

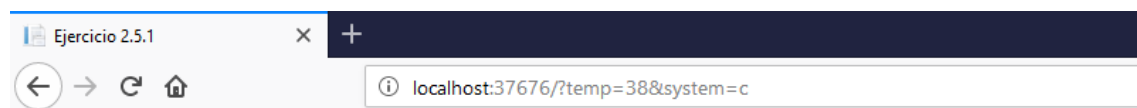
Ejercicio 2.5.1.

Modifica el ejercicio 2.3.1, que convertía una temperatura en grados centígrados a grados Fahrenheit.

Los cambios consistirán en lo siguiente:

- La temperatura se recogerá de la url con un parámetro llamado *temp*, en lugar de *cent*.
- Tendremos otro parámetro *system*, que indique si se trata de una temperatura en grados centígrados o grados Fahrenheit, mediante los valores 'c' y 'f' respectivamente.
- Se creará una clase estática que contenga dos métodos, uno para pasar la temperatura de cada uno de los sistemas al otro.

La aplicación funcionará del mismo modo, mostrando el valor en el sistema original y el valor en el sistema al que se ha convertido, siendo la salida similar a la que sigue:



38 grados centígrados = 100,4 grados Fahrenheit

Solución.

HomeController.cs

```
using Microsoft.AspNetCore.Mvc;
using Ejercicio.Models;

namespace Ejercicio.Controllers
{
    public class HomeController : Controller
    {
        private static int count = 0;

        public IActionResult Index(double temp, char system)
        {
            ViewData["original"] = temp;
            ViewData["system"] = system;

            if (system == 'f' || system == 'F')
            {
                ViewData["converted"] =
                    TemperatureConverter.FahrenheitToCelsius(temp);
            }
            else
            {
                ViewData["converted"] =
                    TemperatureConverter.CelsiusToFahrenheit(temp);
            }

            return View();
        }
    }
}
```

TemperatureConverter.cs

```
namespace Ejercicio.Models
{
    public static class TemperatureConverter
    {
        public static double CelsiusToFahrenheit(double temperatureC)
        {
            // Convert Celsius to Fahrenheit.
            double fahrenheit = (temperatureC * 9 / 5) + 32;

            return fahrenheit;
        }

        public static double FahrenheitToCelsius(double temperatureF)
        {
            // Convert Fahrenheit to Celsius.
            double celsius = (temperatureF - 32) * 5 / 9;

            return celsius;
        }
    }
}
```

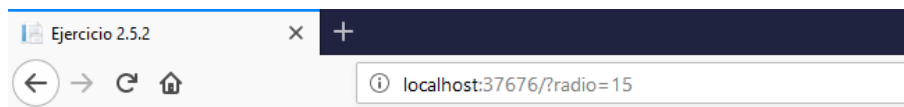
Index.cshtml

```
<html>
<head>
  <title>Ejercicio 2.5.1</title>
</head>
<body>
  @{
    char system = (char)ViewData["system"];
    if (system == 'f' || system == 'F')
    {
      <h2>
        @ViewData["original"] grados Fahrenheit = @ViewData["converted"]
      grados centígrados
      </h2>
    }
    else
    {
      <h2>
        @ViewData["original"] grados centígrados = @ViewData["converted"]
      grados Fahrenheit
      </h2>
    }
  }
</body>
</html>
```

Ejercicio 2.5.2.

Para la realización de este segundo ejercicio crearemos una clase que represente una esfera. Esta clase tendrá un atributo que almacene el radio de la misma. También ofrecerá un método que calcule y devuelva el volumen de la esfera ($\frac{4}{3} * \pi * \text{radio}^3$).

Para probar el funcionamiento de la aplicación pasaremos el radio como parámetro en la url y se mostrará una página en el navegador que ofrezca la información de la esfera, tal y como se muestra a continuación:



Información de la esfera:

- Radio: 15
- Volumen: 10602,8752058656

Solución.

HomeController.cs

```
using Ejercicio.Models;
using Microsoft.AspNetCore.Mvc;

namespace Ejercicio.Controllers
{
    public class HomeController : Controller
    {
        public IActionResult Index(double radio)
        {
            Sphere sphere = new Sphere(radio);

            ViewData["radio"] = radio;
            ViewData["volume"] = sphere.Volume;

            return View();
        }
    }
}
```

Sphere.cs

```
using System;

namespace Ejercicio.Models
{
    public class Sphere
    {
        public double Radio { get; }

        public double Volume
        {
            get
            {
                return 4 / 3 * Math.PI * Radio * Radio * Radio;
            }
        }

        public Sphere(double radio)
        {
            Radio = radio;
        }
    }
}
```

Index.cshtml

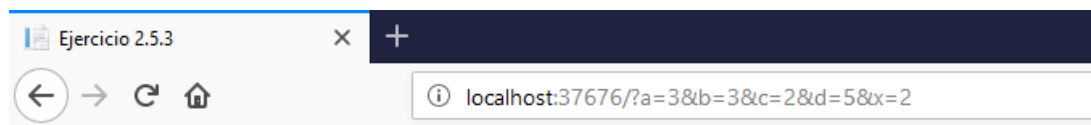
```
<html>
<head>
    <title>Ejercicio 2.5.2</title>
</head>
<body>
    <h2>Información de la esfera:</h2>
    <ul>
        <li>Radio: @ViewData["radio"]</li>
        <li>Volumen: @ViewData["volume"]</li>
    </ul>
</body>
</html>
```

Ejercicio 2.5.3.

Escribir una clase que represente a los polinomios de la forma ax^5+bx^3+cx+d y que tenga un método que devuelva el valor para un valor determinado de x , y un constructor que reciba los cuatro coeficientes.

A partir de esta clase crearemos una aplicación web que recibirá por su url los valores de los coeficientes a , b , c y d , además del valor de x , siendo todos números enteros.

A continuación mostrará en una página el resultado del polinomio, de modo similar a la imagen de ejemplo:



$$3x^5 + 3x^3 + 2x + 5 = 129$$

Solución.

HomeController.cs

```
using Ejercicio.Models;
using Microsoft.AspNetCore.Mvc;

namespace Ejercicio.Controllers
{
    public class HomeController : Controller
    {
        public IActionResult Index(int a, int b, int c, int d, int x)
        {
            Pol pol = new Pol(a, b, c, d);
            ViewData["result"] = pol.Result(x);
            ViewData["pol"] = a + "x^5 + " + b + "x^3 + " + c + "x + " + d;

            return View();
        }
    }
}
```

Pol.cs

```
using System;

namespace Ejercicio.Models
{
    public class Pol
    {
        public int A { get; }
        public int B { get; }
        public int C { get; }
        public int D { get; }

        public Pol(int a, int b, int c, int d)
        {
            A = a;
            B = b;
            C = c;
            D = d;
        }

        public int Result(int x)
        {
            return A * (int)Math.Pow(x, 5) + B
                * (int)Math.Pow(x, 3) + C * x + D;
        }
    }
}
```

Index.cshtml

```
<html>
<head>
  <title>Ejercicio 2.5.3</title>
</head>
<body>
  <h2>@ViewData["po1"] = @ViewData["result"]</h2>
</body>
</html>
```