

Tecnológico Nacional de México

Instituto Tecnológico de Tijuana

Departamento de Sistemas y Computación

Ingeniería en Sistemas Computacionales



Tópicos Avanzados de Programación

Figuras geométricas

Dr. Daniel Eduardo Hernández Morales

Marco Antonio Velázquez Figueroa

Francisco Javier Hernández Ornelas

# Introducción

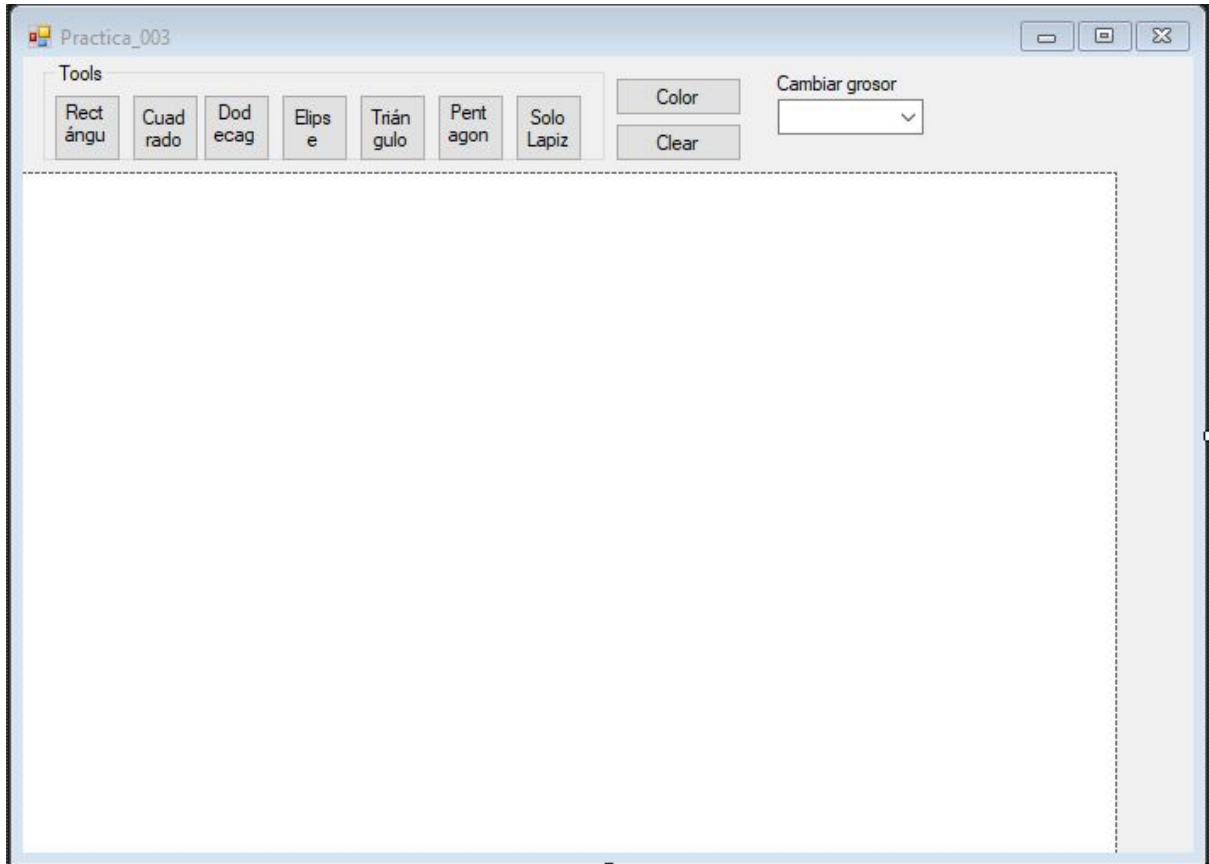
Para la práctica 3 se pidió realizar un “MiniPaint”, que sería una “réplica” de la aplicación de Paint que viene por defecto en los sistemas operativos windows. Esta réplica no es igual que la original, es un poco más ligera que esta, y solo con algunas funciones básicas de la aplicación como poner algunas figuras, o dibujar a mano alzada con un lápiz de diferentes grosores. Para poder realizar esto se utilizaron algunos botones dentro de un groupbox, un label, un combobox y un picturebox. También se utilizó la clase “Graphics” para poder realizar los dibujos en el picturebox, y el control complejo de ColorSelectDialog para cambiar el color con el que se dibuja.

# Marco Teórico

**GUI:** La Interfaz Gráfica de Usuario (GUI, por sus siglas en inglés que significan Graphical User Interface) consiste en la aplicación de una forma en la cual un usuario puede interactuar con el Ordenador, realizando distintas funcionalidades en forma intuitiva y dinámica.

# Desarrollo

Primero diseñamos una interfaz con el editor de forms, buscando que fuera amigable y sencilla para cualquier usuario, llegando al siguiente resultado:



Luego se procedió con el código para hacer que la interfaz funcionara correctamente, que cada botón hiciera lo que se supone que debe hacer, etc. Para esto se utilizaron las siguientes funciones:

```
45 private void drawArea_MouseClick(object sender, MouseEventArgs e)...
```

Esta función se utiliza para escribir en el picturebox cada el que el evento mouseclick sucede en el mismo, pudiendo así escribir en el picturebox cada que el mouse es presionado, pues dentro están definidas las diferentes “herramientas de dibujo” que el usuario puede utilizar.

```
127 private void clearButton_Click(object sender, EventArgs e)...
```

Este es el evento “click” del botón “Clear”, que se encarga de limpiar el picturebox cuando el evento sucede.

```
135 private void ellipseButton_Click(object sender, EventArgs e)...
```

El evento click del botón ellipseButton se encarga de seleccionar la herramienta diseñada para dibujar un círculo, que está definida en la función “drawArea\_MouseClick”.

```
143      +      private void rectButton_Click(object sender, EventArgs e)...
```

El evento click del botón recButton se encarga de seleccionar la herramienta diseñada para dibujar un rectángulo, que está definida en la función “drawArea\_MouseClick”.

```
156      +      private void drawArea_MouseMove_1(object sender, MouseEventArgs e)...
```

El evento “MouseMove\_1” se encarga de dibujar una línea en el picturebox por donde se vaya moviendo el mouse.

```
171      +      private void drawArea_MouseDown(object sender, MouseEventArgs e)...
```

El evento “MouseDown” cambia el valor de la variable “isMousePressed” a “True”, para avisar que el mouse a sido presionado en el área de dibujo.

```
176      +      private void drawArea_MouseUp(object sender, MouseEventArgs e)...
```

El evento “MouseUp” cambia el valor de la variable “isMousePressed” a “False”, para avisar que el mouse ya no se está presionando en el área de dibujo.

```
182      +      private void button4_Click(object sender, EventArgs e)...
```

El evento click del botón “button4” se encarga de seleccionar la herramienta diseñada para dibujar un triángulo, que está definida en la función “drawArea\_MouseClick”.

```
188      +      private void button3_Click(object sender, EventArgs e)...
```

El evento click del botón “button3” se encarga de seleccionar la herramienta diseñada para dibujar un cuadrado, que está definida en la función “drawArea\_MouseClick”.

```
208      +      private void SoloLapizBtt_Click(object sender, EventArgs e)...
```

El evento click del botón “SoloLapizBtt” se encarga de seleccionar la herramienta diseñada para dibujar a mano alzada, que está definida en la función “drawArea\_MouseClick”.

```
219      +      private void ColorBtt_Click(object sender, EventArgs e)...
```

El evento click del botón “ColorBtt” se abre una ventana donde puedes elegir un color para dibujar en el área de dibujo, esto logra mediante el control complejo de ColorSelectDialog.

```
227 private void GrosorCmbx_SelectedIndexChanged(object sender, EventArgs e)...
```

Cuando ocurre el evento “SelectedIndexChanged” del combobox “GrosorCmbx”, se toma el valor elegido en el mismo, y se guarda en la variable “grosor” para modificar el grosor del “pen”.

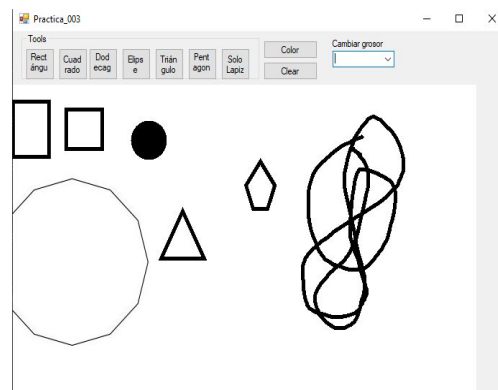
```
237 private void DodecagonoBtt_Click(object sender, EventArgs e)...
```

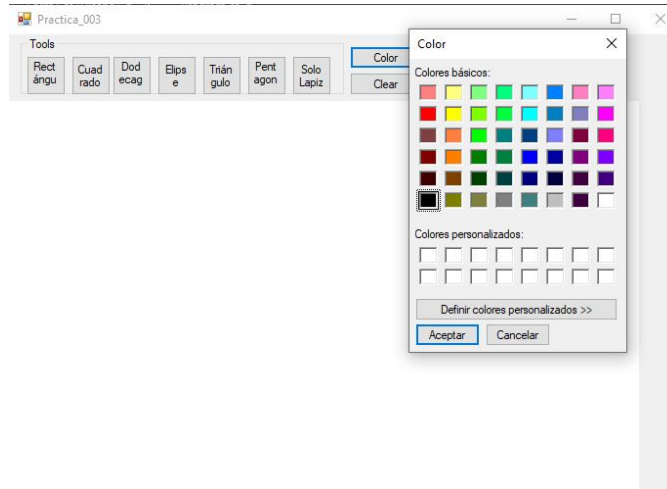
El evento click del botón “DodecagonoBtt” se encarga de seleccionar la herramienta diseñada para dibujar un dodecagono, que está definida en la función “drawArea\_MouseClick”.

```
243 private void PentagonoBtt_Click(object sender, EventArgs e)...
```

El evento click del botón “PentagonoBtt” se encarga de seleccionar la herramienta diseñada para dibujar un pentagono, que está definida en la función “drawArea\_MouseClick”.

## Resultados





# Conclusión

La práctica fue más fácil de lo que esperábamos, logramos realizar todos los requisitos pedidos sin tener algún problema que nos retrasara por mucho tiempo. El ejemplo proporcionado ayudó demasiado a la realización de la práctica, basándose en ese ejemplo fue realmente sencillo lograrlo.

En esta práctica un poco más compleja pudimos notar que las cosas pueden llegar a ponerse realmente complejas al tener tantos eventos, variables, funciones. etc, y la importancia de tener todo organizado de una buena manera al ponerle un nombre adecuado a cada boton, label, combobox, etc, para no llegar a confundirse, y realizar el trabajo de una manera mucho más eficiente.