

UF1. [PAC03] Solución

Para la realización de las siguientes cuestiones deberéis ayudaros del libro de texto, capítulos 4 y 5, y consultar, si lo creéis necesario, internet.

Debéis subir un único archivo comprimido que contenga UN ÚNICO DOCUMENTO PDF con las respuestas del test, y además se adjuntarán las soluciones de la parte práctica, con nombres identificativos.

Todas las preguntas valen lo mismo, cada pregunta tiene un valor de 2 puntos, por lo que las preguntas tipo test tienen un valor de 0,25 cada una.

Las preguntas tipo test se corregirán sólo en la caja que veréis después de las preguntas.

Recordad que la fecha límite para la entrega de la PAC es el día 17 de NOVIEMBRE.

Sugerencia: Para la realización del ejercicio práctico es necesario tener instalado Visual Studio.

Información: Cualquier PAC copiada y/o en la que se haya utilizado “copy-paste” de código ya escrito será puntuada con un 0.

Actividades

Parte teórica

1. Responde a este test en la matriz de respuestas que se encuentra al final de las preguntas:
 - 1.1. Respecto al lenguaje XAML:
 - a. Se trata de un lenguaje orientado únicamente al desarrollo de interfaces web.
 - b. Se trata de un lenguaje orientado únicamente al desarrollo de interfaces móviles.
 - c. Se trata de un lenguaje orientado únicamente al desarrollo de interfaces de escritorio.
 - d. Se trata de un lenguaje orientado al desarrollo de interfaces tanto web como de escritorio.

- 1.2. SVG está pensado para:
- Definir simples especificadas mediante ecuaciones matemáticas o mediante expresiones algebraicas.
 - Definir figuras complejas formadas por la unión de varias figuras simples o complejas.
 - Definir simples especificadas mediante ecuaciones matemáticas o mediante expresiones algebraicas y definir figuras complejas formadas por la unión de varias figuras simples o complejas.
 - Ninguna de las anteriores.
- 1.3. Indica cuáles de los siguientes son elementos de SVG:
- Elementos contenedores, elementos gráficos y elementos de animación.
 - Elementos definitivos, elementos contenedores y elementos de gradiente.
 - Elementos contenedores, elementos de audio y elementos fijos.
 - Elementos contenedores, elementos moleculares y elementos de animación.
- 1.4. Los componentes WPF se crean a partir de la clase UserControl, por lo que heredarán:
- Propiedades y valores.
 - Propiedades, eventos y métodos.
 - Eventos, valores y métodos.
 - Métodos abstractos, valores y eventos predefinidos.
- 1.5. En el componente Button de WPF la propiedad IsCancel define:
- Permite asociar el botón a la pulsación de la tecla Esc del teclado.
 - Permite asociar el botón a la pulsación de la tecla Enter del teclado.
 - Permite asociar el botón a la pulsación de la tecla BackSpace del teclado.
 - Permite asociar el botón a la pulsación de la tecla Supr del teclado.

- 1.6. La propiedad name
- Es importante definirla porque es el texto que mostrará al pulsar.
 - No es importante definirla porque es el texto que mostrará al pulsar.
 - Es importante definirla porque hace referencia al elemento.
 - No es importante definirla porque hace referencia al elemento.
- 1.7. ¿Cuál de las siguientes opciones no es de ClickMode?
- Release
 - Ischecked
 - Press
 - Over
- 1.8. ¿Cuál de estos no es un evento enrutado?
- LostFocus
 - KeyDown
 - RightTapped
 - Todos son eventos enrutados

MATRIZ DE RESPUESTAS

1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
D	A	A	B	A	C	B	D

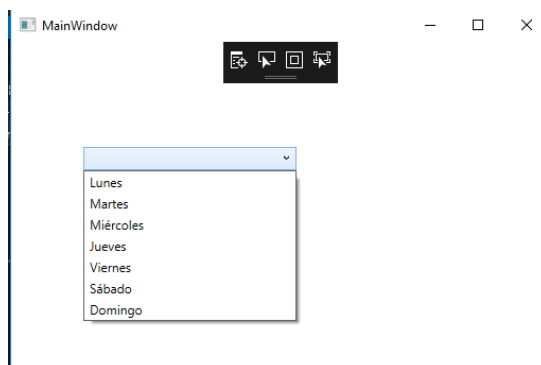
Parte práctica

2. Cree una interfaz mediante XAML donde se puede seleccionar un elemento de una lista desplegable.

```

1 <Window x:Class="Ejercicio5PAC1.MainWindow"
2       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4       xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5       xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6       xmlns:local="clr-namespace:Ejercicio5PAC1"
7       mc:Ignorable="d"
8       Title="MainWindow" Height="350" Width="525">
9     <Grid>
10       <ComboBox x:Name="comboBox" HorizontalAlignment="Left" Margin="68,99,0,0" VerticalAlignment="Top" Width="201">
11         <ComboBoxItem Tag="Lu">Lunes</ComboBoxItem>
12         <ComboBoxItem Tag="Ma">Martes</ComboBoxItem>
13         <ComboBoxItem Tag="Mi">Miércoles</ComboBoxItem>
14         <ComboBoxItem Tag="Ju">Jueves</ComboBoxItem>
15         <ComboBoxItem Tag="Vi">Viernes</ComboBoxItem>
16         <ComboBoxItem Tag="Sa">Sábado</ComboBoxItem>
17         <ComboBoxItem Tag="Do">Domingo</ComboBoxItem>
18       </ComboBox>
19     </Grid>
20 </Window>
21

