

Microsoft
Dev Group
{Cobán}

Comparte | Aprende | Desarrolla



.NET Core

.NET Core MVC vs Razor Pages

¿Cuál utilizar?





Hola!

Geovani de León (Yova)

Ingeniero de Software en PensoTec

Docente Universitario –CUNOR–

<https://github.com/yovafree>

<https://yovadeleon.dev>

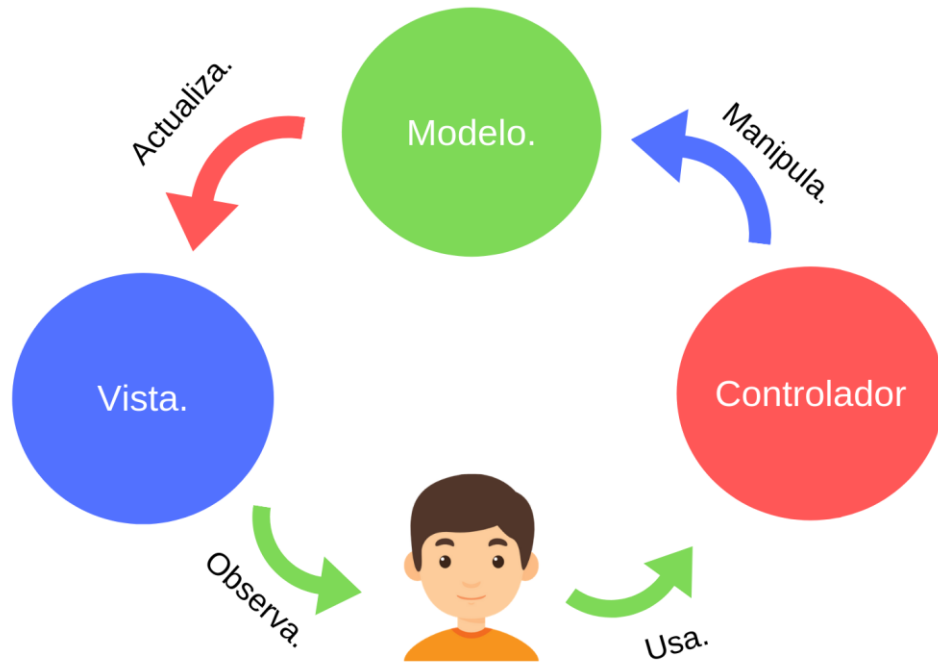
Agenda

- ⊙ Qué es MVC
- ⊙ Conociendo .NET Core MVC
- ⊙ Demo MVC
- ⊙ Razor Pages
- ⊙ Demo Razor Pages
- ⊙ Comparativa .NET Core MVC vs Razor Pages

