

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE  
TELECOMUNICACIÓN

UNIVERSIDAD POLITÉCNICA DE CARTAGENA



TRABAJO FIN DE MÁSTER

**Solución integral para pc, tablet y smartphone dirigida a  
la gestión y lectura de libros electrónicos implementado en  
C# y XAML.**



AUTOR: José Luis Marín Marín  
DIRECTOR: Antonio Javier García Sánchez

Junio 2015



Autor	José Luis Marín Marín
E-mail del Autor	<a href="mailto:Joseluisxs@hotmail.com">Joseluisxs@hotmail.com</a>
Director	Antonio García Sánchez
E-mail del Director	<a href="mailto:antoniojavier.garcia@upct.es">antoniojavier.garcia@upct.es</a>
Título del TFM	<i>Solución integral para pc, tablet y smartphone dirigida a la gestión y lectura de libros electrónicos implementado en C# y XAML.</i>
Resumen	<p>Este proyecto consiste en el desarrollo de una aplicación de gestión y lectura de libros electrónicos para el sistema operativo Windows, tanto libres como protegidos por el sistema Adobe DRM (gestión de derechos digitales). Para ello se emplea la plataforma .NET de Microsoft y el lenguaje de programación C# con XALM (<i>Lenguaje Extensible de Formato para Aplicaciones</i>).</p> <p>Esta es una aplicación universal Windows, es decir, que la misma puede ejecutarse en todos los dispositivos con Windows, tales como ordenadores, tabletas electrónicas y teléfonos inteligentes con el sistema operativo Windows 8.1 o superior.</p> <p>Esta aplicación integra la aplicación Web de OdiloTK existente y desarrollada por Odilo, mediante un contenedor Web. De forma que los usuarios puedan gestionar los libros y descargarlos mediante la OdiloTK.</p>
Titulación	Máster en Ingeniería de Telecomunicación
Intensificación	-
Departamento	Tecnologías de la información y las comunicaciones
Fecha de presentación	Julio 2015



## Contenido

Capítulo I – Introducción acerca del proyecto .....	4
Objetivos .....	4
Descripción .....	5
Servicios que usa la aplicación.....	6
Capítulo II – Aplicaciones para Windows 8.1 .....	7
Windows Phone .....	7
Interfaz de usuario .....	7
Tienda(Store) .....	8
Desarrollo de aplicaciones .....	8
.NET .....	9
Consideraciones .....	9
Componentes.....	10
Common Language Runtime.....	10
C# .....	11
Características del lenguaje C# .....	12
Windows Phone SDK .....	13
WinRT .....	13
Ventajas de esta plataforma.....	14
Otras alternativas .....	15
Capítulo III – Entorno de desarrollo .....	17
Definición .....	17
Visual Studio .....	17
Windows Phone SDK tools .....	18
Instalación Windows 8.1 SDK .....	19
Requisitos del sistema .....	20
Desarrollo de aplicaciones en Visual Studio IDE .....	20
Capítulo IV – Medios empleados durante la implementación y pruebas .....	29
Capítulo V – Introducción a las BBDD relacionales .....	30
Características .....	30
Elementos .....	30
Estructura .....	32
Manipulación de la información .....	33
Ventajas y desventajas .....	33
Diseño de las bases de datos relacionales .....	34
Capítulo VI – Introducción al motor de BBDD SQLite .....	35
Características de SQLite .....	36
Introducción al uso de SQLite .....	37
Capítulo VII – Características de la aplicación .....	40
Introducción .....	40
Aplicación localizada.....	41
Kit de herramientas multilingües .....	42



Diagrama de flujo de la aplicación .....	43
Páginas de la aplicación .....	44
Activación .....	44
Biblioteca .....	46
Información .....	53
LectorDRM .....	54
OdiloTK .....	57
Clases de la aplicación .....	63
Almacenamiento .....	64
Biblioteca .....	64
Funciones .....	64
Libro .....	65
SQLite/SQLiteAsync .....	66
App .....	67
Asistente personal Cortana .....	68
Integración de aplicaciones con Cortana .....	70
Funcionamiento de OdiloTK con Cortana .....	72
Capítulo VIII - Aplicaciones universales .....	76
Introducción. ....	76
Aplicaciones universales en Windows 10 .....	77
Capítulo IX – Diseño de la aplicación.....	78
Introducción a la interfaz gráfica de usuario .....	78
Interfaz gráfica en .NET.....	78
Componentes de las aplicaciones Windows .....	79
System.Windows.Forms .....	79
La clase Form .....	80
Interfaz metro en Windows .....	81
Características de una aplicación metro .....	81
Anexo – Documentación técnica .....	83
Conclusión .....	85
Manual de usuario .....	86
Instalación (ejemplo para Nokia) .....	86
Desinstalación (ejemplo para Nokia) .....	87
Requisitos mínimos .....	87
Funcionamiento de la aplicación .....	88
Problemas y soluciones. ....	93
Glosario de términos utilizados. ....	96
Referencias .....	97



## Capítulo I – Introducción acerca del proyecto

### *Objetivos*

Se desarrolla una aplicación para la **gestión y lectura de libros electrónicos**, tanto libres como protegidos por el sistema **Adobe DRM** (gestión de derechos digitales).

Esta es una **aplicación universal Windows**, es decir, que la misma puede ejecutarse en todos los dispositivos con Windows, tales como ordenadores, tabletas electrónicas y teléfonos inteligentes con el sistema operativo Windows 8.1 o superior.

Esta aplicación integra la aplicación **Web de OdiloTK** existente y desarrollada por Odilo, mediante un contenedor Web. De forma que los usuarios puedan gestionar los libros y descargarlos mediante la OdiloTK.

La aplicación integra **dos lectores** para libros electrónicos: en primer lugar el **lector Web Nubereader** desarrollado por Odilo y accesible directamente desde OdiloTK. Además incorpora un **lector interno** de la aplicación basado en el sistema **DRM de Adobe**, en el que también se van a poder leer libros libres en formato ePub y PDF.



Icono (tile) de la aplicación

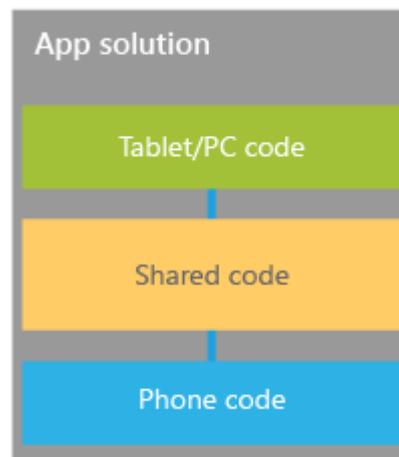
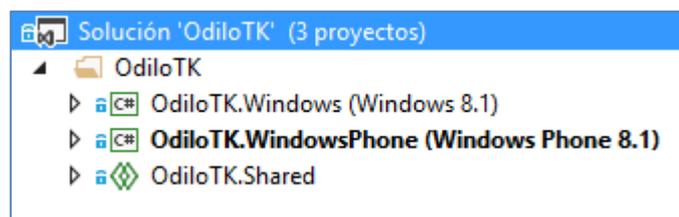




## Descripción

La aplicación se ha desarrollado mediante el entorno de desarrollo Microsoft **Visual Studio 2013** (actualización 4) y el lenguaje de programación C# de la plataforma .NET.

Se ha creado una solución del tipo “**Aplicación universal**”, que permite el diseño simultáneo para ordenadores, tabletas o teléfonos con sistema operativo **Windows 8.1** o superior. Este se compone a su vez de tres partes: el proyecto de Windows 8.1 (escritorio), el proyecto de Windows Phone 8.1 (teléfono) y el **proyecto compartido**, que contiene la **mayor parte del código** de nuestra aplicación.



Solución aplicación universal

El diseño de la interfaz gráfica de usuario se realiza mediante XAML (Lenguaje Extensible de Formato para Aplicaciones), que es un lenguaje declarativo basado en XML. El código C# de cada página XAML, se engloba bajo esta.

Para la implementación se utiliza la **plataforma WinRT**, que es la abreviación de Windows Runtime, un modelo de programación implementado y presentado por Microsoft y que facilita el desarrollo de aplicaciones Software con estilo Metro para su sistema operativo Windows 8.



## ***Servicios que usa la aplicación***

En primer lugar, la aplicación utiliza la **comunicación a través de internet** (WiFi o datos móviles) para el acceso a la aplicación Web OdiloTK y también para poder comprobar que los certificados de los libros electrónicos protegidos con AdobeDRM son válidos y no han caducado. Sin embargo, permite leer sin conexión los libros que no están protegidos.

Para los libros protegidos, se descarga primero un archivo XAML que contiene la información del libro y después contacta con el servidor de DRM. **Es necesaria una licencia válida** de algún proveedor de DRM como Adobe (u otro, como Odilo) para poder descargar/abrir el libro.

Los libros se almacenan en la memoria local del dispositivo y para su gestión interna se utiliza una **base de datos local SQL**, en concreto se utiliza **SQLite** para crear la BBDD, ya que es un estándar mundialmente conocido e integrado en las diferentes plataformas que existen en la actualidad.



La aplicación utiliza el asistente personal **Cortana** de Microsoft, que reconoce el lenguaje natural (sin necesidad de utilizar comandos concretos) y utiliza Bing, Yelp y Foursquare como bases de datos. Sustituye a la búsqueda integrada de Bing en Windows Phone 8.1 y se activa pulsando el botón de búsqueda. A partir de la actualización a Lumia Denim, Cortana puede ser activada diciendo "Ey Cortana". Es el primer asistente que **permite a los desarrolladores integrarse con él**, permitiendo utilizar sus aplicaciones mediante la voz.





## Capítulo II – Aplicaciones para Windows 8.1

### *Windows Phone*

Windows Phone es un sistema operativo móvil desarrollado por Microsoft, como sucesor de la plataforma Windows Mobile. A diferencia de su predecesor, está enfocado en el mercado de consumo generalista en lugar del mercado empresarial. Con Windows Phone, Microsoft ofrece una nueva interfaz de usuario que integra varios servicios propios como OneDrive, Skype y Xbox Live en el sistema operativo.

En primavera de 2014 se lanzó al mercado la última versión del sistema operativo, Windows Phone 8.1 solo para nuevos dispositivos, debido a un cambio completo en el kernel que lo hace incompatible con dispositivos basados en la versión anterior, fragmentando de esta forma el mercado de aplicaciones disponibles. Esta versión incluye nuevas funciones que de acuerdo a Microsoft lo harán competitivo con sistemas operativos como iOS de Apple o Android de Google.

Windows Phone 8.1 es una nueva versión del sistema operativo solo para nuevos dispositivos. Entre las nuevas características se incluyen:

- Multitarea para aplicaciones de terceros
- Uso como unidad de almacenamiento MTP
- Cifrado de datos almacenados
- Aplicaciones en código nativo
- Actualizaciones directamente en el teléfono (sin usar el PC)
- Captura de pantalla
- Búsqueda dentro de una página web (funcionalidad que fue eliminada en la actualización Mango)
- Soporte para VPN, Wi-Fi Direct y Bluetooth 4.0
- Centro de notificaciones
- Teclado Word Flow
- Aplicaciones universales, para integrar Windows Phone, Windows y Xbox One
- Asistente personal: Cortana

### **Interfaz de usuario**

Windows 8.1 cuenta con una nueva interfaz de usuario, llamada Metro. La pantalla de inicio, llamada "Start Screen", se compone de "Live Tiles", mosaicos dinámicos que son enlaces a aplicaciones, características, funciones y objetos individuales (como contactos, páginas web, o archivos multimedia) y que muestran información útil y personalizada para el usuario. Estos mosaicos se actualizan frecuentemente manteniendo informado de cualquier cambio al usuario. La información que se muestra en los mosaicos dinámicos puede ser desde



llamadas, mensajes recibidos, correos electrónicos pendientes, citas previstas, juegos o enlaces rápidos a aplicaciones. La pantalla de inicio y la posición de los mosaicos dinámicos se puede personalizar pulsando y arrastrando los mosaicos a la posición que se desee.

