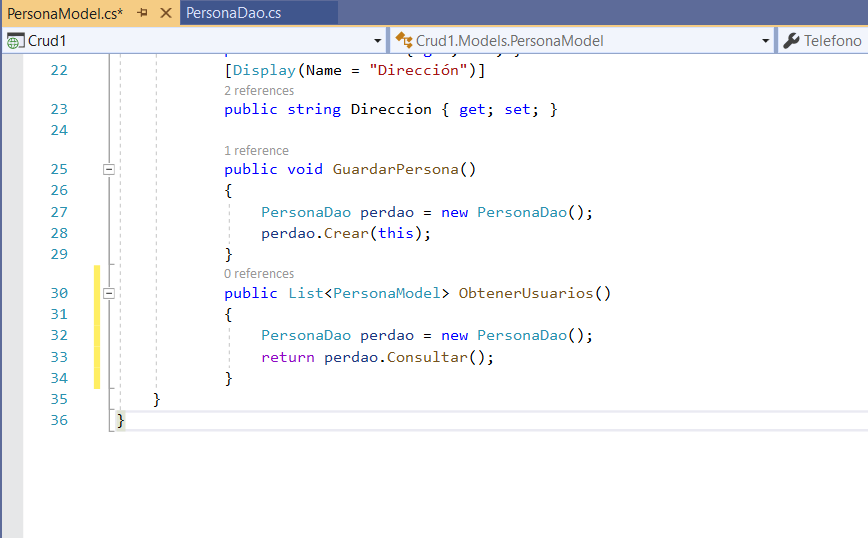




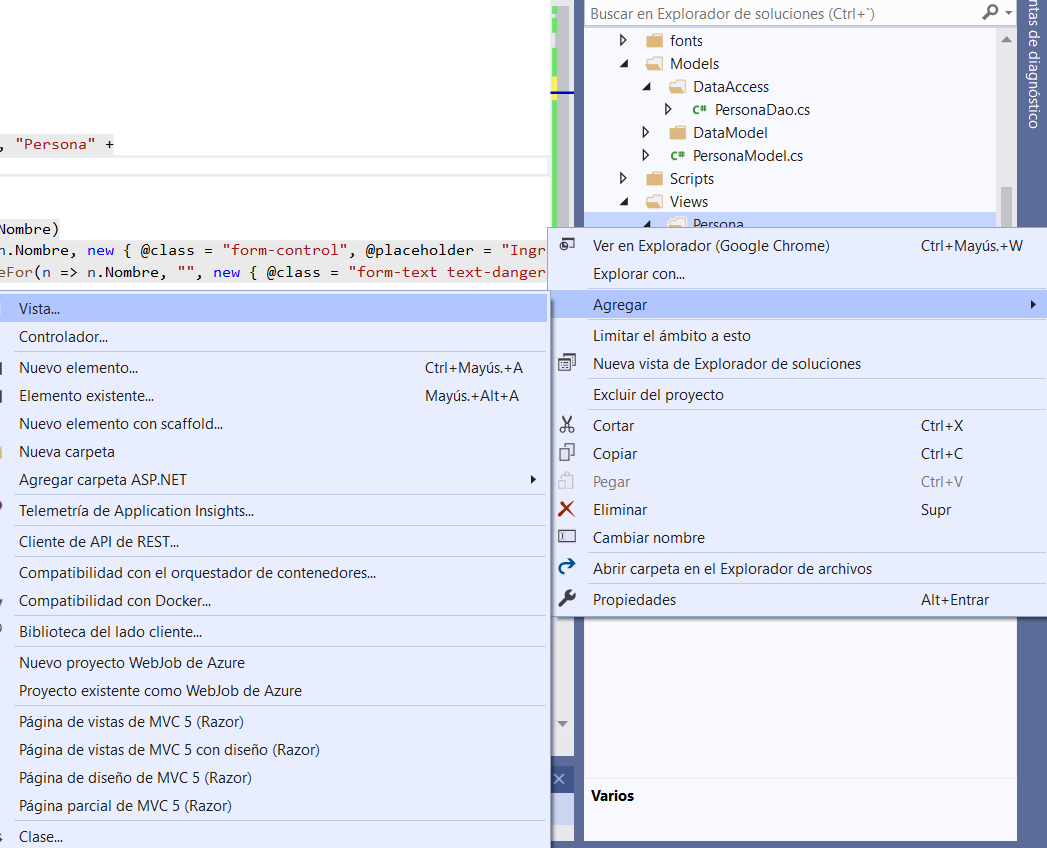
39. Creates a method of type list to make a query on the database



