

Tecnologías de la Información y la Comunicación

Programación de aplicaciones

“Práctica de patrones MVC”

Florentino Daniel Acosta Nava

Marco Antonio Sarabia Barrón

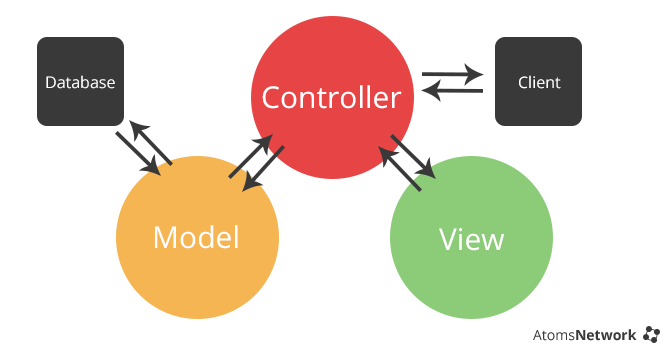
José Manuel Galván Pizano

9° “B”

MI. Ariana Gómez Contreras

23 de agosto de 2019

**Buenas prácticas en programa en C# con MVC**



**Definición de variables por default.**

La declaración de las variables locales de un método, deberá realizarse en el bloque superior de éste. Todas las variables deben estar declaradas en el mismo lugar ósea asta la parte de arriba y deben ser inicializadas para limpiar su valor predeterminado, así se definen las variables que el programa tiene que llevar por default apara que pueda arrancar y tenga un correcto funcionamiento sin esto el programa no podría ser capaz de ni siquiera compilar

**Declaración de variables.**

Declaración de variables C# tipos definidos, ya que muchas veces usamos los tipos definidos en las clases de Framework y podríamos confundirnos Asi que se declara todas las variables necesarias para las funciones necesarias del programa como mostrar las marcas de los coches, su color, etc., etc.

**Métodos y propiedades.**

Se utilizo lo que fueron de preferencia métodos privados y públicos, aunque no es muy recomendado utilizar de origen publico a menos que se necesite acceder de la parte de afuera de la clase.

**Indentación, espaciado y Agrupación de código.**

* Los comentarios están en el mismo nivel que el código, es decir que han de utilizar el mismo nivel de indentación.
* Las llaves de apertura y cierre están en el mismo nivel que el código fuera de éstas.
* Las llaves están en una línea independiente y no en la misma línea que if, for, etc.
* Todo lo anterior para el buen entendimiento y visualización del código a la hora de estar observándolo ya sea para corregir errores para añadir nuevas características, etc.

**Comentarios y notas en código (documentación).**

Se utiliza los comentarios de la manera correcta con // para mostrar que son comentarios y así no los pueda tomar C#, todo esto con un orden y una gramática correcta al igual de una explicación concreta y entendible de lo que se hizo y nos sea de ayuda para no ser de tormento en tratar de entender código y comentario.

**Seguridad**

Para tener seguridad en el manejo de la información se bloquea el código para que una vez en ejecución mediante el MVC no pueda mostrar más que la vista codificando todo este utilizando clases privadas a la medida de lo posible para no compartir toda la información en publicas y siendo un código mas limpio y mas fluido para evitar fugas o fallas de información al codificarlo.

**Práctica**

Se ha realizado una práctica del MVC, donde se crea una lista de coches.

Como se puede ver en la siguiente imagen, el “Explorador de Soluciones” muestra las partes del MVC, en las carpetas “Controllers”, “Models” y “Views”.

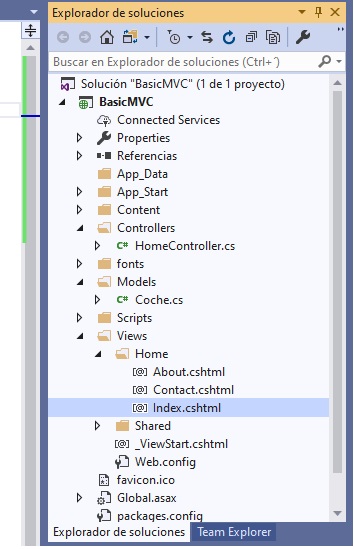


Imagen - Explorador de Soluciones en Visual Estudio. Se muestran las carpetas contenedoras de los archivos para el MVC.

En la carpeta “Models” se crea una clase (nuestro modelo) que se llama “Coche” y le agregamos las propiedades que tendrá tal clase.

