|  |
| --- |
| INSTITUTO TECNOLÓGICO DE ESTUDIOS SUPERIORES DE LA REGIÓN CARBONÍFERA |
| Manual de Prácticas de Graficación |
| Por: Ing. Héctor Javier Padilla Lara |
| Ejercicios propuestos para la materia de Graficación utilizando Windows Presentation Foundation (WPF) y C#.  **Coordinación de Sistemas Computacionales**  Junio de 2010 |
|  |
|  |

|  |
| --- |
|  |

Tabla de contenido

[Unidad 1. 4](#_Toc295494675)

[Ejercicio 1. StackPanel 4](#_Toc295494676)

[Respuesta en XAML: 4](#_Toc295494677)

[Ejercicio 2. Canvas 5](#_Toc295494678)

[Respuesta en XAML: 5](#_Toc295494679)

[Ejercicio 3. Grid 6](#_Toc295494680)

[Respuesta en XAML: 6](#_Toc295494681)

[Ejercicio 4. UniformGrid 8](#_Toc295494682)

[Respuesta en XAML: 8](#_Toc295494683)

[Ejercicio 5. Otros Controles 10](#_Toc295494684)

[Respuesta en XAML: 10](#_Toc295494685)

[Unidad 2. 11](#_Toc295494686)

[Ejercicio 1. Casita 11](#_Toc295494687)

[Respuesta en XAML: 11](#_Toc295494688)

[Ejercicio 2. Bebe 12](#_Toc295494689)

[Respuesta en XAML 12](#_Toc295494690)

[Ejercicio 3. Ying y Yang 13](#_Toc295494691)

[Respuesta en XAML: 13](#_Toc295494692)

[Ejercicio 4. Pacman 14](#_Toc295494693)

[Respuesta en XAML: 14](#_Toc295494694)

[Ejercicio 5. Disquette 15](#_Toc295494695)

[Respuesta en XAML: 15](#_Toc295494696)

[Unidad 3. 16](#_Toc295494697)

[Ejercicio 1. Rehiletes 16](#_Toc295494698)

[Respuesta en XAML: 16](#_Toc295494699)

[Ejercicio 2. Pájaro bebedor 19](#_Toc295494700)

[Respuesta en XAML: 19](#_Toc295494701)

[Ejercicio 3. Torre de Transmisión 21](#_Toc295494702)

[Respuesta en XAML: 21](#_Toc295494703)

[Unidad 4. 24](#_Toc295494704)

[Ejercicio 1. Ahorcado 24](#_Toc295494705)

[Respuesta en XAML: 24](#_Toc295494706)

[Código en C#: 25](#_Toc295494707)

[Ejercicio 2. Termómetro 28](#_Toc295494708)

[Respuesta en XAML: 28](#_Toc295494709)

[Código en C#: 30](#_Toc295494710)

[Ejercicio 3. Notas en WPF 32](#_Toc295494711)

[Respuesta en XAML: 32](#_Toc295494712)

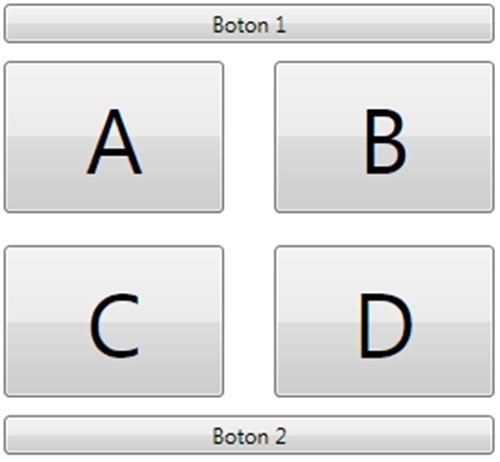
[Código en WPF: 33](#_Toc295494713)

# Unidad 1.

En esta unidad el alumno aprenderá las bases de WPF, el uso de controles y el maquetado (layout).

Realizar los ejercicios después de explicar cada uno de los controles de maquetación y los controles básicos de WPF.

## Ejercicio 1. StackPanel



## Respuesta en XAML:

<StackPanel Margin="3">

<Button>Boton 1</Button>

<UniformGrid Height="200">

<Button Margin="0,5,12,10" FontSize="30">A</Button>

<Button Margin="12,5,0,10" FontSize="30">B</Button>

<Button Margin="0,10,12,5" FontSize="30">C</Button>

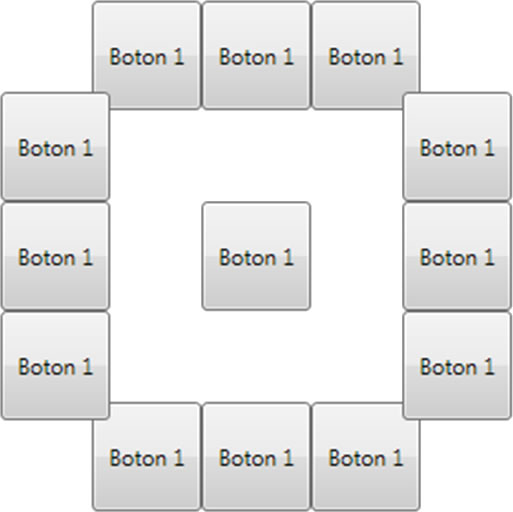
<Button Margin="12,10,0,5" FontSize="30">D</Button>

</UniformGrid>

<Button>Boton 2</Button>

</StackPanel>

## Ejercicio 2. Canvas



## Respuesta en XAML:

<Canvas>

<StackPanel Canvas.Top="0" Orientation="Horizontal" Canvas.Left="40" >

<Button Width="50" Height="50">Boton 1</Button>

<Button Width="50">Boton 1</Button>

<Button Width="50">Boton 1</Button>

</StackPanel>

<StackPanel Canvas.ZIndex="10" Canvas.Top="40" Orientation="Vertical" >

<Button Height="50" Width="50">Boton 1</Button>

<Button Height="50" Width="50">Boton 1</Button>

<Button Height="50">Boton 1</Button>

</StackPanel>

<StackPanel Canvas.ZIndex="10" Orientation="Vertical" Canvas.Left="180" Canvas.Top="40">

