

ESCUELA POLITECNICA DEL EJERCITO
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION
PRACTICAS DE LABORATORIO

PROFESOR	PRACTICA	GRUPO:	NOMBRE DE LA PRACTICA:	DURACIÓN:
ING. CÉSAR VILLACÍS, MGS. ING. MARGARITA ZAMBRANO, MGS.	No. 8	DOS PERSONAS	INTRODUCCIÓN A LA COMPUTACIÓN GRÁFICA CON C# Y GDI+	4 HORAS

NOMBRES (ALUMNOS)

..... CI: FIRMA:

..... CI: FIRMA:

1. INTRODUCCIÓN

El desarrollo de sistemas informáticos básicos enfocados al área de la Computación Gráfica, aplicando técnicas matemáticas y física básica, con un lenguaje de alto nivel como el C#, que den solución a un problema específico de un área del conocimiento de las Ciencias de la Computación, cada vez más creciente, con eficiencia y eficacia en la elaboración de algoritmos y responsabilidad profesional, hace que esta asignatura sea fundamental dentro de la formación de los estudiantes de la carrera de Ingeniería de Sistemas e Informática.

La asignatura de Computación Gráfica corresponde a la primera etapa del eje de formación profesional, proporciona al futuro profesional los principios y fundamentos de la computación gráfica, donde se utilizan computadoras tanto para generar imágenes visuales sintéticamente como integrar o cambiar la información visual y espacial probada del mundo real; y desarrolla en el estudiantes las habilidades de análisis, diseño, construcción y documentación de aplicaciones desarrolladas mediante motores de gráficos en 2D y motores de videojuegos en 2D, como el MonoGame/XNA, NetGore, entre otros.

En la primera unidad temática referente a Mecanismos de Abstracción con C# y GDI+, se van a diseñar y desarrollar aplicaciones de Computación Gráfica que den solución a problemas de ingeniería mediante el uso de clases y objetos, elementos matemáticos y motores de gráficos y juegos con técnicas de programación orientadas a objetos.

2. OBJETIVO (COMPETENCIA)

Diseñar y desarrollar programas que utilicen clases, objetos, librerías gráficas y aplicaciones gráficas utilizando líneas, puntos, funciones y figuras geométricas con GDI+ y C#.

3. FUNDAMENTO

- Conceptos entorno a la utilización de clases y objetos.
- Conceptos entorno a la utilización de librerías gráficas.
- Conceptos entorno a al desarrollo de aplicaciones gráficas utilizando líneas, puntos,

<p style="text-align: center;">ESCUELA POLITECNICA DEL EJERCITO DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION PRACTICAS DE LABORATORIO</p>
--

funciones y figuras geométricas con GDI+ y C#.
--

4. DESARROLLO DE LA PRACTICA (INDICAR LOS COMANDOS UTILIZADOS)

Utilizando la herramienta de desarrollo, implementar una aplicación en modo visual que resuelva lo siguiente:

1. Crear una interfaz gráfica de usuario con controles visuales (botones, cajas de texto, cajas de grupo, etc.) que permita graficar y realizar los cálculos de áreas y perímetros de las siguientes figuras geométricas:
 - a. Círculo.
 - b. Rectángulo.
 - c. Cuadrado.
 - d. Líneas.
 - e. Triángulos.
2. Validar las opciones en el caso que el usuario ingrese un dato inválido.
3. Implementar las clases: CCircle, CRectangle, CSquare, CLine, CTriangle, donde cada clase contemple datos y funciones miembros para cada una de las operaciones indicadas anteriormente.
4. Cada opción deberá llamar a la función correspondiente utilizando un objeto.
5. Mostrar en controles visuales el resultado de cada operación.

Una vez finalizado el taller, el grupo de trabajo guardará en una memoria flash el proyecto realizado, bajo el siguiente formato:

PRACTICA#_GRUPO#_FECHA-PRACTICA
Ejemplo: PRACTICA8_GRUPO3_26102015

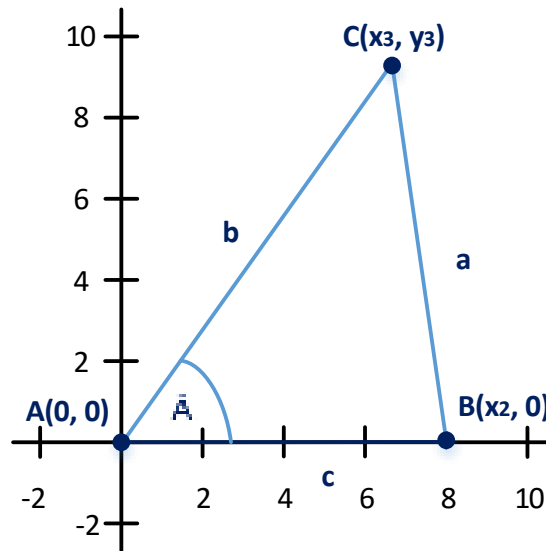
5. RESULTADOS Y CONCLUSIONES

- Aplicar conocimientos de POO en la construcción de aplicaciones de computación gráfica que utilicen clases, objetos, librerías gráficas y aplicaciones gráficas utilizando líneas, puntos, funciones y figuras geométricas con GDI+ y C#.
- Entregar prácticas e informes de laboratorio sobre: a) Clases y objetos; b) Elementos de la computación gráfica; c) Creación de juegos con GDI+; usando el lenguaje C#.

IMPLEMENTACIÓN

Geometría Analítica del Triángulo

Dados los tres lados de un triángulo, encontrar los valores de los vértices en coordenadas rectangulares y graficar las líneas del triángulo, considerando que uno de sus lados se encuentra sobre el eje de las 'x' y uno de sus vértices es el origen.



Solución:

Dados los siguientes puntos:

$$\begin{aligned} A(0,0) &= P_1(x_1, y_1) \\ B(x_B, 0) &= P_2(x_2, y_2) \\ C(x_C, y_C) &= P_3(x_3, y_3) \end{aligned}$$

Calculamos la distancia **AB**:

$$\begin{aligned} d_{AB} &= d_{P_1 P_2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = c \\ c &= \sqrt{(x_2 - 0)^2 + (0 - 0)^2} \\ c &= \sqrt{x_2^2} \\ c &= x_2 \end{aligned} \quad (1)$$

Por lo tanto las coordenadas del punto **P₁** y **P₂** son:

$$A(0,0) = P_1(0,0) \quad (2)$$

$$B(x_B, 0) = P_2(c, 0) \quad (3)$$

ESCUELA POLITECNICA DEL EJERCITO DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION PRACTICAS DE LABORATORIO

Para calcular el valor del ángulo **A**, se utiliza la Ley de Cosenos:

$$\cos(A) = \frac{b^2 + c^2 - a^2}{2bc}$$

$$A = \arccos\left(\frac{b^2 + c^2 - a^2}{2bc}\right)$$

Calculamos el valor de la pendiente **AC**:

$$m_{AC} = \frac{y_3 - y_1}{x_3 - x_1} = tg(A)$$

$$tg(A) = \frac{y_3 - 0}{x_3 - 0}$$

$$tg(A) = \frac{y_3}{x_3}$$

Despejamos el valor de y_3 donde obtenemos la siguiente ecuación:

$$y_3 = x_3 \cdot tg(A) \quad (4)$$

Calculamos la distancia **AC**:

$$d_{AC} = d_{P_1 P_3} = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2} = b$$

$$b = \sqrt{(x_3 - 0)^2 + (y_3 - 0)^2}$$

$$b = \sqrt{x_3^2 + y_3^2} \quad (5)$$

Reemplazamos el valor de y_3 de la ecuación (4) en la ecuación (5) y calculamos el valor de **x₃**:

$$b = \sqrt{x_3^2 + (x_3 \cdot tg(A))^2}$$

$$b = \sqrt{x_3^2 + x_3^2 \cdot tg^2(A)}$$

$$b = \sqrt{x_3^2(1 + tg^2(A))}$$

$$b = x_3 \cdot \sqrt{1 + tg^2(A)}$$

Despejamos el valor de **x₃** y luego racionalizamos la expresión:

$$x_3 = \frac{b}{\sqrt{1 + tg^2(A)}} \cdot \frac{\sqrt{1 + tg^2(A)}}{\sqrt{1 + tg^2(A)}}$$

$$x_3 = \frac{b \cdot \sqrt{1 + tg^2(A)}}{1 + tg^2(A)} \quad (6)$$

<p style="text-align: center;">ESCUELA POLITECNICA DEL EJERCITO DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION PRACTICAS DE LABORATORIO</p>
--

Reemplazamos el valor de x_3 de la ecuación (6) en la ecuación (4) y calculamos el valor de y_3 :

$$y_3 = \frac{b \cdot \sqrt{1 + \operatorname{tg}^2(A)}}{1 + \operatorname{tg}^2(A)} \cdot \operatorname{tg}(A) \quad (7)$$

Por lo tanto las coordenadas del punto P_3 son:

$$C(x_c, y_c) = P_3 \left(\frac{b \cdot \sqrt{1 + \operatorname{tg}^2(A)}}{1 + \operatorname{tg}^2(A)}, \frac{b \cdot \sqrt{1 + \operatorname{tg}^2(A)}}{1 + \operatorname{tg}^2(A)} \cdot \operatorname{tg}(A) \right) \quad (8)$$

Desarrollo e Implementación del Programa

El desarrollo e implementación de la clase `CTriangle`, se presenta a continuación en la Tabla 1, que es parte de la solución del ejercicio.

