

Windows Phone 8 en acción

Amin Espinoza de los Monteros

Desarrollador y consultor de aplicaciones .NET, Microsoft
Certified Technology Specialist Silverlight y Professional
Developer en Windows Phone



Windows Phone 8 en acción
Amin Espinoza de los Monteros

ISBN: 978-84-941272-0-5
EAN: 9788494127205
BIC: UMS

Copyright © 2013 RC Libros
© RC Libros es un sello y marca comercial registrados

Windows Phone 8 en acción

Reservados todos los derechos. Ninguna parte de este libro incluida la cubierta puede ser reproducida, su contenido está protegido por la Ley vigente que establece penas de prisión y/o multas a quienes intencionadamente reprodujeren o plagiaran, en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución en cualquier tipo de soporte existente o de próxima invención, sin autorización previa y por escrito de los titulares de los derechos del copyright. La infracción de los derechos citados puede constituir delito contra la propiedad intelectual. (Art. 270 y siguientes del Código Penal). Dirijase a CEDRO (Centro Español de Derechos Reprográficos) si necesita fotocopiar o escanear algún fragmento de esta obra a través de la web www.conlicencia.com; o por teléfono a: 91 702 19 70 / 93 272 04 47)

RC Libros, el Autor, y cualquier persona o empresa participante en la redacción, edición o producción de este libro, en ningún caso serán responsables de los resultados del uso de su contenido, ni de cualquier violación de patentes o derechos de terceras partes. El objetivo de la obra es proporcionar al lector conocimientos precisos y acreditados sobre el tema tratado pero su venta no supone ninguna forma de asistencia legal, administrativa ni de ningún otro tipo, si se precisase ayuda adicional o experta deberán buscarse los servicios de profesionales competentes. Productos y marcas citados en su contenido estén o no registrados, pertenecen a sus respectivos propietarios.

RC Libros
Calle Mar Mediterráneo, 2. Nave 6
28830 SAN FERNANDO DE HENARES, Madrid
Teléfono: +34 91 677 57 22
Fax: +34 91 677 57 22
Correo electrónico: info@rclibros.es
Internet: www.rclibros.es
Diseño de colección, cubierta y pre-impresión: Grupo RC
Impresión y encuadernación: Service Point
Depósito Legal: M-11460-2013
Impreso en España

17 16 15 14 13 (1 2 3 4 5 6 7 8 9 10 11 12)

PRÓLOGO

HECHOS

Veamos los hechos.

El primer hecho es que vivimos en un mundo en donde estar conectado a una red gigantesca de datos a los que recurrimos con el fin de entretenernos, aprender, trabajar o visitar nuestras redes sociales es una actividad que ha dejado de ser exclusiva de nuestros escritorios o lugares de trabajo favoritos. Hoy en día queremos hacerlo en el taxi que nos lleva al trabajo, al ir caminando con un amigo, en la cama antes de dormir o en cualquier lugar que a usted estimado lector se le pueda ocurrir.

El segundo hecho es que tener un teléfono de última generación llamado “inteligente” no es suficiente, necesitamos su capacidad de datos en todo momento e igual de importante es el caso de contar con la aplicación adecuada para un fin determinado, para ello se puede ofrecer infinidad de ejemplos como ver un itinerario de vuelo, nuestra colección de logros en los videojuegos, redactar un documento oficial y enviarlo a sus destinatarios e incluso reconocer las estrellas de nuestra bóveda celeste.

El tercer hecho es que tengo una de diez posibilidades de equivocarme al asegurar que tanto su computadora del trabajo como su ordenador cuentan con software de Microsoft, una de las gigantes empresas de software mundiales.

CONCLUSIONES

Ahora, si unificamos los hechos anteriores podremos obtener los cabos que al ser unidos darán como resultado este libro que ahora tiene en sus manos. Las conclusiones son muy simples.

Microsoft quiere seguir en la batalla tecnológica mundial y en un momento determinado debió afrontar el hecho de que el sector móvil es sumamente importante como para no considerar su presencia en este mercado. Hubo varias versiones para poder cumplir con este aspecto, la apuesta actual se llama Windows Phone.

Así como Microsoft concibió una plataforma para competir en el sector móvil, también creó el modelo de mercado que sus dos gigantes competidores ya tenían establecido y con mucha más experiencia, me refiero a Apple con su sistema iOS y a Google con Android. De esta manera instauró el Marketplace de aplicaciones para Windows Phone. Dejando en manos de sus desarrolladores el crecimiento de aplicaciones que los usuarios podrían utilizar en función de sus necesidades.

La tercera conclusión fue la que en mi opinión fue la más acertada. Tomó la decisión de unificar su gran ecosistema de ambientes como Windows, Xbox y el gran ejército de aplicaciones bajo un mismo estilo de diseño para que todo tuviera un común denominador. Esto causó una sorpresa general, ahora todos sus productos tendrían una forma de navegación muy similar, así como su visualización. La recepción ha sido muy bien aceptada al menos hasta ahora, aún falta un gran camino por recorrer pero es indiscutible que el inicio de este “efecto dominó” fue Windows Phone.

