

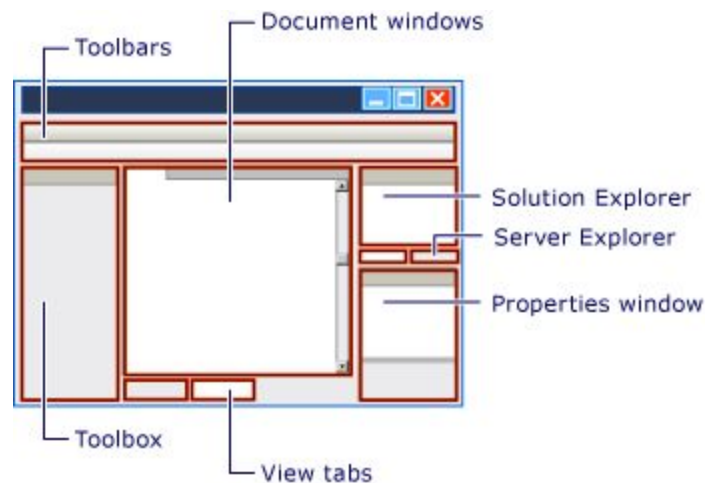
CLASE 1

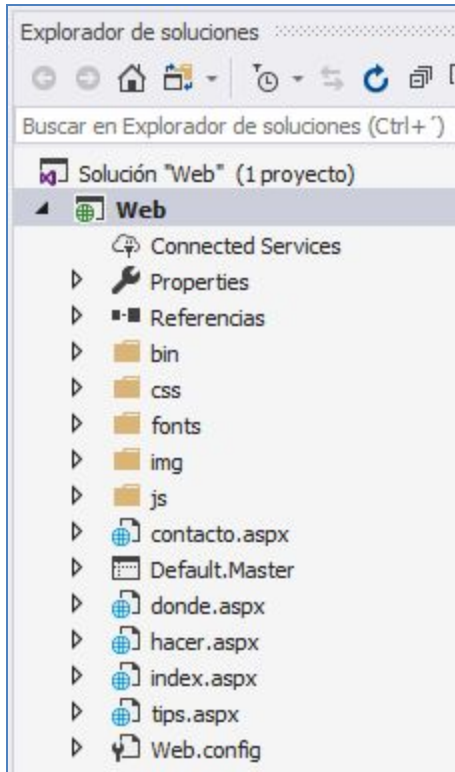
Detallamos los ejercicios de este laboratorio. **Se recomienda realizar como mínimo todos los ejercicios excepto los indicados como opcionales** si tuviera restricciones de tiempo. El objetivo es asegurar la ejercitación adecuada para los contenidos de esta clase:

- Ejercicio 1: Explorar Visual Studio .Net
- Ejercicio 2: Repaso programación básica con C# :
 1. Operaciones básicas
 2. Compra en restaurant
 3. Mayor de dos números
 4. Desglose de billetes
 5. Triángulos (opcional)
 6. Días de la semana
 7. Tabla de multiplicar (con For) (opcional)
 8. Tabla de multiplicar (con While)
 9. Suma de N números
 10. Funciones de cadena
 11. Número primo (opcional)
 12. Arreglo
 13. Número mayor y menor de una matriz (opcional)

Ejercicio 1: Explorar Visual Studio .Net

Antes de comenzar con el ejercicio, recordamos la estructura de ventanas y herramientas del IDE de Visual Studio:

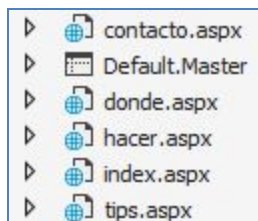




Este ejercicio tiene como objetivo **explorar una breve aplicación web sencilla** con páginas WebForm/HTML, CSS y Bootstrap, para familiarizarse con el IDE y los archivos de código, teniendo en cuenta que realizó el curso anterior **Introducción a C# .Net** o posee los conocimientos equivalentes.

La aplicación muestra información turística de una ciudad. Sólo la interfaz visual (front-end) del proyecto.

- Baje la aplicación **ExplorarVisualStudio**, descomprímala en una carpeta y haga doble click en el archivo **ExplorarVisualStudio.sln**.
- Ejecute la aplicación, y luego al detenerla, recorra a través de la ventana Explorador de soluciones, los distintos archivos del proyecto.



- Hay archivos que son páginas web realizadas con WebForms, que tienen código HTML con hojas de estilo aplicadas (esto se ejecuta en el cliente), y archivos de código C# que se ejecutarán en el servidor.

contacto.aspx X contacto.aspx.cs

ContentPlaceholder1 (Personalizado)

Formulario de Contacto

- Mensaje de error 1.
- Mensaje de error 2.

