

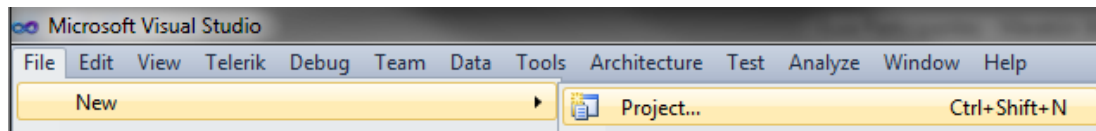
# Hands on Lab: Construyendo el lector RSS desde Visual Studio y Expression Blend

## Etapa 2

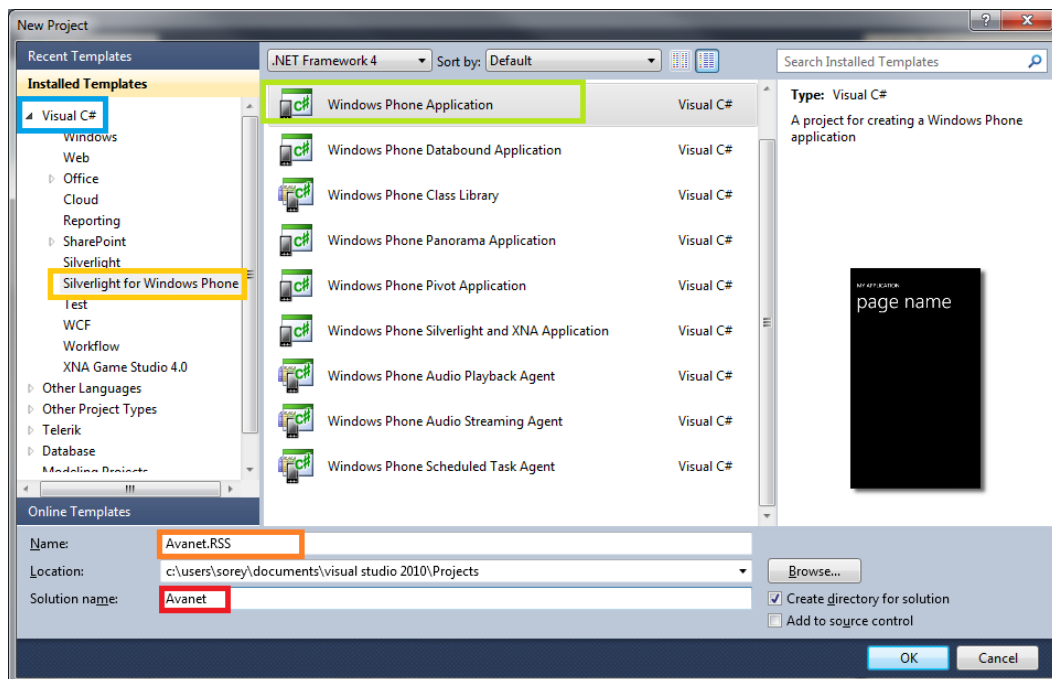
El objetivo de este Hands On Lab es crear una aplicación lectora de RSS, probarla con el Marketplace Test Kit y publicarla en el Marketplace.

### *Configurando la base del proyecto*

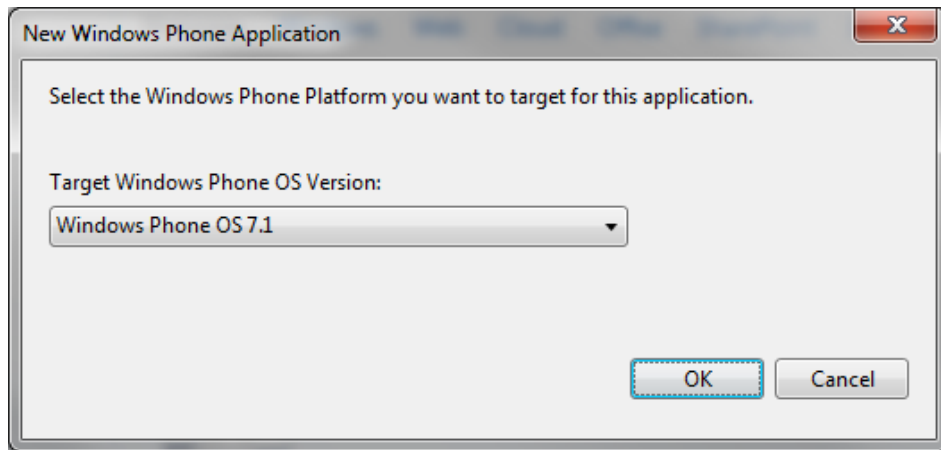
1. Cree un nuevo proyecto



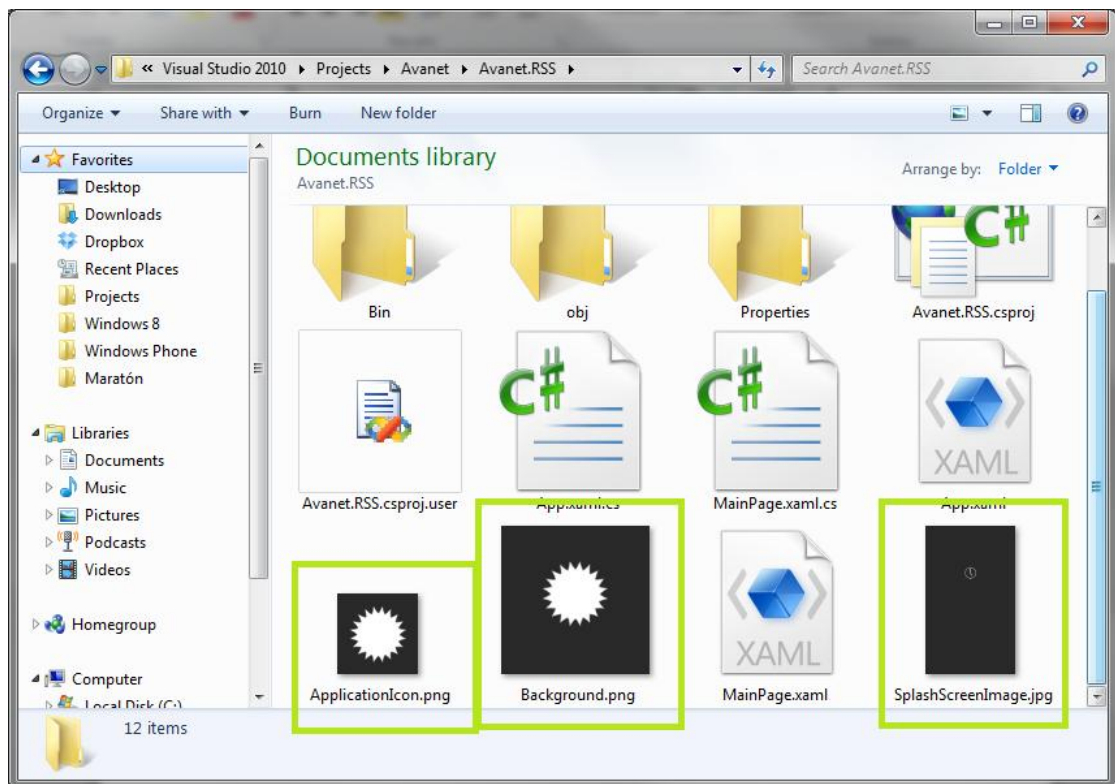
2. Seleccione una plantilla en blanco y elija su lenguaje favorito. Recuerde: Nombrar correctamente el proyecto y la solución.



3. Seleccione el sistema operativo como 7.1



4. Diríjase a la carpeta donde creó su proyecto y remplace las imágenes por las que diseñó.



5. Vuelva al Visual Studio, de clic derecho sobre el nombre del proyecto y seleccione la opción **Propiedades**, y configure allí el título que se verá en la lista general y el título del Tile o que aparece en la parte frontal en la parte inferior de la imagen Background.PNG

Application c:\users\sorey\documents\visual studio 2010\Projects\Avanet\Avanet.RSS\Contexto.cs

Configuration: N/A Platform: N/A

Assembly name: Avanet.RSS Default namespace: Avanet.RSS

Startup object: Avanet.RSS.App Assembly Information...

Xap file name: Avanet.RSS.xap Target Windows Phone OS Version: Windows Phone OS 7.1

