

**Universidad Israel**

**DEPARTAMENTO:** Ciencias de la Ingeniería **PROFESOR:** Mg. Luis Fernando Aguas B.

**CARRERA:** Sistemas de Información **ESTUDIANTE:** Marco Antonio Ayala Lituma

**CURSO:** Octavo **PARALELO:** “A“ **DESCRIPCIÓN:** Tarea Semana 6

**ASIGNATURA:** Plataformas de Desarrollo 2

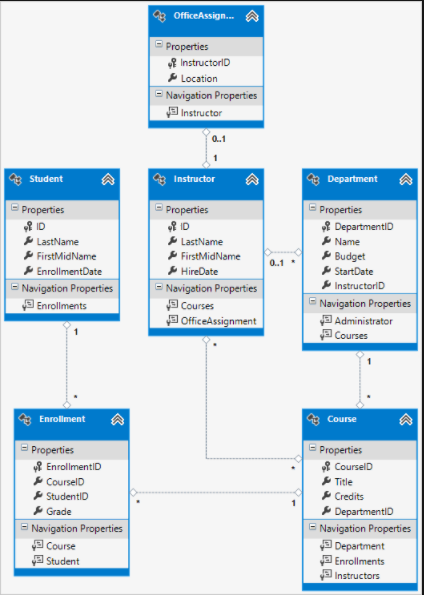
**TEMA:**

Investigue un código en C#, en el cual se aplique MVC

**DESARROLLO:**

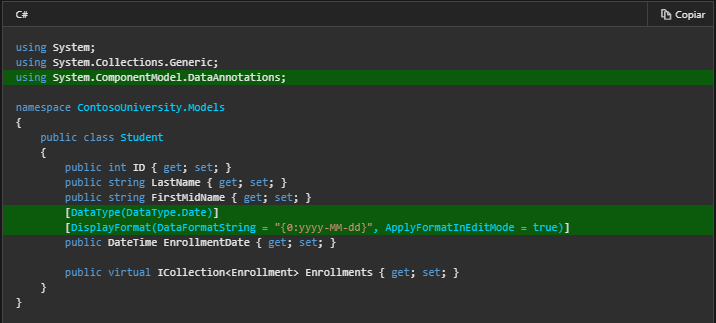
Vamos a investigar el código fuente del proyecto por ejemplo de ContosoUniversity de la publicación de Microsoft.

Empezamos analizando el diagrama de clases del proyecto.

****

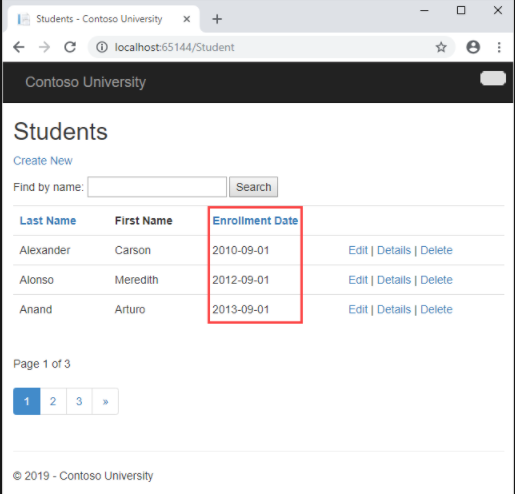
Aquí podemos visualizar el modelo de la clase Student la misma que contiene algunas etiquetas interesantes como los tipos de dato con DataAnnotations.

El atributo DataType se utiliza para especificar un tipo de datos más específico que el tipo intrínseco de la base de datos. En este caso solo se quiere realizar el seguimiento de la fecha, no de la fecha y la hora. La enumeración DataType proporciona muchos tipos de datos, como Date, Time, PhoneNumber, Currency, EmailAddress , etc. El atributo DataType también puede permitir que la aplicación proporcione automáticamente características específicas del tipo. Por ejemplo, se puede crear un vínculo de mailto: para DataType. EmailAddressy se puede proporcionar un selector de fecha para DataType. Date en exploradores compatibles con HTML5. Los atributos DataType emiten atributos de datos HTML 5 (pronunciados con guiones de datos) que los exploradores HTML 5 pueden comprender. Los atributos DataType no proporcionan ninguna validación.

