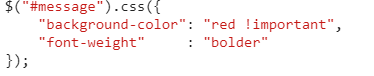
**%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%**

Explique el funcionamiento de los siguientes estilos CSS:

|  |  |
| --- | --- |
|  | R//  Este estilo css tiene un ancla y tiene su valor por defecto,!important da prioridad y se antepone al resto de reglas |
|  | R//  Este estilo de css lo que nos dice es lo siguiente : hay un contenedor cuyo tamaño máximo es de 500 px , el tamaño debe ser menor o igual que 500 px , hay un id llamado wrapper que es quien contiene el contenedor , el screen hace alusión al diseño responsivo |
|  | R//  Tenemos la clase Titulo Principal , esta tiene el tipo de letra sans –serif y tiene un color determinado , el texto mediante la función uppercase permite convertirlo a mayúscula , el tamaño de la letra es 24px , y esta alineado a 30 px a lo alto,diseño del titulo principal,tiene fuente de 400, ,!important da prioridad y se antepone al resto de reglas |
|  | R//  Este estilo de css tiene un alto de 73px , tiene posición por defecto , tiene un 90% del contenido , el valor del ancho es por defecto , el fondo es transparente |

**%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%**

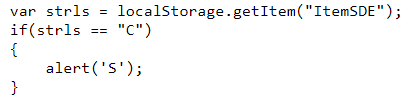
Explique el funcionamiento de los siguientes scripts:



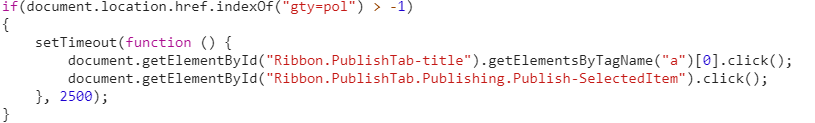
R// Este script lanza un mensaje cuyo color de fondo es de color rojo , y el contenido o texto ocupa el ancho de la caja , se están manejando los estilos css desde Javascript



R// La etiqueta h3 tiene un tamaño medio en html a modo de titulo , lo que hace este script es asignar un nombre a dicho elemento y luego lo cierra u oculta con el hide



R// Se declara la variable strls a la cual se le agrega un valor , la condición dice que si la variable stlrls es idéntica por la notación “==”, idéntica al valor string o texto “C” , si cumple esta condición debe mostrar una alerta o mensaje que diga “S”



R// Este srcript lo que hace es que se ejecuta en un tiempo de 2500 milisengundos una vez se dispara el evento , se agrega mediante un arreglo o vector un valor , en este caso “a” una vez se de el evento click.

**%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%**

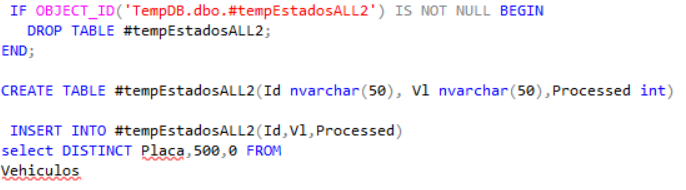
**Explique qué tipos de archivos contienen las siguientes extensiones y en qué casos se**

**Utilizan:**

|  |  |
| --- | --- |
| .xsn | **Servicios de red** |
| **.aspx** | **Esta extensión de archivo se genera cuando estamos trabajando en un proyecto de ASP.Net** |
| **.swf** | **Este archivo se puede reproducir en el navegador y ver e Dreamweaver,pero no editar en flash** |
| **.css** | **Esta extensión de archivo nos indica que dicho archivo es de estilos dentro del proyecto y es quien define los colores,tamaños , formas , tipos de letra etc.** |
| **.js** | **Este tipo de archivo es de Javascript , quiere decir que es un archivo que contiene información dinámica del sitio web** |
| **.bak** | **Este tipo de archivo por lotes , se genera cuando hacemos un backup de nuestra información de base de datos , es una forma en la cual podemos restablecer la BD con un script** |
| **.ascx** |  |
| **.config** | **Este tipo de archivo en un proyecto lo que nos esta diciendo es que tiene las propiedades y configuraciones por defecto del sistema , conexiones a bd etc.** |
| **.wsp** |  |
| **.xsl** | **Lenguaje para expresar hojas de estilo, significa extensible Style Language** |
| **.dll** | **Este tipo de archivo tiene la facultad de modificar propiedades del sistema operativo , tiene propiedades e instrucciones de modificación** |
| **.master** | **Este tipo de archivo lo que nos esta diciendo es que posee atributos generales** |
| **.ashx** |  |
| **.resx** | **Es donde se almacenan los recursos incrustados del formulario, como imágenes etc.** |
| **.cs** | **Este tipo de archivo nos dice que es una clase, generalmente se crea cuando estamos definiendo las clases.** |
| **.svg** | **Este tipo de archivo nos dice que es un tipo de archivo imagen** |
| **.svc** | **Conexión virtual entre dos extremos** |
| **.json** | **Este tipo de archivo contiene información plana del proyecto** |
| **.snk** |  |
| **.sql** | **Este tipo de archivo es el que genera el SMBD cuando realizamos una copia de respaldo de nuestra base de datos-** |
| **.xml** | **Este tipo de archivo almacena información del proyecto y puede ser abierto en un navegador web.** |

**%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%**

En SQL, Explique la siguiente instrucción:



R// Esta instrucción elimina una tabla llamada tempEstadosALL2 , luego la crea y le pasa 3 parámetros , estos parametros poseen tipos de datos string y numéricos , en este orden Id es tipo Nvarchar de tamaño máximo 50 caracteres , V1 tambien es nvarchar con las mismas características y el ultimo atributo es Processed de tipo entedero , quien por defecto es de tamaño 10 caracteres …luego se hace un insert a la tabla pero muestra un error resaltado ya que no están bien definidos los valores , por ejemplo : tanto Id como V1 con nvarchar , estos deben encerrarse en comillas pues Sql server no los reconoco o asocia con el tipo de datos establecido.

**%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%**

Debe crear un proyecto de aplicación ASP.NET usando el lenguaje de programación C#, debe aplicar un buen diseño al sitio, utilizando los conceptos que maneje (CSS, JavaScript, JQuery, Boostrap, MVC, etc.) y los que consulte.

**Debe crear un Servicio web ó Web Api con lo métodos que necesite.**

Cree una Base de datos En SQL llamada Usuarios, cree una tabla Estudiantes así:

\*Correo (No se puede repetir, (Obligatorio))

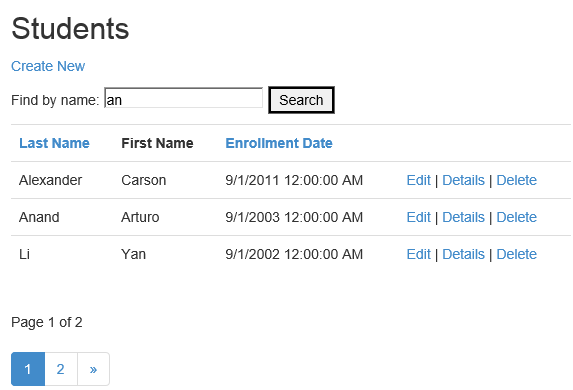
\*Clave (Obligatorio)

\*Nombre (Obligatorio)

\*Apellidos (Obligatorio)

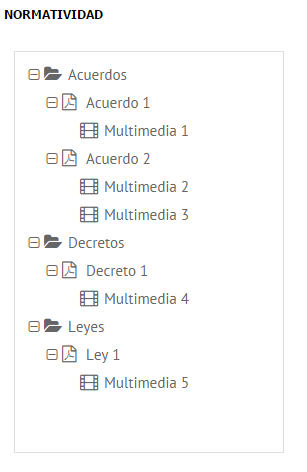
\*Fecha Cumpleaños (Obligatorio)

Debe haber una página que permita administrar los estudiantes registrados, al cargar la página se deben mostrar todos los estudiantes en un control Grid, éste debe contener paginación, permitir buscar por correo y las opciones de: Eliminar estudiantes (pedir confirmación antes de eliminar), Detalles (muestra en un pop-up toda la información del estudiantes), editar (permite modificar la información del estudiantes menos el correo electrónico), así por ejemplo:



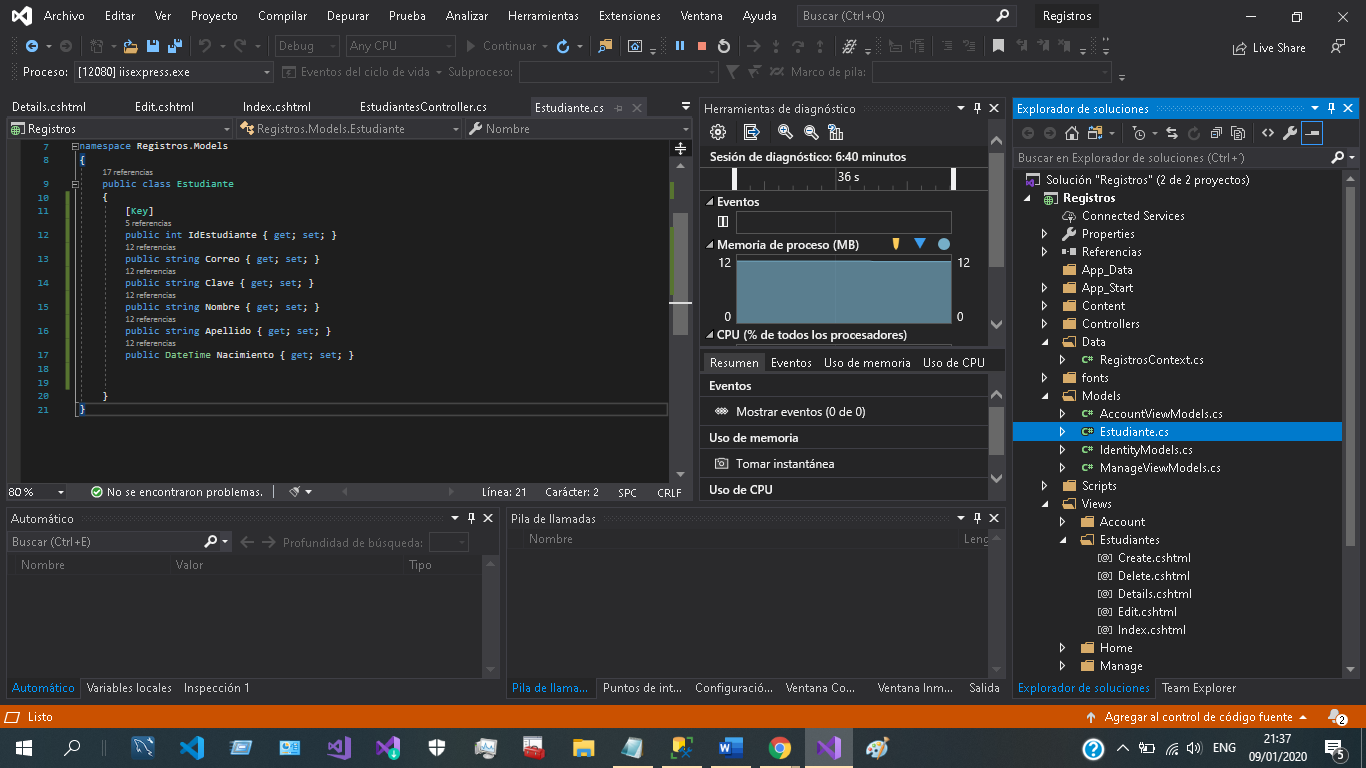
**%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%**

