

# Hands on Lab: Usando el servicio de localización y mapas de Bing

---

## Etapa 4

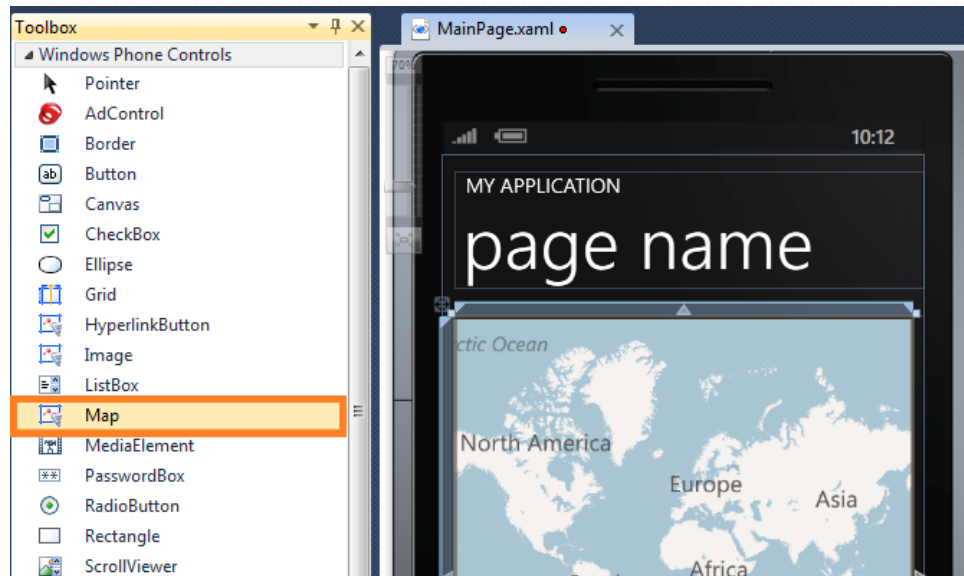
Esta etapa es para los participantes que hayan finalizado la **etapa 3**. En este documento no se guiará sobre la realización de la aplicación paso a paso, sin embargo se darán todas las instrucciones que se necesitan para construirlo, además este reto debe ser complementado con lo aprendido en las etapas anteriores. Si necesita ayuda con alguno de los temas puede preguntar a los entrenadores.

**Reto:** Crear una nueva aplicación para almacenar localmente los lugares favoritos.

1. Crear un nuevo proyecto con una plantilla simple y personalizar las imágenes base
2. Añadir un mapa que muestre con el GPS la ubicación actual.
3. Implementar la confirmación con el usuario para que debe usar el GPS y que si no lo permite controle el error y no deje usar la aplicación.
4. Permitir que el usuario agregue en el mapa un nuevo lugar favorito.
5. Del lugar favorito se deben guardar un nombre, una descripción, una categoría y las coordenadas, por lo que al seleccionar la opción añadir debe navegarse a otro formulario que permite diligenciar estos datos y al guardar, retornar al mapa donde debe verse el ítem guardado ya graficado.
6. Crear en la barra de aplicación una opción para ver los lugares que se tienen almacenados en un listbox.
7. Crear en la barra de aplicación dos ítems de menú para cambiar a vista aérea y a vista de calles.
8. En el listbox de lugares almacenados colocar un menú contextual que permita eliminar ítems de la lista.
9. Personalizar la apariencia de los pushpin que mi ubicación tenga una apariencia diferente, y que cada ubicación graficada tenga una apariencia según la categoría a la que pertenece.
10. Revisar la aplicación con el Marketplace Test Kit.
11. Publicar aplicación en el Marketplace.

## *Añadiendo un Mapa de Bing a la aplicación*

1. Agregar un mapa a nuestra aplicación es muy simple, seleccionamos y arrastramos el control Mapa a la página donde lo vamos a usar desde el cuadro de herramientas en Visual Studio.



2. Antes de empezar vamos obtener un key o clave autorizada para nuestra aplicación, de otro modo cuando ejecutemos la aplicación veremos una advertencia sobre nuestro mapa:



3. Lo que debemos hacer es visitar <http://www.bingmapsportal.com> y crear una nueva cuenta de Windows Live o iniciar sesión con un Live ID existente, que bien puede ser su misma cuenta del APPHUB.
4. Lo primero en el portal de Bing será crear una nueva cuenta para nuestras aplicaciones si es que ya no la tenemos creada.

## Create an account

You need an account to create keys or upload map apps.

### Account details

**\* Account name**  
Your account name

**Contact name**  
Your contact name

**Company name**  
Your company name

**\* Email address**  
Your email address

**Phone number**  
Your phone number

☒ I agree to the [Bing Maps API Terms of Use](#). The information I provide will be used in keeping with the [Microsoft Online Privacy Statement](#) and by Bing Maps to provide me with service updates, maintenance notifications, account management inquiries and/or survey invitations.

ClearSave

5. En mi caso que la tengo creada hace un tiempo accediendo al menú **Create or view key maps**

[Bing Maps Account Center Help](#)  
**My Account**  
[Update or view account details](#)  
[Create or view keys](#)  
[View my Bing Maps API usage](#)  
**Map Apps**  
[Submit a map app](#)  
[View my map apps](#)  
**Resources**  
[Bing Maps Platform](#)  
[Bing Maps SDKs](#)  
[Bing Maps Forums](#)  
[Bing Maps Blog](#)

### My keys

You need a key to authenticate your Bing Maps application. You can create only one Bing Maps key. If you need more keys, please contact [mpnet@microsoft.com](mailto:mpnet@microsoft.com).

Click [here](#) to view/download complete list of keys.

Application name	Key / URL	Update Key
SoreyGarcía	Developer	<a href="#">Update</a>

6. De acuerdo a las instrucciones de MSDN y al mensaje que nos aparece en la vista anterior, podemos crear una clave y si necesitamos más es necesario contactarlos.
7. También en MSDN podemos ver que significa el tipo de aplicación que elegimos y sus restricciones, evidentemente para nuestra aplicación seleccionamos Mobile.

## My keys

You can update the Bing Maps key.

### Update key

**\* Application name**

**Application URL**

**\* Application type**

Developer

Please select a value

Broadcast

Developer

Education

Mobile

Not-for-profit

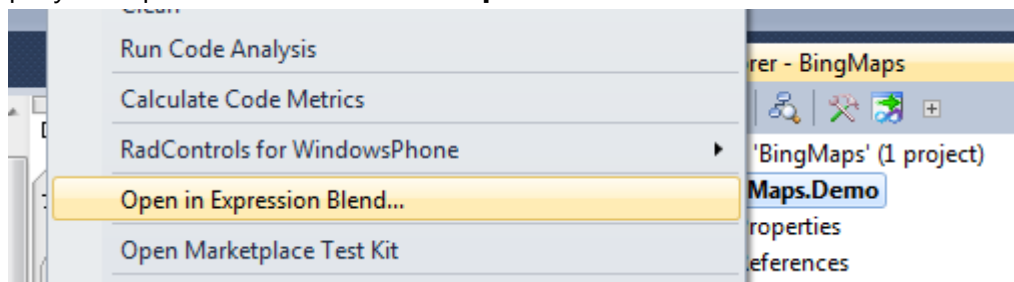
What's this?