Nombres:

***Los nombres solo deben contener letras**

Apellidos:

***Los apellidos solo deben contener letras**

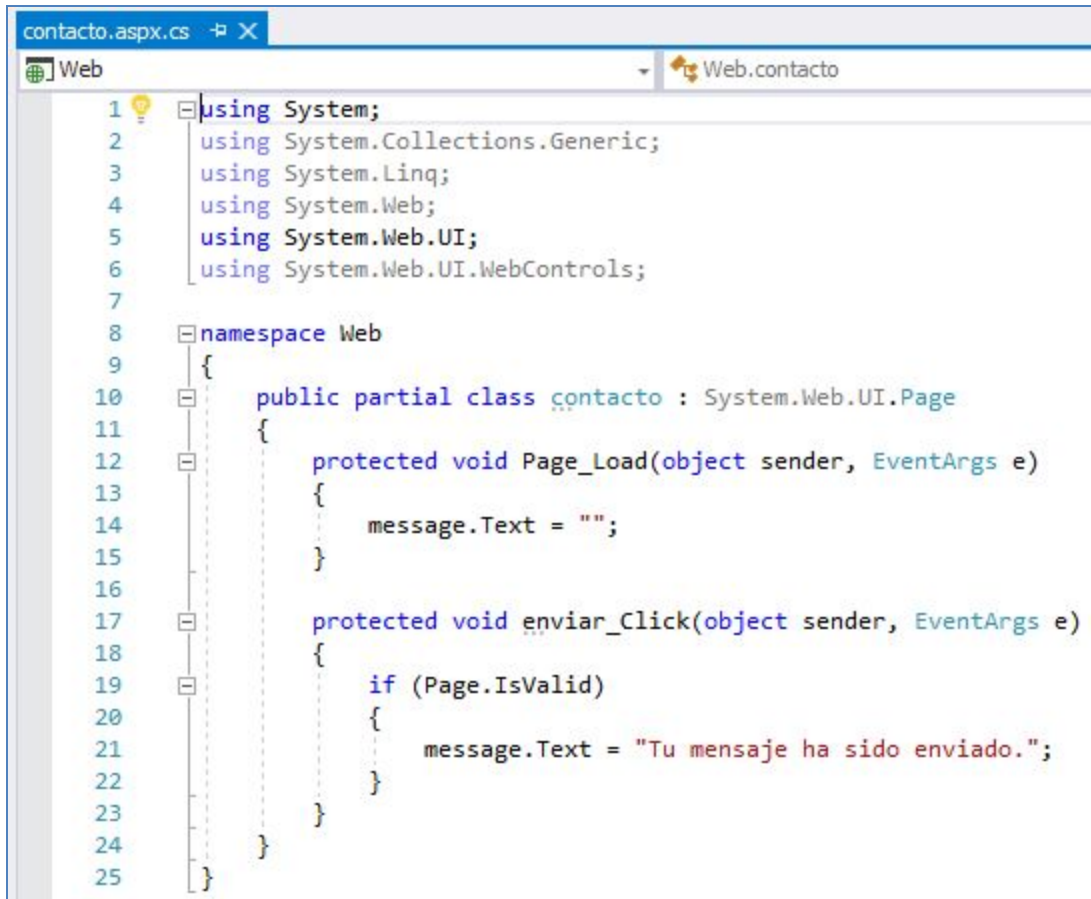
Email:

contacto.aspx X contacto.aspx.cs

```

1  <%@ Page Title="" Language="C#" MasterPageFile="~/Default.Master" AutoEventWireup="true"
2  <asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
3  </asp:Content>
4  <asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceholder1" runat="server">
5      <div class="container cuerpo">
6          <div class="row contacto">
7              <div class="col-md-12">
8                  <h1>Formulario de Contacto</h1>
9                  <asp:ValidationSummary ID="ValidationSummary1" runat="server" Validation
10                 <div class="form-group">
11                     <asp:Label ID="Label1" runat="server" Text="Nombres: "></asp:Label><
12                     <asp:RegularExpressionValidator ID="RegularExpressionValidator2" run
13                 </div>
14                 <div class="form-group">
15                     <asp:Label ID="Label2" runat="server" Text="Apellidos: "></asp:Label
16                 </div>
17                 <div class="form-group">
18                     <asp:Label ID="Label3" runat="server" Text="Email: "></asp:Label><as
19                 </div>
20                 <div class="form-group">
21                     <asp:Label ID="Label4" runat="server" Text="Telefono: "></asp:Label>
22                 </div>