En la carpeta “treeview” La página “Demo.html” contiene un ejemplo de “Vista de Árbol”, el cual debe acomodar de tal forma que la estructura quede de la siguiente manera:



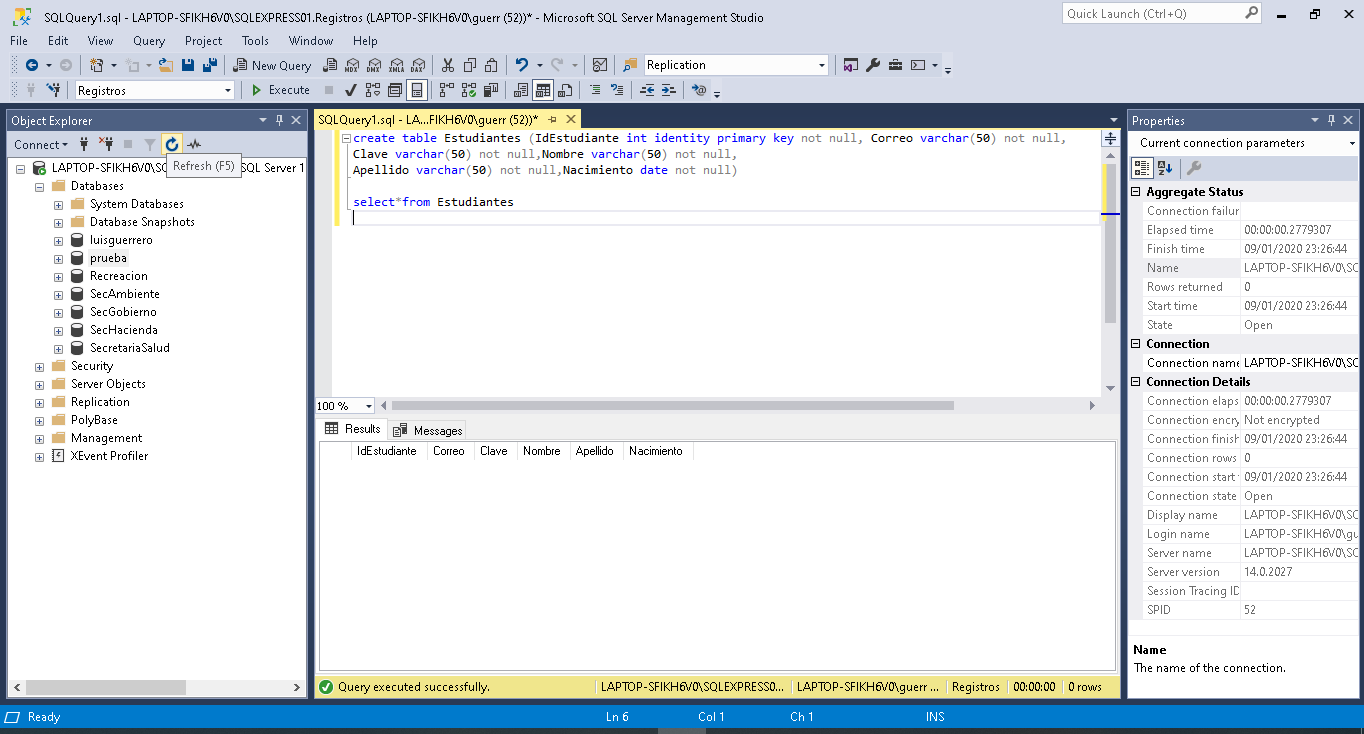
Solución desarrollo aplicación con ASP.NET

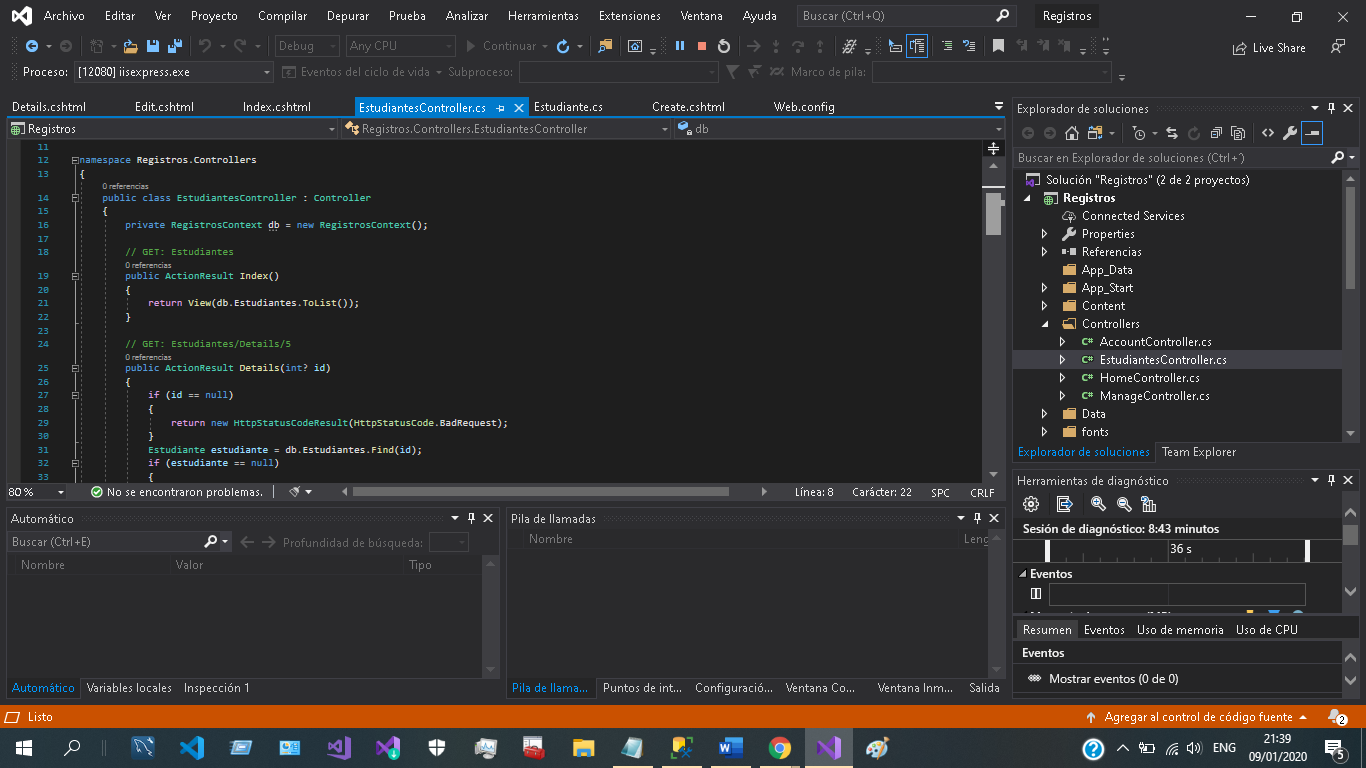
1.



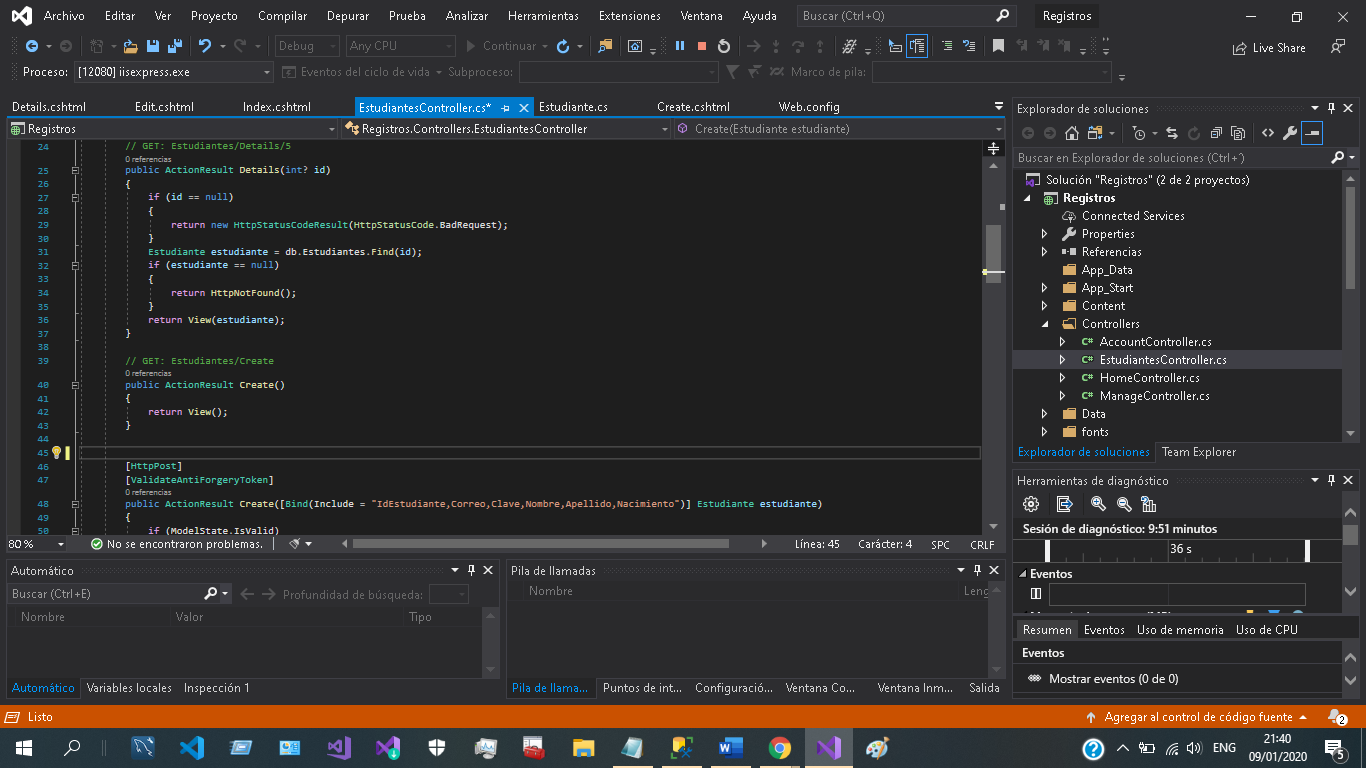
En esta parte inicial creamos en la carpeta Modelo, nuestra clase Estudiante con sus respectivos atributos , si bien en el planteamiento no se especifica el campo IdEstudiante , es bueno que toda clase o tabla en este caso , tenga una llave primaria.

