

**“UNIVERSIDAD DE LAS FUERZAS ARMADAS”**

Cuaderno Digital: Primer Parcial

**Integrantes:**

Altamirano Jonathan

Tualle Bryan

**Docente:** Ing. César Villacís

**Sangolquí, Pichincha, Ecuador**

2017-2018

**Temas**

* Fundamentos de programación orientada a objetos y visual con C#
* Creacion de aplicaciones desktop con Windows forms 🡪 GDI+ (Librería de graficos (Windows.h) Graphics Device Interface with COM+ technology)
* Creacion de aplicaciones con WDF (Windows presentation foundation) 🡪 Motores de graficos(Direct X 11.00-12.00) / Open source Open GL, WEB, WEB GL, TREE JS
* HCI (Human computer interaction)
* Kinect
* Joysticks
* Wii Nintendo
* Nunchuck
* Manejo de Arreglos y matrices
* Manejo de colecciones – Listas enlazadas

**Programación de videojuegos en 2D y 3D**

Realidad Aumentada AR C# PHP y MySQL XML

Unity Java Script Python 75% aplicaciones

Realidad Virtual VR web en el mundo

Crear mundos virtuales

Programar

Blender

3D MaxStudio

Maya

Modelamiento 3D, Renderizado y Animación.

**Mecatronica**

* Hardware
* Ing. Industrial
* Mecanica
* Electrica
* Electronica
* Software
* Sistemas de Ingeniería (Ing. SW) Arquitecturas de Modelamiento
* Computacional 🡪 Algoritmos, BD, IA, Sist. Distribuidos, Multimedia
* Comunicación 🡪 Redes y Seguridad LAN, WAN, móviles
* Control 🡪 Microcontroladores: Arduino, Raspberry PI

Complex systems

**Cyber-Physics Systems**

**Visual Studio .NET**

Incluye lenguajes como C# (lo mejor de C/C++ ,java, V. Basic), Visual Basic, VC++, F#

Framework de desarrollo de aplicaciones

Desarrollar aplicaciones Desktop, A. Web A. Mobile, iOT, A. Multimedia, Videojuegos

Aplicaciones con bases de datos – A. Sistemas distribuidos

GUI (Graphic User Interface) Manejan hilos o threads para la comunicación entre varias interfaces graficas

**Visual C# (tecnología .NET de Windows)**

Windows Form Application

Form 🡪 Formulario / Contenedor / Panel

Name: frm (prefijos) Identificar el tipo de objeto

Ej: frmMessages

Form: Clase Parcial (Partial Class)

Todas las: funciones miembro (métodos)

private

Datos miembros (Atributos

Excepto la función constructor que es public

Manejo de

* Objetos - Clases POO
* Eventos. - Funciones especiales para activarse con acciones específicas, (POE programación orientada a eventos o visual)

**MessageBox (Clase)**

Caja de mensajes 🡪 Permite mostrar mensajes de texto (Strings)

