1. Crear un nuevo proyecto MVC llamado LABMVC07\_EvaluacionParametros.
2. En la carpeta Controllers crear un controlador MVC seleccionando la plantilla **Controlador de MVC 5 en blanco** y llamarlo **Parametros01Controller**.
3. Agregar el método:

public ActionResult evaluarParametro(int cantidad)

{

return View(cantidad);

}

1. Crearle una vista con el nombre por defecto (evaluarParametro) seleccionando la plantilla Empty (sin modelo).
2. Agregarle manualmente a la vista que realice el model binding con Int32 y despliegue un string con el valor del modelo:

@model Int32

@{

ViewBag.Title = "evaluarParametro";

}

<h2>evaluarParametro</h2>

@Model.ToString()

1. Ejecutar la aplicación (en la barra de direcciones debe aparecer: http://localhost:XXXXX/Parametros01/evaluarParametro)

. Verifique que se obtiene un error.   
*The parameters dictionary contains a null entry for parameter 'cantidad' of non-nullable type 'System.Int32' for method 'System.Web.Mvc.ActionResult evaluarParametro(Int32)' in LabMVC07\_EvaluacionParametros.Controllers.Parametros01Controller'. An optional parameter must be a reference type, a nullable type, or be declared as an optional parameter.  
Nombre del parámetro: parameters*



Puede explicar lo que ocurre?

**Solución 1: Pasar el parámetro como query string.**

En la barra de direcciones pasar el parámetro con nombre y valor utilizando el queryString. Probarlo escribiendo en la barra de direcciones:

http://localhost:XXXXX/Parametros01/evaluarParametro/?cantidad=10

Verificar al ejecutar que la vista despliega el valor 10

**Solución 2: Configurar al parámetro un valor por defecto.**

Modificar el código del método en el controlador incluyendo 0 como valor por defecto para cantidad, y verificar que escribiendo en la barra de direcciones **http://localhost:XXXXX/Parametros01/evaluarParametro/**

en la vista obtenemos 0 (el valor por defecto).

public ActionResult evaluarParametro(int cantidad=0)

{

return View(cantidad);

}

**Solución 3: Declarar el parámetro como tipo nullable.**

Hacer que el parámetro sea nullable. Escribir el código en el método evaluarParametro y ejecutar verificando que en la vista visualizamos 100 al no pasar un parámetro.

**public ActionResult evaluarParametro(int? cantidad)**

{

if (cantidad != null)

return View(cantidad);

else

return View(100);

}

¿Qué ocurre si en la barra de direcciones escribimos las siguientes URLs?:

**http://localhost:XXXXX/Parametros01/evaluarParametro/25**

**http://localhost:XXXXX/Parametros01/evaluarParametro/?cantidad=25**

1. **Pruebe lo que ocurre si el parámetro se declara como string y en la URL no se le pasa un valor:**

public ActionResult evaluarParametro(string cantidad)

{

if (cantidad != null)

return View(cantidad);

else

return View(100);

}



¿Es esperable este comportamiento?¿Por qué?

1. **Utilizar un ActionLink para pasarle el parámetro a la acción del controlador.**

En el archivo layout.cshtml debajo de la carpeta Views/Shared incluir junto a los demás enlaces un hipervínculo que nos lleve al método evaluarParametro pasándole un valor a la cantidad, y ejecutar la aplicación. En la vista debe mostrarnos el valor de la cantidad (25 en el código).

<li>@Html.ActionLink(

linkText:"Evaluar los parámeros",

actionName:"evaluarParametro",

controllerName:"Parametros01",

routeValues: new { cantidad=25},

htmlAttributes: null)

</li>

Observar que se utiliza una sobrecarga de ActionLink que incluye un último parámetro en null. Esto es para evitar que asp.net intente utilizar otra sobrecarga. No existe una que reciba el texto del link, la acción, el controlador y los parámetros (routeValues). Se pueden consultar todas las sobrecargas en <https://msdn.microsoft.com/en-us/library/system.web.mvc.html.linkextensions.actionlink(v=vs.118).aspx>

**Nota:** Se podría utilizar la sintaxis que no incluye los parámetros con nombre, pero es más posible que lleve a error:

<li>@Html.ActionLink("Evaluar parámetros", "evaluarParametro", "Parametros01",

new { cantidad = 10 },null)</li>