Segundo Taller:

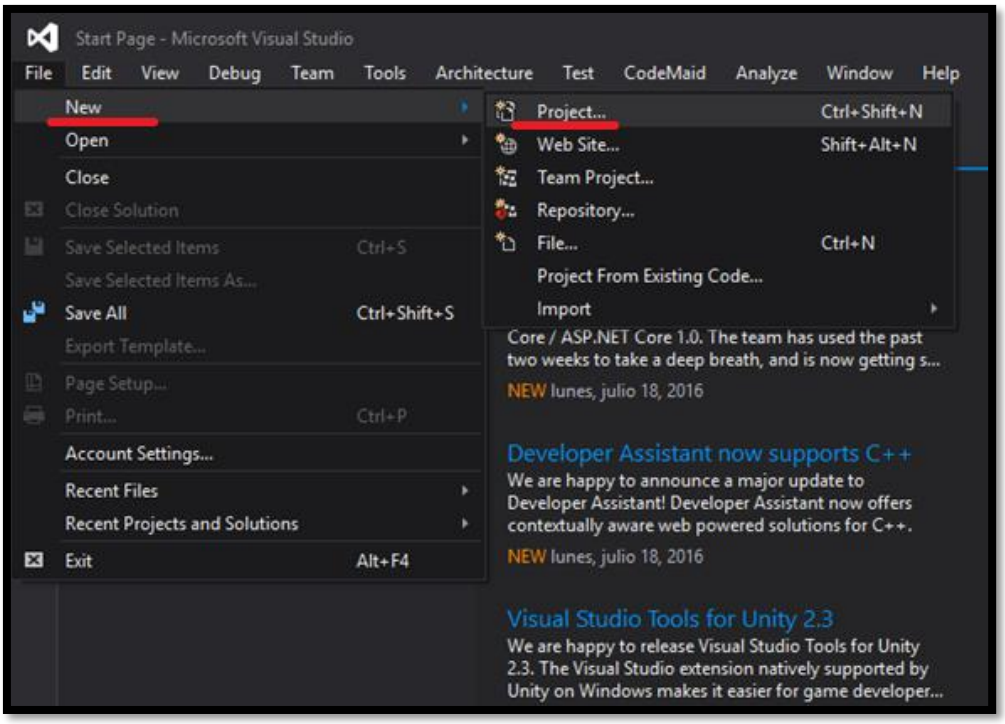
Introducción a ASP.net MVC 5

ASP.NET

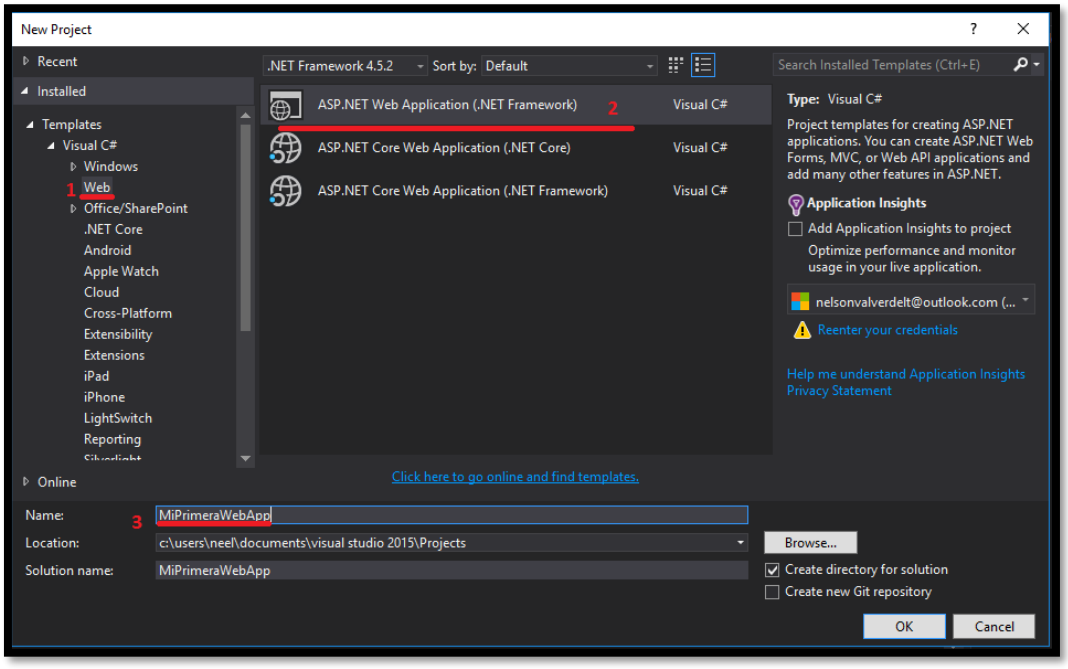
I

UPAO.net

1) Creamos un nuevo proyecto



2) Seleccionamos un nombre a nuestro proyecto, en este caso, crearé un proyecto con el nombre **"MiPrimeraWebApp"**