Por último, si usted ha adquirido este libro es porque quiere (sino es que ya lo es) formar parte del tercer gran componente que determinará el éxito o fracaso de la plataforma: los desarrolladores. Estos contribuyen a crear aplicaciones para el sistema operativo. En este caso las ideas pueden ser muy divergentes y la intención de este libro es allanar el camino que usted debe recorrer para aprovechar lo mejor posible el potencial del sistema operativo móvil, así como darle rienda suelta a su imaginación conociendo los alcances que la plataforma tiene para ofrecer tanto a usted como desarrollador como a sus usuarios.

LOS ANTECEDENTES DE WINDOWS PHONE

Windows Phone no es ni por asomo la primera opción creada por Microsoft para competir en el sector móvil. De hecho han sido ya numerosos los intentos que ha tenido para lograr posicionarse como una opción destacable y a considerar.

La primera opción que colocó en el mercado fue en el año 2002 con un sistema llamado Pocket PC que desde su nacimiento trató de pelear por obtener el emblemático nombre de Windows Mobile pero este “reconocimiento” no llegó hasta el año siguiente. Una nueva versión que surgió como Windows Mobile 2003. De esta forma estableció completamente su independencia del sistema Windows Embedded.

Windows Mobile (con una gran similitud de interfaz a Windows XP) fue creciendo paulatinamente hasta llegar a su versión 6.5. Para entonces Microsoft habría notado dos grandes curiosidades, la primera fue negativa pues el SO comenzaba a perder cuota de mercado frente a sus nuevos opositores con una interfaz sumamente nueva y reluciente y dispositivos aún más atractivos en cuanto a diseño. La segunda curiosidad fue que sus dispositivos fueron orientándose mucho más hacia el sector empresarial, en este aspecto Microsoft decidió seguir con esa línea en el sistema operativo que comenzaba a gestar y que venía como una renovación absoluta, no como una actualización.

El otro producto que desapareció en aras de ceder su espacio para Windows Phone fue el fallido reproductor musical que Microsoft propuso como competencia al iPod, su nombre era Zune y fue quien comenzó con el estilo visual característico de Windows Phone (y consecuentemente de todas las demás plataformas de Microsoft). Así que podemos atribuirle de manera real este cambio y su herencia a Windows Phone.

ANTES DE INICIAR

Estimado lector, si usted adquirió este libro es para comenzar o perfeccionar sus habilidades en el desarrollo de aplicaciones para Windows Phone, debo aclararle que he hecho énfasis en la creación de aplicaciones genéricas dejando de lado a los videojuegos dado que este es un tema tan complejo que bien valdría la elaboración de un nuevo libro.

AGRADECIMIENTOS

Agradezco muchísimo a la vasta comunidad de desarrolladores que dedican su tiempo a escribir artículos de ciertos temas en sus sitios propios o en comunidades, su enorme contribución me dio la motivación suficiente para comenzar a hacer lo mismo y encontré en compartir conocimientos una gran satisfacción, ustedes me inspiran para seguir haciéndolo.

Es justo dedicar unas líneas a todas las personas a las que dejé de ver o frecuentar por el tiempo que dediqué a este libro, amigos y familia que pacientemente me animaban a dedicarle tiempo a esta obra en lugar de pasarlo con ellos. No tengo palabras para expresar lo enormemente agradecido que estoy con ustedes.

Por último, le agradezco a usted estimado lector, por depositar su confianza en esta obra al adquirirla. Espero que disfrute las lecciones y el aprendizaje de los conceptos aquí descritos tanto como yo he disfrutado escribiéndolos.

1 LAS BASES

MEJORES PRÁCTICAS DE DESARROLLO

Es importante aclarar ciertos aspectos antes de comenzar propiamente con el desarrollo. El primero se trata de cómo utilizar una convención de nombres adecuada para la creación de métodos, variables y objetos con el fin de darle la mejor uniformidad posible a nuestro código. Considere también el muy importante hecho de que seguramente en muchos de sus proyectos podría llegar a trabajar en equipo. En este caso el crear elementos con un nombre adecuado es fundamental para poder evitar confusiones con otros miembros del equipo, así como pérdidas de tiempo innecesarias en discusiones acerca de qué es lo que hace cada elemento escrito en el código fuente común.

DECLARACIÓN DE NOMBRES

Hay dos convenciones principales para la declaración de nombres de cualquier tipo en el momento de escribir un código fuente.

Pascal Casing. La primera letra de todas las palabras es mayúscula y todas las demás son minúsculas.

Ejemplo:

NombreDeElemento

MiSegundoNombre

Camel Casing. La primera letra de todas las palabras, excepto la primera palabra, va en mayúsculas y las demás en minúsculas.

Ejemplo:

nombreDeElemento

miSegundoNombre

Para saber cuándo utilizar cada una de estas simplemente deberá considerar los siguientes escenarios.