b6TgwTM7wcZeP64kZG-

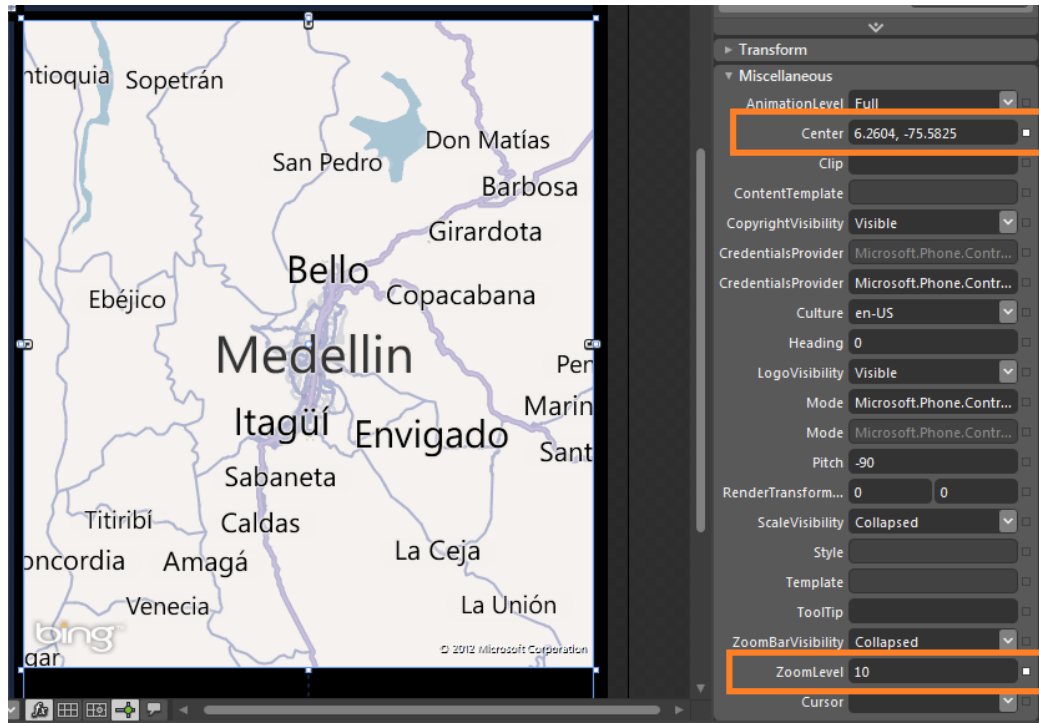
dhbn

Submit

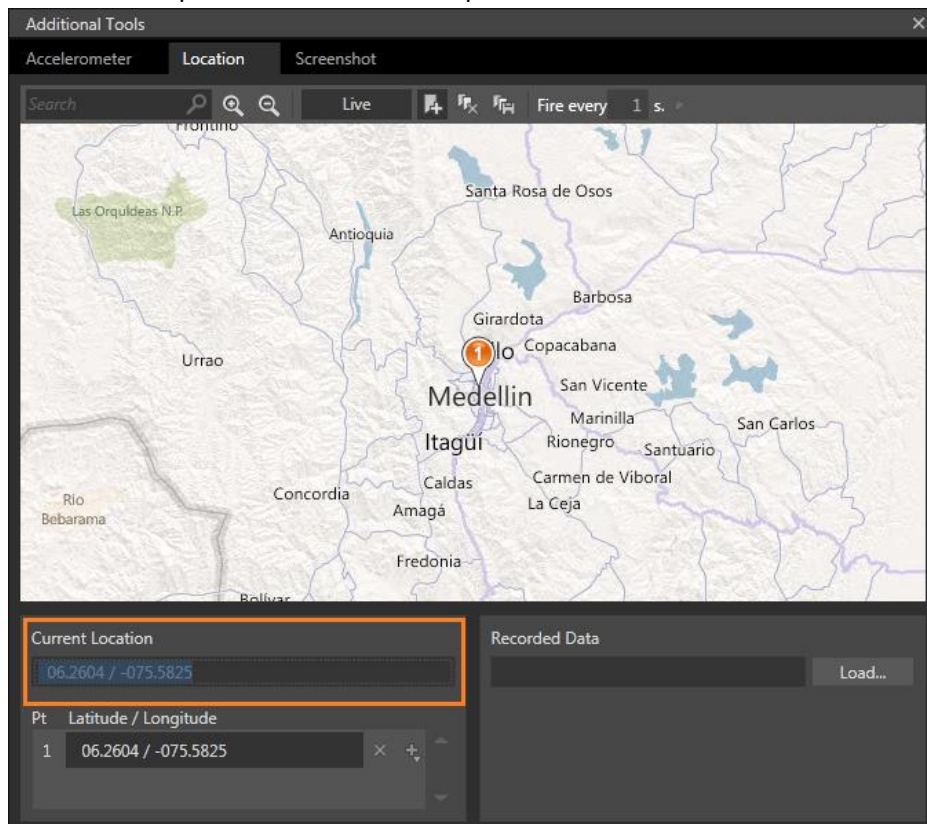
8. Vamos entonces a configurar el estado inicial de nuestro mapa, así que desde Visual Studio podemos abrir nuestro proyecto dando clic derecho sobre el proyecto que deseamos editar en **Expression Blend**.



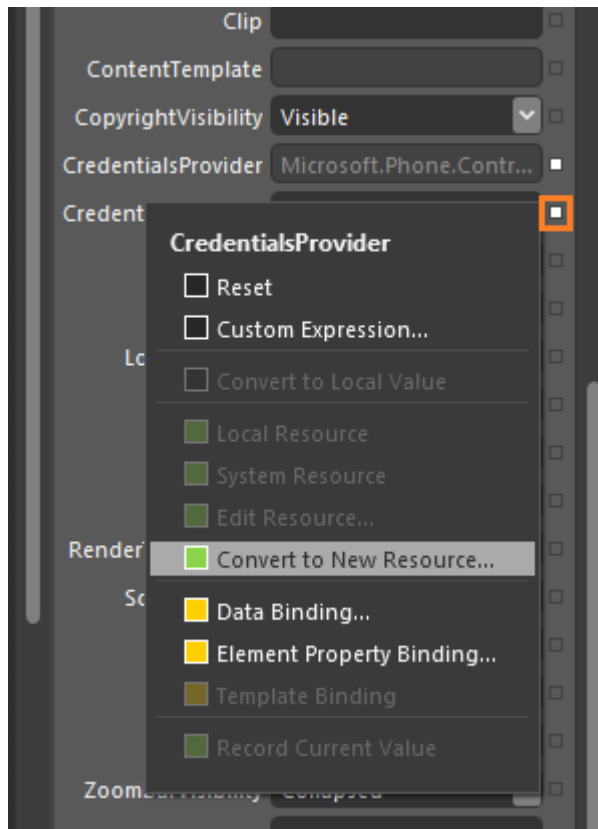
9. Seleccionamos nuestro control y vamos a la pestaña de Misceláneo en las propiedades que típicamente encontramos a la derecha y configuramos el punto central y el zoom inicial.



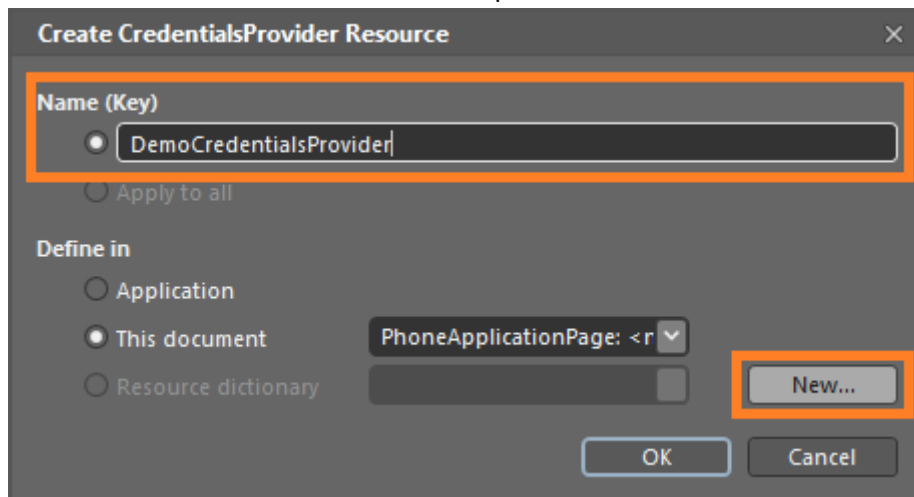
10. Para encontrar el punto inicial central recuerden que podemos usar las herramientas disponibles en el emulador, en este caso el emulador del GPS. Más adelante veremos como establecer programáticamente esta ubicación o bien establecer la posición actual del dispositivo.



