

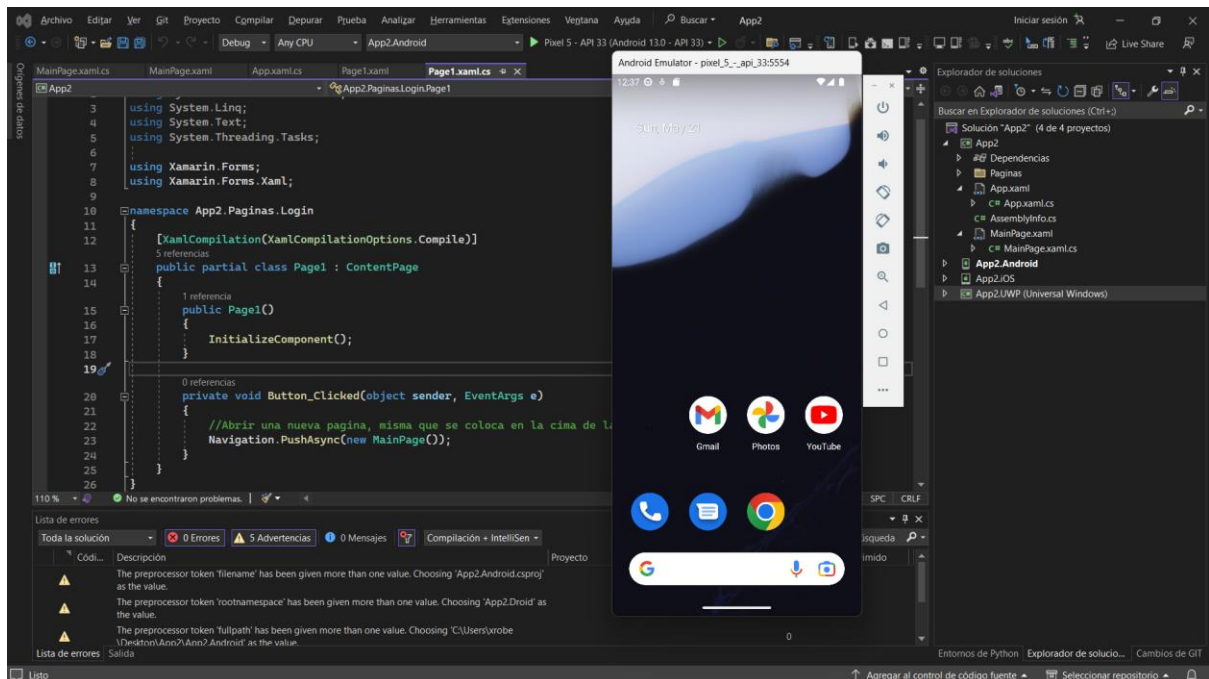


UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE SOFTWARE
ASIGNATURA: APLICACIONES MÓVILES NIVEL: 07

Nombres completos:	Roberth Gabriel Lima Carvajal
Fecha:	20/05/2023
Tema:	Calculadora
Objetivo de esta actividad:	Desarrollar una aplicación móvil con la ayuda de Visual Studio 2022 usando Xamarin.Forms para la creación de una calculadora con sus funciones básicas y entender de mejor forma como funciona esta solución multiplataforma

INDICACIONES:

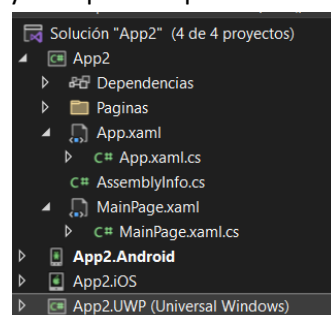
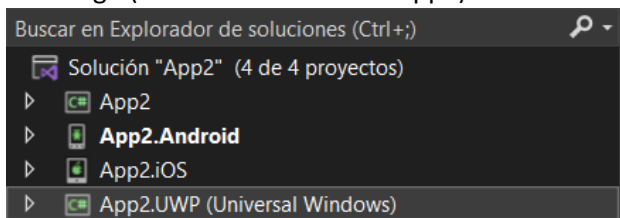
Para la realización de esta actividad se debe de tener el IDE de Visual Studio 2022, e instalado una máquina virtual móvil, para la ejecución del código desarrollado.