```

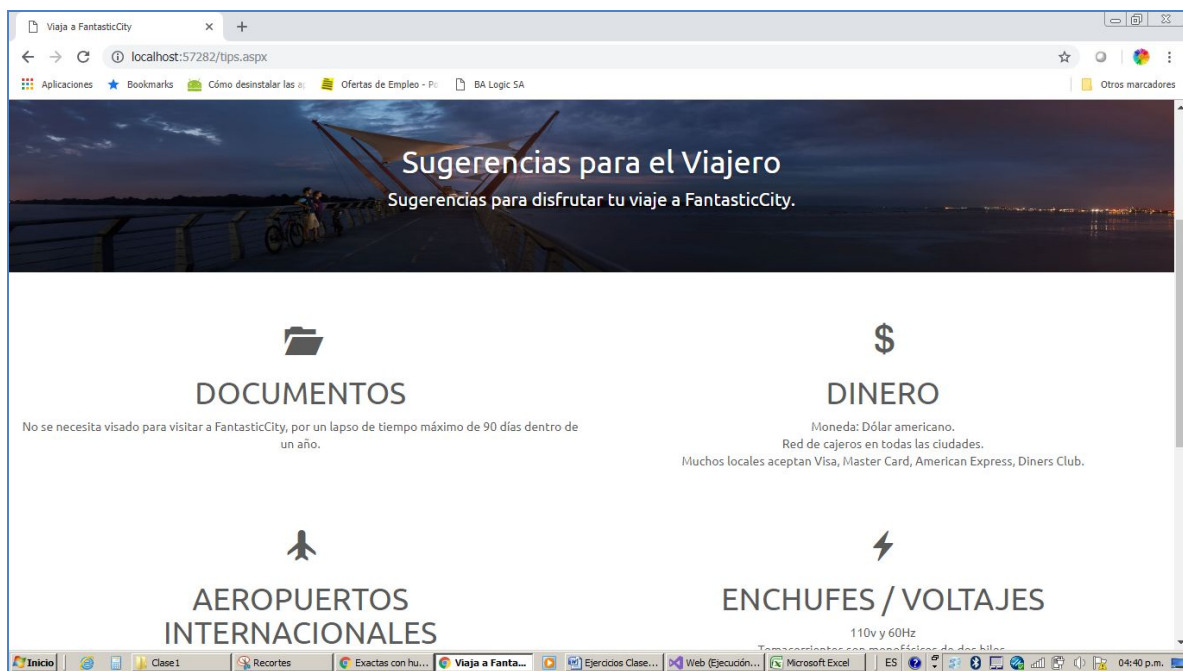
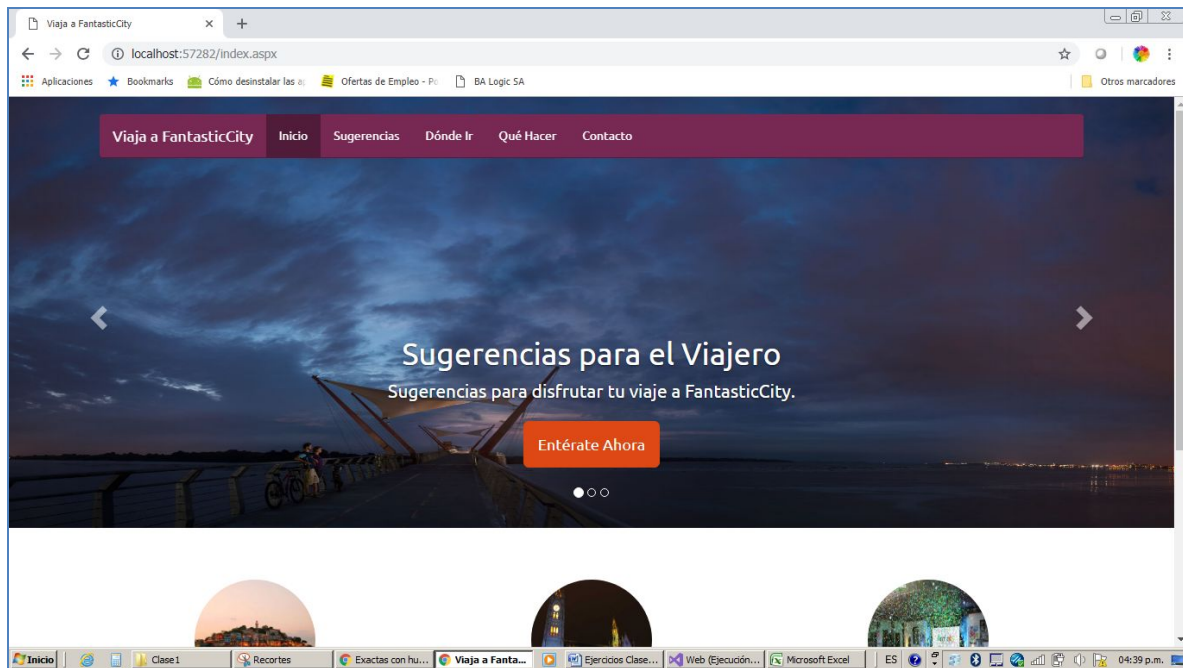