Windows Phone utiliza tecnología multitáctil (Significa que se puede usar con más de un dedo). La interfaz por defecto de Windows Phone tiene un estilo visual negro que prolonga la batería en pantallas OLED ya que los píxeles negros no emiten tanta luz como otros, por lo tanto no gastan tanta batería. El usuario puede elegir un tema claro y también cambiar a varios colores tales como rojo, mango, azul, verde, marrón, rosado, lima, magenta, morado, verde azulado y en la serie de nokia lumia el color nokia blue.

## Tienda (Store)

La tienda de Windows se utiliza para distribuir digitalmente música, contenido de vídeo, podcasts y aplicaciones de terceros para los teléfonos con Windows Phone. El mercado es accesible desde el hub de la tienda de los dispositivos. El mercado está gestionado por Microsoft, que incluye un proceso de aprobación para cada aplicación de terceros, donde se evalúan varios factores estrictos de contenido y performance. Por ejemplo, no se permite contenido que incluya discriminación de ningún tipo, uso de drogas, alcohol o tabaco, o que incluya material sexualmente sugestivo. Desde marzo de 2012, el Windows Phone Store está disponible en 54 países.

## Desarrollo de aplicaciones

El desarrollo de aplicaciones para Windows Phone puede hacerse empleando tres tipos de implementaciones:

- **Microsoft Silverlight** que permite realizar aplicaciones que contengan transiciones y efectos visuales. Silverlight permite el desarrollo de aplicaciones basadas en XAML. Silverlight para Windows Phone incluye el Microsoft .NET Compact Framework, que hereda de la arquitectura .NET Framework, el CLR y la ejecución de código administrado, soporta un subconjunto de las librerías de clases de .NET Framework y contiene clases diseñadas exclusivamente para .NET Compact Framework. Este soporte incluye el Base Class Library, una colección de clases que soportan lectura y escritura de ficheros, manipulación XML y manejo de gráficos. Cada aplicación que es ejecutada en Windows Phone OS 7.0 CTP se ejecuta dentro de un proceso en el motor de ejecución .NET Compact Framework<sup>44</sup>
- **Microsoft XNA Framework** es una implementación nativa de .NET Compact Framework que incluye un amplio conjunto de bibliotecas de clases, específicos para el desarrollo de juegos, por ejemplo para el manejo de dispositivos de entrada, tratamiento de sonidos y vídeos, carga de modelos y texturas, uso de ficheros de forma transparente a la plataforma en la que se ejecute, desarrollo de juegos online, etc...



Permite desarrollar juegos para Windows Phone OS 7.0 CTP, Xbox 360, Zune HD y Windows 7

• **WinRT (Windows Runtime)** es la abreviación de Windows Runtime, un modelo de programación implementado y presentado por Microsoft en el año 2011 y que facilita el desarrollo de aplicaciones Software con estilo Metro para su sistema operativo Windows 8. Desde 2014, se utiliza también para el desarrollo de aplicaciones móviles para el sistema Windows Phone 8.1 o superior.



## ***.NET***

**.NET** es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

**.NET** podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Oracle Corporation y a los diversos framework de desarrollo web basados en PHP. Su propuesta es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones –o como la misma plataforma las denomina, soluciones– permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.

## **Consideraciones**

La plataforma **.NET** de Microsoft es un componente de software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma. Esta solución es el producto principal en la oferta de Microsoft, y pretende ser utilizada por la mayoría de las aplicaciones creadas para la plataforma Windows.



.NET Framework se incluye en Windows Server 2008, Windows Vista y Windows 7. De igual manera, la versión actual de dicho componente puede ser instalada en Windows XP, y en la familia de sistemas operativos Windows Server 2003. Una versión "reducida" de .NET Framework está disponible para la plataforma Windows Mobile, incluyendo teléfonos inteligentes.

La norma (incluido en ECMA-335, ISO/IEC 23271) que define el conjunto de funciones que debe implementar la biblioteca de clases base (BCL por sus siglas en inglés, tal vez el más importante de los componentes de la plataforma), define un conjunto funcional mínimo que debe implementarse para que el marco de trabajo sea soportado por un sistema operativo. Aunque Microsoft implementó esta norma para su sistema operativo Windows, la publicación de la norma abre la posibilidad de que sea implementada para cualquier otro sistema operativo existente o futuro, permitiendo que las aplicaciones corran sobre la plataforma independientemente del sistema operativo para el cual haya sido implementada. El Proyecto Mono emprendido por Xamarin pretende realizar la implementación de la norma para varios sistemas operativos adicionales bajo el marco del código abierto.

## Componentes

Los principales componentes del marco de trabajo son:

- El conjunto de lenguajes de programación.
- La *biblioteca de clases base* o *BCL*.
- El *entorno común de ejecución para lenguajes*, o *CLR* por sus siglas en inglés.

Debido a la publicación de la norma para la infraestructura común de lenguajes (CLI por sus siglas en inglés), el desarrollo de lenguajes se facilita, por lo que el marco de trabajo .NET soporta ya más de 20 lenguajes de programación y es posible desarrollar cualquiera de los tipos de aplicaciones soportados en la plataforma con cualquiera de ellos, lo que elimina las diferencias que existían entre lo que era posible hacer con uno u otro lenguaje.

Algunos de los lenguajes desarrollados para el marco de trabajo .NET son: C#, Visual Basic .NET, Delphi (Object Pascal), C++, F#, J#, Perl, Python, Fortran, Prolog (existen al menos dos implementaciones, el P#1 y el Prolog.NET2 ), Cobol y PowerBuilder.

## Common Language Runtime

El CLR es el verdadero núcleo del framework de .NET, entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios del sistema operativo (W2k y W2003). Permite integrar proyectos en distintos lenguajes soportados por la plataforma .Net, como C++, Visual Basic, C#, entre otros.

La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .NET en un código intermedio, el CIL (Common Intermediate



Language) antes conocido como MSIL (Microsoft Intermediate Language), similar al BYTECODE de Java. Para generarlo, el compilador se basa en la especificación CLS (Common Language Specification) que determina las reglas necesarias para crear el código MSIL compatible con el CLR.

Para ejecutarse se necesita un segundo paso, un compilador JIT (Just-In-Time) es el que genera el código máquina real que se ejecuta en la plataforma del cliente. De esta forma se consigue con .NET independencia de la plataforma de hardware. La compilación JIT la realiza el CLR a medida que el programa invoca métodos. El código ejecutable obtenido se almacena en la memoria caché del ordenador, siendo recompilado de nuevo sólo en el caso de producirse algún cambio en el código fuente.



Estructura interna del entorno de ejecución en lenguaje común.

## C#

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

El nombre C Sharp fue inspirado por la notación musical, donde '#' (sostenido, en inglés sharp) indica que la nota (C es la nota do en inglés) es un semitono más alta, sugiriendo que C# es superior a C/C++. Además, el signo '#' se compone de cuatro signos '+' pegados.

Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco Mono - DotGNU,



actualmente Xamarin, el cual genera programas para distintas plataformas como Windows, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

```
using System;

public class Ejemplo {
    public static void Main(string[] args) {
        Console.WriteLine("Hola mundo");
    }
}
```

Ejemplo "Hola mundo"

## Características del lenguaje C#

La sintaxis de C# es muy expresiva, pero también es sencilla y fácil de aprender. La sintaxis de C# basada en signos de llave podrá ser reconocida inmediatamente por cualquier persona familiarizada con C, C++ o Java. Los desarrolladores que conocen cualquiera de estos lenguajes pueden empezar a trabajar de forma productiva en C# en un plazo muy breve. La sintaxis de C# simplifica muchas de las complejidades de C++ y proporciona características eficaces tales como tipos de valor que admiten valores NULL, enumeraciones, delegados, expresiones lambda y acceso directo a memoria, que no se encuentran en Java. C# admite métodos y tipos genéricos, que proporcionan mayor rendimiento y seguridad de tipos, e iteradores, que permiten a los implementadores de clases de colección definir comportamientos de iteración personalizados que el código cliente puede utilizar fácilmente. Las expresiones Language-Integrated Query (LINQ) convierten la consulta fuertemente tipada en una construcción de lenguaje de primera clase.

Como lenguaje orientado a objetos, C# admite los conceptos de encapsulación, herencia y polimorfismo. Todas las variables y métodos, incluido el método Main que es el punto de entrada de la aplicación, se encapsulan dentro de definiciones de clase. Una clase puede heredar directamente de una clase primaria, pero puede implementar cualquier número de interfaces. Los métodos que reemplazan a los métodos virtuales en una clase primaria requieren la palabra clave override como medio para evitar redefiniciones accidentales. En C#, una struct es como una clase sencilla; es un tipo asignado en la pila que puede implementar interfaces pero que no admite la herencia.

Además de estos principios básicos orientados a objetos, C# facilita el desarrollo de componentes de software a través de varias construcciones de lenguaje innovadoras, entre las que se incluyen las siguientes:

- Firmas de métodos encapsulados denominadas delegados, que habilitan notificaciones de eventos con seguridad de tipos.



- Propiedades, que actúan como descriptores de acceso para variables miembro privadas.
- Atributos, que proporcionan metadatos declarativos sobre tipos en tiempo de ejecución.
- Comentarios en línea de documentación XML.
- Language-Integrated Query (LINQ) que proporciona funciones de consulta integradas en una gran variedad de orígenes de datos.

## ***Windows Phone SDK***

Windows Phone SDK 8.1 es un completo entorno de desarrollo que puede usar para crear aplicaciones y juegos para Windows Phone 8.1, 8 y Windows Phone 7.5. Windows Phone SDK puede funcionar como complemento de Visual Studio 2013 Community(gratuito), Professional, Premium o Ultimate. Con el SDK, puede usar sus conocimientos de programación y el código que ya tiene para generar aplicaciones administradas o con código nativo. Asimismo, el SDK contiene varios emuladores y herramientas adicionales para generar perfiles y probar la aplicación de Windows Phone en condiciones reales. Este SDK sirve para desarrollar aplicaciones con **Silverlight y WinRT**.