Deployment options

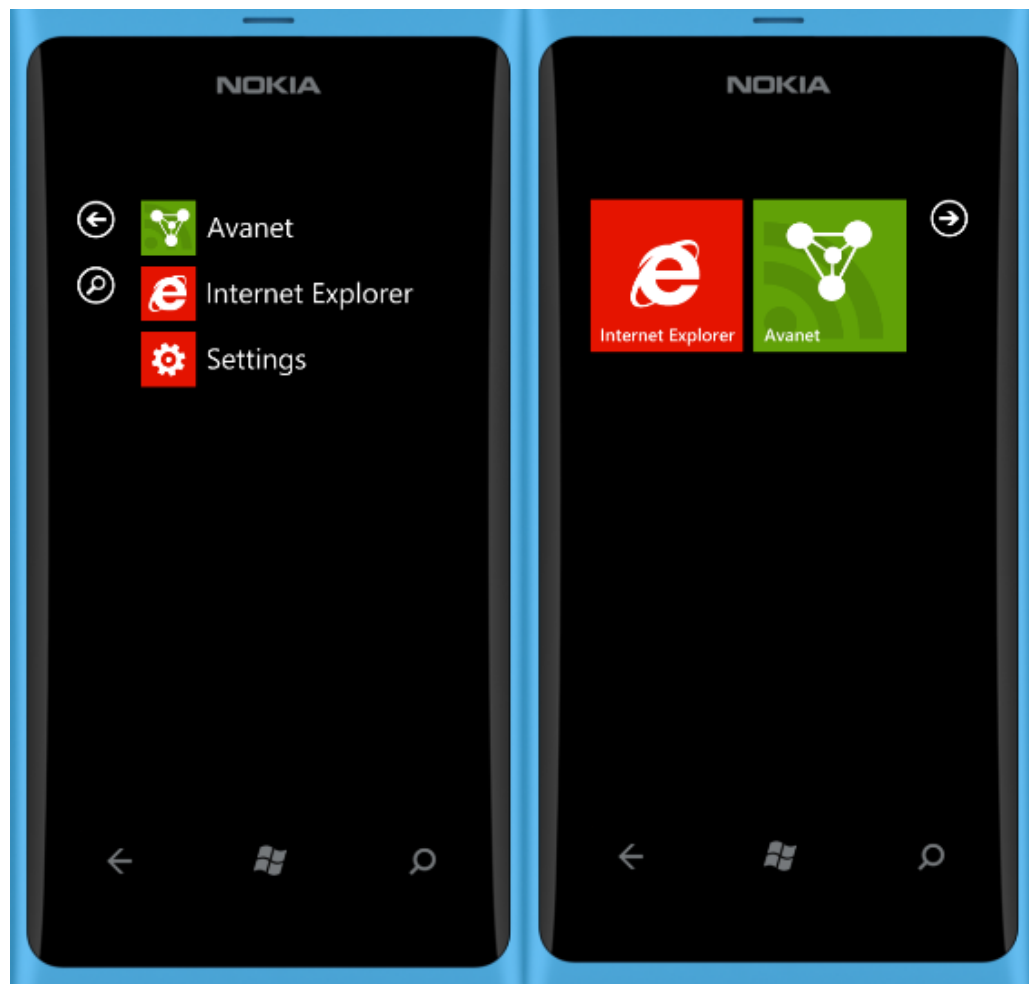
Title: Avanet

Icon: ApplicationIcon.png

Tile options

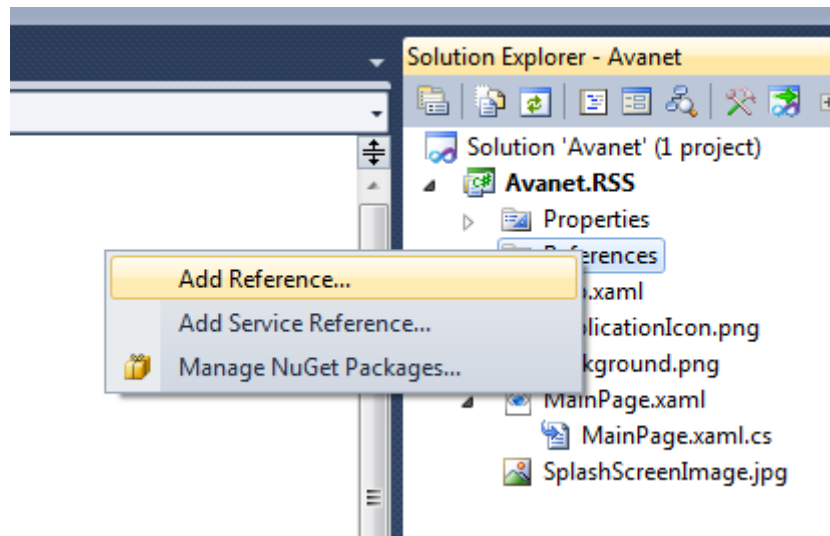
Title: Avanet

Background image: Background.png

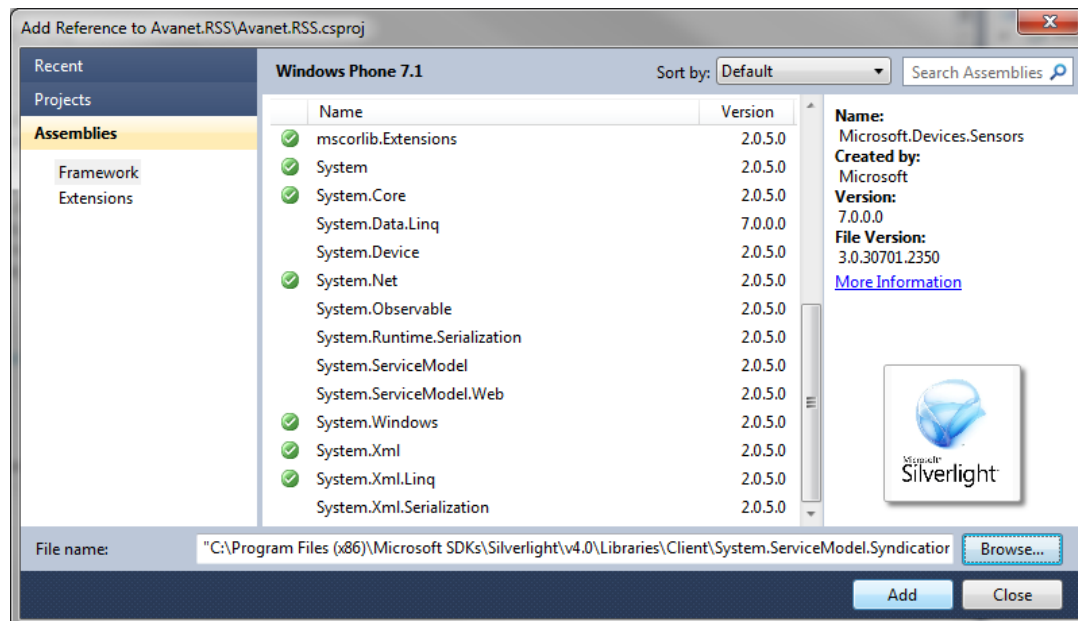


## Configurando los enlaces

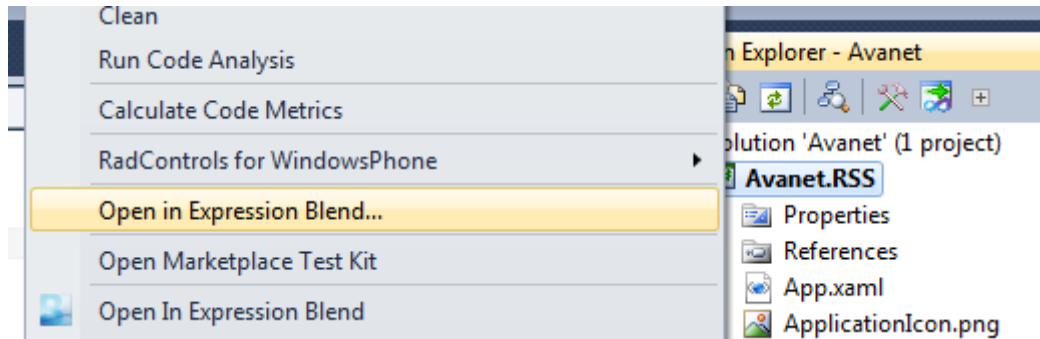
1. Añada una nueva referencia al proyecto, de clic derecho sobre la carpeta de referencias



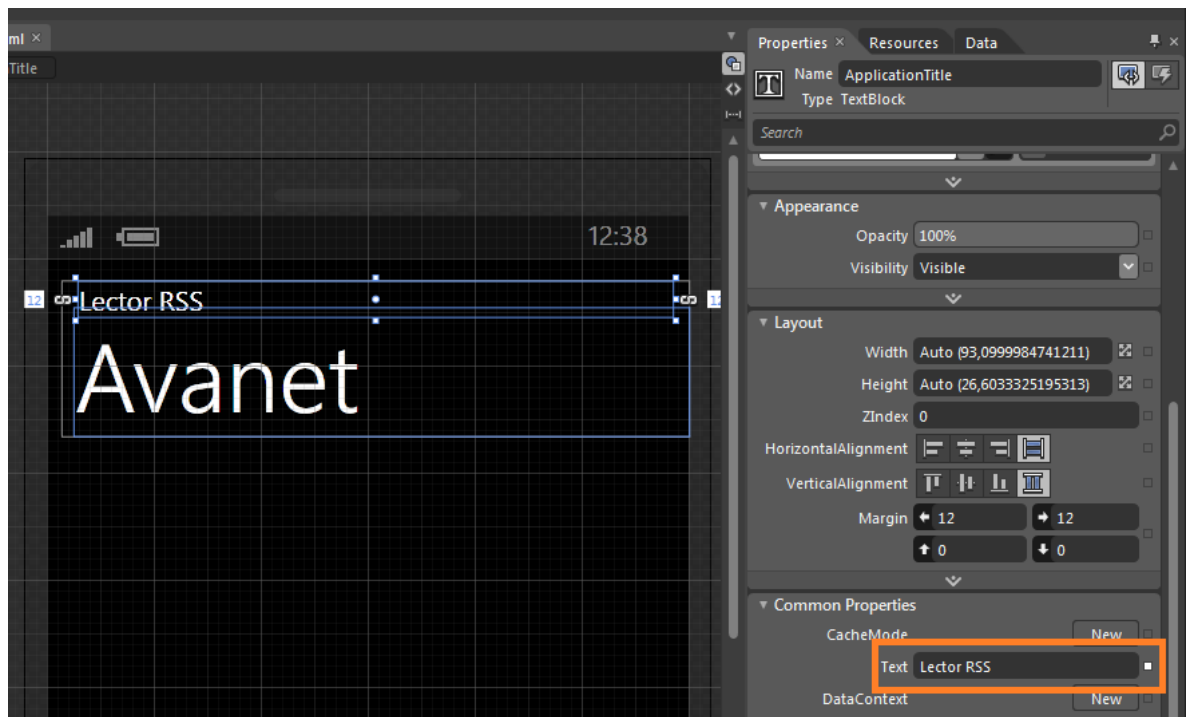
2. Seleccione el botón explorar y busque en su máquina la ruta del ensamblado **C:\Program Files (x86)\Microsoft SDKs\Silverlight\v4.0\Libraries\Client\System.ServiceModel.Syndication.dll**, presione el botón **Adicionar** y luego **Cerrar**.



