

# Favoritos mi primera app de Windows® Phone 7. Como se hizo...

---

Ya está disponible en el Microsoft Marketplace mi primera aplicación para Windows® Phone 7.5.

Se llama "Favoritos" y es gratuita, es una sencilla aplicación que permite guardar contactos como favoritos. Una funcionalidad que no viene de serie en Windows® Phone 7...



## Características:

- Permite añadir contactos de tu agenda como favoritos, hasta 100!
- Cuando añades un favorito puedes dejar el nombre del contacto de tu agenda o cambiarlo
- Permite realizar llamadas a tus favoritos
- Permite enviar SMS a tus favoritos
- Permite borrar tus favoritos, esta acción no borra el contacto de tu agenda
- El tema de la aplicación se corresponde con el tema de tu teléfono
- Es multiidioma. Actualmente soporta inglés y castellano. El idioma se aplica en función de la región de tu teléfono

## Como se hizo Favoritos:

Favoritos ha sido una aplicación fácil de desarrollar, básicamente lo que tenía que hacer era acceder a la agenda, guardar el contacto seleccionado de alguna manera y usar los lanzadores de aplicaciones para llamar o enviar SMS. Una vez desarrollado todo el eje central de la aplicación el resto es adornarlo: hacerla multiidioma, que el tema sea dinámico y hacer una interfaz vistosa de fácil uso para el usuario.

Se me hace difícil calcular cuántas horas he invertido en su desarrollo ya que la he ido haciendo en ratos libres, pero con una buena planificación yo diría que esta aplicación se puede desarrollar en 3 o 4 días. En mi caso donde he dedicado más tiempo es en toda la parte del diseño ya que no tengo casi experiencia trabajando con .xaml. Toda la parte lógica ha sido muy sencilla ya que apenas hay código, gracias a las plantillas de proyecto de WP7, las propiedades dependientes, el sistema de bindado de Silverlight y las propias funciones de WP7. Y el toolkit de Silverlight para WP7 de codeplex me ha resultado de gran ayuda para poder controlar fácilmente el tema del tap, el doubletap y el menú contextual para eliminar los favoritos.

Y ahora al lío!

**Nota: El siguiente texto va dirigido a desarrolladores de WP7 y/o Silverlight, doy por hecho que se tienen unos conocimientos mínimos de las tecnologías utilizadas**

Para Favoritos he utilizado la plantilla de proyecto Panorama, la aplicación cuenta con 2 panorama items, uno para mostrar los favoritos y otro para mostrar el about. Esto es una parte del código .xaml del primer panorama item:

```
<ListBox ItemsSource="{Binding Items}" Name="lstContacts" >
<ListBox.ItemTemplate>
<DataTemplate>
<Canvas Width="400" Height="100" toolkit:TiltEffect.IsTiltEnabled="True" Margin="15,0,0,20"
UseLayoutRounding="True">
<Border CornerRadius="20,20,0,0" Canvas.Left="0" Canvas.Top="0" Width="400" Height="100">
<Border.Background>
<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
<GradientStop Color="{StaticResource PhoneAccentColor}" Offset="0"/>
<GradientStop Color="{StaticResource PhoneBackgroundColor}" Offset="1"/>
</LinearGradientBrush>
</Border.Background>
<toolkit:ContextMenuService.ContextMenu>
<toolkit:ContextMenu>
<toolkit:MenuItem
DataContext="{Binding Alias}"
Header="{Binding Path=LocalizedResources.litClear, Source={StaticResource LocalizedStrings}}"
Click="MenuItem_Click"/>
</toolkit:ContextMenu>
</toolkit:ContextMenuService.ContextMenu>
</Border>
<Image Name="imgIconPhone" Stretch="Fill" Source="{Binding IconPhone}" Canvas.Left="17" Canvas.Top="25" />
<TextBlock Name="txtAlias" Text="{Binding Alias}" FontFamily="Arial" Canvas.Top="10" Canvas.Left="85"
FontSize="40" Foreground="{StaticResource PhoneContrastBackgroundBrush}"></TextBlock>
<TextBlock Name="txtPhoneNumber" Text="{Binding PhoneNumber}" FontFamily="Arial" Canvas.Top="61"
Canvas.Left="88" FontSize="20" Foreground="{StaticResource PhoneContrastBackgroundBrush}"></TextBlock>
<toolkit:GestureService.GestureListener>
<toolkit:GestureListener DoubleTap="OnDoubleTap" Tap="OnTap" />
</toolkit:GestureService.GestureListener>
</Canvas>
</DataTemplate>
</ListBox.ItemTemplate>
</ListBox>
```

En el listbox se puede observar que tiene una colección de objetos (Items) bindada, dicha colección es una clase con propiedades como alias y teléfono, cada vez que añadimos o quitamos un contacto se actualiza la colección de ítems vía código y el listbox se refresca automáticamente, esta es la grandeza de Silverlight.

Cada aplicación de WP7 tiene un espacio reservado en el teléfono, en dicho espacio se puede almacenar lo que queramos, como Favoritos no necesita mucho espacio ni guarda mucha información he utilizado la clase IsolatedStorage para almacenar los contactos. Esta clase nos permite guardar una key + un valor en el fichero de configuración interno de la app. La clase tiene métodos para recuperar estos valores o para eliminarlos.