Utilice Pascal casing en caso de declarar una clase.

public class MiClase

```
{  
}
```

Cuando se trate de declarar nombres de métodos.

private void MetodoDeClase()

```
{  
}
```

Utilice Camel casing para las variables y argumentos en los métodos.

string primerNombre = string.Empty;

private void MetodoConParametros(string argumentoUno, string argumentoDos)

```
{  
}
```

TABULACIONES Y ESPACIOS

La idea de que usted tenga un orden en los espacios en el momento de escribir su código es que después de muchas líneas le será mucho más cómodo poder ubicar un elemento o un fragmento en particular simplemente con un recorrido visual rápido. En este libro las mejores prácticas para este apartado serán las siguientes.

Los comentarios deben encontrarse al mismo nivel que el código, respetando de esta forma la misma tabulación.

//Esta es la clase que utilizarás en la lista

//No es necesario colocar atributos adicionales

```
public class Elemento
{
    public int Identificador { get; set; }
    public string Nombre { get; set; }
    public string Mensaje { get; set; }
    public string Imagen { get; set; }
}
```

Las llaves {} deben estar en el mismo nivel que el código sobre ellas.

```
public class Elemento
{
    public int Identificador { get; set; }
    public string Nombre { get; set; }
    public string Mensaje { get; set; }
    public string Imagen { get; set; }
}
```

Utilice una línea en blanco para poder separar los segmentos de códigos.

```
private void ObtenerListaDeElementos(UIElement padre, Type tipoPresentacion,
Type tipoVideo, Type tipoImagen, Type tipoNavegador)
{
    int contador = VisualTreeHelper.GetChildrenCount(padre);

    for (int i = 0; i < contador; i++)
    {
        UIElement hijo = (UIElement)VisualTreeHelper.GetChild(padre, i);

        if (hijo.GetType() == tipoPresentacion)
        {
            int valorIndice = master.LayoutRoot.Children.IndexOf(hijo);
            ModificarPosicionElementosPresentacion(valorIndice);
        }
    }
}
```

```
else if (hijo.GetType() == tipoVideo)
{
    int valorIndice = master.LayoutRoot.Children.IndexOf(hijo);
    ModificarPosicionElementosVideo(valorIndice);
}
}
```

Debería dejar una línea de separación entre cada método dentro de la clase.

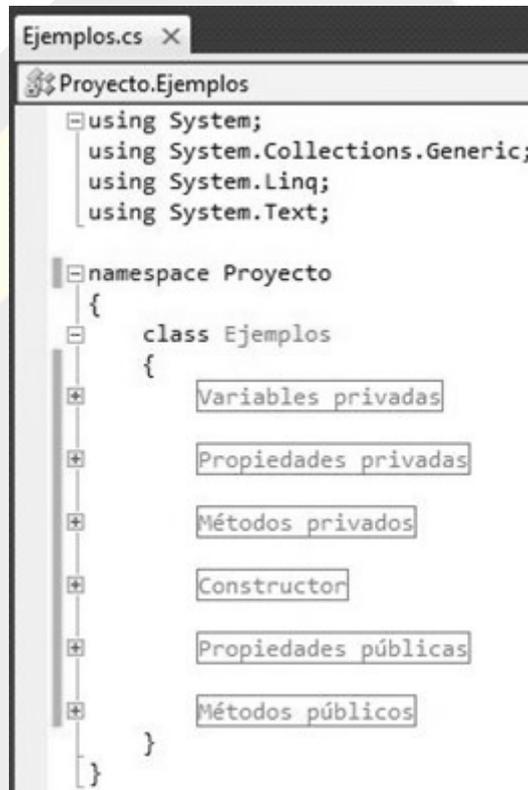
```
private void PrimerMetodo(object argumento)
{
    //Funcionalidad de método
}
```

```
private void SegundoMetodo()
{
    //Funcionalidad de método
}
```

Las llaves que se utilizan para la apertura y cierre de un método deberán encontrarse en una línea separada y no en la misma sentencia de inicio.

```
private void Metodo(object argumento)
{
    if (elemento == true)
    {
    }
}
```

El uso de regiones será muy recurrente y la secuencia para la colocación de todos los elementos involucrados en cualquier ejercicio será de esta manera.



```
Ejemplos.cs x
Proyecto.Ejemplos
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Proyecto
{
    class Ejemplos
    {
        Variables privadas
        Propiedades privadas
        Métodos privados
        Constructor
        Propiedades públicas
        Métodos públicos
    }
}
```

MANEJO Y CREACIÓN DE MÉTODOS

Esta obra ha sido escrita con fines demostrativos y académicos y no para elementos de producción, por ello para una mejor explicación y comprensión los métodos podrán contar con más de **25** líneas de código. En un entorno productivo esto no es recomendable y en caso de que suceda, entonces lo ideal es refactorizar la funcionalidad en métodos más pequeños.