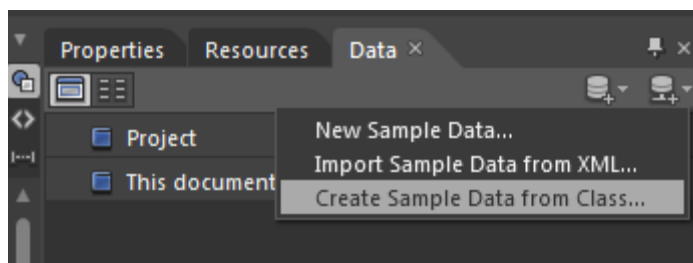
3. Guarde los cambios y **abra el proyecto en Expression Blend** dando clic sobre el nombre del proyecto.



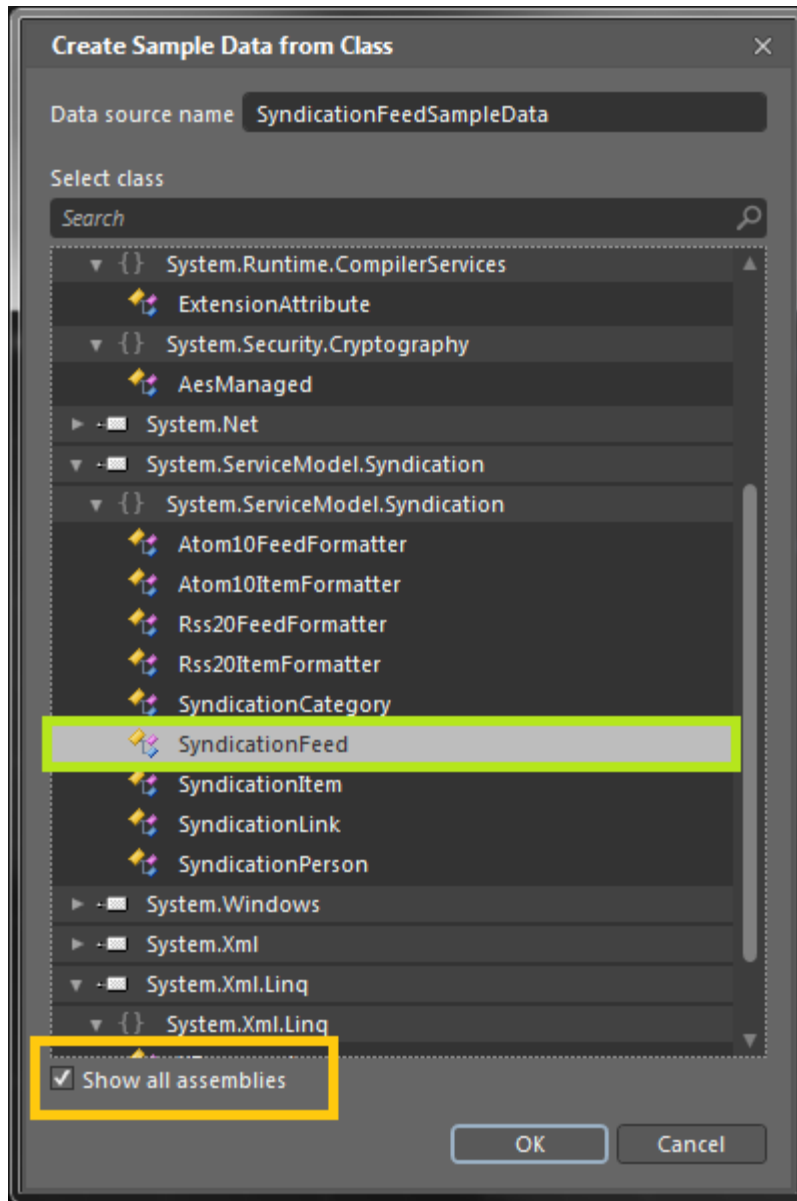
4. Cambie los títulos de su página desde Expression Blend seleccionando cada elemento y cambiando la propiedad **Text**.



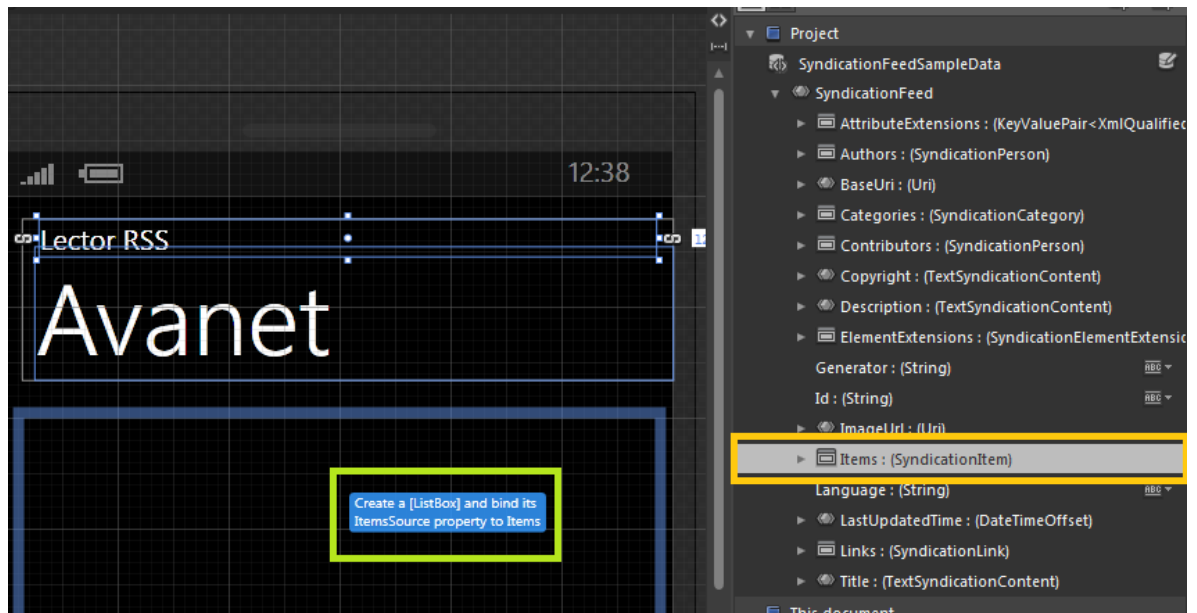
5. En la pestaña **Data** en la parte derecha superior cree un nuevo **Sample Data** a partir de una clase.



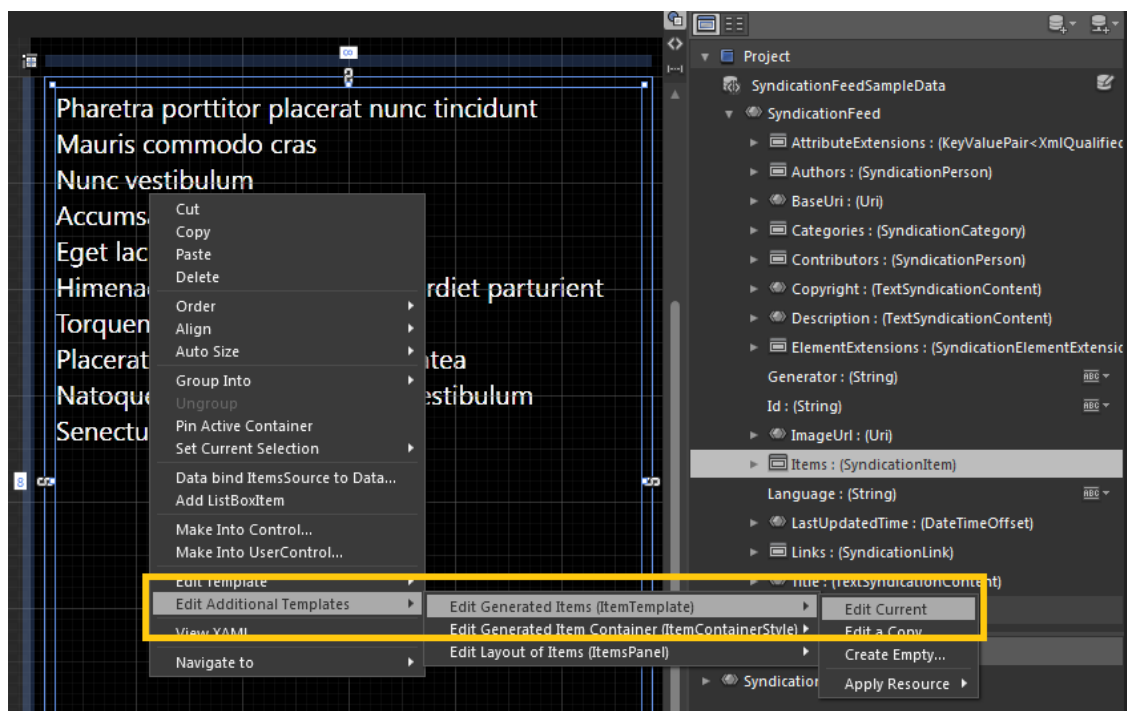
6. Seleccione **ver todos los ensamblados** y busque **SyndicationFeed** desde la nueva referencia que añadió.