El lanzador para sacar la agenda es `PhoneNumberChooserTask`. Con 4 líneas de código se puede lanzar la agenda del teléfono y controlar que contacto se ha seleccionado:

```
phoneNumberChooserTask = new PhoneNumberChooserTask();
phoneNumberChooserTask.Completed += new EventHandler<PhoneNumberResult>(phoneNumberChooserTask_Completed);

//Muestra la agenda
phoneNumberChooserTask.Show();

//Evento que se dispara cuando seleccionamos un contacto
void phoneNumberChooserTask_Completed(object sender, PhoneNumberResult e)
{
    if (e.TaskResult == TaskResult.OK)
    {
        if (e.PhoneNumber != string.Empty)
        {
            //TODO, en mi caso llamo a la página de añadir contacto, donde
            //controlo el nombre y guardo el nuevo favorito
        }
    }
}
```

El lanzador para llamar a un número es el siguiente:

```
PhoneCallTask phoneCallTask = new PhoneCallTask();
phoneCallTask.DisplayName = "pepe";
phoneCallTask.PhoneNumber = "555-555-555";
phoneCallTask.Show();
```

Y el lanzador para enviar un SMS es el siguiente:

```
SmsComposeTask composeSMS = new SmsComposeTask();
composeSMS.Body = "Hola pepe";
composeSMS.To = "555-555-555";
composeSMS.Show();
```

Para sacar el menú contextual que permite eliminar favoritos simplemente me he descargado/instalado el toolkit de Silverlight para WP7 de codeplex <http://silverlight.codeplex.com/releases/view/52297> y ya te lo da casi todo hecho, basta con añadir en el .xaml:

```
<toolkit:ContextMenuService.ContextMenu>
<toolkit:ContextMenu>
<toolkit:MenuItem
    DataContext="{Binding Alias}"
    Header="{Binding Path=LocalizedResources.litClear, Source={StaticResource LocalizedStrings}}"
    //recurso Eliminar
    Click="MenuItem_Click"/>
```

```
</toolkit:ContextMenu>
</toolkit:ContextMenuService.ContextMenu>
```

Con estas pocas líneas ya se muestra el menú contextual al presionar unos segundos sobre cualquier favorito y al seleccionar el menulitem se dispara el evento MenuItem\_Click donde podemos obtener que favorito se ha seleccionado.

Con el toolkit también controlo los gestos tap y doubleTap:

```
<toolkit:GestureService.GestureListener>
<toolkit:GestureListener DoubleTap="OnDoubleTap" Tap="OnTap" />
</toolkit:GestureService.GestureListener>
```

Cuando se realice cualquiera de esas 2 acciones se dispara el evento OnTap o OnDoubleTap, el problemilla que hay es que cuando haces un doble toque también estás haciendo un toque, por lo tanto por código nos corresponde la tarea de controlarlo. Con un simple Thread.Sleep y un par de métodos asíncronos se puede hacer.

Respecto a controlar el tema dinámico del teléfono (Si el teléfono tiene tema azul que los botones de la app se vean azules por ejemplo) lo único que hay que es utilizar los recursos estáticos propios de WP7, por ejemplo:

Que el fondo sea como el del telefono:

```
Background="{StaticResource PhoneBackgroundBrush}"
```

Que el fondo de un textblock sea del color del tema del teléfono:

```
Foreground="{StaticResource PhoneAccentBrush}"
```

O conseguir un degradado con el color del tema en un botón:

```
<Button.Background>
<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
<GradientStop Color="{StaticResource PhoneAccentColor}" Offset="0"/>
<GradientStop Color="{StaticResource PhoneBackgroundColor}" Offset="1"/>
</LinearGradientBrush>
</Button.Background>
```

Por ultimo está el multiidioma, para que la aplicación soporte varios idiomas lo que hay que hacer es:

- Añadir un fichero de recurso por idioma y uno que será el internacional.  
En mi caso he añadido 3: AppResource.resx, AppResource.es-ES.resx y AppResource.en-GB.resx.  
Donde AppResource.resx es el internacional que será el en-US.

- Añadir en el fichero App.xaml un recurso de aplicación:

```
<!--Application Resources-->
<Application.Resources>
<local:LocalizedStrings xmlns:local="clr-namespace:Favoritos"
x:Key="LocalizedStrings" />
</Application.Resources>
```

- Añadir la clase, a la que se hace referencia en el recurso de aplicación, encargada de devolver los literales

```
public class LocalizedStrings
{
    public LocalizedStrings() {}
}
```

```
private static Favoritos.AppResource localizedresources = new Favoritos.AppResource();
public Favoritos.AppResource Localizedresources { get { return localizedresources; } }
}
```

- Y con esto ya se pueden usar los recursos utilizando el potente bindado de Silverlight. Por ejemplo para controlar el literal de un panorama ítem:

```
<controls:PanoramaItem Name="panContacts" Header="{Binding
Path=Localizedresources.PanTitle, Source={StaticResource LocalizedStrings}}" >
```

Y poco más que añadir, si tienes cualquier duda no dudes en ponerte en contacto conmigo

@saludos y perdón por el tocho 😊.