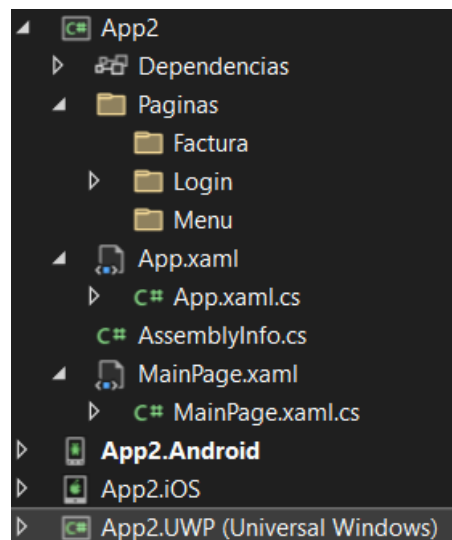


Una vez tenido todo esto, deberemos de programar una aplicación móvil usando Xamarin.Forms que funcione como calculadora.

DESARROLLO:

1. Crear un Proyecto Xamarin.Forms, y que esté cargado y listo para empezar a escribir líneas de código (En este caso se llama App2):





2. Código dentro de cada archivo:

- App.xaml.css

```
using System;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace App2
{
    7 referencias
    public partial class App : Application
    {
        3 referencias
        public App()
        {
            InitializeComponent();

            MainPage = new NavigationPage( new App2.Paginas.Login.Page1());

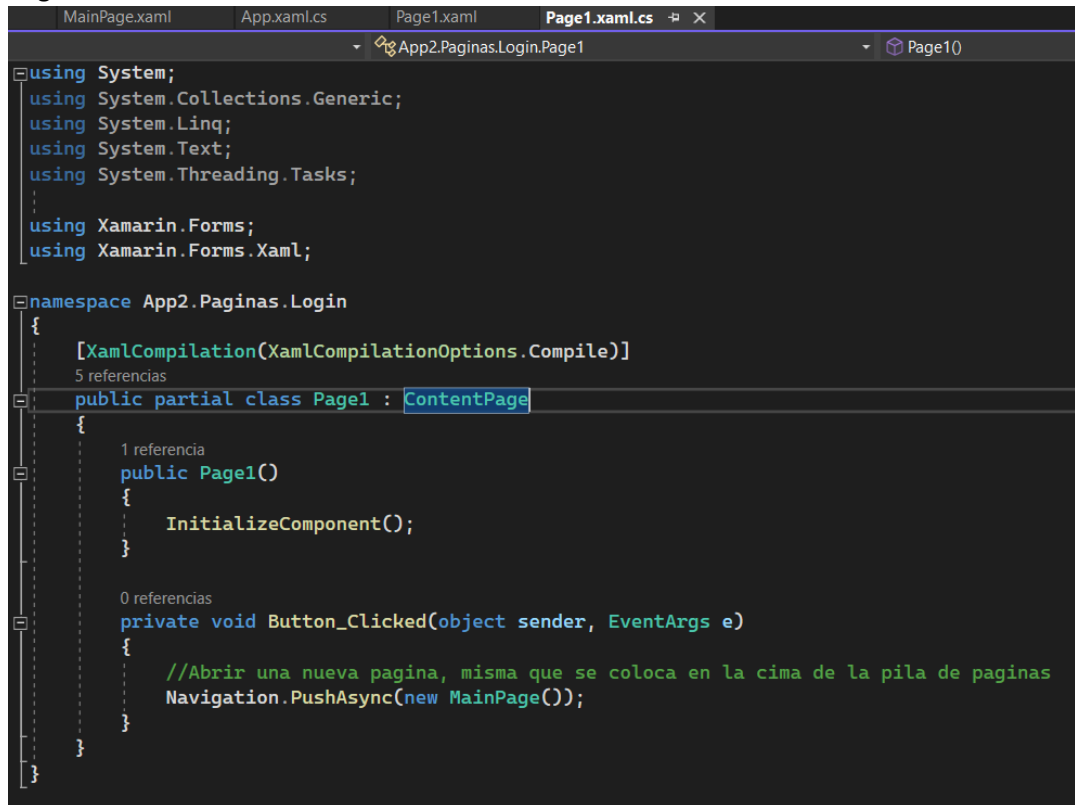
            //
            MainPage = new MainPage();
        }

        0 referencias
        protected override void OnStart()
        {
        }

        0 referencias
        protected override void OnSleep()
        {
        }

        0 referencias
        protected override void OnResume()
        {
        }
    }
}
```