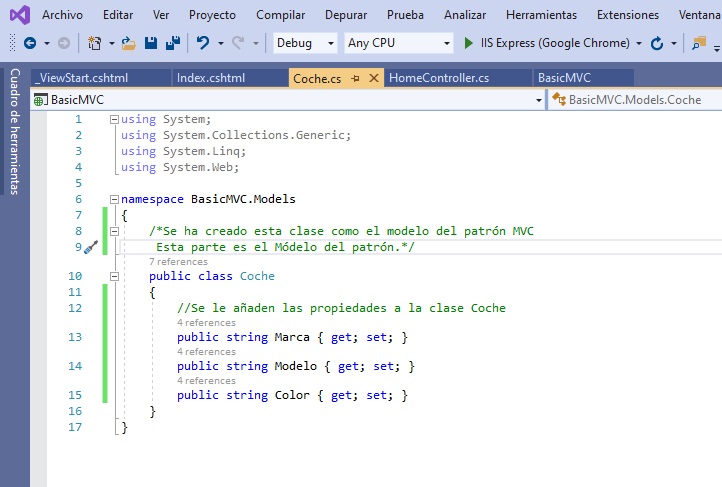


Imagen - La clase Coche es el modelo del patrón. Donde simplemente se le asignan sus propiedades, pero no se ha creado ningún objeto de tal clase.

En nuestro controlador “Controllers/HomeController.cs” creamos la lista correspondiente donde instanciamos todos los coches que la vista va a mostrar. En esta parte, le asignamos a las propiedades de los coches sus valores correspondientes:

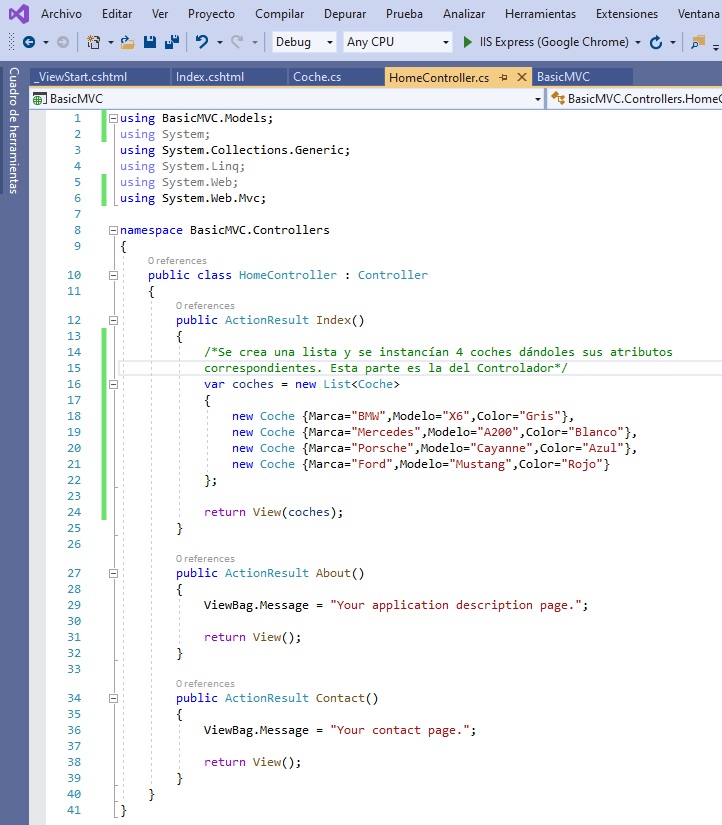


Imagen - En el controlador, se instancian los objetos que se quieren mostrar en la vista. Se le asignan los valores a las propiedades de cada objeto de la clase Coche.

Como se dijo anteriormente, la vista es la encargada de mostrar el contenido que el controlador le permite:

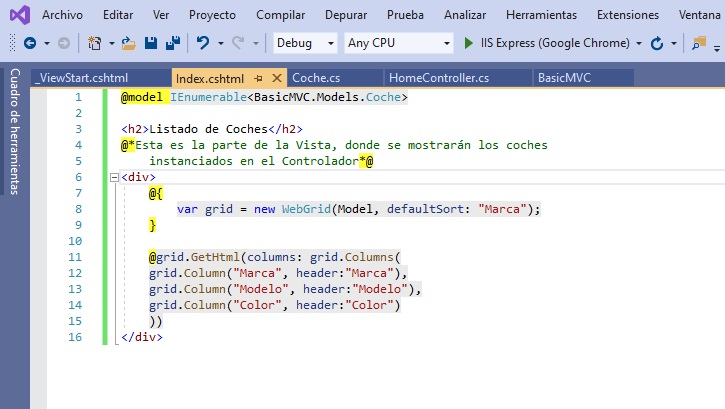


Imagen - En la vista, se hace referencia a la lista del Controlador, de manera que sea lo único que se muestre en pantalla.

Una vez ejecutado el programa, se abre el navegador que muestra lo siguiente:

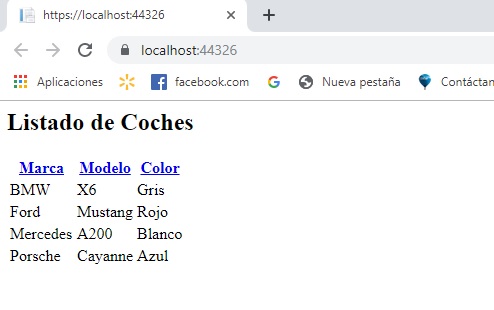


Imagen - Al ejecutar el programa, el navegador muestra lo que la vista tiene para mostrar, de acuerdo a lo que el controlador le permite.