



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

Departamento de Computación

Computación Gráfica 2283

Introducción

El presente informe documentará el desarrollo de una aplicación en C# que permite realizar cálculos de áreas y perímetros de diversas figuras geométricas mediante el uso de formularios. Esta aplicación, implementada bajo el entorno de desarrollo .NET Framework y utilizando formularios de Windows Forms, tiene como propósito familiarizarse con el entorno de VS así como repasar fundamentos claves de programación orientada a objetos y del uso de varios formularios y subformularios

El proyecto incluye formularios dedicados a cada tipo de figura geométrica, como cuadrados, rectángulos, círculos y triángulos, donde el usuario puede ingresar los parámetros correspondientes, por ejemplo, lados, radio o base y altura. A partir de estos datos, la aplicación realiza los cálculos necesarios y muestra los resultados de manera clara y dinámica. Además, se incorpora un diseño basado en principios de programación estructurada y orientada a objetos, haciendo uso de patrones como el Singleton para optimizar el manejo de ventanas y evitar redundancias en la ejecución.

Requerimientos

Entrada de datos

La entrada de datos para cada formulario está dada por solo los inputs de cada dato que se necesita de la figura y en la parte de las clases se guarda en variables flotantes

Formularios

txtNombreDelInput /*Objeto Tipo TextBox*/

Clases

mNombredelDato /*Objeto Tipo Flotante*/

Salida de datos

La salida de datos está dada por cajas de texto las cuales están inhabilitadas para editar y bloqueadas para poder mover las cajas

Formularios

txtNombredelOutput /*Objeto Tipo TextBox*/

Diseños de la solución

Dentro de los diseños de la solución, cada figura cuenta con sus respectivos textbox para las entradas de datos y va a depender de cada figura, también se aplica un patrón de diseño singleton en los formularios de cada figura para que se cree una única instancia de esta forma evitando formularios repetidos, dentro de la clase del formulario se encuentra los métodos de cada botón y su método para inicializar datos.

Cuadrado

```
public partial class FrmSquare : Form
{
    Square square = new Square();

    private static FrmSquare instance;

    1 referencia
    public FrmSquare()
    {
        InitializeComponent();
    }

    2 referencias
    public static FrmSquare GetInstance() ...

    2 referencias
    public void InitializeData() ...

    1 referencia
    private void btnExit_Click(object sender, EventArgs e) ...

    1 referencia
    private void btnReset_Click(object sender, EventArgs e) ...

    1 referencia
    private void btnCalculate_Click(object sender, EventArgs e) ...

    1 referencia
    private void FrmSquare_Load(object sender, EventArgs e) ...
}
```

Triangulo

Menu

Figuras

Triangulo

Entradas

Lado 1

Lado 2

Lado 3

Altura

Procesos

Calcular Resetear Regresar

Salidas

Perimetro

Area

```
7 referencias
public partial class FrmTriangle : Form
{
    Triangle triangle = new Triangle();

    private static FrmTriangle instance;

    1 referencia
    public FrmTriangle()
    {
        InitializeComponent();
    }

    1 referencia
    public static FrmTriangle GetInstance() ...

    2 referencias
    public void InitializeData() ...

    1 referencia
    private void FrmTriangle_Load(object sender, EventArgs e) ...

    1 referencia
    private void btnCalculate_Click(object sender, EventArgs e) ...

    1 referencia
    private void btnReset_Click(object sender, EventArgs e) ...

    1 referencia
    private void btnExit_Click(object sender, EventArgs e) ...
}
}
```

Rectángulo

The screenshot shows a Windows application window titled "Figuras". Inside the window is a smaller form titled "Rectangulo". The "Rectangulo" form is divided into three sections: "Entradas" (Inputs), "Procesos" (Processes), and "Salidas" (Outputs). In the "Entradas" section, there are two text boxes labeled "Ancho" (Width) and "Altura" (Height). In the "Procesos" section, there are three buttons: "Calcular" (Calculate), "Resetear" (Reset), and "Regresar" (Return). In the "Salidas" section, there are two text boxes labeled "Perimetro" (Perimeter) and "Area". The "Calcular" button is highlighted with a blue border.

```
namespace Practica_6
{
    7 referencias
    public partial class FrmRectangle : Form
    {
        Rectangle rectangle = new Rectangle();

        private static FrmRectangle instance;

        1 referencia
        public FrmRectangle() ...

        1 referencia
        public static FrmRectangle GetInstance() ...

        2 referencias
        public void InitializeData() ...
        1 referencia
        private void FrmRectangle_Load(object sender, EventArgs e) ...

        1 referencia
        private void btnReset_Click(object sender, EventArgs e) ...

        1 referencia
        private void btnExit_Click(object sender, EventArgs e) ...

        1 referencia
        private void btnCalculate_Click(object sender, EventArgs e) ...
    }
}
```