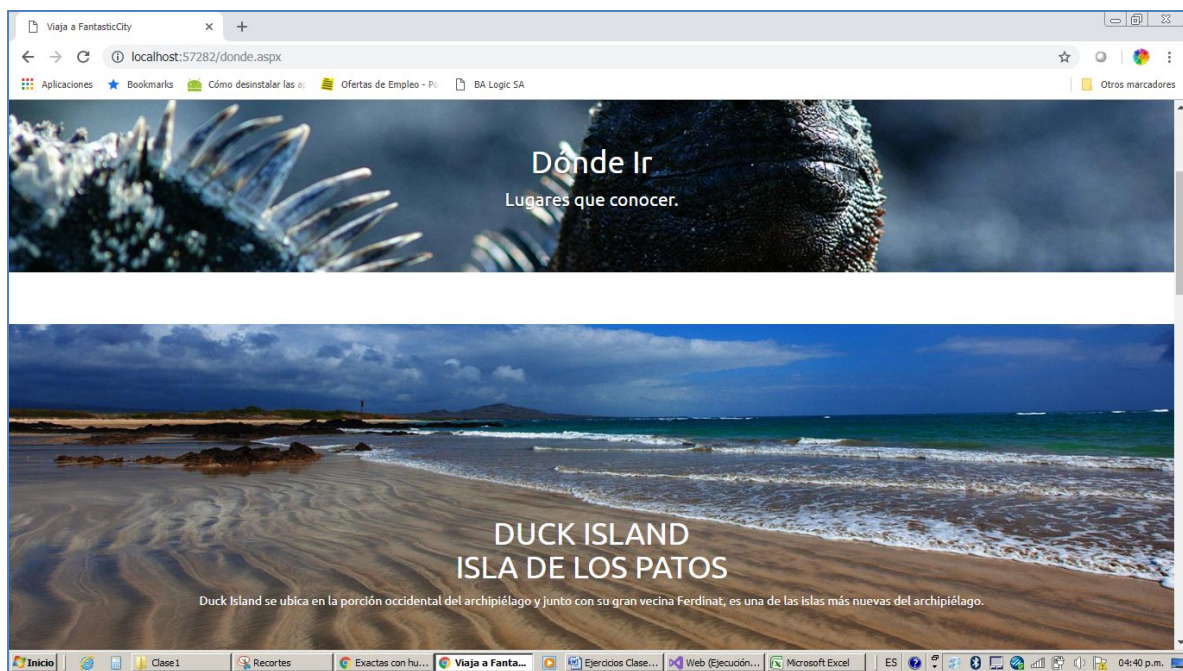
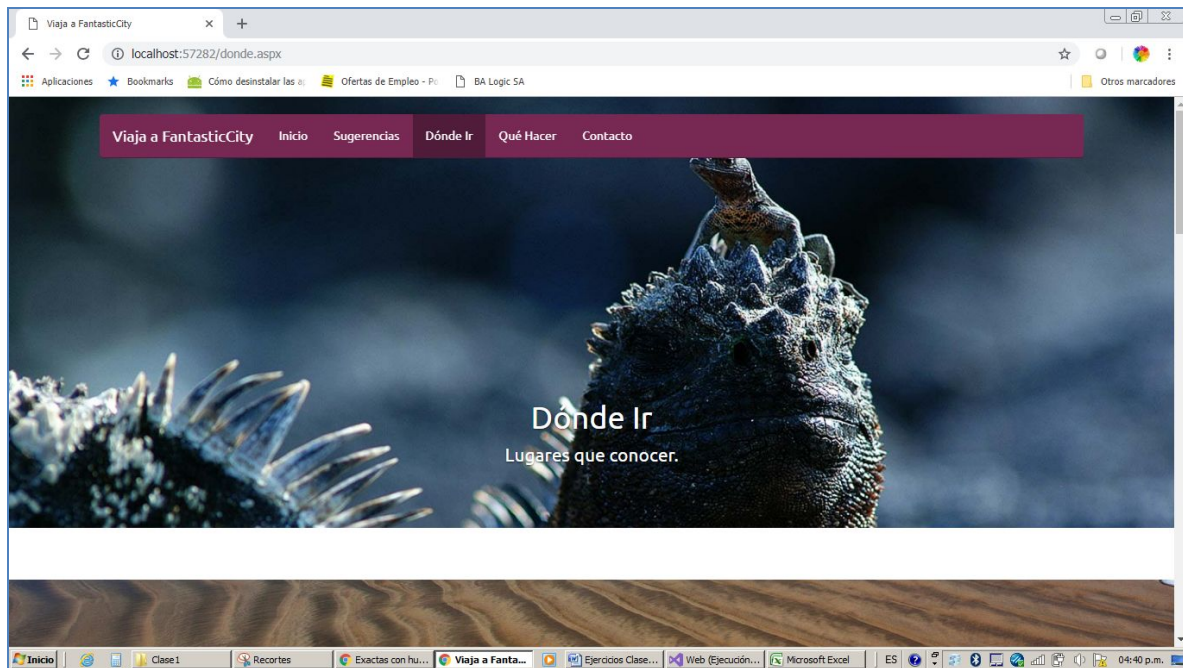
```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.UI;
6 using System.Web.UI.WebControls;
7
8 namespace Web
9 {
10     public partial class contacto : System.Web.UI.Page
11     {
12         protected void Page_Load(object sender, EventArgs e)
13         {
14             message.Text = "";
15         }
16
17         protected void enviar_Click(object sender, EventArgs e)
18         {
19             if (Page.IsValid)
20             {
21                 message.Text = "Tu mensaje ha sido enviado.";
22             }
23         }
24     }
25 }
```