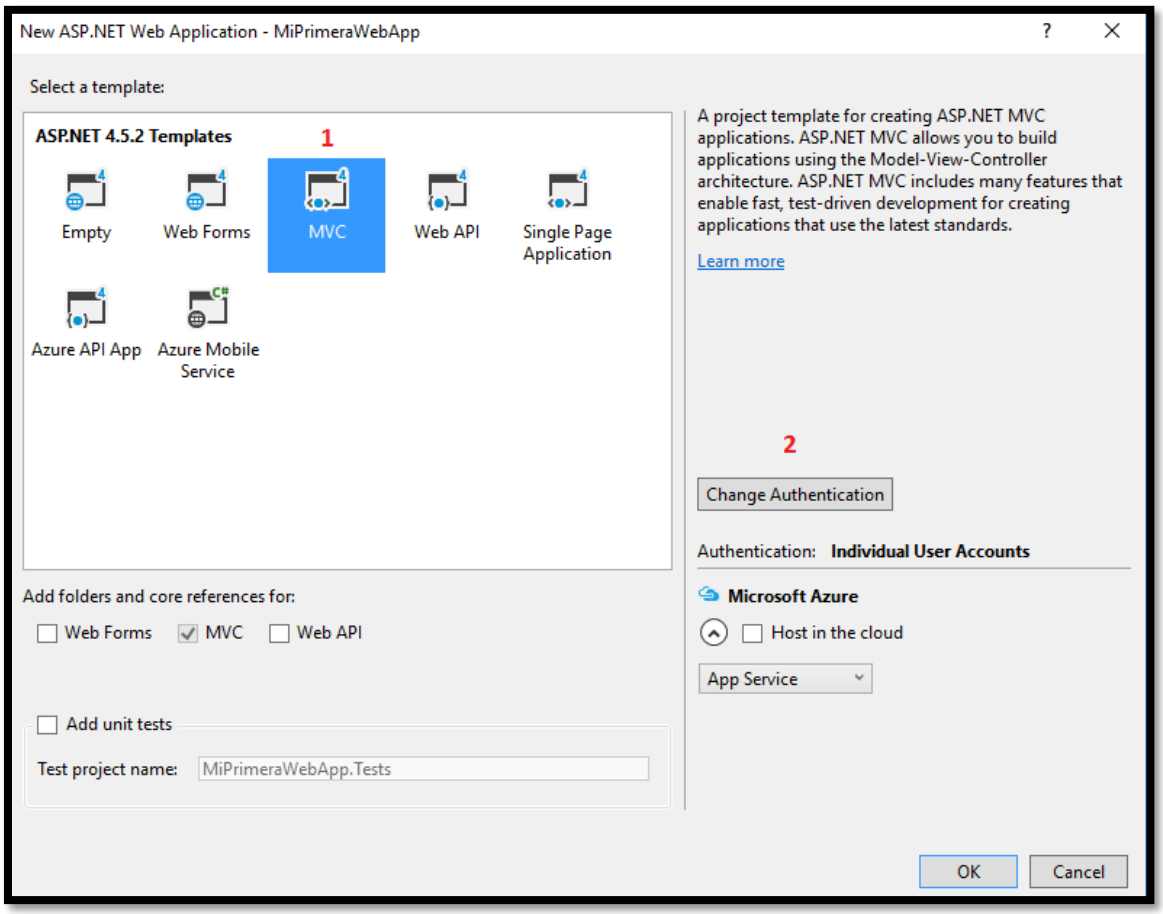
3) Seleccionamos **MVC** (Modelo, Vista, Controlador)

Explicación:

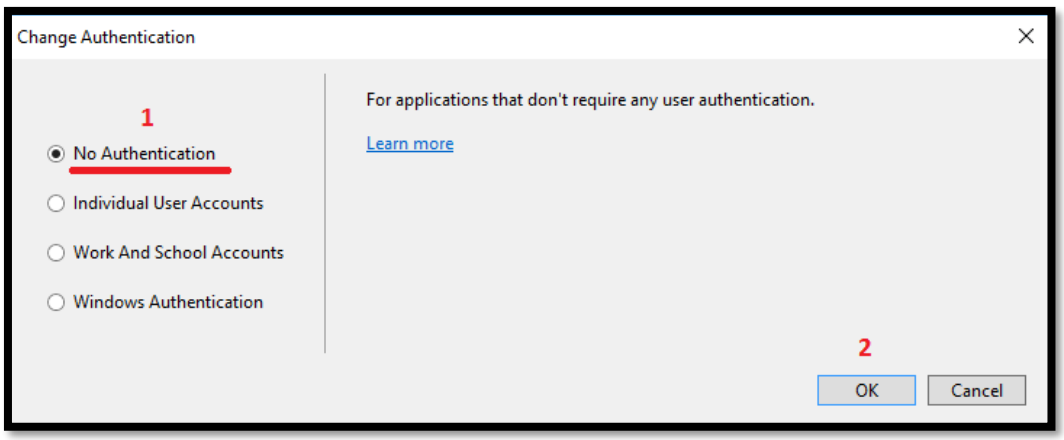
M → Modelo → Es una simple clase con atributos.

V → Vista → Es lo que va a ver nuestro usuario, en este caso una página.

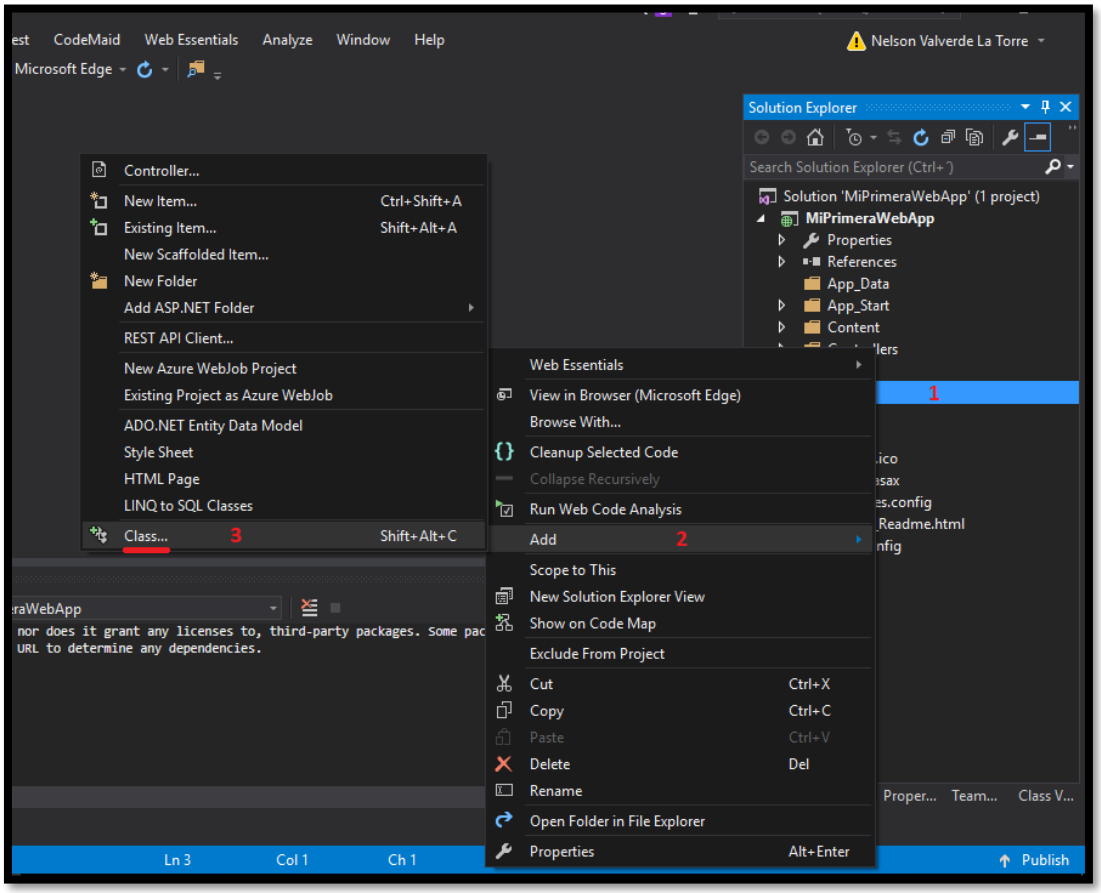
C → Controlador → Es la parte lógica de nuestra aplicación.