Paralelogramo

Menu

Figuras

Paralelogramo

Entradas

Base

Lado

Altura

Procesos

Calcular Resetear Regresar

Salidas

Perimetro

Area

```
namespace Practica_6
{
    7 referencias
    public partial class FrmParalelogram : Form
    {
        Paralelogram paralelogram = new Paralelogram();

        private static FrmParalelogram instance;

        1 referencia
        public FrmParalelogram() ...

        1 referencia
        public static FrmParalelogram GetInstance() ...

        2 referencias
        public void InitializeData() ...

        1 referencia
        private void grbInputs_Enter(object sender, EventArgs e) ...

        1 referencia
        private void btnCalculate_Click(object sender, EventArgs e) ...

        1 referencia
        private void btnReset_Click(object sender, EventArgs e) ...

        1 referencia
        private void FrmParalelogram_Load(object sender, EventArgs e) ...

        1 referencia
        private void btnExit_Click(object sender, EventArgs e) ...
    }
}
```

Rombo

Menu

Figuras

Rombo

Entradas

Diagonal Superior.

Diagonal Inferior

Lado

Procesos

Calcular Resetear Regresar

Salidas

Perimetro

Area

```
namespace Practica_6
{
    7 referencias
    public partial class FrmDiamond : Form
    {
        Diamond diamond = new Diamond();

        private static FrmDiamond instance;
        1 referencia
        public FrmDiamond() ...

        1 referencia
        public static FrmDiamond GetInstance() ...

        2 referencias
        public void InitializeData() ...

        1 referencia
        private void FrmDiamond_Load(object sender, EventArgs e) ...

        1 referencia
        private void btnCalculate_Click(object sender, EventArgs e) ...

        1 referencia
        private void btnReset_Click(object sender, EventArgs e) ...

        1 referencia
        private void btnExit_Click(object sender, EventArgs e) ...
    }
}
```

Cometa

Menu

Figuras

Cometa

Entradas

Diagonal Superior

Diagonal Inferior

Lado Menor

Lado Mayor

Procesos

Calcular Resetear Regresar

Salidas

Perimetro

Area

```
namespace Practica_6
{
    7 referencias
    public partial class FrmComet : Form
    {
        Comet comet = new Comet();

        private static FrmComet instance;

        1 referencia
        public FrmComet() ...

        1 referencia
        public static FrmComet GetInstance() ...

        2 referencias
        public void InitializeData() ...

        1 referencia
        private void FrmComet_Load(object sender, EventArgs e) ...

        1 referencia
        private void btnReset_Click(object sender, EventArgs e) ...

        1 referencia
        private void btnExit_Click(object sender, EventArgs e) ...

        1 referencia
        private void btnCalculate_Click(object sender, EventArgs e) ...
    }
}
```

Trapezio

Menu

Figuras

Trapezio

Entradas

Base Mayor

Base Menor

Cateto 1

Cateto 2

Altura

Procesos

Salidas

Perimetro

Area

```
namespace Practica_6
{
    7 referencias
    public partial class FrmTrapeze : Form
    {

        Trapeze trapeze = new Trapeze();

        private static FrmTrapeze instance;

        1 referencia
        public FrmTrapeze() ...

        1 referencia
        public static FrmTrapeze GetInstance() ...

        2 referencias
        public void InitializeData() ...

        1 referencia
        private void FrmTrapeze_Load(object sender, EventArgs e) ...

        1 referencia
        private void btnReset_Click(object sender, EventArgs e) ...

        1 referencia
        private void btnExit_Click(object sender, EventArgs e) ...

        1 referencia
        private void btnCalculate_Click(object sender, EventArgs e) ...

    }
}
```


Circulo

The screenshot shows a Windows application window titled "Figuras". Inside the window is a smaller form titled "Circulo". The "Circulo" form has three sections: "Entradas" (Inputs) with a "Radio" text box; "Procesos" (Processes) with three buttons labeled "Calcular", "Resetear", and "Regresar"; and "Salidas" (Outputs) with two text boxes labeled "Perimetro" and "Area".

```
namespace Practica_6
{
    7 referencias
    public partial class FrmCircle : Form
    {
        Circle circle = new Circle();

        private static FrmCircle instance;

        1 referencia
        public FrmCircle() ...

        1 referencia
        public static FrmCircle GetInstance() ...

        2 referencias
        public void InitializeData() ...

        1 referencia
        private void FrmCircle_Load(object sender, EventArgs e) ...

        1 referencia
        private void btnReset_Click(object sender, EventArgs e) ...

        1 referencia
        private void btnExit_Click(object sender, EventArgs e) ...

        1 referencia
        private void btnCalculate_Click(object sender, EventArgs e) ...
    }
}
```