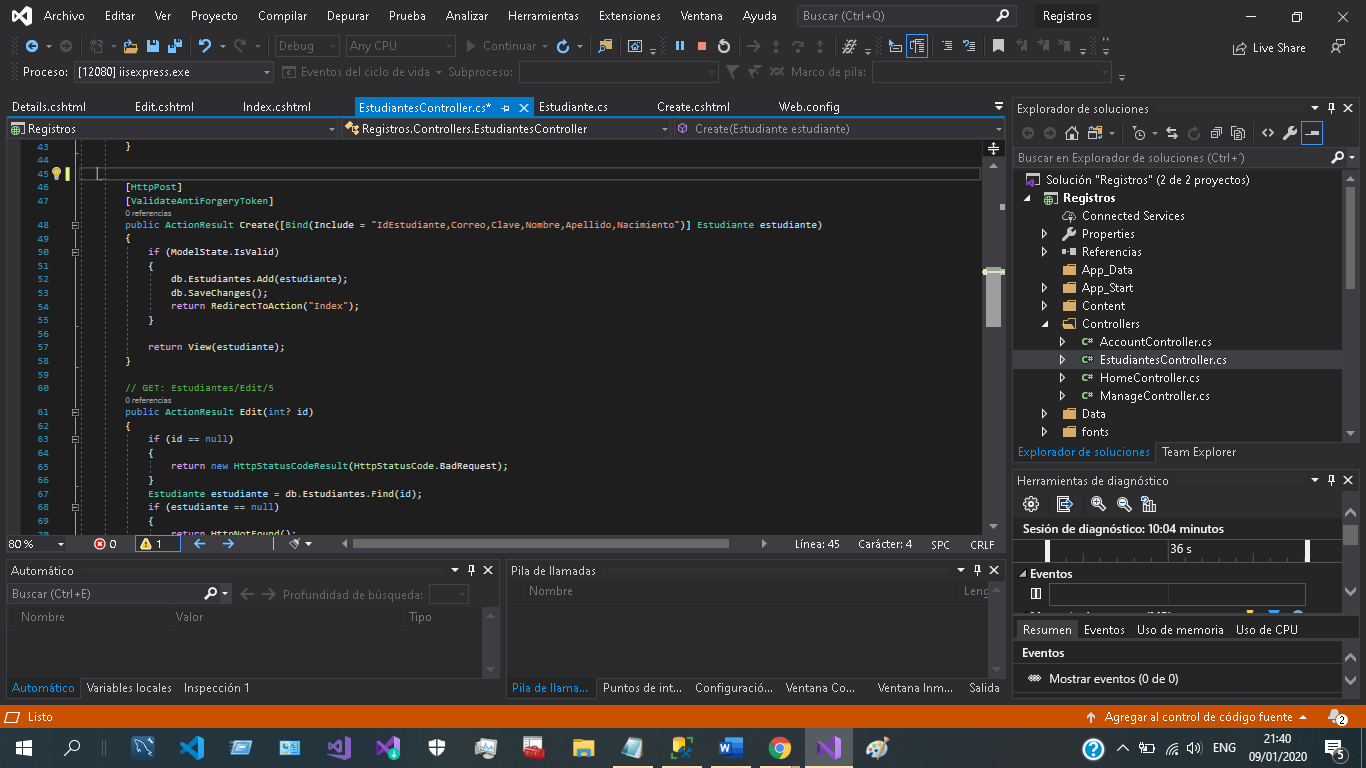
Creacion Base de datos

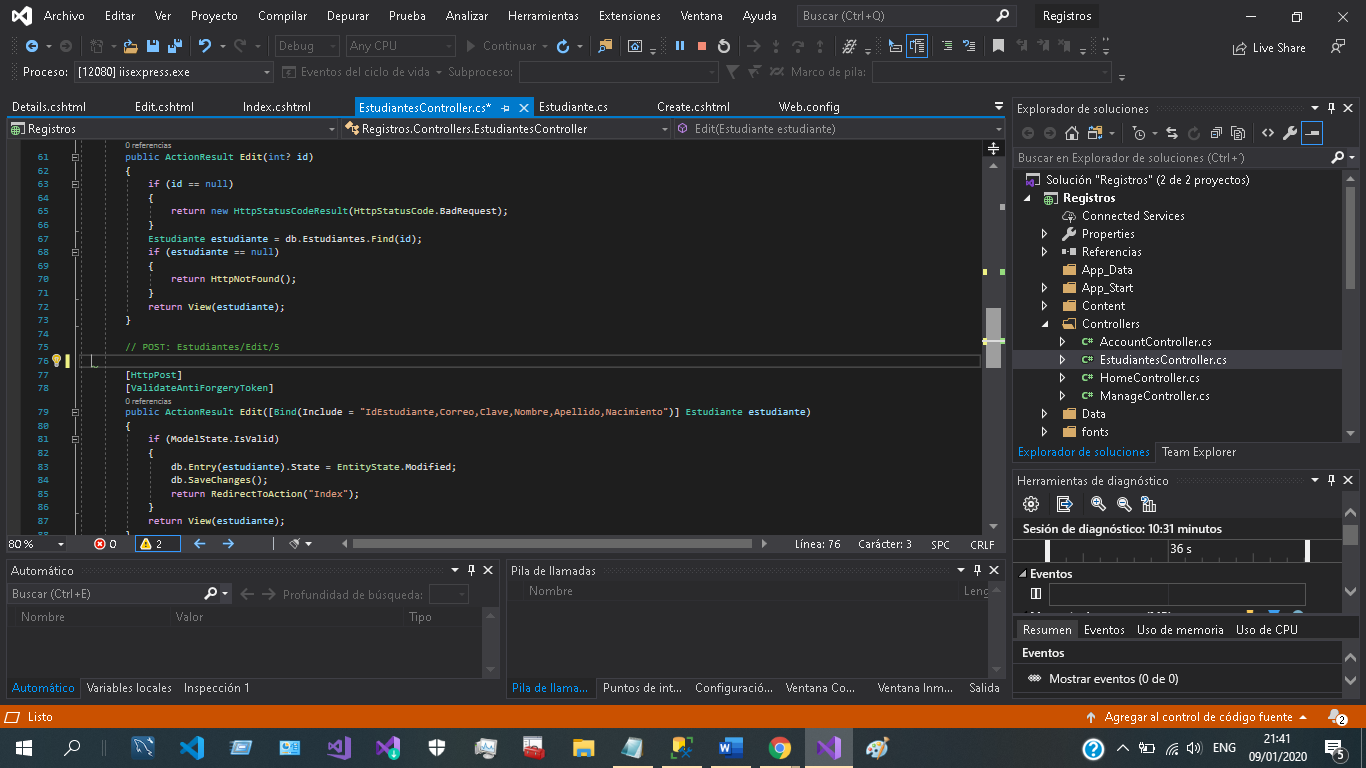


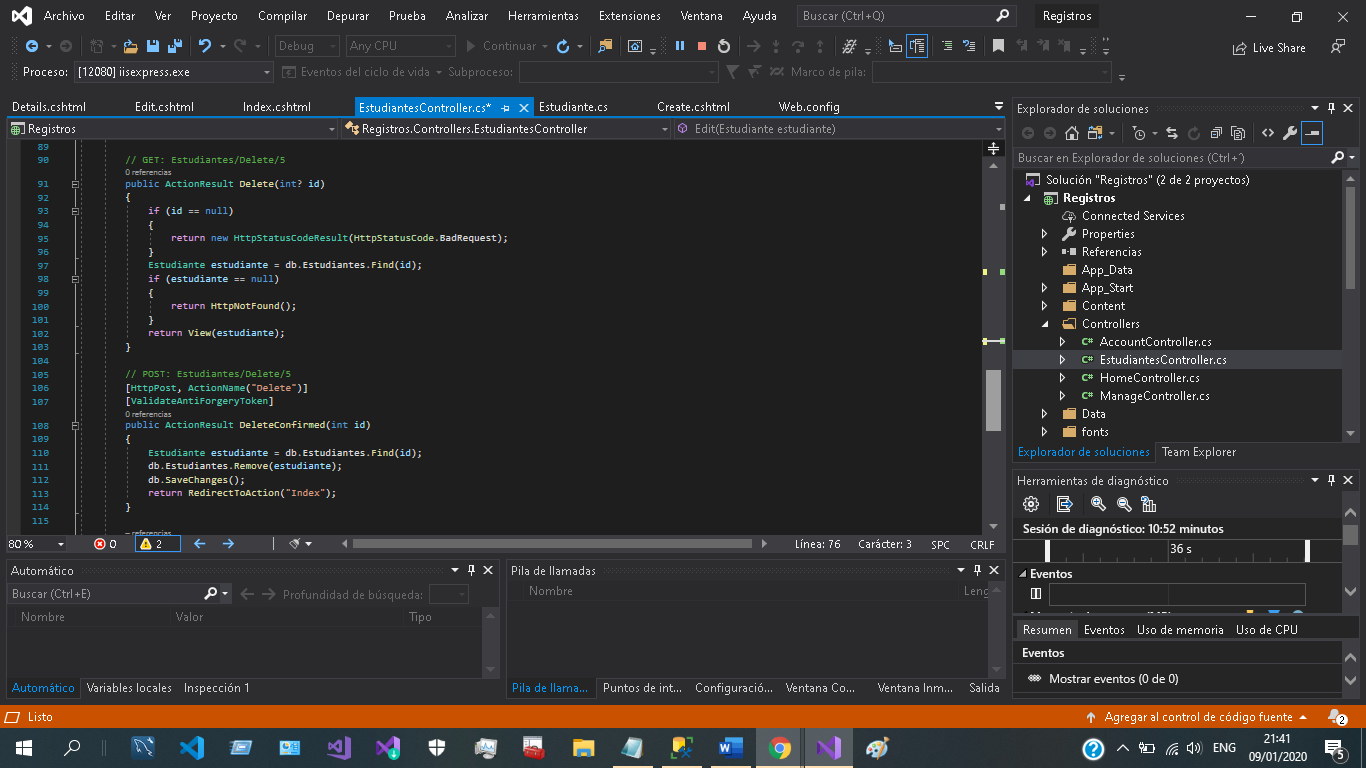
2. 

