# Proyectos web

## Tipos

1. Websites
   1. Mezclan el código de servidor y el de la interfaz, similar a lo que se hacia en asp classic o PHP
2. Webform
   1. Tiene un estilo de programación similar al de las aplicaciones de escritorio
   2. El estado de los controles de la pagina se guarda en un campo hidden llamado viewstate
3. MVC
   1. Puede utilizar el motor de vistas webform o razor
   2. Utiliza el patrón Model View Controller para separar las resposabilidades y mejorar la reutilización

### Versiones de MVC

Version 1 disponible en 2009 en asp.net 3.5

Version 2 disponible en 2010 en asp.net 3.5

* Data annotations
* Validadores de modelo
* Validadores del lado del cliente
* Soporte para asp.net 4.0

Version 3 disponible en 2011 en asp.net 4

* Motor de vistas razor
* Validaciones mediante jquery
* Soporte para entity framework code first

Version 4 disponible en 2012 en asp.net 4

* Web Api
* Incorpora la funcionalidad de bundle y minification en tiempo de ejecución

## Componentes

Controlador: es una clase que hereda de la clase Controller de asp.net y que contiene una método para cada ruta que queremos resolver en el mismo

Vistas: son archivos escritos en razor que permiten visualizar datos, generar formfularios, etc. Estas vista se renderizan como código html estándar cuando son enviadas al cliente

Modelo: en este componente encontramos 2 diferencias.

* Modelo de datos: son las clases con las que intereactuamos con nuestro repositorio de datos y obtenemos entidades de negocio
* Modelo de vista: es una clase que contiene propiedades que son utilizadas en las vista para mostrar o completar datos

Controladores WebApi: son similares a los anteriores pero utilizan los verbos del protocolo HTTP para ejecutar los métodos correspondientes

Motor de vistas razor

1. Para diferenciar entre el código html y el código razor utilizamos @
2. Las estucturas de control (if,for,while, etc) deben escribirse con llaves por mas que sean de una sola línea
3. Razor nos expone algunos métodos para ayudarnos a escribir las vistas
   1. ActionLink nos permite hacer un link a un método de un controlador
   2. BeginForm y EndForm nos sirven para crear formularios
   3. TextArea, TextBox, CheckBox, RadioButton, ListBox, DropDownList, Hidden, Password nos sirven para crear los campos del formulario
   4. Action este método nos sirve para ejecutar una acción del controlador e incluir el resultado en la vista actual
   5. Partial nos permite incluir otra vista dentro de la actual sin necesidad de pasar por el controlador

Modelo

El modelo nos permite agregar a las propiedades algunas marcas para mejorar la validación

* Required
* DisplayName
* StringLength
* Range
* RegularExpression
* EmailAddress
* Url

Cada uno de estos atributos permite configurar el mensaje de error devuelto mediante la propiedad ErrorMessage

# Entity Framework

## Versiones

EF 3.5 Fue la versión inicial incluida en el framework 3.5

EF 4 Incluida en el framework 4, incluye la compatibilidad con clases poco, carga diferida y la forma de trabajo model first

EF 4.1 se separa del framework de .net incluye un nuevo dbContext y la forma de trabajo code first

EF 4.2 correccion de errores

EF 4.3 incluye la posibilidad de gestionar los cambios en el modelo mediante las Migrations

EF 5 incluye la compatibilidad con los frameworks 4 y 4.5, compatibilidad con enumerados

EF 6 soporte para SP para insertar, actualizar y borrar, soporte para interceptores

EF7 actualmente está en beta, soporte para bases de datos no relacionales, soporta proyectos que no están escritos en full framework

## Componenetes

Contexto: es lo que nos permite manejar la conexión con la base de datos y el comportamiento de EF, solo es necesario crear una clase que herede de DbContext

Dbsets: dbset es una clase genérica que nos permite decirle al EF que entidades queremos conectar con la base de datos

## DataAnnotations

Para poder modificar el comportamiento de EF utilizamos atributos sobre las tablas y los campos para describirlos

* Table se usa sobre una clase para indicar el nombre de la tabla que representa
* MaxLength se utiliza en campos de texto para indicar su largo
* ConcurrencyCheck marca que propiedades se van a utilizar para controlar la concurrencia
* TimeStamp si se usa una columna timestamp para controlar la concurrencia
* Key para indicar la clave principal de la tabla

## Configuracion

Para decirle a EF a que base de datos debe conectarse tenemos varias opciones

1. Por defecto EF va a intentar conectarse con la cadena de conexión que este nombrada como DefaultConection en el archivo de configuración.
2. Podemos en el constructor de nuestra clase contexto llamar al constructor de dbcontext y pasarle el nombre de la cadena de conexión que queremos utilizar.
3. Podemos en el constructor de nuestra clase contexto llamar al constructor de dbcontext y pasarle la cadena de conexión que queremos utilizar.