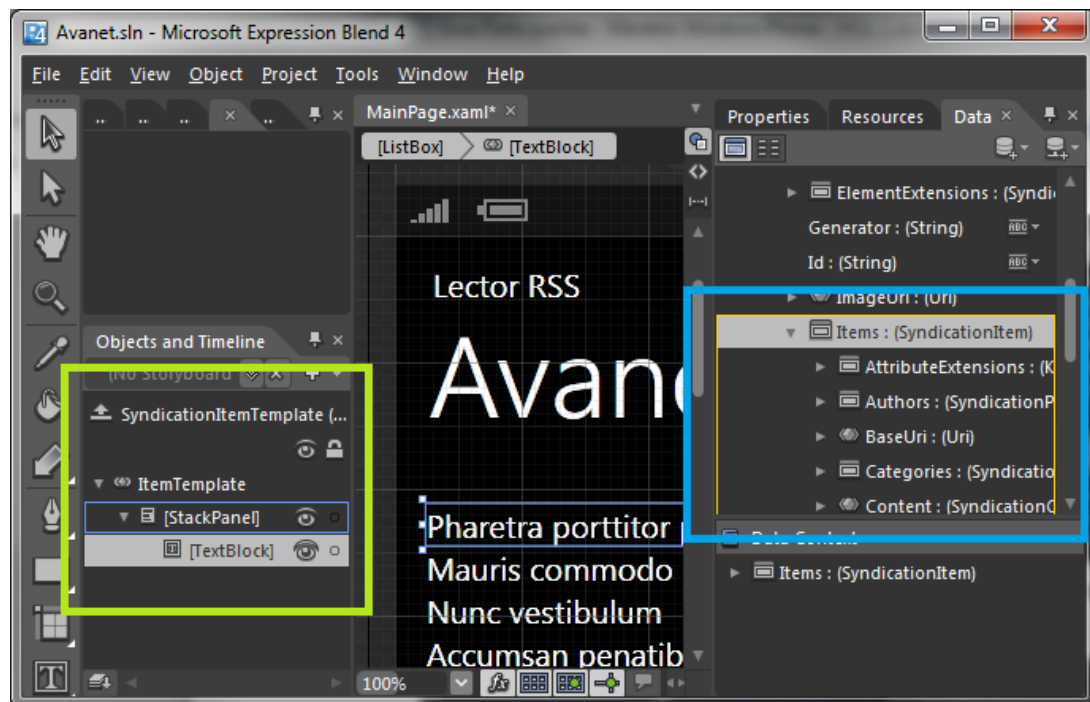
7. Cuando carguen todos los ítems en la sección proyecto justo en la parte inferior, busque la propiedad **Items** y con clic sostenido arrastre y suelte sobre el contenido central de su página. Verifique que el mensaje le indica que se creará un listbox enlazado a los Items, si ya tiene un listbox verifique que el mensaje indica que se enlazarán a la lista.



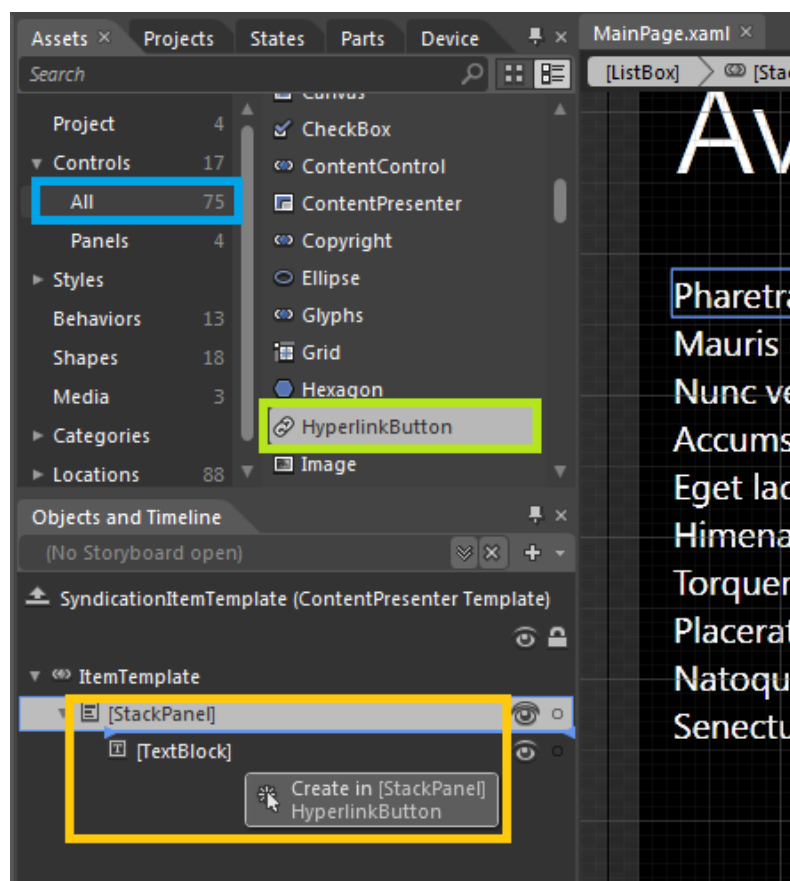
8. Se creará una lista automáticamente, ordene las márgenes de la lista de clic derecho para editar **la plantilla de los ítems**



9. En la parte izquierda de Expression Blend encontrará la jerarquía de elementos que conforman **cada uno de los ítems**, esta plantilla puede modificarse adicionando más campos o cambiando el estilo y propiedades de los existentes.

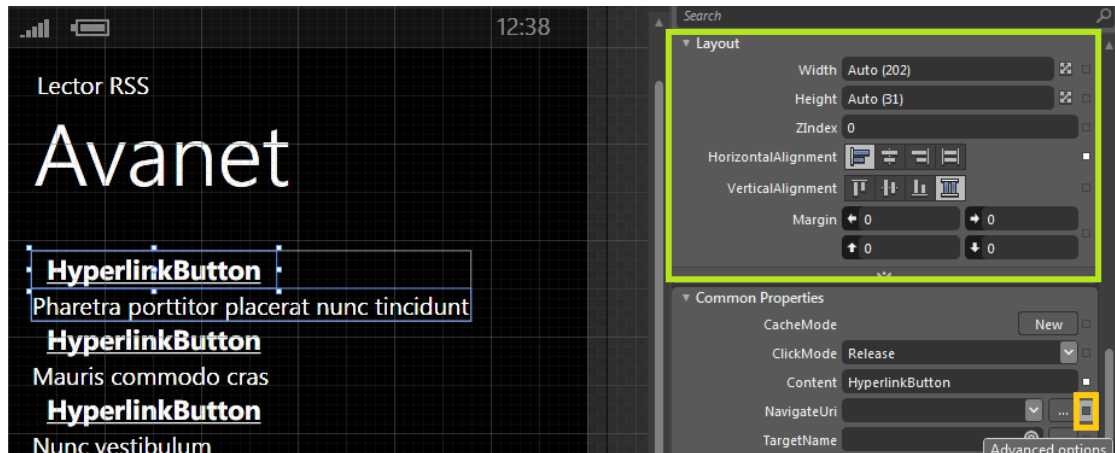


10. Añada un **HiperlinkButton** a la jerarquía que hace parte de la plantilla del ítem, arrastrando y soltando en el lugar adecuado desde la lista de controles.

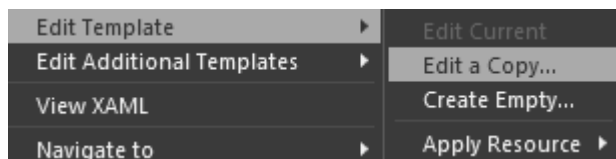




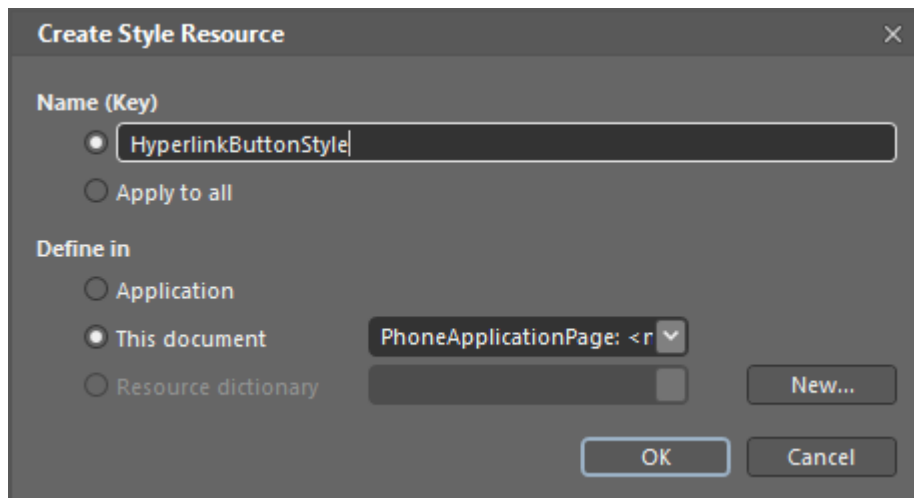
11. Tenga cuidado con la alineación, recuerde que la estética es uno de los criterios para ser aceptado en el Marketplace.



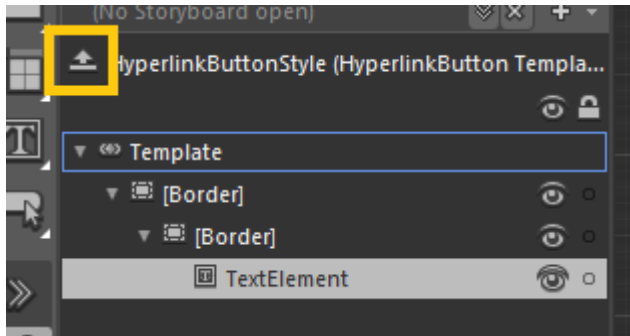
12. Para editar el estilo del **HiperlinkButton** puede dar clic derecho sobre el control y buscar la **opción editar plantilla y editar una copia**.