40. PersonaModel, creates the method that calls the enquery of the Dao

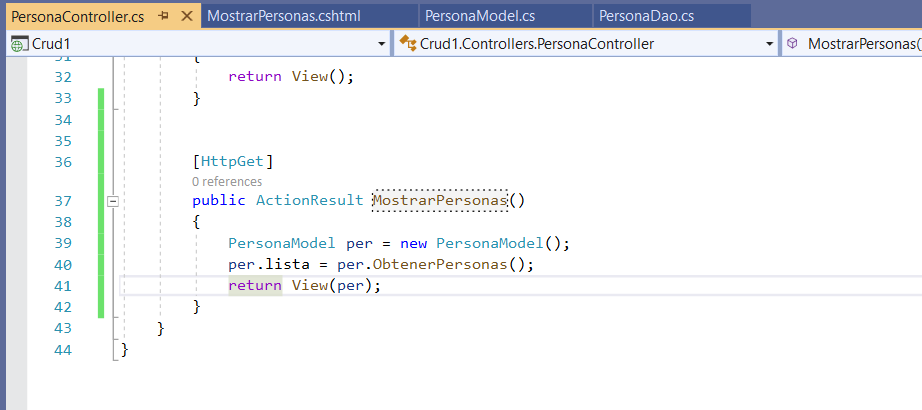


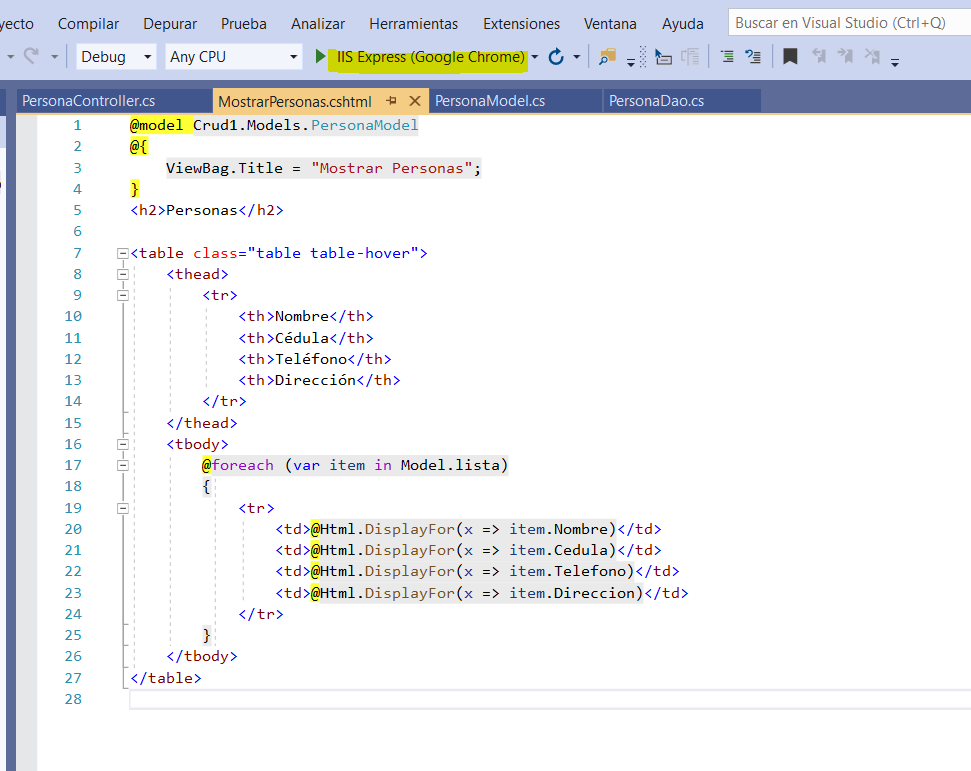
41. Adds a new view called MostrarPersonas, here a table will be created, where they are and are called fields and data that has our database

