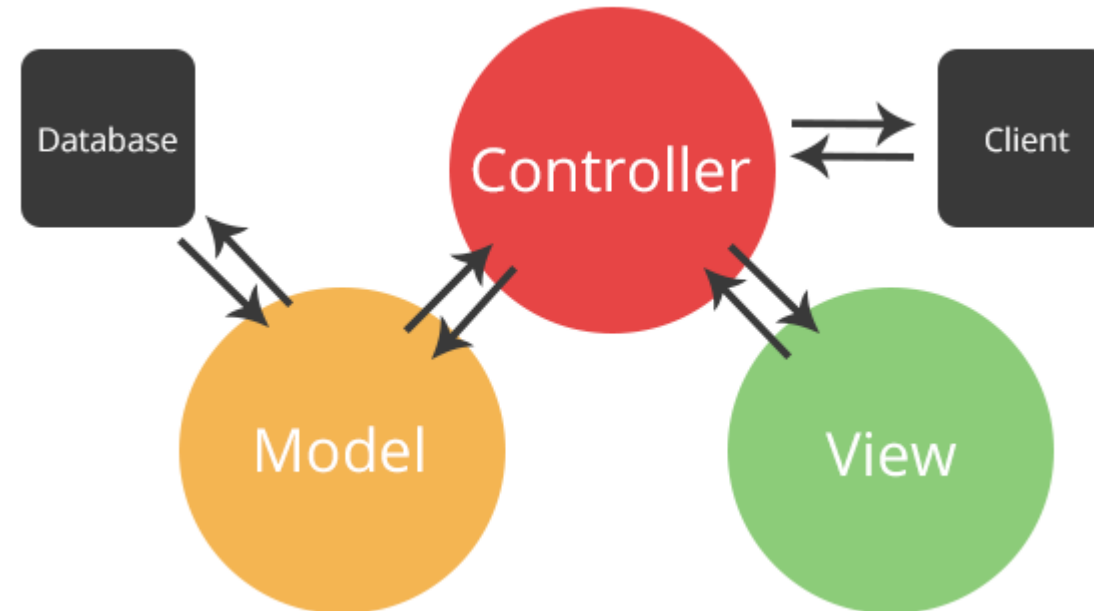


# PATRÓN MVC CON .NET8



# EL PATRÓN MVC EN .NET8

MVC es un patrón, no una tecnología.