****

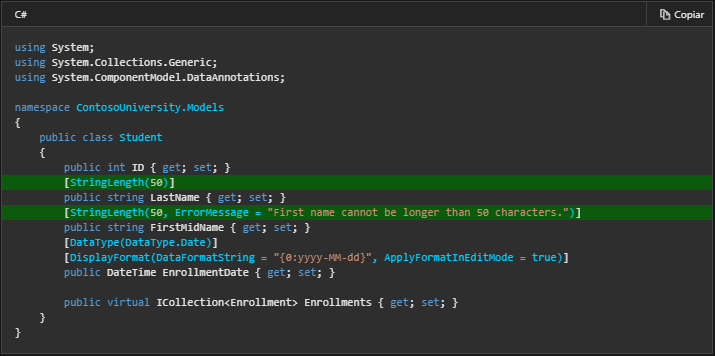
Una visualización rápida del código generado de la vista Student.

El valor ApplyFormatInEditMode especifica que el formato especificado también debe aplicarse cuando el valor se muestra en un cuadro de texto para su edición. (Es posible que no desee que en algunos campos, por ejemplo, para los valores de moneda, no desee que el símbolo de moneda en el cuadro de texto se edite).

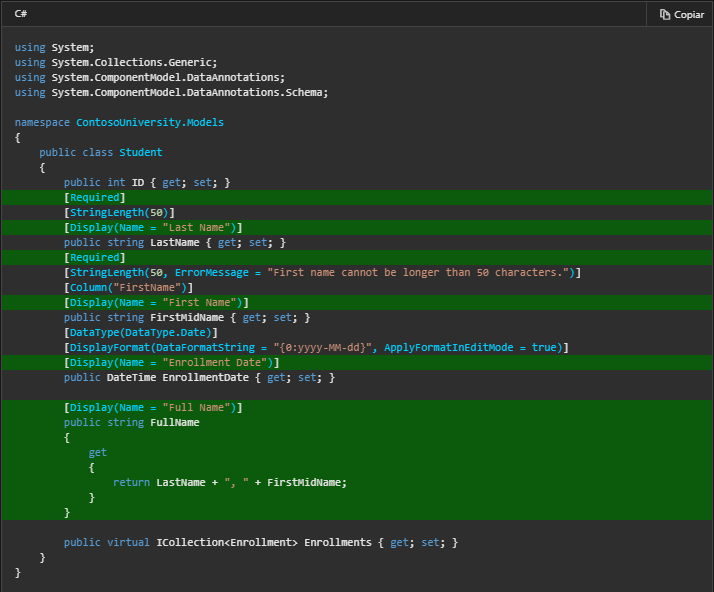
****

También puede especificar reglas de validación de datos y mensajes de error de validación mediante atributos. El atributo StringLength establece la longitud máxima en la base de datos y proporciona la validación del lado cliente y del lado servidor para ASP.NET MVC. En este atributo también se puede especificar la longitud mínima de la cadena, pero el valor mínimo no influye en el esquema de la base de datos.

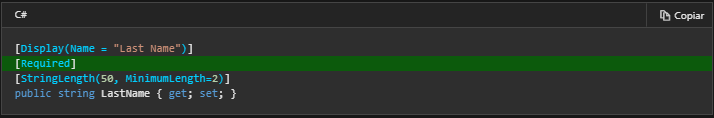
Imagine que quiere asegurarse de que los usuarios no escriban más de 50 caracteres para un nombre. Para agregar esta limitación, agregue los atributos StringLength a las propiedades LastName y FirstMidName, como se muestra en el ejemplo siguiente:

****

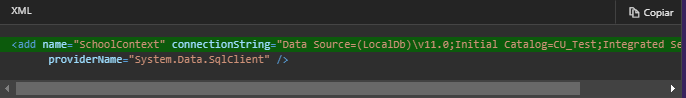
Ahora en la clase Student editamos con los siguientes campos y atributos personalizados.

****

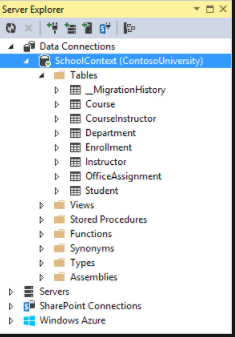
Revisamos el atributo required hace que las propiedades de nombre sean campos obligatorios. El Required attribute no es necesario para los tipos de valor como DateTime, int, Double y Float. A los tipos de valor no se les puede asignar un valor null, por lo que se tratan de forma inherente como campos obligatorios.