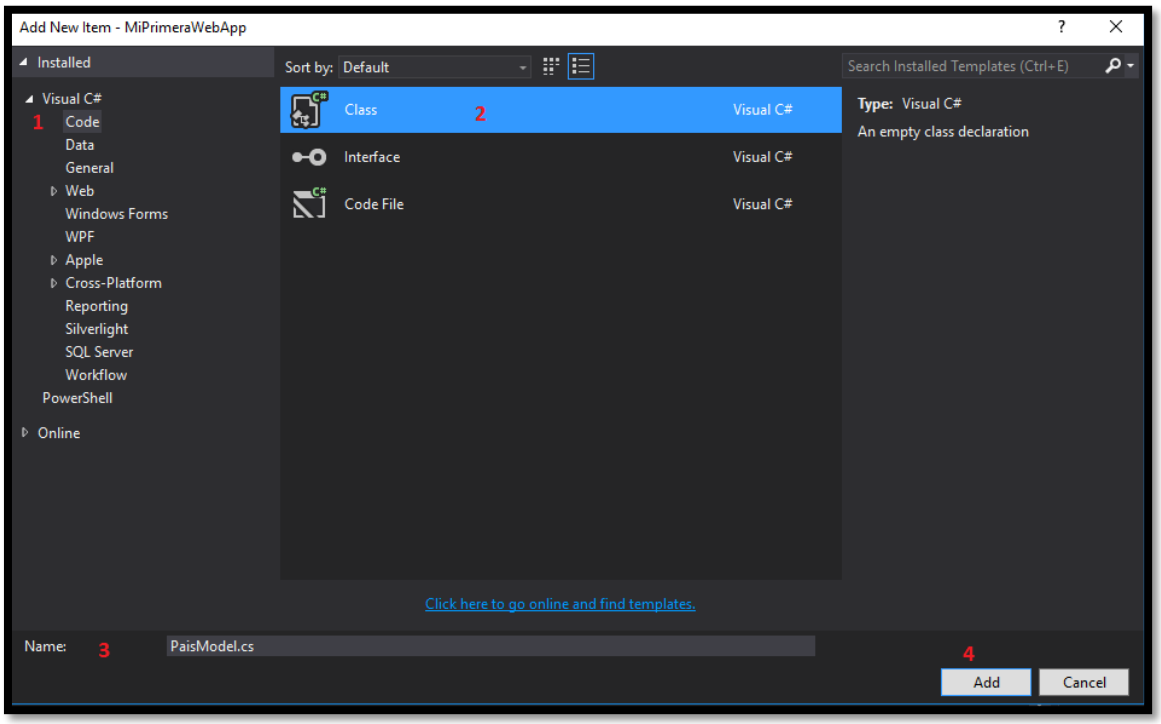
4) Para este ejemplo no haremos uso de autenticaciones



5) Creamos 3 clases dentro de nuestra carpeta Models, recordar que cada carpeta está dividida de manera lógica y se recomienda seguir con esta estructura



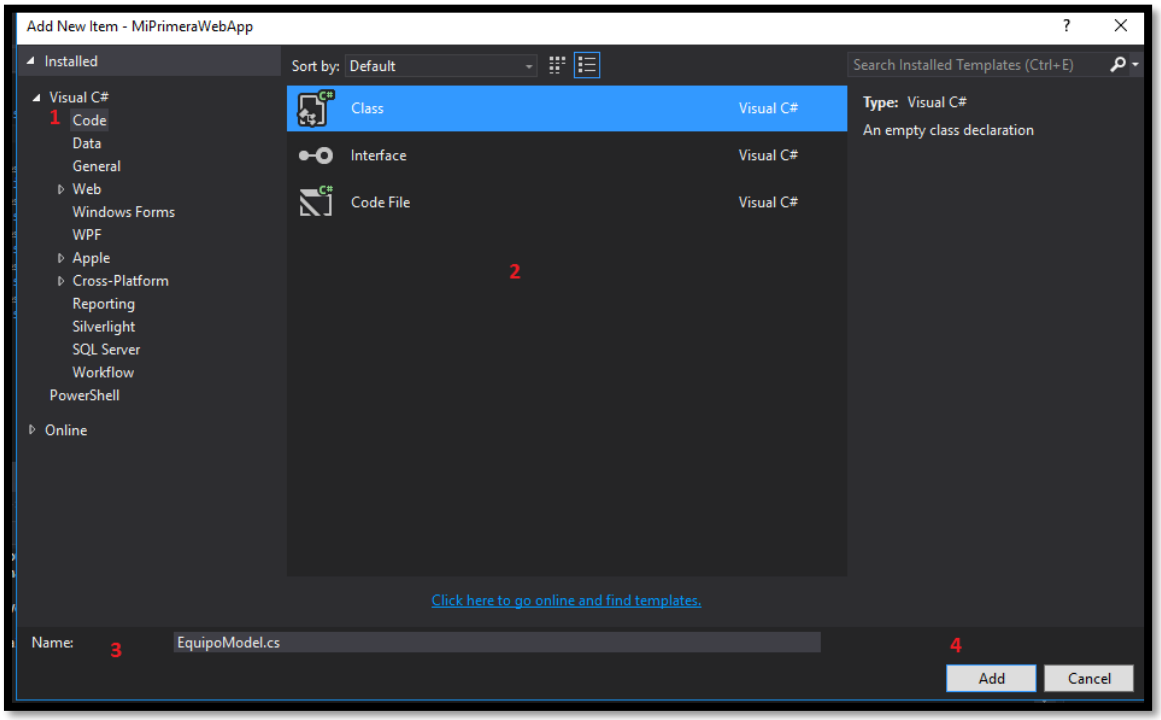
6) Creamos nuestra primera clase **PaisModel.cs**



7) Agregamos los atributos correspondientes a **PaisModel.cs**

```
using System.ComponentModel.DataAnnotations;                    1

namespace MiPrimeraWebApp.Models
{
    0 references
    public class PaisModel                                      2
    {
        [Key]
        0 references
        public int IDPais { get; set; }
        0 references
        public string  Nombre { get; set; }
    }
}
```