- Page1.xaml.css



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

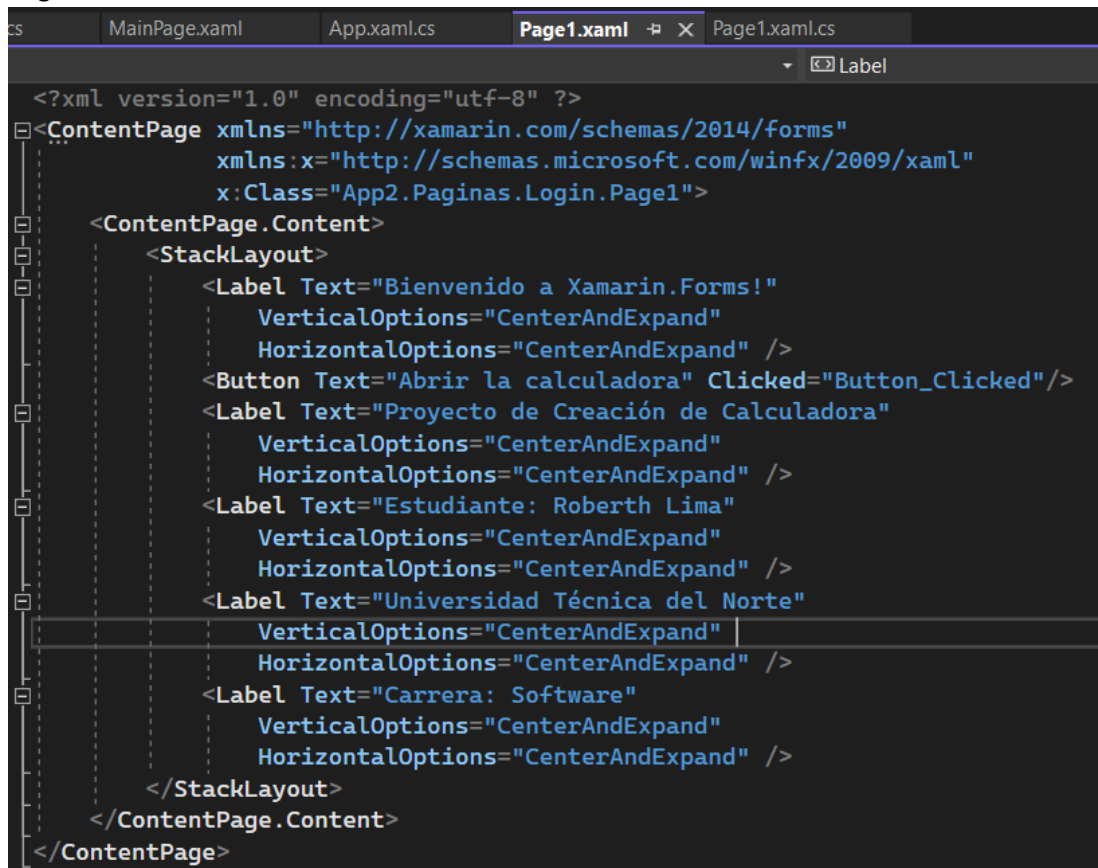
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace App2.Paginas.Login
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    5 referencias
    public partial class Page1 : ContentPage
    {
        1 referencia
        public Page1()
        {
            InitializeComponent();
        }

