



---

# ACTIVIDAD EVALUABLE

## 3

---

Isidro gutierrez gonzalez



7 DE OCTUBRE DE 2025

# ÍNDICE

Introducción breve del objetivo de la actividad.....2

Descripción de la  
solución.....2

Fragmentos de código  
comentados.....3

Capturas de pantalla con  
evidencias.....5

## **Introducción breve del objetivo de la actividad.**

El objetivo de esta actividad es practicar el desarrollo de aplicaciones web utilizando ASP.NET MVC, integrando la manipulación de datos mediante formularios, arrays de objetos y la generación dinámica de contenido en vistas Razor.

Se busca consolidar conceptos como: manejo de formularios POST, uso de ViewBag y ViewData, bucles @foreach, condicionales @if, y la integración de todas las vistas con un Layout común.

## **Descripción de la solución:**

- ASP.NET MVC: Permite estructurar la aplicación en Modelo-Vista-Controlador, separando la lógica de negocio, la presentación y el flujo de la aplicación.
- C#: Lenguaje de programación para implementar la lógica del controlador y las funciones auxiliares.
- Razor: Motor de vistas de ASP.NET que permite generar HTML dinámico mezclando código C# con etiquetas HTML.

# Fragmentos de código comentados

## Estructura del Layout

Se ha utilizado un **\_Layout.cshtml** común en la carpeta Views/Shared, que define la estructura general de todas las páginas:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>@ViewData["Title"] - Actividad 3</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.
min.css" />
</head>
<body>
    <div class="container mt-4">
        <h1>Actividad 3 - ASP.NET MVC</h1>
        <hr />
        @RenderBody() <!-- Aquí se inserta el contenido de cada vista
-->
    </div>
</body>
</html>
```

- Todas las vistas usan este layout mediante la directiva `Layout = "_Layout";`.
- Permite mantener un estilo consistente en todas las páginas.

## Formulario POST y función auxiliar

- **Formulario HTML** para introducir los datos de un partido de fútbol:

```
<form asp-action="Ejercicio1" method="post">
  <input type="text" name="equipoLocal" required />
  <input type="text" name="equipoVisitante" required />
  <input type="number" name="golesLocal" required />
  <input type="number" name="golesVisitante" required />
  <button type="submit">Enviar</button>
</form>
```

- **Función auxiliar en el controlador:**

```
private string CalcularResultado(string local, string visitante, int
golesLocal, int golesVisitante)
{
    if (golesLocal > golesVisitante)
        return $"Gana {local} ({golesLocal}-{golesVisitante})";
    else if (golesLocal < golesVisitante)
        return $"Gana {visitante} ({golesLocal}-{golesVisitante})";
    else
        return $"Empate ({golesLocal}-{golesVisitante})";
}
```

- Permite determinar el resultado del partido y mostrarlo en la vista usando ViewBag.Resultado.

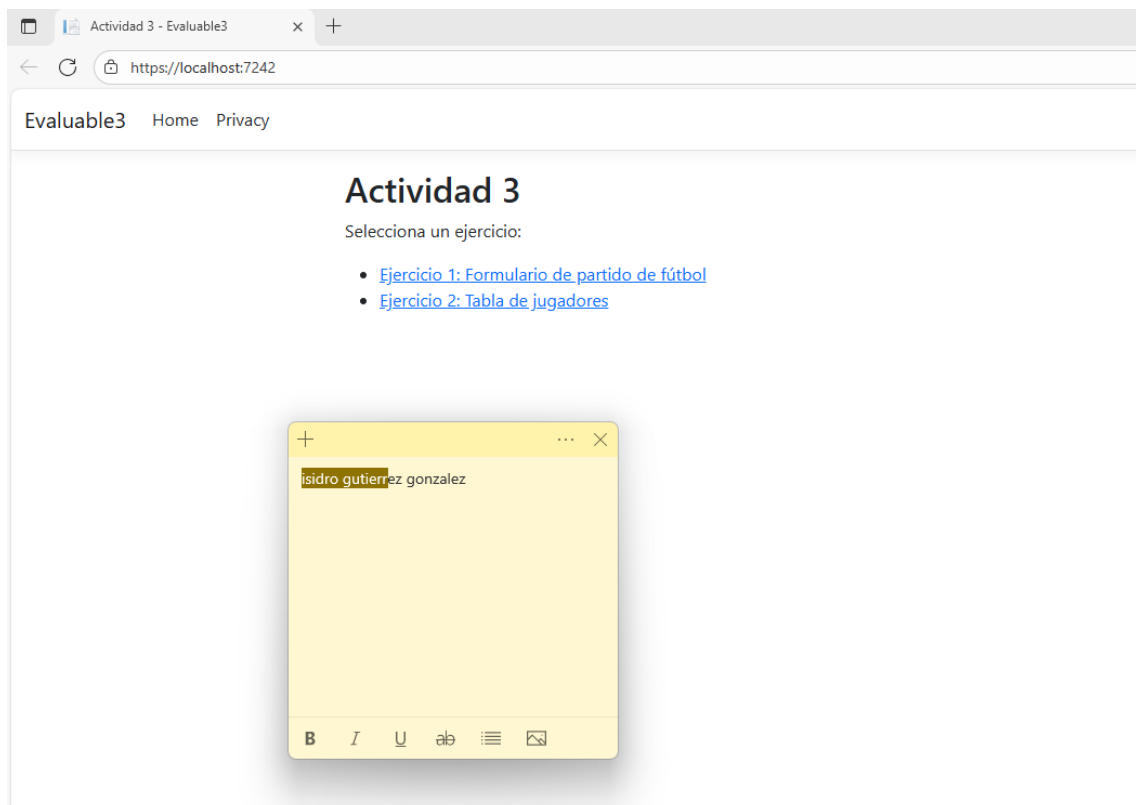
## Vista Ejercicio 2

```
@foreach(var jugador in Model)
{
    <tr style="@ (jugador.Posicion=="Portero"? "background-
color:#FFD700;":"")">
        <td>
            @* Nombre en negrita si dorsal < 10 *@
            @if(jugador.Dorsal < 10)
            {
                <strong>@jugador.Nombre</strong>
            }
            else
            {
                @jugador.Nombre
            }
        </td>
    </tr>
}
```

- ViewBag demuestra paso de datos desde el controlador.

- `@foreach` recorre arrays de objetos.
- `@if` permite aplicar condicionales para resaltar contenido.

## Capturas de pantalla con evidencias



Ejercicio 1 - Partido de fútbol - Ejerc

+

←

↻

https://localhost:7242/Ejercicio/Ejercicio1

Evaluable3HomePrivacy

Formulario de partido de fútbol

Equipo Local:

Getafe

Equipo Visitante:

Almería

Goles Local:

2

Goles Visitante:

3

Enviar

Resultado: Gana Almería (2-3)

Ejercicio 2 - Tabla de jugadores - Ejerc

+

←

↻

https://localhost:7242/Ejercicio/Ejercicio2

☆

☆

Evaluable3HomePrivacy

Tabla de jugadores

Nombre	Dorsal	Posición
Juan	1	Portero
Luis	9	Delantero
Pedro	11	Centrocampista
Miguel	7	Delantero