Una de las prácticas más recomendables para la declaración de métodos y que será muy útil para esta obra, consiste en darle a los métodos un nombre adecuado en función de la tarea que realiza. Es decir, si un método va a leer un archivo XML, entonces colocarle el nombre de “LeerDatosXML” podría ser una idea mucho mejor que simplemente “LeerDatos”.

Otra práctica que verá a lo largo de este libro será el uso de métodos destinados a una sola actividad. La idea es poder organizar y reutilizar pequeñas funciones en grandes procesos.

Un aspecto sumamente importante es evitar en todo momento y a toda costa líneas de código estático (comúnmente conocido como **hardcoded**). Para las ocasiones en donde se requiera un valor, la opción preferible será utilizar una variable constante en lugar de utilizar únicamente el valor.

```
private const int numeroUno = 1;
```

```
private void MetodoDeEjemplo()  
{  
    if (numeroUno == 1)  
    {  
    }  
}
```

Utilizando el punto anterior para el caso de los textos al principio del libro comenzaremos con la forma común de trabajar con los textos. En el capítulo 5 veremos la forma correcta de trabajar con el manejo de textos permitiendo además la facilidad de poder traducir una aplicación a múltiples idiomas. Por el momento solo lo tendremos presente y en el capítulo adecuado comenzaremos a implementarlo.

Para el efecto de cualquier sentencia de selección múltiple, la mejor práctica es utilizar un enumerador para enlistar las posibilidades de elección.

```
public enum Objetos
```

```
{  
    pantalla,  
    gabinete,  
    teclado,  
    mouse  
}
```

```
private void ElegirDispositivo(Objetos argumento)
```

```
{  
    switch (argumento)  
    {  
        case Objetos.pantalla:
```

```

        break;
    case Objetos.gabinete:
        break;
    case Objetos.teclado:
        break;
    case Objetos.mouse:
        break;
    default:
        break;
    }
}

```

Por último, evite en todo momento el uso de funcionalidad en un manejador de eventos. Es decir, cuando el usuario presiona un botón se maneja un evento (en este caso podríamos mencionar el evento Tap). Ese evento desencadena ciertas acciones si tiene un manejador de eventos. El objetivo de una buena práctica a seguir es que el manejador de eventos no debe hacer otra cosa más que invocar a un método y no hacer la tarea para la que el método fue creado.

```

private void WindowLoaded(object sender, RoutedEventArgs e)
{
    IniciarAplicacion();
}

private void IniciarAplicacion()
{
    //Funcionalidad de la tarea.
}

```

COMENTARIOS Y DESCRIPCIONES

La costumbre de utilizar comentarios es una de las mejores prácticas que como programador puede utilizar y afortunadamente una de las más recurridas en prácticamente todos los desarrolladores de software. En el caso de los ejercicios realizados en este libro, los ejercicios básicos omitirán los comentarios permitiendo que nos enfoquemos en la funcionalidad y explicando de manera aislada toda situación. Solo me resta recomendarle, apreciado lector, que es una muy buena idea colocar comentarios en todos los ejercicios por realizar para poder recurrir a una sección en particular y no general en el momento de retomar proyectos pasados.

Configuración inicial

El software que va a requerir para la elaboración de todos los ejercicios contenidos en este libro será el siguiente.

Visual Studio Express para Windows Phone.

<http://www.microsoft.com/visualstudio/eng/products/visual-studio-express-for-windows-phone>

En caso de contar con una licencia para utilizar cualquier otra versión de Visual Studio necesitará instalar primero el kit de desarrollo de Windows Phone.

<http://dev.windowsphone.com/en-us/downloadsdk>

Para el desarrollo de aplicaciones nos basaremos en el último lanzamiento del SDK de Windows Phone que corresponde a la versión 8.0 con lo que solo será necesaria esa edición de Visual Studio 2012.

Además de Visual Studio, utilizaremos Expression Blend for Visual Studio, el cual se instala de manera automática en el momento de instalar el IDE en su versión 2012, así no deberá preocuparse por esta parte y solo por considerar de nuevo el hecho de que debe de instalar Visual Studio 2012 para que el complemento se instale ahí y no en una versión separada y Express. Hay que mencionar que esta aplicación puede ser considerada como opcional, pero también le puede ahorrar muchísimo tiempo al crear interfaces, animaciones y dar estilo a los controles.

EL EMULADOR

En las situaciones en que no cuente con un dispositivo físico puede utilizar un emulador para poder ejecutar y probar sus aplicaciones. Esto obviamente cuenta con ciertas ventajas y desventajas; estas son:

Ventajas

Facilidad de pruebas en caso de no contar con un dispositivo físico.

Posibilidad de probar una aplicación en los cuatro posibles escenarios.