****

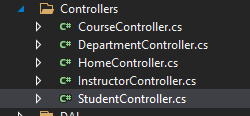
Para generar la base de datos se puede actualizar el modelos de la base de datos con la cadena de conexión.

****

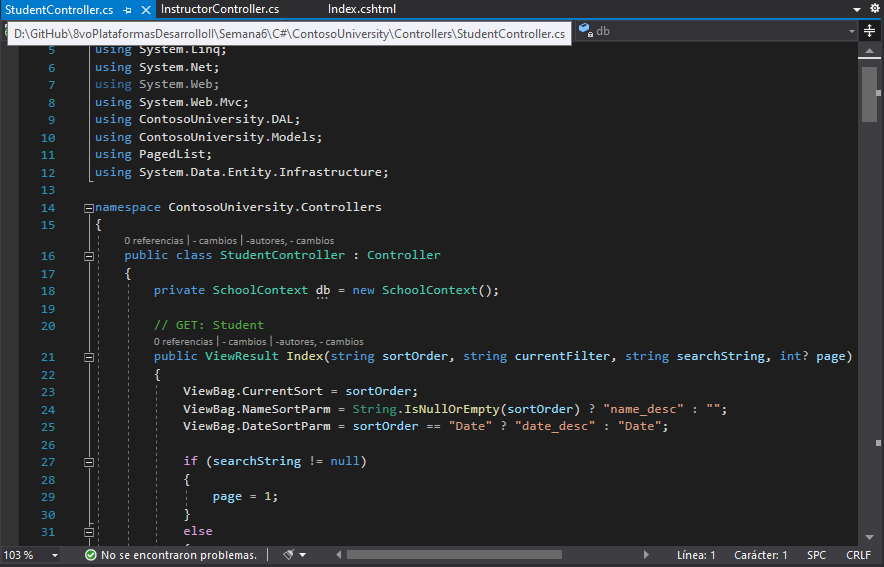
El modelo de la base de datos ya se genera automáticamente con el script.

****

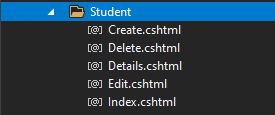
Ahora revisamos que esta carpeta de Controladores contiene la clase StudentControler.



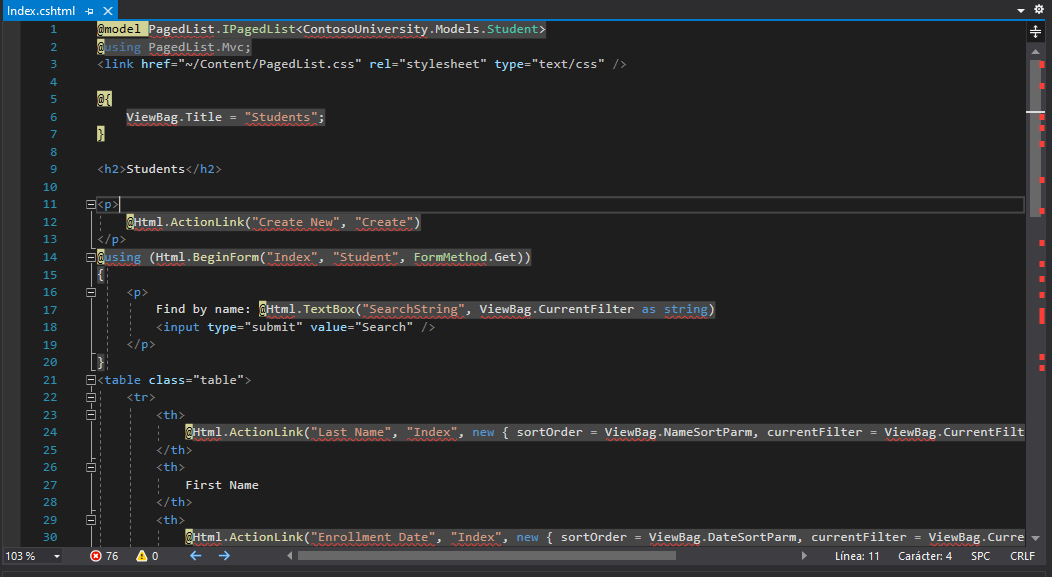
Si revisamos el archivo podemos visualizar el código fuente.



