



**UNIVERSIDAD LATINA
DE COSTA RICA**
LAUREATE INTERNATIONAL UNIVERSITIES®

Universidad
Latina

Facultad De Tecnologías De Información y Comunicaciones
Escuela De Ingeniería De Sistemas Informáticos

“Tarea 3”

Tópicos Selectos de Computación
(BSI-16)

Estudiantes:

Alonso Mesén

Fabián Rodríguez

José Zúñiga

Aaron Vargas

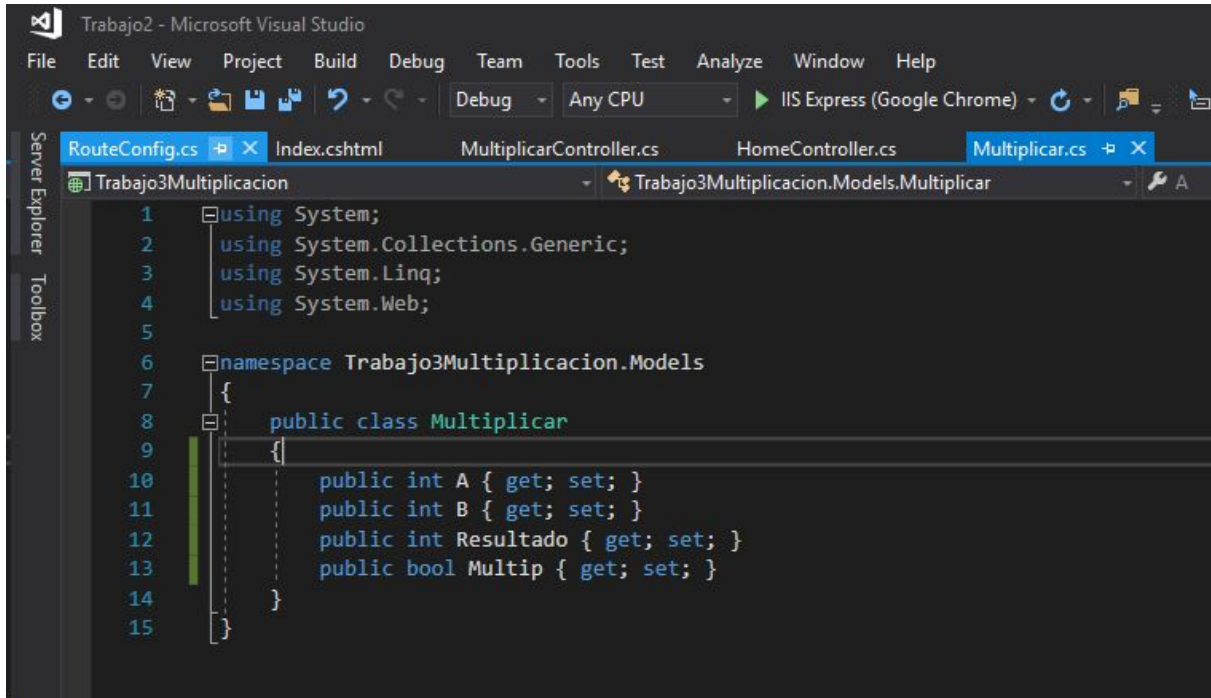
Profesor:

Hector Fernandez Mendez

San José, 9 de Marzo de 2019.

Se realiza una aplicación web utilizando el patrón Modelo Vista Controlador en el cual se deben realizar distintas operaciones matemáticas.