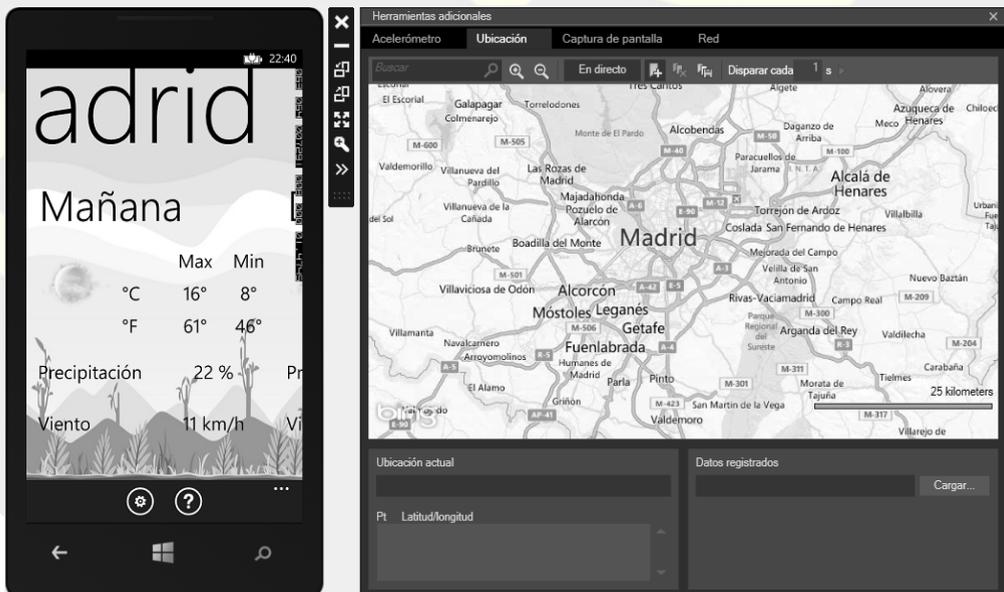
Emulador WVGA 512 MB: Este emulador cuenta con 512 MB de RAM, doble procesador y resolución de 480 x 800 píxeles.

Emulador WVGA: Este emulador cuenta con las mismas características que el anterior pero con la diferencia de usar 1 GB de RAM.

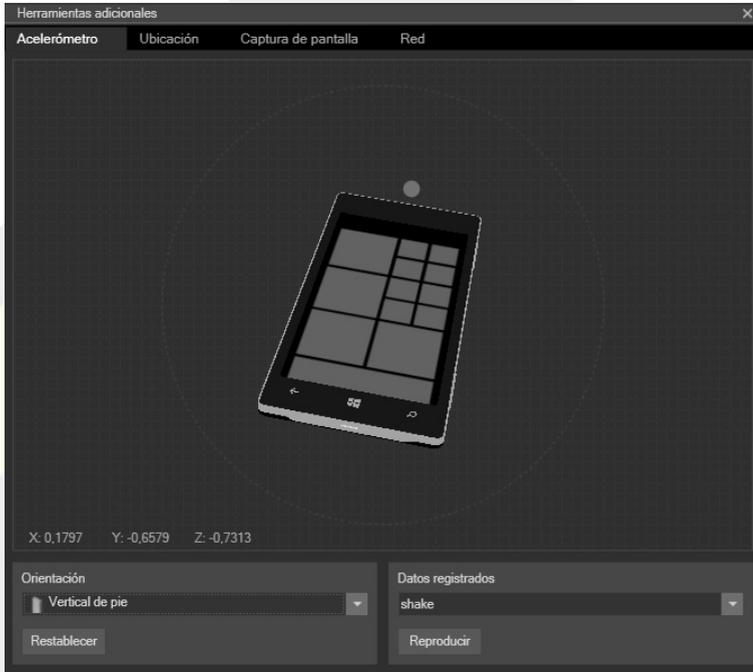
Emulador WXGA: Cuenta con 1 GB de RAM, un doble procesador y su resolución es de 1280 x 768 píxeles.

Emulador 720p: Cuenta con las mismas características del emulador anterior pero con una resolución de 1280 x 720 píxeles.

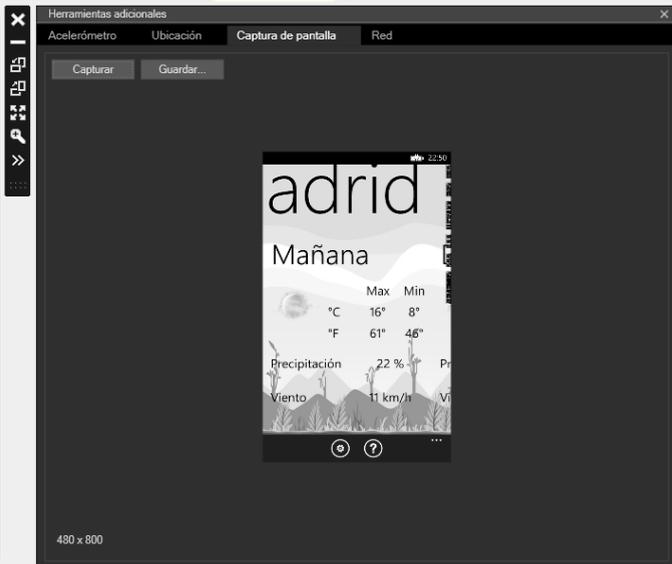
Aun contando con un dispositivo físico, la realización de pruebas con el sensor GPS es un tanto complicada por diversas y lógicas razones. Para solucionarlo, podrá utilizar un emulador de posicionamiento que viene como una herramienta complementaria para el emulador.