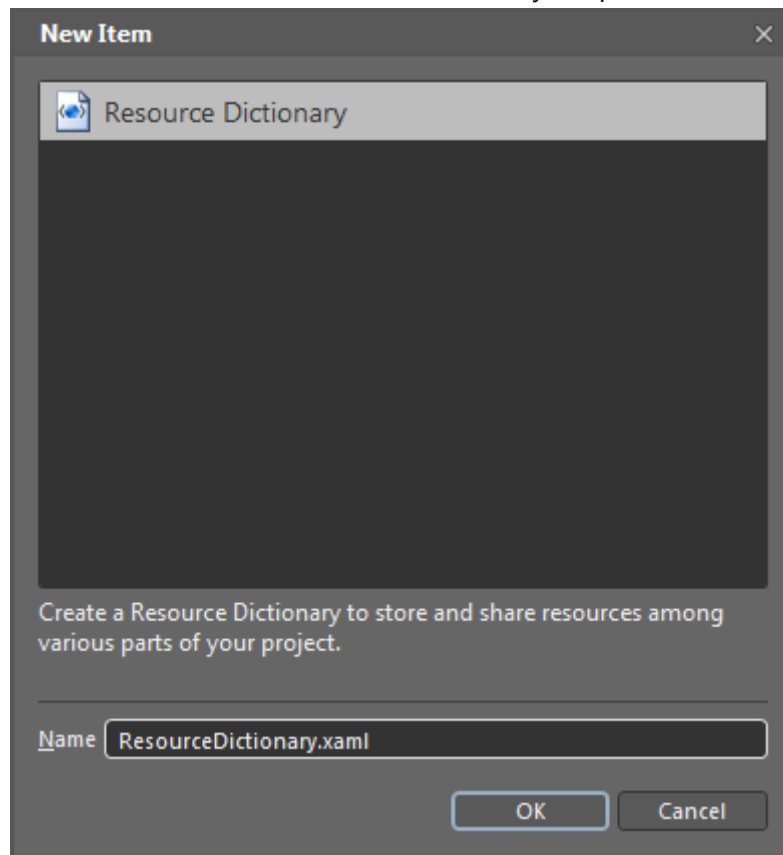
11. Y para terminar nuestra primera parte, configuramos la clave que obtuvimos para el mapa, con el fin de que funcione correctamente, para eso crearemos un nuevo recurso, así:



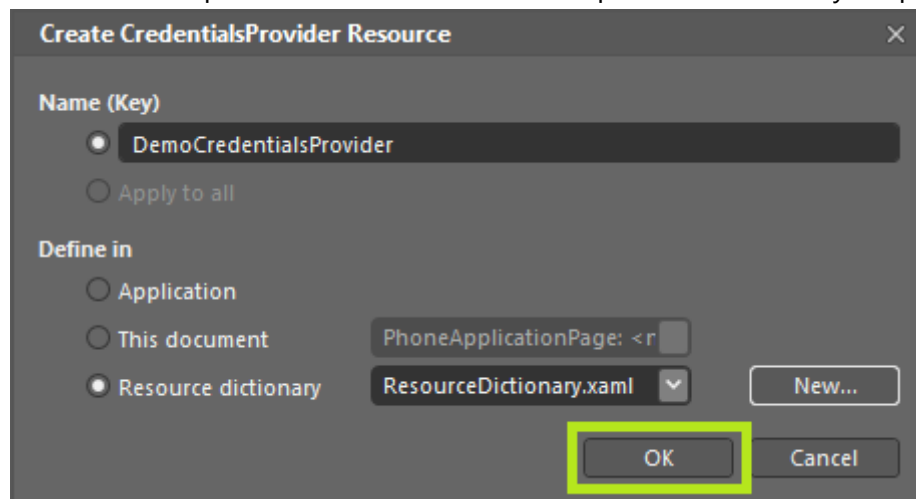
12. La recomendación y para el código de nuestras páginas quede más limpio, es que usemos diccionarios de recursos, así que crearemos uno.



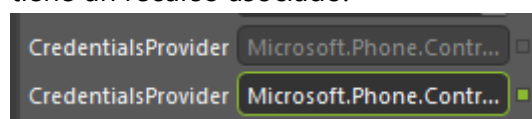
13. Establecemos el nombre del diccionario y aceptamos.



14. Vemos como aparece ahora el diccionario disponible en la lista y aceptamos.

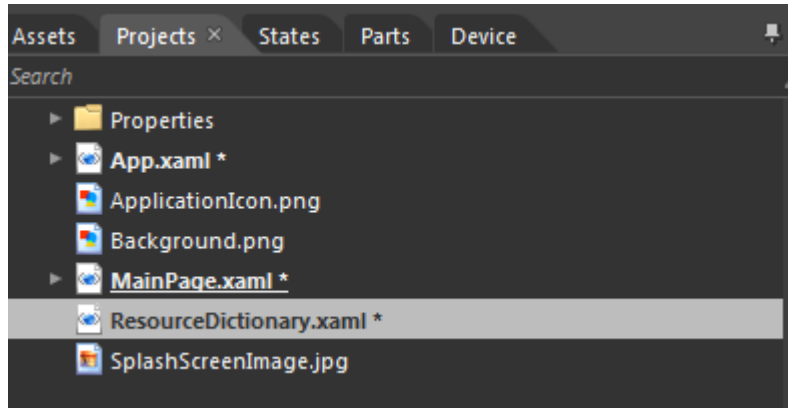


15. Además la propiedad que estábamos enlazando cambia de color indicándonos que tiene un recurso asociado.

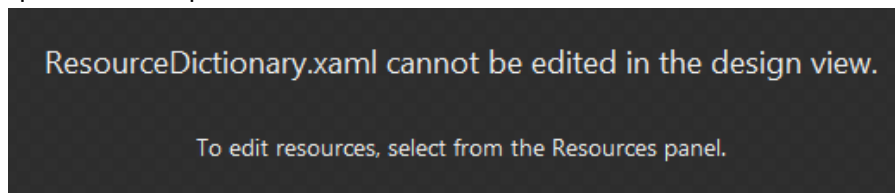




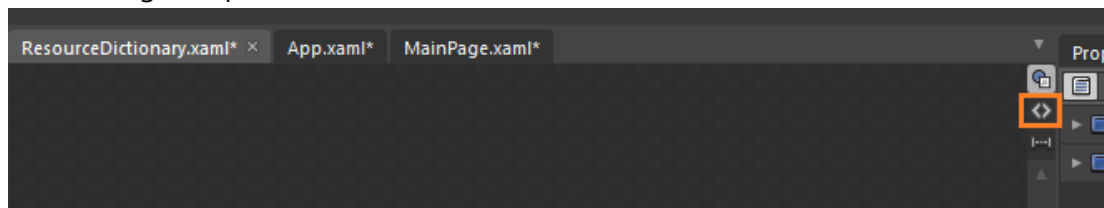
16. Ahora buscamos en la vista de proyectos generalmente en la zona izquierda superior el archivo de recursos para establecer nuestra clave.