Asistencia



Patrón MVC – Modelo - Vista - Controlador



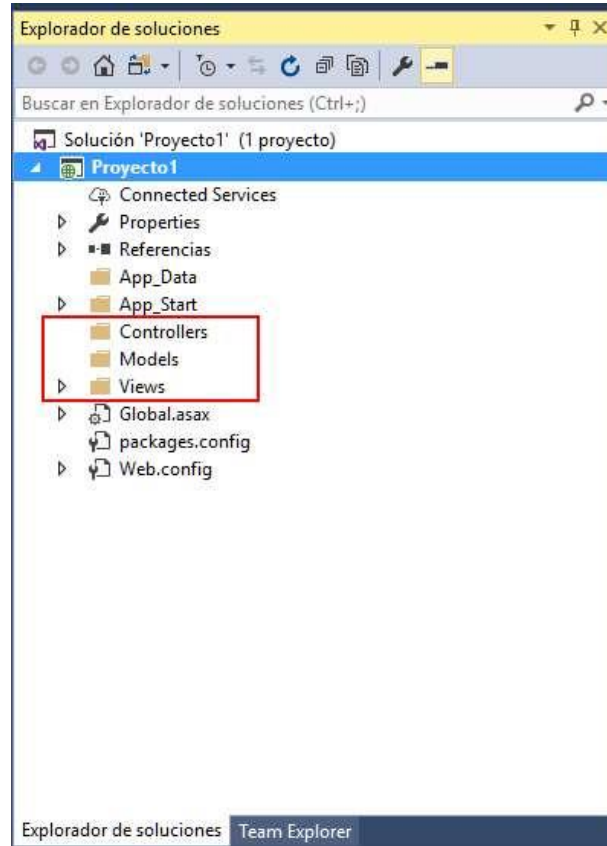
ASP.NET MVC

- ◎ Microsoft tiene dos tecnologías fundamentales para el desarrollo de sitios web:
 - Framework ASP.NET WebForm.
 - Framework ASP.NET MVC.
- ◎ El Framework ASP.NET MVC utiliza el patrón Modelo-Vista-Controlador.

Interioridades

- ◎ Que disponemos en cada una de estas carpetas:
 - Models: Se implementan las clases y estructuras de datos que resuelven la lógica de negocios de nuestra aplicación.
 - Views: Definimos cómo se mostrará la interfaz de usuario de la aplicación.
 - Controllers: Es un conjunto de clases que coordinan la comunicación entre las peticiones que hace el cliente (Navegador generalmente), el modelo que procesa los datos que llegan del cliente y comunica a las vistas para que muestren los datos peticionados por el cliente.

Distribución de un proyecto en MVC



Demo MVC



```
dotnet new mvc -n mvcApp1  
dotnet run
```

Razor Pages

- © Una página Razor es muy similar al componente de vista al que están acostumbrados los desarrolladores de ASP.NET MVC. Tiene la misma sintaxis y funcionalidad.

Ventajas

- ⦿ Principio de responsabilidad única
- ⦿ Get y Post directos
- ⦿ Vistas HTML

Demo Razor Pages



```
dotnet new razor -n razorApp1  
dotnet run
```



ASP.NET MVC vs Razor Pages

ASP.NET MVC	Razor Page
<ul style="list-style-type: none">└─ RazorPageTest<ul style="list-style-type: none">└─ Connected Services└─ Dependencies└─ Properties└─ wwwroot└─ Controllers<ul style="list-style-type: none">└─ C# HomeController.cs└─ Models<ul style="list-style-type: none">└─ C# PageClass.cs└─ Views<ul style="list-style-type: none">└─ Home<ul style="list-style-type: none">└─ About.cshtml└─ Contact.cshtml└─ Index.cshtml└─ ManagePage.cshtml└─ Shared<ul style="list-style-type: none">└─ _ViewImports.cshtml└─ _ViewStart.cshtml└─ appsettings.json└─ bower.json└─ bundleconfig.json└─ C# Program.cs└─ C# Startup.cs	<ul style="list-style-type: none">└─ RazorPageTest2<ul style="list-style-type: none">└─ Connected Services└─ Dependencies└─ Properties└─ wwwroot└─ Pages<ul style="list-style-type: none">└─ _Layout.cshtml└─ _ValidationScriptsPartial.cshtml└─ _ViewImports.cshtml└─ _ViewStart.cshtml└─ About.cshtml└─ Contact.cshtml└─ Error.cshtml└─ Index.cshtml└─ ManagePage.cshtml<ul style="list-style-type: none">└─ C# ManagePage.cshtml.cs└─ appsettings.json└─ bower.json└─ bundleconfig.json└─ C# Program.cs└─ C# Startup.cs

Siguientes de sesiones

- ◎ Sesión #3: Entity Framework Core y LINQ
- ◎ Sesión #4: Sync, Async y await – Manejo de Excepciones
- ◎ Sesión #5: .NET Core Web API
- ◎ Sesión #6: SignalR en .NET Core – Parte 1 – Web Chat
- ◎ Sesión #7: SignalR en .NET Core – Parte 2 – Web Chat



Gracias!

¿Preguntas?

Puedes contactarme mediante:

@yovafree

erwingeo1021@gmail.com

<https://github.com/yovafree>



Repositorio Github de la Comunidad