Usa método Show 🡪 Mostrar un mensaje

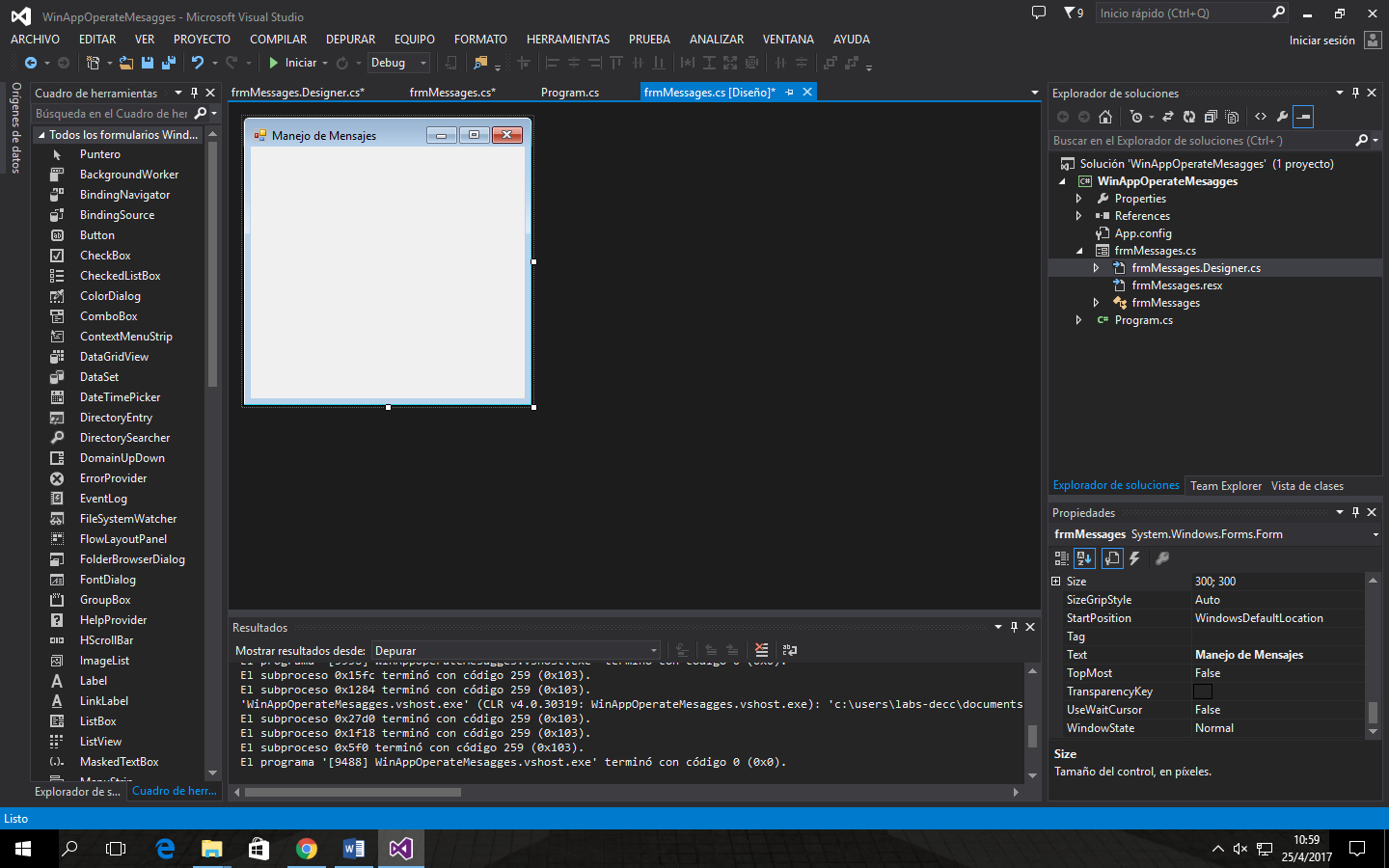
**Form**

Consta de dos partes:

* Archivo Físico (Disco)

Correspondencia (Coincidir el nombre)

* Archivo Lógico (Designer)

**Propiedades (Properties)** Funciones especiales para manejar características de los objetos

* **Name:** frmMessages
* **Text:** manejo de mensajes
* **Size:** 300;300
* **BackColor:** 192; 192; 255

(RGB Red+Green+Blue)

**Group Box** 🡪Caja de grupo / sub contenedor sub panel

**Propiedades**

* **Name:** grbMessages
* **Text:** [BLANK]
* **BackColor:** 192; 192; 255 (mismo que el formulario)

**Button**

**Propiedades**

* **Name:** btnMessage1, btnMessage2, btnMessage3
* **Text:** Desplegar primer mensaje, Desplegar segundo mensaje, salir

**Evento Click (clic)** 🡪 Presionar botón

Ejecuta una o varias acciones en secuencia

**WinAppOperateMesagges**

namespace WinAppOperateMesagges

{

partial class frmMessages

{

/// <summary>

/// Variable del diseñador requerida.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Limpiar los recursos que se estén utilizando.

/// </summary>

/// <param name="disposing">true si los recursos administrados se deben desechar; false en caso contrario.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Código generado por el Diseñador de Windows Forms

/// <summary>

/// Método necesario para admitir el Diseñador. No se puede modificar

/// el contenido del método con el editor de código.

/// </summary>

private void InitializeComponent()

{

this.grbMessages = new System.Windows.Forms.GroupBox();

this.btnMessage1 = new System.Windows.Forms.Button();

this.btnMessage2 = new System.Windows.Forms.Button();

this.btnMessage3 = new System.Windows.Forms.Button();

this.grbMessages.SuspendLayout();

this.SuspendLayout();

//

// grbMessages

//

this.grbMessages.Controls.Add(this.btnMessage3);

this.grbMessages.Controls.Add(this.btnMessage2);

this.grbMessages.Controls.Add(this.btnMessage1);

this.grbMessages.Location = new System.Drawing.Point(12, 12);

this.grbMessages.Name = "grbMessages";

this.grbMessages.Size = new System.Drawing.Size(260, 237);

this.grbMessages.TabIndex = 0;

this.grbMessages.TabStop = false;

//

// btnMessage1

//

this.btnMessage1.Location = new System.Drawing.Point(32, 20);

this.btnMessage1.Name = "btnMessage1";

this.btnMessage1.Size = new System.Drawing.Size(200, 30);

this.btnMessage1.TabIndex = 0;

this.btnMessage1.Text = "Desplegar primer mensaje";

this.btnMessage1.UseVisualStyleBackColor = true;

this.btnMessage1.Click += new System.EventHandler(this.btnMessage1\_Click);