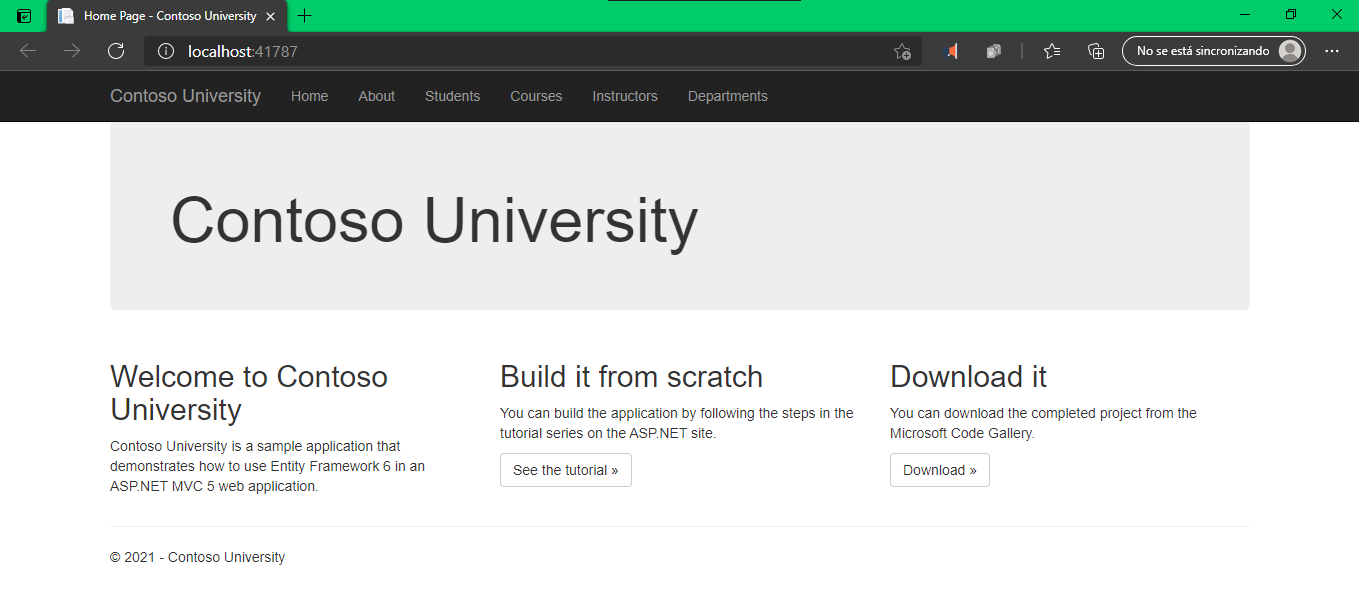
En esta parte s encuentra generado vistas de la clase de Student del modelo



Si abrimos el archivo index de student se visualiza de esta manera, las misma que maneja etiquetas de razor.



El proyecto ejecutándose se visualiza de esta manera

****

**COMENTARIO:**

La rama de la ingeniería del software se preocupa por crear procesos que aseguren calidad en los programas que se realizan y esa calidad atiende a diversos parámetros que son deseables para todo desarrollo, como la estructuración de los programas o reutilización del código, lo que debe influir positivamente en la facilidad de desarrollo y el mantenimiento.

Los ingenieros del software se dedican a estudiar de qué manera se pueden mejorar los procesos de creación de software y una de las soluciones a las que han llegado es la arquitectura basada en capas que separan el código en función de sus responsabilidades o conceptos. Por tanto, cuando estudiamos MVC lo primero que tenemos que saber que son buenas practicas.

**BIBLIOGRAFÍA:**

Creación aplicación ASP.NET MVC, recuperador en.

[Tutorial: creación de un modelo de datos más complejo para una aplicación ASP.NET MVC | Microsoft Docs](https://docs.microsoft.com/es-es/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/creating-a-more-complex-data-model-for-an-asp-net-mvc-application)