<Button Height="50">Boton 1</Button>

<Button Height="50" Width="50">Boton 1</Button>

<Button Height="50">Boton 1</Button>

</StackPanel>

<StackPanel Canvas.Top="180" Canvas.Left="40" Orientation="Horizontal" >

<Button Height="50" Width="50">Boton 1</Button>

<Button Width="50">Boton 1</Button>

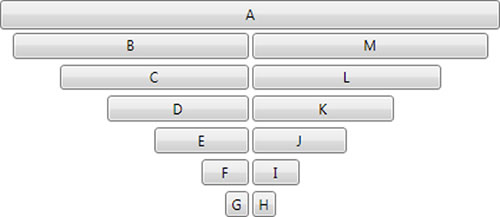
<Button Width="50">Boton 1</Button>

</StackPanel>

<Button Width="50" Height="50" Canvas.Top="90" Canvas.Left="90">Boton1</Button>

</Canvas>

## Ejercicio 3. Grid



## Respuesta en XAML:

<Viewbox Stretch="Uniform">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition/><RowDefinition/>

<RowDefinition/><RowDefinition/>

<RowDefinition/><RowDefinition/><RowDefinition/>

</Grid.RowDefinitions>

<Button Grid.ColumnSpan="2" Width="490">A</Button>

<Button Grid.Row="1" Width="223" HorizontalAlignment="Right" Margin="0,2,1,0">B</Button>

<Button Grid.Row="2" Width="203" HorizontalAlignment="Right" Margin="0,2,1,0">C</Button>

<Button Grid.Row="3" Width="163" HorizontalAlignment="Right" Margin="0,2,1,0">D</Button>

<Button Grid.Row="4" Width="113" HorizontalAlignment="Right" Margin="0,2,1,0">E</Button>

<Button Grid.Row="5" Width="53" HorizontalAlignment="Right" Margin="0,2,1,0">F</Button>

<Button Grid.Row="6" Width="23" HorizontalAlignment="Right" Margin="0,2,1,0">G</Button>

<Button Grid.Column="1" Grid.Row="1" Width="223" HorizontalAlignment="Left" Margin="1,2,0,0">M</Button>

<Button Grid.Column="1" Grid.Row="2" Width="203" HorizontalAlignment="Left" Margin="1,2,0,0">L</Button>

<Button Grid.Column="1" Grid.Row="3" Width="163" HorizontalAlignment="Left" Margin="1,2,0,0">K</Button>

<Button Grid.Column="1" Grid.Row="4" Width="113" HorizontalAlignment="Left" Margin="1,2,0,0">J</Button>

<Button Grid.Column="1" Grid.Row="5" Width="53" HorizontalAlignment="Left" Margin="1,2,0,0">I</Button>

<Button Grid.Column="1" Grid.Row="6" Width="23" HorizontalAlignment="Left" Margin="1,2,0,0">H</Button>

</Grid>

</Viewbox>

## Ejercicio 4. UniformGrid



## Respuesta en XAML:

<UniformGrid Columns="3" Rows="4" Margin="5">

<Button FontSize="35">1</Button>

<Button>

<TextBlock FontSize="35">2<TextBlock FontSize="18" FontStyle="Italic">abc</TextBlock></TextBlock></Button>

<Button>

<TextBlock FontSize="35">3<TextBlock FontSize="18" FontStyle="Italic">def</TextBlock></TextBlock>

</Button>

<Button>

<TextBlock FontSize="35">4<TextBlock FontSize="18" FontStyle="Italic">ghi</TextBlock></TextBlock>

</Button>

<Button>

<TextBlock FontSize="35">5<TextBlock FontSize="18" FontStyle="Italic">jkl</TextBlock></TextBlock>

</Button>

<Button>

<TextBlock FontSize="35">6<TextBlock FontSize="18" FontStyle="Italic">mno</TextBlock></TextBlock>

</Button>

<Button>

<TextBlock FontSize="35">7<TextBlock FontSize="18" FontStyle="Italic">pqrs</TextBlock></TextBlock>

</Button>

<Button>

<TextBlock FontSize="35">8<TextBlock FontSize="18" FontStyle="Italic">tuv</TextBlock></TextBlock>

</Button>

<Button>

<TextBlock FontSize="35">9<TextBlock FontSize="18" FontStyle="Italic">wxyz</TextBlock></TextBlock>

</Button>

<Button>

<TextBlock FontSize="45">\*</TextBlock>

</Button>

<Button>

<TextBlock FontSize="35">0<TextBlock FontSize="18" FontStyle="Italic">\_</TextBlock></TextBlock>

</Button>

<Button>

<TextBlock FontSize="35">#</TextBlock>

</Button>

</UniformGrid>

## Ejercicio 5. Otros Controles



## Respuesta en XAML:

<ComboBox Height="20">

<ComboBoxItem FontSize="25" FontFamily="Arial">Arial</ComboBoxItem>

<ComboBoxItem FontSize="25" FontFamily="Courier New">Courier New</ComboBoxItem>

<ComboBoxItem FontSize="25" FontFamily="Comic Sans MS">Comic Sans MS</ComboBoxItem>

<ComboBoxItem FontSize="25" FontFamily="Monotype Corsiva">Monotype Corsiva</ComboBoxItem>

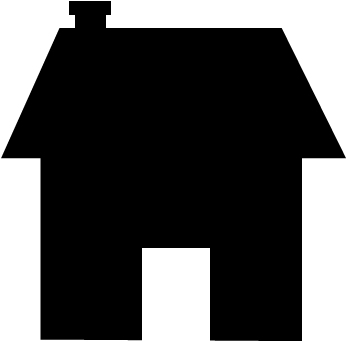
<ComboBoxItem FontSize="25" FontFamily="Times New Roman">Times New Roman</ComboBoxItem>

</ComboBox>

# Unidad 2.

El alumno aprenderá a trabajar con formas básicas (Líneas, Elipses, Rectángulos, Polígonos) y con rutas (Paths) para graficar y crear controles.

