



El futuro digital  
es de todos

MinTIC



# Razor



Universidad de Caldas

# Hola:

En el mercado existen diferentes marcos de trabajo o *frameworks* que tienen sus propias herramientas y facilitan su implementación, uno de ellos es el *Razor* de .net. Para la interactividad en una página web es necesario, además de las etiquetas, un conjunto de instrucciones escritas en código fuente JavaScript. Sin embargo, ante la imposibilidad de que todos los programadores utilicen JavaScript, se crean *frameworks* que combinan HTML con lenguajes de programación existentes o, incluso, lenguajes nuevos que dan más flexibilidad, desempeño y potencia al desarrollo *frontend*. En el caso de .net se usa C#. Esta combinación da origen a *Razor*, que utiliza un nuevo lenguaje de etiquetas (no estándar) llamado CSHTML, que son archivos con HTML con código C#. Adicionalmente, pueden utilizar clases, como las que se han hecho para las entidades del dominio.

En el siguiente caso se encuentran párrafos HTML marcados con la etiqueta P y, internamente, hay instrucciones C# con el símbolo arroba. En otro caso, se define una variable llamada message con el mensaje: “Hello World Razor”. Luego, se ubica esta variable en una etiqueta de título HTML y el símbolo @ indicará que ese código es C#. Así que, cuando se encuentra una página construida de esta manera es importante recordar que no es HTML, sino una página *Razor*, en las que no solamente se puede utilizar C# para declarar y mostrar variables. También se pueden utilizar condicionales, ciclos y operaciones matemáticas, es decir, se puede emplear todo el potencial de funciones de .NET en aplicaciones web.

```
1 <p>The total is:  
2     @Model.Sum</p>  
3  
4 <p>The time is:  
5     @DateTime.Now</p>  
6
```

```
1 @{  
2     string message = "Hello World Razor!";  
3 }  
4 <h2>@message </h2>
```

## Hello World Razor!

```
<div class="product-widget">  
    @if(today == "Monday") {  
        <p>Monday Widget</p>  
    }  
    @foreach(var product in products) {  
        <p>@product.Name</p>  
    }  
    @{  
        var sum = 1 + 2;  
    }  
</div>
```

La posibilidad de utilizar C# directamente en las páginas HTML abre una cantidad de opciones que permiten crear páginas dinámicas rápidamente. Sin embargo, esto puede volverse ilegible dada la cantidad de líneas en un archivo y, aunque esto puede hacerse así, las páginas razor pueden dividirse en 2 partes (archivos). Una parte, en donde solo se encuentra el código C# para clases, propiedades, lógica y la otra parte, con CSSHTML donde se utiliza C# para tareas de visualización.

La posibilidad de utilizar C# directamente en las páginas HTML abre una cantidad de opciones que permiten crear páginas dinámicas rápidamente. Sin embargo, esto puede volverse ilegible dada la cantidad de líneas en un archivo y, aunque esto puede hacerse así, las páginas razor pueden dividirse en 2 partes (archivos). Una parte, en donde solo se encuentra el código C# para clases, propiedades, lógica y la otra parte, con CSSHTML donde se utiliza C# para tareas de visualización.

.

```

4
5 namespace HelloWorldMvcApp
6 {
7     public class MenuModel
8     {
9         public List<string> Menu { get; set; }
10        public void OnGet()
11        {
12            Menu = new List<string> {"Tacos", "Pizza"};
13        }
14    }
15
16    @page
17    @model MenuModel
18
19    <div>
20        @foreach(var item in Model.Menu)
21        {
22            <p>@item.Name</p>
23        }
24    </div>
25
26
```

Model



Cuando un cliente hace una petición de un recurso a un servidor web con .net core, se ejecuta Middleware, que es un intermediario, es decir, permite la interacción y el intercambio de información entre diferentes programas y con diferentes formatos respectivamente. El middleware ubica la página C# asociada al recurso, ejecuta la lógica definida y genera un conjunto con la página CSSHTML que será enviada de nuevo al middleware para que lo convierta a HTML CSS y JavaScript, antes de ser enviado al cliente/navegador.

.

Mapeo al modelo

URL: http://razor.com/pagina.cshtml

Modelo de la página

Ejecuta el código C# asociada a la página

Middleware

HTML CSS JS

Página con etiquetas HTML

Renderiza a lenguaje etiquetado

En conclusión, los marcos de trabajo o *frameworks* permiten la combinación de etiquetas y códigos para facilitar el desarrollo de aplicaciones *frontend*. También está Razor, que es el *framework* de .net que se va a utilizar en el curso y, el concepto de middleware, que es importante a la hora de entender el desarrollo web actual.

.