```



3. Cree un componente que permita simular un conversor de temperatura indistintamente de grados centígrados a grados fahrenheit.

XAML

```

<Window x:Class="Ejercicio2PAC3.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:local="clr-namespace:Ejercicio2PAC3"
  mc:Ignorable="d"
  Title="Conversor temperatura" Height="350" Width="525">
  <Grid>
    <Label x:Name="label" Content="Grados Centígrados:"
      HorizontalAlignment="Left" Margin="65,35,0,0" VerticalAlignment="Top" Height="54"
      Width="204" FontSize="20"/>
  </Grid>
</Window>

```

```

        <Label x:Name="label1" Content="Grados Farenheit:"
HorizontalAlignment="Left" Margin="65,94,0,0" VerticalAlignment="Top" Height="52"
Width="174" FontSize="20"/>
        <TextBox x:Name="txtCentigrados" HorizontalAlignment="Left" Height="23"
Margin="274,48,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"
RenderTransformOrigin="0.633,-0.5"/>
        <TextBox x:Name="txtFarenheit" HorizontalAlignment="Left" Height="23"
Margin="274,107,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>
        <Button x:Name="btnCalcular" Content="Calcular"
HorizontalAlignment="Left" Margin="65,192,0,0" VerticalAlignment="Top"
Width="107" FontSize="20" Height="46" Click="btnCalcular_Click"/>

    </Grid>

</Window>

```

C#

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Ejercicio2PAC3
{
    /// <summary>
    /// Lógica de interacción para MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void btnCalcular_Click(object sender, EventArgs e)
        {
            if (this.txtCentigrados.Text.Length > 0)
            {
                float gCentigrados = float.Parse(this.txtCentigrados.Text);
                float gFarenheit = (gCentigrados * 9 / 5) + 32;
                this.txtFarenheit.Text = gFarenheit.ToString();
            } else if (this.txtFarenheit.Text.Length > 0)
            {
                float gFarenheit = float.Parse(this.txtFarenheit.Text);
                float gCentigrados = (gFarenheit - 32) * 5 / 9;
                this.txtCentigrados.Text = gCentigrados.ToString();
            }
        }
    }
}

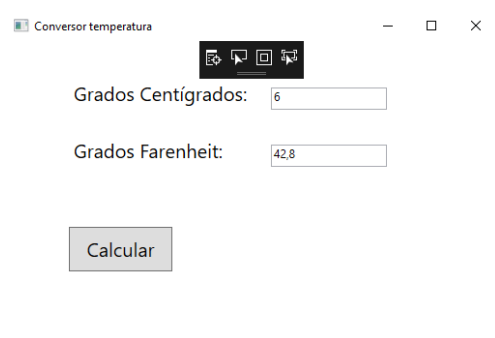
```

```

    } else
    {
        this.txtCentigrados.Text = "0";
        btnCalcular_Click(sender, e);
    }
}
}
}

```

Resultado



4. Cree una tarjeta de crédito de forma que permita ingresar o reintegrar cantidades, pero debe generar un evento en la aplicación cuando se sobrepase el saldo disponible y entre en números rojos. El evento únicamente mostrará un mensaje por pantalla.

XAML

```

<Window x:Class="Ejercicio3PAC3.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Ejercicio3PAC3"
        mc:Ignorable="d"
        Title="MainWindow" Height="350" Width="525">
    <Grid>
        <Label x:Name="label" Content="Cantidad" HorizontalAlignment="Left"
            Margin="67,73,0,0" VerticalAlignment="Top" Height="34" Width="94" FontSize="20"/>
        <TextBox x:Name="txtCantidad" HorizontalAlignment="Left" Height="34"
            Margin="166,73,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="214"/>
        <Button x:Name="btnIngresar" Content="Ingresar"
            HorizontalAlignment="Left" Margin="86,160,0,0" VerticalAlignment="Top" Width="75"
            Click="btnIngresar_Click"/>
        <Button x:Name="btnReintegro" Content="Reintegro"
            HorizontalAlignment="Left" Margin="234,160,0,0" VerticalAlignment="Top"
            Width="75" Click="btnReintegro_Click"/>
    </Grid>

```

</Window>

C#