## Ejercicio 1. Casita



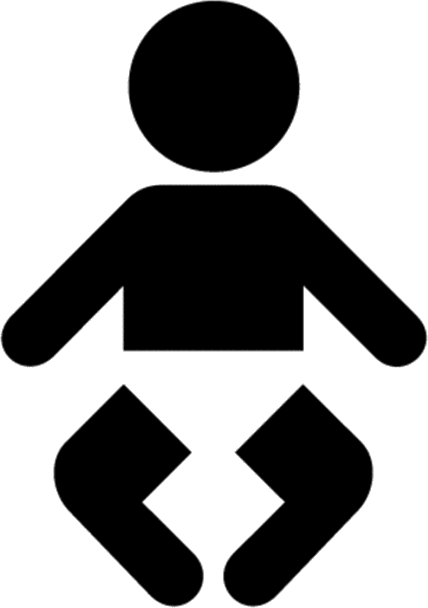
## Respuesta en XAML:

<Canvas>

<Path Canvas.Top="23" Canvas.Left="20" Stroke="Black" Fill="Black" Data="M0,100 L40,15 L200,15 L240,100 L210,100 L210,220 L30,220 L30,100 Z M100,220 L100,160 L140,160 L140,220 Z M50,0 L80,0 L80,5 L75,5 L75,15 L55,15 L55,5 L50,5" ></Path>

</Canvas>

## Ejercicio 2. Bebe



## Respuesta en XAML

<Canvas Margin="10">

<Ellipse Fill="Black" Canvas.Left="60" Stroke="Black" Height="82" Width="82"></Ellipse>

<Path Stroke="Black" Fill="Black" Data="M74,87 L129,87 Q140,90 145,97 L199,151 A10,10 0 1 1 179,171 L144,136 L144,167 L58,167 L58,136 L23,172 A10,10 0 1 1 3,152 L57,97 Q63,90 74,87 "></Path>

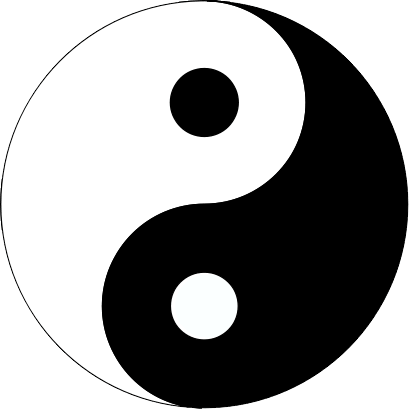
<Path Stroke="Black" Fill="Black" Data="M58,182 L91,215 L68,238 L95,266 A5,5 0 0 1 75,286 L30,240 Q21,224 30,210 Z"></Path>

<Path Stroke="Black" Fill="Black" Data="M144,182 L111,215 L134,238 L108,266 A5,5 0 0 0 128,286 L172,240 Q181,224 172,210 Z">

</Path>

</Canvas>

## Ejercicio 3. Ying y Yang



## Respuesta en XAML:

<Canvas>

<Path Stroke="Black" Fill="White" Data="M180,0 A173.5,173.5 0 0 0 180,347 A90,85 0 0 1 173.5,173.5 A90,85 0 0 0 180,0"></Path>

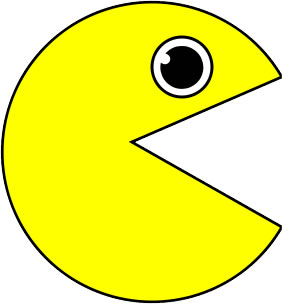
<Path Stroke="Black" Fill="Black" Data="M180,0 A173.5,173.5 0 0 1 180,347 A90,85 0 0 1 173.5,173.5 A90,85 0 0 0 180,0"></Path>

<Ellipse Canvas.Top="227" Canvas.Left="150" Width="60" Height="60" Fill="White"></Ellipse>

<Ellipse Canvas.Top="60" Canvas.Left="150" Width="60" Height="60" Fill="Black"></Ellipse>

</Canvas>

## Ejercicio 4. Pacman



## Respuesta en XAML:

<Grid>

<Path Stroke="Black" Fill="Yellow" Data="M196,53 L93,99 L196,159 A100,100 0 1 1 196,53"></Path>

<Ellipse Stroke="Black" Fill="White" Height="43" Width="43" HorizontalAlignment="Left" Margin="104,24,0,0" VerticalAlignment="Top"></Ellipse>

<Path Stroke="Black" Fill="Black" Data="M115,37 A14,14 0 1 1 112,44 M115,37 A3,3 0 0 1 112,44"></Path>

</Grid>

## Ejercicio 5. Disquette



## Respuesta en XAML:

<Canvas Margin="10">

<Path Stroke="Black" Fill="Black" Data="M0,0 H176 L192,16 V192,192 H0 M171,167 H182 V178 H171 Z"></Path>

<Rectangle Width="108" Height="70" Fill="Gray" Canvas.Left="42"></Rectangle>

<Rectangle Fill="Black" Width="24" Height="45" RadiusX="5" RadiusY="5" Canvas.Left="109" Canvas.Top="17"></Rectangle>

<Rectangle Fill="White" Width="142" Height="84" Canvas.Left="25" Canvas.Top="84"></Rectangle>

<Rectangle Fill="Red" Width="142" Height="16" Canvas.Left="25" Canvas.Top="167"></Rectangle>

<Line Opacity=".1" Stroke="Black" X1="30" Y1="108" X2="162" Y2="108"></Line>

<Line Opacity=".1" Stroke="Black" X1="30" Y1="124" X2="162" Y2="124"></Line>

<Line Opacity=".1" Stroke="Black" X1="30" Y1="140" X2="162" Y2="140"></Line>

<Line Opacity=".1" Stroke="Black" X1="30" Y1="156" X2="162" Y2="156"></Line>

<Line Opacity=".1" Stroke="Black" X1="30" Y1="92" X2="162" Y2="92"></Line>

<Line X1="43" Y1="103" X2="145" Y2="103" Stroke="Black" StrokeDashArray="1" StrokeThickness="2" ></Line>