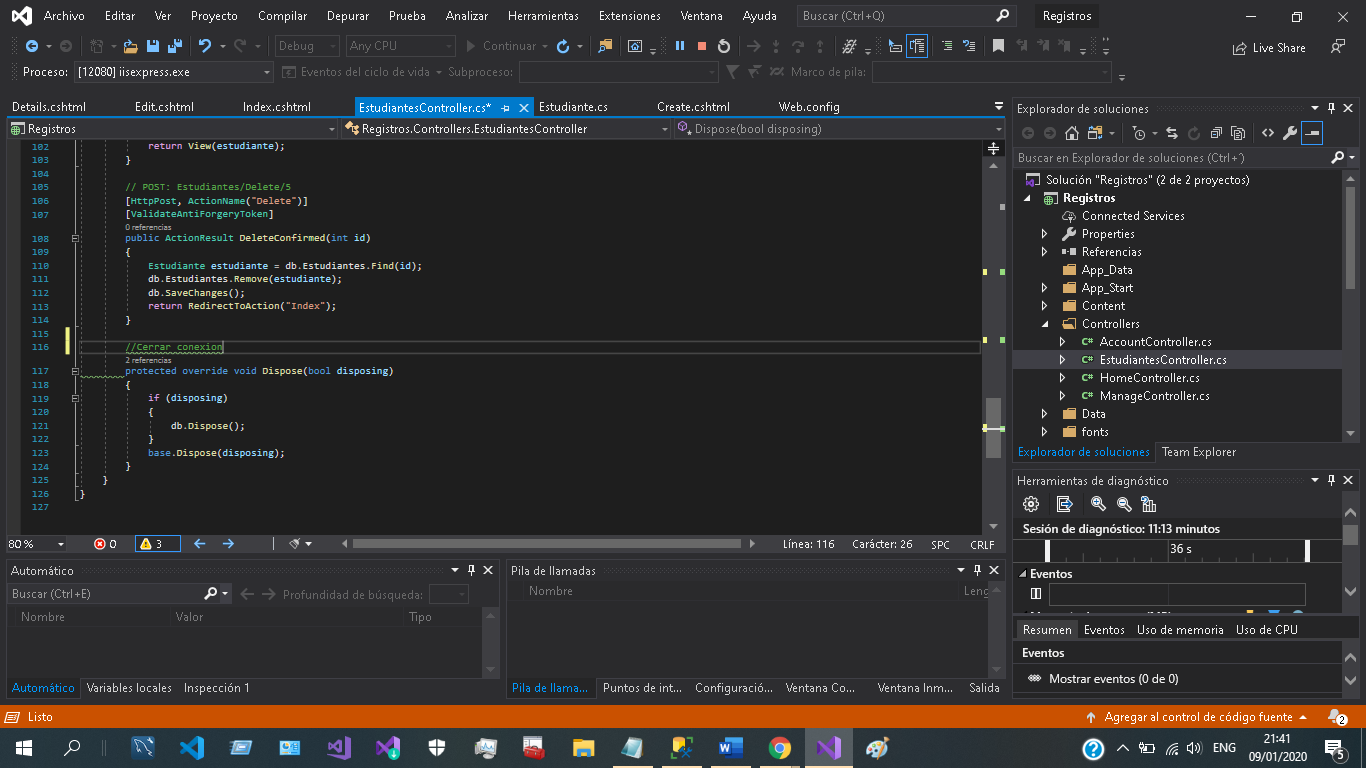
Creamos en base a nuestro modelo ,el controlador de la aplicación



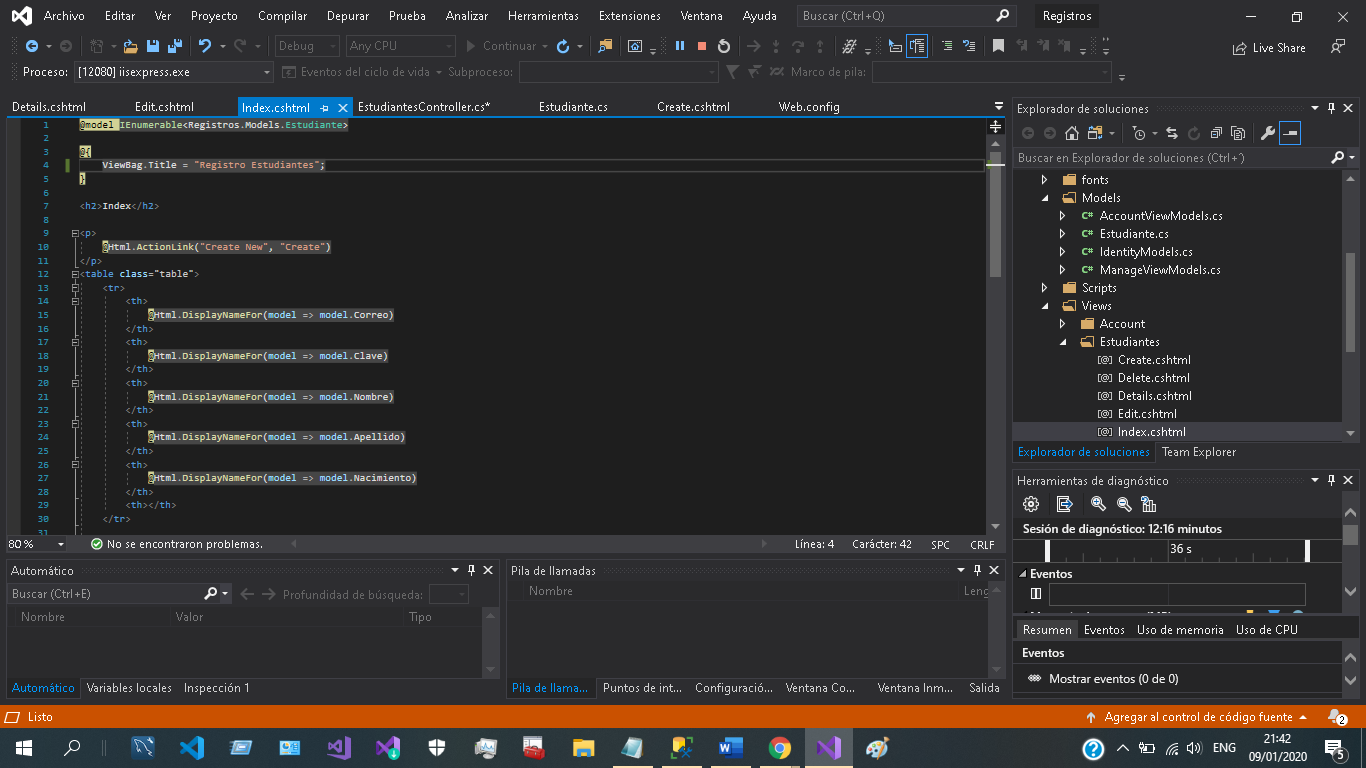






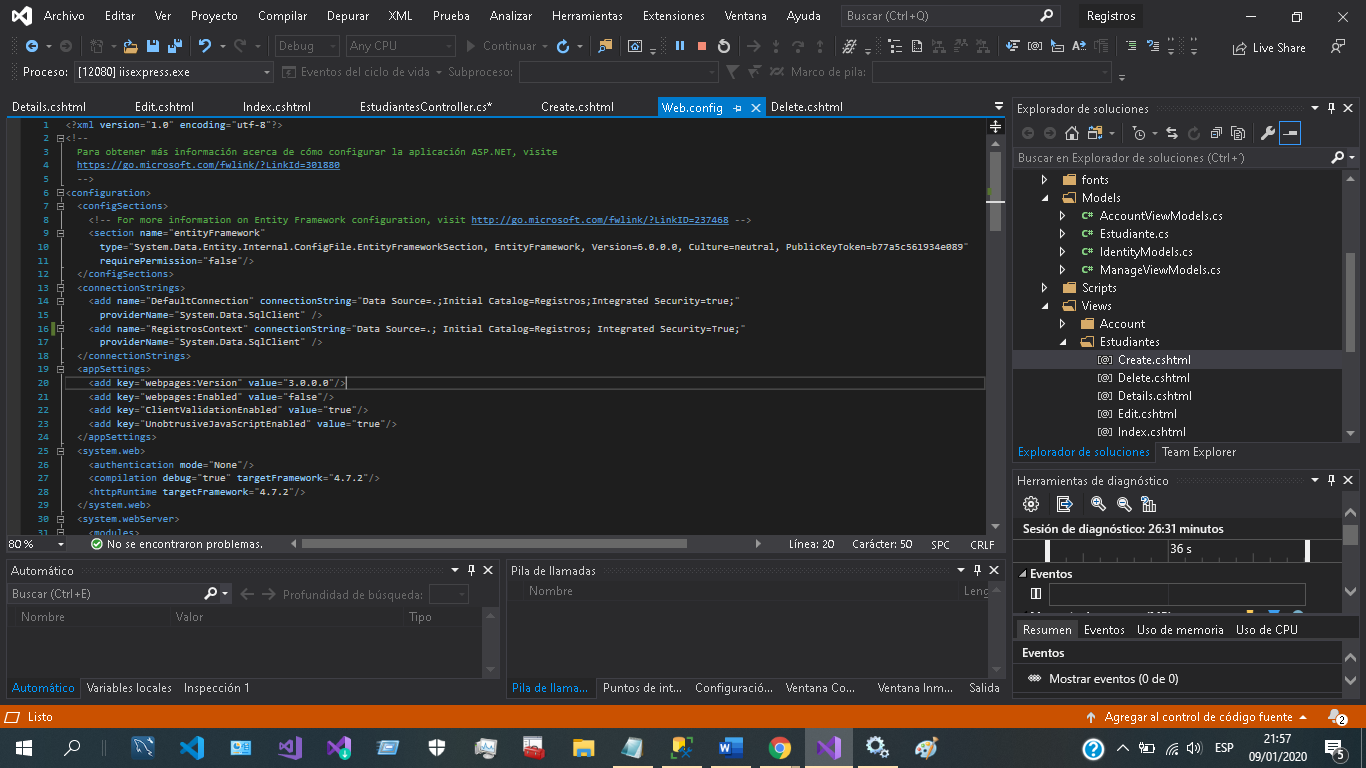


3.



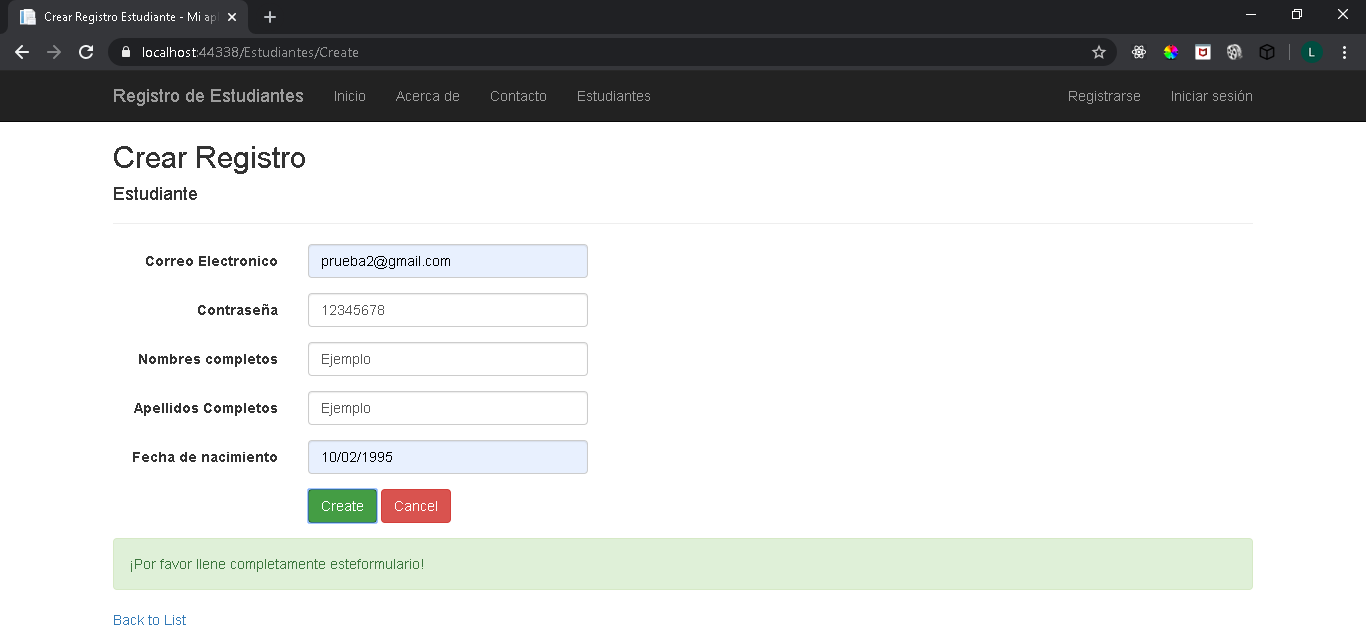
En la carpeta vistas generamos los archivos necesarios para nuestro CRUD

4.Establecer conexión



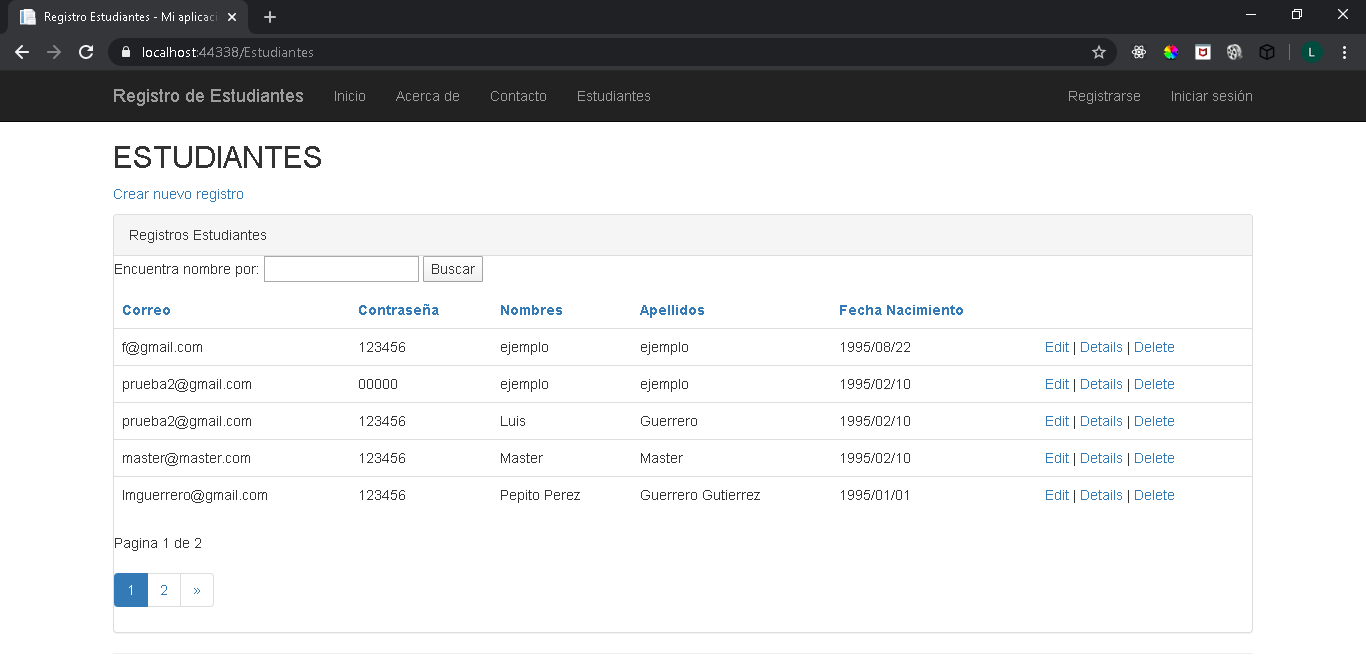
Vamos al archivo Web.config y asignamos la conexión a una base de datos local ¿Cómo? R/Luego del Data Source= colocamos un . (punto) y ahí estamos diciéndole a .NET que vamos a utilizar una cadena de conexión en una base de datos local , asignamos el nombre de la base de datos , y configuramos como “true” el parámetro Integrated Security

5.



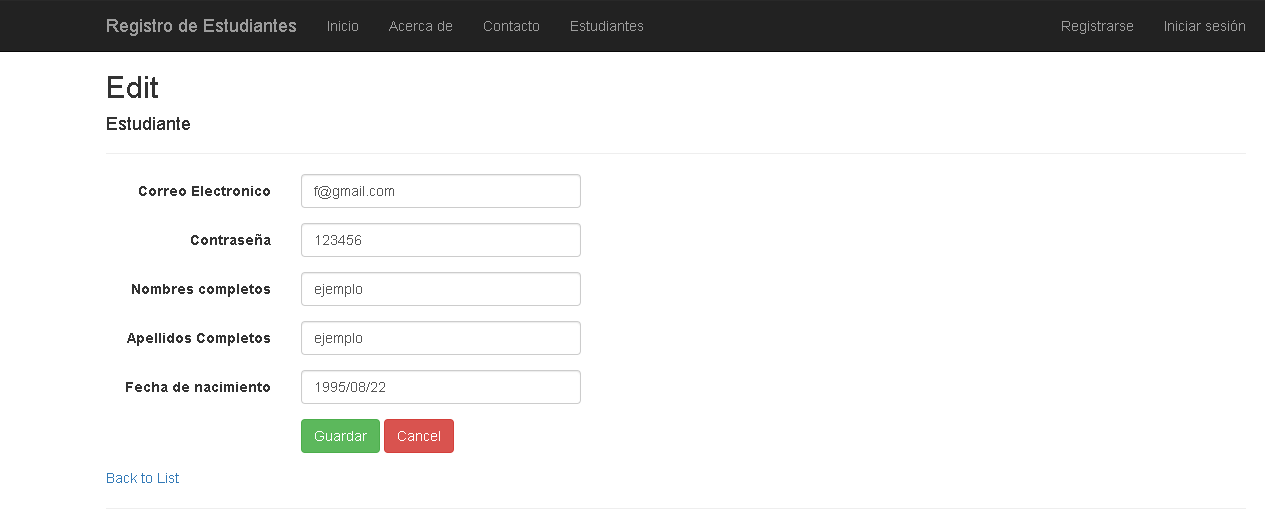
Create – Registro estudiante

6.

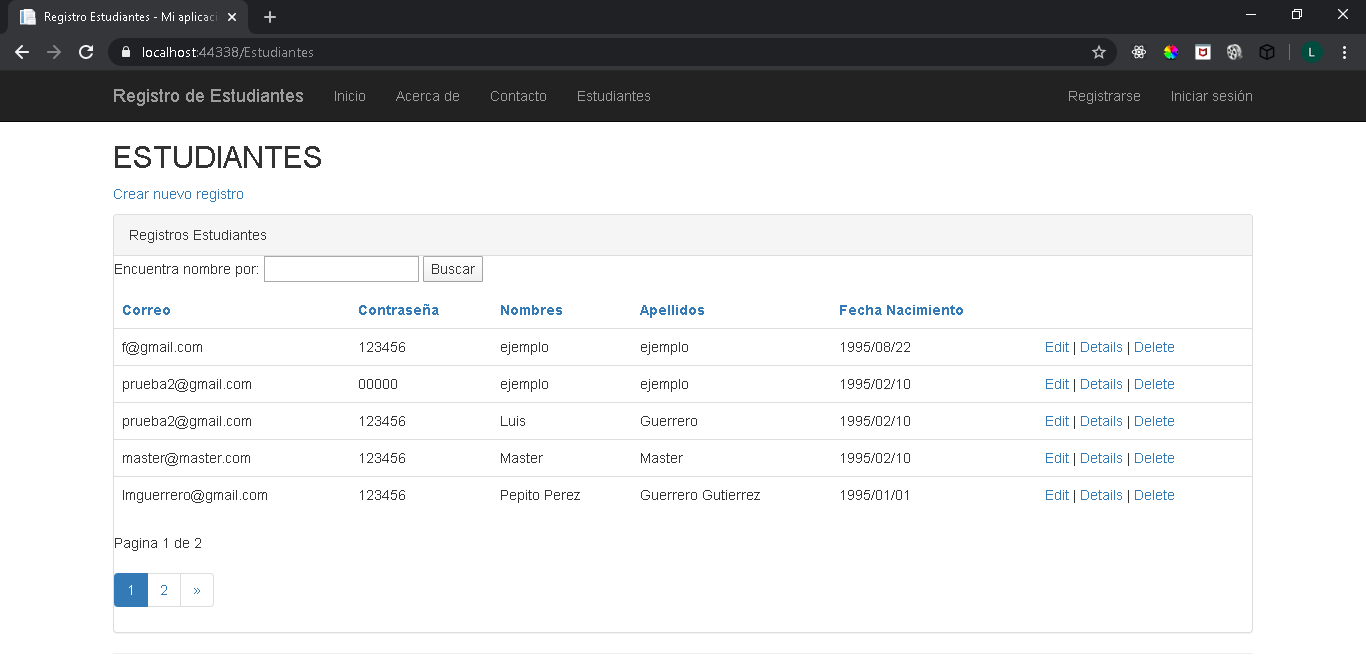


En esta vista podemos ver los detalles del registro recién insertado(

7.

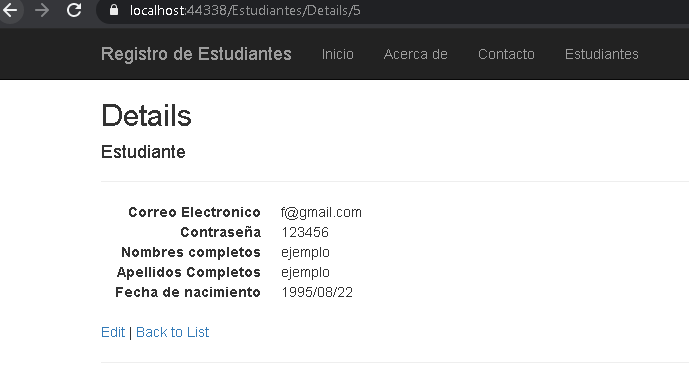


En esta vista podemos editar los atributos del estudiante



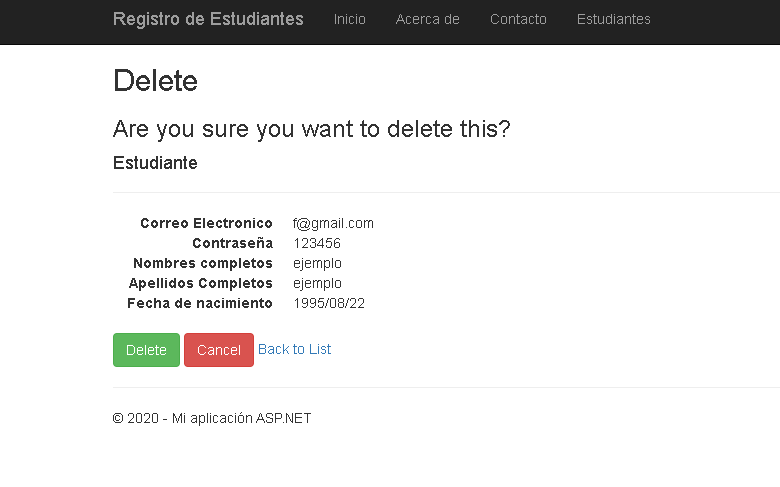
Como podemos observar , los cambios se realizaron con éxito.

8.