Logo Windows Phone SDK

## ***WinRT***

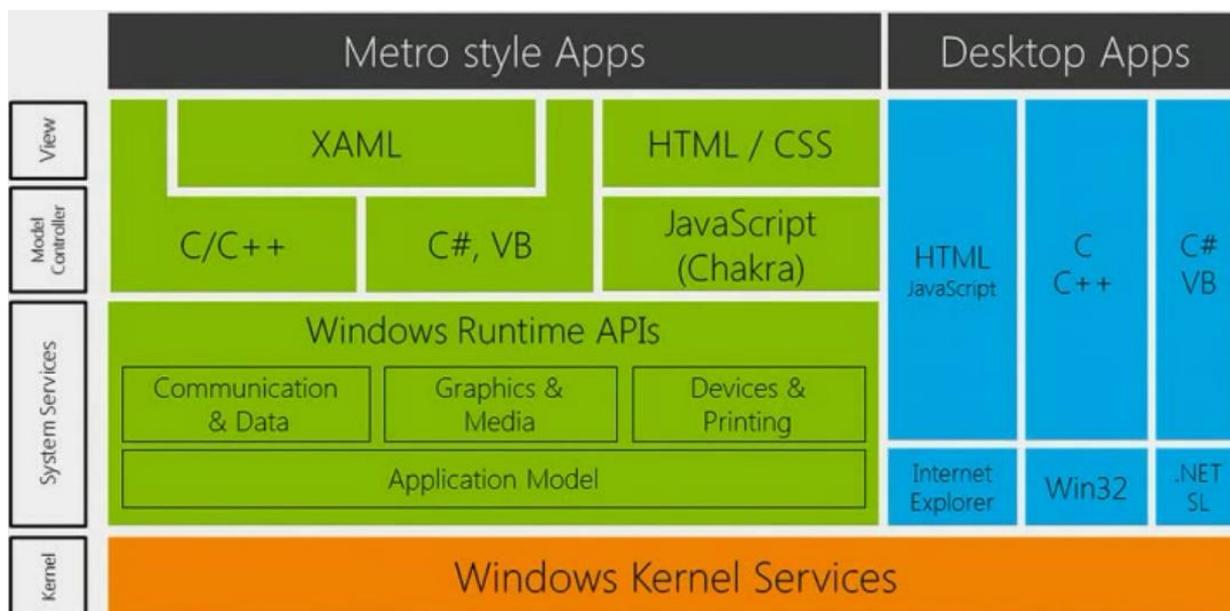
WinRT es la abreviación de Windows Runtime, un modelo de programación implementado y presentado por Microsoft en el año 2011 y que facilita el desarrollo de aplicaciones Software con estilo Metro para su sistema operativo Windows 8.

WinRT soporta tanto C++, como lenguajes de código manejado como JavaScript o bajo el prisma de .NET, con lenguajes de programación bien conocidos por todos, como son VB.NET y C#.La bondad de WinRT es que permite escribir aplicaciones tanto para el chip Intel como para el chip ARM que como todos sabemos, consume hoy día menos recursos y energía que el primero de ellos.Pero lo realmente interesante es que el espectro que cubre



XAML se ha ampliado, o lo que es lo mismo, que podemos utilizar XAML para crear aplicaciones en WPF, en Silverlight o con C++, y por otro lado está HTML y Javascript. Es decir, más opciones, más posibilidades.

WinRT es una API basada en COM (Component Object Model) que podría ser definida como la plataforma encargada de servir aplicaciones Metro. De hecho, WinRT ha sido diseñada para trabajar únicamente en sistemas operativos Windows 8 (y futuros sistemas operativos Windows). El único "pero" de WinRT con respecto al desarrollo .NET con el que estábamos trabajando desde hace más de 10 años es que WinRT es meramente un consumidor de Servicios Web, pero no actuará como proveedor de Servicios Web tal y como podíamos hacer con .NET.



Componentes Windows

## ***Ventajas de esta plataforma***

La principal ventaja es que la plataforma .NET es multilenguaje, por lo que nuestra aplicación funciona igual independientemente del lenguaje en el que esté escrita, ya que .NET se encarga de la traducción intermedia. De esta manera cada desarrollador puede elegir el lenguaje que conozca o que más se adapte a sus preferencias (gran ventaja respecto a otras plataformas).

La versión que se usa en nuestro caso es Windows SDK 8.1 con el lenguaje C# y XAML para la interfaz gráfica. De esta forma nos aseguramos que nuestra aplicación funciona en cualquier dispositivo que funcione con el sistema operativo Windows 8.1,



independientemente de las características específicas de cada modelo como memoria interna, RAM, velocidad de procesamiento... además de la marca del dispositivo.

En la actualidad existen diversos fabricantes que incorporan el sistema Windows (Phone) 8.1 en sus dispositivos, tales como Nokia, HTC, Samsung y Huawei. Podemos ver la lista de dispositivos en la Web: <http://www.windowsphone.com/es-es/phones>

## ***Otras alternativas***

Para el desarrollo de aplicaciones ejecutables en dispositivos móviles, existen otros lenguajes de programación o plataformas aparte de .NET para Windows, con diferentes características y propios del tipo de dispositivo, tales como: Symbian, J2ME, Android (Java), IOS (objective C)...

Actualmente los sistemas con más presencia en el mercado son Android, IOS y Windows Phone, por orden de nº de dispositivos, hemos elegido Windows por ser un sistema muy nuevo, con un aspecto moderno y que está empezando con muy buena aceptación. Tanto su número de dispositivos mundial como el de aplicaciones en la tienda Windows Store, está creciendo continuamente y poco a poco se abre camino entre sus competidores.

Pero dependiendo de cuál se escoja, hay que estudiar sus características y sus posibilidades, que en cada caso pueden llegar a ser bastante diferentes. En nuestro caso (Windows Phone) hemos de usar un lenguaje admitido por la plataforma .NET como C#, Visual Basic, J#, Delphi, Fortran... Siendo el más utilizado para el desarrollo de aplicaciones móviles C#, seguido de C++ y Visual Basic.

Hemos escogido C# porque es un lenguaje moderno y flexible, fácil de aprender y muy potente, además de disponer de un extenso abanico de documentación en MSDN y ser el más utilizado en el desarrollo de aplicaciones móviles y de escritorio para el sistema Windows, permitiendo desarrollar las “aplicaciones universales” Windows, que incluso funcionarán en la videoconsola Xbox One.



Móvil con Windows Phone



## Capítulo III – Entorno de desarrollo

### *Definición*

Un entorno de desarrollo integrado o, en inglés, Integrated Development Environment ('IDE'), es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

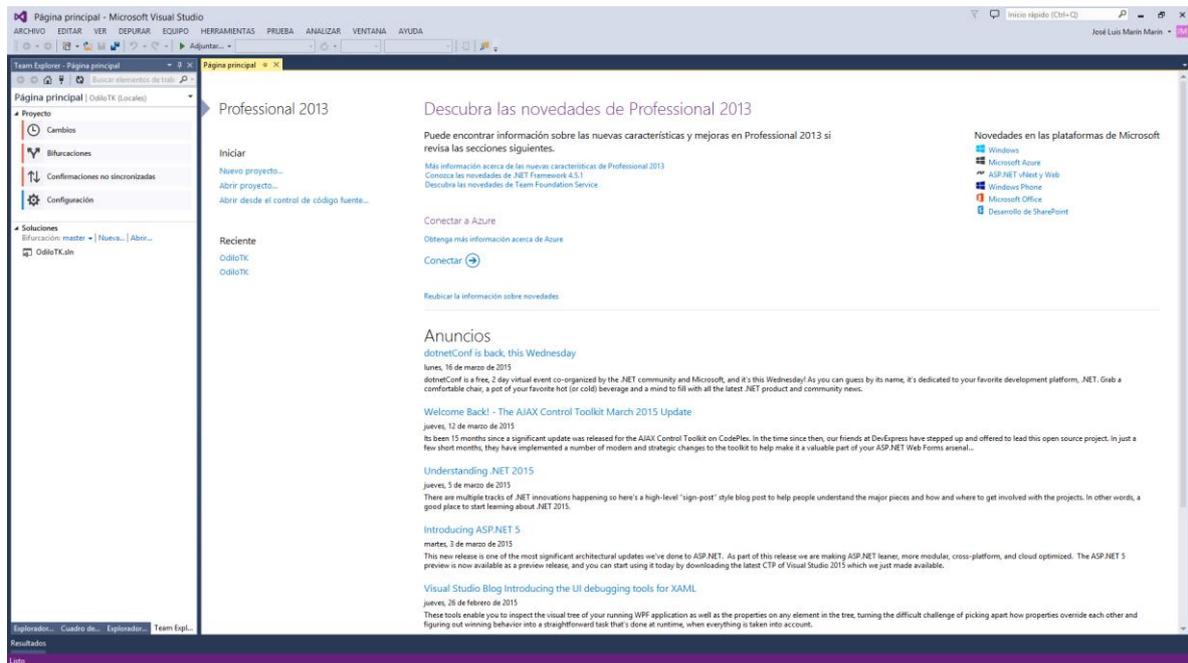
### *Visual Studio*

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET. aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

Visual Studio es un conjunto de herramientas de desarrollo y de otras tecnologías basado en componentes para crear aplicaciones eficaces de alto rendimiento. Además, Visual Studio está optimizado para diseño basado en equipos, desarrollo e implementación de soluciones empresariales.

Hemos utilizado la versión 2013 de Visual Studio debido a que se requiere esta versión para desarrollar aplicaciones universales para Windows 8.1 (con Visual Studio 2012 se pueden desarrollar para Windows Phone 8), en concreto la versión Professional, aunque se podría usar la versión Community (totalmente gratuita)



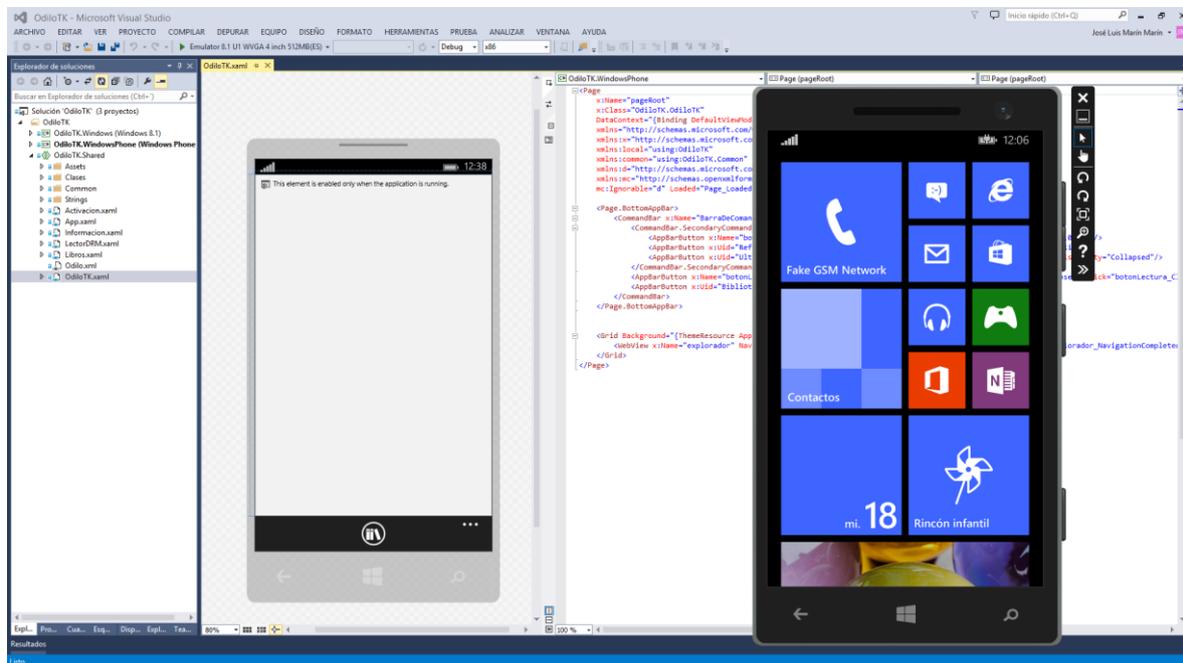
IDE Visual Studio

## ***Windows Phone SDK tools***

Las herramientas de Windows Phone consisten en Expression Blend para Windows Phone, Visual Studio 2012 para Windows Phone Express, XNA Game Studio y el emulador de Windows Phone que juntos ofrecen una experiencia de desarrollo sin fisuras. Experimentar, personalizar y estilo tu UX con sutiles animaciones y transiciones que agrega riqueza y vida a tus aplicaciones.