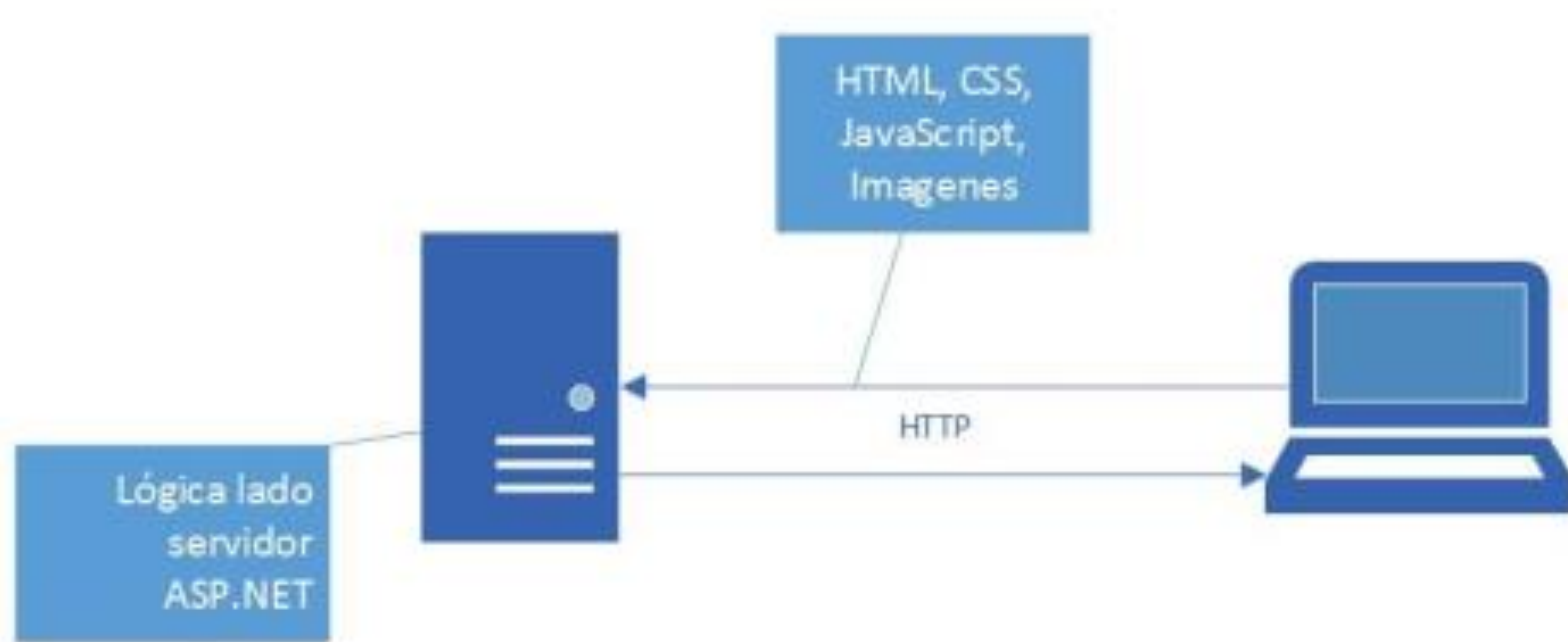
**Vista:** La vista es el conjunto de markup de la aplicación, es decir el HTML. En ASP.NET MVC este markup es pre renderizado a través de una tecnología llamada Razor, la cual permite, entre otras cosas, ejecutar lógica previa a renderizar el HTML que será enviado al cliente.

**Modelo:** Son los objetos de dominio o negocio de la aplicación, más adelante, veremos que no necesariamente puede estar en el mismo proyecto que la aplicación web.

**Controlador:** El controlador se encarga de recibir solicitudes y ser el punto de entrada de la aplicación, a través de lo que se conoce como acciones (Actions). Se basa en solicitudes HTTP y puede responder con datos o con contenido, como HTML.

# EL PATRÓN MVC EN .NET8

En el desarrollo web, se trabaja con un servidor que responde a archivos e información y un navegador que ejecuta y lee los archivos.



# EL PATRÓN MVC EN .NET8

El tipo de solicitud se define a través de un verbo HTTP que indica el tipo de solicitud, los principales verbos que se deben conocer son:

**GET:** usualmente conocido como verbo “obtener”, no incluye cuerpo.

**POST:** solicitud de envío de información, debe ir acompañado de un cuerpo de contenido, por ejemplo una imagen o un JSON

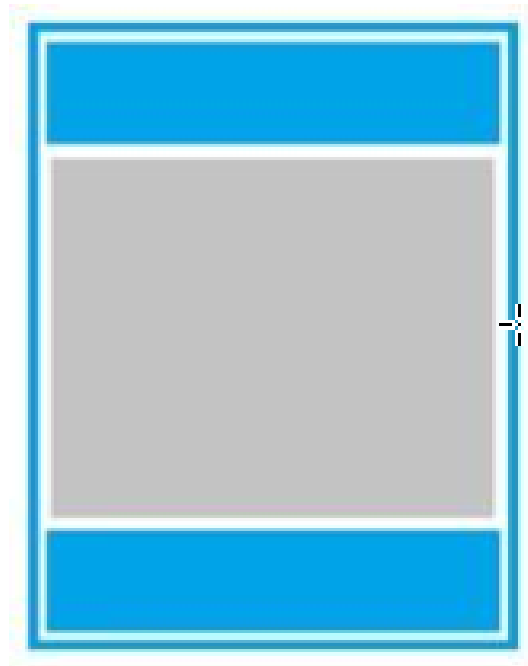
**PUT:** similar al POST, sin embargo se utiliza para cuando se realizan actualización de recursos.