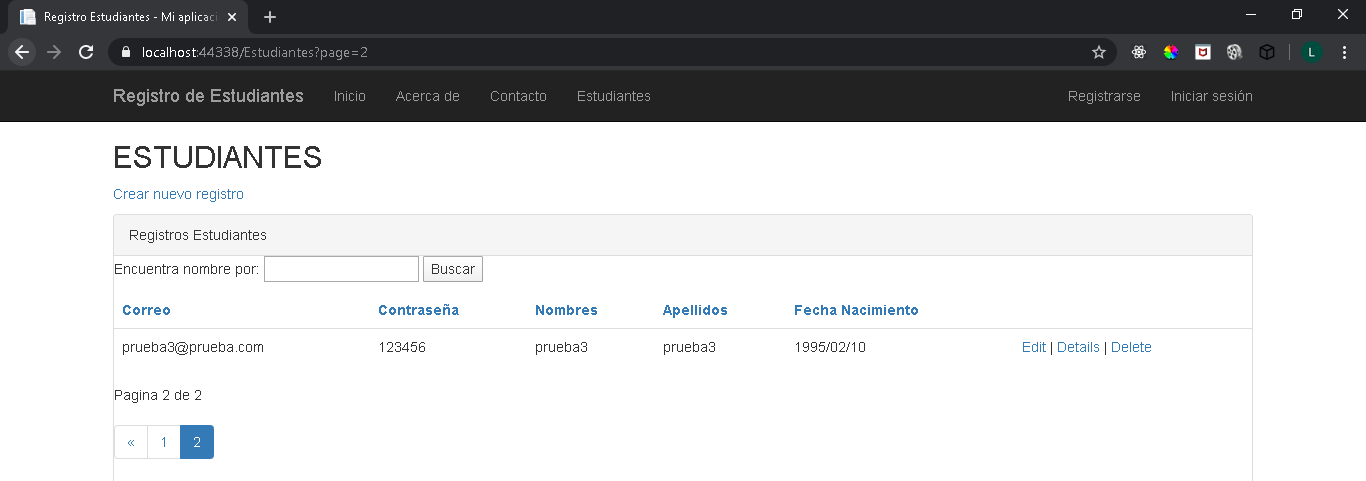


En esta vista podemos observar los detalles del estudiante

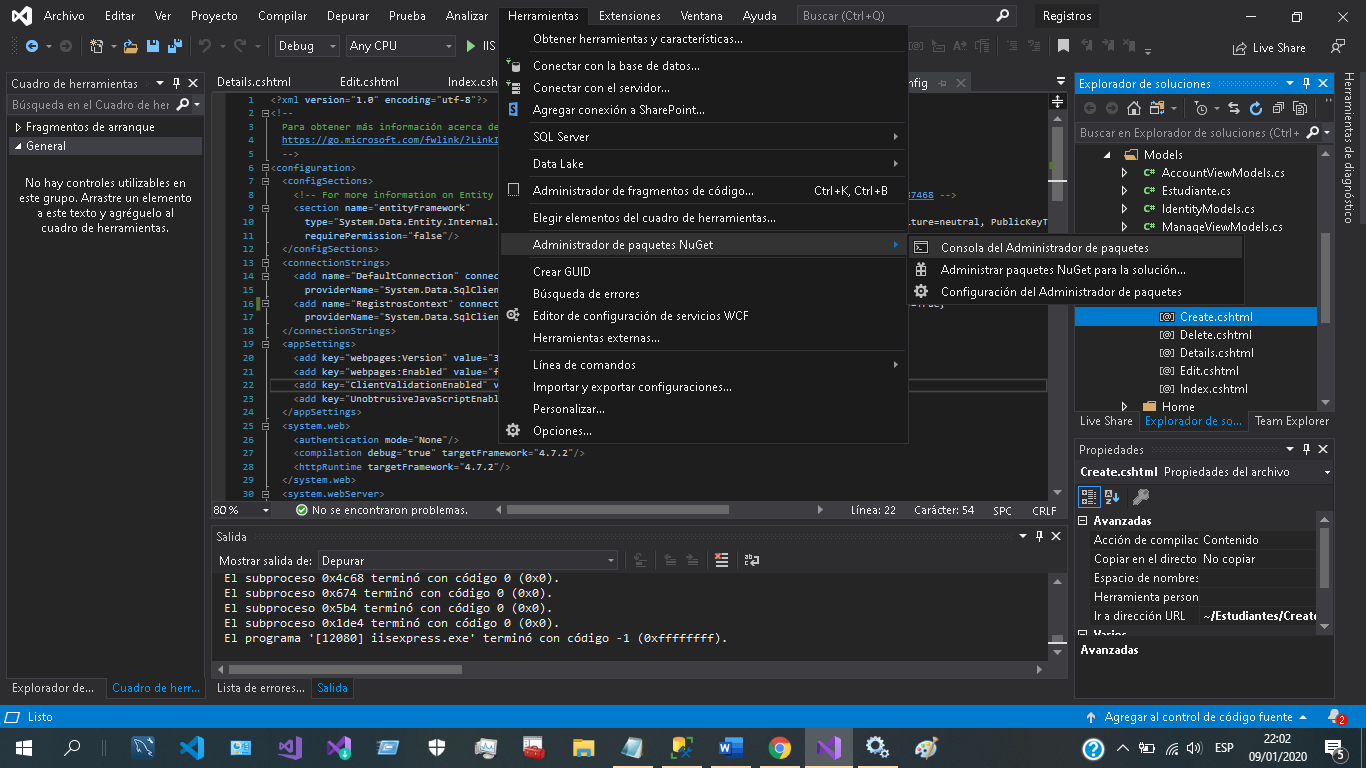
9.DELETE



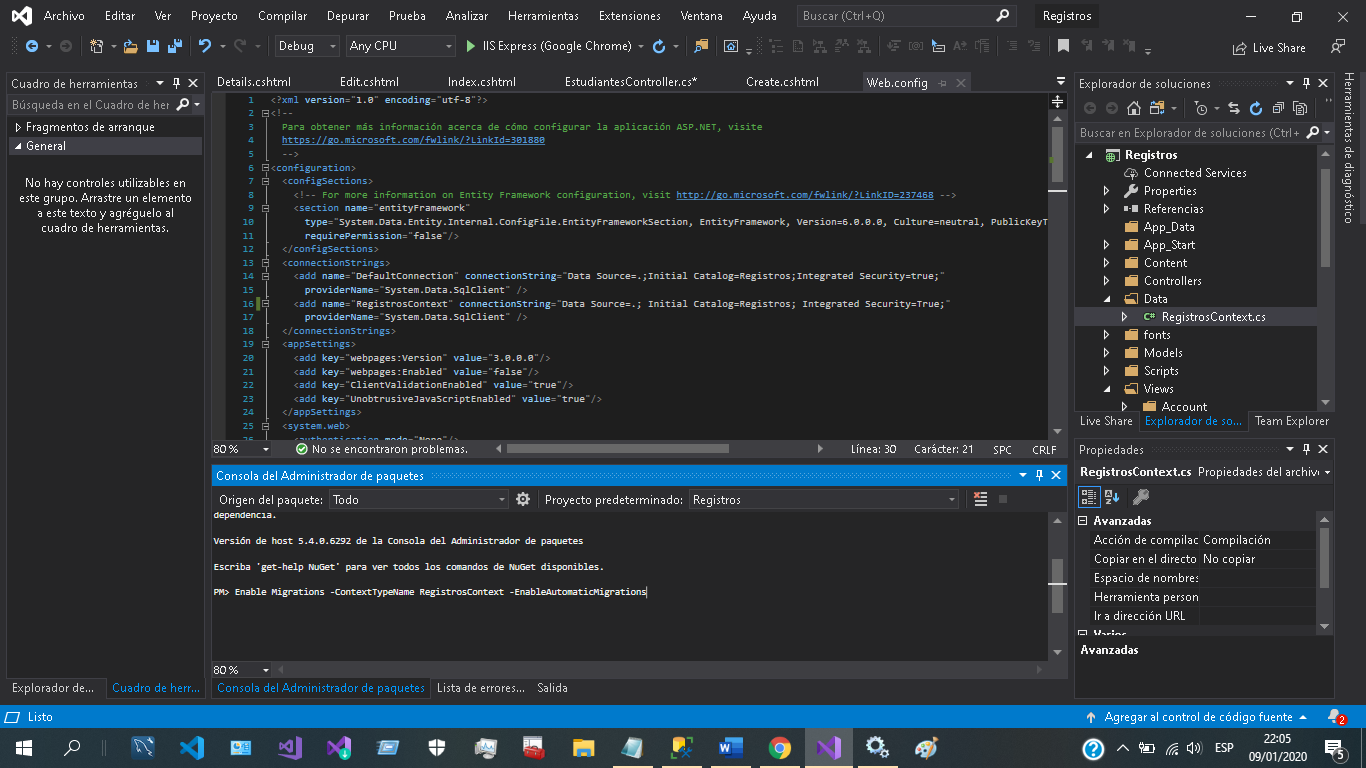
Paginación



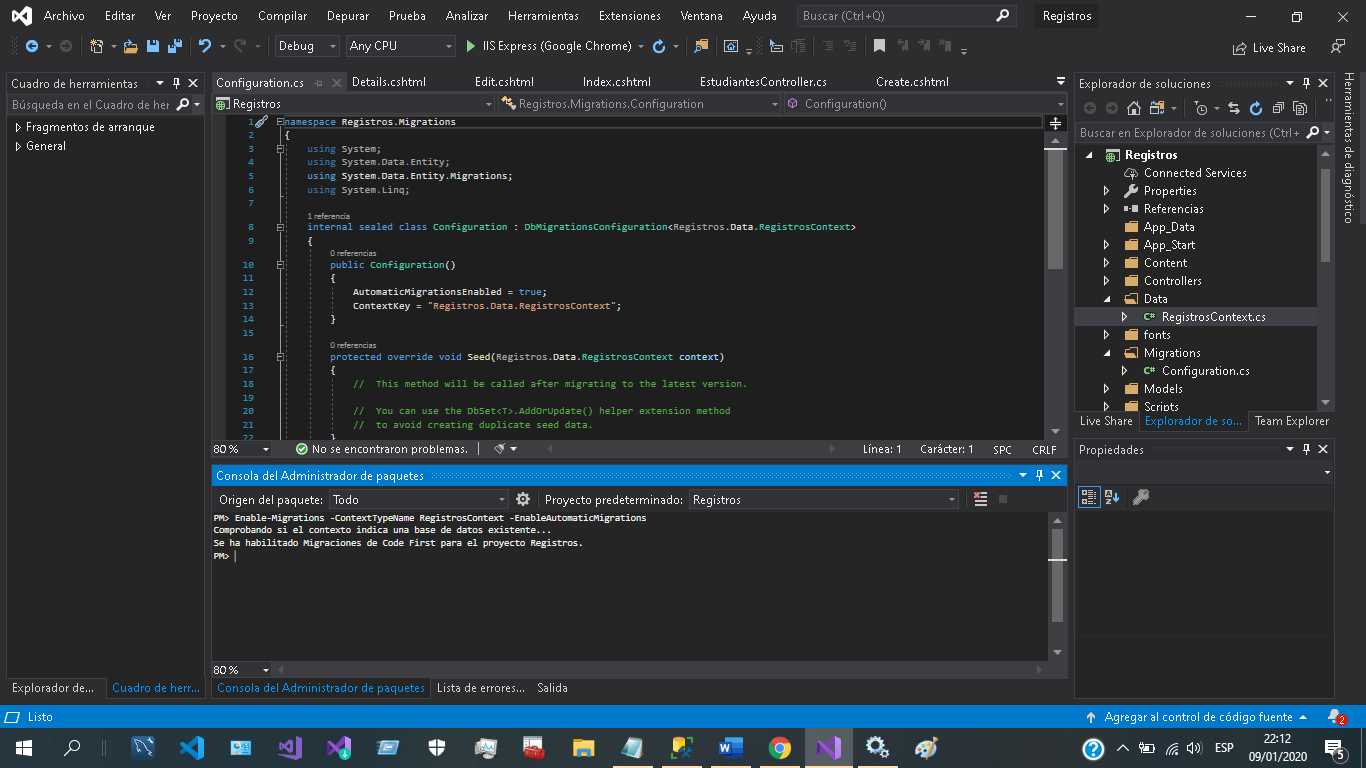
**Migraciones automáticas para la base de datos**