El SDK de Windows Phone incluye Expression Blend para el emulador de Windows Phone, Visual Studio Express para Windows Phone y Windows. Otros componentes clave incluidos en el paquete SDK son:

- SDK de Windows Phone
- Emulador de Windows Phone
- Silverlight 4 SDK y DRT
- Extensiones SDK de Windows Phone para XNA Game Studio
- Microsoft Expression Blend Preview SDK para Windows Phone
- Microsoft Advertising SDK para Windows Phone



Desarrollo con Windows Phone SDK

## Instalación Windows 8.1 SDK

Para la instalación de Windows 8.1 SDK en español, ha de dirigirse al siguiente enlace del centro de descargar de Microsoft:

<https://msdn.microsoft.com/es-es/windows/desktop/bg162891.aspx/>

Esto le descargará la última versión a fecha de hoy, que incluye todas las herramientas necesarias para el desarrollo de aplicaciones para Windows 8.1 y Windows Phone.

Esto incluye Visual Studio 2013 para Windows Phone, Microsoft Expression Blend, emulador de Windows Phone para Windows Phone 8.1, 8 y emulador para Windows Phone 7.5.

Pero se **recomienda** directamente **descargar la nueva versión de Visual Studio 2013 Community**, que es **gratuita** e incluye todas las herramientas para el desarrollo de aplicaciones móviles, escritorio, servicios web, etc.

Para la instalación e Visual Studio 2013 Community en español, ha de dirigirse al siguiente enlace:



[https://www.visualstudio.com/es-es/downloads/download-visual-studio-vs#DownloadFamilies\\_2](https://www.visualstudio.com/es-es/downloads/download-visual-studio-vs#DownloadFamilies_2)

Si desea realizar otra actualización aparte, en el mismo enlace aparecen otras descargas relacionadas debajo. Por ejemplo puede descargar “[Actualización del SDK de Windows Phone para Windows Phone 7.8](#)” para que se instale el emulador de Windows Phone 7.8 a su instalación de Visual Studio, si va a implementar aplicaciones con esta versión.

## Requisitos del sistema

- **Sistema operativo compatible**
  - Windows 8.1; Windows 8.1 Pro
- **Tipo de sistema operativo:**
  - Versiones de cliente de Windows 8.1 de 64 bits (x64)
- **Hardware:**
  - 6,5 GB de espacio disponible en el disco duro
  - 4 GB de RAM
  - CPU de 64 bits (x64)
- **Emulador de Windows Phone 8.1:**
  - Windows 8.1 Pro Edition o versiones posteriores
  - Requiere un procesador que admita la traducción de direcciones de segundo nivel (SLAT)

Si el equipo se ajusta a los requisitos del sistema operativo y del hardware pero no a los del emulador de Windows Phone 8, se instalará y ejecutará Windows Phone SDK 8. Sin embargo, el emulador de Windows Phone 8 no funcionará y no se podrá implementar ni probar aplicaciones en él.

## ***Desarrollo de aplicaciones en Visual Studio IDE:***

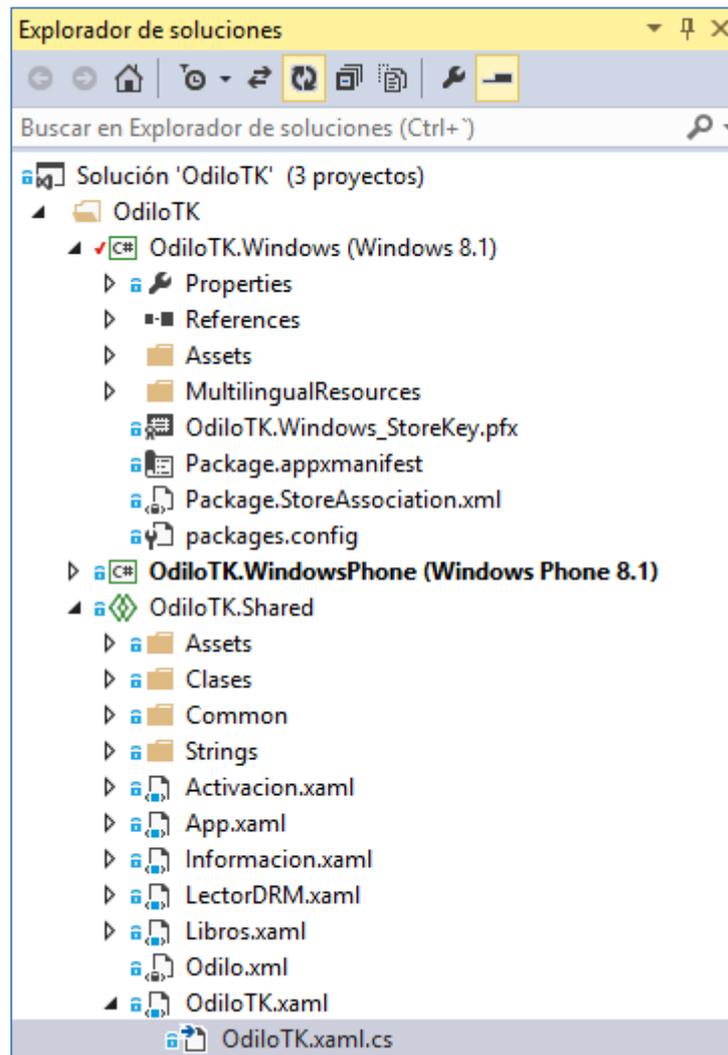
Como se puede observar, en esta herramienta es posible realizar todas las fases de desarrollo de aplicaciones para Windows/Windows Phone.

- Esta herramienta dispone de un cuadro llamado “Explorador de soluciones”, donde podemos tener uno o varios proyectos abiertos, siempre asociados a nuestra solución. Desde aquí podemos abrir todos los archivos de nuestro proyecto y empezar a editarlos, así como consultar las propiedades del proyecto.

Los diferentes archivos que nos encontramos son:



- Archivos XAML: Definen las diferentes páginas de nuestra aplicación mediante su interfaz gráfica. La página App.xaml siempre existe en la aplicación y no tiene interfaz gráfica, esto es debido a que es la que se encarga del funcionamiento y del flujo de la aplicación.
- Archivos .CS: Definen el código C# de nuestra aplicación. Desplegando cada una de las páginas XAML podemos ver si archivo .CS asociado, en donde se incluyen las diferentes propiedades y métodos de esa página, por ejemplo, los manejadores de eventos. También podemos tener clases independientes con funciones que usa nuestra aplicación, por ejemplo, en la carpeta clases tenemos algunas con métodos estáticos que se pueden llamar desde cualquier página.
- Referencias: Aquí tenemos las diferentes referencias incluidas en nuestro proyecto. Ejemplo: Si incluimos una DLL externa, debemos referenciarla, así como si tenemos un proyecto que necesita llamar a código de otro proyecto dentro de nuestra solución.
- Diferentes elementos que usa nuestra aplicación, tales como iconos, imágenes, sonidos, etc...: estos se encuentran en la carpeta "Assets" o en la raíz del proyecto. Además los archivos de recursos que usa la aplicación se encuentran en la carpeta strings.



Explorador de soluciones

- Además dispone de un cuadro en el que se visualizan los archivos de todo tipo del proyecto, mencionados anteriormente y que sirve de editor de texto. A continuación vemos un archivo .CS abierto. Como podemos observar se pueden abrir varios archivos separados por pestañas y dispone de unas listas desplegables arriba, que nos sirven para ir directamente a una clase y método dentro del archivo:



```
OdiloTK.xaml.cs
OdiloTK.WindowsPhone
OdiloTK.OdiloTK
direcciones

using OdiloTK.Common;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Windows.Storage;
using Windows.UI.Popups;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Navigation;
using Windows.Web.Http;
using Windows.Web.Http.Headers;

// La plantilla de elemento Página básica está documentada en http://go.microsoft.com/fwlink/?linkid=234237

namespace OdiloTK
{
    /// <summary>
    /// Página básica que proporciona características comunes a la mayoría de las aplicaciones.
    /// </summary>
    public sealed partial class OdiloTK : Page
    {
        public List<string> direcciones;
        private string ultimaFicha, ultimolibro, pagina, dir_biblioteca;
        private NavigationHelper navigationHelper;
        private ObservableDictionary defaultViewModel = new ObservableDictionary();
        private bool detectar, leyendo = false;

        /// <summary>
        /// Este puede cambiarse a un modelo de vista fuertemente tipada.
        /// </summary>
        public ObservableDictionary DefaultViewModel
        {
            get { return this.defaultViewModel; }
        }

        /// <summary>
        /// NavigationHelper se usa en cada una de las páginas para ayudar con la navegación y
        /// la administración de la duración de los procesos
        /// </summary>
        public NavigationHelper NavigationHelper
        {
            get { return this.navigationHelper; }
        }

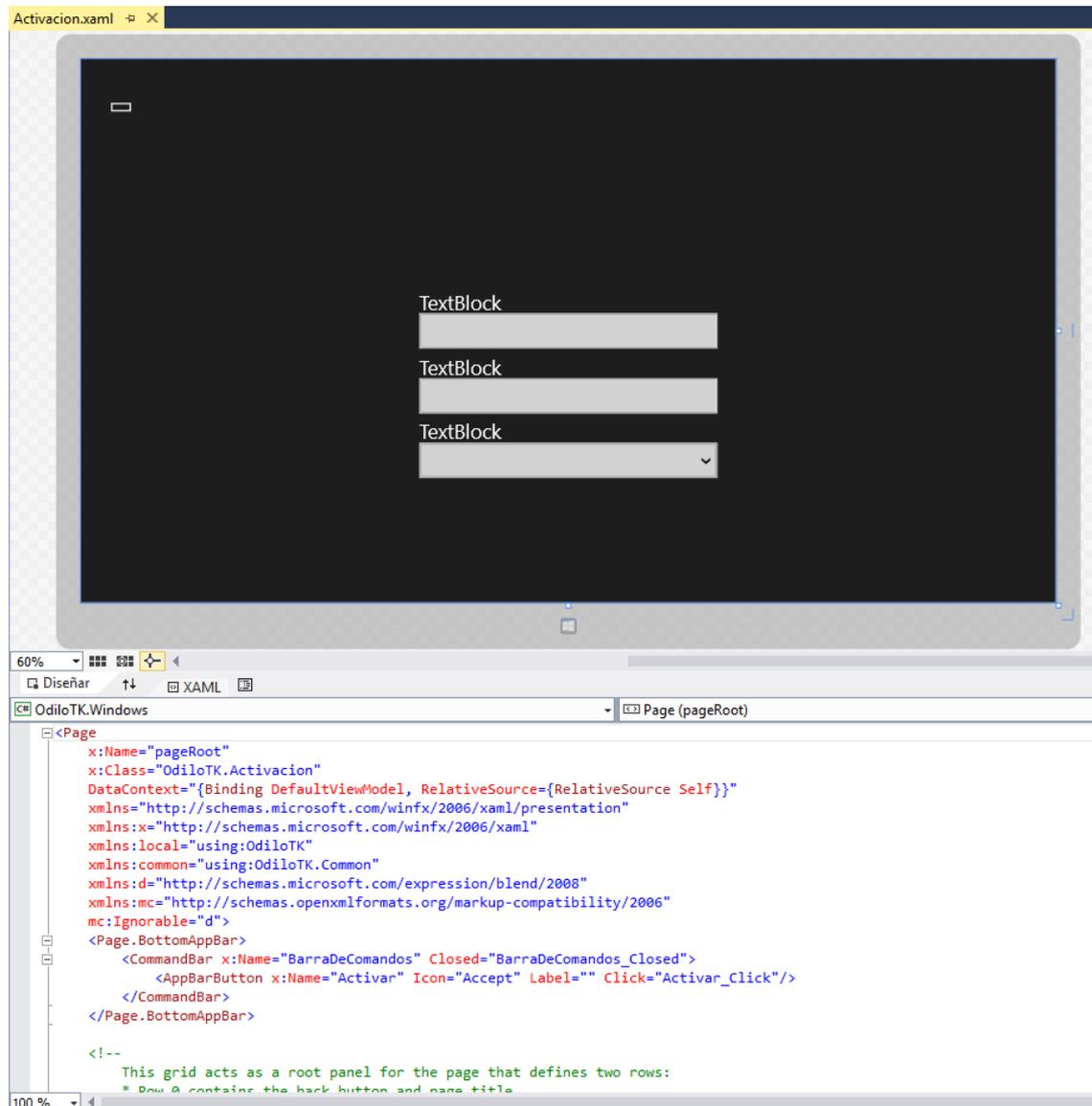
        public OdiloTK()
        {
            this.InitializeComponent();
            this.navigationHelper = new NavigationHelper(this);
            this.navigationHelper.LoadState += navigationHelper_LoadState;
            this.navigationHelper.SaveState += navigationHelper_SaveState;
        }

        #if WINDOWS_PHONE_APP
        botonAtras.Visibility = Visibility.Collapsed;
        #elif WINDOWS_APP
    }
}

Editor de texto
```

