

PRACTICO 3

PARÁMETROS DE LOS MÉTODOS DE ACCIÓN.

INTRODUCCION

Veremos cómo se reciben en un método de controlador los valores enviados a través del método HTTP GET, típicamente desde una URL escrita en la barra de direcciones del navegador, desde un link, o desde un formulario que tenga el atributo METHOD =GET (o no tenga METHOD seteado, ya que por defecto para HTTP es GET).



Para probar es conveniente poner un punto de parada en cada método de acción, así podrá inspeccionar las variables y hacer el seguimiento del flujo de ejecución.

CREAR EL PROYECTO

- Siguiendo el mismo camino de los laboratorios anteriores, crear un proyecto llamado ParametrosActionResult.

PROBAR EL FUNCIONAMIENTO DE UN MÉTODO CON UN PARÁMETRO LLAMADO ID.

- Crear un **controlador** en blanco llamado **ParametrosController**.
- Dejar el método Index() que crea por defecto tal cual :

```
public class ParametrosController : Controller
{
    // GET: Parametros
    public ActionResult Index()
    {
        return View();
    }
}
```

- Agregar una vista **Index** con clic derecho sobre el método Index, Agregar Vista. En la vista escribir el código:

```
@{
    ViewBag.Title = "Index";
}

<h2>Index de Parámetros</h2>
<a href="/parametros/ProbarParametroXId/54">Probar pasar un valor por la
URL a un parámetro llamado Id</a>
<p>Mostrando el valor recibido en ProbarParametrosXId
@ViewBag.Mensaje01</p>
```

- Agregar un método en ParametrosController que reciba un parámetro llamado Id (en este caso en un int, pero puede ser de cualquier tipo).

```
public ActionResult ProbarParametroXId(int id)
{
    ViewBag.Mensaje01 = "Recibido el valor" + id.ToString();
    return View("Index");
}
```

PRACTICO 3

PARÁMETROS DE LOS MÉTODOS DE ACCIÓN.

- Probar la aplicación, y verificar que muestra el valor 54, que coincide con el que pasamos en la URL.

Incluir en la vista Index otro link idéntico al anterior, pero que utilice el símbolo de comienzo de Query String (el signo de interrogación):

```
<a href="/parametros/ProbarParametroXId?Id=877">Probar pasar un valor por la URL a un parámetro llamado Id utilizando el signo de interrogación</a>
```

Refrescar la página en el navegador (cuando se hacen cambios en la vista no es necesario cortar la ejecución, alcanza con refrescar la página) y verificar que muestra 877. Esto muestra que acepta cualquiera de las dos sintaxis.

PROBAR EL FUNCIONAMIENTO DE UN MÉTODO CON UN PARÁMETRO QUE NO SE LLAME ID

- Agregar otro método en ParametrosController que reciba un parámetro llamado **edad**

```
public ActionResult ProbarParametroConOtroNombre(int edad)
{
    if (edad != null)
        ViewBag.Mensaje02 = "Recibido el valor" + edad.ToString();
    else
        ViewBag.Mensaje02 = "No se recibió ningún valor";
    return View("Index");
}
```

- En la vista Index, modificar el código agregando un link para acceder al nuevo método y mostrar el valor que se recibió.

```
<h2>Index de Parámetros</h2>
<p><a href="/parametros/ProbarParametroXId/54">Probar pasar un valor por la URL a un parámetro llamado Id</a></p>
<p><a href="/parametros/ProbarParametroXId?Id=877">Probar pasar un valor por la URL a un parámetro llamado Id utilizando el signo de interrogación</a></p>
<p>Mostrando el valor recibido en ProbarParametroXId @ViewBag.Mensaje01</p>
<p><a href="/parametros/ProbarParametroConOtroNombre/98">Probar pasar un valor por la URL a un parámetro que no se llame Id (x ej. edad)</a></p>
<p><a href="/parametros/ProbarParametroConOtroNombre?edad=98">Probar pasar un valor por la URL a un parámetro que no se llame Id (x ej. edad)</a></p>
<p>Mostrando el valor recibido en ProbarParametroConOtroNombre en el parámetro edad</p>
<p>Usando querystring explícito</p>@ViewBag.Mensaje02
<p>Usando url amigable (sin el parámetro explícito)</p>@ViewBag.Mensaje02
```

Probar la aplicación accediendo a /parámetros/ desde la barra de direcciones del navegador, y hacer clic en ambos links que acabamos de agregar.