**DELETE:** similar a la estructura al GET, pero el servidor debe entender que se busca remover el recurso que se consulta en lugar de retornarlo.

# IMPLEMENTADO MVC EN .NET6

## VISTA

Todo website tiene ciertos componentes que se repiten entre páginas, tipo “marco”. En Web Forms se le conocía como MasterPage, en MVC se le conoce como `_Layout`. Usualmente estas secciones incluyen partes como encabezados, pie de página y menús de navegación.

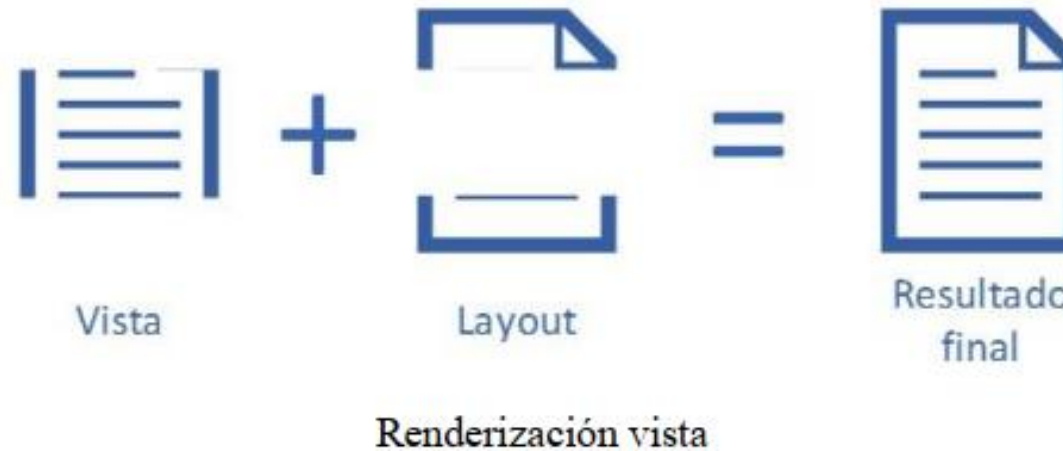


Al investigar un poco más el archivo se puede encontrar la sentencia **@RenderBody()** en esta parte del HTML se renderizará el resto de páginas que utilicen.

# IMPLEMENTADO MVC EN .NET6

## VISTA

La manera de asignar este layout a diferentes páginas es a través de una variable llamada ***Layout***.



La página que fue retornada ya traía un layout y contenido renderizado, esto es porque por defecto cada vista creada tiene un layout por defecto.

Esta configuración se realiza en el archivo `_ViewStart.cshtml`.

# IMPLEMENTADO MVC EN .NET6

## CONTROLADORES

Un controlador se conforma de una clase con métodos que responden a solicitudes HTTP. La clase equivale al nombre del controlador y la función equivale al nombre de la acción.

Nombre de Clase = Controlador

Nombre de Función = Acción

Por ejemplo en el archivo  
Controllers/HomeController

Controlador: Home.

Acción : Index o About.

```
0 references
public class HomeController : Controller
{
    0 references controlador
    public IActionResult Index()
    {
        return View();
    }
    0 references
    public IActionResult About()
    {
        acciones
    }
}
```

