

# DESARROLLO DE INTERFACES

## Testing en MAUI



Marcos Zahonero

## Índice

<b>Introducción.....</b>	<b>3</b>
<b>Tiempo utilizado.....</b>	<b>4</b>
<b>Material utilizado.....</b>	<b>5</b>
Mi ordenador de casa.....	5
Portátil de clase.....	5
<b>Objetivos de la práctica.....</b>	<b>6</b>
<b>Desarrollo de la Actividad.....</b>	<b>7</b>
Página - MainPage.....	7
Pantalla inicial.....	7
Pantalla con una operación.....	7
Pantalla intentado dividir entre cero.....	8
Códigos de los archivos.....	9
MainPage.....	9
MainPage.xaml.....	9
MainPage.xaml.cs.....	9
MainPageViewModel.cs.....	10
Clases extras.....	11
(models) Operacions.cs.....	11
(UnitTestProject) OperacionsTests.cs.....	12
Resultado Final – Vista Móvil.....	13
Resultado Final – Vista Ordenador.....	13
GitHub.....	14
<b>Conclusión.....</b>	<b>15</b>
<b>Problemas y/o sugerencias en la actividad.....</b>	<b>16</b>
1# - Mi extra a la actividad.....	16
<b>Webgrafía.....</b>	<b>17</b>

## **Introducción**

En esta actividad veremos como seguir utilizando los ICommandds pero lo más destacable y novedoso de la actividad sería la utilización de clases testing para hacer pruebas a nuestra aplicación y ver si funciona como esperamos.

## **Tiempo utilizado**

He utilizado 3 horas, y para hacer el documento y vídeos aproximadamente 30/50 minutos, así que sería aproximadamente 4 horas.

## Material utilizado

- El portátil de clase, con mi ordenador personal de casa.
- Visual Studio.

### Mi ordenador de casa

Componente	Nombre
Sistema Operativo	Windows 11 Pro
Procesador	AMD Ryzen 5 2600X
Tarjeta gráfica	NVIDIA GeForce GTX 1650 4GB
Memoria RAM	16 GB
Placa base	B450M-A PRO MAX
Disco duro 1#	KIOXIA EXCERIA G2 SSD 1 TB
Disco duro 2#	ST1000DM010-2EP102 HDD 1 TB

### Portátil de clase

Componente	Nombre
Sistema Operativo	Windows 10 Pro
Procesador	Intel(R) Core(TM) i5-7300U @ 2.60GHz 2.71 GHz
Tarjeta gráfica	(Integrada)
Modelo portátil	HP EliteBook 840 G5
Memoria RAM	16 GB
Placa base	HP 83B2
Disco duro	Disco duro HDD 256GB

## Objetivos de la práctica

- Realizar Testing entendiendo tu programa
- Entender el funcionamiento de como usarlos ICommand.
- Entender el funcionamiento de los Binding y utilizarlos.
- Diseñar una interfaz interesante de ver.

## Desarrollo de la Actividad

Esta aplicación consta de 1 página con los siguientes elementos:

- 2 Entrys
- 4 Buttons
- 2 Labels
- 2 Frames

### Página – MainPage

#### Pantalla inicial

En este momento no contiene ningún número puesto ni ninguno calculado, así que así sería la pantalla al iniciar la aplicación.



#### Pantalla con una operación

Aquí hacemos  $10 + 50$  y daría 60, y sale abajo centrado.