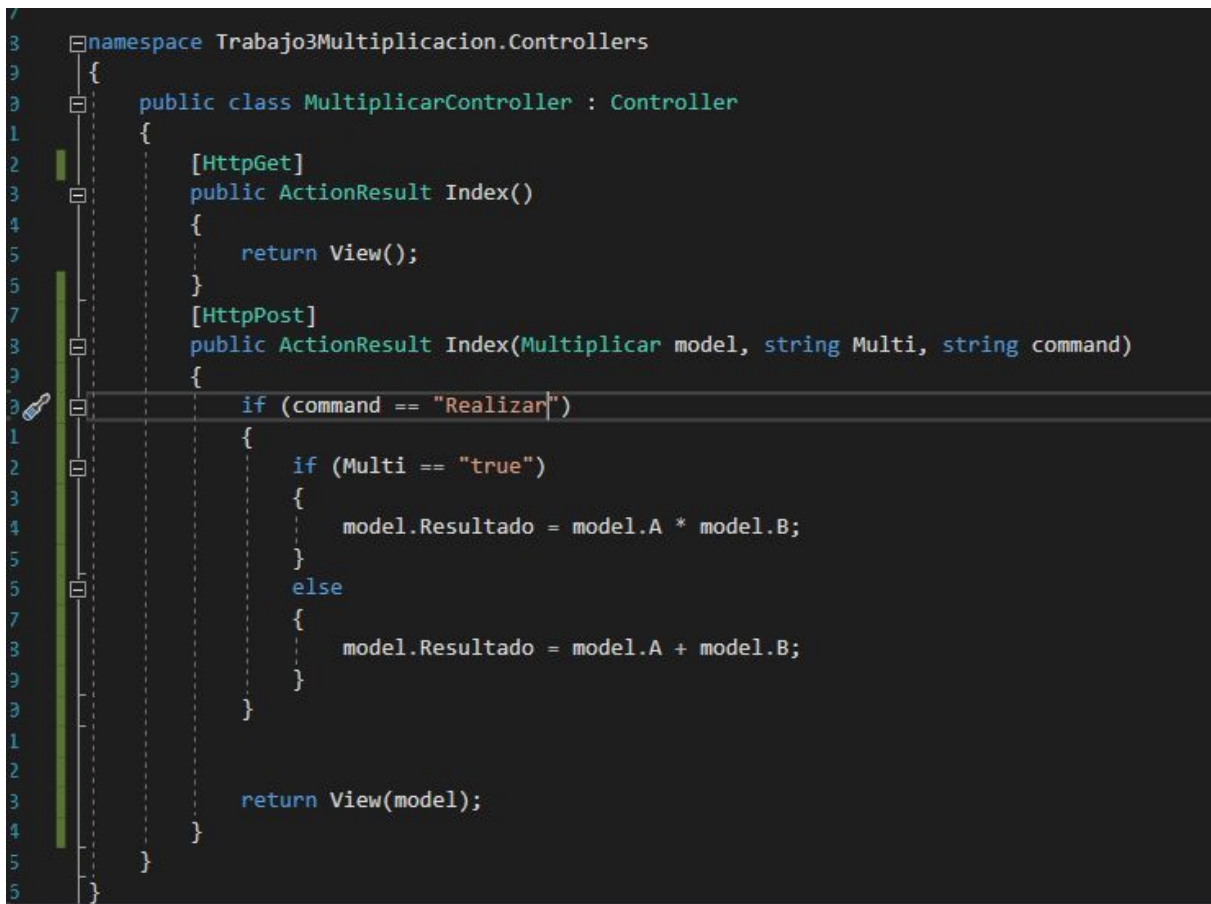
El modelo de la aplicación es el siguiente:



The screenshot shows the Visual Studio IDE with the file 'Multiplicar.cs' open. The code defines a class 'Multiplicar' within the namespace 'Trabajo3Multiplicacion.Models'. The class has four public properties: 'A' (int), 'B' (int), 'Resultado' (int), and 'Multip' (bool), each with get and set accessors. The code is as follows:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5
6 namespace Trabajo3Multiplicacion.Models
7 {
8     public class Multiplicar
9     {
10         public int A { get; set; }
11         public int B { get; set; }
12         public int Resultado { get; set; }
13         public bool Multip { get; set; }
14     }
15 }
```

El controlador de la aplicación es el siguiente:



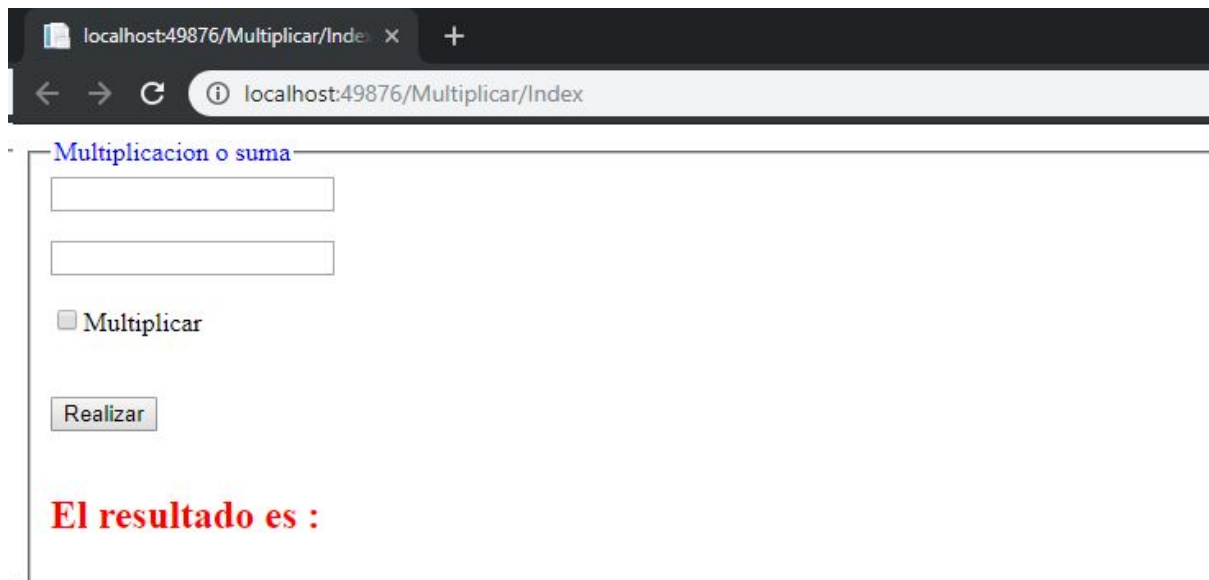
The screenshot shows the Visual Studio IDE with the file 'MultiplicarController.cs' open. The code defines a class 'MultiplicarController' within the namespace 'Trabajo3Multiplicacion.Controllers', inheriting from 'Controller'. It has two actions: 'Index' (HttpGet) which returns the view, and 'Index' (HttpPost) which performs multiplication or addition based on the 'Multi' parameter and the 'command' value. The code is as follows:

```
1 namespace Trabajo3Multiplicacion.Controllers
2 {
3     public class MultiplicarController : Controller
4     {
5         [HttpGet]
6         public ActionResult Index()
7         {
8             return View();
9         }
10        [HttpPost]
11        public ActionResult Index(Multiplicar model, string Multi, string command)
12        {
13            if (command == "Realizar")
14            {
15                if (Multi == "true")
16                {
17                    model.Resultado = model.A * model.B;
18                }
19                else
20                {
21                    model.Resultado = model.A + model.B;
22                }
23            }
24
25            return View(model);
26        }
27    }
28 }
```

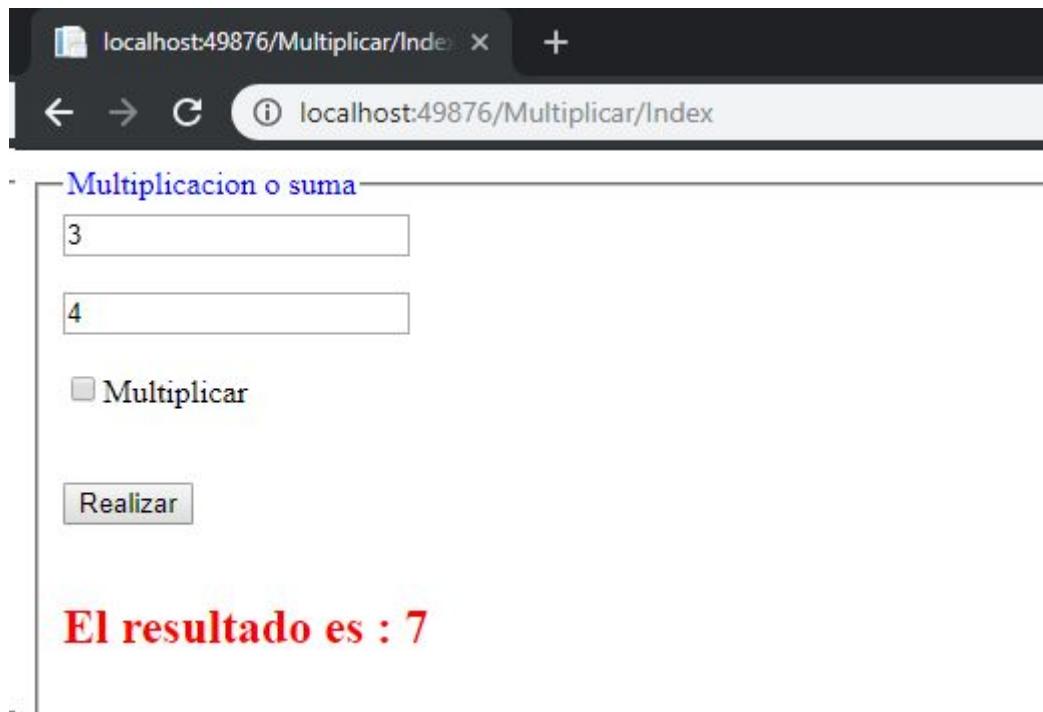
La vista que se está utilizando es la siguiente:

```
RouteConfig.cs | Index.cshtml | MultiplicarController.cs | HomeController.cs | Multiplicar.cs
1  @if
2  ViewBag.Title = "Home Page";
3  }
4  @model Trabajo3Multiplicacion.Models.Multiplicar
5  @if
6  Layout = null;
7  }
8
9  @using (Html.BeginForm())
10 {
11     <fieldset>
12         <legend style="color:blue"> Multiplicacion o suma </legend>
13         @Html.EditorFor(x => x.A) <br /> @Html.EditorFor(x => x.B) <br /> <br />
14         <input type="checkbox" name="Multi" value="true"> Multiplicar <br />
15         <br /> <br />
16
17         <input type="submit"
18             value="Realizar"
19             name="command" />
20         <br /> <br />
21         <h2 style="color:red">
22             @Html.Label("El resultado es :")
23             @Html.DisplayFor(x => x.Resultado)
24         </h2>
25     </fieldset>
26 }
```

Ahora veremos la aplicación funcionando:



Si no se selecciona el checkbox multiplicar los números se suman



localhost:49876/Multiplicar/Inde x +

localhost:49876/Multiplicar/Index

Multiplicacion o suma

3


4

☐ Multiplicar

Realizar

El resultado es : 7

En caso de seleccionar el checkbox los números se multiplican



localhost:49876/Multiplicar/Inde x +

localhost:49876/Multiplicar/Index

Multiplicacion o suma

3

4

☒ Multiplicar

Realizar

El resultado es : 12