Polígono Regular

The screenshot shows a Windows application window titled "Figuras". Inside the window is a smaller window titled "Poligono". This window is divided into three sections:

- Entradas:** Contains three input fields labeled "Apotema", "Lado", and "Número de Lados".
- Procesos:** Contains three buttons labeled "Calcular", "Resetear", and "Regresar".
- Salidas:** Contains two output fields labeled "Perimetro" and "Area".

```
namespace Practica_6
{
    7 referencias
    public partial class FrmRegularPoligon : Form
    {
        RegularPoligon regularPoligon = new RegularPoligon();

        private static FrmRegularPoligon instance;

        1 referencia
        public FrmRegularPoligon()...

        1 referencia
        public static FrmRegularPoligon GetInstance() ...

        2 referencias
        public void InitializeData()...
        1 referencia
        private void FrmRegularPoligon_Load(object sender, EventArgs e)...

        1 referencia
        private void btnReset_Click(object sender, EventArgs e)...

        1 referencia
        private void btnExit_Click(object sender, EventArgs e)...

        1 referencia
        private void btnCalculate_Click(object sender, EventArgs e)...
    }
}
```

Corona Circular

Menu

Figuras

Corona Circular

Entradas

Radio Mayor

Radio Menor

Procesos

Calcular Resetear Regresar

Salidas

Area

```
namespace Practica_6
{
    7 referencias
    public partial class FrmCrown : Form
    {
        Crown crown = new Crown();

        private static FrmCrown instance;

        1 referencia
        public static FrmCrown GetInstance() ...

        1 referencia
        public FrmCrown() ...

        2 referencias
        public void InitializeData() ...

        1 referencia
        private void FrmCrown_Load(object sender, EventArgs e) ...

        1 referencia
        private void btnReset_Click(object sender, EventArgs e) ...

        1 referencia
        private void btnExit_Click(object sender, EventArgs e) ...

        1 referencia
        private void btnCalculate_Click(object sender, EventArgs e) ...
    }
}
```

Sector Circular

The screenshot shows a Windows application window titled "Figuras" with a standard Windows title bar (minimize, maximize, close buttons). Inside the window is a smaller form titled "Sector Circular". This form is divided into three sections: "Entradas" (Inputs), "Procesos" (Processes), and "Salidas" (Outputs). The "Entradas" section contains two text input fields labeled "Radio" and "Grados". The "Procesos" section contains three buttons: "Calcular" (highlighted with a blue border), "Resetear", and "Regresar". The "Salidas" section contains a single text output field labeled "Area".

```
namespace Practica_6
{
    7 referencias
    public partial class FrmCircleSector : Form
    {
        CircleSector sector = new CircleSector();

        private static FrmCircleSector instance;

        1 referencia
        public FrmCircleSector()...

        1 referencia
        public static FrmCircleSector GetInstance() ...

        2 referencias
        public void InitializeData()...

        1 referencia
        private void FrmCircleSector_Load(object sender, EventArgs e)...

        1 referencia
        private void btnCalculate_Click(object sender, EventArgs e)...

        1 referencia
        private void btnReset_Click(object sender, EventArgs e)...

        1 referencia
        private void btnExit_Click(object sender, EventArgs e)...
    }
}
```

Implementación

Dentro de la implementación de la solución, se encuentran las clases de cada figura, se tiene un constructor público para inicializar las variables de la clase, cada clase tiene su método para recibir los respectivos datos de la figura, asignarlos a las variables, realizar los cálculos del perímetro y del área y por último el método para escribir dentro del formulario

Cuadrado

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Practica_6
{
    3 referencias
    internal class Square
    {
        private float mSide;
        private float mPerimeter;
        private float mArea;

        1 referencia
        public Square() ...

        1 referencia
        public void ReadData(TextBox txtSide) ...

        1 referencia
        public void CalculatePerimeter() ...

        1 referencia
        public void CalculateArea() ...

        1 referencia
        public void WriteData(TextBox txtPerimeter, TextBox txtArea) ...
    }
}
```

Triangulo

```
namespace Practica_6
{
    3 referencias
    internal class Triangle
    {
        private float mSide1;
        private float mSide2;
        private float mSide3;
        private float mHeight;
        private float mPerimeter;
        private float mArea;

        1 referencia
        public Triangle() ...