<TextBlock Foreground="Red" FontWeight="Bold" FontSize="16" Height="27" FontStyle="Italic" Canvas.Left="51" Canvas.Top="105" Width="115">Informacion</TextBlock>

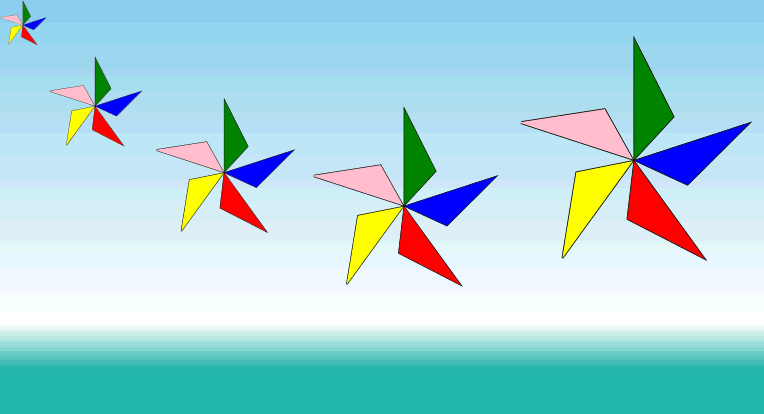
<TextBlock Foreground="Navy" FontWeight="Bold" FontSize="16" Height="27" Canvas.Left="46" Canvas.Top="123" Width="105">Confidencial</TextBlock>

</Canvas>

# Unidad 3.

El alumno aprenderá animaciones básicas y el uso de efectos y transformaciones geométricas.

# Ejercicio 1. Rehiletes



## Respuesta en XAML:

<Window.Background>

<LinearGradientBrush StartPoint="0,0" EndPoint="0,1">

<GradientStop Color="SkyBlue" Offset="0"></GradientStop>

<GradientStop Color="White" Offset=".75"></GradientStop>

<GradientStop Color="DarkTurquoise" Offset="1"></GradientStop>

</LinearGradientBrush>

</Window.Background>

<Canvas>

<Canvas.Resources>

<VisualBrush x:Key="fondo">

<VisualBrush.Visual>

<Canvas>

<Polygon Fill="Green" Stroke="Black" Points="0,0 50,93 0,141 "></Polygon>

<Polygon Fill="Blue" Stroke="Black" Points="0,0 50,93 0,141 ">

<Polygon.RenderTransform>

<RotateTransform CenterX="0" CenterY="141" Angle="70"/>

</Polygon.RenderTransform>

</Polygon>

<Polygon Fill="Red" Stroke="Black" Points="0,0 50,93 0,141 ">

<Polygon.RenderTransform>

<RotateTransform CenterX="0" CenterY="141" Angle="140"/>

</Polygon.RenderTransform>

</Polygon>

<Polygon Fill="Yellow" Stroke="Black" Points="0,0 50,93 0,141 ">

<Polygon.RenderTransform>

<RotateTransform CenterX="0" CenterY="141" Angle="205"/>

</Polygon.RenderTransform>

</Polygon>

<Polygon Fill="Pink" Stroke="Black" Points="0,0 50,93 0,141 ">

<Polygon.RenderTransform>

<RotateTransform CenterX="0" CenterY="141" Angle="280"/>

</Polygon.RenderTransform>

</Polygon>

</Canvas>

</VisualBrush.Visual >

</VisualBrush>

</Canvas.Resources>

<Rectangle Width="40" Height="40" Fill="{StaticResource fondo}" Canvas.Top="1" Canvas.Left="1" >

<Rectangle.RenderTransform>

<RotateTransform CenterX="20" CenterY="20" x:Name="gira1"/>

</Rectangle.RenderTransform>

<Rectangle.Triggers>

<EventTrigger RoutedEvent="Rectangle.Loaded">

<BeginStoryboard>

<Storyboard RepeatBehavior="Forever">

<DoubleAnimation From="0" To="360" Duration="0:0:2" Storyboard.TargetName="gira1" Storyboard.TargetProperty="Angle"/>

<DoubleAnimation From="0" To="360" Duration="0:0:2" Storyboard.TargetName="gira2" Storyboard.TargetProperty="Angle"/>

<DoubleAnimation From="0" To="360" Duration="0:0:2" Storyboard.TargetName="gira3" Storyboard.TargetProperty="Angle"/>

<DoubleAnimation From="0" To="360" Duration="0:0:2" Storyboard.TargetName="gira4" Storyboard.TargetProperty="Angle"/>

<DoubleAnimation From="0" To="360" Duration="0:0:2" Storyboard.TargetName="gira5" Storyboard.TargetProperty="Angle"/>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

</Rectangle.Triggers>

</Rectangle>

<Rectangle Canvas.Left="70" Width="75" Height="75" Fill="{StaticResource fondo}" Canvas.Top="45" >

<Rectangle.RenderTransform>

<RotateTransform CenterX="37.5" CenterY="37.5" x:Name="gira2"/>

</Rectangle.RenderTransform>

</Rectangle>

<Rectangle Canvas.Left="182" Width="100" Height="100" Fill="{StaticResource fondo}" Canvas.Top="92" >

<Rectangle.RenderTransform>

<RotateTransform CenterX="50" CenterY="50" x:Name="gira3"/>

</Rectangle.RenderTransform>

</Rectangle>

<Rectangle Canvas.Left="330" Width="125" Height="125" Fill="{StaticResource fondo}" Canvas.Top="105">

<Rectangle.RenderTransform>

<RotateTransform CenterX="62.5" CenterY="62.5" x:Name="gira4"/>

</Rectangle.RenderTransform>

</Rectangle>

<Rectangle Canvas.Left="516" Width="175" Height="175" Fill="{StaticResource fondo}" Canvas.Top="22">