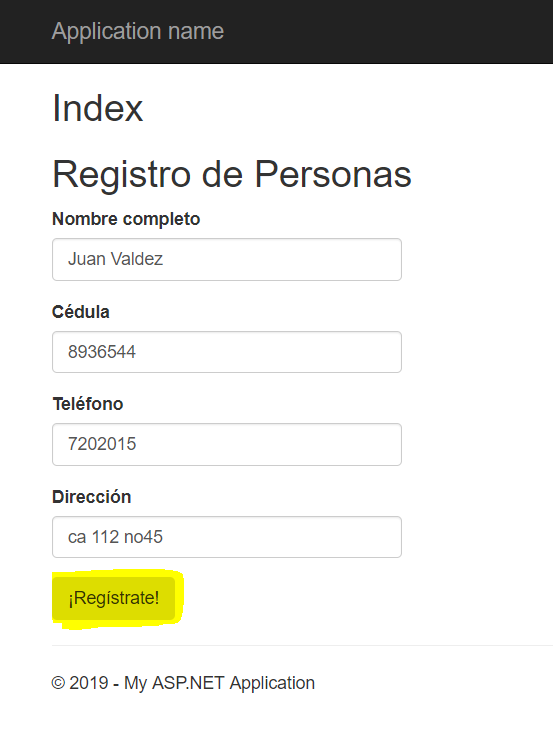


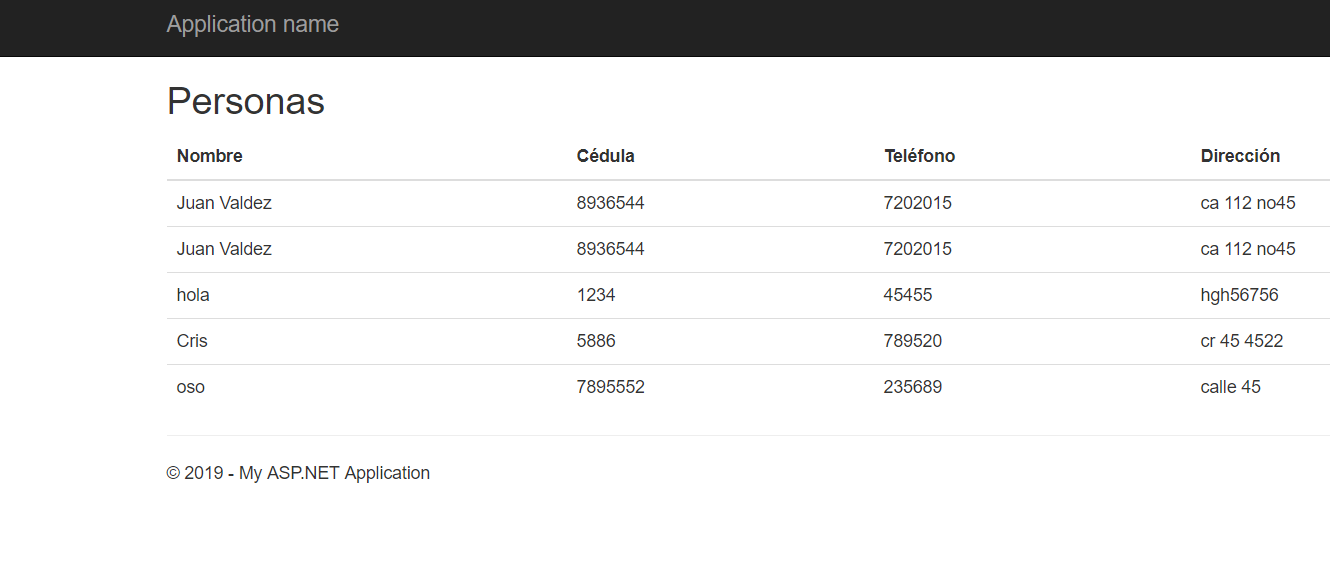


42. By filling out the form of the Index, and click on register, effectively it redirects to the view MostrarPersonas