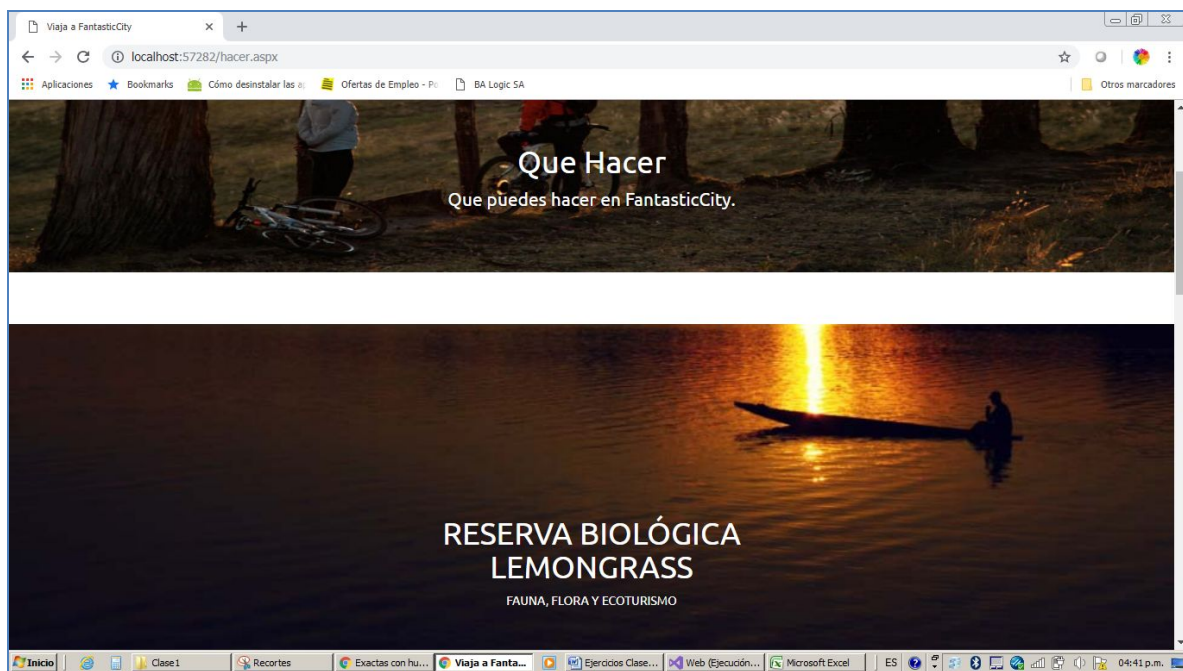
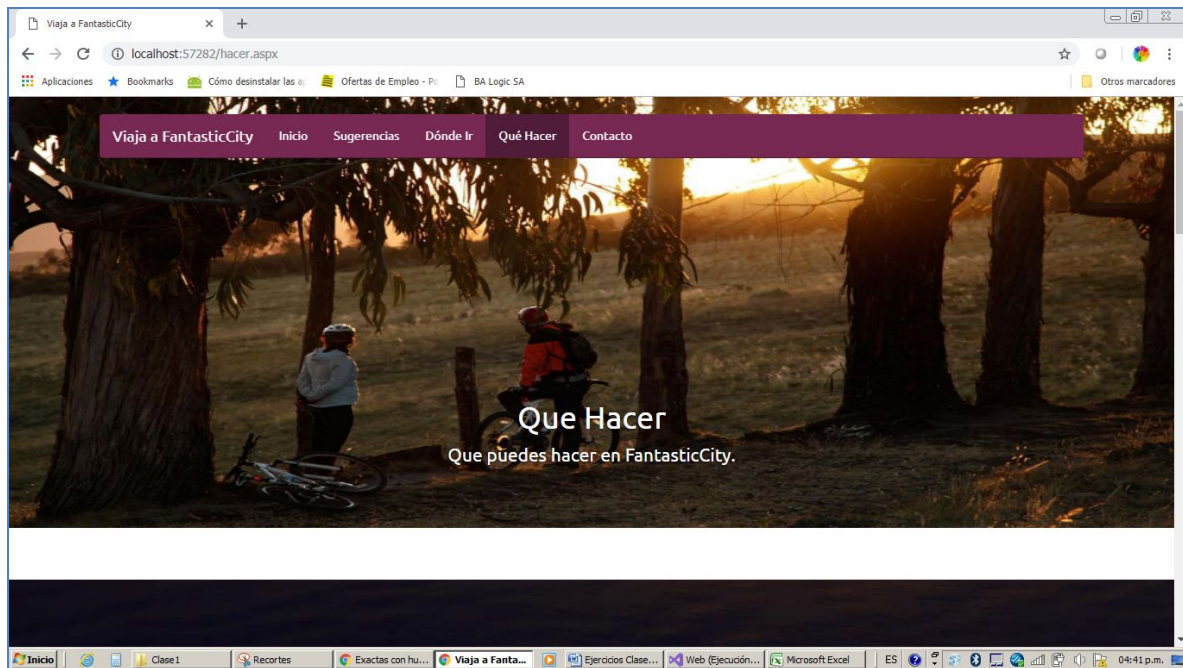
- A lo largo del curso y en las primeras clases, completará el estudio del **lenguaje C#** y la **Paradigma a objetos (POO) aplicado al lenguaje C#**, y luego abordará la creación de aplicaciones web sencillas con WebForms, estética atractiva y simple con CSS y acceso a datos con ADO.Net, básico de Entity Framework, y uso de bases de datos Sql Server.