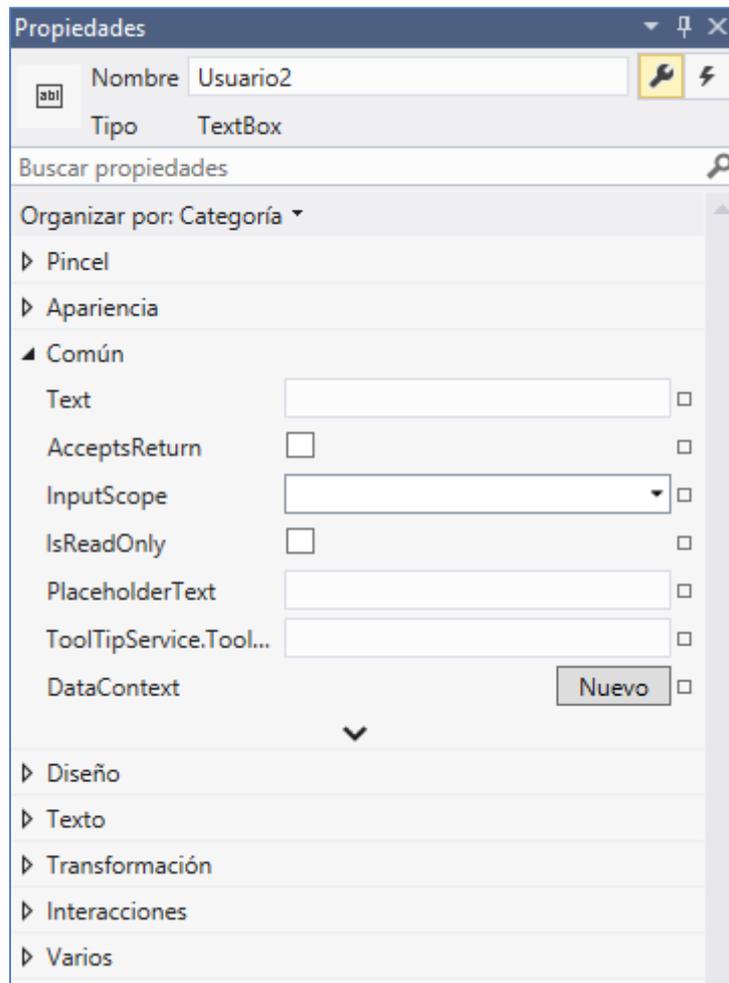
Si abrimos un archivo .XAML nos aparece una ventana dividida en dos partes, el editor de código y un panel con la interfaz gráfica, donde podemos diseñar el aspecto visual de nuestra página. Para ello dispone de tres ventanas desplegadas en el panel gráfico. Estas son:

- Cuadro de herramientas: Ventana con los controles que disponemos para usar en Windows Phone (se pueden crear nuevos o descargar librerías con nuevos controles/modificados). Para colocar un control, tan solo seleccionar y arrastrar al panel del dispositivo.
- Dispositivo: Podemos modificar el aspecto del dispositivo a mostrar en el panel de edición gráfica, ejemplo: la orientación, colores, resolución...
- Esquema del documento: Podemos ver gráficamente cada uno de los elementos descritos mediante XAML de los que se compone nuestra página, ir directamente a uno de ellos y editarlo o añadir nuevos, así como sus plantillas.



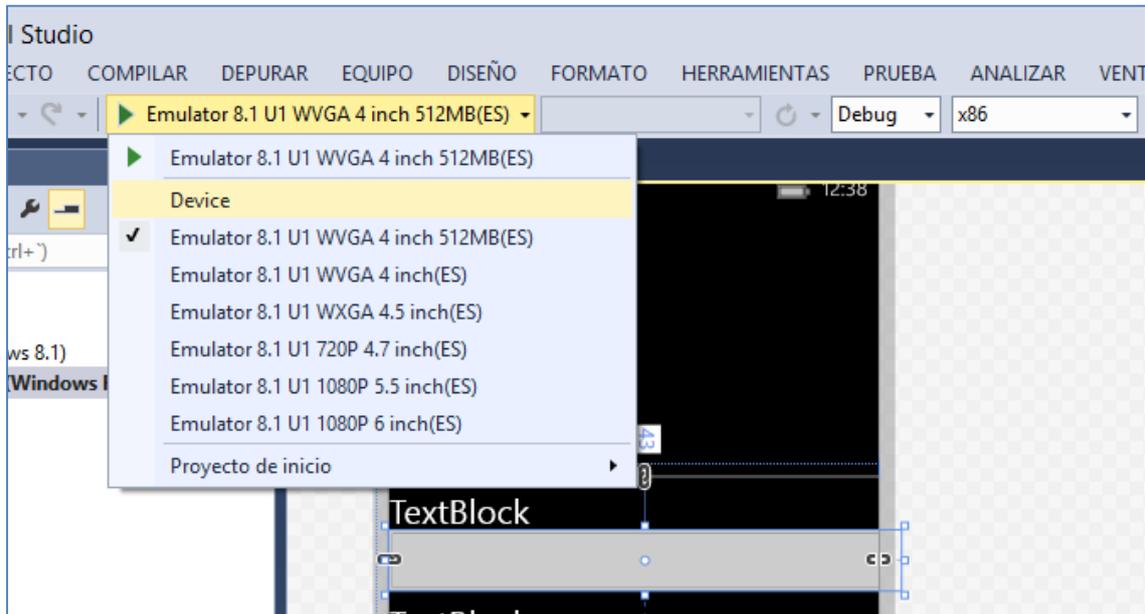
Panel gráfico y editor XAML

- Otro cuadro muy importante durante el desarrollo es el de “propiedades”, el cual nos muestra las propiedades del elemento que tengamos seleccionado en ese momento, pudiendo ordenarlas por categorías y modificar aquí directamente lo que necesitemos, ya que Visual Studio modifica el código XAML automáticamente a partir de las propiedades que modifiquemos. Cabe destacar que en esta ventana está seleccionado el icono de “herramienta”, pero podemos seleccionar el icono de “rayo” para que nos muestre la lista de eventos disponibles para el elemento seleccionado, volviendo a la lista de propiedades al pulsar sobre la “herramienta.”