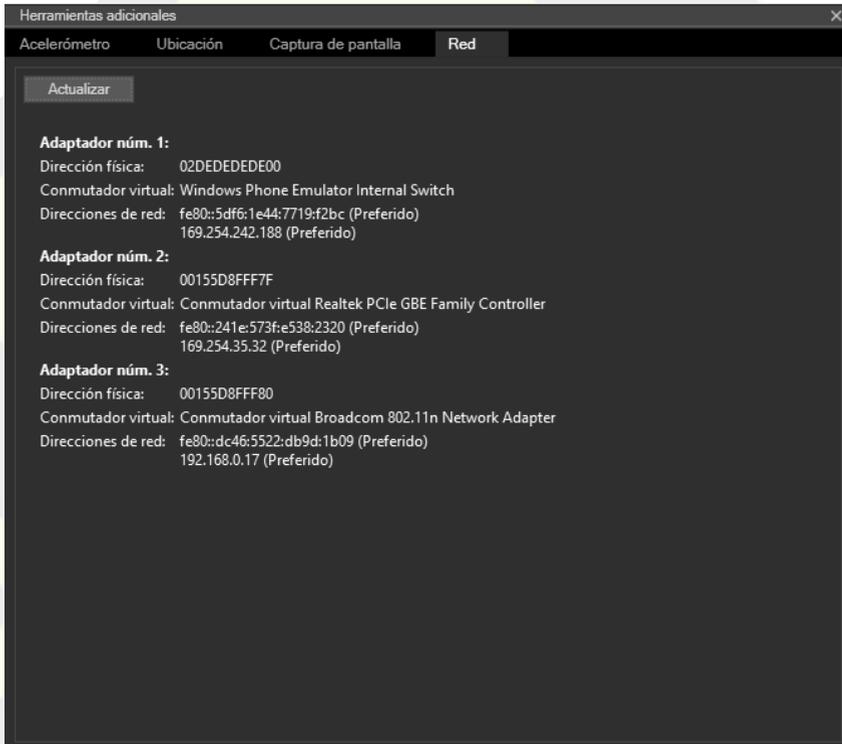
La segunda herramienta complementaria para el emulador es un simulador de acelerómetro que le permitirá darle valores del sensor real para poder simular el movimiento del dispositivo.



Otra herramienta de suma utilidad es la captura de pantallas para aplicaciones en el emulador. Esta utilidad aplica desde el uso de imágenes para su publicidad hasta poder completar los requisitos de publicación para la tienda de aplicaciones.



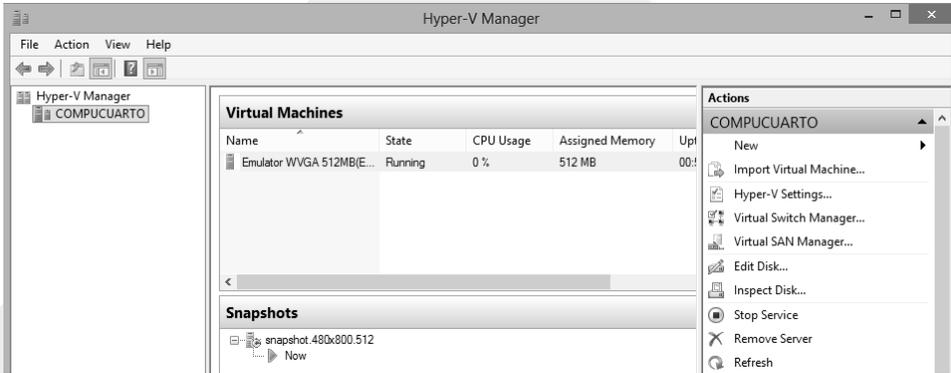
La última herramienta que viene con el emulador es el inventario de puertos de conexión que el mismo emulador está utilizando. Esto le servirá para tener de una manera más visible el control sobre sus conexiones y la idea de saber si es por conectividad que su aplicación pudiera estar fallando.



Desventajas

Hay dos grandes desventajas en el uso del emulador de Windows Phone. La primera corresponde al acceso desde la cámara, la brújula y el giroscopio. Debido a ello es que será prácticamente imposible poder considerar el tipo de aplicaciones de realidad aumentada.

La segunda y más importante a considerar es que a partir del SDK 8.0, cualquier emulador hará uso de Hyper-V que es el entorno de virtualización por excelencia de Microsoft. El usar este entorno es un paso a considerar con mucha cautela por dos razones. La primera es que su equipo de desarrollo debe contar de manera indispensable con la función de virtualización asistida por hardware. Esta función se encuentra en procesadores que incluyen una opción de virtualización ya sea Intel Virtualization Technology o AMD Virtualization.