<Rectangle.RenderTransform>

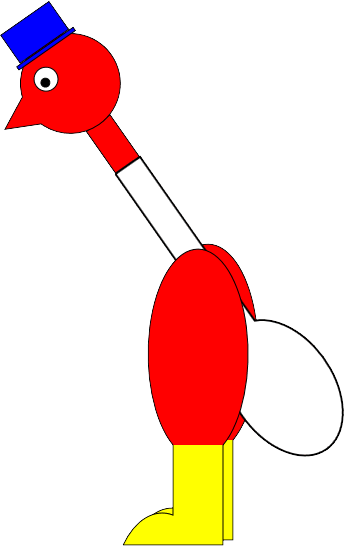
<RotateTransform CenterX="87.5" CenterY="87.5" x:Name="gira5"/>

</Rectangle.RenderTransform>

</Rectangle>

</Canvas>

## Ejercicio 2. Pájaro bebedor



## Respuesta en XAML:

<Canvas Margin="80 0 0 0">

<Ellipse Height="125" Width="60" Stroke="Black" Fill="Red" Canvas.Left="94" Canvas.Top="148" />

<Path Canvas.Left="80" Canvas.Bottom="2" Stroke="Black" Fill="Yellow" Data="M0,327 H59 V267 H29 V310 Q15,300 0,327"/>

<Canvas Canvas.Left="78" Width="75" Height="320">

<Rectangle Width="17" Height="32" Stroke="Black" Fill="Red" Canvas.Left="35" Canvas.Top="79" />

<Path Stroke="Black" Fill="White" Data="M52,230 C77,248 77,300 52,315 A12,5 0 0 1 35,315 C7,300 7,248 35,230 V110 H52 V230 "/>

<Path Stroke="Black" Fill="Red" Canvas.Top="20">

<Path.Data>

<CombinedGeometry >

<CombinedGeometry.Geometry1>

<EllipseGeometry Center="45,30" RadiusX="30" RadiusY="30" />

</CombinedGeometry.Geometry1>

<CombinedGeometry.Geometry2>

<PathGeometry Figures="M0,30 L20,20 L20,40" />

</CombinedGeometry.Geometry2>

</CombinedGeometry>

</Path.Data>

</Path>

<Ellipse Stroke="Black" Fill="White" Width="15" Height="15" Canvas.Left="22" Canvas.Top="33" />

<Ellipse Fill="Black" Width="5" Height="5" Canvas.Left="25" Canvas.Top="39" />

<Rectangle Stroke="Black" Fill="Blue" Width="36" Height="20" Canvas.Left="27" Canvas.Top="2" />

<Rectangle Stroke="Black" Fill="Blue" Width="42" Height="5" Canvas.Left="23.5" Canvas.Top="21" />

<Canvas.RenderTransform>

<RotateTransform CenterX="37.5" CenterY="180" Angle="0" x:Name="girar"/>

</Canvas.RenderTransform>

<Canvas.Triggers>

<EventTrigger RoutedEvent="Rectangle.Loaded">

<BeginStoryboard>

<Storyboard RepeatBehavior="Forever">

<DoubleAnimation From="0" To="-90" Storyboard.TargetProperty="Angle" Storyboard.TargetName="girar" Duration="00:00:02"></DoubleAnimation>

<DoubleAnimation From="-90" To="25" Storyboard.TargetProperty="Angle" Storyboard.TargetName="girar" BeginTime="00:00:02" Duration="00:00:02"></DoubleAnimation>

<DoubleAnimation AutoReverse="True" RepeatBehavior="2x" From="25" To="-20" Storyboard.TargetProperty="Angle" Storyboard.TargetName="girar" BeginTime="00:00:04" Duration="00:00:01"></DoubleAnimation>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

</Canvas.Triggers>

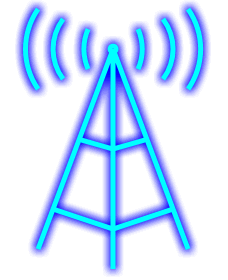
</Canvas>

<Ellipse Height="125" Width="60" Stroke="Black" Fill="Red" Canvas.Left="88" Canvas.Top="150" />

<Path Canvas.Left="74" Stroke="Black" Fill="Yellow" Data="M0,327 H59 V267 H29 V310 Q15,300 0,327"/>

</Canvas>

## Ejercicio 3. Torre de Transmisión



## Respuesta en XAML:

<Window.Resources>

<Style TargetType="Path">

<Setter Property="Path.BitmapEffect">

<Setter.Value>

<OuterGlowBitmapEffect GlowColor="Blue" GlowSize="5"></OuterGlowBitmapEffect>

</Setter.Value>

</Setter>

</Style>

</Window.Resources>

<Canvas>

<Path Stroke="Aqua" StrokeThickness="5" Data="M37,249 L111,53 A3,3 0 1 1 114,53 L188,249 M112.5,53 L112.5,270 M79,138 L112,148 L146,138 M53,208 L112,229 L172,208 "/>

<Path Stroke="Aqua" StrokeThickness="5" Data="M88,29 Q80,47 88,69 " Opacity="0">

<Path.Triggers>

<EventTrigger RoutedEvent="Path.Loaded">

<BeginStoryboard>

<Storyboard RepeatBehavior="Forever">

<DoubleAnimation From="1" To="1" Duration="0:0:1" Storyboard.TargetProperty="Opacity"/>

<DoubleAnimation From="1" To="0" BeginTime="0:0:1" Duration="0:0:1" Storyboard.TargetProperty="Opacity"/>

<DoubleAnimation From="1" To="1" BeginTime="0:0:1" Duration="0:0:1" Storyboard.TargetName="L2" Storyboard.TargetProperty="Opacity"/>

<DoubleAnimation From="1" To="0" BeginTime="0:0:2" Duration="0:0:1" Storyboard.TargetName="L2" Storyboard.TargetProperty="Opacity"/>

<DoubleAnimation From="1" To="1" BeginTime="0:0:2" Duration="0:0:1" Storyboard.TargetName="L3" Storyboard.TargetProperty="Opacity"/>