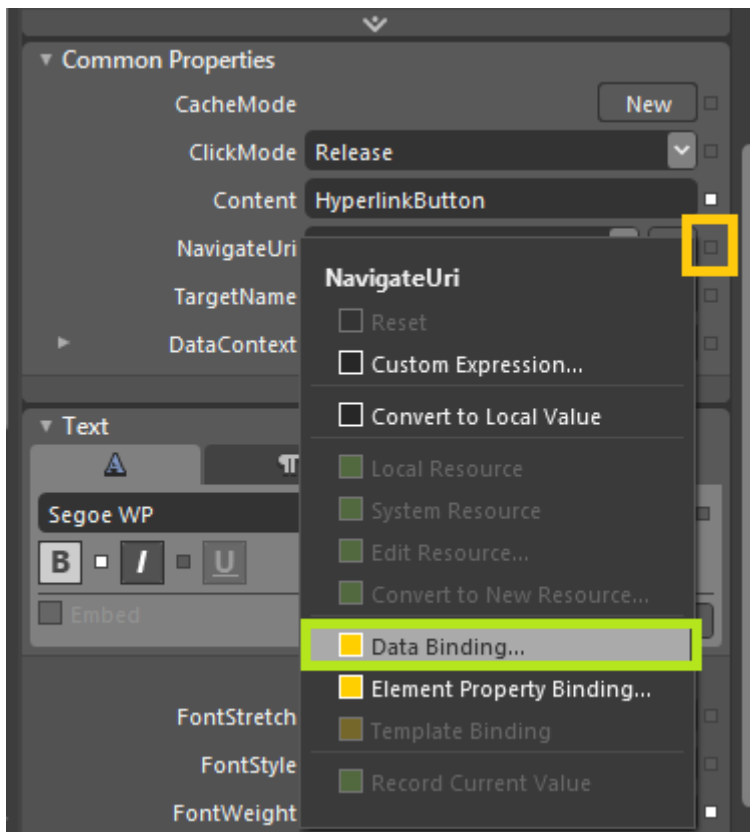
13. Se le pedirá asignar un nombre y decidir si creará el estilo en esta página o en el contexto de la aplicación, también se le permitirá crear un diccionario de recursos.



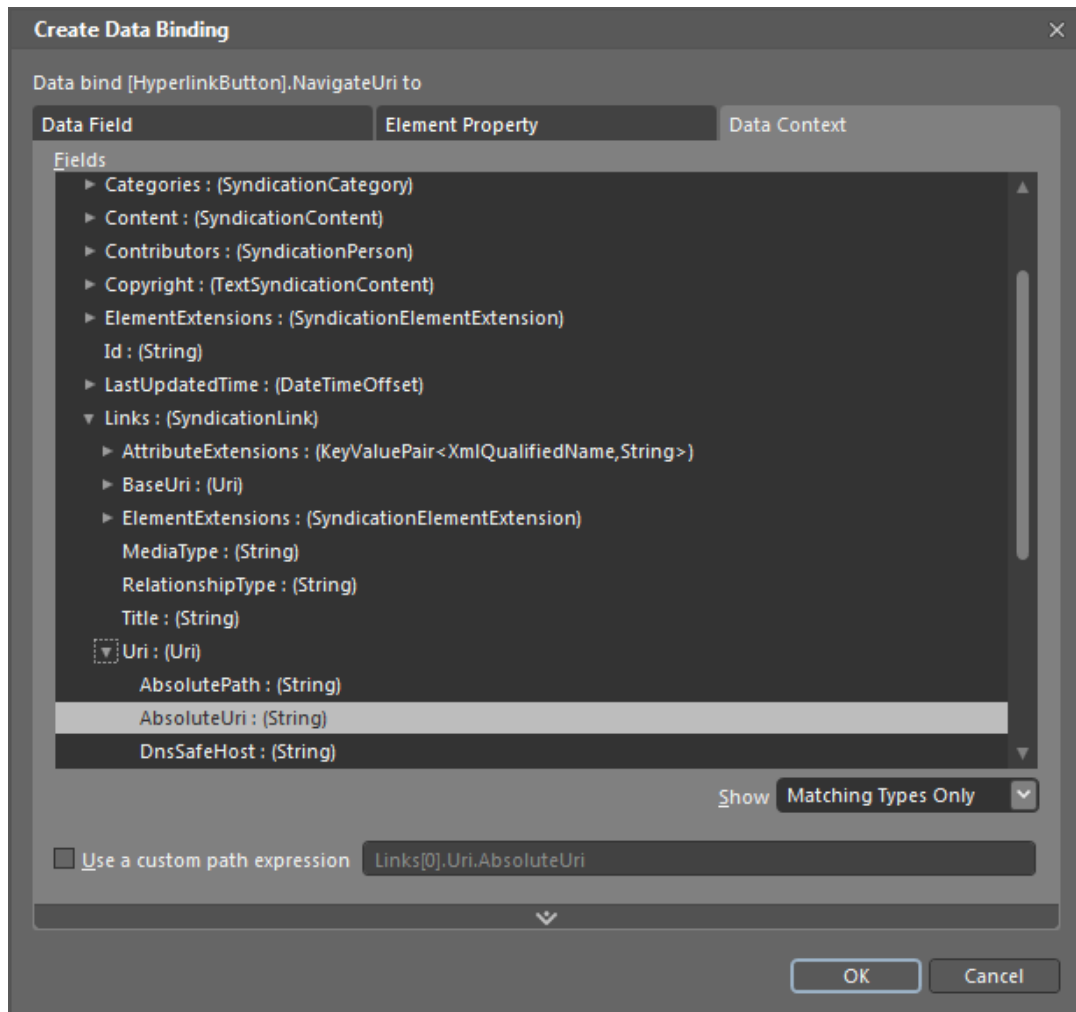
14. A continuación verá que se muestra la jerarquía de elementos que existen dentro de un **HiperlinkButton** y usted puede modificarla como desee. Cuando termine podrá regresar a **continuar editando la plantilla de la lista de ítems**.



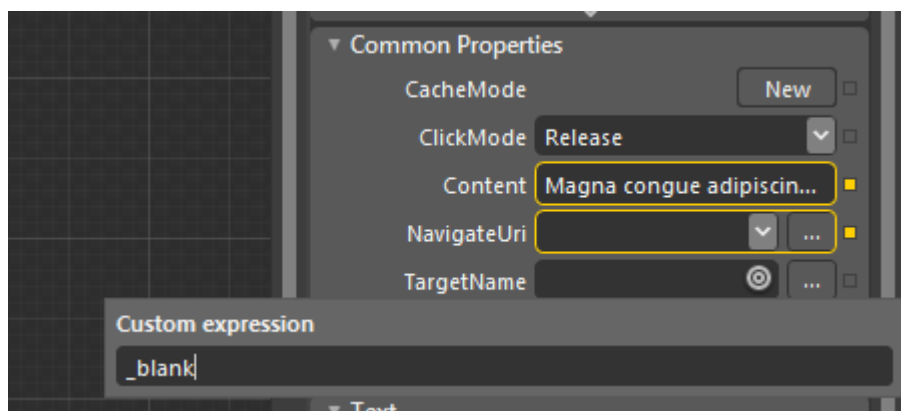
15. Seleccione el **HiperlinkButton** en la **plantilla de la lista de ítems** y en la pestaña de **propiedades** a la derecha de la pantalla busque la propiedad **NavigateUri**



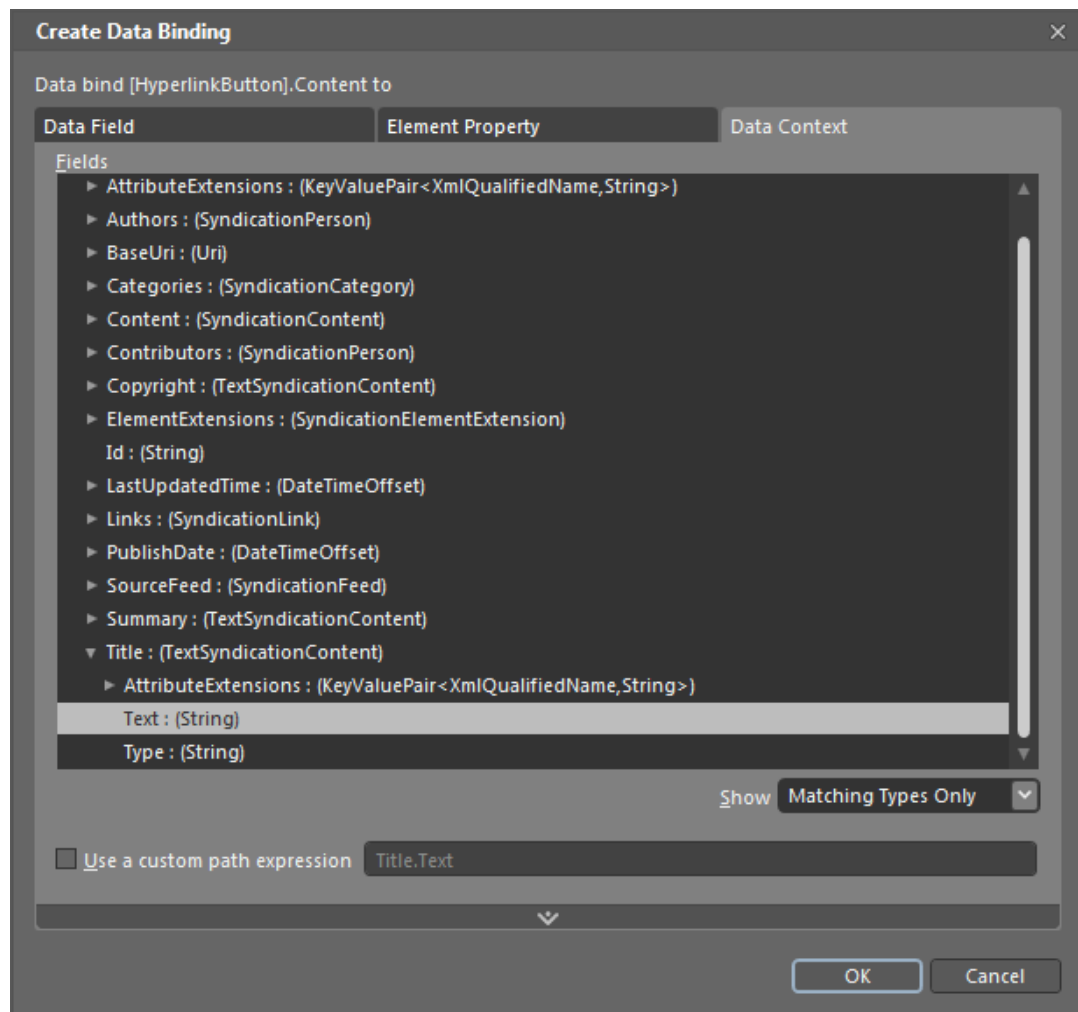
16. Como la lista está ya enlazada a los Ítems se presentarán todas las propiedades disponibles para un ítem. Ahí debe buscar la que le devuelve el link necesario para navegar al origen del ítem, es decir a la página que relaciona el RSS. Guíese de la siguiente imagen para saber cual es el campo, ya que hay varios links disponibles en la clase.