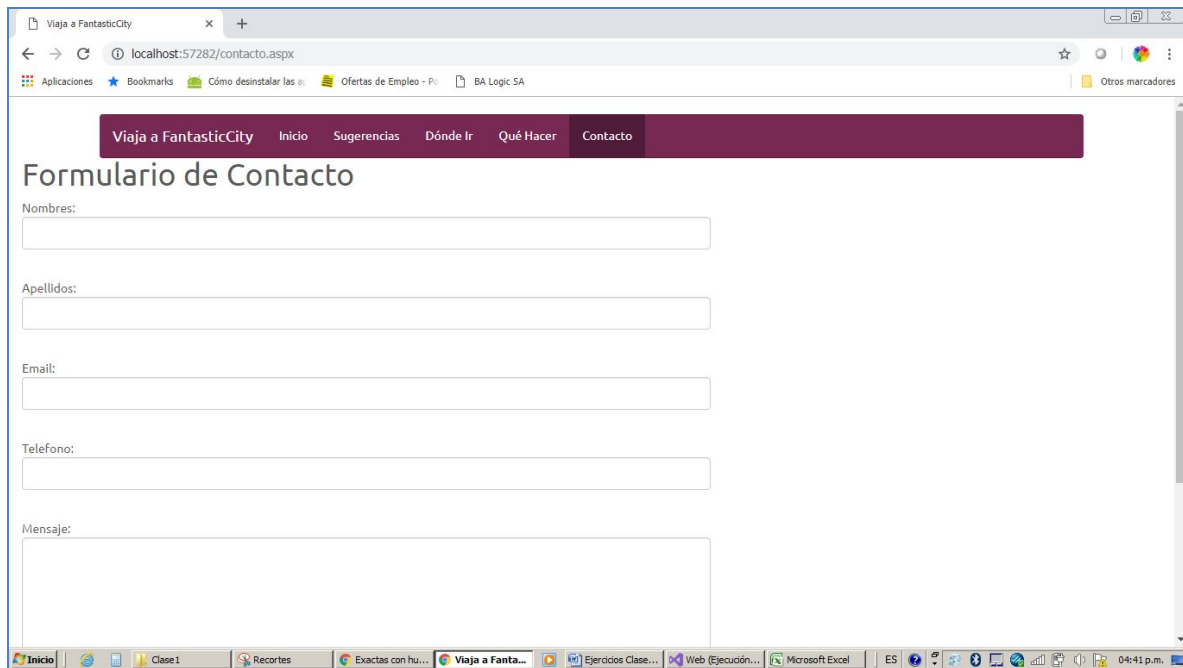
Características del Proyecto

- Proyecto realizado en Visual Studio 2015/2017.
- ASP.NET Web Forms con C#.
- Formulario de contacto con validación de campos requeridos, sólo números, sólo texto, formato de correo.
- Tema Bootstrap para estilos principales.
- Plantilla [Carousel](#) para diagramar la página.
- Contenido y fotos de imágenes de turismo.









Ejercicio 2: Repaso programación básica con C#

Los siguientes ejercicios, se presentan a modo de repaso del curso anterior, **Introducción a la programación con C#**, y como introducción a este curso. Enumeramos varios ejemplos que los ejercitará en un proyecto de consola, repaso de:

- estructuras de control:
 - secuencia
 - decisión simple: if
 - decisión múltiple: switch
 - ciclos: while, do while, for
 - variables
 - funciones y parámetros

Operaciones básicas

```
// OPERACIONES ARITMETICAS BASICAS
```

```
int NUM1, NUM2, RESUL;  
string linea;
```

```
Console.WriteLine("PRIMER NÚMERO :");  
linea = Console.ReadLine();  
NUM1 = int.Parse(linea);
```

```
Console.WriteLine("SEGUNDO NÚMERO :");  
linea = Console.ReadLine();
```



```

NUM2 = int.Parse(linea);
Console.WriteLine();

RESUL = NUM1 + NUM2;
Console.WriteLine("LA SUMA ES {0}: ", RESUL);

RESUL = NUM1 - NUM2;
Console.WriteLine("LA RESTA ES: {0} - {1} = {2} ", NUM1, NUM2, RESUL);

RESUL = NUM1 * NUM2;
Console.WriteLine("LA MULTIPLICACIÓN ES: " + RESUL);

RESUL = NUM1 / NUM2;
Console.WriteLine("LA DIVISIÓN ES: " + RESUL);

RESUL = NUM1 % NUM2;
Console.WriteLine("EL RESIDUO ES: " + RESUL);

Console.Write("Pulse una Tecla:");
Console.ReadKey();

```

Compra en restaurant

```

// Calcula el total de la compra de cada ítem
byte CantBebida, CantHamburguesa, CantPapas;
double APagar;
string linea;

const double PrecioB = 0.8;
const double PrecioH = 2;
const double PrecioP = 1.2;

Console.Write("CANTIDAD DE HAMBURGUESAS :"); linea = Console.ReadLine();
CantHamburguesa = byte.Parse(linea);

Console.Write("CANTIDAD DE PAPAS :"); linea = Console.ReadLine();
CantPapas = byte.Parse(linea);

Console.Write("CANTIDAD DE BEBIDAS :"); linea = Console.ReadLine();
CantBebida = byte.Parse(linea);
Console.WriteLine();

APagar = (CantHamburguesa * PrecioH) + (CantPapas * PrecioP) + (CantBebida *
PrecioB);
Console.WriteLine("VALOR A PAGAR: " + APagar);

Console.Write("Pulse una Tecla:");
Console.ReadKey();

```

Mayor de dos números

```

// Se ingresan dos números y se muestra el mayor, el menor o si son iguales

```

```

int NUM1, NUM2;
string linea;

Console.Write("NÚMERO 1 :"); linea = Console.ReadLine();
NUM1 = int.Parse(linea);
Console.Write("NÚMERO 2 :"); linea = Console.ReadLine();
NUM2 = int.Parse(linea);

if ((NUM1 > NUM2))
{
    Console.WriteLine("{0} ES MAYOR QUE {1}", NUM1, NUM2);
}
else
{
    if ((NUM1 == NUM2))
    {
        Console.WriteLine("{0} ES IGUAL A {1}", NUM1, NUM2);
    }
    else
    {
        Console.WriteLine("{0} ES MENOR QUE {1}", NUM1, NUM2);
    }
}
Console.WriteLine();
Console.WriteLine("OTRA MANERA");
string RESUL;
if (NUM1 > NUM2)
{
    RESUL = "MAYOR";
}
else
if (NUM1 == NUM2)
{
    RESUL = "IGUAL";
}
else
{
    RESUL = "MENOR";
}

Console.WriteLine("{0} ES {1} QUE {2}", NUM1, RESUL, NUM2);
Console.Write("Pulse una Tecla:");
Console.ReadKey();