<DoubleAnimation From="1" To="0" BeginTime="0:0:3" Duration="0:0:1" Storyboard.TargetName="L3" Storyboard.TargetProperty="Opacity"/>

<DoubleAnimation From="1" To="1" Duration="0:0:1" Storyboard.TargetName="L12" Storyboard.TargetProperty="Opacity"/>

<DoubleAnimation From="1" To="0" BeginTime="0:0:1" Duration="0:0:1" Storyboard.TargetName="L12" Storyboard.TargetProperty="Opacity"/>

<DoubleAnimation From="1" To="1" BeginTime="0:0:1" Duration="0:0:1" Storyboard.TargetName="L22" Storyboard.TargetProperty="Opacity"/>

<DoubleAnimation From="1" To="0" BeginTime="0:0:2" Duration="0:0:1" Storyboard.TargetName="L22" Storyboard.TargetProperty="Opacity"/>

<DoubleAnimation From="1" To="1" BeginTime="0:0:2" Duration="0:0:1" Storyboard.TargetName="L32" Storyboard.TargetProperty="Opacity"/>

<DoubleAnimation From="1" To="0" BeginTime="0:0:3" Duration="0:0:1" Storyboard.TargetName="L32" Storyboard.TargetProperty="Opacity"/>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

</Path.Triggers>

</Path>

<Path Stroke="Aqua" StrokeThickness="5" Data="M63,18 Q50,47 63,79 " Name="L2" Opacity="0"/>

<Path Stroke="Aqua" StrokeThickness="5" Data="M38,9 Q20,47 38,89 " Name="L3" Opacity="0"/>

<Path Stroke="Aqua" StrokeThickness="5" Data="M137,29 Q145,47 137,69 " Name="L12" Opacity="0"/>

<Path Stroke="Aqua" StrokeThickness="5" Data="M162,18 Q175,47 162,79 " Name="L22" Opacity="0"/>

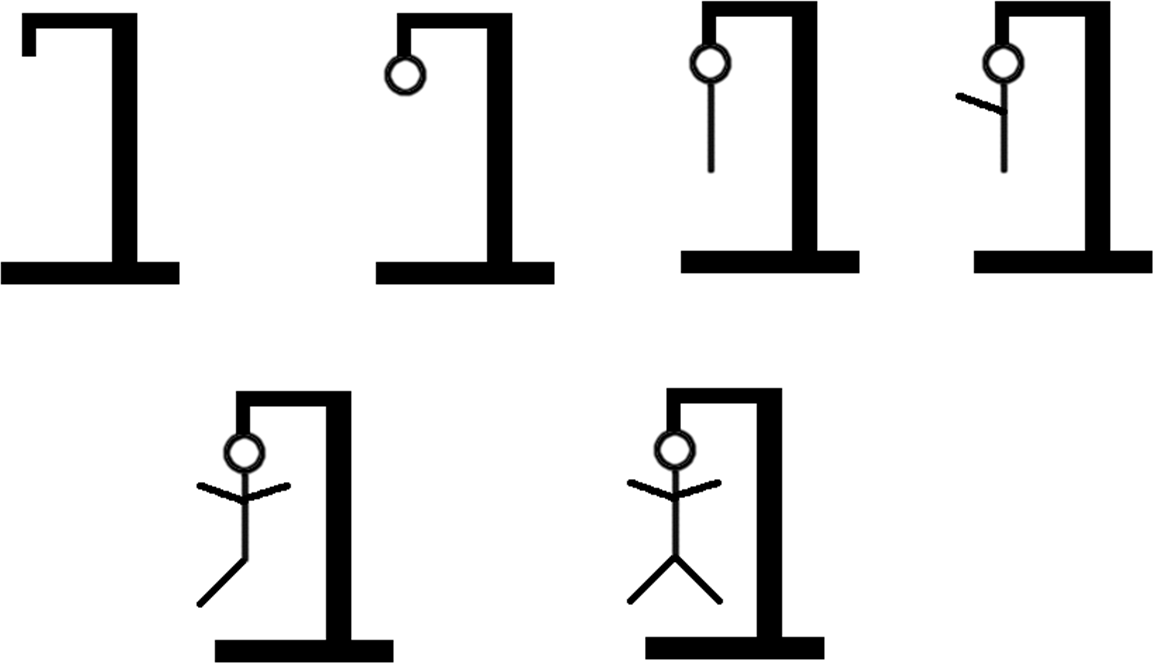
<Path Stroke="Aqua" StrokeThickness="5" Data="M187,9 Q205,47 187,89 " Name="L32" Opacity="0"/>

</Canvas>

# Unidad 4.

El alumno aprenderá a reunir los conocimientos adquiridos anteriormente para formar controles de usuario para poder usar en sus aplicaciones.

# Ejercicio 1. Ahorcado



## Respuesta en XAML:

<UserControl x:Class="Ejercicios\_U4.UC\_Ahorcado"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

mc:Ignorable="d"

d:DesignHeight="300" d:DesignWidth="219">

<Canvas Width="214">

<Path Stroke="Black" Fill="Black" Data="M40,43 V0 H156 V249 H198 V271 H19 V249 H130 V16 H54 V43 Z "/>

<Ellipse Name="cabeza" Width="40" Height="40" Stroke="Black" StrokeThickness="5" Canvas.Left="28" Canvas.Top="42" Visibility="Hidden" />

<Line Name="cuerpo" X1="0" Y1="0" X2="0" Y2="90" StrokeEndLineCap="Round" Stroke="Black" StrokeThickness="5" Canvas.Left="47" Canvas.Top="80" Visibility="Hidden"/>

<Line Name="b1" X1="47" Y1="112" X2="5" Y2="96" StrokeEndLineCap="Round" Stroke="Black" StrokeThickness="5" Visibility="Hidden" />

<Line Name="b2" X1="47" Y1="112" X2="90" Y2="96" StrokeEndLineCap="Round" Stroke="Black" StrokeThickness="5" Visibility="Hidden" />

<Line Name="p1" X1="47" Y1="169" X2="5" Y2="214" StrokeEndLineCap="Round" Stroke="Black" StrokeThickness="5" Visibility="Hidden" />

<Line Name="p2" X1="47" Y1="169" X2="90" Y2="214" Stroke="Black" StrokeThickness="5" StrokeEndLineCap="Round" Visibility="Hidden" />

</Canvas>

</UserControl>

## Código en C#:

public partial class UC\_Ahorcado : UserControl

{

public UC\_Ahorcado()

{

InitializeComponent();

}

private int gradoError;

public int GradoError

{

get { return gradoError; }

set

{

gradoError = value;

switch (gradoError)

{

case 0:

{

cabeza.Visibility = Visibility.Hidden;

cuerpo.Visibility = Visibility.Hidden;

b1.Visibility = Visibility.Hidden;

b2.Visibility = Visibility.Hidden;

p1.Visibility = Visibility.Hidden;

p2.Visibility = Visibility.Hidden;

} break;

case 1:

{

cabeza.Visibility = Visibility.Visible;

cuerpo.Visibility = Visibility.Hidden;

b1.Visibility = Visibility.Hidden;

b2.Visibility = Visibility.Hidden;

p1.Visibility = Visibility.Hidden;

p2.Visibility = Visibility.Hidden;

} break;

case 2:

{

cabeza.Visibility = Visibility.Visible;

cuerpo.Visibility = Visibility.Visible;

b1.Visibility = Visibility.Hidden;

b2.Visibility = Visibility.Hidden;

p1.Visibility = Visibility.Hidden;

p2.Visibility = Visibility.Hidden;

} break;

case 3:

{

cabeza.Visibility = Visibility.Visible;

cuerpo.Visibility = Visibility.Visible;

b1.Visibility = Visibility.Visible;

b2.Visibility = Visibility.Hidden;

p1.Visibility = Visibility.Hidden;

p2.Visibility = Visibility.Hidden;

} break;

case 4:

{

cabeza.Visibility = Visibility.Visible;

cuerpo.Visibility = Visibility.Visible;

b1.Visibility = Visibility.Visible;

b2.Visibility = Visibility.Visible;

p1.Visibility = Visibility.Hidden;

p2.Visibility = Visibility.Hidden;

} break;

case 5:

{

cabeza.Visibility = Visibility.Visible;

cuerpo.Visibility = Visibility.Visible;

b1.Visibility = Visibility.Visible;

b2.Visibility = Visibility.Visible;

p1.Visibility = Visibility.Visible;

p2.Visibility = Visibility.Hidden;

} break;

case 6:

{

cabeza.Visibility = Visibility.Visible;

cuerpo.Visibility = Visibility.Visible;

b1.Visibility = Visibility.Visible;

b2.Visibility = Visibility.Visible;

p1.Visibility = Visibility.Visible;

p2.Visibility = Visibility.Visible;

} break;

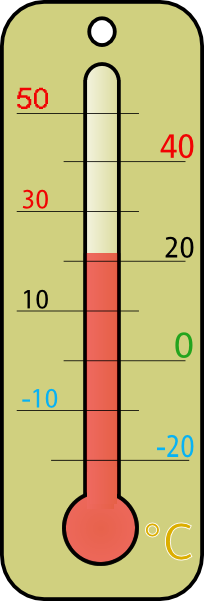
}

}

}

}

## Ejercicio 2. Termómetro



## Respuesta en XAML:

<UserControl x:Class="Ejercicios\_U4.UC\_Termometro"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

mc:Ignorable="d"

d:DesignHeight="300" d:DesignWidth="101">

<Canvas>

<Rectangle Width="100" Height="300" Stroke="Black" Fill="#d0d07f" RadiusX="20" RadiusY="20"/>

<Ellipse Stroke="Black" Fill="White" Width="13" Height="13" Canvas.Left="43" Canvas.Top="8"/>

<Ellipse Fill="Red" Canvas.Left="31" Canvas.Top="232" Width="37" Height="38"/>

<Rectangle Name="liq" Width="17" Height="0" Fill="Red" Canvas.Left="41" Canvas.Bottom="59"/>

<Path Stroke="Black" Fill="White" Canvas.Left="41" Canvas.Top="31" Opacity=".5">

<Path.Data>

<CombinedGeometry>

<CombinedGeometry.Geometry1>

<EllipseGeometry RadiusX="19" RadiusY="19" Center="8.5,220"/>

</CombinedGeometry.Geometry1>

<CombinedGeometry.Geometry2>

<RectangleGeometry Rect="0,0 17,210" RadiusX="10" RadiusY="8"/>

</CombinedGeometry.Geometry2>

</CombinedGeometry>

</Path.Data>

</Path>

<TextBlock Text="°C" Foreground="#dcae00" FontSize="20" FontWeight="Bold" Height="27" Canvas.Left="74" Canvas.Top="259" Width="20" />

<TextBlock Foreground="Red" Canvas.Left="15" Canvas.Top="32" FontSize="16">50</TextBlock>

<TextBlock Foreground="Red" Canvas.Left="70.06" Canvas.Top="55" FontSize="16">40</TextBlock>

<TextBlock Foreground="Red" Canvas.Left="15" Canvas.Top="77" FontSize="16">30</TextBlock>

<TextBlock Canvas.Left="68.06" Canvas.Top="99" FontSize="16">20</TextBlock>

<TextBlock Canvas.Left="15" Canvas.Top="121" FontSize="16">10</TextBlock>

<TextBlock Canvas.Left="78" Canvas.Top="143" FontSize="16">0</TextBlock>

<TextBlock Foreground="#08b2f4" Canvas.Left="10" Canvas.Top="166" FontSize="16">-10</TextBlock>

<TextBlock Foreground="#08b2f4" Canvas.Left="65" Canvas.Top="188" FontSize="16">-20</TextBlock>

<Line X1="0" Y1="0" X2="45" Y2="0" Stroke="Black" Canvas.Left="15" Canvas.Top="53.22" />