DESBLOQUEO DE DISPOSITIVOS

Para poder utilizar un dispositivo físico en las pruebas de los proyectos de Windows Phone, es necesario desbloquearlo en modo de desarrollador. La manera de poder lograr esto es usando la herramienta de registro de desarrolladores de Windows Phone.



Además de contar con una cuenta de desarrollador, esta cuenta le dará la facilidad de poder desbloquear hasta 3 dispositivos para su uso en pruebas. Para poder registrarse como desarrollador es necesario ingresar en el centro de desarrolladores de Windows Phone.

Windows Phone Developer Center

El centro de desarrolladores de Windows Phone es indispensable para los desarrolladores de esta plataforma pues cumple con muchas funciones necesarias como las siguientes.

La dirección del centro de desarrolladores es: <http://dev.windowsphone.com/en-us/home>

Documentación y proyectos de ejemplo

Esta sección cumple con dos propósitos primordiales al iniciar el camino en el desarrollo de aplicaciones. Funciona como repositorio de ejemplos que pueden ser útiles en caso de dudas específicas y además, con los mismos ejemplos nos da muchas ideas para poder pensar en nuevas aplicaciones.

Inventario de los dispositivos desbloqueados

Esta sección le permitirá tener el control sobre los dispositivos que están desbloqueados para desarrollo, así como la posibilidad de volver a bloquearlos con el propósito de tener la disponibilidad para otro teléfono.

Forma de ingreso de aplicaciones a la tienda de aplicaciones de Windows Phone

La única manera que usted tendrá de publicar sus aplicaciones en en la tienda de aplicaciones es por este medio. Los requisitos que deberá cumplir para que su aplicación sea publicada serán los siguientes:

- Un alias para la aplicación.
- Una categoría.
- Un precio base (aquí es donde puede considerar si su aplicación es de paga o gratuita).
- Distribución de mercado (cuál es el mercado o mercados que piensa alcanzar con su aplicación).
- Alcance de la aplicación (si la aplicación tendrá un alcance público o será solo una versión beta).
- El archivo .XAP generado a partir de la compilación de la aplicación.
- Descripción y lenguaje.
- Una imagen en formato PNG de 300 x 300 píxeles.
- Una imagen en formato PNG de 1000 x 800 píxeles.
- De una a ocho capturas de pantalla de la aplicación con las siguientes resoluciones opcionales:

Resolución	Píxeles	Versión
WVGA	480 x 800 píxeles	Windows Phone OS 7.1 Windows Phone 8
WXGA	768 x 1280 píxeles	Windows Phone 8
HD720p	720 x 1280 píxeles	Windows Phone 8

Para profundizar más sobre el tema y considerando que este paso es el último dentro de todo el proceso de su aplicación, en el capítulo 12 se explicará con mayor detalle el procedimiento a seguir.

Seguimiento de aplicaciones publicadas en la tienda de aplicaciones de Windows Phone

En este apartado podrá seguir monitoreando sus aplicaciones al estar ya publicadas. Además de poder modificar cualquiera de sus opciones como su costo o cualquier actualización. El objetivo principal es no considerar que el proceso haya terminado al publicar la aplicación, sino que deberá seguir considerando mejoras en su proyecto y cualquier corrección de errores, así como su difusión y alcance a más usuarios.

Tienda de aplicaciones Windows Phone

En la tienda de aplicaciones de Windows Phone los usuarios podrán encontrar toda la gama de aplicaciones que podrán ser capaces de instalar en sus dispositivos. Estas aplicaciones se dividen en cuatro tipos de opciones, las cuales son:

Aplicaciones gratuitas

Se trata de las aplicaciones que el usuario podrá descargar sin pagar nada por tenerlas, en este modelo de aplicaciones se pueden obtener ganancias por medio de la publicidad o simplemente por asociarla a una aplicación mejorada que incluya un costo. Aquí es donde radica aproximadamente el 67 por ciento de las aplicaciones en general.

Aplicaciones de prueba

Las aplicaciones de prueba son aplicaciones de pago pero antes de ser compradas, permiten su uso ya sea por tiempo ilimitado o con cierta cantidad de funciones restringidas. En cualquier caso, es decisión del usuario poder elegir con un mejor criterio la posibilidad de comprarla o no ya que ha visto su funcionalidad. Dentro de esta categoría radica un 10 por ciento de las aplicaciones aproximadamente.

Aplicaciones de cobro

Las aplicaciones de cobro pueden variar dependiendo del nivel de complejidad de la aplicación, y por supuesto, de su reputación. Esto hará que los costos puedan oscilar desde un euro hasta los quinientos. En este punto es necesario que usted como desarrollador deberá evaluar la capacidad de compra que pueda tener su aplicación. Es lógico pensar que cuanto más multifuncional y capaz de realizar tareas complejas sea, más elevado podrá ser su costo.

