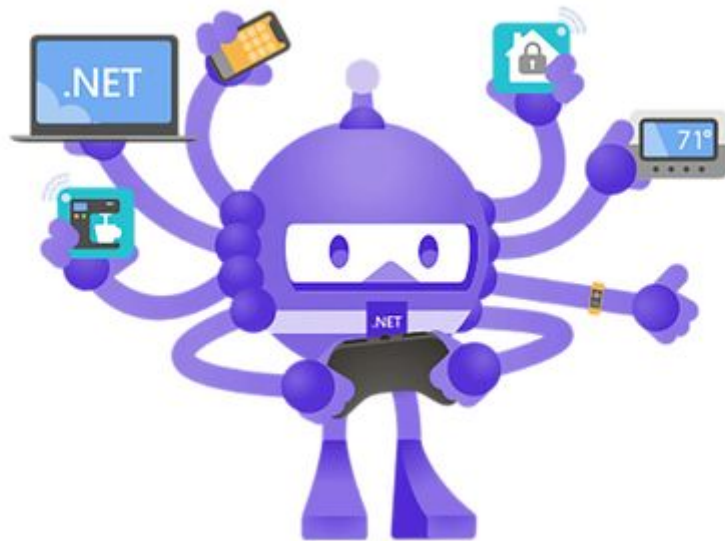


APLICACIONES CON MAUI .NET



Índice

1. Introducción.....	3
2. Objetivos.....	3
3. Actividades.....	4
3.1 Apartado 1. Utilización de StackLayout.....	4
3.2 Apartado 2. Utilización de Grid.....	7
4. Problemas.....	9
5. Conclusión.....	9

1.Introducción

En esta práctica desarrollaremos una aplicación con Maui .NET donde utilizaremos dos formas de introducir las etiquetas, el primero de ellos con StackLayout donde se apilan los objetos uno debajo del otro, predeterminadamente, pero también podemos cambiar la configuración para que dicha orientación se realice de manera horizontal.

En segundo lugar, organizaremos la estructura con algo más complejo como es el caso de un Grid, este organiza los elementos en filas y columnas, aunque predeterminadamente esta estructura viene delimitada por una fila y una columna, aunque se pueden añadir más de ellas.

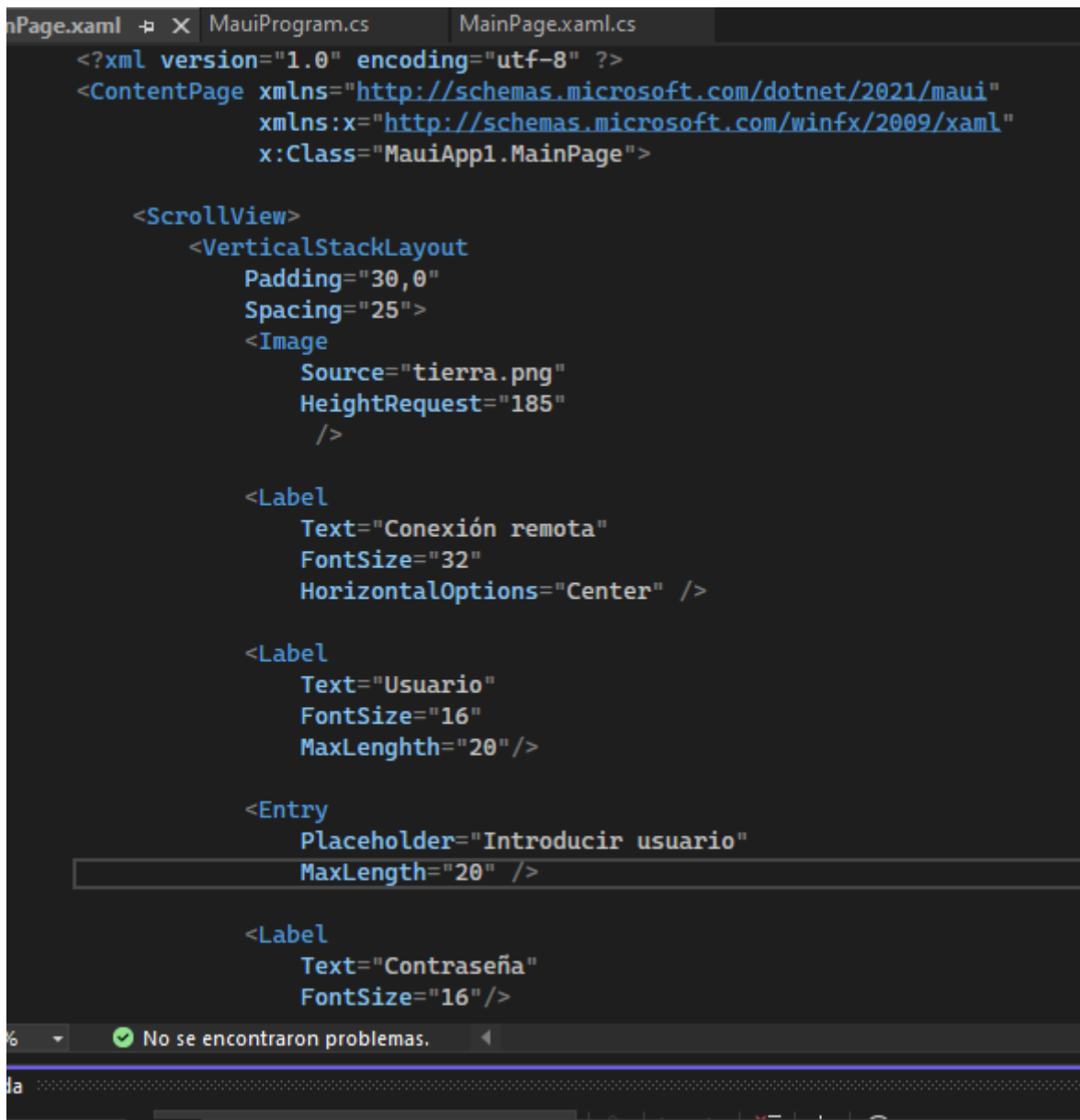
2.Objetivos

Los objetivos de la actividad son realizar diseños dentro de la aplicación de Visual Studio Community, probando diferentes elementos y formatos dentro de los elementos y a su vez, poder ver diferentes maneras de organizar la estructura para que nos salga lo mismo, según nos pidan o nosotros nos manejemos mejor.

3.Actividades

3.1 Apartado 1. Utilización de StackLayout

En este apartado vamos a visualizar el procedimiento para diseñar en Maui .NET con el método de <StackLayout> en el cual, se apilarán los objetos/elementos de manera vertical, uno encima del otro, de forma predeterminada, ya que también tenemos la opción de poder poner objetos uno al lado del otro.



```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
x:Class="MauiApp1.MainPage">

    <ScrollView>
        <VerticalStackLayout
            Padding="30,0"
            Spacing="25">
            <Image
                Source="tierra.png"
                HeightRequest="185"
            />

            <Label
                Text="Conexión remota"
                FontSize="32"
                HorizontalOptions="Center" />

            <Label
                Text="Usuario"
                FontSize="16"
                MaxLength="20" />

            <Entry
                Placeholder="Introducir usuario"
                MaxLength="20" />

            <Label
                Text="Contraseña"
                FontSize="16" />
        </VerticalStackLayout>
    </ScrollView>
</ContentPage>
```

Lo primero que encontramos en el código es el elemento *<ScrollView>*, este es muy útil para cuando tenemos muchas cosas en la pantalla que no caben y nos permite desplazarnos para visualizar lo que hay más abajo y que no esté todo amontonado, sin que quede bien. Lo ponemos antes que nada para abarcar todos los elementos que pondremos en el diseño.

Lo siguiente que vemos es el elemento *<VerticalStackLayout>*, es muy importante la utilidad de esta herramienta ya que organiza los elementos de la interfaz de una manera vertical, de forma ordenada, uno encima del otro.

A continuación pondremos los elementos que queremos visualizar en la pantalla de manera organizada, ya que irá uno encima del otro:

ELEMENTOS

- **<Image>**: Esta etiqueta es útil para introducir una imagen en la aplicación. Se puede introducir o bien la ruta de la imagen o meter la misma en la carpeta *Resource/Images* así solo colocamos el nombre de la imagen seguido del tipo.
- **<Label>**: Esta etiqueta es útil si queremos colocar cualquier cuadro de texto. Dentro de la misma se podrán modificar parámetros según nuestras necesidades.
- **<Entry>**: Con este elemento se puede introducir cualquier tipo de texto, ya sea un usuario, una contraseña como en nuestro caso, o lo que queramos.
- **<Checkbox>**: Con este elemento se generan cajas para poder seleccionar o deseleccionar opciones según a gusto de los usuarios.
- **<RadioButton>**: Permite que el usuario seleccione una opción de las que se proporcionan.
- **<Button>**: Permite la creación de un botón.

Hay más elementos que se podrán ir viendo a lo largo de las actividades.

```
Page.xaml  MauiProgram.cs  MainPage.xaml.cs

<Label
    Text="Contraseña"
    FontSize="16" />

<Entry
    IsPassword="True"
    MaxLength="20"
    Placeholder="*****" />

<StackLayout Orientation="Horizontal">
    <Label Text="¿Guardar contraseña?" VerticalOptions="Center" />
    <CheckBox IsChecked="True" HorizontalOptions="Center" />
</StackLayout>

<Label Text="Elige servidor:" FontSize="16" />
<StackLayout Orientation="Horizontal" HorizontalOptions="Center">
    <RadioButton Content="Madrid" />
    <RadioButton Content="Barcelona" />
    <RadioButton Content="Valencia" IsChecked="True" />
</StackLayout>

<Button
    x:Name="CounterBtn"
    Text="Acceder"
    VerticalOptions="EndAndExpand"
    HorizontalOptions="Center" />
</VerticalStackLayout>
</ScrollView>

</ContentPage>
```

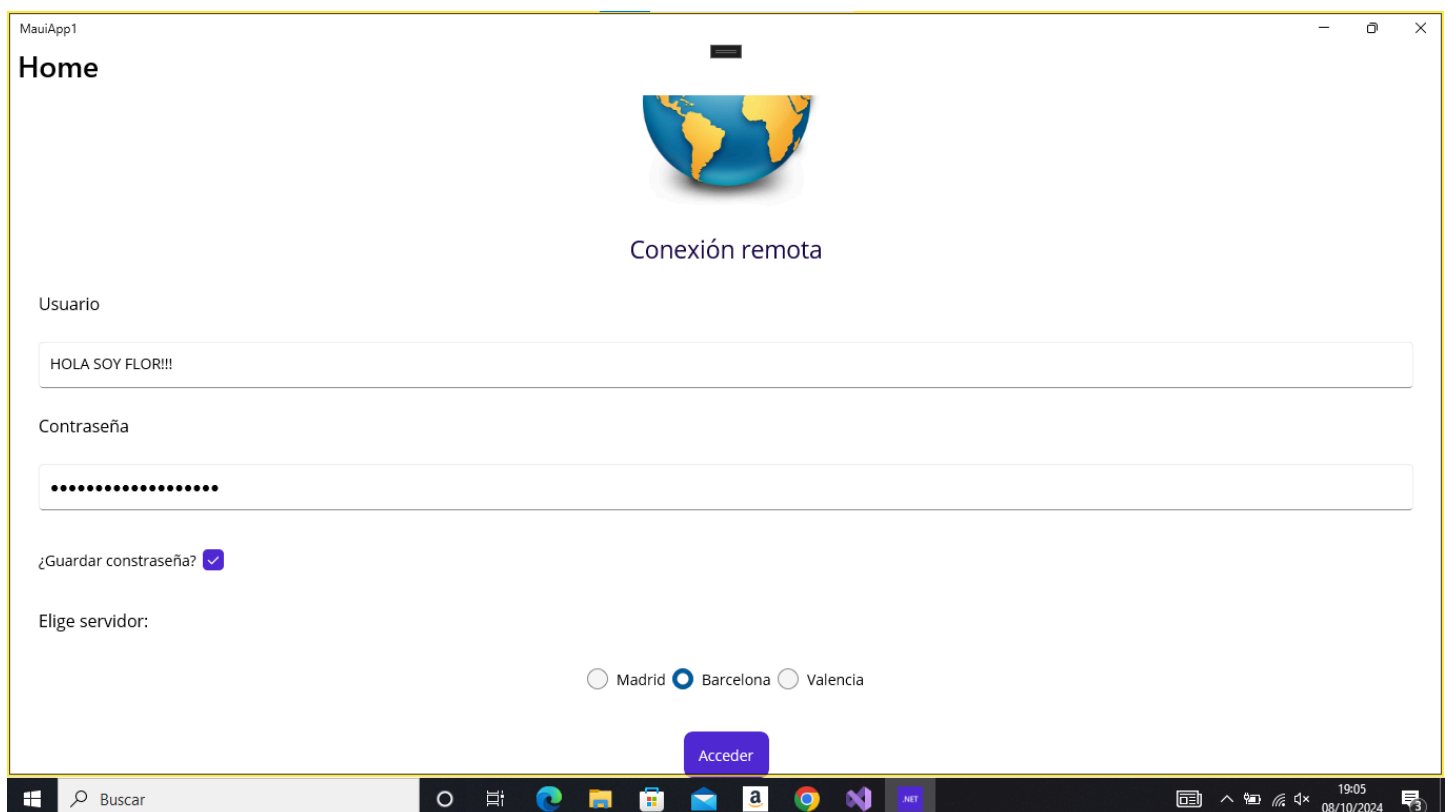
✓ No se encontraron problemas.

También saber que dentro de cada elemento se puede modificar a gusto de quien está realizando la interfaz.

PARÁMETROS

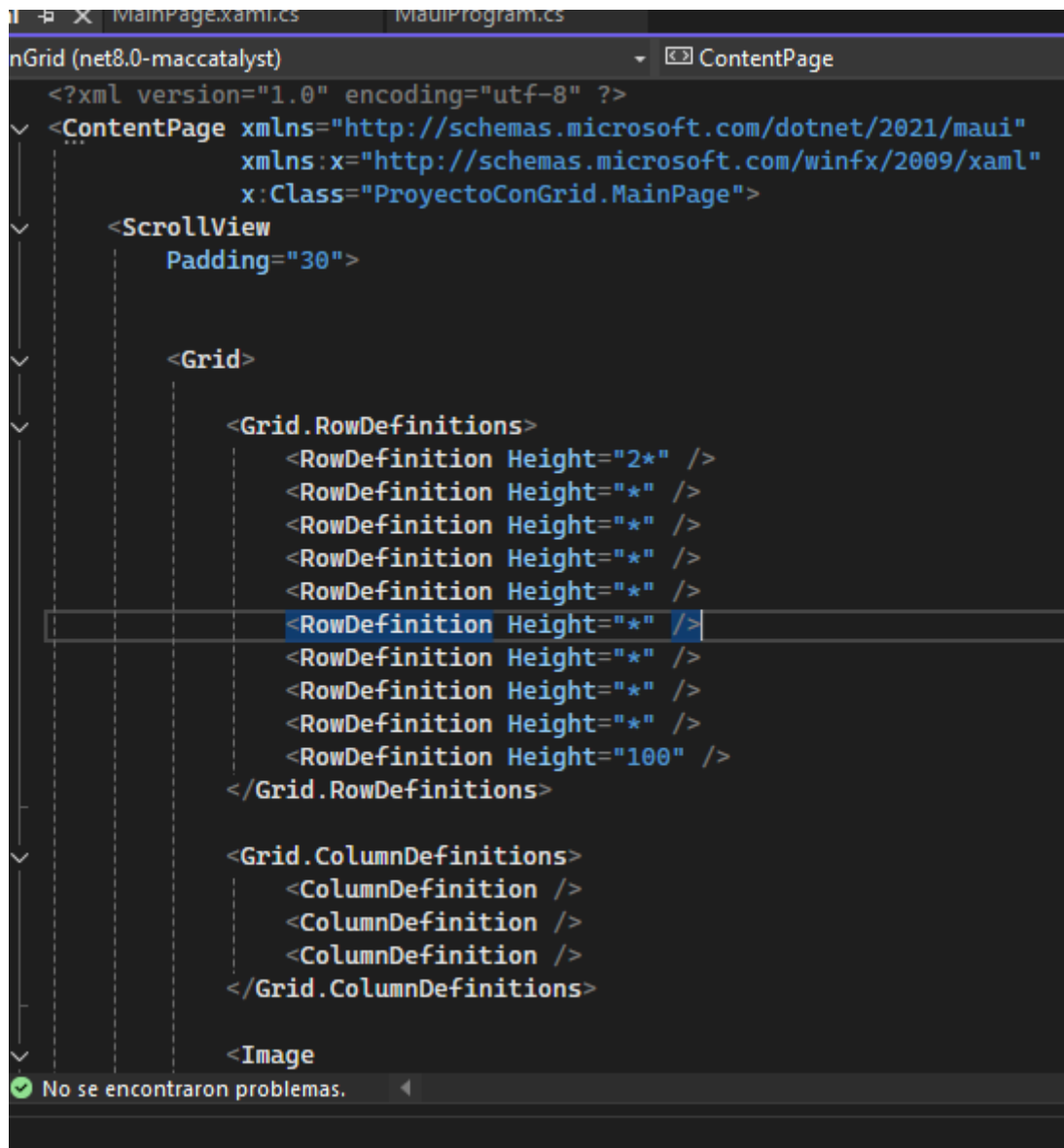
- **Text** = : Para escribir cualquier texto que queramos que aparezca en pantalla.
- **FontSize** = : Podemos decidir el tamaño del texto.
- **HorizontalOptions** = : En el eje de la x podemos decidir donde situar el elemento, ya sea a la izquierda, al centro o al final.
- **VerticalOptions** = : Lo mismo que el anterior pero en el eje y.
- **MaxLength** = : Se determina el número de caracteres a teclear por el usuario.
- **Placeholder** = : Se utiliza en campos como el "Entry" para dejar un mensaje en el cuadro de texto, ya sea indicando una instrucción de cómo rellenar la caja de texto o alguna otra cosa.
- **IsPassword** = : También se utiliza dentro del campo "Entry" para una mayor seguridad al teclear en el cuadro de texto, devolviendo puntos en lugar de letras.
- **IsChecked** = : Utilizado en campos como "CheckBox" o "RadioButton" para predeterminar la selección en alguna opción.
- **Content** = : Para darle una descripción, donde lo veremos por pantalla, como por ejemplo en los "RadioButton"
- **x:Name** = : Sirve para darle un identificativo a un elemento.
- **Padding** = : Gracias a este parámetro podemos establecer un espacio interno alrededor del contenido de un contenedor como sería el caso de un <StackLayout>

Con todos estos elementos, organizados debidamente en donde tocan, nuestra aplicación se vería de esta manera:



3.2 Apartado 2. Utilización de Grid

Al igual que en el apartado anterior, la etiqueta <Grid> sirve para implementar un contenedor y organizar la visualización del contenido introducido en el mismo. Como diferencia, este contenedor emplea una estructura de filas y columnas que debemos diseñar antes de decidir cómo meter los diferentes elementos en él.



```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
              xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
              x:Class="ProyectoConGrid.MainPage">
    <ScrollView
        Padding="30">
        <Grid>
            <Grid.RowDefinitions>
                <RowDefinition Height="2*" />
                <RowDefinition Height="*" />
                <RowDefinition Height="*" />
                <RowDefinition Height="*" />
                <RowDefinition Height="*" />
                <RowDefinition Height="*" />
                <RowDefinition Height="*" />
                <RowDefinition Height="*" />
                <RowDefinition Height="*" />
                <RowDefinition Height="100" />
            </Grid.RowDefinitions>
            <Grid.ColumnDefinitions>
                <ColumnDefinition />
                <ColumnDefinition />
                <ColumnDefinition />
            </Grid.ColumnDefinitions>
            <Image
```

Implementamos el contenedor dentro del ScrollView para que el contenido a la hora de que no podamos visualizarlo completamente en la pantalla no se nos apile uno encima de otro.