//

// btnMessage2

//

this.btnMessage2.Location = new System.Drawing.Point(32, 54);

this.btnMessage2.Name = "btnMessage2";

this.btnMessage2.Size = new System.Drawing.Size(200, 30);

this.btnMessage2.TabIndex = 1;

this.btnMessage2.Text = "Desplegar segundo mensaje";

this.btnMessage2.UseVisualStyleBackColor = true;

this.btnMessage2.Click += new System.EventHandler(this.btnMessage2\_Click);

//

// btnMessage3

//

this.btnMessage3.Location = new System.Drawing.Point(32, 87);

this.btnMessage3.Name = "btnMessage3";

this.btnMessage3.Size = new System.Drawing.Size(200, 30);

this.btnMessage3.TabIndex = 2;

this.btnMessage3.Text = "Salir";

this.btnMessage3.UseVisualStyleBackColor = true;

this.btnMessage3.Click += new System.EventHandler(this.btnMessage3\_Click);

//

// frmMessages

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(192)))), ((int)(((byte)(192)))), ((int)(((byte)(255)))));

this.ClientSize = new System.Drawing.Size(284, 161);

this.Controls.Add(this.grbMessages);

this.Name = "frmMessages";

this.Text = "Manejo de Mensajes";

this.Load += new System.EventHandler(this.frmMessages\_Load);

this.grbMessages.ResumeLayout(false);

this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.GroupBox grbMessages;

private System.Windows.Forms.Button btnMessage3;

private System.Windows.Forms.Button btnMessage2;

private System.Windows.Forms.Button btnMessage1;

}

}

**frmMessages.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WinAppOperateMesagges

{

public partial class frmMessages : Form

{

public frmMessages()

{

InitializeComponent();

}

private void frmMessages\_Load(object sender, EventArgs e)

{

MessageBox.Show("Hola Mundo");

}

private void btnMessage1\_Click(object sender, EventArgs e)

{

MessageBox.Show("Hola Mundo","Control de mensajes");

}

private void btnMessage2\_Click(object sender, EventArgs e)

{

MessageBox.Show("Hola Mundo",

"Control de mensajes",

MessageBoxButtons.OKCancel, //Botones

MessageBoxIcon.Information); //Iconos

}

private void btnMessage3\_Click(object sender, EventArgs e)

{

Close(); // Cerrar un formulario

//Application.Exit(); // Cerrar toda la aplicacion (varios formularios)

}

}

}

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WinAppOperateMesagges

{

static class Program

{

/// <summary>

/// Punto de entrada principal para la aplicación.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