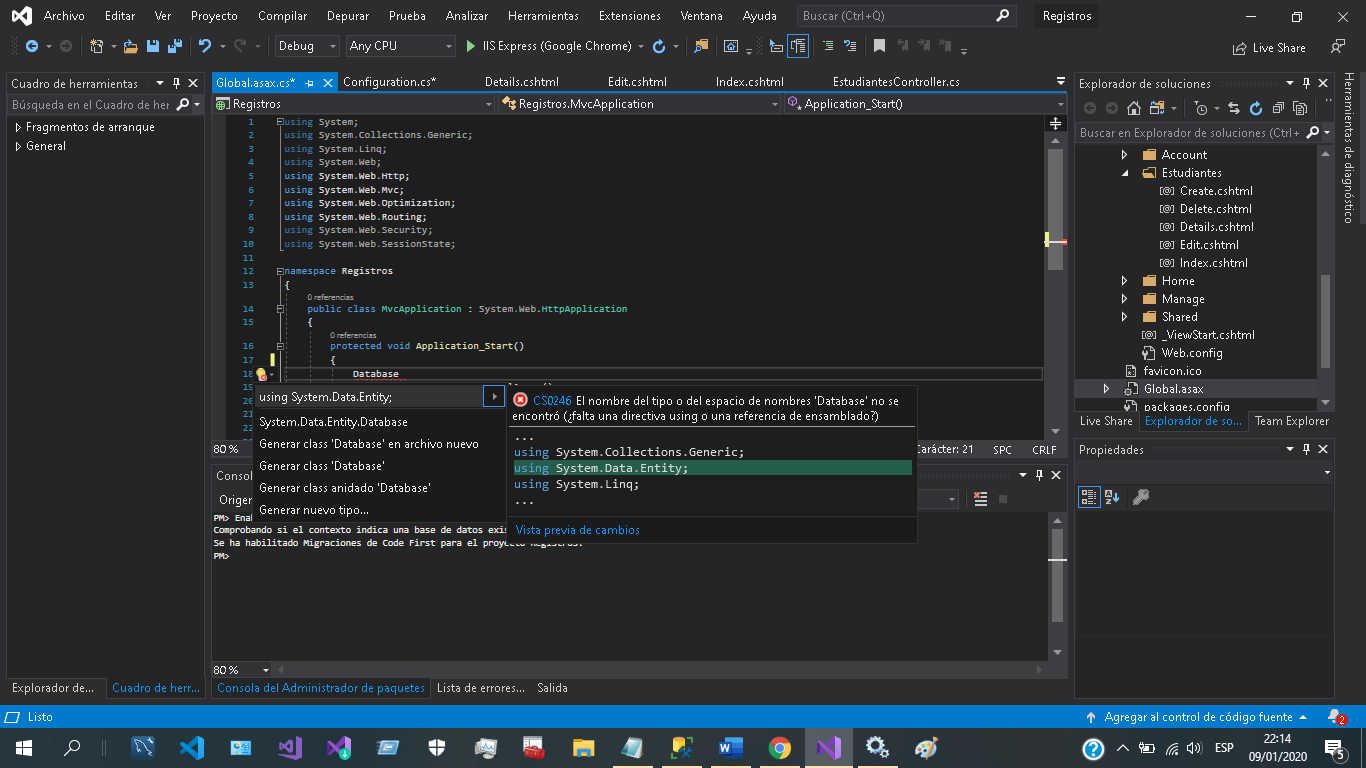
Vamos a la opción “Herramientas” -> luego a “Adminsitracion de paquetes Nuget” y escogemos la primer opción “Consola” y escribimos :

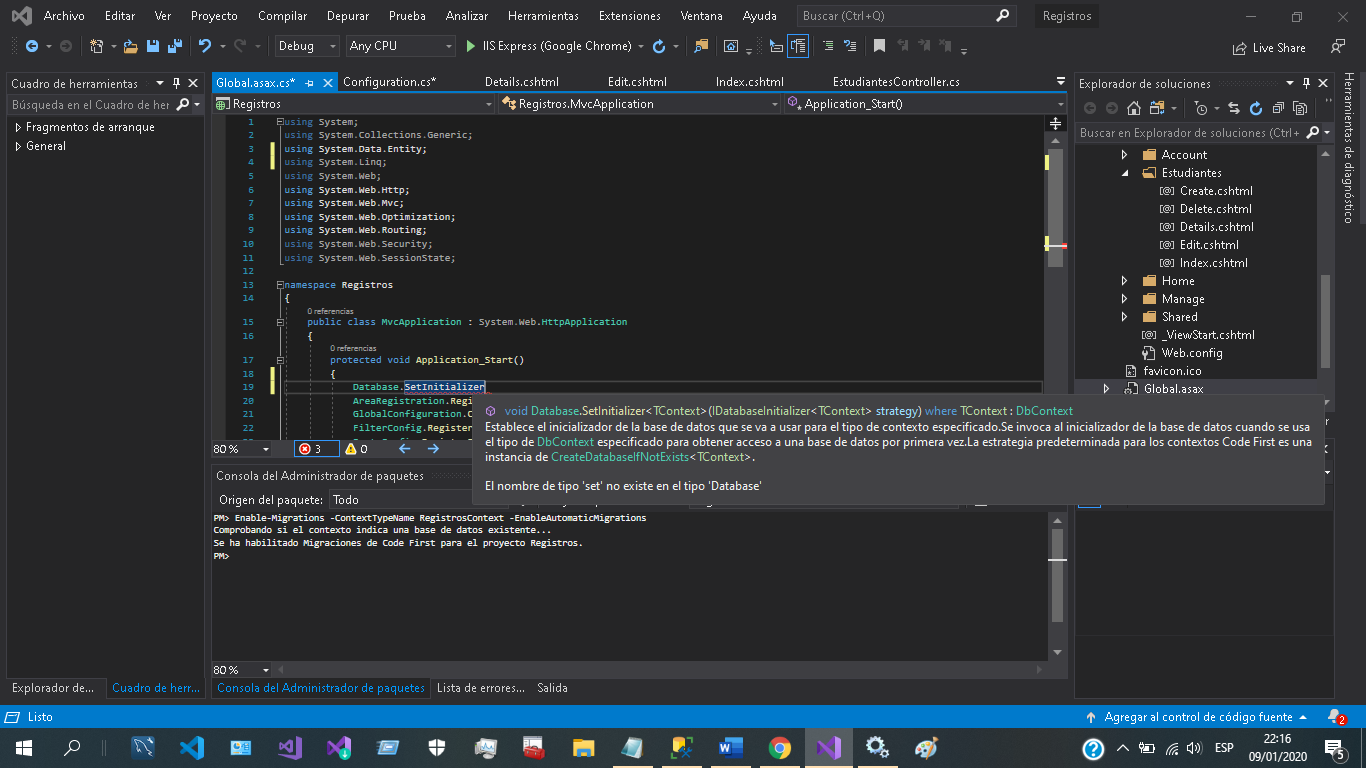
Enable-Migrations -ContextTypeName RegistrosContext -EnableAutomaticMigrations

Copiamos el nombre de nuestro context para habilitar las migraciones “esto con el fin de que a futuro si se hace algún cambio en el modelo(que viene siendo la tabla) SQL Server no lance errores pues habrían atributos que no existirían en la base de datos”



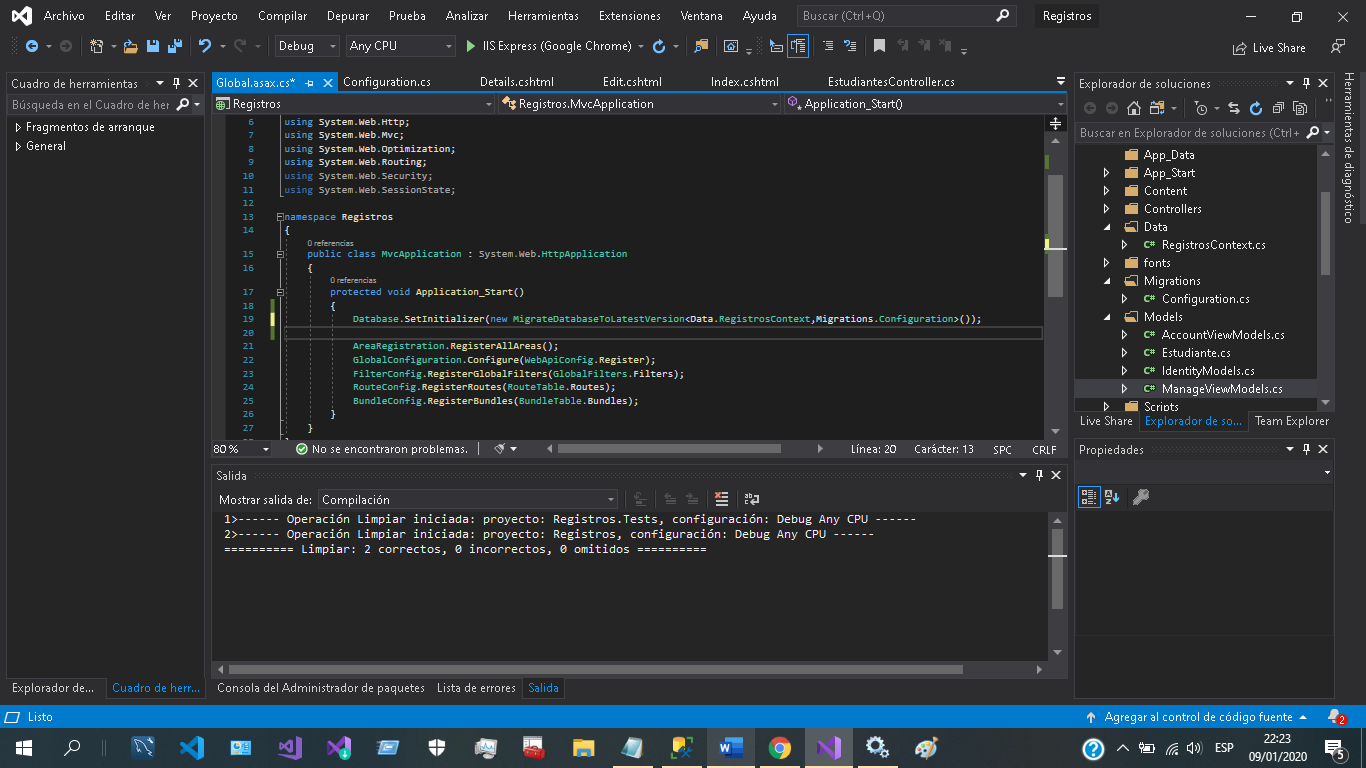
Una vez validado el proceso , nos crea la carpeta Migrations y la clase Configurations , vamos al archivo Global, este ultimo lo que hará será validar si el modelo cambio , y si cambio lo debe actualizar , usamos la librería “using System.Data.Entity” de la clase “Database” como se observa en la imagen