Los atributos principales que se tienen en la clase `CTriangle` son los tres lados que tiene un triángulo representados por los datos miembros mA , mB y mC que son de tipo `float`. Además se han declarado un grupo de datos miembro que permiten operar con el modo gráfico de la librería `System.Drawing` de los componentes de `GDI+`.

Esta clase también posee un grupo de métodos que de manera similar a la clase `CRectangle`, realizan diferentes operaciones. Se definirán algunos métodos para la clase `CTriangle` de acuerdo a las acciones que puede hacer y que básicamente serán funciones para poder representar y operar un triángulo en el plano.

Tabla 1: Código de la clase `CTriangle` del programa.

```
class CTriangle
{
    // Datos miembro - Atributos.
    private float mA, mB, mC;
    private float mPerimeter, mArea;

    // Datos miembro que operan con el modo gráfico.
    private Graphics mGraph;
    private Pen mPen;
    private const float SF = 20;
    private PointF mP1, mP2, mP3;
    private float mAngleA;

    // Funciones miembro - Métodos.

    // Constructor por defecto.
    public CTriangle()
```

ESCUELA POLITECNICA DEL EJERCITO
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION
PRACTICAS DE LABORATORIO

```
{
    mA = 0.0f; mB = 0.0f; mC = 0.0f;
    mPerimeter = 0.0f; mArea = 0.0f;
}

// Función que permite leer los tres lados del triángulo.
public void ReadData(TextBox txtSideA,
                    TextBox txtSideB,
                    TextBox txtSideC)
{
    mA = float.Parse(txtSideA.Text);
    mB = float.Parse(txtSideB.Text);
    mC = float.Parse(txtSideC.Text);
}

// Función que permite calcular el semi-perímetro del triángulo.
private float SemiperimeterTriangle()
{
    return ((mA + mB + mC) / 2);
}

// Función que permite calcular el perímetro del triángulo.
public void PerimeterTriangle()
{
    float s = SemiperimeterTriangle();
    mPerimeter = 2 * s;
}

// Función que permite calcular el área del triángulo.
public void AreaTriangle()
{
    float s = SemiperimeterTriangle();
    mArea = (float)Math.Sqrt(s * (s - mA) * (s - mB) * (s - mC));
}

// Función que permite imprimir el perímetro y el área del triángulo.
public void PrintData(TextBox txtPerimeter, TextBox txtArea)
{
    txtPerimeter.Text = mPerimeter.ToString();
    txtArea.Text = mArea.ToString();
}

// Función que permite inicializar los datos y controles que operan en
// la GUI del triángulo.
public void InitializeData(TextBox txtSideA,
                        TextBox txtSideB,
                        TextBox txtSideC,
                        TextBox txtPerimeter,
                        TextBox txtArea,
                        PictureBox picCanvas)
{
    txtSideA.Text = ""; txtSideB.Text = ""; txtSideC.Text = "";
}
```

ESCUELA POLITECNICA DEL EJERCITO
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION
PRACTICAS DE LABORATORIO

```
txtPerimeter.Text = ""; txtArea.Text = "";

// Mantiene el cursor titilando en una caja de texto.
txtSideA.Focus();

mA = 0.0f; mB = 0.0f; mC = 0.0f;
mPerimeter = 0.0f; mArea = 0.0f;

picCanvas.Refresh();
}

// Función que permite validar la existencia de un triángulo.
public bool CheckTriangle()
{
    if ((mA + mB > mC) && (mA + mC > mB) && (mB + mC > mA))
    {
        return (true);
    }
    else // !((mA + mB > mC) && (mA + mC > mB) && (mB + mC > mA))
        return (false);
}

// Función que permite calcular el ángulo A del triángulo, utilizando
// la Ley de Cosenos.
private void CalculateAngleA()
{
    mAngleA = (float)Math.Acos((mB * mB + mC * mC - mA * mA) / (2 * mB * mC));
}

// Función que permite calcular los valores de los tres vértices del triángulo,
// utilizando fórmulas de Geometría Analítica.
private void CalculateVertexC()
{
    mP1.X = 0.0f; mP1.Y = 0.0f;
    mP2.X = mC; mP2.Y = 0.0f;

    CalculateAngleA();

    mP3.X = mB * (float)Math.Cos(mAngleA);
    mP3.Y = mB * (float)Math.Sin(mAngleA);
}

// Función que permite graficar un triángulo en base a los valores de los tres
// vértices representados por tres puntos en un plano.
public void GraphShape(PictureBox picCanvas)
{
    mGraph = picCanvas.CreateGraphics();
    mPen = new Pen(Color.Blue, 3);

    CalculateVertexC();

    mGraph.DrawLine(mPen, mP1.X * SF, mP1.Y * SF, mP2.X * SF, mP2.Y * SF);
}
```