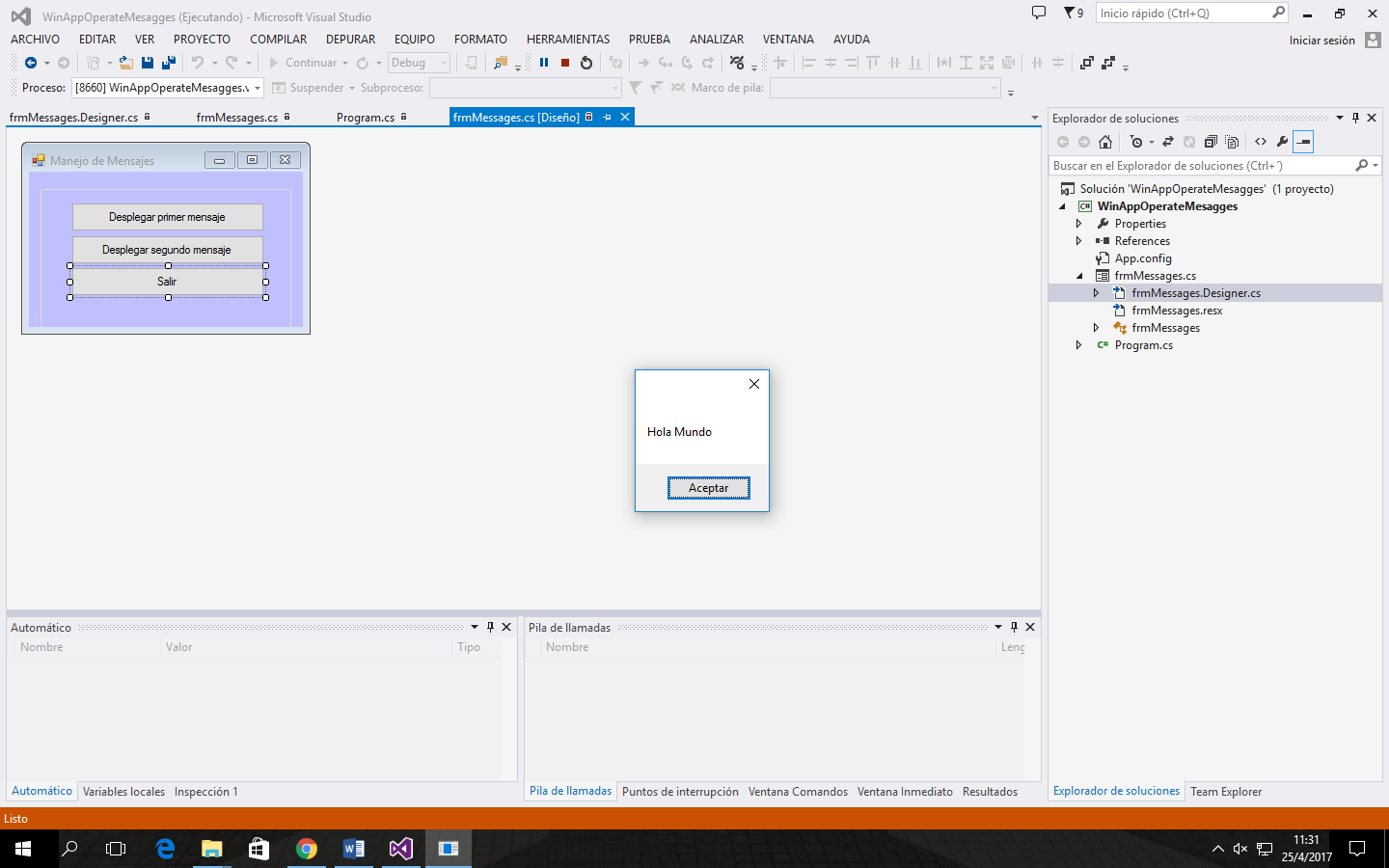
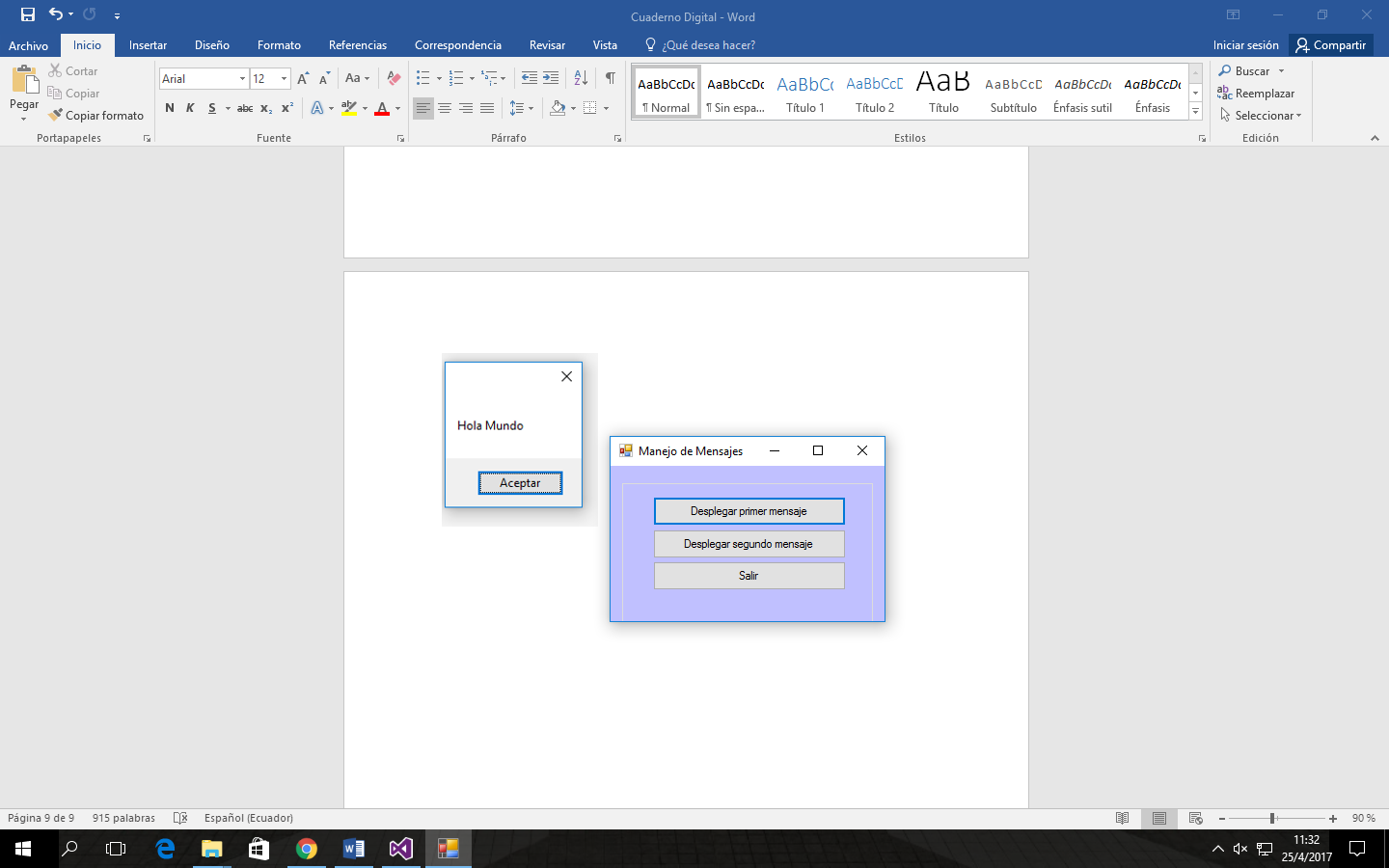
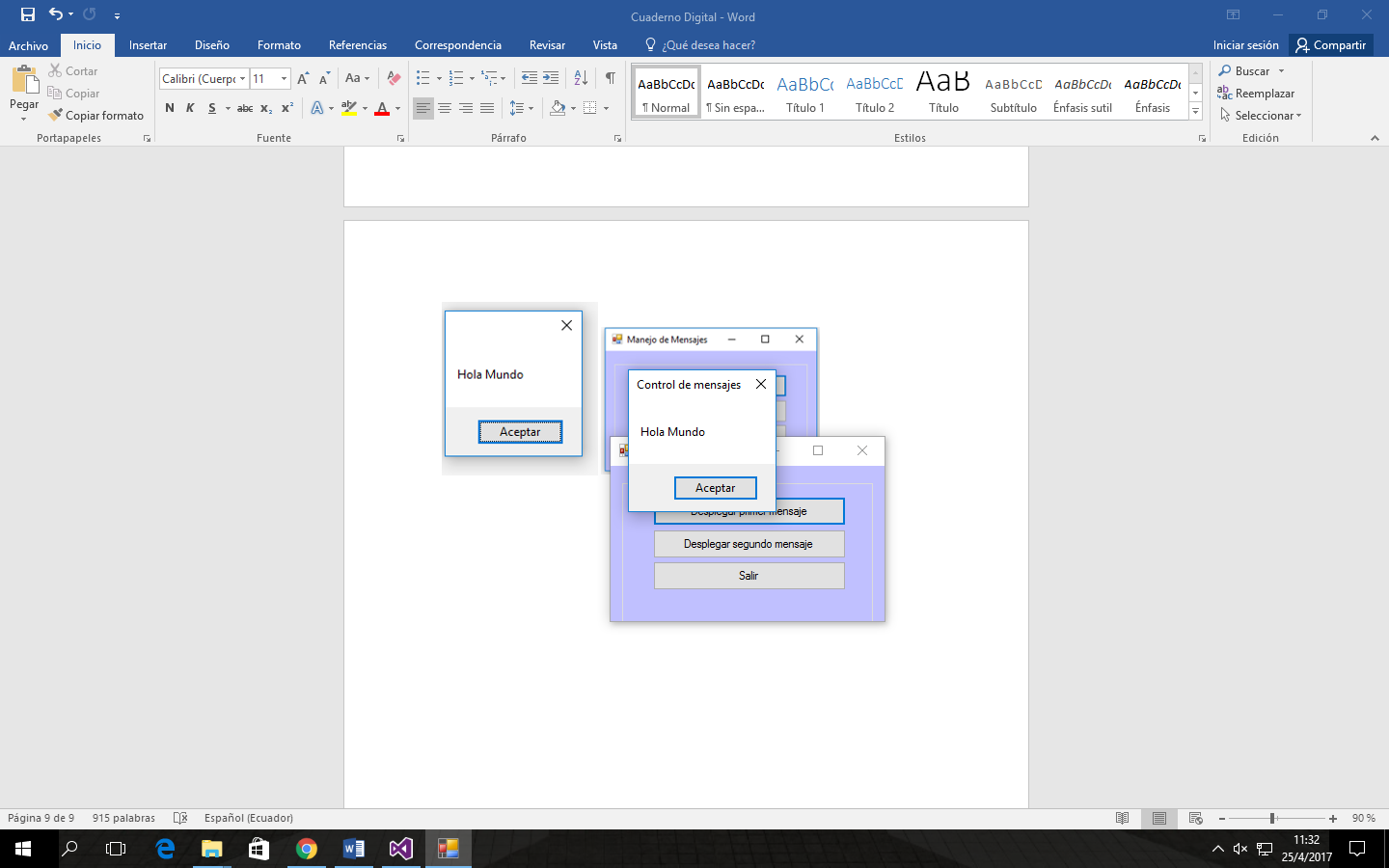