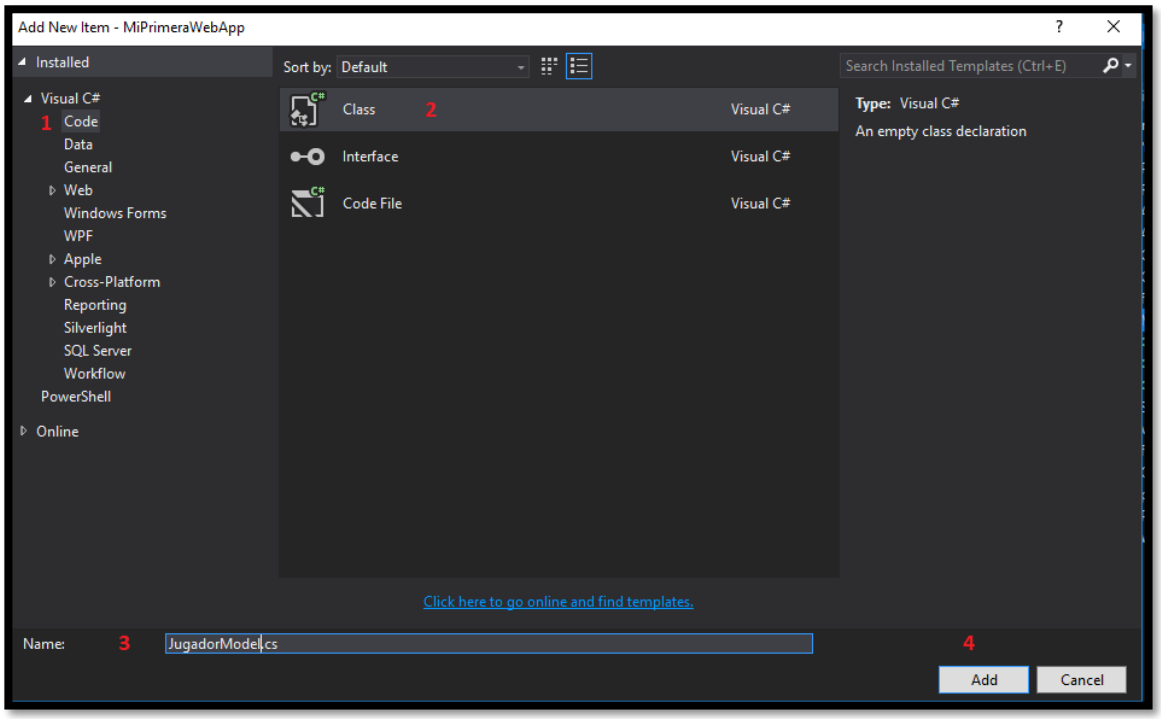
8) Creamos nuestra Segunda clase **EquipoModel.cs**



9) Agregamos los atributos correspondientes a **EquipoModel.cs**

```
using System.ComponentModel.DataAnnotations;                    1

namespace MiPrimeraWebApp.Models
{
    0 references
    public class EquipoModel
    {
        [Key]
        0 references
        public int IDEquipo { get; set; }
        0 references
        public string Nombre { get; set; }
        0 references
        public int  copas { get; set; }                          2
        0 references
        public int IDPais { get; set; }
        0 references
        public PaisModel Pais { get; set; }
    }
}
```