        0 referencias
        private void Button_Clicked(object sender, EventArgs e)
        {
            //Abrir una nueva pagina, misma que se coloca en la cima de la pila de paginas
            Navigation.PushAsync(new MainPage());
        }
    }
}

```

- Page1.xaml



```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="App2.Paginas.Login.Page1">
    <ContentPage.Content>
        <StackLayout>
            <Label Text="Bienvenido a Xamarin.Forms!"
                VerticalOptions="CenterAndExpand"
                HorizontalOptions="CenterAndExpand" />
            <Button Text="Abrir la calculadora" Clicked="Button_Clicked"/>
            <Label Text="Proyecto de Creación de Calculadora"
                VerticalOptions="CenterAndExpand"
                HorizontalOptions="CenterAndExpand" />
            <Label Text="Estudiante: Roberth Lima"
                VerticalOptions="CenterAndExpand"
                HorizontalOptions="CenterAndExpand" />
            <Label Text="Universidad Técnica del Norte"
                VerticalOptions="CenterAndExpand"
                HorizontalOptions="CenterAndExpand" />
            <Label Text="Carrera: Software"
                VerticalOptions="CenterAndExpand"
                HorizontalOptions="CenterAndExpand" />
        </StackLayout>
    </ContentPage.Content>
</ContentPage>

```

- MainPage.xaml.css

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

namespace App2
{
    5 referencias
    public partial class MainPage : ContentPage
    {
        private string currentOperator;
        private double firstNumber;
        private double secondNumber;

        1 referencia
        public MainPage()
        {
            InitializeComponent();
            numberEntry.Focused += (sender, e) => numberEntry.Unfocus();
        }

        // Método para eliminar un número al tocar el botón "C"
        0 referencias
        private void OnClearClicked(object sender, EventArgs e)
        {
            if (!string.IsNullOrEmpty(numberEntry.Text))
            {
                numberEntry.Text = numberEntry.Text.Substring(0, numberEntry.Text.Length - 1);
            }
        }

        // Método para eliminar todo al tocar el botón "AC"
        0 referencias
        private void OnAllClearClicked(object sender, EventArgs e)
        {
            numberEntry.Text = string.Empty;
            resultLabel.Text = string.Empty;
        }

        1 referencia
        private double CalculateResult()
        {
            double result = 0;

            switch (currentOperator)
            {
                case "+":
                    result = firstNumber + secondNumber;
                    break;
                case "-":
                    result = firstNumber - secondNumber;
                    break;
                case "x":
                    result = firstNumber * secondNumber;
                    break;
                case "/":
                    result = firstNumber / secondNumber;
                    break;
            }

            return result;
        }
    }
}
```

```

0 referencias
private void OnButtonClicked(object sender, EventArgs e)
{
    Button button = (Button)sender;
    string buttonText = button.Text;

    if (buttonText == "+" || buttonText == "-" || buttonText == "x" || buttonText == "/")
    {
        currentOperator = buttonText;
        firstNumber = double.Parse(numberEntry.Text);
        numberEntry.Text = string.Empty;
    }
    else if (buttonText == "=")
    {
        secondNumber = double.Parse(numberEntry.Text);
        double result = CalculateResult();
        resultLabel.Text = result.ToString();
        numberEntry.Text = result.ToString();
    }
    else
    {
        numberEntry.Text += buttonText;
    }
}
}