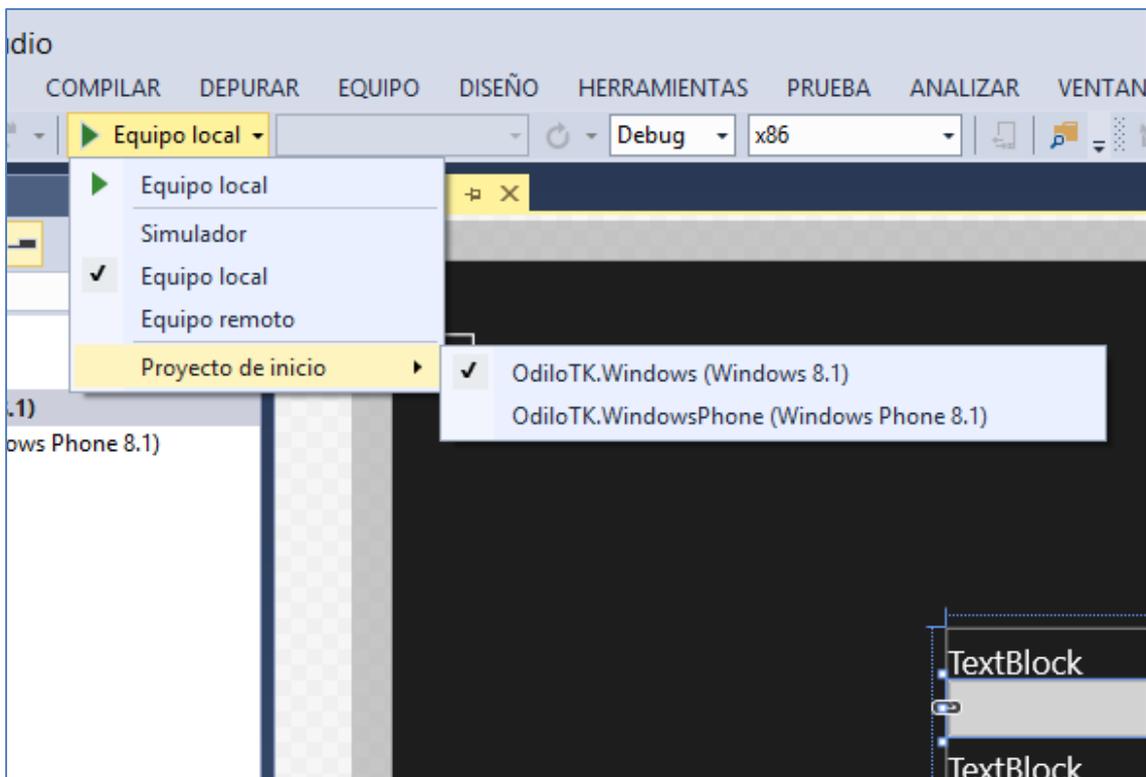


Cuadro de propiedades

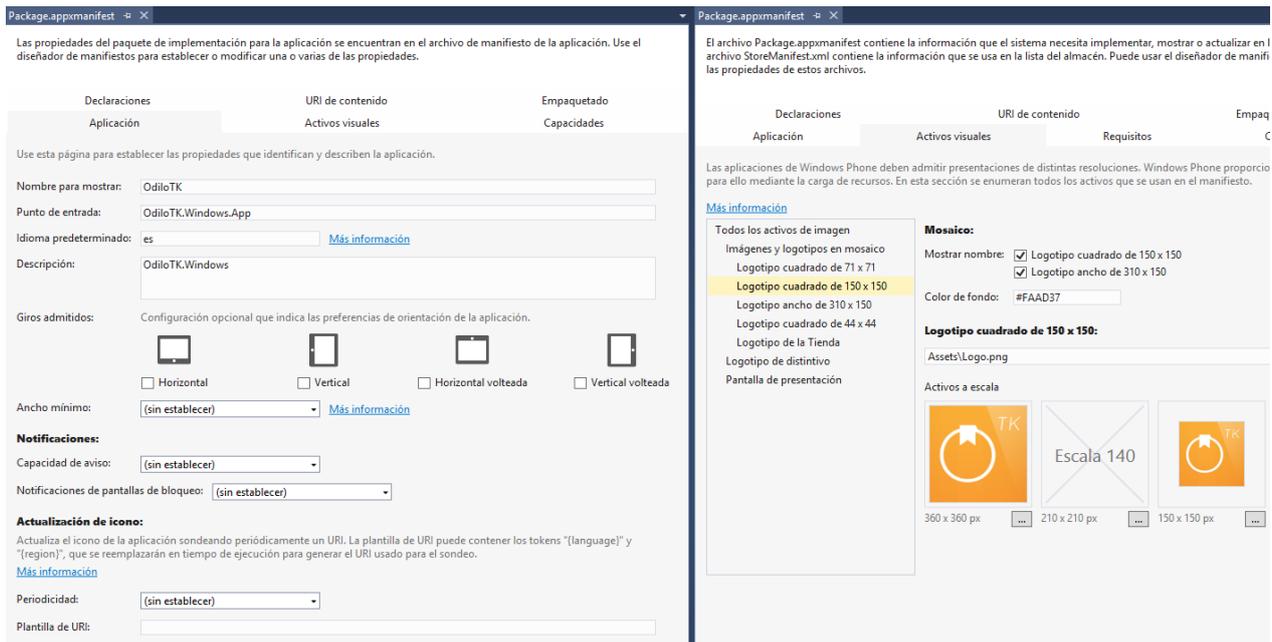
- Respecto a compilar la aplicación, se puede realizar mediante el menú “Compilar” y “compilar solución” (se compila toda la solución completa, hay que elegir cual es el proyecto de inicio si hay varios). Pero si procede a ejecutar la aplicación, es decir, desplegarla sobre el emulador o sobre el dispositivo físico, se realiza automáticamente la compilación y el guardado de las modificaciones, para ello tenemos el menú “Depurar” con las opciones “Iniciar depuración” e “Iniciar sin depurar”, depende de cómo queramos ejecutar el despliegue. Por último debemos elegir si el despliegue es en modo “debug” o “release” (release se usa para el dispositivo físico y siempre que sea para subir a la tienda) y dónde vamos a desplegar la aplicación (podemos tener varios emuladores, dependiendo de nuestra configuración/instalación de Visual Studio).



Selección de emulador o dispositivo para ejecutar

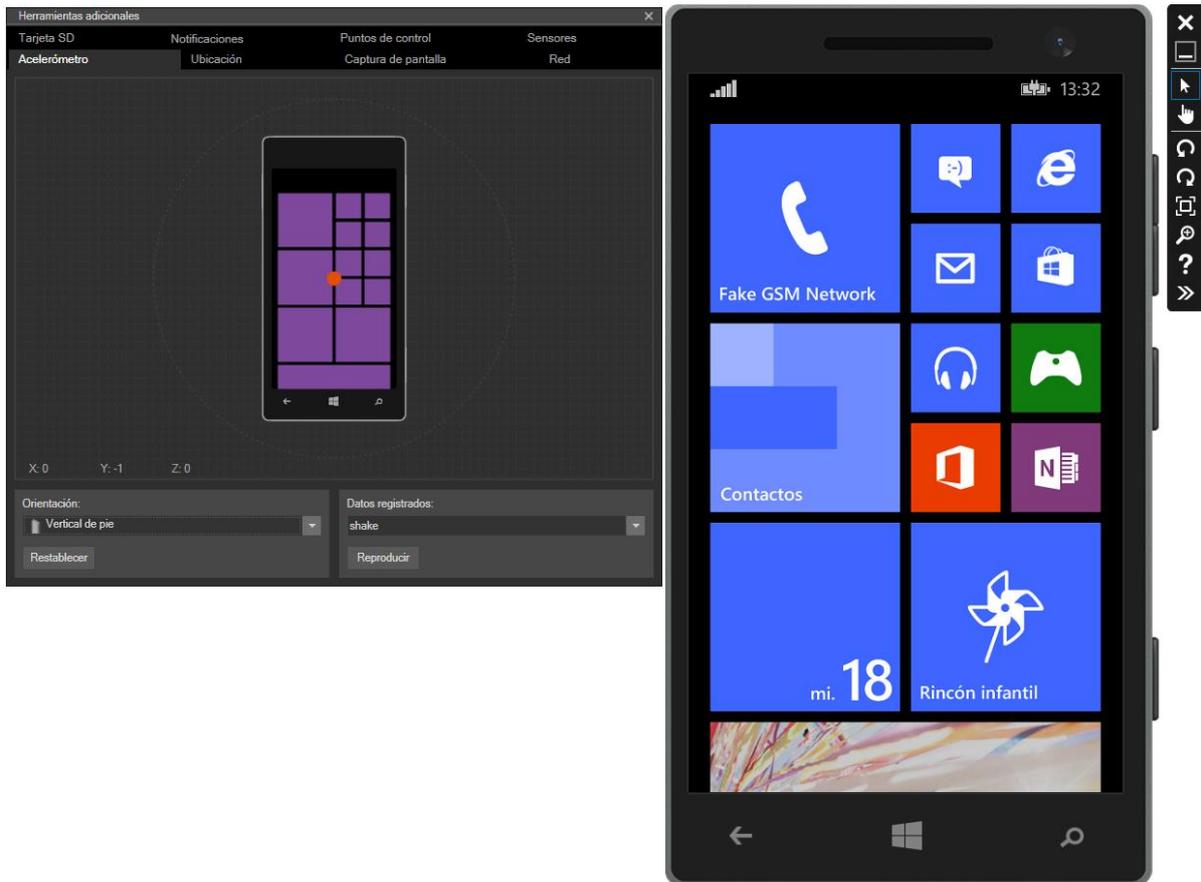


Despliegue en simulador o equipo local, proyecto Windows



Manifiestos de la aplicación (Windows- Windows Phone)

- Emulador de Windows Phone 8.1: es una máquina virtual ejecutando el sistema operativo Windows Phone como si de un dispositivo físico se tratase, muy potente y dispone de prácticamente todas las funciones que tienen los dispositivos: acceso a internet, localización, giroscopio, acceso a la tienda, simulación de llamadas, etc... Dispone de una barra de herramientas que nos permite girar el dispositivo, modificar su tamaño o desplegar un panel de “herramientas adicionales” mediante la flecha, que se muestra continuación:



Emulador Windows Phone con panel de herramientas adicionales



## Capítulo IV – Medios empleados durante la implementación y pruebas

Para la implementación y pruebas se ha empleado el entorno de desarrollo Visual Studio Edición Profesional para las pruebas de las diversas funciones durante su desarrollo, usando el emulador que incorpora de Windows Phone y el simulador/equipo local para Windows.

Para que la aplicación soporte la orientación horizontal (landscape) en todas sus páginas, se ha utilizado también el editor de Visual Studio (XAML), que nos permite realizar diseños compactos y autoadaptables a diferentes resoluciones y orientación.

El completo funcionamiento de la aplicación se ha ido probando tanto en el emulador como en móviles Nokia Lumia con sistema operativo Windows Phone 8.1. Gracias al completo emulador que nos proporciona el IDE, podemos desplegar la aplicación en un dispositivo físico como: Ordenador, teléfono tableta, además de pruebas en simulador (Windows), emulador (Windows Phone).

La aplicación necesita el uso de la conexión a Internet, se ha comprobado que funcione correctamente tanto con conexión de datos móviles como vía Wi-Fi. Por otro lado, si no se dispone de conexión alguna, el programa está adecuado para que permita funcionar sin línea y detectar cuando ésta esté disponible de nuevo.



Emulador Windows Phone



Móvil Nokia Lumia



## Capítulo V – Introducción a las BBDD relacionales

Una Base de Datos Relacional, es una base de datos que cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas. Permiten establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas, de ahí proviene su nombre: "Modelo Relacional". Tras ser postuladas sus bases en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos.

### *Características*

- Una Base de Datos Relacional se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y ajenas (o foráneas).
- Las claves primarias son la clave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos.
- Las claves ajenas se colocan en la tabla hija, contienen el mismo valor que la clave primaria del registro padre; por medio de éstas se hacen las relaciones.

### *Elementos*

#### *- Relaciones Base y Derivadas*

En una base de datos relacional, todos los datos se almacenan y se accede a ellos por medio de relaciones. Las relaciones que almacenan datos son llamadas "relaciones base" y su implementación es llamada "tabla". Otras relaciones no almacenan datos, pero son calculadas al aplicar operaciones relacionales. Estas relaciones son llamadas "relaciones derivadas" y su implementación es llamada "vista" o "consulta". Las relaciones derivadas son convenientes ya que expresan información de varias relaciones actuando como si fuera una sola.

#### *- Restricciones*

Una restricción es una limitación que obliga el cumplimiento de ciertas condiciones en la base de datos. Algunas no son determinadas por los usuarios, sino que son inherentemente definidas por el simple hecho de que la base de datos sea relacional. Algunas otras restricciones las puede definir el usuario, por ejemplo, usar un campo con valores enteros entre 1 y 10.

Las restricciones proveen un método de implementar reglas en la base de datos. Las restricciones limitan los datos que pueden ser almacenados en las tablas. Usualmente se



definen usando expresiones que dan como resultado un valor booleano, indicando si los datos satisfacen la restricción o no.

Las restricciones no son parte formal del modelo relacional, pero son incluidas porque juegan el rol de organizar mejor los datos. Las restricciones son muy discutidas junto con los conceptos relacionales.

- ***Dominios***

Un dominio describe un conjunto de posibles valores para cierto atributo. Como un dominio restringe los valores del atributo, puede ser considerado como una restricción. Matemáticamente, atribuir un dominio a un atributo significa "todos los valores de este atributo deben ser elementos del conjunto especificado".

Distintos tipos de dominios son: enteros, cadenas de texto, fecha, no procedurales etc.

Cada tabla puede tener uno o más campos cuyos valores identifican de forma única cada registro de dicha tabla, es decir, no pueden existir dos o más registros diferentes cuyos valores en dichos campos sean idénticos. Este conjunto de campos se llama clave única. Pueden existir varias claves únicas en una determinada tabla, y a cada una de éstas suele llamársele candidata a clave primaria.

- ***Clave primaria***

Una clave primaria es una clave única elegida entre todas las candidatas que define unívocamente a todos los demás atributos de la tabla, para especificar los datos que serán relacionados con las demás tablas. La forma de hacer esto es por medio de claves foráneas.

- ***Clave foránea***

Una clave foránea es una referencia a una clave en otra tabla, determina la relación existente en dos tablas. Las claves foráneas no necesitan ser claves únicas en la tabla donde están y sí a donde están referenciadas.

Por ejemplo, el código de departamento puede ser una clave foránea en la tabla de empleados. Se permite que haya varios empleados en un mismo departamento, pero habrá uno y sólo un departamento por cada clave distinta de departamento en la tabla de empleados.

- ***Clave índice***

Las claves índice surgen con la necesidad de tener un acceso más rápido a los datos. Los índices pueden ser creados con cualquier combinación de campos de una tabla. Las consultas que filtran registros por medio de estos campos, pueden encontrar los registros de forma no secuencial usando la clave índice.

Las bases de datos relacionales incluyen múltiples técnicas de ordenamiento, cada una de ellas es óptima para cierta distribución de datos y tamaño de la relación.