17. Para poder navegar desde el **HiperlinkButton** a una página externa es necesario establecer la propiedad **TargetName** con la expresión **\_blank**

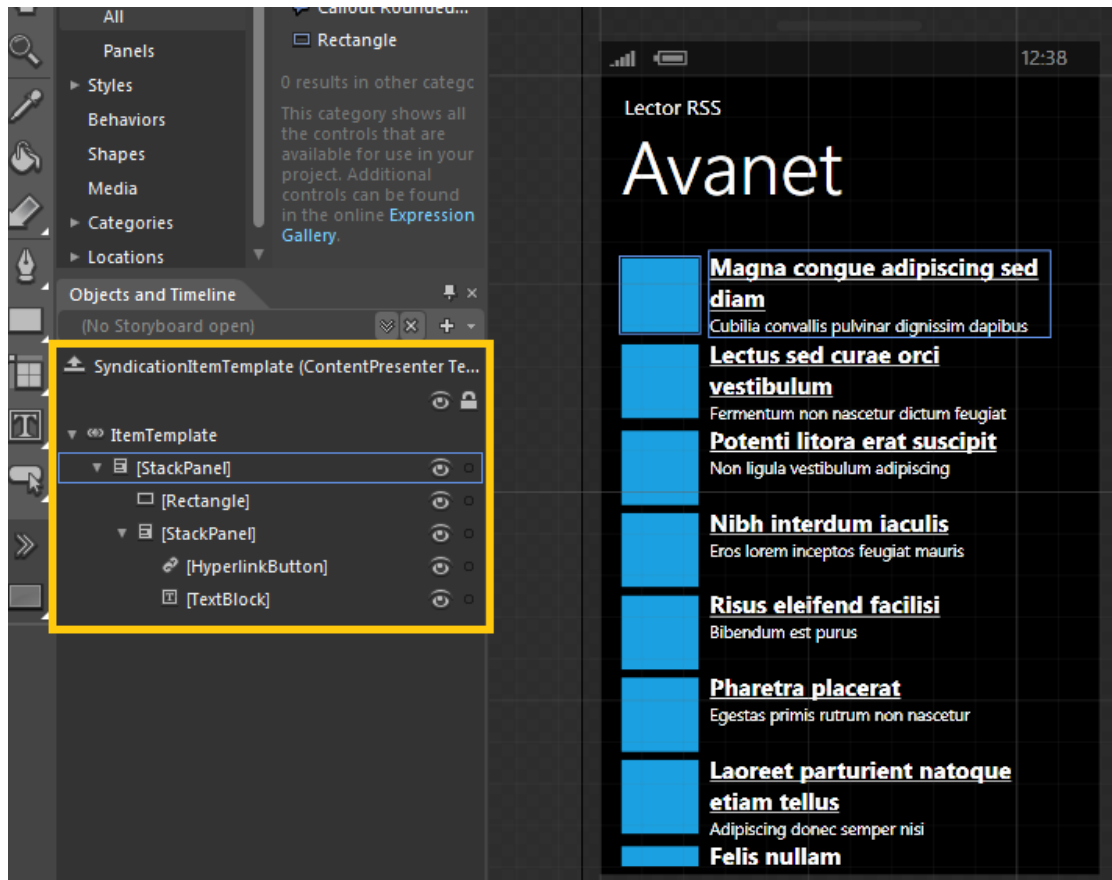


18. Enlace el campo de texto en la propiedad **content**, cree un enlace con la propiedad **Text** del campo **Title**, tal como lo hizo antes.

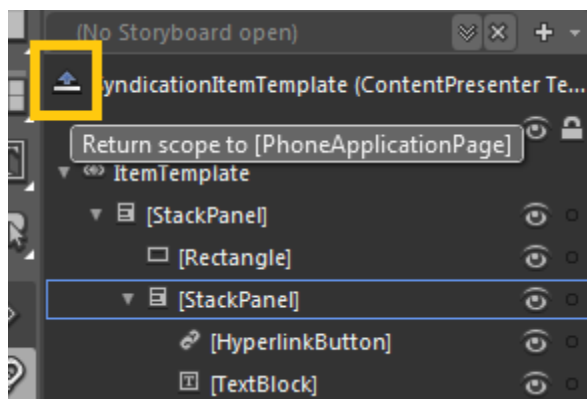


19. En este punto usted puede trabajar en el estilo de su lista añadiendo más controles o campos que conozca desde la lista de enlaces. En los siguientes pasos verá como el ejemplo se encuentra personalizado. La personalización es un plus para su prueba.

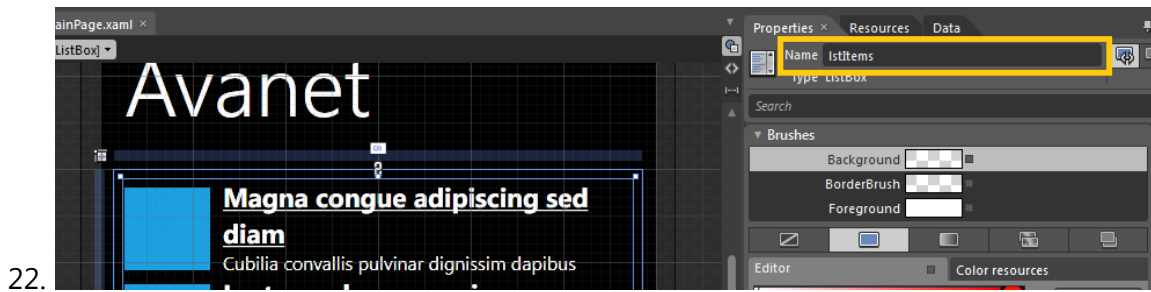
Observe en el ejemplo la jerarquía de elementos si necesita ayuda con los contenedores busque un entrenador.



20. Cuando termine puede salir de la edición de la plantilla usando la flecha que se ubica al lado del nombre de la plantilla, tal como se indica en la siguiente imagen.

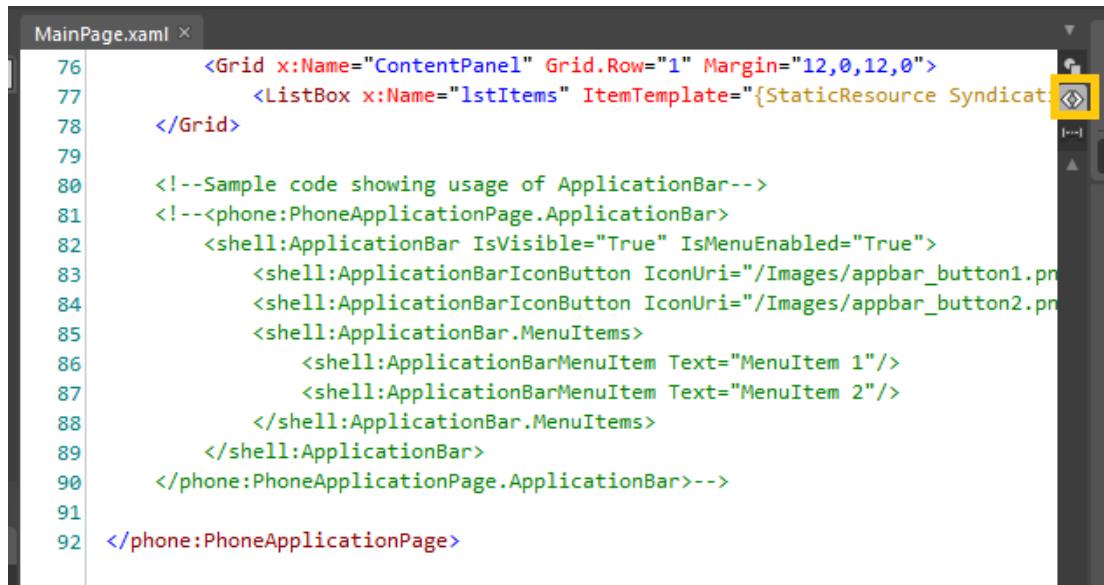


21. Antes de continuar establezca un nombre a la lista de ítems para poder referenciarla en el código más adelante.

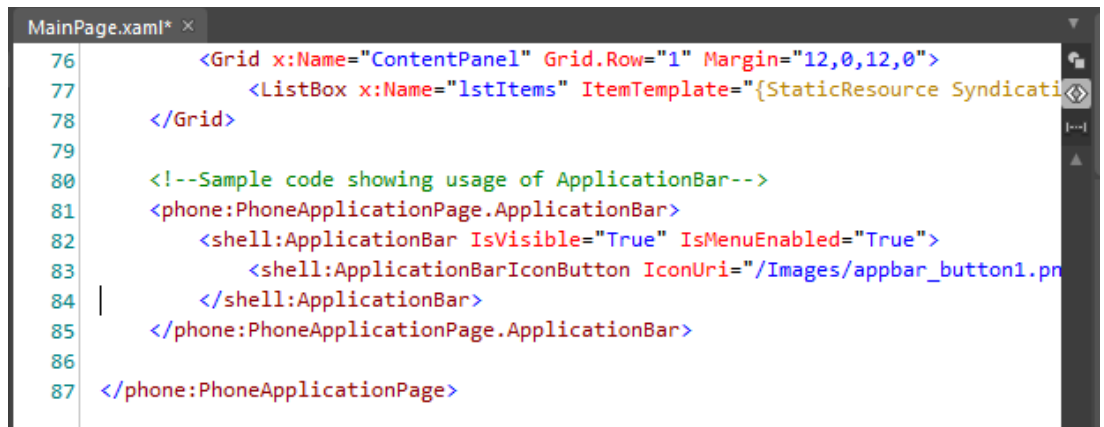


### Configurando la barra de aplicaciones

1. Ahora navegue al XAML o marcado de la página usando el botón que se indica en la imagen y busque el código que ve en la parte inferior del archivo.