## Pantalla intentado dividir entre cero

Calculadora

10

0

+

-

×

÷

Error: División por cero



# MainPage

## MainPage.xaml

```
verticalStackLayout>
```

## MainPage.xaml.cs

## MainPageViewModel.cs

```

public partial class MainPageViewModel : INotifyPropertyChanged
{
    private int _num1;
    4 referencias
    public int Num1
    {
        get => _num1;
        set
        {
            if (_num1 == value) return;
            _num1 = value;
        }
    }

    private int _num2;
    4 referencias
    public int Num2
    {
        get => _num2;
        set
        {
            if (_num2 == value) return;
            _num2 = value;
        }
    }

    private String _result;
    5 referencias
    public String Result
    {
        get => _result;
        set
        {
            if (_result == value) return;
            _result = value;
            OnPropertyChanged(nameof(Result));
        }
    }

    1 referencia
    public ICommand SumarCommand { get; set; }
    1 referencia
    public ICommand RestarCommand { get; set; }
    1 referencia
    public ICommand MultiplicarCommand { get; set; }
    1 referencia
    public ICommand DividirCommand { get; set; }
}

1 referencia
public MainPageViewModel()
{
    SumarCommand = new Command(Sumar);
    RestarCommand = new Command(Restar);
    MultiplicarCommand = new Command(Multiplicar);
    DividirCommand = new Command(Dividir);
    Debug.WriteLine("ddd");
}

public event PropertyChangedEventHandler? PropertyChanged;
1 referencia
protected void OnPropertyChanged([CallerMemberName] string propertyName = null)
{
    PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));
}

private void Sumar()
{
    Result = Operacions.Sumar(Num1, Num2).ToString();
}

private void Restar()
{
    Result = Operacions.Restar(Num1, Num2).ToString();
}

private void Multiplicar()
{
    Result = Operacions.Multiplicar(Num1, Num2).ToString();
}

private void Dividir()
{
    Result = Operacions.Dividir(Num1, Num2);
}

```

## Clases extras

### (Models) Operacions.cs

```
public static class Operacions
{
    2 referencias
    public static int Sumar(int a, int b)
    {
        return a + b;
    }

    2 referencias
    public static int Restar(int a, int b)
    {
        return a - b;
    }

    2 referencias
    public static int Multiplicar(int a, int b)
    {
        return a * b;
    }

    3 referencias
    public static string Dividir(int a, int b)
    {
        try
        {
            return (a / b).ToString();
        }
        catch (DivideByZeroException)
        {
            return "Error: División por cero";
        }
    }
}
```

## (UnitTestProject) OperacionsTests.cs

```
0 referencias
public class OperacionsTests
{
    // Test para la suma
    [Theory]
    [InlineData(3, 4, 7)] // 3 + 4 = 7
    [InlineData(-1, 5, 4)] // -1 + 5 = 4
    0 referencias
    public void Sumar_ShouldReturnCorrectResult(int num1, int num2, int expected)
    {
        var result = Operacions.Sumar(num1, num2);
        Assert.Equal(expected, result);
    }

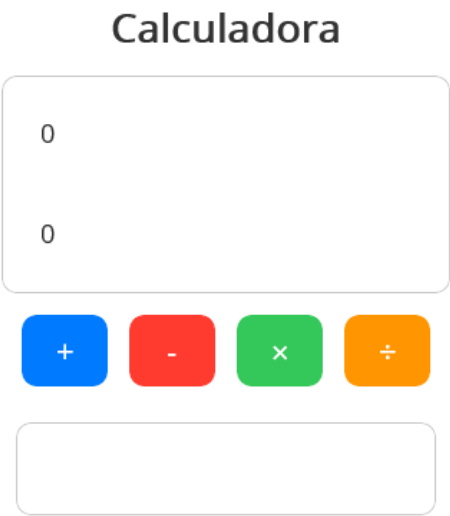
    // Test para la resta
    [Theory]
    [InlineData(5, 3, 2)] // 5 - 3 = 2
    [InlineData(0, 2, -2)] // 0 - 2 = -2
    0 referencias
    public void Restar_ShouldReturnCorrectResult(int num1, int num2, int expected)
    {
        var result = Operacions.Restar(num1, num2);
        Assert.Equal(expected, result);
    }

    // Test para la multiplicación
    [Theory]
    [InlineData(2, 3, 6)] // 2 * 3 = 6
    [InlineData(-2, 4, -8)] // -2 * 4 = -8
    0 referencias
    public void Multiplicar_ShouldReturnCorrectResult(int num1, int num2, int expected)
    {
        var result = Operacions.Multiplicar(num1, num2);
        Assert.Equal(expected, result);
    }

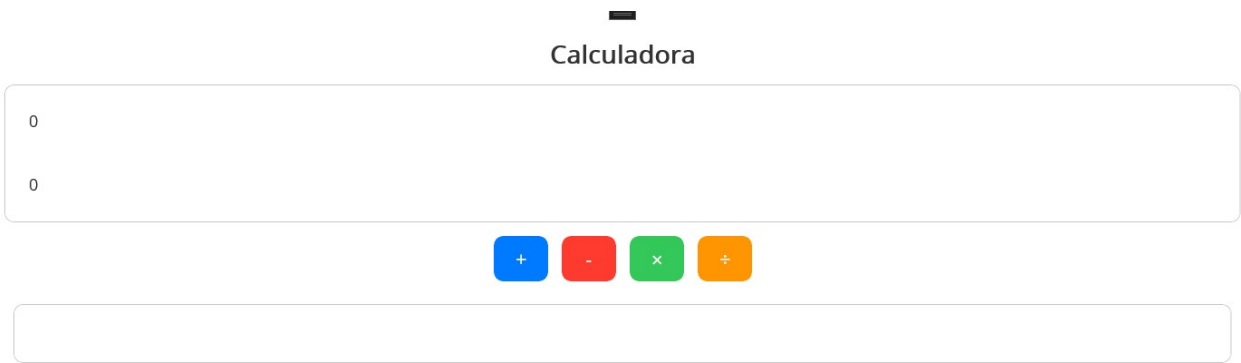
    // Test para la división
    [Theory]
    [InlineData(6, 2, "3")] // 6 / 2 = 3
    [InlineData(5, 2, "2")] // 5 / 2 = 2 (división entera)
    0 referencias
    public void Dividir_ShouldReturnCorrectResult(int num1, int num2, string expected)
    {
        var result = Operacions.Dividir(num1, num2);
        Assert.Equal(expected, result);
    }

    // Test para la división por cero
    [Theory]
    [InlineData(5, 0, "Error: División por cero")] // División por cero
    0 referencias
    public void Dividir_ByZero_ShouldReturnError(int num1, int num2, string expected)
    {
        var result = Operacions.Dividir(num1, num2);
        Assert.Equal(expected, result);
    }
}
```