10) Creamos nuestra Tercera clase **JugadorModel.cs**

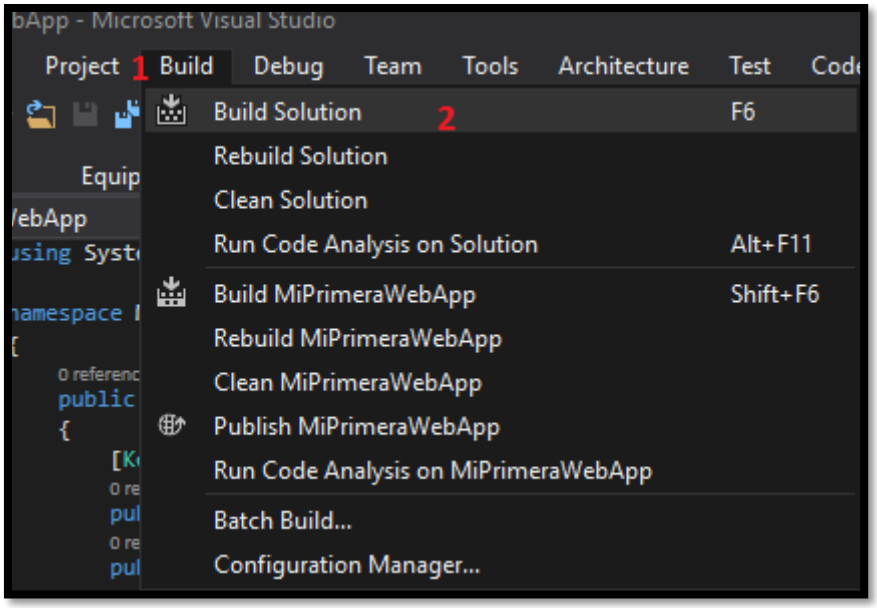


11) Agregamos los atributos correspondientes a **JugadorModel.cs**

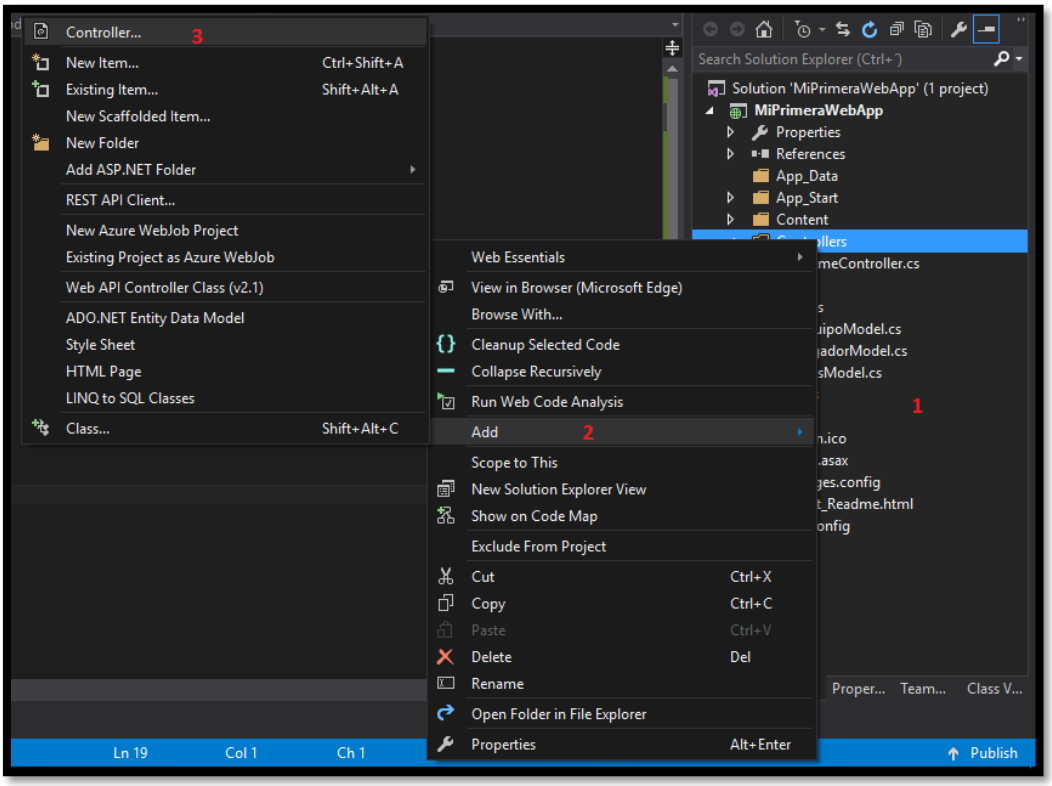
```
using System.ComponentModel.DataAnnotations;

namespace MiPrimeraWebApp.Models
{
    7 references
    public class JugadorModel
    {
        [Key]
        0 references
        public int IDJugador { get; set; }
        0 references
        public string Nombre { get; set; }
        0 references
        public int Edad { get; set; }
        0 references
        public string Posicion { get; set; }
        0 references
        public double Estatura { get; set; }
        3 references
        public int IDEquipo { get; set; }
        1 reference
        public EquipoModel Equipo { get; set; }
    }
}
```

12) Hacemos Build o Generamos nuevamente nuestra solución del proyecto.

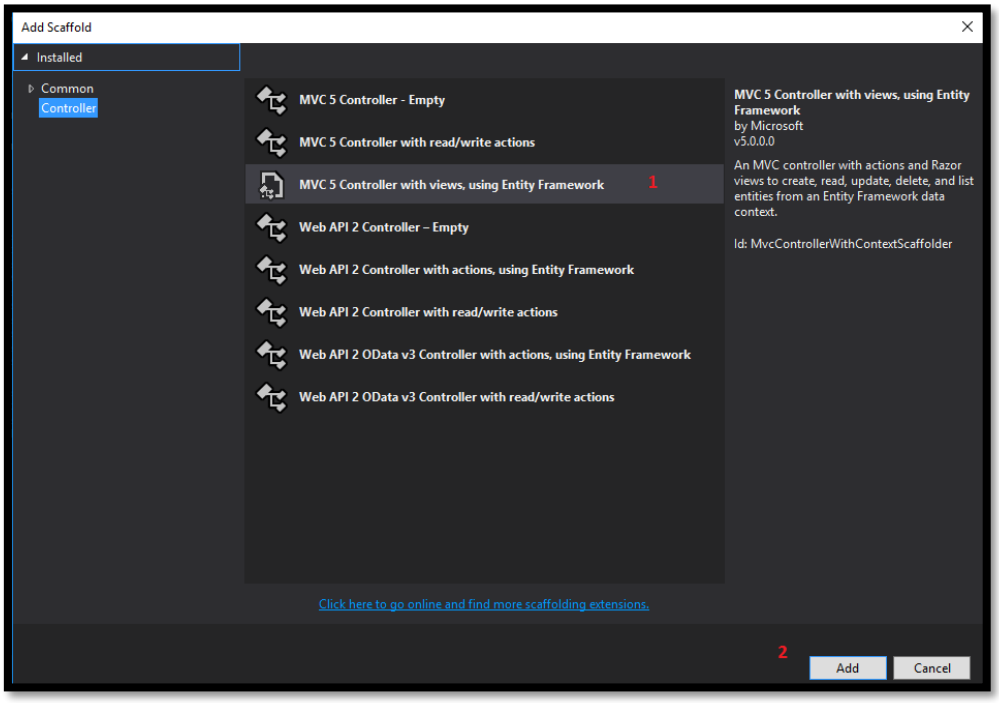
Nota: Esto es importante para que, al momento de crear nuestros controladores pueda reconocer nuestros modelos y generar nuestros controladores sin ningún tipo de problemas o errores.



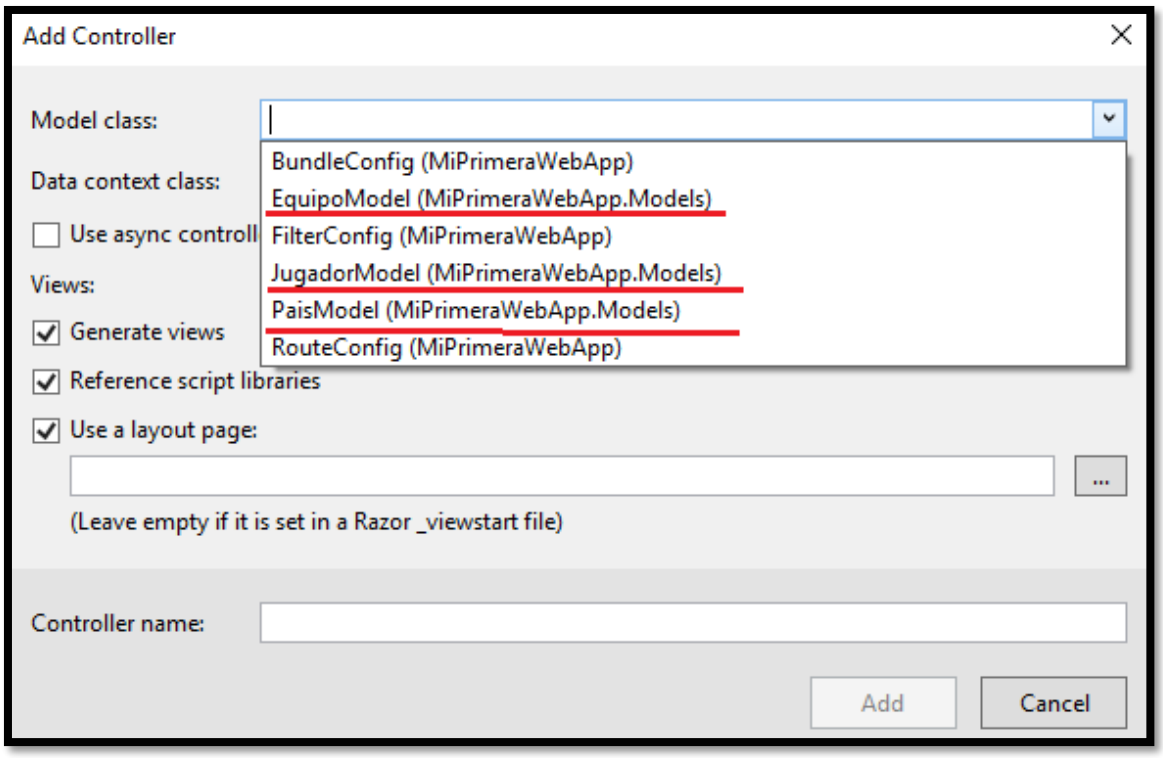
13) Ahora llego el momento de crear nuestros controladores, nuevamente creamos nuestros controladores dentro de la carpeta **Controlle**r:



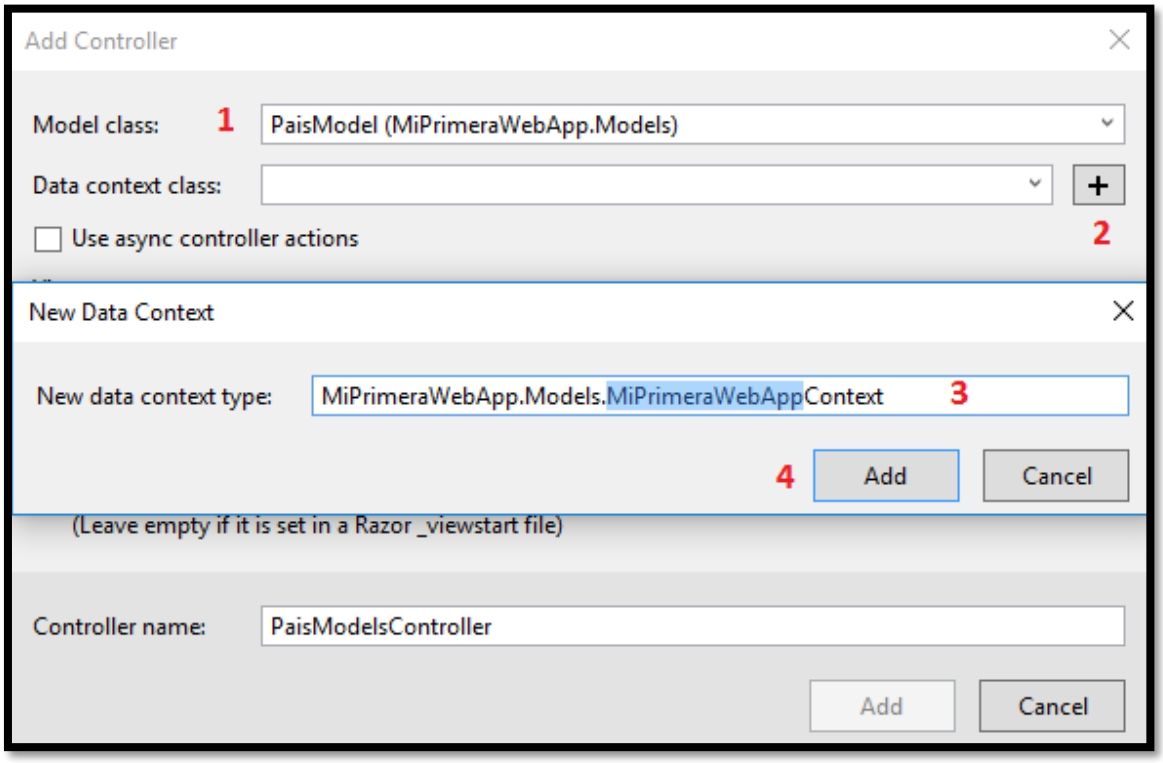
14) Seleccionamos crear controlador MVC 5 o MVC 4 Con vistas y usando Entity Framework.



14) Crean un controlador para **JugadorModel, EquipoModel y PaisModel**

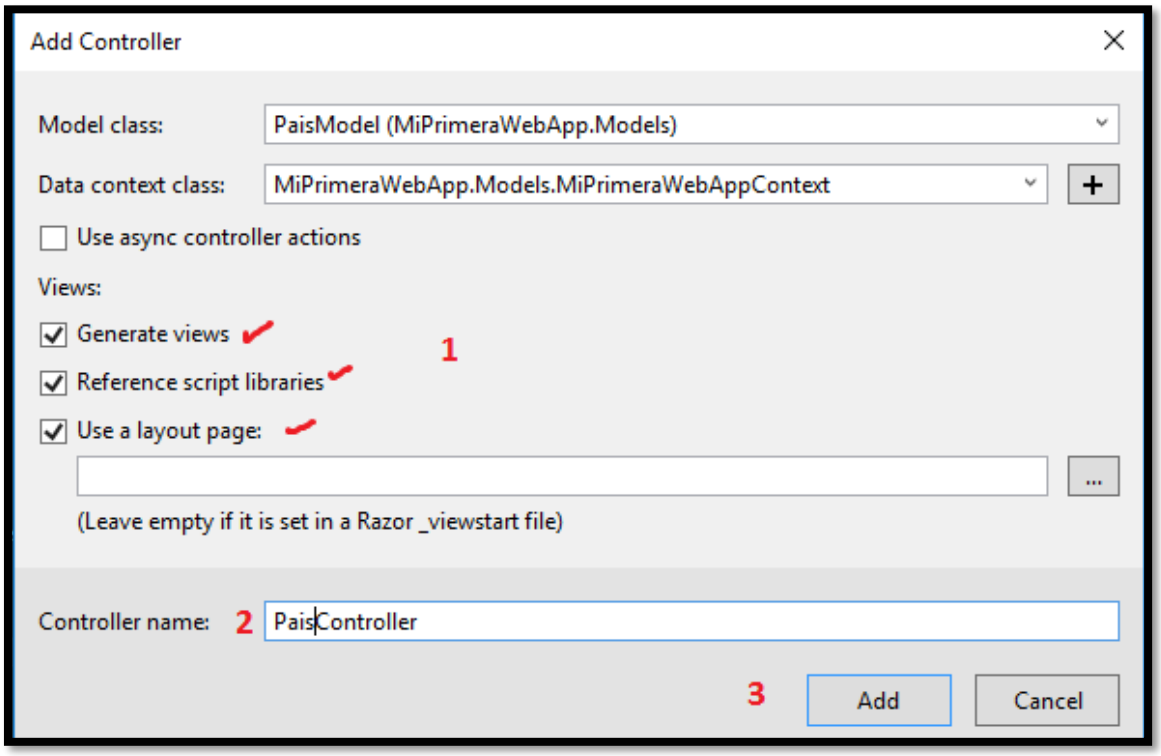


14) Agregamos nuestros Data Context, esto nos permitirá comunicarnos con nuestra base de datos, en este caso seleccionamos la que nos genera por defecto



15) Seleccionamos Generate Views o Generar vistas, agregamos los scripts necesarios para que nuestra página tenga un diseño en específico, y el layout que hace referencia al cuerpo o la parte compartida de nuestra página web.

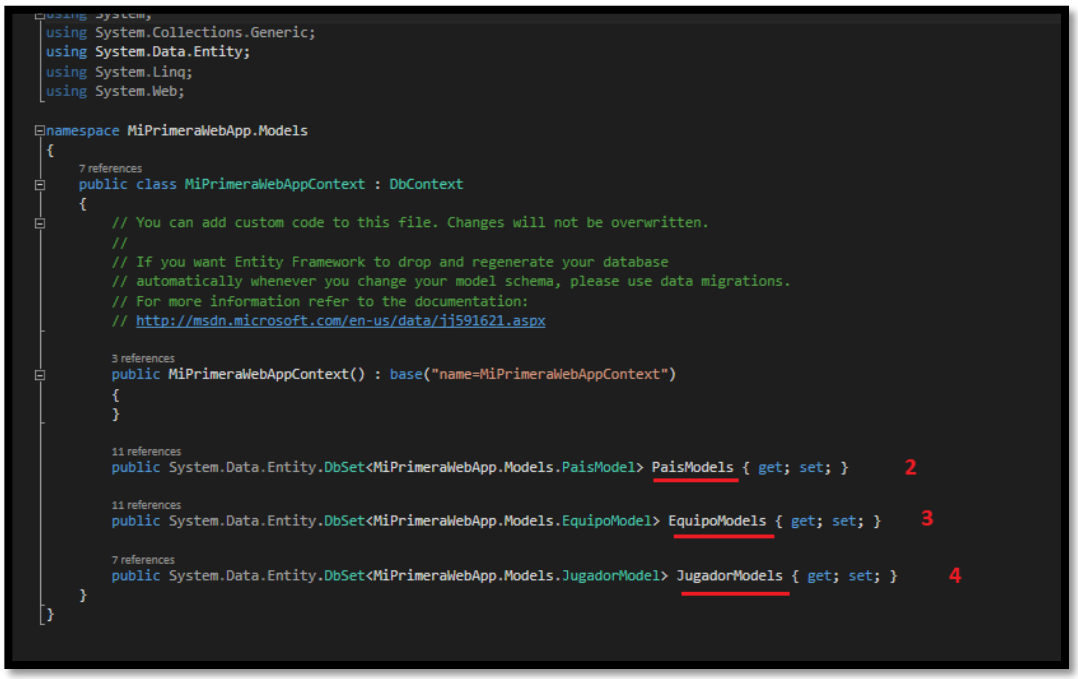
    En este caso no utilizamos controladores asíncronos, ya que los controladores asíncronos son útiles cuando una acción debe realizar varias operaciones de larga ejecución independientes y para este ejemplo no hará falta.



De La misma forma crean los 2 controladores restantes, en este caso hace falta crear para JugadorModel y EquipoModel.

Nota: En Data Context class, seleccionan la misma que ya habían registrado

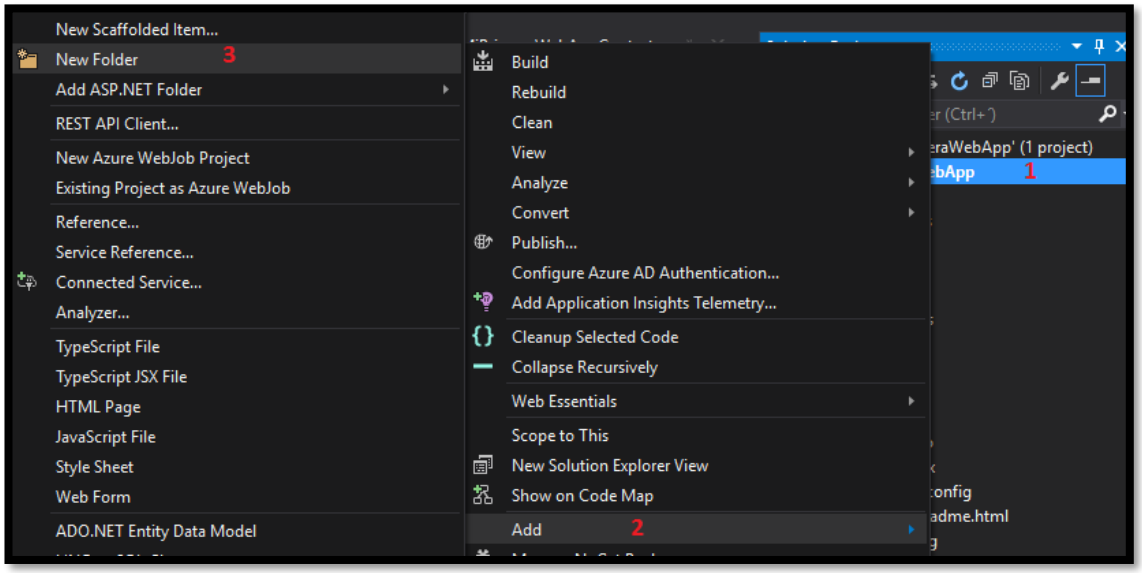
16) Esta clase nos permite realizar nuestra conexión a nuestra base de datos y realizar las funciones que necesitemos realizar dentro de nuestra aplicación, es bueno percatarnos que tenemos todo lo que necesitamos para poder iniciar nuestro proyecto, esta clase nos genera automáticamente, así que no se preocupen:')



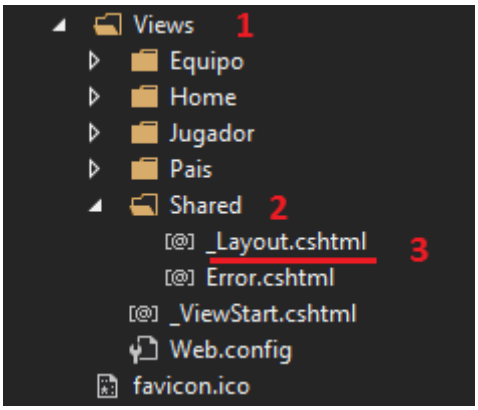
17) Seleccionamos la raíz de nuestro proyecto y creamos una carpeta con el nombre "Images" y dentro de carpeta Images pegamos o agregamos una imagen, en este caso yo descargare:

http://www-asp.azureedge.net/v-2016-06-24-004/images/ui/asplogo-square.png

renombramos el nombre de la imagen por logo.png y lo guardamos.