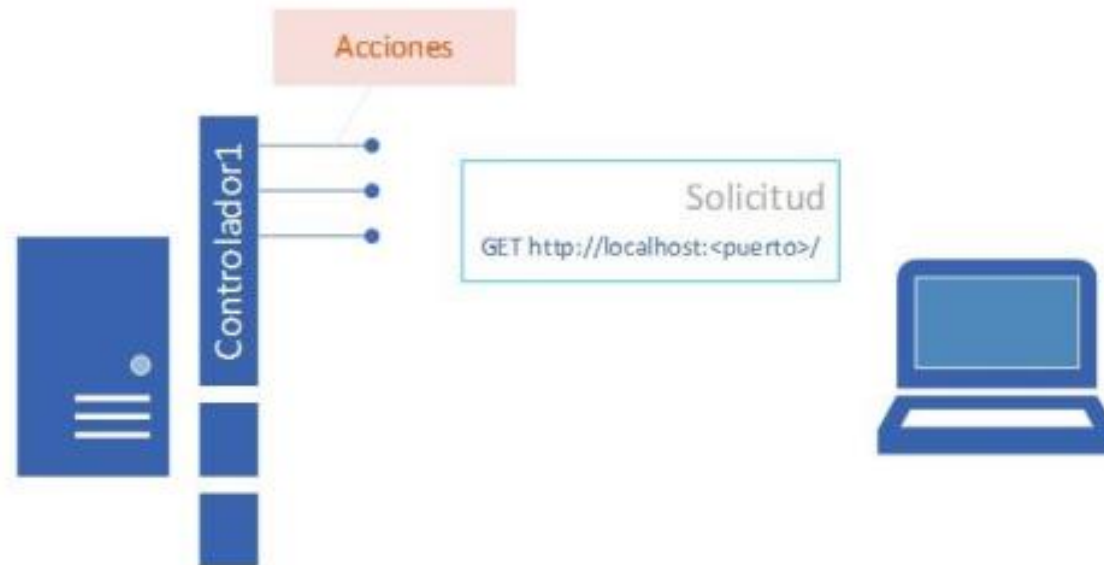
# IMPLEMENTADO MVC EN .NET6

## CONTROLADORES

`http://localhost:<puerto>/Home/Index`

`http://localhost:<puerto>/Home/About`

Así las solicitudes web se resuelven de la siguiente forma:



Procesamiento solicitudes web



# IMPLEMENTADO MVC EN .NET6

## CONTROLADORES

El controlador en el tipo de función define qué tipo de dato se retornará al cliente. Por ejemplo en el homecontroller, todas las acciones responden un tipo de dato ActionResult, en este caso todas las funciones retornan un objeto de tipo View.

```
public IActionResult About()  
{  
    ViewBag.Message = "Your app  
    return View();  
}
```

Plantilla de ruteo

La función **View** se puede colocar sin pasar parámetros por lo cual la función buscará una vista que esté en el folder Views/<Nombre de Controlador>/<Nombre de Acción>. En este caso retornará la vista que esté ubicada en Views/Home/About.

Sin embargo también se puede indicar explícitamente el nombre de la vista que se desea retornar por ejemplo **View("About")**

# IMPLEMENTADO MVC EN .NET8

## MODELOS

Los modelos son clases que representan objetos de dominio y pueden ser utilizados por la lógica de negocio y acceso a datos para manipular información. Están ubicado en la plantilla en el folder Models

```
public class Item
{
    public int Id { get; set; }
    public string Nombre { get; set; }
}
```

# PRÁCTICA

- Implementar una aplicación web con plantilla ASP.NET core MVC Net 8 y agregar una vista razor y acción Persona. Luego de realizar click en el botón enviar, carga la misma vista presentado los datos ingresados.

AppWebEjemplo Home Privacy Personal

### Registra Persona

Nombre

Edad

Enviar

AppWebEjemplo Home Privacy Personal

### Registra Persona

Juan Pérez

25

Enviar

Sr(a) Juan Pérez tiene 25 de edad

# PRÁCTICA

- En la aplicación web anterior, agregar una vista razor y acción Notas. Luego de realizar click en botón Promedio, carga la misma vista presentado los datos ingresados y promediar las 2 notas con mayor nota.

AppWebEjemplo [Home](#) [Privacy](#) [Personal](#)

## Registra Notas