Resultado Final – Vista Móvil



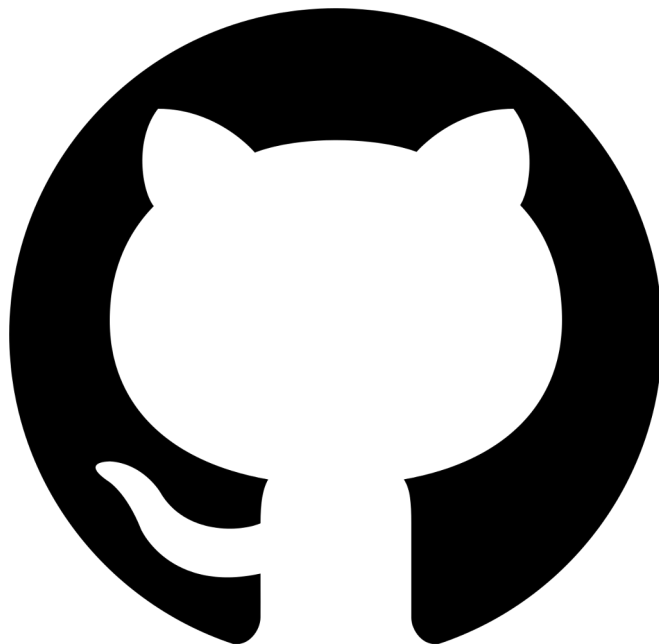
Resultado Final – Vista Ordenador



## GitHub

En esta actividad he añadido GitHub para añadir el proyecto completo, junto a una descripción en el [README.md](#) para añadir una breve explicación y preview de cada actividad.

Enlace: [Maek0s/2DAM DesarrolloInterfaces](#)



## **Conclusión**

En conclusión, el testing es algo sencillo si tu código está bien planteado y si es algo fácil de ver y probar, es decir, puede llegar a ser muy útil e imprescindible en algunos casos pero en otros no tanto.

## Problemas y/o sugerencias en la actividad

### 1# - Mi extra a la actividad

En este caso hice todo en un solo vídeo, no hay mucho que comentar así que decidí agruparlo todo en un solo vídeo.

Vídeo: <https://youtu.be/e4QgTeA5mLI>





## Webgrafía

- <https://visualstudio.microsoft.com/es/vs/community/>
- <https://dotnet.microsoft.com/es-es/apps/maui>
- **ICommand:**  
<https://learn.microsoft.com/en-us/dotnet/maui/fundamentals/data-binding/commanding?view=net-maui-9.0>
- **ViewModels:**  
<https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm>