

# Tutorial: usa Grid y StackPanel para crear una aplicación meteorológica sencilla.

19/05/2017    Tiempo de lectura: 9 minutos

## En este artículo

Requisitos previos

Paso 1: Crear una aplicación en blanco

Paso 2: Definir una cuadrícula

Paso 3: Dar color a la cuadrícula

Paso 4: Organizar el contenido mediante elementos StackPanel

Paso 5: Agregar un icono de imagen

Artículos relacionados

Usa XAML para crear el diseño de una aplicación meteorológica sencilla con los elementos **Grid** y **StackPanel**. Con estas herramientas, puedes crear aplicaciones con un aspecto fantástico que funcionen en cualquier dispositivo que ejecute Windows10. Este tutorial dura entre 10 y 20 minutos.

**API importantes:** [Clase Grid](#), [Clase StackPanel](#)

## Requisitos previos

- Windows10 y Microsoft Visual Studio2015. Haz clic aquí para obtener información sobre cómo iniciar la configuración con Visual Studio.
- Conocimientos acerca de cómo crear una aplicación de "Hello World" básica mediante XAML y C#. Si aún no los tienes, haz clic aquí para aprender a crear una aplicación "Hello World".

## Paso 1: Crear una aplicación en blanco

1. En el menú de Visual Studio, selecciona **Archivo > Nuevo proyecto**.
2. En el panel izquierdo del cuadro de diálogo **Nuevo proyecto**, selecciona **Visual C# > Windows > Universal** o **Visual C++ > Windows > Universal**.
3. En el panel central, selecciona **Aplicación vacía**.
4. En el cuadro **Nombre**, escribe **WeatherPanel** y selecciona **Aceptar**.
5. Para ejecutar el programa, selecciona **Depurar > Iniciar depuración** en el menú o presiona F5.

## Paso 2: Definir una cuadrícula

En XAML, un elemento **Grid** (una cuadrícula) se compone de una serie de filas y columnas. Al especificar la fila y la columna de un elemento incluido en un elemento **Grid**, puedes colocar y separar otros elementos dentro de una interfaz de usuario. Las filas y las columnas se definen con los elementos **RowDefinition** y **ColumnDefinition**.

Para empezar a crear un diseño, abre **MainPage.xaml** mediante el uso del **Explorador de soluciones** y sustituye el elemento **Grid** generado automáticamente por este código.

```

<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="3*" />
    <ColumnDefinition Width="5*" />
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="2*" />
    <RowDefinition Height="*" />
  </Grid.RowDefinitions>
</Grid>

```

El nuevo elemento **Grid** crea un conjunto de 2 filas y columnas que define el diseño de la interfaz de la aplicación. La primera columna tiene un **Width**(ancho) de "3\*", mientras que el segundo tiene "5\*", lo que divide el espacio horizontal entre las 2 columnas en una proporción de 3:5. Del mismo modo, las 2 filas tienen **Height** (alto) de "2\*" y "\*" respectivamente, por lo tanto, el elemento **Grid** asigna el doble de espacio a la primera fila que a la segunda ("\*" equivale a "1\*"). Estas relaciones se mantienen incluso si se cambia el tamaño de la ventana o se cambia el dispositivo.

Para obtener información sobre otros métodos de definir el tamaño de las filas y columnas, consulta [Definir diseños de página con XAML](#).

Si ejecutas la aplicación ahora, no verás nada,excepto una página en blanco, porque ninguna de las áreas de **Grid** tienen ningún contenido. Para mostrar el elemento **Grid**, vamos a asignarle el color.

### Paso 3: Dar color a la cuadrícula

Para dar color al elemento **Grid**, agregamos tres elementos **Border**, cada uno con un color de fondo diferente. Cada uno de ellos también se asigna a una fila y columna en el elemento primario **Grid** mediante el uso de los atributos **Grid.Row** y **Grid.Column**. De forma predeterminada, los valores de estos atributos son 0, por lo que no tienes que asignarlos al primer elemento **Border**. Agrega el siguiente código al elemento **Grid** después de las definiciones de fila y columna.

XML

```

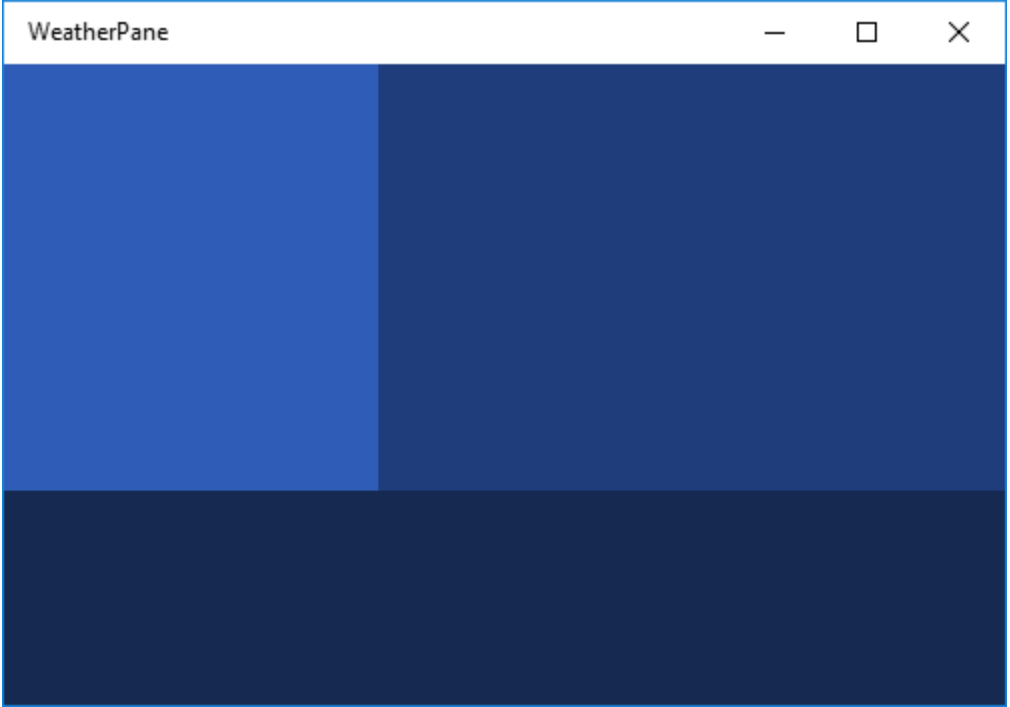
<Border Background="#2f5cb6" />
<Border Grid.Column ="1" Background="#1f3d7a" />
<Border Grid.Row="1" Grid.ColumnSpan="2" Background="#152951" />

```

Copiar

Ten en cuenta que para el tercer elemento **Border** usamos un atributo adicional, **Grid.ColumnSpan**, lo que hace que este **Border** ocupe ambas columnas en la fila inferior. Puedes usar **Grid.RowSpan** de la misma forma y, juntos, estos atributos te permiten extender un elemento sobre cualquier número de filas y columnas. La esquina superior izquierda de dicha expansión es siempre los atributos **Grid.Column** y **Grid.Row** especificados en los atributos del elemento.

Si ejecutas la aplicación, el resultado será similar al siguiente.



## Paso 4: Organizar el contenido mediante elementos StackPanel

**StackPanel** es el segundo elemento de la interfaz de usuario que usaremos para crear nuestra aplicación meteorológica. El elemento **StackPanel** es una parte fundamental de muchos diseños de aplicación básicos, que permite apilar los elementos vertical u horizontalmente.

En el siguiente código, crearemos dos elementos **StackPanel** y los rellenaremos con tres **TextBlocks**. Agrega estos elementos **StackPanel** al elemento **Grid** debajo de los elementos **Border** del paso 3. Esto hace que los elementos **TextBlock** se representen en la parte superior de elemento **Grid** coloreado que hemos creado anteriormente.

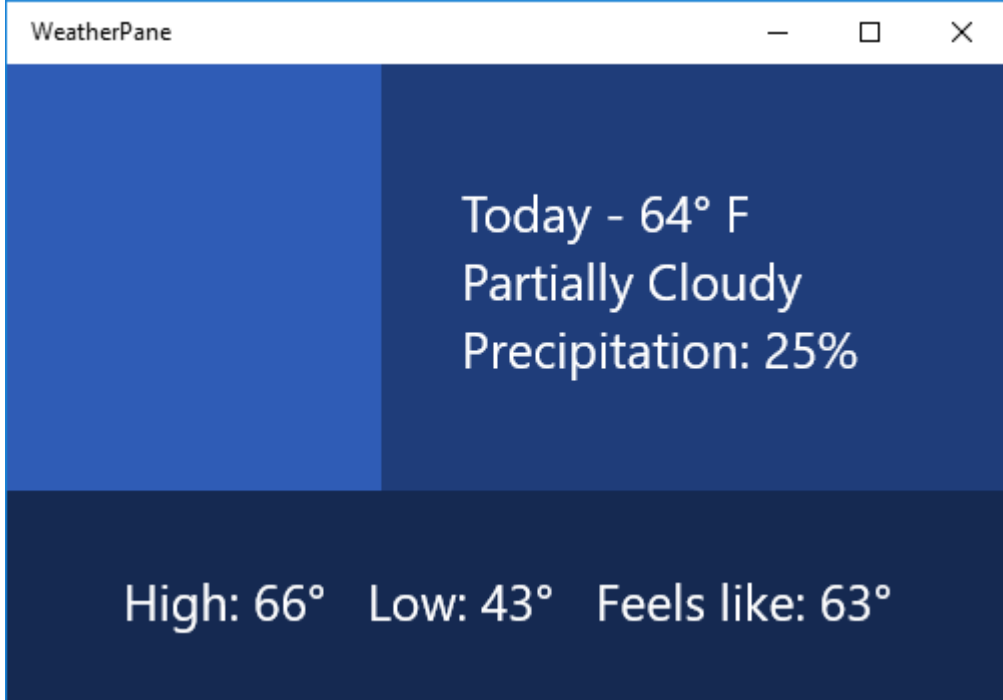
XML

```
<StackPanel Grid.Column="1" Margin="40,0,0,0" VerticalAlignment="Center">
  <TextBlock Foreground="White" FontSize="25" Text="Today - 64° F"/>
  <TextBlock Foreground="White" FontSize="25" Text="Partially Cloudy"/>
  <TextBlock Foreground="White" FontSize="25" Text="Precipitation: 25%"/>
</StackPanel>
<StackPanel Grid.Row="1" Grid.ColumnSpan="2" Orientation="Horizontal"
  HorizontalAlignment="Center" VerticalAlignment="Center">
  <TextBlock Foreground="White" FontSize="25" Text="High: 66°" Margin="0,0,20,0"/>
  <TextBlock Foreground="White" FontSize="25" Text="Low: 43°" Margin="0,0,20,0"/>
  <TextBlock Foreground="White" FontSize="25" Text="Feels like: 63°"/>
</StackPanel>
```

Copiar

En el primer **Stackpanel**, cada **TextBlock** se apila verticalmente debajo del siguiente. Este es el comportamiento predeterminado de un **StackPanel**, así que no necesitamos establecer el atributo **Orientation**. En el segundo **StackPanel**, queremos que los elementos secundarios se apilen horizontalmente de izquierda a derecha, por lo que establecemos el atributo **Orientation** en "Horizontal". También debemos establecer el atributo **Grid.ColumnSpan** en "2", para que el texto se centre en el elemento **Border** inferior.

Si ejecutas la aplicación ahora, verás algo parecido a esto.



## Paso 5: Agregar un icono de imagen

Por último, vamos a rellenar la sección vacía de nuestro elemento **Grid** con una imagen que represente el tiempo de hoy, algo que dice "parcialmente nuboso".

Descarga la siguiente imagen y guárdala como un archivo PNG denominado "partially-cloudy".



En el **Explorador de soluciones**, haz clic en la carpeta **Assets** y selecciona **Agregar -> Elemento existente....** Busca partially-cloudy.png en el explorador que se muestra, selecciónalo y haz clic en **Agregar**.

A continuación, en **MainPage.xaml**, agrega el siguiente elemento **Image** debajo de los elementos StackPanel del paso 4.

XML

Copiar

```
<Image Margin="20" Source="Assets/partially-cloudy.png"/>
```

Como queremos el elemento Image en la primera fila y la primera columna, no necesitamos establecer sus atributos **Grid.Row** o **Grid.Column**, por lo que podemos dejarlos en su valor predeterminado de "0".

Y eso es todo. Has creado correctamente el diseño de una aplicación meteorológica sencilla. Si presionas **F5** y ejecutas la aplicación, deberías ver algo parecido a esto:



Today - 64° F  
Partially Cloudy  
Precipitation: 25%

High: 66° Low: 43° Feels like: 63°

Si lo deseas, experimenta con el diseño anterior y explora diferentes maneras en las que podrías representar los datos meteorológicos.

## Artículos relacionados

Para obtener una introducción al diseño de diseños para aplicaciones para UWP, consulta [Introducción al diseño de aplicaciones para UWP](#).

Para obtener información sobre cómo crear diseños dinámicos que se adapten a diferentes tamaños de pantalla, consulta [Definir diseños de página con XAML](#).