```

- MainPage.xaml

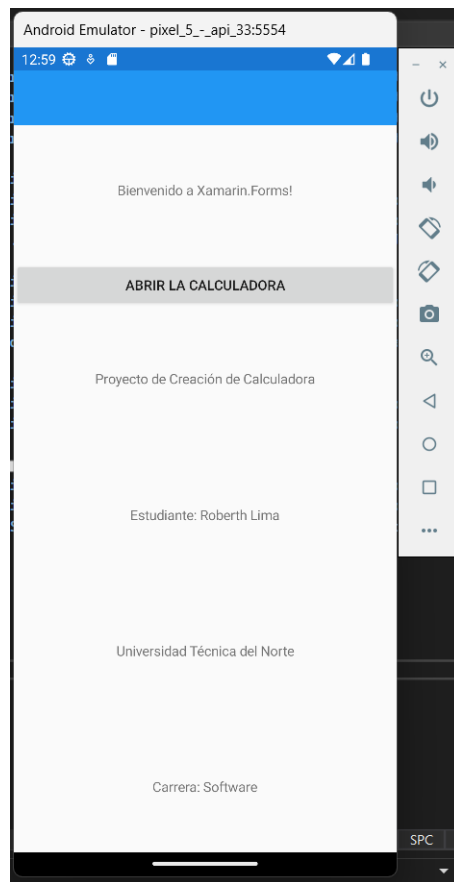
```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="App2.MainPage"
    BackgroundColor="#1e2952">
    <ContentPage.Content>
        <Grid>
            <Grid.RowDefinitions>
                <RowDefinition Height="2*" />
                <RowDefinition Height="3*" />
            </Grid.RowDefinitions>
            <Entry x:Name="numberEntry" BackgroundColor="#162138" TextColor="White" FontSize="36" HorizontalTextAlignment="End" VerticalTextAlignment="Center" Grid.Row="0" />
            <Grid BackgroundColor="#162138" Grid.Row="1">
                <Label x:Name="resultLabel" TextColor="White" FontSize="36" HorizontalTextAlignment="End" VerticalTextAlignment="Center" />
                <Label x:Name="resultDisplayLabel" TextColor="White" FontSize="36" HorizontalTextAlignment="End" VerticalTextAlignment="Center" />
            </Grid>
            <Frame BackgroundColor="#1e2952" Padding="10" CornerRadius="20" Grid.Row="1">
                <Grid>
                    <Grid.RowDefinitions>
                        <RowDefinition Height="Auto" />
                        <RowDefinition Height="*" />
                        <RowDefinition Height="*" />
                        <RowDefinition Height="*" />
                    </Grid.RowDefinitions>
                    <Grid.ColumnDefinitions>
                        <ColumnDefinition Width="*" />
                        <ColumnDefinition Width="*" />
                        <ColumnDefinition Width="*" />
                    </Grid.ColumnDefinitions>
                    <Button Text="+" BackgroundColor="Black" TextColor="Yellow" Grid.Row="0" Grid.Column="0" Clicked="OnButtonClicked" />
                    <Button Text="-" BackgroundColor="Black" TextColor="Yellow" Grid.Row="0" Grid.Column="1" Clicked="OnButtonClicked" />
                    <Button Text="x" BackgroundColor="Black" TextColor="Yellow" Grid.Row="0" Grid.Column="2" Clicked="OnButtonClicked" />
                    <Button Text="/" BackgroundColor="Black" TextColor="Yellow" Grid.Row="0" Grid.Column="3" Clicked="OnButtonClicked" />
                    <Button Text="7" BackgroundColor="Black" TextColor="White" Grid.Row="1" Clicked="OnButtonClicked" />
                    <Button Text="8" BackgroundColor="Black" TextColor="White" Grid.Row="1" Grid.Column="1" Clicked="OnButtonClicked" />
                    <Button Text="9" BackgroundColor="Black" TextColor="White" Grid.Row="1" Grid.Column="2" Clicked="OnButtonClicked" />
                    <Button Text="C" BackgroundColor="Black" TextColor="Red" Grid.Row="1" Grid.Column="3" Clicked="OnClearClicked" />
                    <Button Text="0" BackgroundColor="Black" TextColor="White" Grid.Row="2" Clicked="OnButtonClicked" />
                    <Button Text="5" BackgroundColor="Black" TextColor="White" Grid.Row="2" Grid.Column="1" Clicked="OnButtonClicked" />
                    <Button Text="6" BackgroundColor="Black" TextColor="White" Grid.Row="2" Grid.Column="2" Clicked="OnButtonClicked" />
                    <Button Text="AC" BackgroundColor="Black" TextColor="Red" Grid.Row="2" Grid.Column="3" Clicked="OnAllClearClicked" />
                    <Button Text="1" BackgroundColor="Black" TextColor="White" Grid.Row="3" Clicked="OnButtonClicked" />
                    <Button Text="2" BackgroundColor="Black" TextColor="White" Grid.Row="3" Grid.Column="1" Clicked="OnButtonClicked" />
                    <Button Text="3" BackgroundColor="Black" TextColor="White" Grid.Row="3" Grid.Column="2" Clicked="OnButtonClicked" />
                    <Button Text="Roberth Lima" BackgroundColor="Black" TextColor="White" Grid.Row="4" />
                    <Button Text="0" BackgroundColor="Black" TextColor="White" Grid.Row="4" Grid.Column="1" Clicked="OnButtonClicked" />
                    <Button Text="." BackgroundColor="Black" TextColor="White" Grid.Row="4" Grid.Column="2" Clicked="OnButtonClicked" />
                    <Button Text="=" BackgroundColor="Red" TextColor="White" FontSize="24" Grid.Row="4" Grid.RowSpan="2" Grid.Column="3" Clicked="OnButtonClicked" />
                </Grid>
            </Frame>
        </Grid>
    </ContentPage.Content>
</ContentPage>

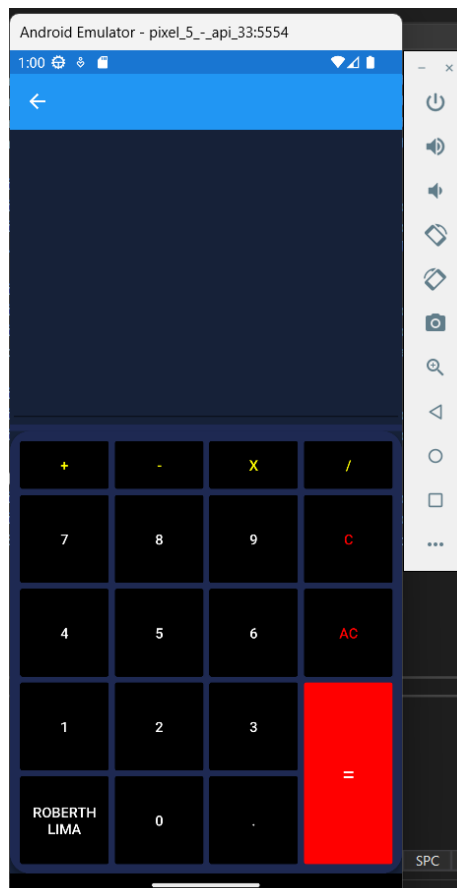
```