<p style="text-align: center;">ESCUELA POLITECNICA DEL EJERCITO</p> <p style="text-align: center;">DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION</p> <p style="text-align: center;">PRACTICAS DE LABORATORIO</p>
--

```

        mGraph.DrawLine(mPen, mP1.X * SF, mP1.Y * SF, mP3.X * SF, mP3.Y * SF);
        mGraph.DrawLine(mPen, mP2.X * SF, mP2.Y * SF, mP3.X * SF, mP3.Y * SF);
    }

}

```

La implementación de la clase *frmTriangle*, se presenta a continuación en la Tabla 2, que es la parte que representa a la interfaz gráfica de usuario de la figura geométrica conocida como Triángulo.

Tabla 2: Código de la clase *frmTriangle* del programa.

```

public partial class frmTriangle : Form
{
    // Datos miembro - atributos.
    private CTriangle ObjTriangle = new CTriangle();

    // Constructor del formulario.
    public frmTriangle()
    {
        InitializeComponent();
        ObjTriangle.InitializeData(txtSideA, txtSideB, txtSideC,
                                   txtPerimeter, txtArea, picCanvas);
    }

    // Función que maneja el evento Load (Carga) del formulario.
    private void frmTriangle_Load(object sender, EventArgs e)
    {
        ObjTriangle.InitializeData(txtSideA, txtSideB, txtSideC,
                                   txtPerimeter, txtArea, picCanvas);
    }

    // Función que maneja el evento Click (Clic) del botón btnCalculate.
    private void btnCalculate_Click(object sender, EventArgs e)
    {
        ObjTriangle.ReadData(txtSideA, txtSideB, txtSideC);

        bool varControl = ObjTriangle.CheckTriangle();

        if (varControl == true)
        {
            ObjTriangle.PerimeterTriangle();
            ObjTriangle.AreaTriangle();
            ObjTriangle.PrintData(txtPerimeter, txtArea);

            ObjTriangle.GraphShape(picCanvas);
        }
        else // varControl != true
            MessageBox.Show("Error. El triángulo no existe...",
                            "Mensaje de control");
    }
}

```


ESCUELA POLITECNICA DEL EJERCITO
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION
PRACTICAS DE LABORATORIO

```
// Función que maneja el evento Click (Clic) del botón btnReset.  
private void btnReset_Click(object sender, EventArgs e)  
{  
    ObjTriangle.InitializeData(txtSideA, txtSideB, txtSideC,  
                               txtPerimeter, txtArea, picCanvas);  
}  
  
// Función que maneja el evento Click (Clic) del botón btnExit.  
private void btnExit_Click(object sender, EventArgs e)  
{  
    Close();  
}  
}
```

En la Tabla 3, se muestra una de las salidas o corridas de este programa en el cual se ha instanciado un objeto *ObjTriangle*, que invoca a algunos métodos de la clase *CTriangle*.

Tabla 3: Salida o corrida del programa con el formulario *frmTriangle*.

The screenshot shows a Windows application window titled "Triángulo". The window is divided into three main sections on the left and a large canvas on the right.

- Entradas (Inputs):** Three text boxes labeled "Ingrese el lado a:", "Ingrese el lado b:", and "Ingrese el lado c:". All three boxes contain the value "6".
- Proceso (Process):** Three buttons: "Calcular" (highlighted with a blue border), "Resetear", and "Salir".
- Salidas (Outputs):** Two text boxes. The first is labeled "Perimetro:" and contains the value "18". The second is labeled "Área:" and contains the value "15,58846".
- Canvas:** A large rectangular area on the right side of the window displaying a blue outline of an equilateral triangle.