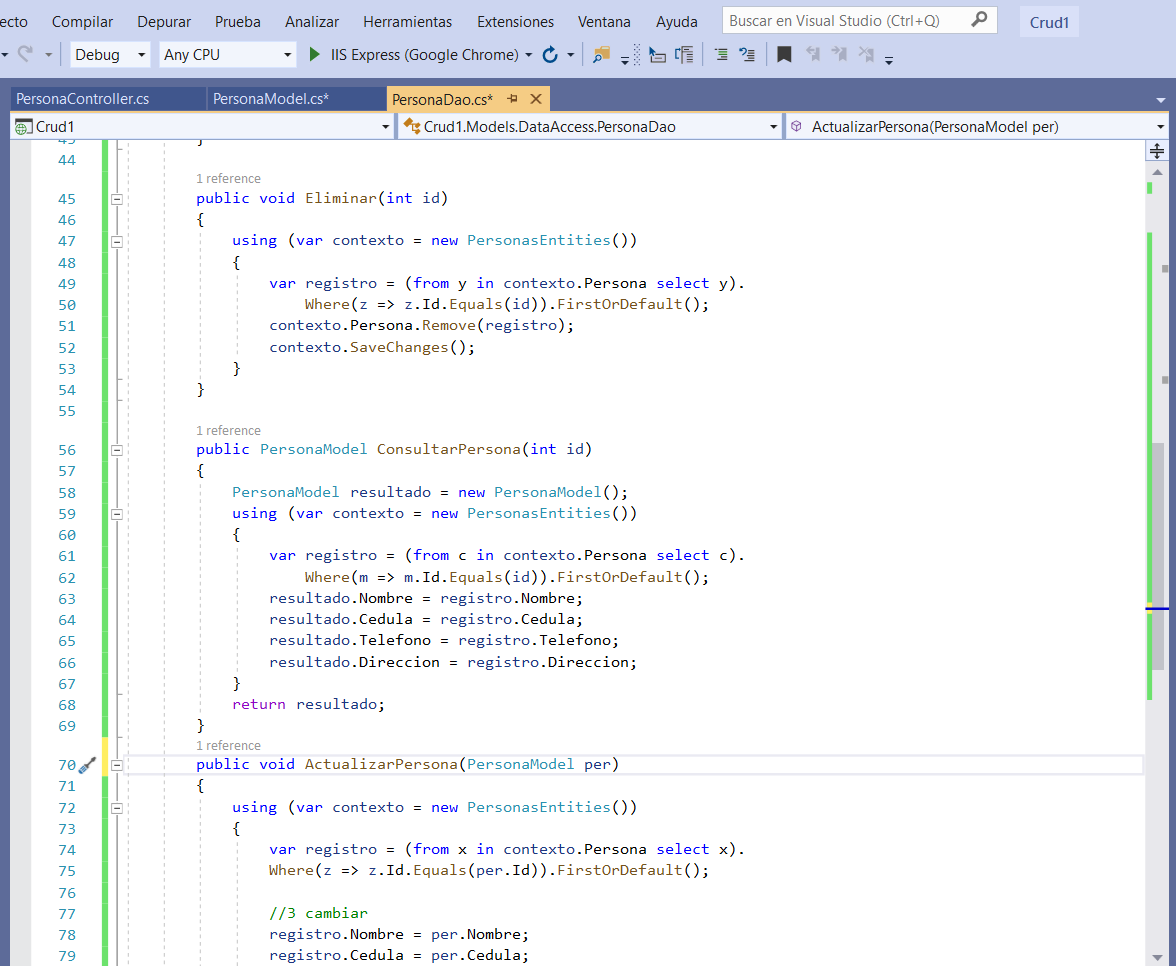


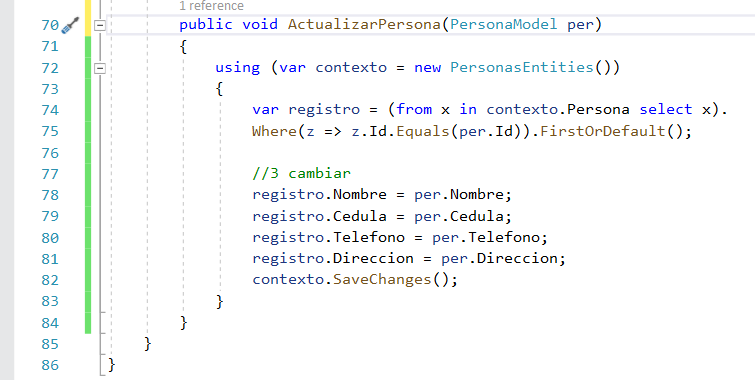


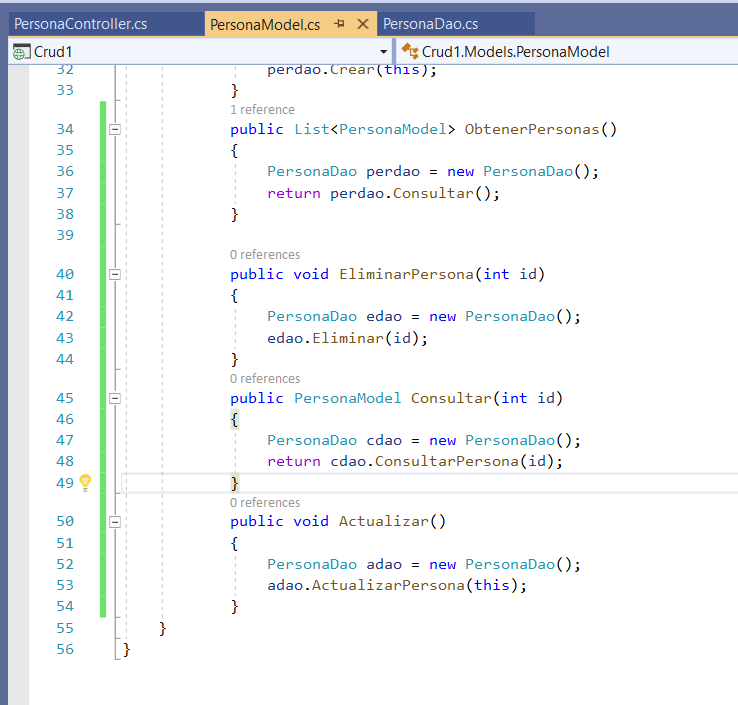


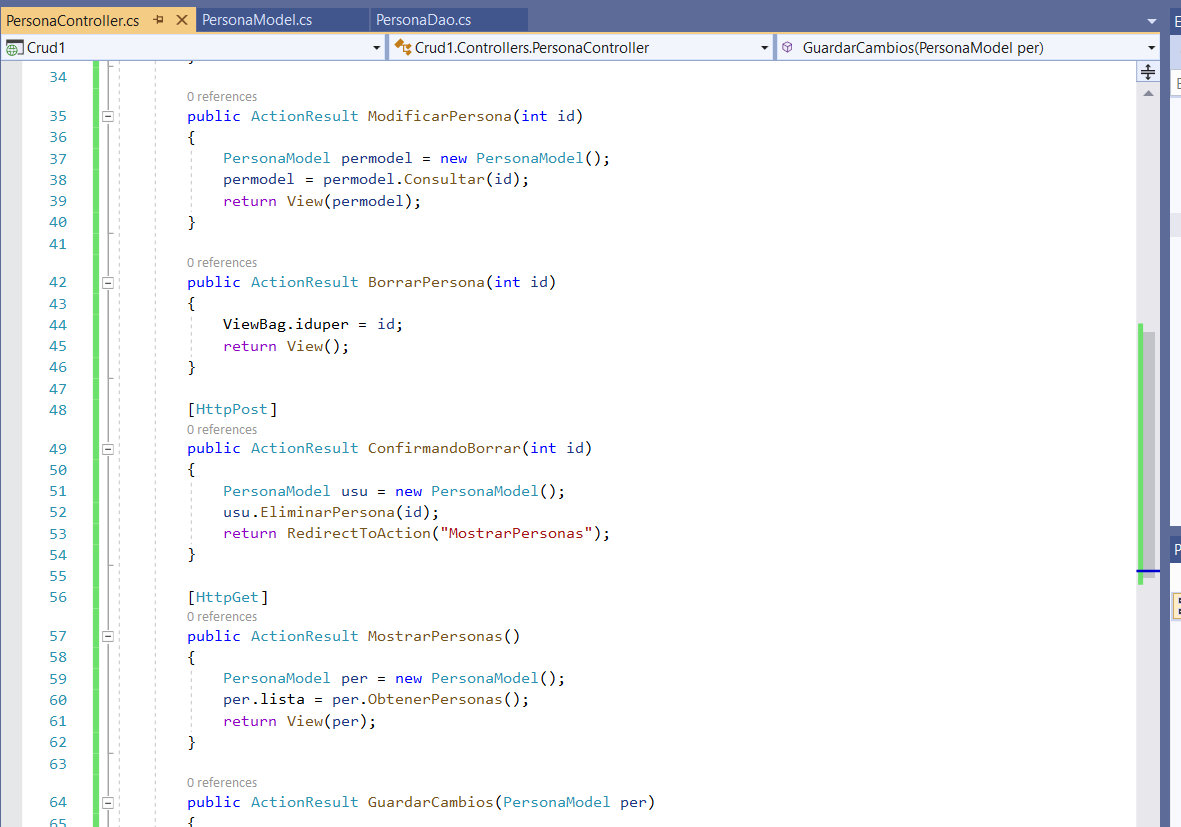


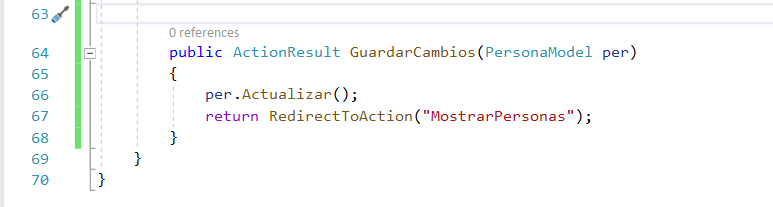
43. The same thing with other methods, is as delete and modify, taking into account that the method that contains the logic is in the DataAccess, PersonaModel model called, and the driver with the help of the view executes it and show it in screen.



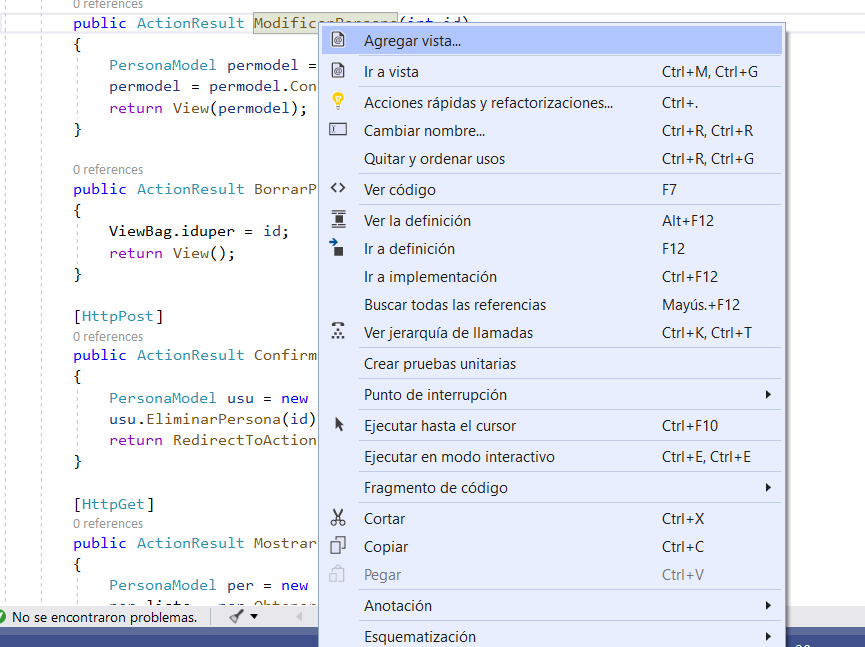


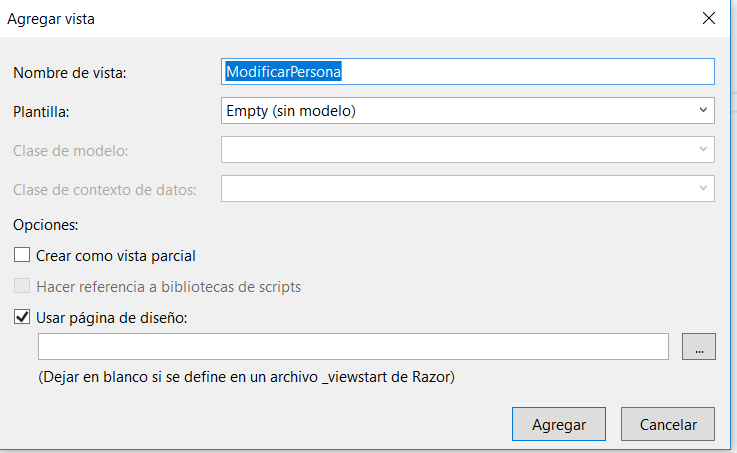


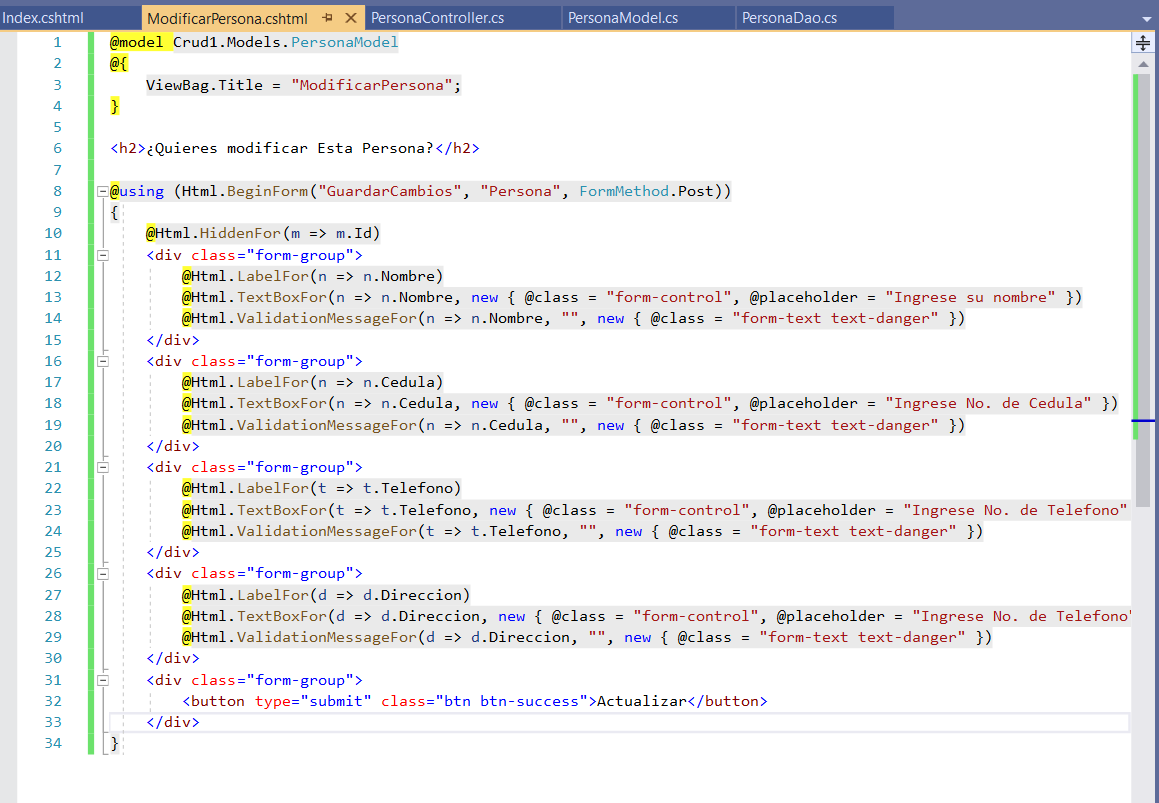




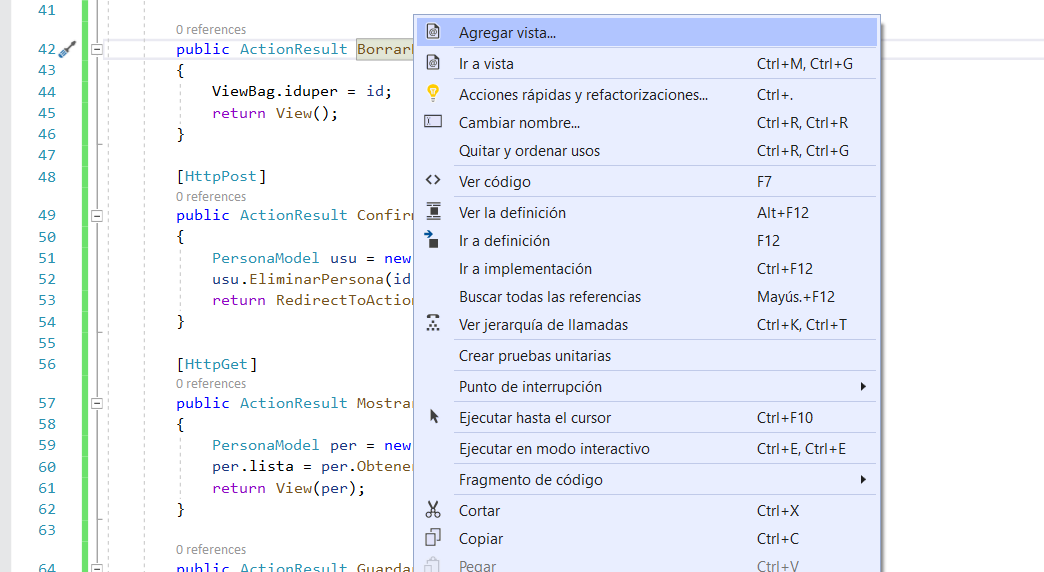
44. Adds a view called ModificarPersona, which is the same form of the Index

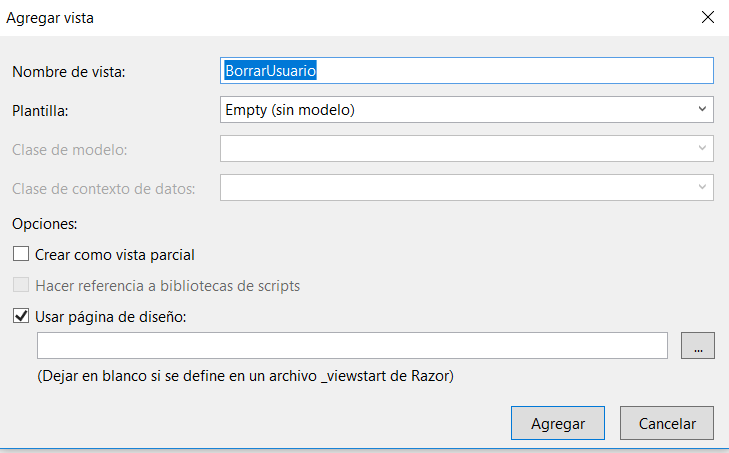


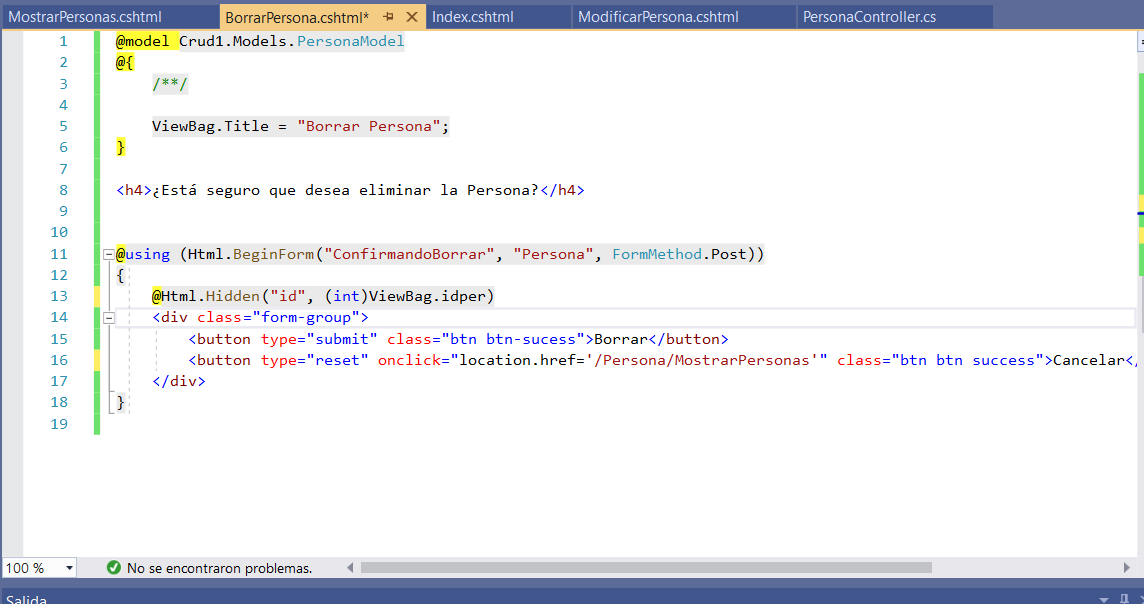




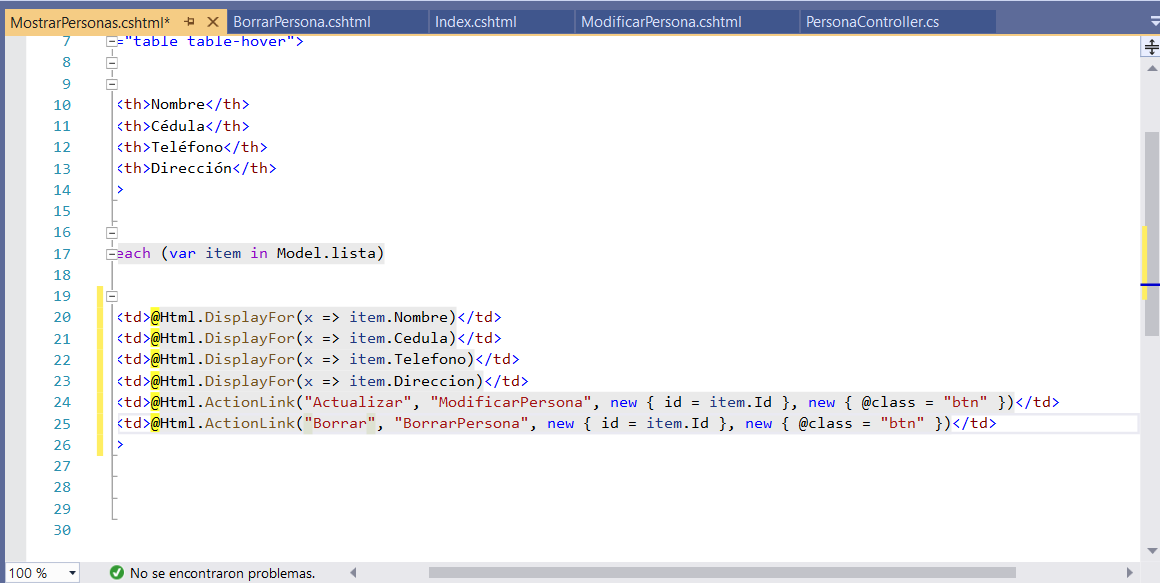
45. Also added a view called BorrarPersona, where is a verification for the deletion to be effective.



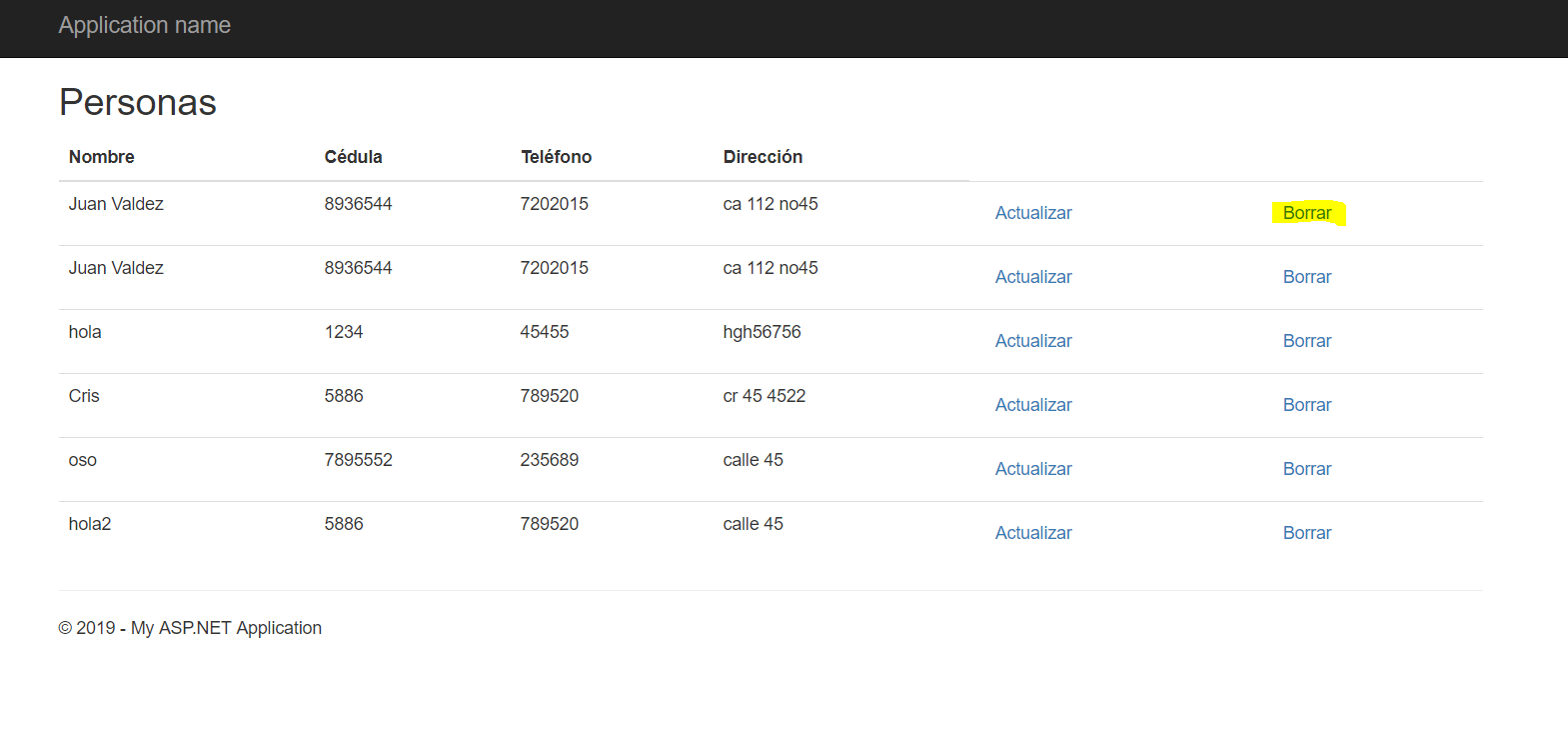


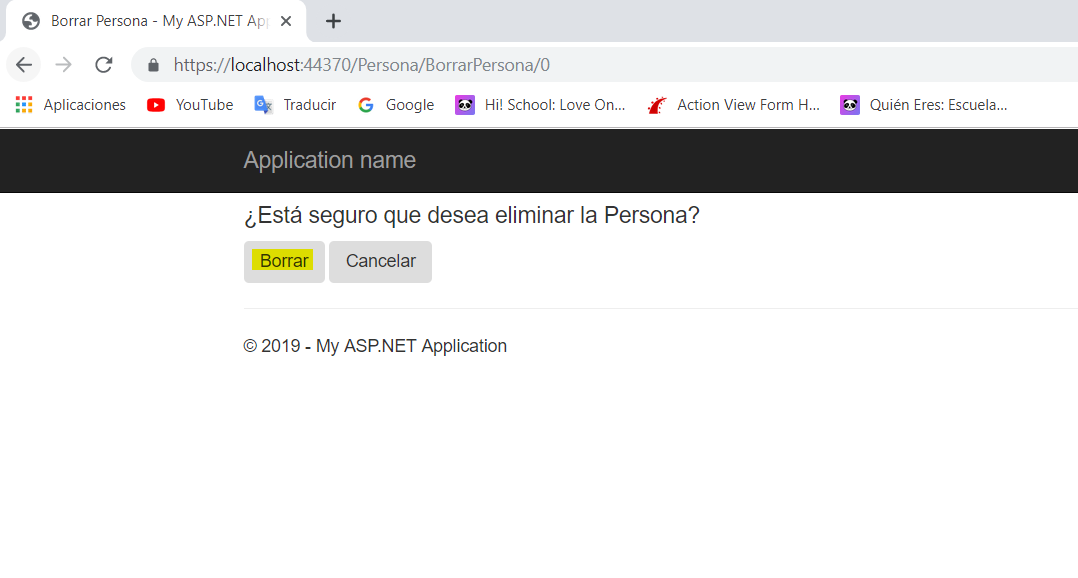


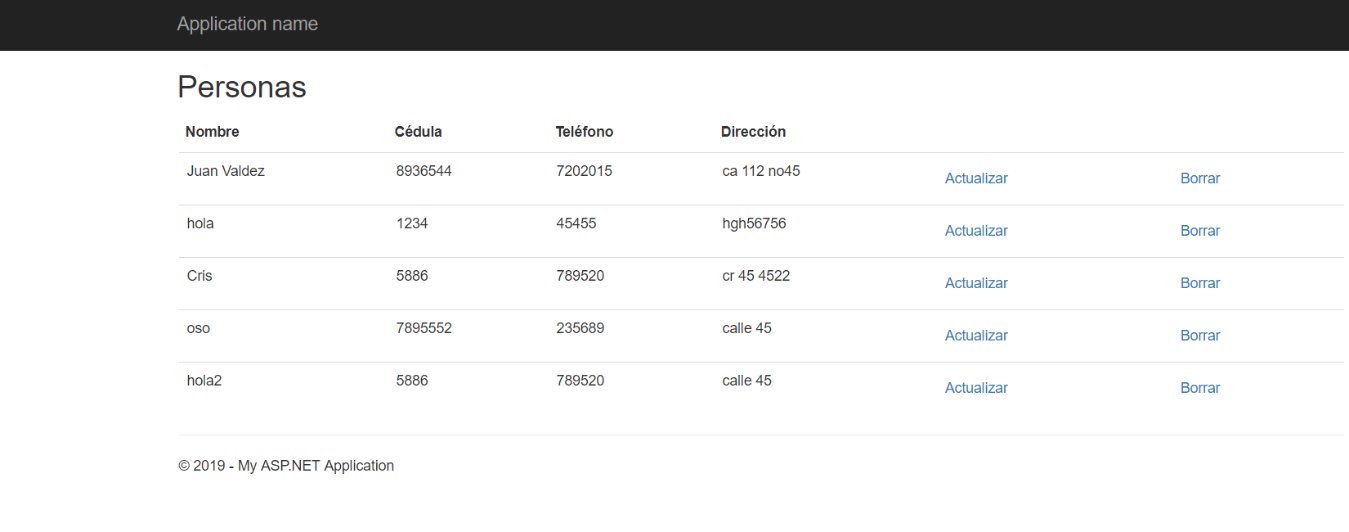
46. In the view show people added two lines of code, so the window shows two hyperlinks that allow us to do the delete and update actions.



47. When you click the option delete, get the verification message, and then the file was removed from our database.







Works in the same way if you want to do another operation.

Thank you

BY JESSICA CHILES