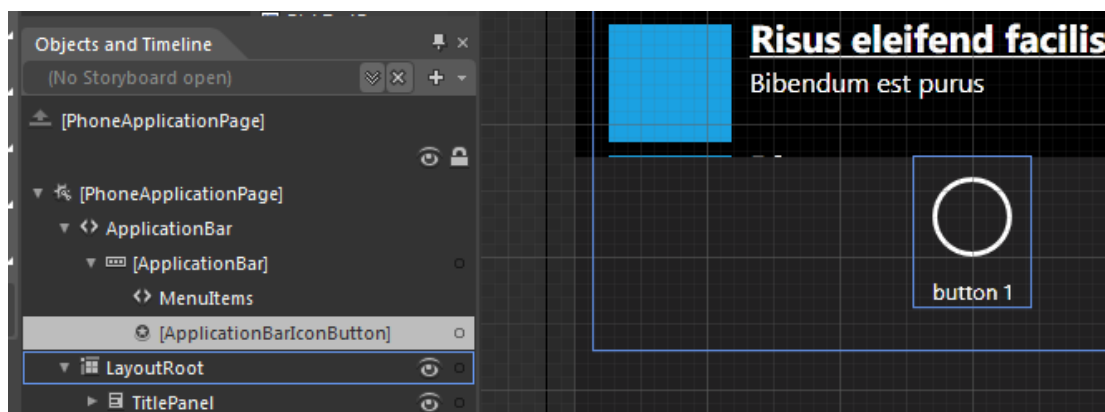
2. Elimine los comentarios y los ítems de menú y botón sobrante, deje solo una opción.



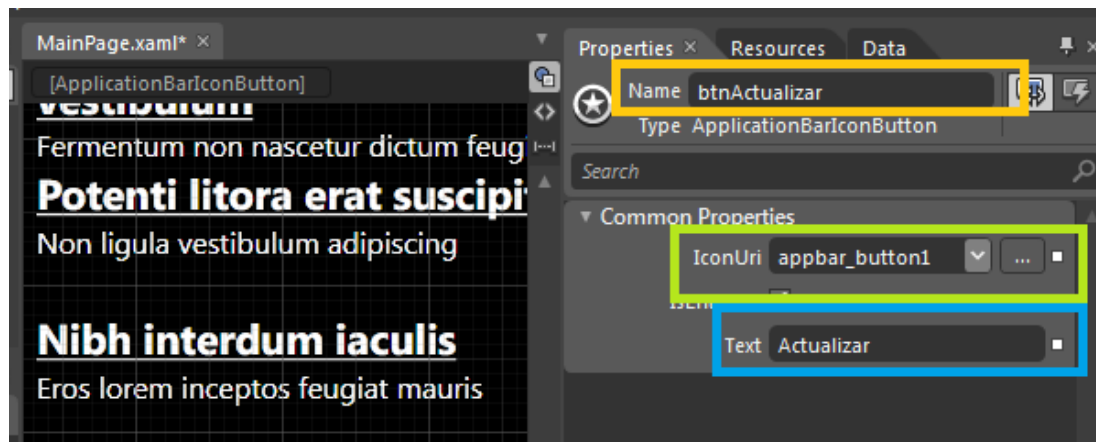
3. Regrese a la vista de diseño



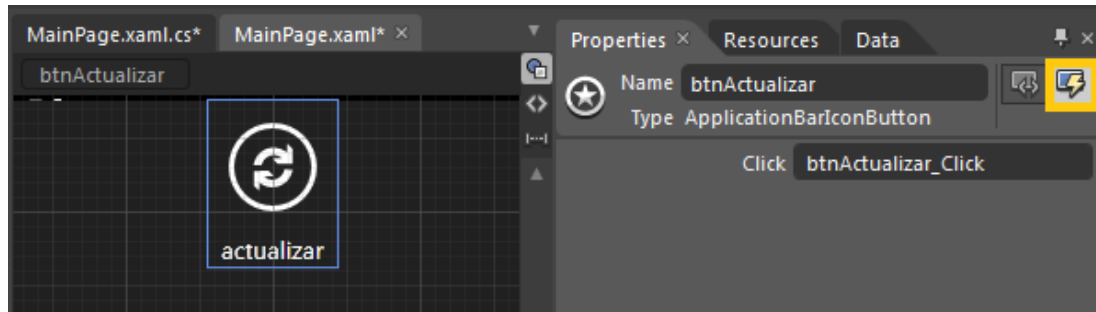
4. Observe en la parte derecha donde se muestra la jerarquia de elementos como se a activado el **Application Bar** los elementos y en la vista de diseño en la parte inferior apareció un botón-



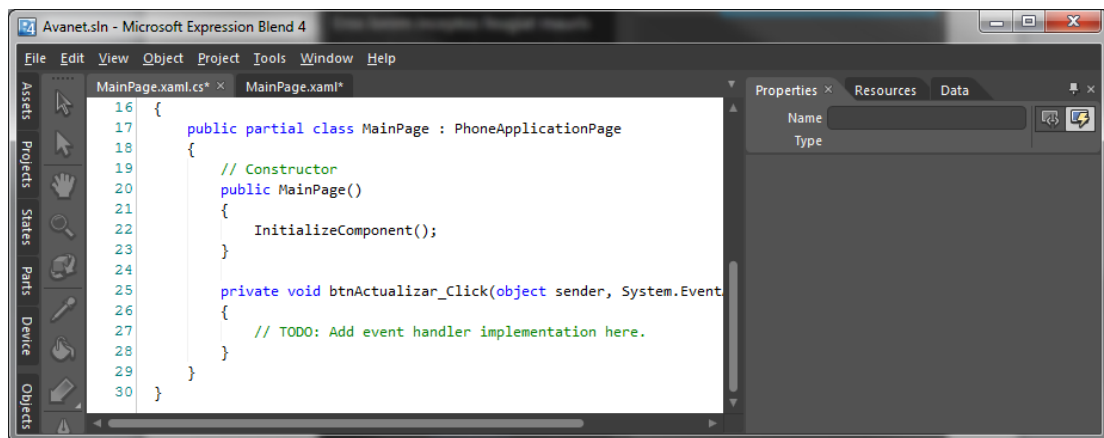
5. Asigne un nombre al botón, además de un texto, observe que no importa como coloque el texto este se muestra en minúsculas. Recuerde elegir palabras cortas y asignar un icono al botón.



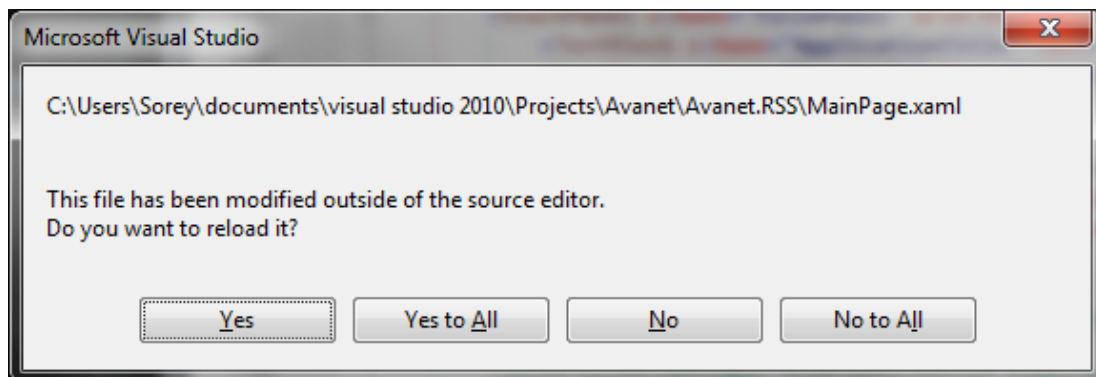
6. Pase a la vista de eventos y de un doble clic sobre el evento que aparece disponible y que verá en blanco, después del doble clic aparecerá el nombre como se ve en el ejemplo.



7. Además se generará el código que se ejecuta cuando se presione el botón.



8. Guarde los cambios y regrese a Visual Studio.

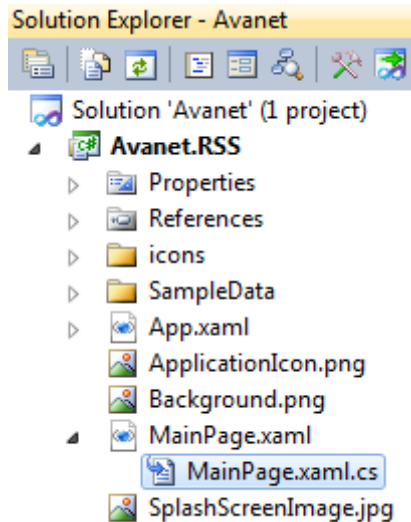


9. Recuerde aceptar que se carguen los cambios si le aparece la alerta, porque de otro modo perderá el trabajo realizado en **Expression Blend**.



## Consumiendo el RSS

1. De doble clic sobre el archivo de código de la página principal



2. Ubíquese en el constructor de la página y suscríbase al evento **Loaded**. Observe que cuando escribe el signo igual le aparece un mensaje **presione dos veces TAB**.

```
public partial class MainPage : PhoneApplicationPage
{
    // Constructor
    public MainPage()
    {
        InitializeComponent();
        this.Loaded +=
            new RoutedEventHandler(MainPage_Loaded); (Press TAB to insert)
    }

    private void btnActualizar_Click(object sender, System.EventArgs e)
    {
        // TODO: Add event handler implementation here.
    }
}
```

3. Verá que automáticamente se crea el método asociado con la firma requerida y este será el lugar donde empezaremos a consumir el RSS.

```
public partial class MainPage : PhoneApplicationPage
{
    // Constructor
    public MainPage()
    {
        InitializeComponent();
        this.Loaded += new RoutedEventHandler(MainPage_Loaded);
    }

    void MainPage_Loaded(object sender, RoutedEventArgs e)
    {
        throw new NotImplementedException();
    }

    private void btnActualizar_Click(object sender, System.EventArgs e)
    {
        // TODO: Add event handler implementation here.
    }
}
```

4. Dentro del método que se creó consuma el RSS usando la URL que ha conseguido, recuerde usar la técnica de dar doble TAB cuando se suscriba al evento **DownloadStringAsync**.