```
using System;
using System.Windows;

namespace Ejercicio3PAC3
{
    /// <summary>
    /// Lógica de interacción para MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        private float _money;

        public event EventHandler MoneyChanged;

        public MainWindow()
        {
            InitializeComponent();
        }

        protected virtual void OnMoneyChanged()
        {
            if (MoneyChanged != null)
                MoneyChanged(this, EventArgs.Empty);
            else
                if (Money < 0)
                    MessageBox.Show("Estás en números rojos: " +
Money.ToString(), "Mensaje informativo", MessageBoxButton.OK,
MessageBoxImage.Exclamation);
        }

        public float Money
        {
            get
            {
                return _money;
            }

            set
            {
                _money = value;
                OnMoneyChanged();
            }
        }

        private void btnIngresar_Click(object sender, RoutedEventArgs e)
        {
            float cantidad = float.Parse(this.txtCantidad.Text);
            this.txtCantidad.Text = "";
            Money = Money + cantidad;
        }

        private void btnReintegro_Click(object sender, RoutedEventArgs e)
        {

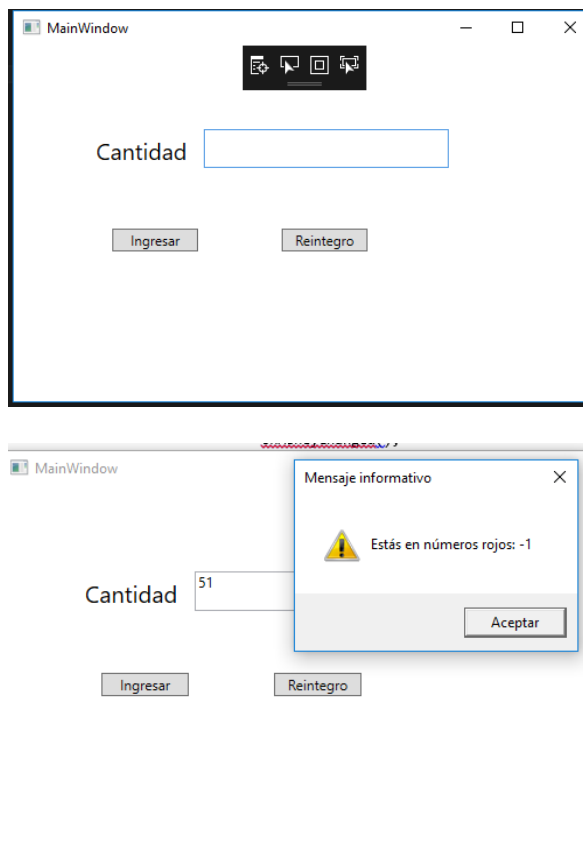
```

```

        float cantidad = float.Parse(this.txtCantidad.Text);
        this.txtCantidad.Text = "";
        Money = Money - cantidad;
    }
}

```

Resultado



5. Cree una interfaz en donde se muestren al usuario las siguientes opciones para seleccionar. La interfaz permite reservar un viaje en el cual se incluyen diferentes extras: Servicio Taxi + 50€, Alojamiento VIP + 200€, Pensión Completa + 140€. Las opciones de vuelo deberán determinar solamente una de las siguientes opciones: Bussiness, Turista, Chárter.

XML

```

<Window x:Class="Ejercicio4PAC3.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

```



```

xmlns:local="clr-namespace:Ejercicio4PAC3"
mc:Ignorable="d"
Title="MainWindow" Height="350" Width="525">

    <Grid Margin="10,-18,-10,18">
        <Canvas HorizontalAlignment="Left" Height="178" Margin="10,23,0,0"
VerticalAlignment="Top" Width="219">
            <Label Content="Vuelo de avión" HorizontalAlignment="Left"
VerticalAlignment="Top" Canvas.Left="10" Canvas.Top="10" FontWeight="Bold"/>
            <RadioButton Content="Bussiness" Canvas.Left="56" Canvas.Top="66"
Width="130"/>
            <RadioButton Content="Turista" Canvas.Left="56" Canvas.Top="101"
Width="130"/>
            <RadioButton Content="Charter" Canvas.Left="56" Canvas.Top="133"
Width="130"/>
        </Canvas>
        <Label Content="Extras&#xD;&#xA;" HorizontalAlignment="Left"
Margin="310,33,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.356,-0.353"
Width="114" Height="25" FontWeight="Bold"/>
        <CheckBox Content="Servicio Taxi + 50€" HorizontalAlignment="Left"
Margin="310,91,0,0" VerticalAlignment="Top" Width="152"/>
        <CheckBox Content="Alojamiento VIP + 200€" HorizontalAlignment="Left"
Margin="310,112,0,0" VerticalAlignment="Top" Width="152"/>
        <CheckBox Content="Pension Completa + 140€" HorizontalAlignment="Left"
Margin="310,133,0,0" VerticalAlignment="Top" Width="152"/>
    </Grid>

</Window>

```

Resultado