Aplicaciones beta

La funcionalidad de publicación de aplicaciones beta en la tienda de aplicaciones de Windows Phone obedece a un objetivo principal: las pruebas de la aplicación en diferentes usuarios.

Aplicaciones empresariales

Hasta la versión 7.1 del SDK de Windows Phone no existía una manera factible para poder desplegar aplicaciones empresariales de una manera sencilla, era necesario desbloquear todos los dispositivos involucrados con una cuenta de desarrollo. Este impedimento traía un gran coste a las empresas interesadas, por ello apareció una herramienta llamada Chevron que permitía la posibilidad de desbloquear el teléfono sin todos los beneficios de contar con una cuenta de desarrollador. Pero con esta versión será posible poder publicar aplicaciones de manera privada y solo puedan ser descargadas por medio de un enlace privado. De esta manera se cumple una de las más amplias necesidades para la plataforma, dado que un gran porcentaje del uso de los dispositivos móviles está destinado al desempeño laboral.

Utilidades

Al igual que cualquier teléfono considerado “inteligente” en nuestros tiempos. Cualquier Windows Phone debe contar con ciertos requerimientos al salir al mercado como una amplia variedad de sensores que pueden ser aprovechados en las aplicaciones de terceros. Uno de esos sensores es el acelerómetro, que permite al usuario la posibilidad de distinguir entre una posición vertical u horizontal. Por último, todos los dispositivos deberán contar con tres botones colocados en la parte inferior del teléfono. Sus usos son diferentes pero suficientes para no ver la necesidad de utilizar nada más.

SENSORES DEL DISPOSITIVO

Todos los terminales que cuenten con el sistema operativo deben de cumplir con las directivas establecidas por Microsoft para poder completar el proceso de aprobación. Hay algunos sensores obligatorios y algunos otros opcionales. Por otro lado aunque veremos en el capítulo 3 en profundidad este tema es importante mencionar que muchos elementos se encontraban fuera de nuestro alcance como desarrolladores ordinarios como el de Bluetooth; sin embargo, a partir de la versión 8 del sistema operativo y del SDK tenemos un rango mucho más amplio para poder trabajar, para todos estos elementos que es posible que usted pueda utilizar es lo que vamos a enfocar más adelante.

ORIENTACIÓN

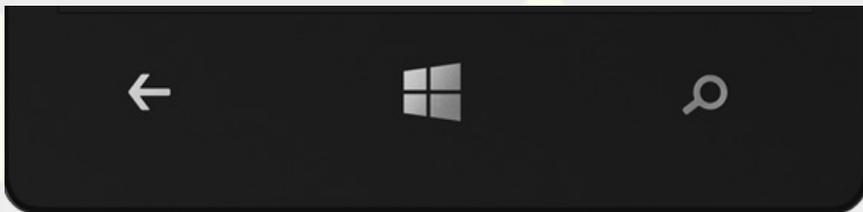
Hay dos tipos de orientación en los que el teléfono puede estar en el momento de ejecutar una aplicación. Dependiendo de su posición es posible que la interfaz se acomode para acoplarse mejor. La posición vertical (portrait) u horizontal (landscape) son opcionales; es decir, usted podrá fijar una posición única de acuerdo con el uso de su aplicación o podrá permitir el manejo de ambas posiciones.





LOS BOTONES

De manera esencial, cualquier dispositivo con Windows Phone debe contar en la parte inferior con tres botones sin importar si son vía hardware o software.



Botón de retroceso: Este es el botón más multifuncional de los tres. El principal uso, como es de esperar, permite al usuario navegar hacia la pantalla anterior. El teléfono va guardando de manera secuencial cada una de las pantallas de las aplicaciones así como las páginas web visitadas, al ir regresando entre aplicaciones, este registro va desapareciendo hasta llegar al menú principal y no deja ninguna otra aplicación abierta.

En el menú principal, mantener el botón de retroceso presionado abrirá el conmutador de aplicaciones, esto es, un menú donde se encuentran todas las aplicaciones abiertas y poder colocar una seleccionada en primer plano.

En el caso de las aplicaciones, cumplirá con dos propósitos si omito el primero ya mencionado. Uno de esos es que debe desaparecer cualquier tipo de menú contextual o aviso si existiera alguno en la pantalla. El segundo es más recurrido en los juegos y es la opción de permitir que el juego sea puesto en pausa.

Botón de menú principal o de inicio: El botón de menú principal cumple exclusivamente con dos funciones. La primera es permitirle al usuario regresar de manera inmediata al menú principal del teléfono mandando así a la aplicación que estuviera en uso a un segundo plano. La segunda función es la de permitir al usuario el uso de comandos de voz, para esto solo bastará mantener el botón de inicio presionado tres segundos.

Botón de búsqueda: El botón de búsqueda enviará cualquier aplicación a un segundo plano y colocará de manera inmediata una pantalla del buscador Bing, así como diversas opciones de uso dependiendo de la región e idioma en el que el dispositivo esté configurado.