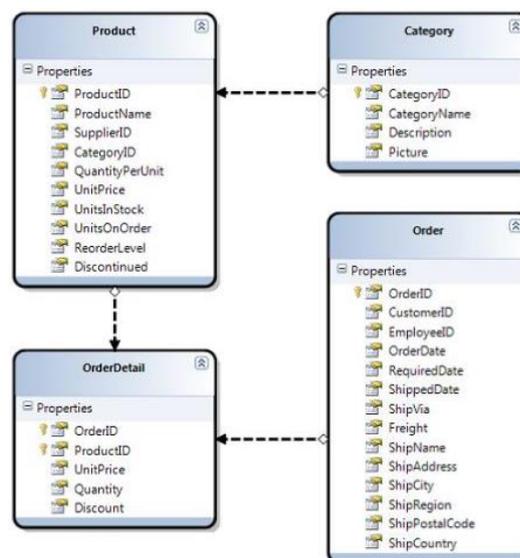


Los índices generalmente no se consideran parte de la base de datos, pues son un detalle agregado. Sin embargo, las claves índices son desarrolladas por el mismo grupo de programadores que las otras partes de la base de datos.

- **Procedimientos almacenados**

Un procedimiento almacenado es código ejecutable que se asocia y se almacena con la base de datos. Los procedimientos almacenados usualmente recogen y personalizan operaciones comunes, como insertar un registro dentro de una tabla, recopilar información estadística, o encapsular cálculos complejos. Son frecuentemente usados por un API por seguridad o simplicidad.

Los procedimientos almacenados no son parte del modelo relacional, pero todas las implementaciones comerciales los incluyen.



BDD relacional SQL Server

## ***Estructura***

La base de datos se organiza en dos marcadas secciones; el esquema y los datos (o instancia).

El esquema es la definición de la estructura de la base de datos y principalmente almacena los siguientes datos:

- El nombre de cada tabla
- El nombre de cada columna



- El tipo de dato de cada columna
- La tabla a la que pertenece cada columna

Las bases de datos relacionales pasan por un proceso al que se le conoce como normalización, el resultado de dicho proceso es un esquema que permite que la base de datos sea usada de manera óptima.

Los datos o instancia es el contenido de la base de datos en un momento dado. Es en sí, el contenido de todos los registros.

## ***Manipulación de la información***

Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el álgebra relacional y el cálculo relacional. El álgebra relacional permite describir la forma de realizar una consulta, en cambio, el cálculo relacional sólo indica lo que se desea devolver.

El lenguaje más común para construir las consultas a bases de datos relacionales es SQL (Structured Query Language), un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales integradas.

En el modelo relacional los atributos deben estar explícitamente relacionados a un nombre en todas las operaciones, en cambio, el estándar SQL permite usar columnas sin nombre en conjuntos de resultados, como el asterisco taquigráfico (\*) como notación de consultas.

Al contrario del modelo relacional, el estándar SQL requiere que las columnas tengan un orden definido, lo cual es fácil de implementar en una computadora, ya que la memoria es lineal.

Es de notar, sin embargo, que en SQL el orden de las columnas y los registros devueltos en cierto conjunto de resultado nunca está garantizado, a no ser que explícitamente sea especificado por el usuario.

## **Ventajas**

- Provee herramientas que garantizan evitar la duplicidad de registros.
- Garantiza la integridad referencial, así, al eliminar un registro elimina todos los registros relacionados dependientes.
- Favorece la normalización por ser más comprensible y aplicable.



## **Desventajas**

- Presentan deficiencias con datos gráficos, multimedia, CAD y sistemas de información geográfica.
- No se manipulan de forma manejable los bloques de texto como tipo de dato.
- Las bases de datos orientadas a objetos (BDOO) se propusieron con el objetivo de satisfacer las necesidades de las aplicaciones anteriores y así, complementar pero no sustituir a las bases de datos relacionales.
- 

## ***Diseño de las bases de datos relacionales***

El primer paso para crear una base de datos, es planificar el tipo de información que se quiere almacenar en la misma, teniendo en cuenta dos aspectos: la información disponible y la información que necesitamos.

La planificación de la estructura de la base de datos, en particular de las tablas, es vital para la gestión efectiva de la misma. El diseño de la estructura de una tabla consiste en una descripción de cada uno de los campos que componen el registro y los valores o datos que contendrá cada uno de esos campos.

Los campos son los distintos tipos de datos que componen la tabla, por ejemplo: nombre, apellido, domicilio. La definición de un campo requiere: el nombre del campo, el tipo de campo, el ancho del campo, etc.

Los registros constituyen la información que va contenida en los campos de la tabla, por ejemplo: el nombre del paciente, el apellido del paciente y la dirección de este. Generalmente los diferentes tipos de campos que se pueden almacenar son los siguientes: Texto (caracteres), Numérico (números), Fecha / Hora, Lógico (informaciones lógicas si/no, verdadero/falso, etc.), imágenes.

En resumen, el principal aspecto a tener en cuenta durante el diseño de una tabla es determinar claramente los campos necesarios, definirlos en forma adecuada con un nombre especificando su tipo y su longitud.



## Capítulo VI – Introducción al motor de BBDD SQLite

SQLite es un proyecto de dominio público, su creador es D. Richard Hipp, el cual implementa una pequeña librería de aproximadamente 500kb, programado en el lenguaje C, de dominio público, totalmente libre y que tiene con función hacer de un sistemas de bases de datos relacional.

Uno de las primeras diferencia entre los motores de Bases de datos convencionales es su arquitectura cliente/servidor, pues SQLite es independiente, simplemente se realizan llamadas a sub rutinas o funciones de las propias librerías de SQLite, lo cual reduce ampliamente la latencia en cuanto al acceso a las bases de datos. Con lo cual podemos decir que las base de datos compuesta por la definición de las tablas, índices y los propios datos son guardados por un solo fichero estándar y en un solo ordenador.

Conozcamos la historia detrás del proyecto SQLite:

Cuando D. Richard Hipp trabajaba desarrollando software para la fuerza naval de los Estados Unidos, comenzó a desarrollar SQLite, según él cuenta con sus propias palabras: El proyecto SQLite surgió de una necesidad personal, para mi propio uso.

En enero de 2000 D. Richard Hipp estaba trabajando con su equipo de la General Dynamics en la Fuerza naval de los Estados Unidos, en un proyecto de software, el cual se conectaba a una base de datos Informix, el motor funcionaba muy bien, pero habían tenido problemas para hacer una reconfiguración cuando el sistema se reiniciaba. Luego cambiaron a PostgreSQL, pero administrar la base de datos era un poco más complejo. Fue en ese momento cuando surgió la idea de escribir un simple motor de base de datos SQL que permitiera leer los archivos del disco duro, y luego ser llamados en diferentes solicitudes.

Cinco meses más tarde comenzó a escribir las primeras versiones de lo que hoy conocemos como SQLite, con el pensamiento de que sería útil en algún problema similar.

Es claro que SQLite tiene la capacidad de reemplazar a grandes motores de Bases de Datos y acoplarse al desarrollo de nuestros proyectos informáticos, ya sea en ambientes de prototipos de sistemas como así también en complejos y robustos software.





## ***Características de SQLite***

Veamos algunas razones para escoger SQLite como una herramienta de desarrollo:

- Tamaño: SQLite tiene una pequeña memoria y una única biblioteca es necesaria para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
- Rendimiento de base de datos: SQLite realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.
- Portabilidad: se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.
- Estabilidad: SQLite es compatible con ACID, reunión de los cuatro criterios de Atomicidad, Consistencia, Aislamiento y Durabilidad.
- SQL: implementa un gran subconjunto de la ANSI – 92 SQL estándar, incluyendo subconsultas, generación de usuarios, vistas y triggers.
- Interfaces: cuenta con diferentes interfaces del API, las cuales permiten trabajar con C++, PHP, Perl, Python, Ruby, Tcl, groovy, etc.
- Costo: SQLite es de dominio público, y por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente.

Alguna de las características más relevantes de SQLite son las siguientes:

### ***No posee configuración***

De la forma en que fue creado y diseñado SQLite, NO necesita ser instalado. NO prender, reiniciar o apagar un servidor, e incluso configurarlo. Esta cualidad permite que no haya un administrador de base de datos para crear las tablas, vistas, asignar permisos. O bien la adopción de medidas de recuperación de servidor por cada caída del sistema.

### ***Portabilidad***

SQLite puede ser ejecutado en diferentes sistemas operativos, como ser Windows, Linux, BSD, Mac OS X, Solaris, HPUNIX, AIX o estar embebido en muchos otros como QNX, VxWorks, Symbian, Palm OS, Windows CE. Se puede notar que muchos de ellos trabajan a 16, 32 y 64 Bits. La portabilidad no está dada en sí por el software, sino por la base de datos condensada en un solo fichero, que puede estar situado en cualquier directorio, trayendo como ventaja que la base de datos puede ser fácilmente copiada a algún dispositivo USB o ser enviada vía correo electrónico.

### ***Registros de longitud variable***

Generalmente los motores asignan una cantidad fija de espacio en disco para cada fila en la mayoría de los campos de una determinada tabla. Por ejemplo, tomemos un campo de tipo VARCHAR(255), esto significa que el motor le asignará 255 bytes de espacio fijo en disco, independientemente de la cantidad de información que se almacene en ese campo. En cambio, SQLite aplica su tecnología y realizará todo lo contrario, utilizando para ello la cantidad de espacio en disco necesario para almacenar la información real del campo.



Tomando el ejemplo anterior, si quisiera almacenar un solo carácter en un campo definido como VARCHAR(255), entonces un único byte de espacio de disco se consume.

El uso de registros de longitud variable por SQLite, tiene una serie de ventajas, entre ellas el resultado de un pequeño archivo de base de datos y optimización de la velocidad de la misma, puesto que hay menos información desperdiciada que leer y recorrer.

Así como encontramos algunas ventajas y características realmente asombrosas, también cuenta con algunas limitaciones:

- Limitaciones en Where: esta limitación está dada por el soporte para clausuras anidadas.
- Falta de Clave Foránea: se hace caso omiso de las claves foráneas; esto quiere decir, cuando se realice la creación de la tabla desde el modo consola, está permitiendo el uso de la clausura, aunque no realizara el chequeo de la misma.
- Poca documentación en español: si bien ya contamos con comunidades hispanas de SQLite, sería importante encontrar más documentación, libros, reviews, etc. como muchos otros motores de bases de datos cuentan hoy en día.

```
danyx@hp:~$ sqlite3 prueba.db
SQLite version 3.4.2
Enter ".help" for instructions
sqlite> CREATE TABLE personal (
  ..> id smallint primary key,
  ..> nombre varchar(25),
  ..> apellido varchar(25),
  ..> direccion varchar(25)
  ..> );
sqlite> █
```

Creación de tabla desde consola SQLite

## ***Introducción al uso de SQLite***

SQLite, al igual que muchos de los motores de bases de datos, posee su administración predefinida a través del uso de la consola o terminal, aquellas personas que pudieron administrar antes otros motores como por ejemplo es el caso de MySQL, sabrán que no es el único administrador que se conoce sino más bien existen una amplia gama para la elección, por ejemplo phpMyAdmin.

Con SQLite sucede lo mismo: es posible administrarlo desde varios sistemas, como es el caso de phpSQLiteAdmin, VisualSQLite, realSqlserver.



## Cómo instalar SQLite

Su instalación se puede realizar por medio de los repositorios o a través de su fuente, como observaremos a continuación.