        1 referencia
        public void ReadData(TextBox txtSide1, TextBox txtSide2, TextBox txtSide3, TextBox txtHei

        1 referencia
        public void CalculatePerimeter() ...

        1 referencia
        public void CalculateArea() ...

        1 referencia
        public void WriteData(TextBox txtPerimeter, TextBox txtArea) ...
    }
}
```

Rectángulo

```
namespace Practica_6
{
    3 referencias
    internal class Rectangle
    {
        private float mWidth;
        private float mHeight;
        private float mPerimeter;
        private float mArea;

        1 referencia
        public Rectangle()...

        1 referencia
        public void ReadData(TextBox txtWidth, TextBox txtHeight)...

        1 referencia
        public void CalculatePerimeter()...

        1 referencia
        public void CalculateArea()...

        1 referencia
        public void WriteData(TextBox txtPerimeter, TextBox txtArea)...
    }
}
```

Paralelogramo

```
namespace Practica_6
{
    3 referencias
    internal class Parallelogram
    {
        private float mBase;
        private float mHeight;
        private float mSide;
        private float mPerimeter;
        private float mArea;

        1 referencia
        public Parallelogram()...

        1 referencia
        public void ReadData(TextBox txtBase, TextBox txtSide, TextBox txtHeight)...

        1 referencia
        public void CalculatePerimeter()...

        1 referencia
        public void CalculateArea()...

        1 referencia
        public void WriteData(TextBox txtPerimeter, TextBox txtArea)...
    }
}
```

Rombo

```
namespace Practica_6
{
    3 referencias
    internal class Diamond
    {
        private float mMajorDiagonal;
        private float mMinorDiagonal;
        private float mSide;
        private float mPerimeter;
        private float mArea;

        1 referencia
        public Diamond()...

        1 referencia
        public void ReadData(TextBox txtMajorDiagonal, TextBox txtMinorDiagonal, TextBox txtSide

        1 referencia
        public void CalculatePerimeter()...

        1 referencia
        public void CalculateArea()...

        1 referencia
        public void WriteData(TextBox txtPerimeter, TextBox txtArea)...
```

Cometa

```
namespace Practica_6
{
    3 referencias
    internal class Comet
    {
        private float mMajorDiagonal;
        private float mMinorDiagonal;
        private float mMinorSide;
        private float mMajorSide;
        private float mPerimeter;
        private float mArea;

        1 referencia
        public Comet()...

        1 referencia
        public void ReadData(TextBox txtMajorDiagonal, TextBox txtMinorDiagonal, TextBox txtMinor

        public void CalculatePerimeter()...

        public void CalculateArea()...

        public void WriteData(TextBox txtPerimeter, TextBox txtArea)...
```

Trapecio

```
namespace Practica_6
{
    3 referencias
    internal class Trapeze
    {
        private float mSide1;
        private float mSide2;
        private float mMajorBase;
        private float mMinorBase;
        private float mHeight;
        private float mPerimeter;
        private float mArea;

        1 referencia
        public Trapeze()...

        1 referencia
        public void ReadData(TextBox txtMajorBase, TextBox txtMinorBase,
            |   TextBox txtSide1, TextBox txtSide2, TextBox txtHeight)...

        1 referencia
        public void CalculatePerimeter()...

        1 referencia
        public void CalculateArea()...

        1 referencia
        public void WriteData(TextBox txtPerimeter, TextBox txtArea)...
```

Circulo

```
namespace Practica_6
{
    3 referencias
    internal class Circle
    {
        private float mRadius;
        private float mPerimeter;
        private float mArea;
        private const float pi = 3.1416F;

        1 referencia
        public Circle()...

        1 referencia
        public void ReadData(TextBox txtRadius)...
```

```
        1 referencia
        public void CalculatePerimeter()...

        1 referencia
        public void CalculateArea()...

        1 referencia
        public void WriteData(TextBox txtPerimeter, TextBox txtArea)...
```


Polígono Regular

```
namespace Practica_6
{
    3 referencias
    internal class RegularPoligon
    {
        private float mSide;
        private float mApotema;
        private int mSideNumber;
        private float mPerimeter;
        private float mArea;

        1 referencia
        public RegularPoligon()...

        1 referencia
        public void ReadData(TextBox txtApotema, TextBox txtSide, TextBox txtSideNumber)...

        1 referencia
        public void CalculatePerimeter()...

        1 referencia
        public void CalculateArea()...

        1 referencia
        public void WriteData(TextBox txtPerimeter, TextBox txtArea)...
    }
}
```

Corona Circular

```
namespace Practica_6
{
    3 referencias
    internal class Crown
    {
        private float mMajorRadius;
        private float mMinorRadius;
        private float mArea;
        private const float pi = 3.1416F;

        1 referencia
        public Crown()...

        1 referencia
        public void ReadData(TextBox txtMajorRadius, TextBox txtMinorRadius)...

        1 referencia
        public void CalculateArea()...

        1 referencia
        public void WriteData(TextBox txtArea)...
    }
}
```

Sector Circular

```
namespace Practica_6
{
    3 referencias
    internal class CircleSector
    {
        private float mRadius;
        private float mGrade;
        private float mArea;
        private const float pi = 3.1416F;

        1 referencia
        public CircleSector()...

        1 referencia
        public void ReadData(TextBox txtRadius, TextBox txtGrade)...

        1 referencia
        public void CalculateArea()...

        1 referencia
        public void WriteData(TextBox txtArea)...
```

Pruebas y Resultados

Después de haber implementado las clases de cada figura en su respectivo formulario, se ingresaron valores en cada figura para comprobar que realizara los cálculos correctos del área y del perímetro, también se ve que los datos se muestran de manera correcta en los textbox correspondientes

Cuadrado

Menu

Figuras

Cuadrado

Entradas

Lado 4

Procesos

Calcular Resetear Regresar

Salidas

Perimetro 16

Area 16

Triangulo

Menu

Figuras

Triangulo

Entradas

Lado 1 4

Lado 2 1

Lado 3 2

Altura 4

Procesos

Calcular Resetear Regresar

Salidas

Perimetro 7

Area 2

Rectángulo

Menu

Figuras

Rectangulo

Entradas

Ancho3

Altura4

Procesos

Calcular

Resetear

Regresar

Salidas

Perimetro14

Area12

Paralelogramo

Menu

Figuras

Paralelogramo

Entradas

Base2

Lado4

Altura9

Procesos

Calcular

Resetear

Regresar

Salidas

Perimetro18

Area12

Rombo

Menu

Figuras

Rombo

Entradas

Diagonal Superior.

3

Diagonal Inferior

2

Lado

1

Procesos

Calcular

Resetear

Regresar

Salidas

Perimetro

4

Area

3

Cometa

Menu

Figuras

Cometa

Entradas

Diagonal Superior 3

Diagonal Inferior 2

Lado Menor 2

Lado Mayor 4

Procesos

Calcular Resetear Regresar

Salidas

Perimetro 12

Area 3

Trapezio

Menu

Figuras

Trapezio

Entradas

Base Mayor

3

Base Menor

1

Cateto 1

2

Cateto 2

2

Altura

5

Procesos

Calcular

Resetear

Regresar

Salidas

Perimetro

8

Area

10

Circulo

Menu

Figuras

Circulo

Entradas

Radio

2

Procesos

Calcular

Resetear

Regresar

Salidas

Perimetro

12.5664

Area

12.5664

Polígono Regular

Menu

Figuras

Poligono

Entradas

Apotema4

Lado2

Número de Lados3

Procesos

Calcular

Resetear

Regresar

Salidas

Perimetro6

Area12

Corona Circular

Menu

Figuras

Corona Circular

Entradas

Radio Mayor4

Radio Menor2

Procesos

Calcular

Resetear

Regresar

Salidas

Area37.6992

Sector Circular

Menu

Figuras

Sector Circular

Entradas

Radio

2

Grados

30

Procesos

Calcular

Resetear

Regresar

Salidas

Area

1,0472

Conclusiones

- La aplicación desarrollada en C# demostró ser efectiva para realizar cálculos precisos de áreas y perímetros de diversas figuras geométricas, facilitando el aprendizaje y la resolución de problemas matemáticos.
- El uso de formularios de Windows Forms permitió crear una interfaz interactiva y fácil de usar, adecuada tanto para usuarios principiantes como para avanzados. La separación de formularios por figura geométrica simplificó la experiencia de navegación.
- La implementación del patrón Singleton garantizó que solo se abriera una instancia de cada formulario, mejorando el manejo de los recursos y evitando redundancias.

Recomendaciones

- Ampliar la aplicación para incluir cálculos de figuras más complejas como polígonos irregulares, elipses o incluso figuras tridimensionales como prismas y cilindros.
- Asegurarse de que los datos ingresados por los usuarios sean válidos mediante validaciones avanzadas, como restricciones para evitar conflictos con los cálculos, especialmente en figuras como triángulos.
- Manejar la lógica de programación de mejor manera para que a la hora de refactorizarlo no nos de tantos problemas.

Bibliografía

Microsoft. (n.d.). *Windows Forms Overview*. Microsoft Learn. Recuperado el 11 de noviembre de 2024, de <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/>