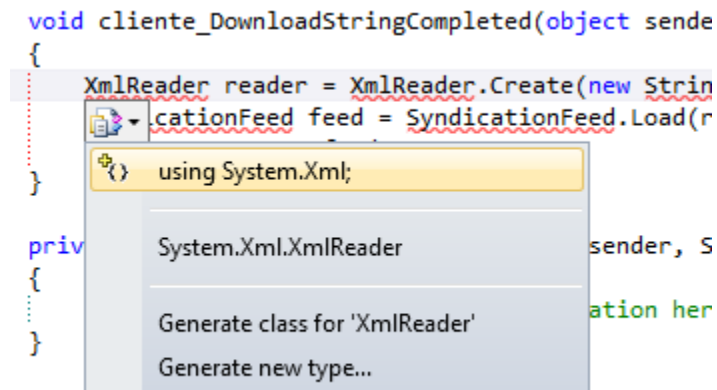
*Observe que estamos concatenando al final de la URL la fecha del dispositivo, esto es por que si se usa la misma URL el dispositivo devuelve lo que tiene cargado en memoria y simulamos con este truco que la URL es diferente para cuando presionen el botón actualizar.*

```
void MainPage_Loaded(object sender, RoutedEventArgs e)
{
    WebClient cliente = new WebClient();
    cliente.DownloadStringAsync(new Uri("http://avanet.org/blog924rss.aspx"
        + "?Trick=" + DateTime.Now.ToShortDateString() + DateTime.Now.ToShortTimeString()));
    cliente.DownloadStringCompleted +=
        new DownloadStringCompletedEventHandler(cliente_DownloadStringCompleted); (Press TAB to insert)
}
```

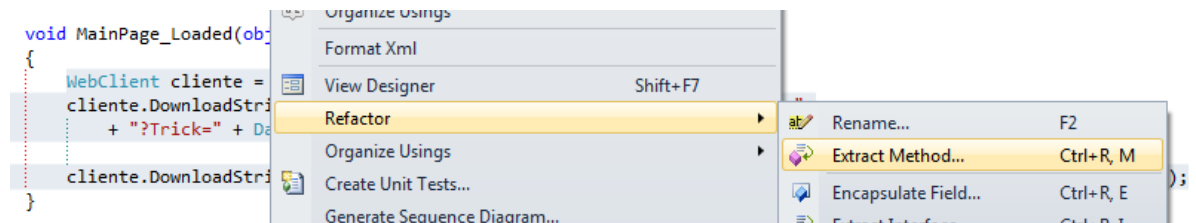
5. En el método que se crea implemente la asignación. Observe que algunas clases no se reconocen aún, cuando se ubica sobre ellas una pequeña línea azul aparece.

```
void cliente_DownloadStringCompleted(object sender, DownloadStringCompletedEventArgs e)
{
    XmlReader reader = XmlReader.Create(new StringReader(e.Result));
    SyndicationFeed feed = SyndicationFeed.Load(reader);
    // The type or namespace name 'SyndicationFeed' could not be found (are you missing a using directive or an assembly reference?)
}
```

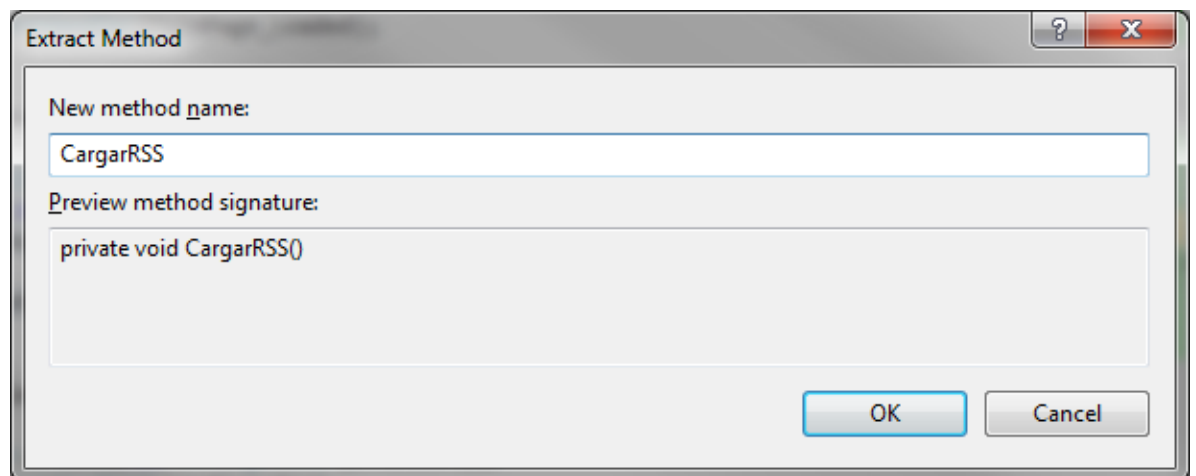
6. Al pasar con el mouse sobre la línea azul aparecerá un botón con el que puede usar los espacios de nombre a los que corresponde la clase.



- 7.
8. Debido a que el método que consume el RSS necesitamos invocarlo en el **Loaded** de la página y en el clic del botón, refactorice para crear un método, para esto seleccione el código que se ha creado dentro del **Loaded**, presione clic derecho y busque la opción refactorizar.



9. Asigne un nombre al método



10. Observe que se crea un método y en el **Loaded** se escribe automáticamente su invocación. Copie esta invocación también en el código asociado al botón actualizar.

```
void MainPage_Loaded(object sender, RoutedEventArgs e)
{
    CargarRSS();
}

private void CargarRSS()
{
    WebClient cliente = new WebClient();
    cliente.DownloadStringAsync(new Uri("http://avanet.org/blog924rss.aspx"
        + "?Trick=" + DateTime.Now.ToShortDateString() + DateTime.Now.ToShortTimeString()));
    cliente.DownloadStringCompleted += new DownloadStringCompletedEventHandler(cliente_DownloadStringCompleted);
}

void cliente_DownloadStringCompleted(object sender, DownloadStringCompletedEventArgs e)
{
    XmlReader reader = XmlReader.Create(new StringReader(e.Result));
    SyndicationFeed feed = SyndicationFeed.Load(reader);
    this.DataContext = feed;
}

private void btnActualizar_Click(object sender, System.EventArgs e)
{
    CargarRSS();
}
```

11. Si llega a probar la aplicación en este momento notará que al iniciar la aplicación parece que no hace nada durante unos segundos, razón por la que es adecuado crear una barra de progreso. Declare un **ProgressIndicator** en la parte superior de la clase e instáncielo en el constructor.

```
public partial class MainPage : PhoneApplicationPage
{
    ProgressIndicator barraProgreso;

    // Constructor
    public MainPage()
    {
        InitializeComponent();
        this.Loaded += new RoutedEventArgs(MainPage_Loaded);

        this.barraProgreso = new ProgressIndicator()
        {
            IsIndeterminate = true,
            Text = "Descargando RSS.."
        };

        SystemTray.SetProgressIndicator(this, barraProgreso);
    }
}
```

12. En el método CargarRSS haga que la barra de progreso sea visible

```
private void CargarRSS()
{
    this.barraProgreso.IsVisible = true;

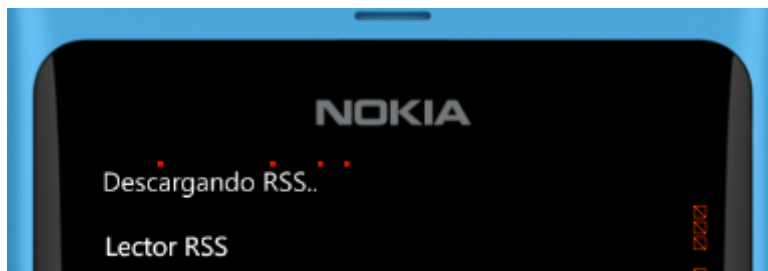
    WebClient cliente = new WebClient();
    cliente.DownloadStringAsync(new Uri("http://avanet.org/blog924rss.aspx"
        + "?Trick=" + DateTime.Now.ToShortDateString() + DateTime.Now.ToShortT
    cliente.DownloadStringCompleted += new DownloadStringCompletedEventHandler
}
```

13. Y en el método donde se ha finalizado la carga del RSS haga que vuelva a ser invisible.

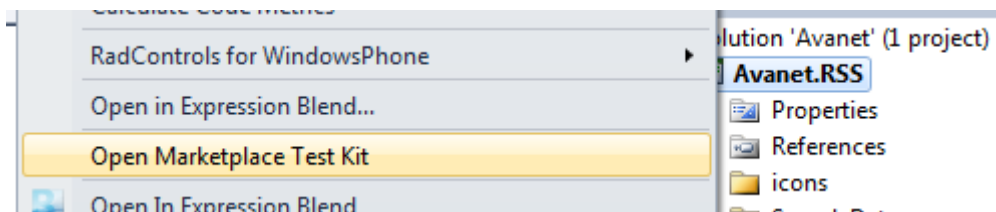
```
void cliente_DownloadStringCompleted(object sender, DownloadStringCompletedEventArgs e)
{
    XmlReader reader = XmlReader.Create(new StringReader(e.Result));
    SyndicationFeed feed = SyndicationFeed.Load(reader);
    this.DataContext = feed;

    barraProgreso.IsVisible = false;
}
```

14. De esa forma cuando la aplicación este trabajando el usuario verá el indicador de progreso.



15. Recuerde controlar los posibles puntos de error de su aplicación usando bloques **Try Catch** y presentando los mensajes de error. Personalice la apariencia de su aplicación para tener más puntos a su favor en las pruebas de la maratón.
16. Recuerde probar su aplicación con el **Marketplace Test Kit**, de clic derecho sobre el proyecto y siga los pasos



A continuación puedes ver como se vería la aplicación finalizada.