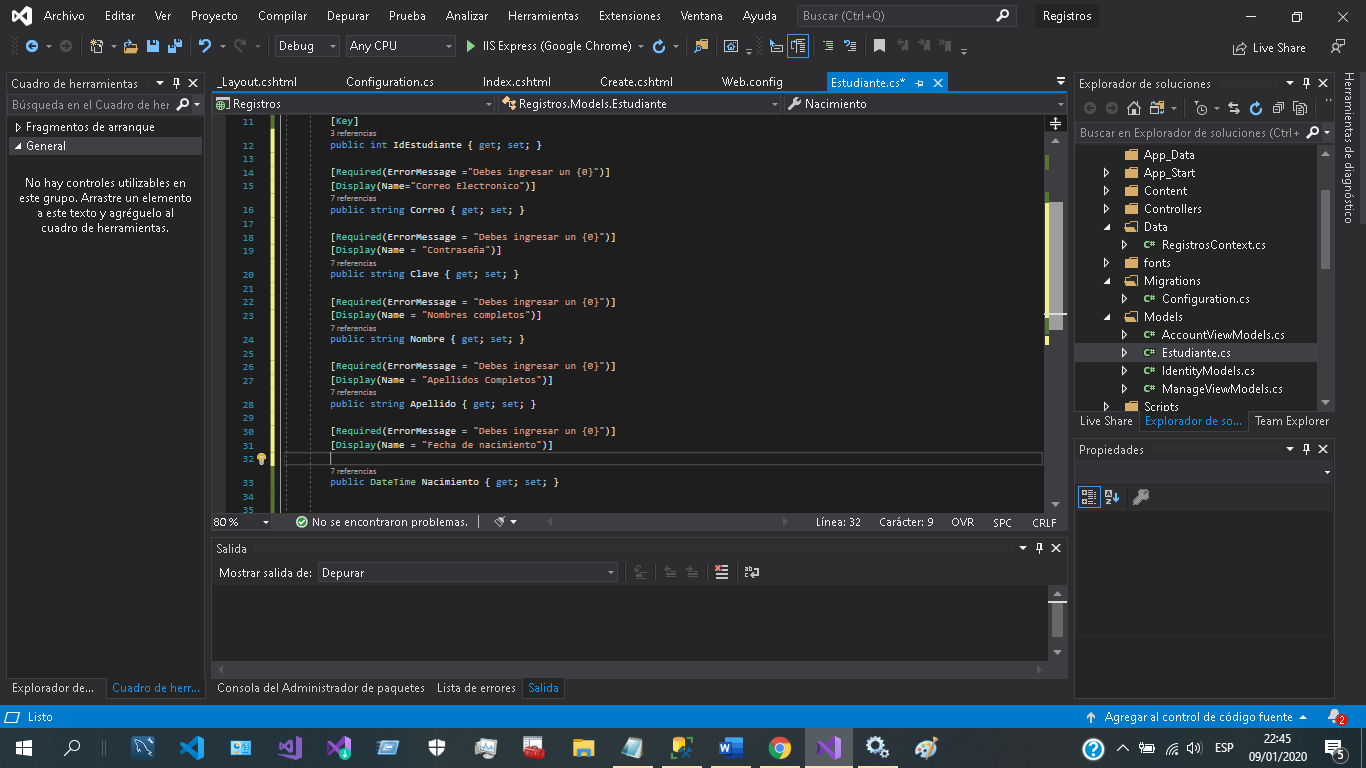


Luego usamos el método “set initializer” y le pasamos el siguiente parámetro

Database.SetInitializer(new MigrateDatabaseToLatestVersion<Data.RegistrosContext,Migrations.Configuration>());



Definimos los campos obligatorios del modelo y por “estetica” le agregamos otros nombres a las variables



Y se veran asi:

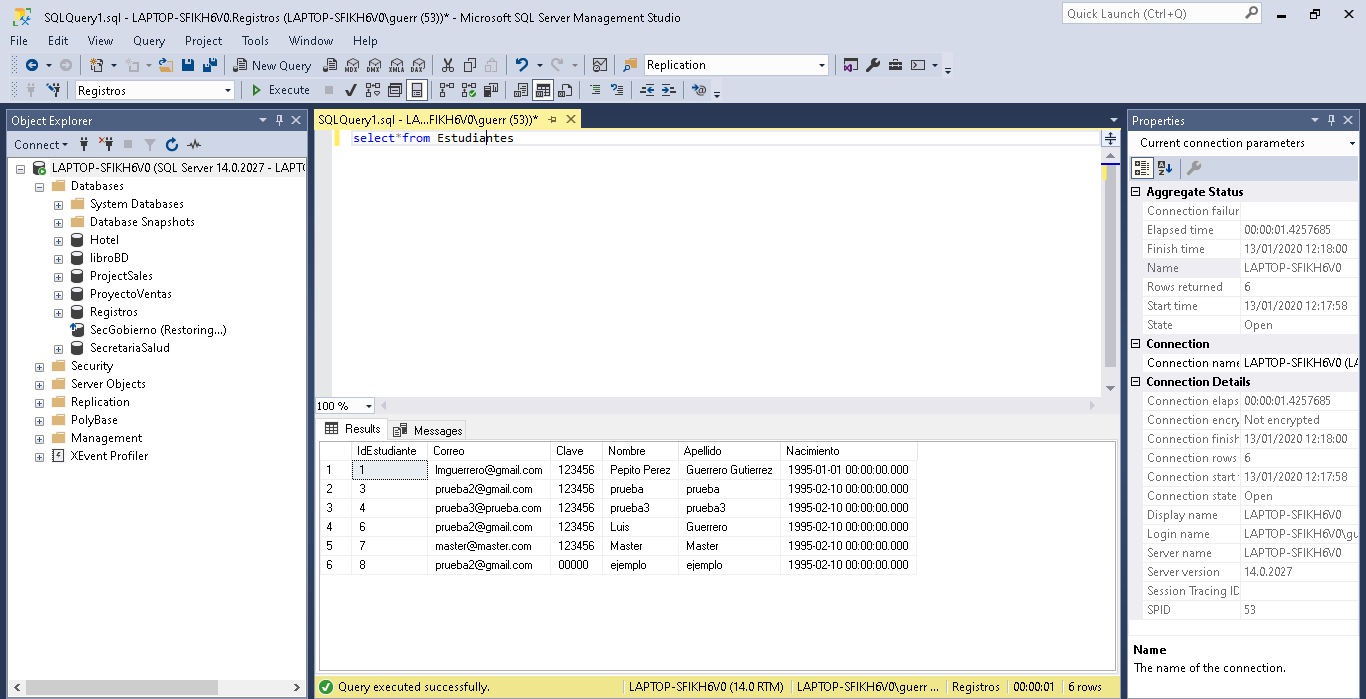


Validacion campos obligatorios

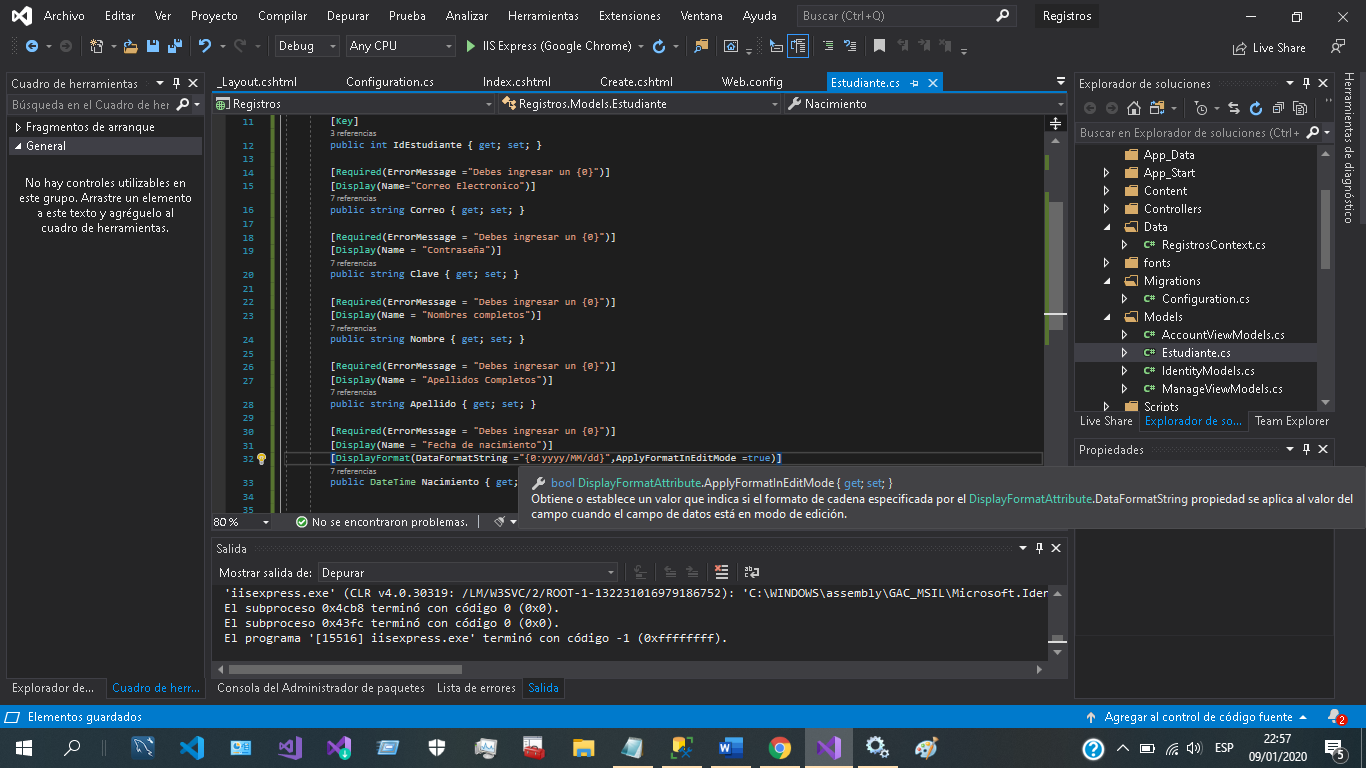




Base de datos



**Configuracion campo fecha de Nacimiento**



Con esta linea de Código sobre el campo “Nacimiento ”lo que hacemos es evitar que por defecto , al insertarse un registro se agregen los minutos y segundos y solo agregue año,mes,dia

[DisplayFormat(DataFormatString ="{0:yyyy/MM/dd}",ApplyFormatInEditMode =true)]