17. Recibiremos un mensaje que nos indica que no se puede editar en vista de diseño y que vamos al panel de recursos



18. Sin embargo lo que nosotros haremos es ver el XAML del archivo



19. Allí establecemos la clave que obtuvimos en el portal de mapas de Bing, en el atributo **ApplicationId**

```
1 <ResourceDictionary
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:Microsoft_Ph
4     <!-- Resource dictionary entries should be defined here. -->
5     <Microsoft_Phone_Controls_Maps:ApplicationIdCredentialsProvider
6         x:Key="DemoCredentialsProvider" ApplicationId="--- Clave Bing ---"/>
7 </ResourceDictionary>
8
```

20. Ahora podemos ejecutar de nuevo nuestra aplicación y ver que el mapa funciona correctamente.



## Usando el GPS para establecer la ubicación actual

1. Ingrese al código asociado a la página principal. Declare un **GeoCoordinateWatcher** y un **Pushpin** o marcador. Genere los métodos asociados al **PositionChanged** y el **StatusChanged** tal como se hizo en la etapa 2, haciendo doble TAB después de escribir el igual. Además inicie la instancia del **GeoCoordinateWatcher** en el evento asociado al **Loaded** de la página.

```
public partial class MainPage : PhoneApplicationPage
{
    GeoCoordinateWatcher watcher;
    Pushpin actualPosition;

    // Constructor
    public MainPage()
    {
        InitializeComponent();
        watcher = new GeoCoordinateWatcher ();
        watcher.PositionChanged += new EventHandler<GeoPositionChangedEventArgs<GeoCoordinate>>(watcher_PositionChanged);
        watcher.StatusChanged += new EventHandler<GeoPositionStatusChangedEventArgs>(watcher_StatusChanged);

        watcher.MovementThreshold = 20;

        this.Loaded += new RoutedEventHandler(MainPage_Loaded);
    }

    void MainPage_Loaded(object sender, RoutedEventArgs e)
    {
        watcher.Start();
    }
}
```

Ahora implemente el **StatusChanged**, con el código sugerido en MSDN

```
void watcher_StatusChanged(object sender, GeoPositionStatusChangedEventArgs e)
{
    switch (e.Status)
    {
        case GeoPositionStatus.Disabled:
            if (watcher.Permission == GeoPositionPermission.Denied)
            {
                statusTextBlock.Text = "La aplicación no tiene permisos sobre el servicio de localización";
            }
            else
            {
                statusTextBlock.Text = "El servicio de localización no esta funcionando en el dispositivo";
            }
            break;

        case GeoPositionStatus.Initializing:
            break;

        case GeoPositionStatus.NoData:
            statusTextBlock.Text = "Los datos de localización no están disponibles";
            break;

        case GeoPositionStatus.Ready:
            statusTextBlock.Text = "El servicio de localización está disponible";
            break;
    }
}
```

2. Implemente también el **PositionChanged**

```

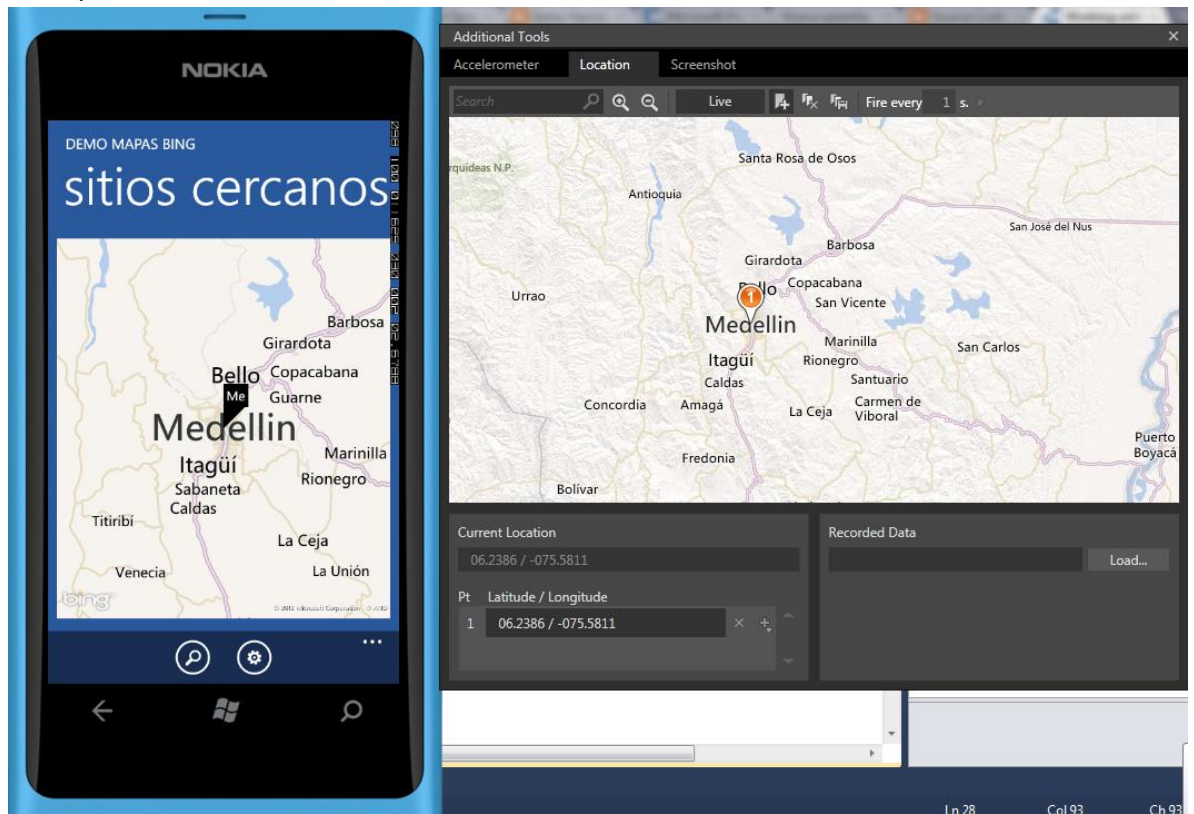
void watcher_PositionChanged(object sender, GeoPositionChangedEventArgs<GeoCoordinate> e)
{
    if (actualPosition != null)
        map1.Children.Remove(actualPosition);

    actualPosition = new Pushpin();
    actualPosition.Location = e.Position.Location;
    actualPosition.Content = "Yo";
    map1.Children.Add(actualPosition);

    map1.SetView(actualPosition.Location, 14);
}

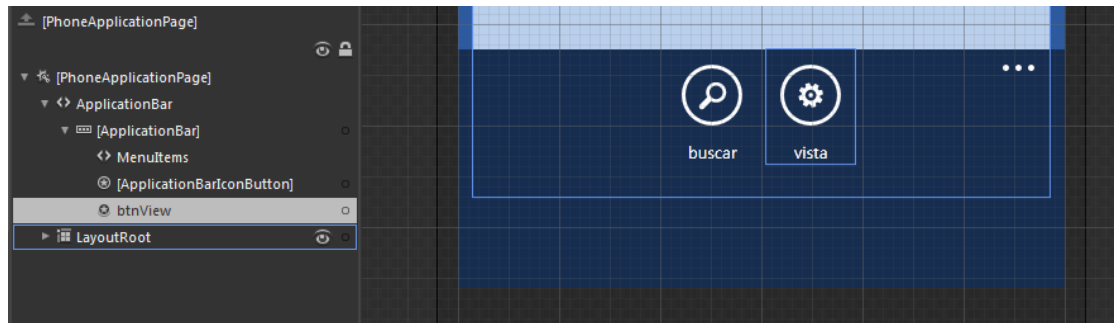
```

3. Recuerde que puede probar el funcionamiento de este código usando el emulador y generando puntos, la aplicación debería graficar el pushpin desplazándolo en cada punto



## Cambiando el tipo de vista

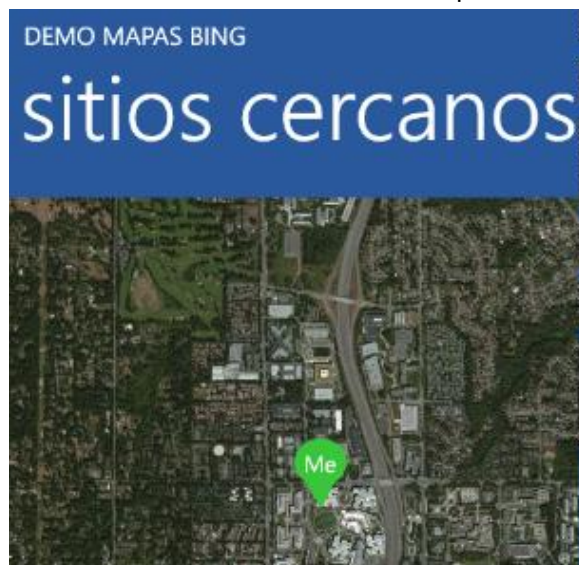
1. Ahora vamos a implementar uno de los botones que tenemos en la barra de aplicaciones, usándolo para cambiar la vista del Mapa de Bing. Configuramos el botón como lo hicimos en la etapa 2