Tras esto lo que debemos hacer es definir la cantidad de filas y la cantidad de columnas que contendrá nuestra interfaz. En nuestro caso tendrá diez filas, numeradas del 0 al 9 y tres columnas, numeradas del 0 al 2. En este paso podemos definir la altura de las filas, siendo la primera más alta ya que tendrá la imagen de la tierra.

```

MainPage.xaml.cs MauiProgram.cs
ContentPage
</Grid.ColumnDefinitions>

<Image
  Source="tierra.png"
  Grid.Row="0" Grid.ColumnSpan="3"
  HeightRequest="100"
  HorizontalOptions="Center"
  VerticalOptions="Center" />

<Label
  Text="Conexión remota"
  FontSize="32"
  HorizontalOptions="Center"
  Grid.Row="1" Grid.ColumnSpan="3"/>

<Label
  Text="Usuario"
  FontSize="16"
  Grid.Row="2" Grid.ColumnSpan="3" />

<Entry
  Placeholder="Introduce usuario"
  MaxLength="20"
  Grid.Row="3" Grid.ColumnSpan="3"/>

<Label
  Text="Contraseña"
  Grid.Row="4" Grid.ColumnSpan="3" />

<Entry

```

Lo único diferente es que agregamos 4 parámetros más que en el anterior contenedor no teníamos, los cuales son los indicativos de las filas y las columnas y de como actúan las mismas.

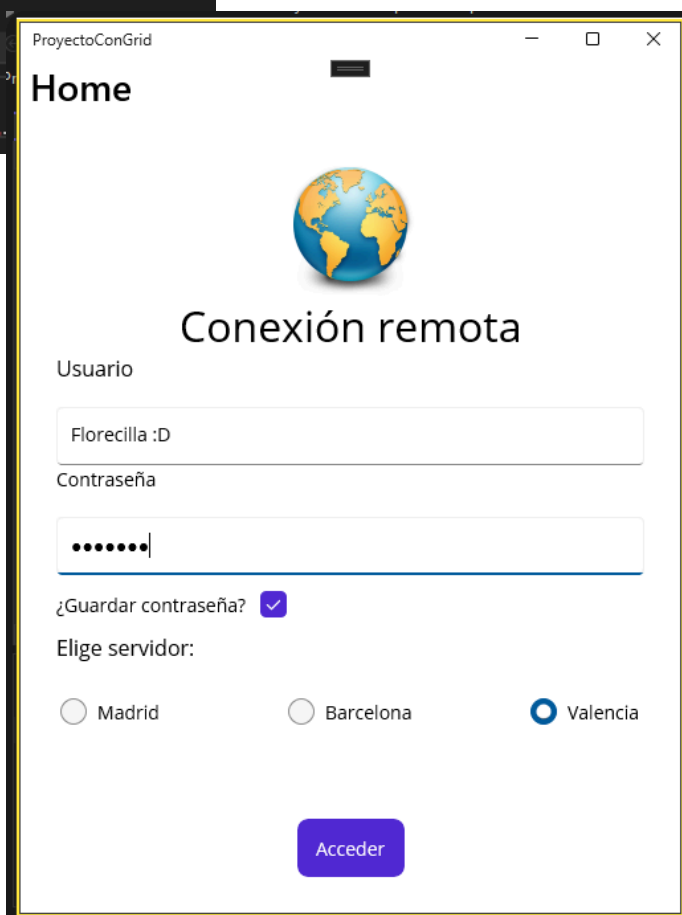
Con *Grid.Row* y *Grid.Column* podemos indicar en cual columna y en cuál elemento se encuentra nuestro elemento. Por ejemplo si queremos situar una etiqueta "Entry" en la cuarta fila y en la columna dos pondremos:

Grid.Row = "3" *Grid.Column* = "1" ya que vamos del 0 al 9 en filas, siendo la primera la fila 0 y de 0 a 2 en columnas, siendo la primera columna la número 0.

Para que un elemento utilice varias columnas o filas utilizaremos los parámetros: *Grid.RowSpan* = y *Grid.ColumnSpan* = seguido de la cantidad de filas o columnas que se necesitarán.

Por lo demás, las etiquetas siguen siendo las mismas.

Y así se vería nuestra aplicación utilizando el contenedor Grid.



4.Problemas

El único problema que quizás he encontrado y no es un problema casi, es que al no estar familiarizada con el contenedor grid me cuesta más a la hora de organizar los elementos dentro del mismo, pero por lo demás ha sido demasiado rápido e intuitivo.

5.Conclusión

Ahora, con los parámetros y las etiquetas claras podemos crear diseños de aplicaciones y organizarlos de diferentes maneras para poder tenerlo todo más “limpio” en la pantalla.

Eso sí, utilizaremos los contenedores Grid para proyectos más complejos y que se puedan ver correctamente en diferentes dispositivos.