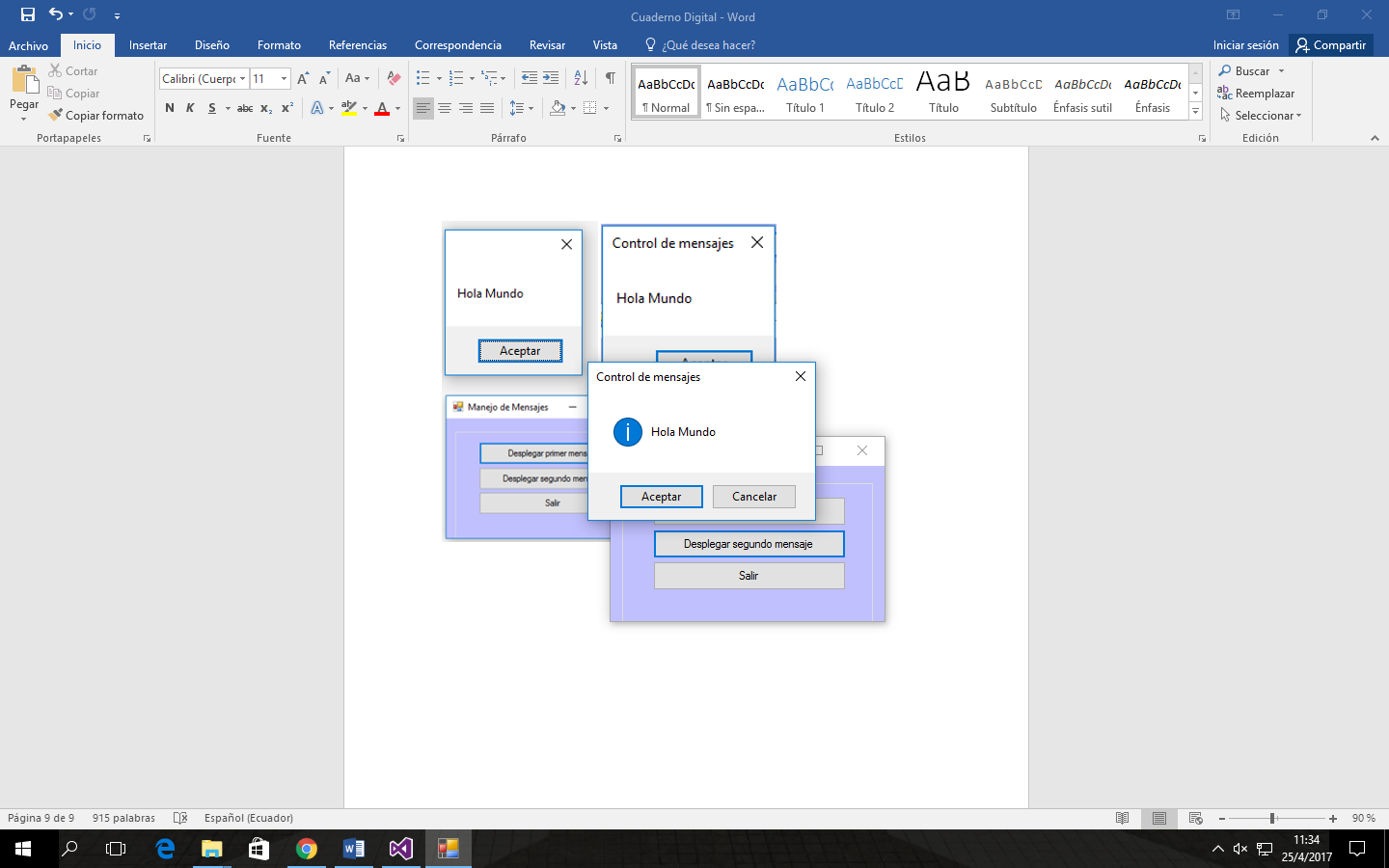
Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new frmMessages());

}

}

}



**Circulo**

p=2π r

A= π r2

**Rectangulo**

P=2a+2b

A=a.b

a^b > 0

**Cuadrado**

P=4a

A=a2

**Triangulo**

B

a

C

A

c

b

S=

A=

Formula de Herón

a, b, c > 0

**Teorema de existencia de un triangulo**

La suma de cualquiera de los dos lados de un triángulo abc tiene que ser mayor que el tercero

(a+b>c)^(a+c>b)^(b+c>a)

1

1

1

1

1

1

5

4

3

**Figuras Geométricas**

frmMenu 🡪 grbMenu

frmCircle btnCircle

frmSquare btnSquare

frmRectangle btnRectangle

frmTriangle btnTriangle

btnExit

grbInput grbProcess grbOutput

Label 🡪 Etiqueta

**Datos Miembro**

M 🡪 Prefijo

Ej: mRadius

Entradas salidas

Proceso p

A

Auxiliares π

**Castings – conversión de tipo de dato**

float.Parse (txtPerimeter.Text)

int.Parse (txtPerimeter.Text) 🡪 Cadena de caracteres

(float)Math.PI

txtPerimeter.Text = mPerimeter.ToString();

Enabled 🡪 Habilitar / deshabilitar

Propiedades True(default) / false

Locked 🡪 Bloquear / desbloquear

True / false(default)

FrmCircle

**WinAppGEometricSharpesV1**

frmMenu

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WinAppGeometricShapes

{

public partial class frmMenu : Form

{

public frmMenu()

{

InitializeComponent();

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void btnCircle\_Click(object sender, EventArgs e)

{

frmCircle ObjForm = new frmCircle();

ObjForm.Show();

}

private void btnSquare\_Click(object sender, EventArgs e)

{

frmSquare ObjForm = new frmSquare();

ObjForm.Show();

}

private void btnRectangle\_Click(object sender, EventArgs e)

{

frmRectangle ObjForm = new frmRectangle();

ObjForm.Show();

}

private void btnTriangle\_Click(object sender, EventArgs e)

{

frmTriangle ObjForm = new frmTriangle();

ObjForm.Show();

}

private void btnExit\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void frmMenu\_Load(object sender, EventArgs e)

{

}

}

}

**frmCircle**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WinAppGeometricShapes

{

public partial class frmCircle : Form

{

//Datos miembro - Atributos

private float mRadius;

private float mPerimeter;

private float mArea;

//Funciones Miembro - Metodos

//Constructor de la clase

public frmCircle()

{

InitializeComponent();

}

private void ReadData()

{

try

{

mRadius = float.Parse(txtRadius.Text);

}

catch

{

MessageBox.Show("Ingreso no válido...", "Mensaje de error");

}

}

private void PerimeterCircle()

{

mPerimeter = 2 \* (float)Math.PI \* mRadius;

}

private void AreaCircle()

{

//mPerimeter = 2 \* (float)Math.PI \* mRadius \* mRadius;

mArea = 2 \* (float)Math.PI \* (float)Math.Pow(mRadius,2);

}

private void PrintData()

{

txtPerimeter.Text = mPerimeter.ToString();

txtArea.Text = mArea.ToString();

}

private void Initialize()

{

txtPerimeter.Text = "";

txtArea.Text = "";

txtRadius.Text = "";

// Mantener el cursor titilando en una caja de texto.

txtRadius.Focus();

mRadius = 0.0f; mPerimeter = 0.0f; mArea = 0.0f;

}

private void frmCircle\_Load(object sender, EventArgs e)

{

Initialize();

}

private void btnCalculate\_Click(object sender, EventArgs e)

{

//Leer el valor del radio del circulo

ReadData();

//Calcular el perimetro del circulo

PerimeterCircle();

// Calcuar el area del circulo

AreaCircle();

// Imprimir los valores del perimetro y del area del circulo

PrintData();

}

private void btnReset\_Click(object sender, EventArgs e)

{

Initialize();

}

private void btnExit\_Click(object sender, EventArgs e)

{

Close();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

frmCricle

namespace WinAppGeometricShapes

{

public partial class frmCircle : Form

{

//Datos miembro - Atributos

private float mRadius;

private float mPerimeter;

private float mArea;

//Funciones Miembro - Metodos

//Constructor de la clase

public frmCircle()

{

InitializeComponent();

}

private void ReadData()

{

try

{

mRadius = float.Parse(txtRadius.Text);

}

catch

{

MessageBox.Show("Ingreso no válido...", "Mensaje de error");

}

}

private void PerimeterCircle()

{

mPerimeter = 2 \* (float)Math.PI \* mRadius;

}

private void AreaCircle()

{

//mPerimeter = 2 \* (float)Math.PI \* mRadius \* mRadius;

mArea = 2 \* (float)Math.PI \* (float)Math.Pow(mRadius,2);

}

private void PrintData()

{

txtPerimeter.Text = mPerimeter.ToString();

txtArea.Text = mArea.ToString();

}

private void Initialize()

{

txtPerimeter.Text = "";

txtArea.Text = "";

txtRadius.Text = "";

// Mantener el cursor titilando en una caja de texto.

txtRadius.Focus();

mRadius = 0.0f; mPerimeter = 0.0f; mArea = 0.0f;

}

private void frmCircle\_Load(object sender, EventArgs e)

{

Initialize();

}

private void btnCalculate\_Click(object sender, EventArgs e)

{

//Leer el valor del radio del circulo

ReadData();

//Calcular el perimetro del circulo

PerimeterCircle();

// Calcuar el area del circulo

AreaCircle();

// Imprimir los valores del perimetro y del area del circulo

PrintData();

}

private void btnReset\_Click(object sender, EventArgs e)

{

Initialize();

}

private void btnExit\_Click(object sender, EventArgs e)

{

Close();

}

}

}

**Principio de Encapsulamiento**

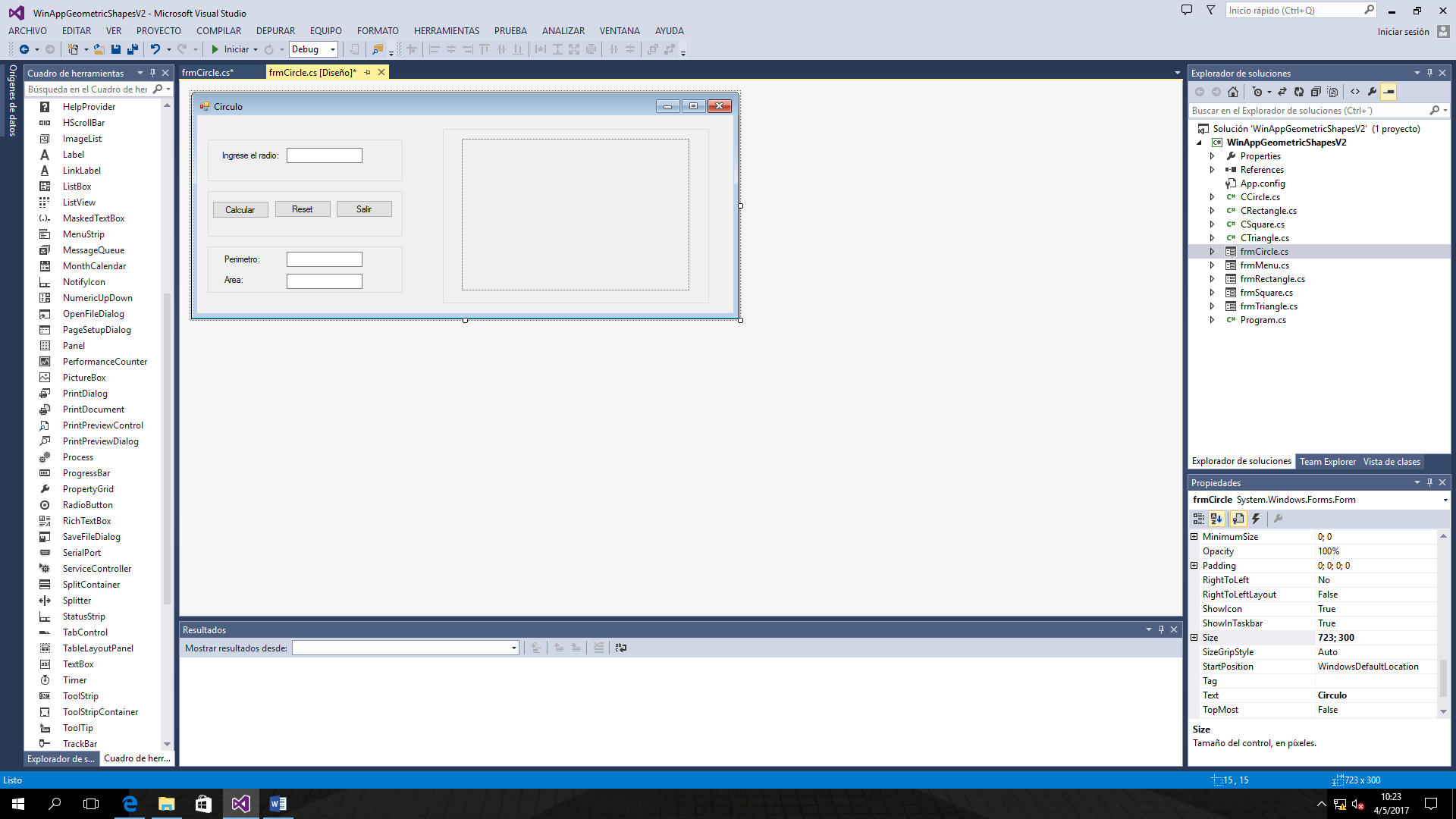
Un objeto puede acceder a funciones publicas

Acceder Acceder

Objeto 🡪 funciones públicas 🡪 datos miembro Privados

Modificar Funciones miembro publicas

Funciones miembro privadas



200 height

(y)

Largo

200 width

(x)

Ancho

(300;200)

(150;100)

(150;100)

(150;0)

(300;100)

(0;0)

(0;200)

(0;100)

PictureBox

picCanvas

(Lienzo) 🡪 Area de trabajo(dibujo)

Size: 300x200

Resolucion 300x200

(300;100)

**Ejes Coordenados (2D)**

(0;0)

Mundo Real

y(+)

x(+)

Cm y nm (metro)

(0;0)

Mundo De la Computación Grafica

y(+)

x(+)

Pixel (punto) nm

Graficacion de Figuras

PictureBox 🡪 Lienzo

mGraph 🡪 Activa y maneja el modo grafico (Graphics)

mPen 🡪Pluma Virtual (Pen)

Factor de escalamiento 🡪 Scale Factor(Zoom-In/Zoom out)

P1(b , 0) P2(b+l , 0) P0(0 , 0) Pm(2b+l , 0)

P3(0 , Ap) P4(2b+2l , Ap) Pn(0 , 2Ap) Pq(2b+l , 2Ap)

P5(b , 2Ap) P6(b+l , 2Ap)

**Sucesiones y Series**

Sucesión infinita transformar a sucesión finita

Sucesión infinita transformar a sucesión finita