18) Nos dirigimos a **Shared**, esta carpeta representa la parte "Compartida" y hace referencia a que todo el cuerpo de nuestra web va a ser estático (nunca va a cambiar) a diferencia de la parte dinámica, que vendría a ser solo el contenido de nuestra web y eso lo encontramos en **_Layout.cshtml**

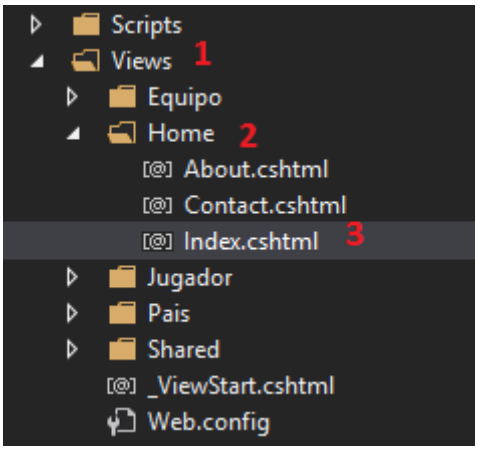


19) No hay mucho que hablar aquí, solo agregar unas pequeñas líneas de código para mostrar y comprobar algunos cambios dentro de nuestra página,

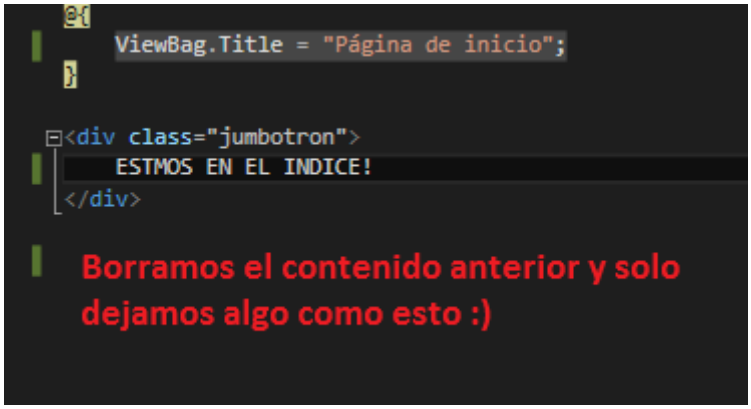
@RenderBody: En las páginas de diseño, hace que la porción de una página de contenido que no está dentro de una sección denominada o hace referencia al contenido o la parte dinámica de nuestra pagina

```
<div class="container body-content">

    <h3>Registro de Equipos y Jugadores de Futbol</h3>     1

    <img src="~/Images/logo.png" alt="Logo de Futbol" />     2

    <br/>  3

    @RenderBody()
    <hr />
    <footer>
        <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
    </footer>
</div>
```

20) Entramos a **Views/Home/** y editamos nuestro **index.cshtml**



21) Por último, editamos y agregamos algo como esto, tampoco entraremos mucho en detalle, eso es parte de HTML5 y conocimiento básico para generar una página sencilla.



22) Ejecutamos el proyecto, nos redirección a la página raíz de nuestro proyecto, en este caso es el índice.

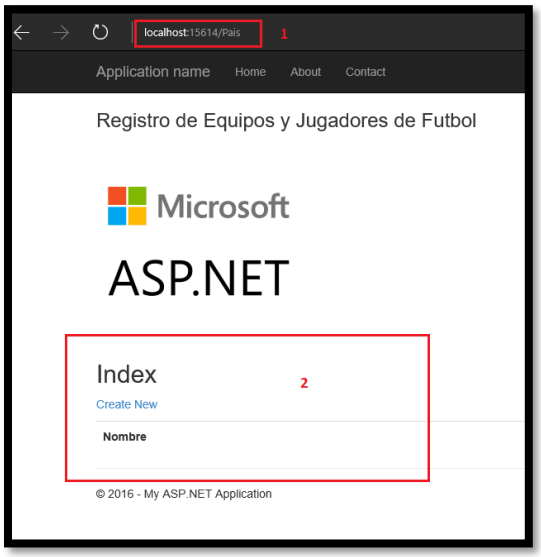
23) Ahora en la parte de la dirección agregamos "/Pais" para verificar si nos generó nuestras vistas.

Nota: la dirección que estoy utilizando es **localhost:15614/Pais,**

Localhost: se refiere a que estoy corriendo mi servicio localmente

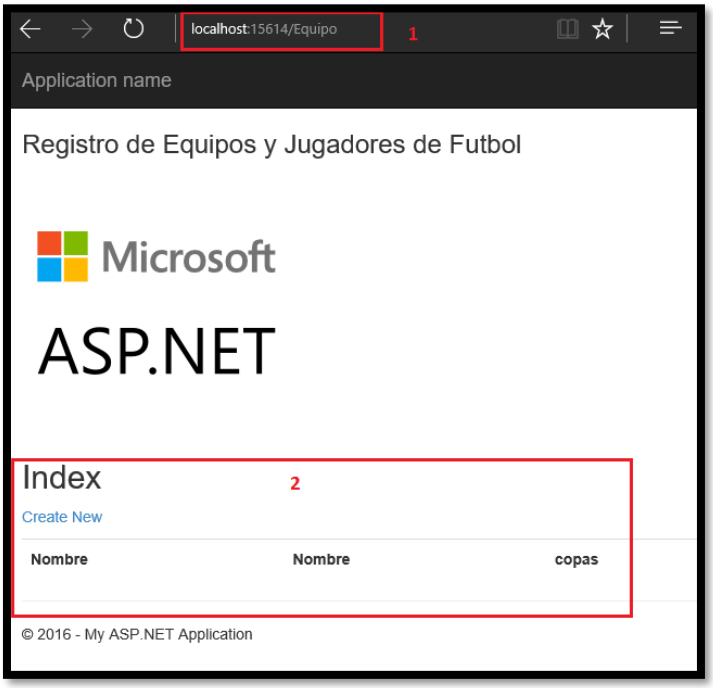
15614: Es el puerto que me habilito IIS(Internet Information Service), a ustedes les debería de generar otro puerto y con ese puerto abren su página web desde Visual Studio

País: Nos referimos a nuestro controlador



24) De la misma forma agregamos en la ruta de dirección "/Equipo".

Ejemplo: **localhost:15614/Equipo**



26) Y por último agregamos en la ruta de dirección "/Jugador".

Ejemplo: **localhost:15614/Jugador**