PRACTICO 3

PARÁMETROS DE LOS MÉTODOS DE ACCIÓN.

Observar que al hacer clic en el que usa la url sin signo de interrogación, se produce una excepción. Esto es porque el parámetro edad queda en null, ya que al no llamarse id MVC no puede hacer la correspondencia por nombre.

En cambio al hacer clic en el último link, edad queda cargada.

EVITAR QUE SE PRODUZCA UNA EXCEPCIÓN CUANDO UN PARÁMETRO RECIBIDO SEA NULL

Los tipos por valor no aceptan null (en general los numéricos, bool, char, enum y struct – como DateTime).

Si reciben un valor null en un método de acción, se producirá una excepción. Para evitar esto, existen los tipos nullable.

Se declaran incluyendo el operador ? y podremos preguntar si son null antes de usarlos, evitando excepciones.

Para probarlo modificar el código del método **ProbarParametroConOtroNombre** y volver a probar el enlace que provocaba la excepción:

```
public ActionResult ProbarParametroConOtroNombre(int? edad)
{
    if (edad != null)
        ViewBag.Mensaje02 = "Recibido el valor" +
edad.ToString();
    else
        ViewBag.Mensaje02 = "No se recibió ningún valor";
    return View("Index");
}
```

Probar la aplicación y verificar que el problema quedó solucionado.

PROBAR CÓMO SE RECIBEN LOS PARÁMETROS DESDE UN FORM.

Probaremos cómo se reciben los parámetros cuando se pasan con un formulario mediante el método GET.

- En ParametrosController escribir este método de acción:

```
public ActionResult ProbarParametroFormXGet(string nombre, int? edad)
{
    if (edad != null && !string.IsNullOrEmpty(nombre))
        ViewBag.ValoresForm = "Nombre: " + nombre + " Edad " +
edad.ToString();
    else
        ViewBag.ValoresForm = "El nombre o la edad son nulos";
    return View("Index");
}
```

PRACTICO 3

PARÁMETROS DE LOS MÉTODOS DE ACCIÓN.

- Modificar la vista Index para incluir el formulario y mostrar los valores recibidos, agregando al final el código resaltado:

```
<h2>Index de Parámetros</h2>
<p><a href="/parametros/ProbarParametroXId/54">Probar pasar un valor por la URL a un parámetro llamado Id</a></p>
<p><a href="/parametros/ProbarParametroXId?Id=877">Probar pasar un valor por la URL a un parámetro llamado Id utilizando el signo de interrogación</a></p>
<p>Mostrando el valor recibido en ProbarParametrosXId @ViewBag.Mensaje01</p>
<p><a href="/parametros/ProbarParametroConOtroNombre/98">Probar pasar un valor por la URL a un parámetro que no se llame Id (x ej. edad)</a></p>
<p><a href="/parametros/ProbarParametroConOtroNombre?edad=98">Probar pasar un valor por la URL a un parámetro que no se llame Id (x ej. edad)</a></p>
<p>Mostrando el valor recibido en ProbarParametroConOtroNombre en el parámetro edad</p>
<p>Usando querystring explícito</p>@ViewBag.Mensaje02
<p>Usando url amigable (sin el parámetro explícito)</p>@ViewBag.Mensaje02

<div>
  <h2>Recibir los parámetros desde un formulario:</h2>
  <form method="GET" action="/parametros/ProbarParametroFormXGet">
    <p>Ingrese un nombre:</p><p><input type="text" name="nombre" /></p>
    <p>Ingrese una edad</p><p><input type="text" name="edad" /></p>
    <p><input type="submit" value="Enviar el formulario x get"></p>
  </form>
  <p>Los datos ingresados en el formulario son:</p>
  @ViewBag.ValoresForm
</div>
```

- Probar la aplicación ingresando los distintos casos. Verificar qué ocurre cuando se incluye un texto en la edad.

Observar la barra de direcciones del navegador.

Ahora cambiar en el tag <form> el atributo method para que el formulario viaje por post (method="POST") y verificar qué ocurre.

Observar nuevamente la barra de direcciones del navegador.