```

Desglose de billetes

```

//Dado un importe, indica la cantidad de billetes de cada valor
//para llegar a ese importe
int CAN, C100, C50, C20, C10, C5;
C100 = 0;
C50 = 0;
C20 = 0;
C10 = 0;

C5 = 0;

```

```

string linea;
Console.Write("DIGITE UNA CANTIDAD :"); linea = Console.ReadLine();
CAN = int.Parse(linea);

if ((CAN >= 100))
{
    C100 = (CAN / 100);
    CAN = CAN - (C100 * 100);
}
if ((CAN >= 50))
{
    C50 = (CAN / 50);
    CAN = CAN - (C50 * 50);
}
if ((CAN >= 20))
{
    C20 = (CAN / 20);
    CAN = CAN - (C20 * 20);
}
if ((CAN >= 10))
{
    C10 = (CAN / 10);
    CAN = CAN - (C10 * 10);
}
if ((CAN >= 5))
{
    C5 = (CAN / 5);
    CAN = CAN - (C5 * 5);
}
Console.WriteLine("BILLETES DE A 100: " + C100);
Console.WriteLine("BILLETES DE A 50 : " + C50);
Console.WriteLine("BILLETES DE A 20 : " + C20);
Console.WriteLine("BILLETES DE A 10 : " + C10);
Console.WriteLine("BILLETES DE A 5 : " + C5);
Console.WriteLine("BILLETES DE A 1 : " + CAN);
Console.Write("Pulse una Tecla:");
Console.ReadKey();

```

Triángulos

```

//Según el tamaño de los datos, indica si el triángulo es
//equilátero, isósceles o escaleno
int LAD01, LAD02, LAD03;
string linea;

Console.Write("DIGITE LADO 1 :"); linea = Console.ReadLine();
LAD01 = int.Parse(linea);
Console.Write("DIGITE LADO 2 :"); linea = Console.ReadLine();
LAD02 = int.Parse(linea);
Console.Write("DIGITE LADO 3 :"); linea = Console.ReadLine();
LAD03 = int.Parse(linea);

if ((LAD01 == LAD02) & (LAD02 == LAD03))
{
    Console.WriteLine("TRIÁNGULO EQUILÁTERO. TODOS IGUALES");
}

```

```

    }
    else
    {
        if ((LAD01 != LAD02) & (LAD01 != LAD03) & (LAD02 != LAD03))
        {
            Console.WriteLine("TRIÁNGULO ESCALENO. NINGUNO IGUAL");
        }
        else
        {
            Console.WriteLine("TRIÁNGULO ISÓSCELES. DOS IGUALES");
        }
    }
}

Console.Write("Pulse una Tecla:");
Console.ReadKey();

```

Días de la semana

```

//Según el número de día de la semana, muestra su nombre
int num;
string linea;

Console.WriteLine();
Console.WriteLine("DIAS DE LA SEMANA");
Console.WriteLine();
Console.Write("Ingrese un numero del 1 al 7 :");

linea = Console.ReadLine();
num = int.Parse(linea);

switch (num)
{
    case 1:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso corresponde al día
DOMINGO");
        break;
    case 2:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso corresponde al día
LUNES");
        break;
    case 3:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso corresponde al día
MARTES");
        break;
    case 4:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso corresponde al día
MIERCOLES");
        break;
    case 5:
        Console.WriteLine();

```



```

        Console.WriteLine("El numero que ingreso corresponde al día
JUEVES");
        break;
    case 6:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso corresponde al día
VIERNES");
        break;
    case 7:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso corresponde al día
SABADO");
        break;
    default:
        Console.WriteLine();
        Console.WriteLine("El numero que ingreso esta fuera de rango");
        break;
    }
    Console.ReadKey();

```

Tabla de multiplicar (con For)

```

//Muestra la tabla de multiplicar de un número
byte NUM, I;
int RESULT;
string linea;

Console.Write("DIGITE NÚMERO:"); linea = Console.ReadLine();
NUM = byte.Parse(linea);

for (I = 1; I <= 12; I++)
{
    RESULT = NUM * I;
    Console.WriteLine("{0} * {1} = {2}", NUM, I, RESULT);
}
Console.Write("Pulse una Tecla:");
Console.ReadKey();

```

TABLA DE MULTIPLICAR CON WHILE

Tabla de multiplicar (con While)

```

//Muestra la tabla de multiplicar de un número
byte NUM, I;
int RESULT;
string linea;

Console.Write("DIGITE NÚMERO:"); linea = Console.ReadLine();
NUM = byte.Parse(linea);

I = 1;
while ((I <= 12))
{
    RESULT = NUM * I;
    Console.WriteLine("{0} * {1} = {2}", NUM, I, RESULT);
    I = I + 1;
}

```

```

    }
    Console.Write("Pulse una Tecla:");
    Console.ReadKey();

```

Suma de N números

```

//Suma N números ingresados
byte CAN, K;
int NUM;
int SUM = 0;
string linea;

Console.Write("LÍMITE:"); linea = Console.ReadLine();
CAN = byte.Parse(linea);

for (K = 1; K <= CAN; K++)
{
    Console.Write("DIGITE UN NÚMERO:"); linea = Console.ReadLine();
    NUM = int.Parse(linea);
    SUM += NUM;
}

Console.WriteLine("SUMA TOTAL ES : " + SUM);
Console.WriteLine("MEDIA ARITMÉTICA: " + SUM / CAN);
Console.Write("Pulse una Tecla:");
Console.ReadKey();