<Line X1="0" Y1="0" X2="45" Y2="0" Stroke="Black" Canvas.Left="41" Canvas.Top="75.44" />

<Line X1="0" Y1="0" X2="45" Y2="0" Stroke="Black" Canvas.Left="15" Canvas.Top="97.66"/>

<Line X1="0" Y1="0" X2="45" Y2="0" Stroke="Black" Canvas.Left="41" Canvas.Top="119.88" />

<Line X1="0" Y1="0" X2="45" Y2="0" Stroke="Black" Canvas.Left="15" Canvas.Top="142.1"/>

<Line X1="0" Y1="0" X2="45" Y2="0" Stroke="Black" Canvas.Left="41" Canvas.Top="164.32"/>

<Line X1="0" Y1="0" X2="45" Y2="0" Stroke="Black" Canvas.Left="15" Canvas.Top="186.54"/>

<Line X1="0" Y1="0" X2="45" Y2="0" Stroke="Black" Canvas.Left="41" Canvas.Top="208.76"/>

</Canvas>

</UserControl>

## Código en C#:

public partial class UC\_Termometro : UserControl

{

public UC\_Termometro()

{

InitializeComponent();

}

private int grado;

public int Grado

{

get { return grado; }

set

{

if (value >= -20 && value <= 50)

{

grado = value;

liq.Height = (grado \* 2.22) + 76.66;

}

else

{

if (value < -20)

{

liq.Height = (-20 \* 2.22) + 76.66;

}

else

liq.Height = (50 \* 2.22) + 76.66;

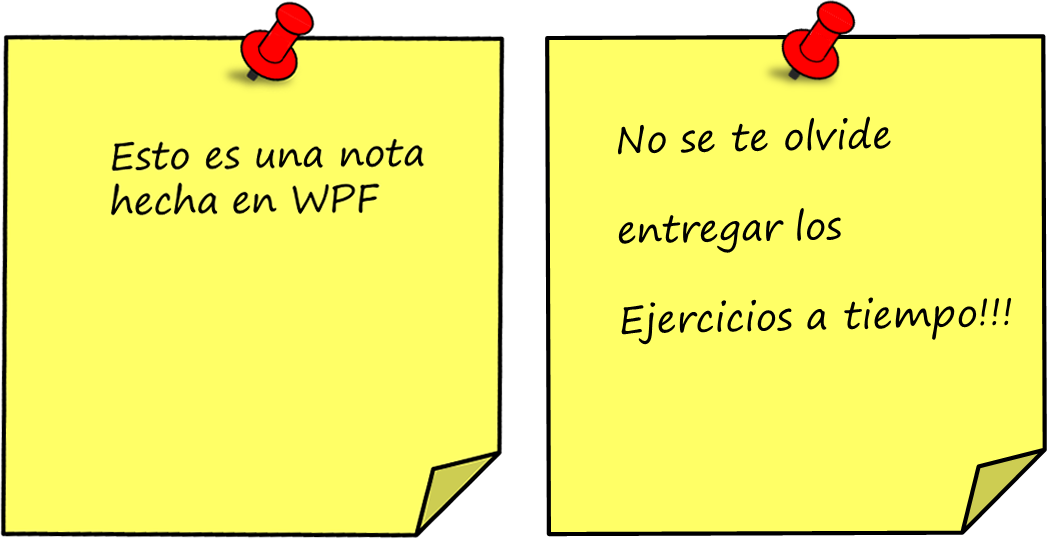
}

}

}

}

## Ejercicio 3. Notas en WPF



## Respuesta en XAML:

<UserControl x:Class="Ejercicios\_U4.UC\_Notas"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

mc:Ignorable="d"

d:DesignHeight="282" d:DesignWidth="262">

<Canvas Width="253" Height="272">

<Polygon Stroke="Black" Fill="#FFFF66" Points="0,18 250,18 250,226 208,267 0,267"/>

<Polygon Stroke="Black" Fill="DarkGoldenrod" Points="208,267 250,226 216,233 "/>

<Ellipse Width="30" Height="10" Canvas.Left="105" Canvas.Top="35" Opacity=".5">

<Ellipse.Fill>

<RadialGradientBrush>

<GradientStop Color="Black" Offset="0"/>

<GradientStop Color="#FFFF66" Offset="1"/>

</RadialGradientBrush>

</Ellipse.Fill>

</Ellipse>

<Canvas Canvas.Left="140" Canvas.Top="-10">

<Line X1="0" Y1="0" X2="0" Y2="5" StrokeEndLineCap="Round" Stroke="Black" StrokeThickness="5" Width="5" Height="10" Canvas.Left="15" Canvas.Top="43" />

<Ellipse Stroke="Black" Fill="Red" Width="30" Height="21" Canvas.Left="0" Canvas.Top="24">

</Ellipse>

<Path Stroke="Black" Fill="Red" Data="M0,18 V0 H13 V18 A2,2 0 0 1 0,18" Canvas.Left="8.5" Canvas.Top="13" />

<Ellipse Stroke="Black" Fill="Red" Width="18" Height="13" Canvas.Top="3" Canvas.Left="5.8"/>

<Canvas.RenderTransform>

<RotateTransform Angle="40"/>

</Canvas.RenderTransform>

</Canvas>

<Rectangle Width="210" Height="185" Canvas.Left="20" Canvas.Top="43">

<Rectangle.Fill>

<VisualBrush Stretch="Uniform" >

<VisualBrush.Visual>

<TextBlock FontFamily="Segoe Print" TextWrapping="Wrap" x:Name="texto" Text="" Width="210" Height="185"/>

</VisualBrush.Visual>

</VisualBrush>

</Rectangle.Fill>

</Rectangle>

</Canvas>

</UserControl>

## Código en WPF:

public partial class UC\_Notas : UserControl

{

public UC\_Notas()

{

InitializeComponent();

}

private string text;

public string Texto

{

get { return text; }

set

{

text = value;

texto.Text = text;

}

}

}