2. Vamos al código asociado al botón y escribimos el código que verifica la vista actual y la cambia por la opuesta.

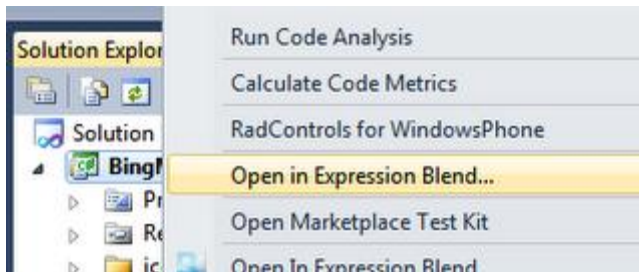
```
private void btnChangeView_Click(object sender, System.EventArgs e)
{
    if (map1.Mode is RoadMode)
    {
        map1.Mode = new AerialMode();
    }
    else
    {
        map1.Mode = new RoadMode();
    }
}
```

3. De este modo la vista de nuestra aplicación cambia cuando presionemos el botón

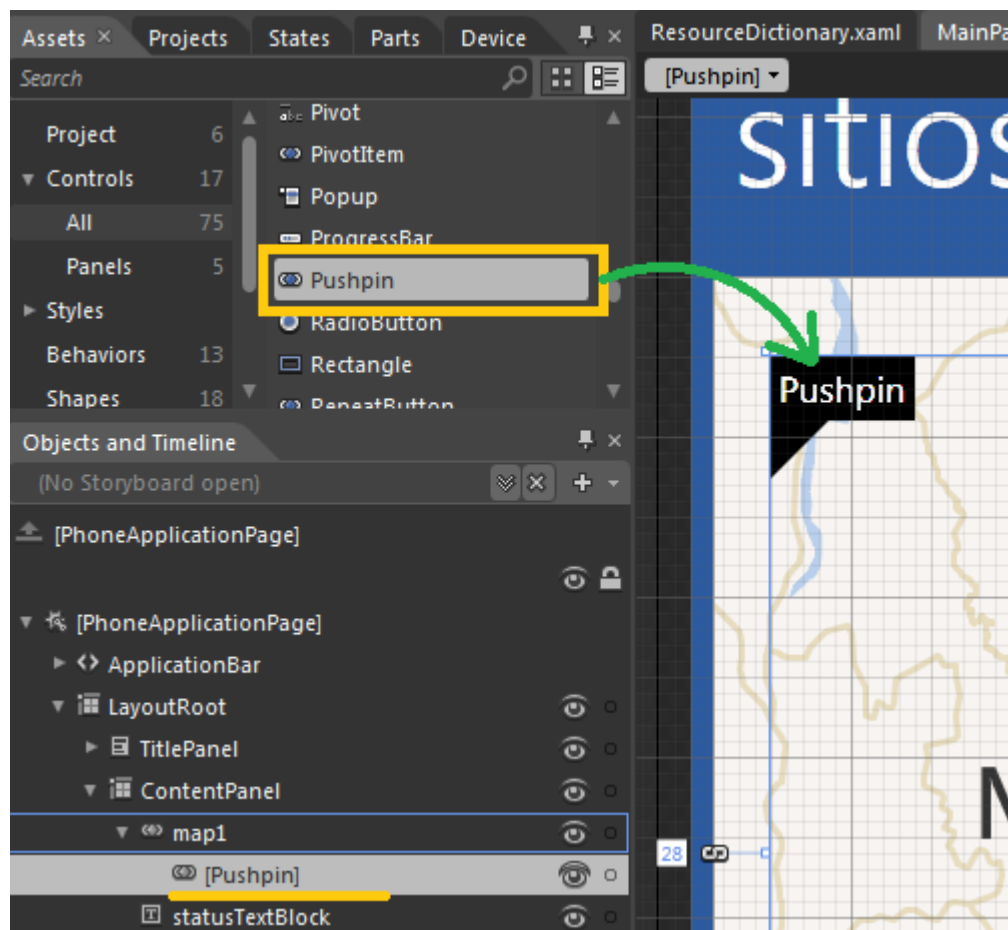


## Personalizando la apariencia de los marcadores

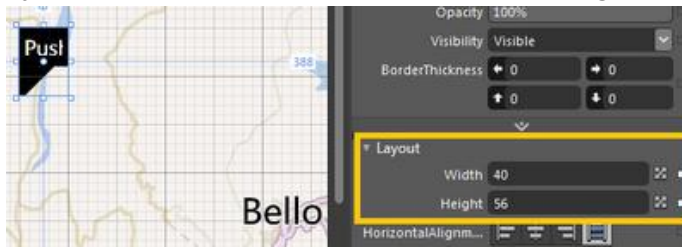
1. Continuando en esto de aprender a añadir funcionalidades a los mapas de Bing, ahora vamos a darle un toque personal a los marcadores o pushpin que colocamos en el mapa. Para esto vamos a usar nuevamente **Expression Blend**.



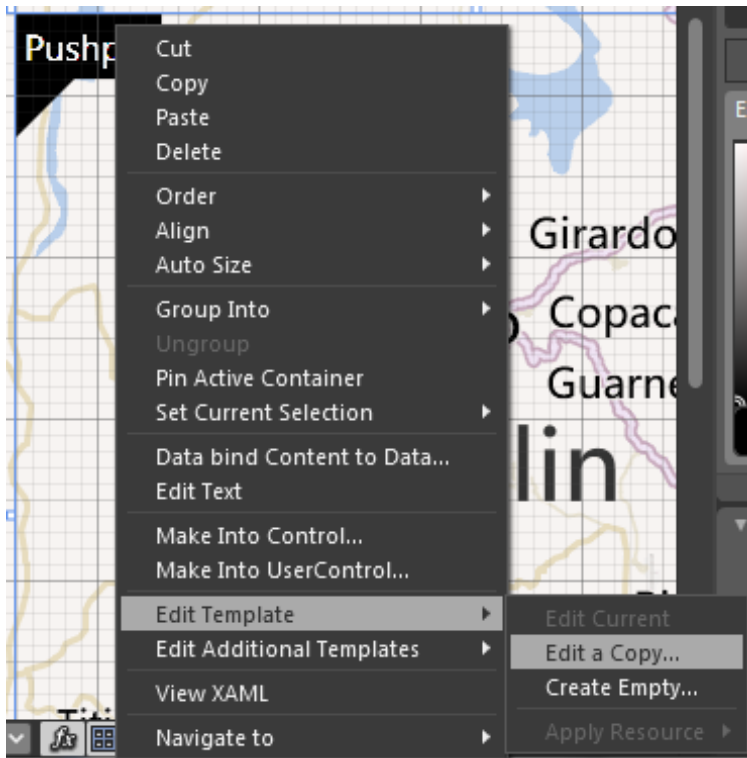
Cuando nuestro proyecto se abra buscamos en la lista de controles el **Pushpin** y lo añadimos sobre el mapa.



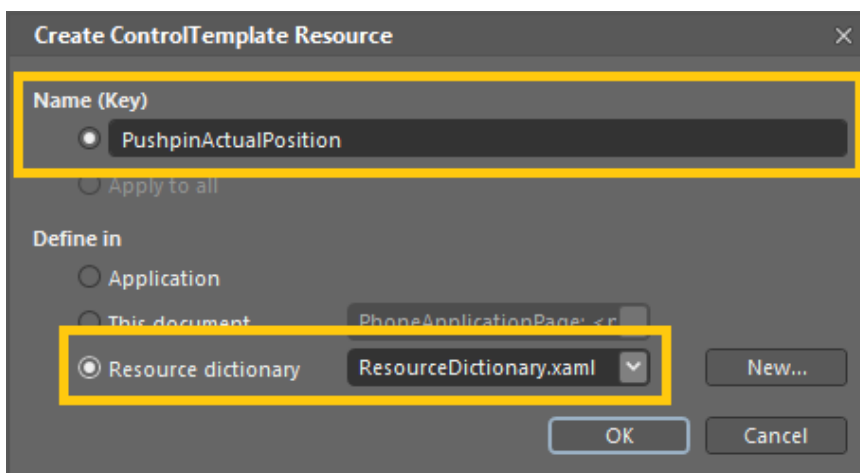
2. Ajustamos el tamaño del pushpin al de la imagen que tenemos para personalizarlo



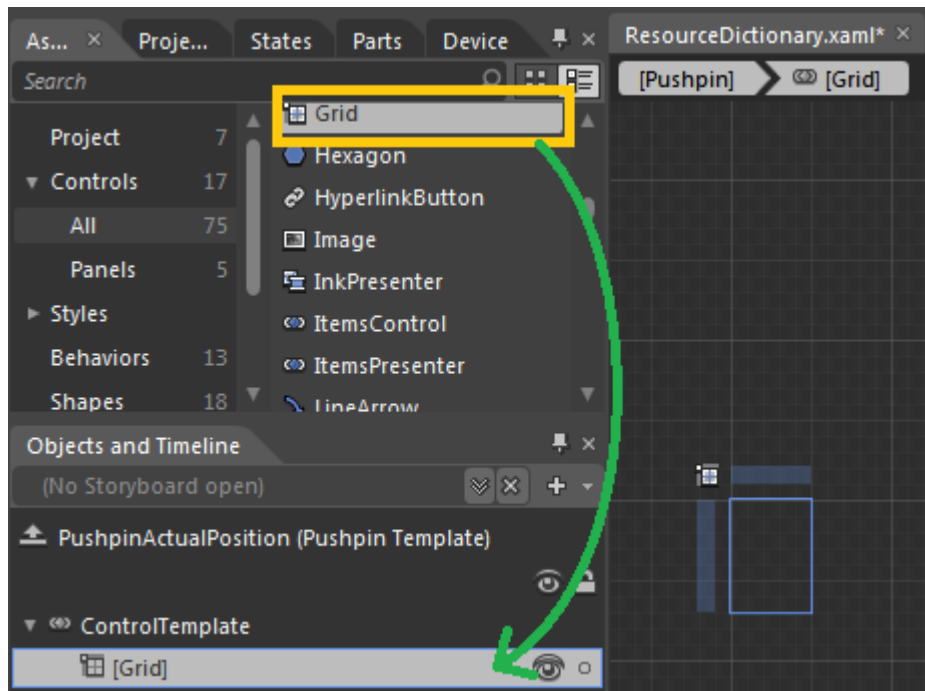
3. Damos clic derecho sobre el pushpin o marcador para editar su plantilla asociada



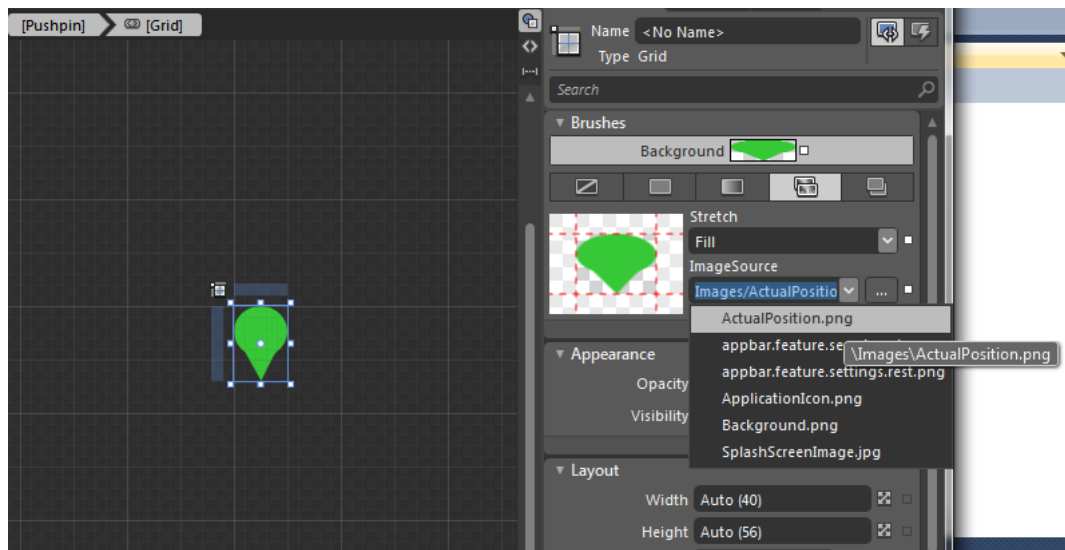
4. Se asigna un nombre a la copia de la plantilla y si han seguido los pasos hasta este momento deberán tener un diccionario de recursos en su aplicación



5. Al entrar a la edición de nuestro marcador agregamos un Grid o Canvas si es que queremos ubicar más controles dentro en una posición determinada.

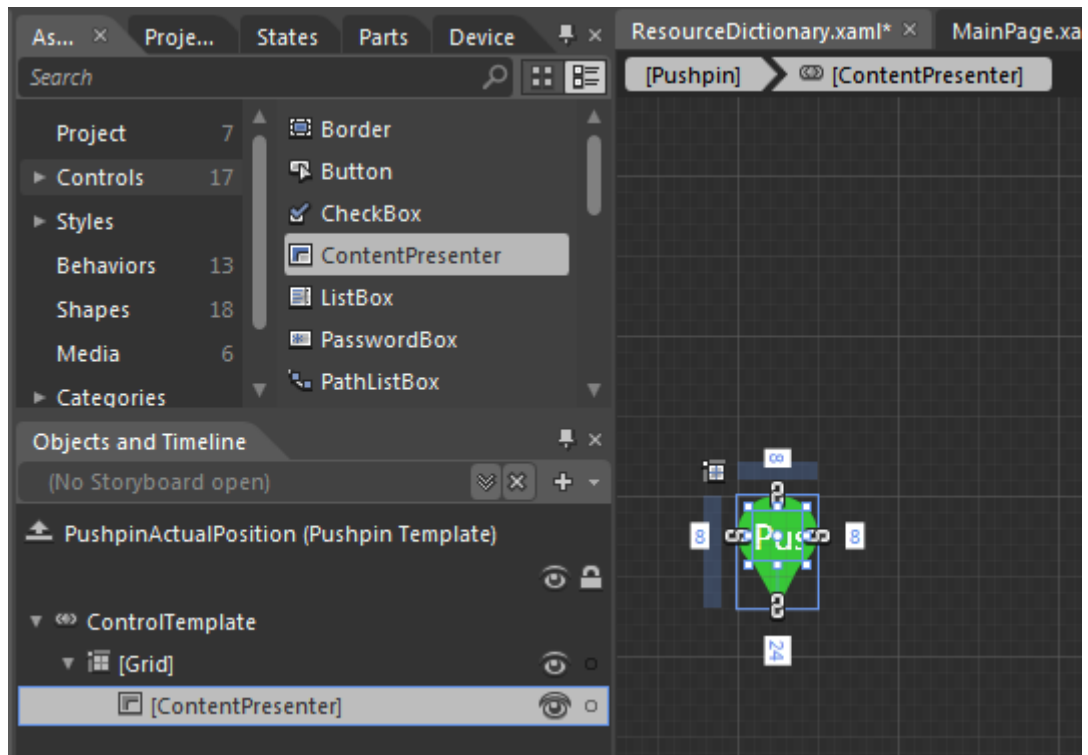


6. Luego establecemos su apariencia, en mi caso usaré una imagen personalizada

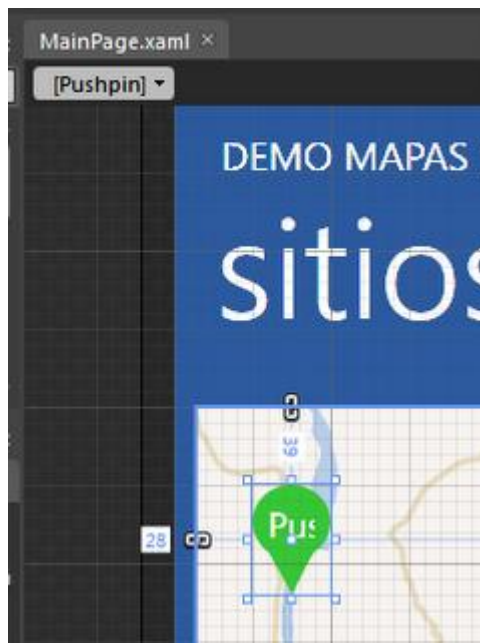


7. Si deseamos que el texto de la propiedad Content del Pushpin se muestre debemos añadir un **ContentPresenter**.





8. Podemos ir a la vista de diseño para ver como queda, sin embargo allí mismo removeremos este marcador ya que los marcadores los crearemos en tiempo de ejecución



9. Guardamos en Expression Blend y podemos volver a Visual Studio. Buscamos el lugar donde estamos instanciando nuestro pushpin, es decir en el **PositionChanged** y allí añadimos la línea que asigna al marcador la plantilla que creamos.

```
void watcher_PositionChanged(object sender, GeoPositionChangedEventArgs<GeoCoordinate> e)
{
    if (actualPosition != null)
        map1.Children.Remove(actualPosition);

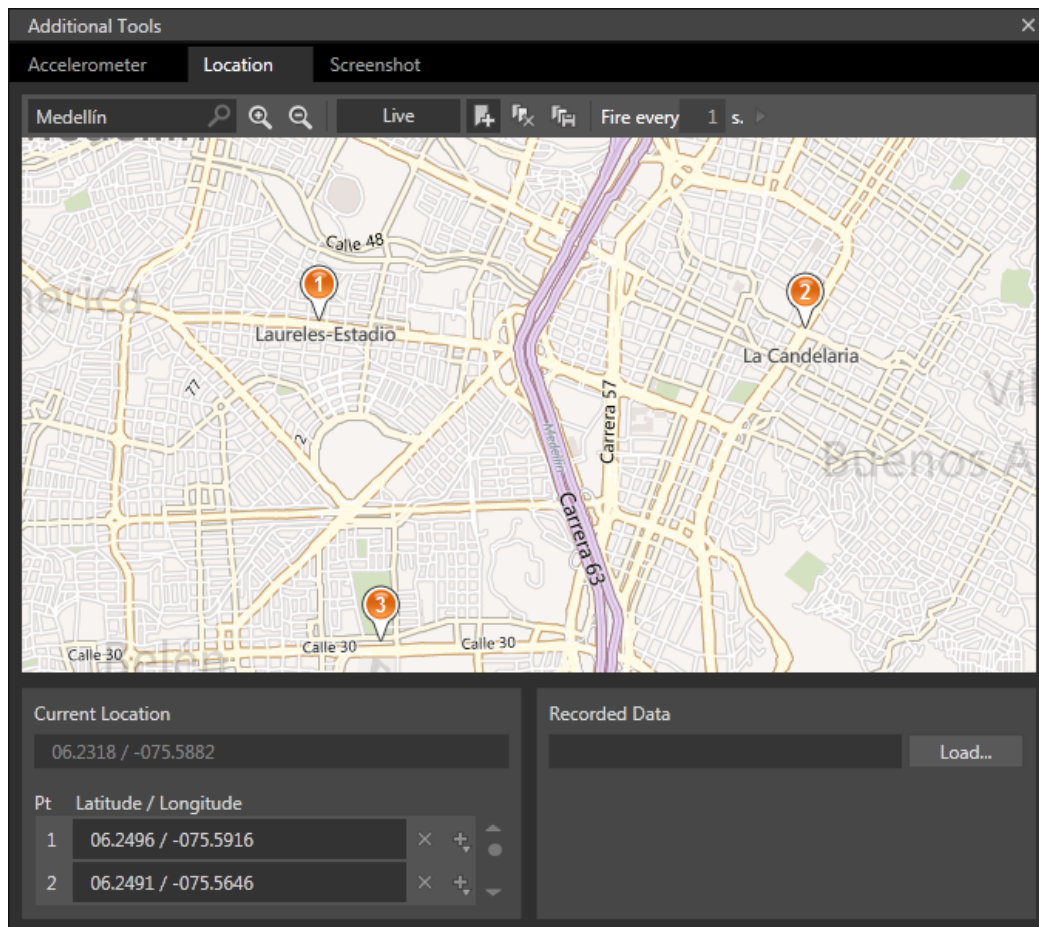
    actualPosition = new Pushpin();
    actualPosition.Template = (ControlTemplate)(App.Current.Resources["PushpinActualPosition"]);
    actualPosition.Location = e.Position.Location;
    actualPosition.Content = "Yo";
    map1.Children.Add(actualPosition);

    map1.SetView(actualPosition.Location, 14);
}
```

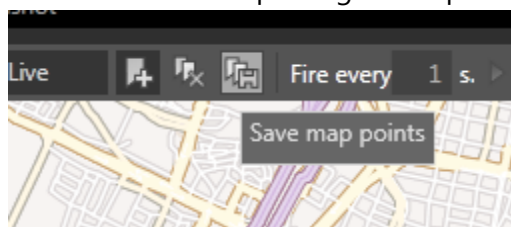
10. Probamos nuestra aplicación y ya tenemos nuestro marcador o pushpin personalizado.



11. Ahora vamos a continuar personalizando los pushpin o marcadores de nuestra aplicación Windows Phone. Para esto vamos a graficar una lista de puntos fijos (en el post) pero que bien pudimos haber obtenido a través de un servicio que haga parte del back end de nuestra aplicación, **al final de este documento se el reto de indicará donde deben estar almacenados esos puntos**, y para obtener esos puntos vamos a usar el emulador.



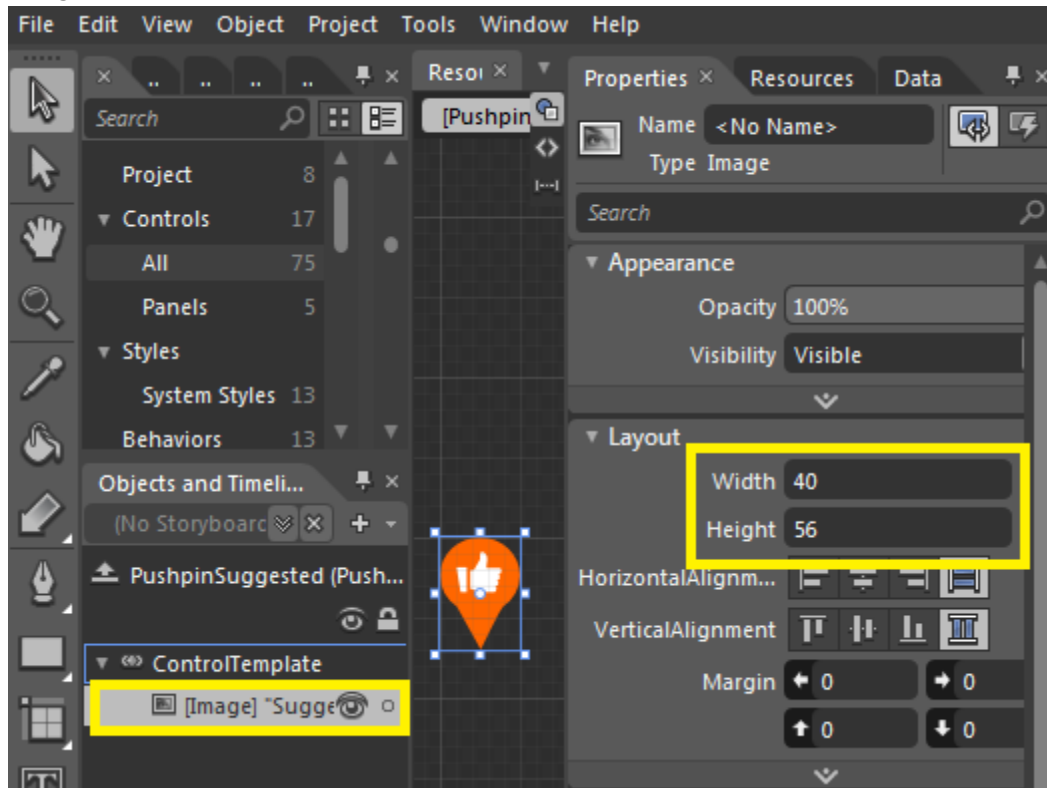
12. Seleccionamos la opción guardar para almacenar las coordenadas en nuestro disco.



13. En el archivo de coordenadas veremos algo como esto.

```
<WindowsPhoneEmulator xmlns="http://schemas.microsoft.com/WindowsPhoneEmulator/
  <SensorData>
    <Header version="1" />
    <GpsData latitude="6.24957407984444" longitude="-75.5916213989258" />
    <GpsData latitude="6.24914747662375" longitude="-75.5645847320557" />
    <GpsData latitude="6.23182709239987" longitude="-75.5881881713867" />
  </SensorData>
</WindowsPhoneEmulator>
```

14. Si recuerdan nuestro objetivo es que los puntos alrededor queden con un estilo diferente a nuestra posición actual, por eso debemos repetir los pasos donde aprendemos a personalizar los pushpin, en el caso de este pushpin que no tiene texto al interior, pueden colocar como contenido en la plantilla simplemente una imagen, pero no olviden colocar las dimensiones del pushpin.



15. Ahora al nivel de la clase de nuestro ejercicio anterior, vamos a declarar para propósitos de este ejercicio un diccionario.

```
public partial class MainPage : PhoneApplicationPage
{
    GeoCoordinateWatcher watcher;
    Pushpin actualPosition;

    Dictionary<int, GeoCoordinate> placesList;
```

16. Luego en el constructor agregamos datos a la lista y vamos a graficarlos cuando la pantalla esté cargada, por lo que debemos suscribirnos al evento Loaded de esta.

```
placesList = new Dictionary<int, GeoCoordinate>();
placesList.Add(1, new GeoCoordinate(6.24957407984444, -75.5916213989258));
placesList.Add(1, new GeoCoordinate(6.24914747662375, -75.5645847320557));
placesList.Add(1, new GeoCoordinate(6.23182709239987, -75.5881881713867));

this.Loaded += new RoutedEventHandler(MainPage_Loaded);
}

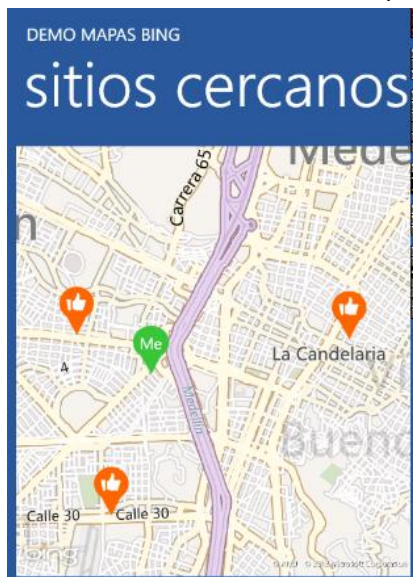
void MainPage_Loaded(object sender, RoutedEventArgs e)
{
}
}
```

17. Y lo implementamos para que muestre los puntos de la lista, usando nuestra plantilla nueva para el pushpin y estamos usando la propiedad **Content** para almacenar la clave, ya que este ejemplo no usamos el Content para mostrar algo como si lo hicimos en el ejercicio anterior .

```
void MainPage_Loaded(object sender, RoutedEventArgs e)
{
    foreach (int key in this.placesList.Keys)
    {
        Pushpin newLocation = new Pushpin();
        newLocation.Template = (ControlTemplate)(App.Current.Resources["PushpinSuggested"]);
        newLocation.Content = key;
        newLocation.Location = this.placesList[key];
        map1.Children.Add(newLocation);
    }

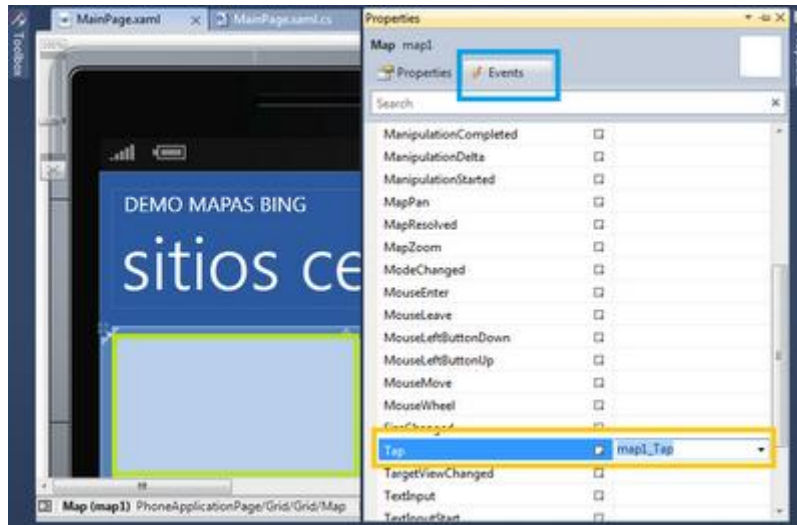
    watcher.Start();
}
}
```

18. De esta forma vemos como aparecen los pushpin alrededor con otro estilo.



## Permitiendo crear nuevos marcadores

1. Ahora vamos a permitir que se agreguen nuevos puntos usando el gesto Tap dentro del Mapa de Bing, para ello vamos al mapa en Visual Studio y en la sección de propiedades en los eventos, creamos el código asociado al Tap dando doble clic.



2. Luego de esto creamos el código que nos permite determinar las coordenadas del mapa usando las coordenadas del Tap

```
private void map1_Tap(object sender, GestureEventArgs e)
{
    var puntoPantalla = new Point(e.GetPosition(map1).X, e.GetPosition(map1).Y);
    var puntoMapa = map1.ViewportPointToLocation(puntoPantalla);

    Pushpin newLocation = new Pushpin();
    newLocation.Template = (ControlTemplate)(App.Current.Resources["NewPushpin"]);
    newLocation.Location = puntoMapa;
    map1.Children.Add(newLocation);
}
```

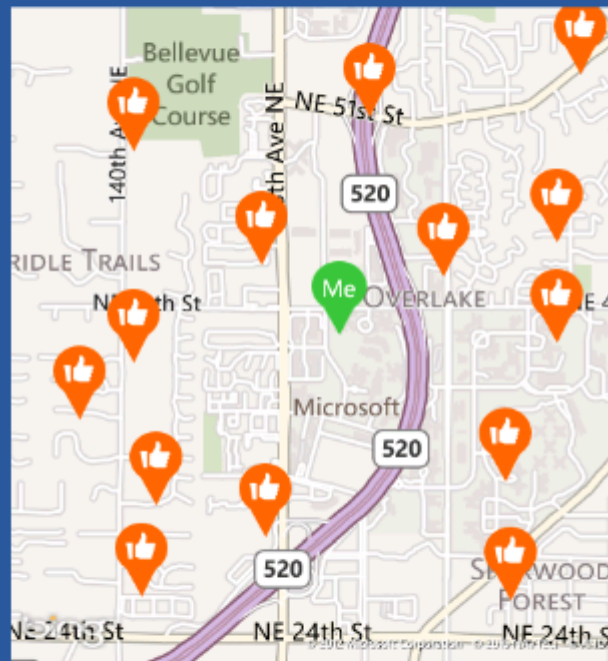
3. Como ve en el punto anterior se hizo referencia a una plantilla con nombre **NewPushpin** que no existe, se debe hacer es crear esta nueva plantilla tal como se le explicó en la sección anterior.
4. De esta forma podemos ver como cada que se da clic o tap en el mapa se crea un nuevo pushpin, si dejamos la misma plantilla que en el anterior, pasará que los puntos que añade el usuario no se distinguen.



NOKIA

DEMO MAPAS BING

# sitios cercanos



location data is available.