```

Funciones de cadena

```

//Dado un string ingresado, se le aplican diversas funciones
//para manejo de string que tiene .Net
//También se invocan dos funciones propias, con parámetros
public void FuncionesCadena()
{
    string NOM;
    byte K;
    Console.Write("DIGITE SU NOMBRE : "); NOM = Console.ReadLine();

    Console.WriteLine("LONGITUD : " + NOM.Length);
    Console.WriteLine("ESTA DENTRO : " + NOM.Contains("ARIEL"));
    Console.WriteLine("REEMPLAZO VOCAL A: " + NOM.Replace("A", "X"));
    Console.WriteLine("EN MINUSCULAS : " + NOM.ToLower());
    Console.WriteLine("EN MAYÚSCULAS : " + NOM.ToUpper());
    Console.WriteLine("REMOVER 4 LETRAS : " + NOM.Remove(3, 4));
    Console.WriteLine("EXTRAER 4 LETRAS : " + NOM.Substring(3, 4));
    Console.WriteLine("IZQUIERDA 4 LETRAS : " + Left(NOM, 4));
    Console.WriteLine("DERECHA 4 LETRAS : " + Right(NOM, 4));
    Console.Write("Pulse una Tecla:"); Console.ReadLine();
    Console.WriteLine();
    Console.WriteLine("DESDE LA IZQUIERDA");

    for (K = 1; K <= NOM.Length; K++)
    {
        Console.WriteLine(Left(NOM, K));
    }
}

```

```

    }
    Console.WriteLine();
    Console.WriteLine("DESDE LA DERECHA");
    for (K = 1; K <= NOM.Length; K++)
    {
        Console.WriteLine(Right(NOM, K));
    }
    Console.Write("Pulse una Tecla:");
    Console.ReadKey();
}

public static string Left(string param, int length)
{
    string result = param.Substring(0, length);
    return result;
}

public static string Right(string param, int length)
{
    int value = param.Length - length;
    string result = param.Substring(value, length);
    return result;
}

```

Número primo

```

//Dado un número, indica si es primo
int n, x, sw, resi;
string linea;
x = 2;
sw = 0;

Console.WriteLine("NUMERO PRIMO");
Console.WriteLine();
Console.Write("Ingrese el numero:");

linea = Console.ReadLine();
n = int.Parse(linea);

while (x < n && sw == 0)
{
    resi = n % x;
    if (resi == 0)
    {
        sw = 1;
    }
    else
    {
        x = x + 1;
    }
}

if (sw == 0)
{
    Console.WriteLine();
}

```

```

        Console.WriteLine("El numero es PRIMO");
    }
    else
    {
        Console.WriteLine();
        Console.WriteLine("El numero no es PRIMO");
    }
}
Console.ReadKey();

```

Arreglo

```

// FORMA 1
// asignando valores al array al mismo tiempo que se declara,
// la lista de valores debe ir encerrada entre llaves
string[] VECTOR = { "LORENA", "ESTEBAN", "ARIEL" };
byte K = 0;

for (K = 0; K <= 2; K++)
{
    Console.WriteLine(VECTOR[K]);
}
Console.WriteLine();

// FORMA 2
// declarar primero la variable que contendrá el array,
// asignar valores al array al mismo tiempo que se instancia
// la lista de valores debe ir encerrada entre llaves
int[] MILISTA = null;
MILISTA = new int[] { 15, 20, 30, 10, 50, 60, 10, 150, 69 };

for (K = 0; K <= MILISTA.Length - 1; K++)
{
    Console.WriteLine(MILISTA[K]);
}
Console.WriteLine();
Console.Write("Pulse una Tecla:");
Console.ReadLine();

```

Número mayor y menor de una matriz

```

//Muestra el número mayor y menor de una matriz
int F = 0;
int C = 0;
int N = 0;
int MAY = 0;
int MEN = 0;
string linea;

// INGRESO
// PARA COMODIDAD GENERAMOS VALORES PARA LA MATRIZ
Console.Write("TAMAÑO DE LA MATRIZ:");
Random rnd = new Random();
linea = Console.ReadLine();
N = int.Parse(linea);

```



```

Console.Clear();

int[,] MAT = new int[N + 1, N + 1];
for (F = 1; F <= N; F++)
{
    for (C = 1; C <= N; C++)
    {
        MAT[F, C] = rnd.Next(0, 99);
        Console.SetCursorPosition(C * 4, F + 1);
        Console.Write(MAT[F, C]);
    }
}

// PROCESO
MAY = MAT[1, 1];
MEN = MAT[1, 1];
for (F = 1; F <= N; F++)
{
    for (C = 1; C <= N; C++)
    {
        if ((MAT[F, C] > MAY))
            MAY = MAT[F, C];
        if ((MAT[F, C] < MEN))
            MEN = MAT[F, C];
    }
}

// SALIDA
Console.WriteLine();
Console.WriteLine("NÚMERO MAYOR ES: " + MAY);
Console.WriteLine("NÚMERO MENOR ES: " + MEN);
Console.Write("Pulse una Tecla:");
Console.ReadLine();

```