3. Pantallas creadas:

- Pantalla Principal



- Pantalla de Calculadora



4. Probando funcionalidades

- Suma, Resta, Multiplicación y División



CONCLUSIONES:

Xamarin.Forms es una opción increíblemente útil para desarrollar aplicaciones que funcionen en múltiples plataformas. Con esta herramienta, puedes escribir el código una vez y utilizarlo en iOS, Android y Windows, lo cual ahorra tiempo y recursos. No es necesario crear interfaces de usuario separadas para cada plataforma, ya que Xamarin.Forms permite crear interfaces compartidas que se adaptan automáticamente a cada sistema operativo.

A pesar de ser una solución multiplataforma, Xamarin.Forms ofrece una experiencia de usuario nativa en cada plataforma. Esto se logra al utilizar controles y componentes nativos de iOS, Android y Windows. La ventaja es que tus aplicaciones se verán y funcionarán de manera similar a las aplicaciones desarrolladas específicamente para cada plataforma, brindando a los usuarios una experiencia fluida y familiar.

Xamarin.Forms se basa en el sólido framework de desarrollo .NET, lo que te brinda acceso a una amplia gama de bibliotecas y herramientas para facilitar el proceso de desarrollo. Puedes aprovechar el poder y la flexibilidad de .NET para crear aplicaciones con características avanzadas. Además, Xamarin.Forms se integra sin problemas con otras tecnologías de Microsoft, como Azure y Xamarin Test Cloud, lo que te permite implementar y probar tus aplicaciones de forma más eficiente.