### - A través de los repositorios

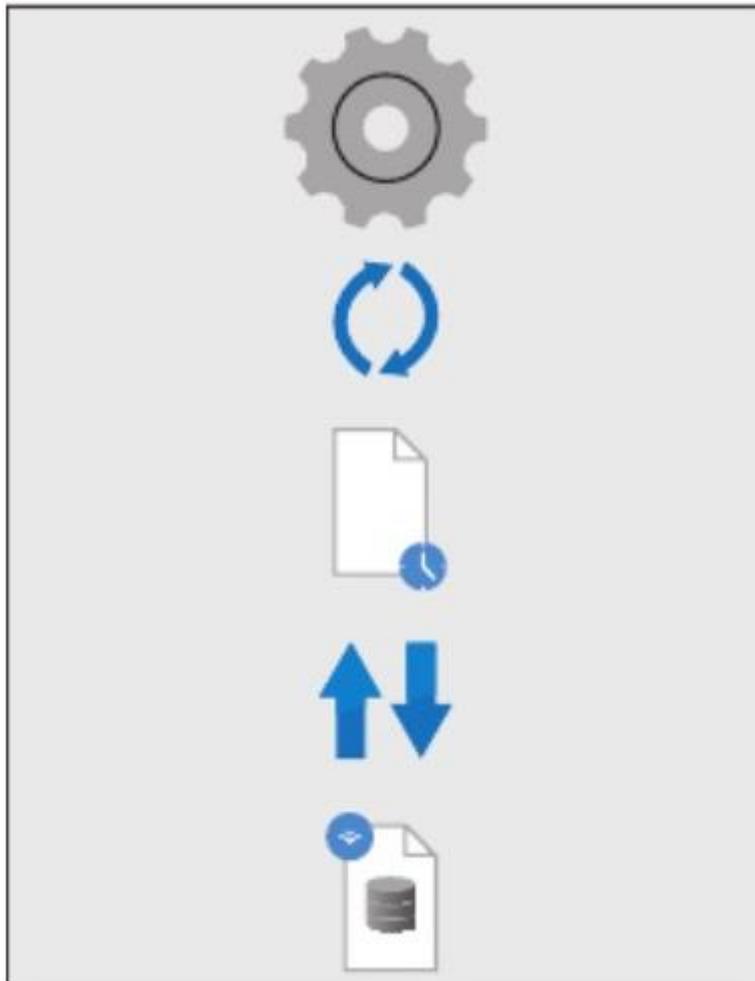
Como ya es costumbre, si tenemos nuestra distribución predilecta y contamos con acceso a Internet o tenemos los DVD de repositorios y contamos con los permisos de súper usuario root, escribimos lo siguiente:

```
aptitude install sqlite3
```

Si contamos con los códigos fuente descargados desde el sitio oficial de SQLite <http://www.sqlite.org/download.html>, simplemente deberemos realizar los siguientes pasos:

- Descomprimos los ficheros fuentes de SQLite en alguna carpeta con `tar xzfsqlite.tar.gz`
- Ejecutamos el script. `./configure`
- Ejecutamos `make install` para realizar la instalación en el sistema.

Si deseamos instalar SQLite en el sistema operativo **Windows**, la instalación se realizará de la siguiente forma: descargamos los ejecutables de SQLite, usamos la consola de línea de comandos de Windows `cmd`, nos posicionamos en la carpeta y ejecutamos el archivo `sqlite3.exe`, el cual disparará la consola de `sqlite3`.



Esquema de SQLite



## Capítulo VII – Características de la aplicación

### *Introducción*

En este capítulo vamos a explicar en detalle el funcionamiento de la aplicación “OdiloTK”. Lo primero de todo es destacar que la aplicación es única, incluye toda la lógica del proceso, exceptuando el contenido de la aplicación web que se ejecuta sobre el “Web view”, pero realiza tratamientos/modificaciones sobre está por ejemplo inyectando código JavaScript. Además se realiza llamadas a algunos servicios web REST propiedad de Odilo.

Todo el código de la aplicación está implementado en C# y la interfaz gráfica definida con XAML. Esta interfaz se compone de una serie de páginas que vamos a detallar a continuación, definiendo el flujo y el contenido de cada una.

*Nota:* Cada vez que se vaya a realizar una acción que involucre acceso a internet en la aplicación, se comprueba si se dispone de acceso mediante una función definida en la clase “Funciones.cs”, de forma que si no hay conexión se avisa al usuario mediante un cuadro de mensaje y no se permite la acción.

Claves a tener en cuenta:

- En todas las páginas existe la posibilidad de volver a la página anterior mediante el botón de vuelta (back). Si es la primera página de la aplicación, pulsar el botón back la cierra, en caso de ejecutarse sobre un dispositivo móvil. Por el contrario, si se ejecuta sobre un ordenador o tableta, aparecerá en la esquina superior izquierda un botón de atrás que realiza las mismas funciones que el botón físico del teléfono.
- Puede suceder que el usuario pulse el botón central del dispositivo con lo que se saldrá de la aplicación, pero no se cerrará (a no ser que el sistema no disponga de memoria suficiente y la cierre) si no que se quedará dormida, esto también sucede cuando recibimos una llamada. Por ello hemos modificado la opción por defecto de la aplicación para que cuando se abra de nuevo, se resume volviendo al estado en el que estaba y no se cree una nueva instancia.
- La aplicación contiene una base de datos local implementada en SQLite, que es la encargada de almacenar la información de los diferentes libros.
- Con la aplicación se pueden leer tanto libros públicos como privados (públicos con lector DRM), pero es necesario que el usuario active la aplicación con la cuenta de su biblioteca para poder empezar a usarla.
- Los libros descargados (PDF, ePub, acsm) se pueden abrir de varias maneras, una de ellas es mediante el explorador de archivos de Windows, otra es mediante el selector de archivos de Windows (abierto desde el programa), otra mediante los tiles secundarios que se permiten crear por cada libro; por último se puede abrir el



último libro mediante voz y el asistente personal Cortana (solo en Windows Phone de momento).

## Aplicación “localizada”

Que una aplicación esté “localizada” significa que está totalmente preparada para soportar diferentes idiomas y de una manera muy sencilla podemos añadir nuevos idiomas a nuestra aplicación de entre los soportados por Windows.

Nuestra aplicación “OdiloTK” está localizada, incluyendo todas sus cadenas de texto en el archivo de recursos Resources.resw, de forma que para añadir un nuevo idioma, solo habría que traducir las cadenas del archivo de recursos creando un nuevo archivo con el prefijo del idioma (Ej: Resources\_EN.resw) y marcar en las propiedades del proyecto el nuevo idioma.

A continuación podemos ver una parte del archivo de recursos de la aplicación, para el idioma principal- Español.

Nombre	Valor	Comentario
Aceptar	Aceptar	Botón de mensaje de aviso
Activacion2.Text	Activación	Bloque de texto
Activar	Activar	Botón barra de comandos
Agregar.Label	Añadir archivo	Botón barra de comandos
Anclar	Anclar a Inicio	Botón barra de comandos
Atras.Label	Atrás	Botón barra de comandos
Autor.Content	Autor	Elemento de la caja de selección
Autor1.Text	Autor	Bloque de texto
AutorOpcion	Autor	Opción de ordenación
Biblio.Text	Biblioteca	Bloque de texto
Biblioteca.Label	Biblioteca	Botón barra de comandos
Borrar.Text	Eliminar	Elemento menú flotante
Cancelar	Cancelar	Botón de mensaje de aviso
Contrasena.Text	Contraseña	Bloque de texto
Cuenta.Label	Cuenta	Botón barra de comandos
Desactivar	Desactivar	Botón barra de comandos
Desanclar	Quitar de Inicio	Botón barra de comandos

Fichero de recursos

Windows 8.1 soporta, en el momento de su salida, más de 50 idiomas de pantalla seleccionables desde las opciones del dispositivo. Además, Windows 8.1 permite desarrollar aplicaciones que muestran texto RTL (de derecha a izquierda).

Para el resto de idiomas, en principio inglés, he utilizado para la traducción el **Kit de herramientas para aplicaciones multilingües** de Microsoft.



## Kit de herramientas para aplicaciones multilingües

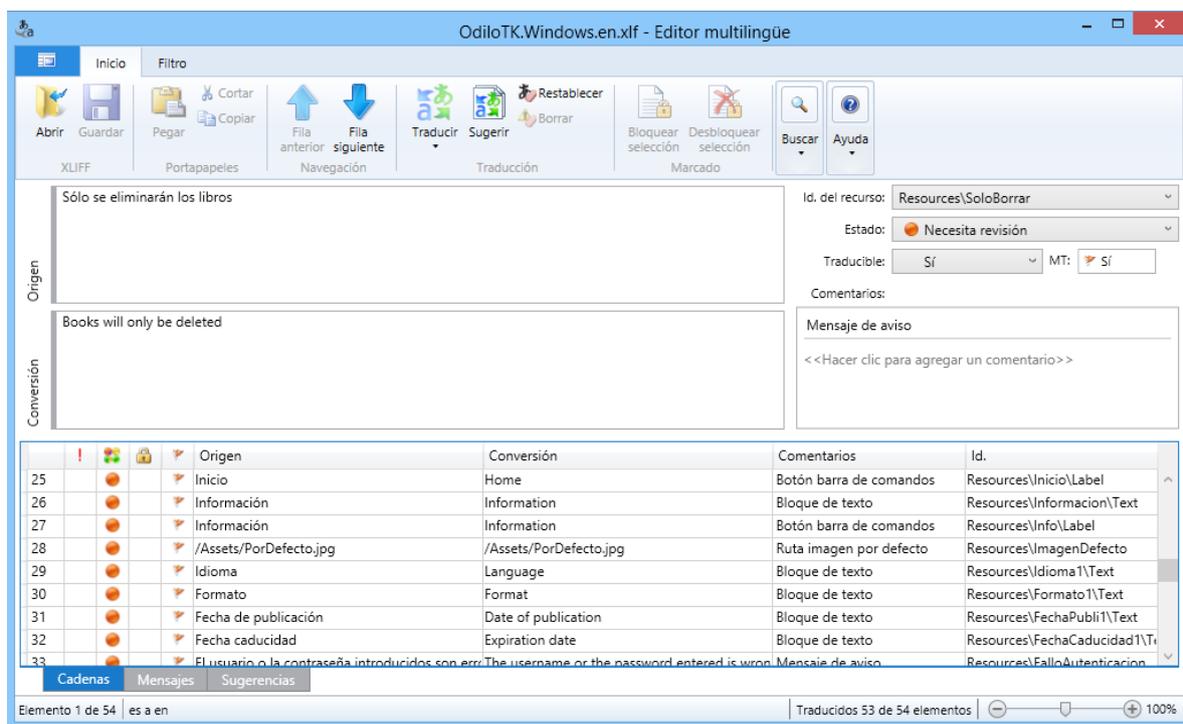
El Kit de herramientas para aplicaciones multilingües y Visual Studio trabajan en conjunto para optimizar el flujo de localización de las aplicaciones de la Tienda Windows y de la Tienda de Windows Phone. El kit de herramientas te ayuda a localizar tus aplicaciones con administración de archivos de localización, compatibilidad para traducciones y herramientas de edición.

Dispone de ciertas características como:

- Integración con Visual Studio IDE.
- Integración con pseudoidiomas.
- Integración con el Portal lingüístico de Microsoft.
- Integración con el traductor automático.
- Ciclo de ida y vuelta de exportación e importación de archivos de traducción.
- Editor de localización dedicado.

Para más información puede acudir a la documentación oficial de Microsoft sobre este conjunto de herramientas de traducción, en el siguiente enlace:

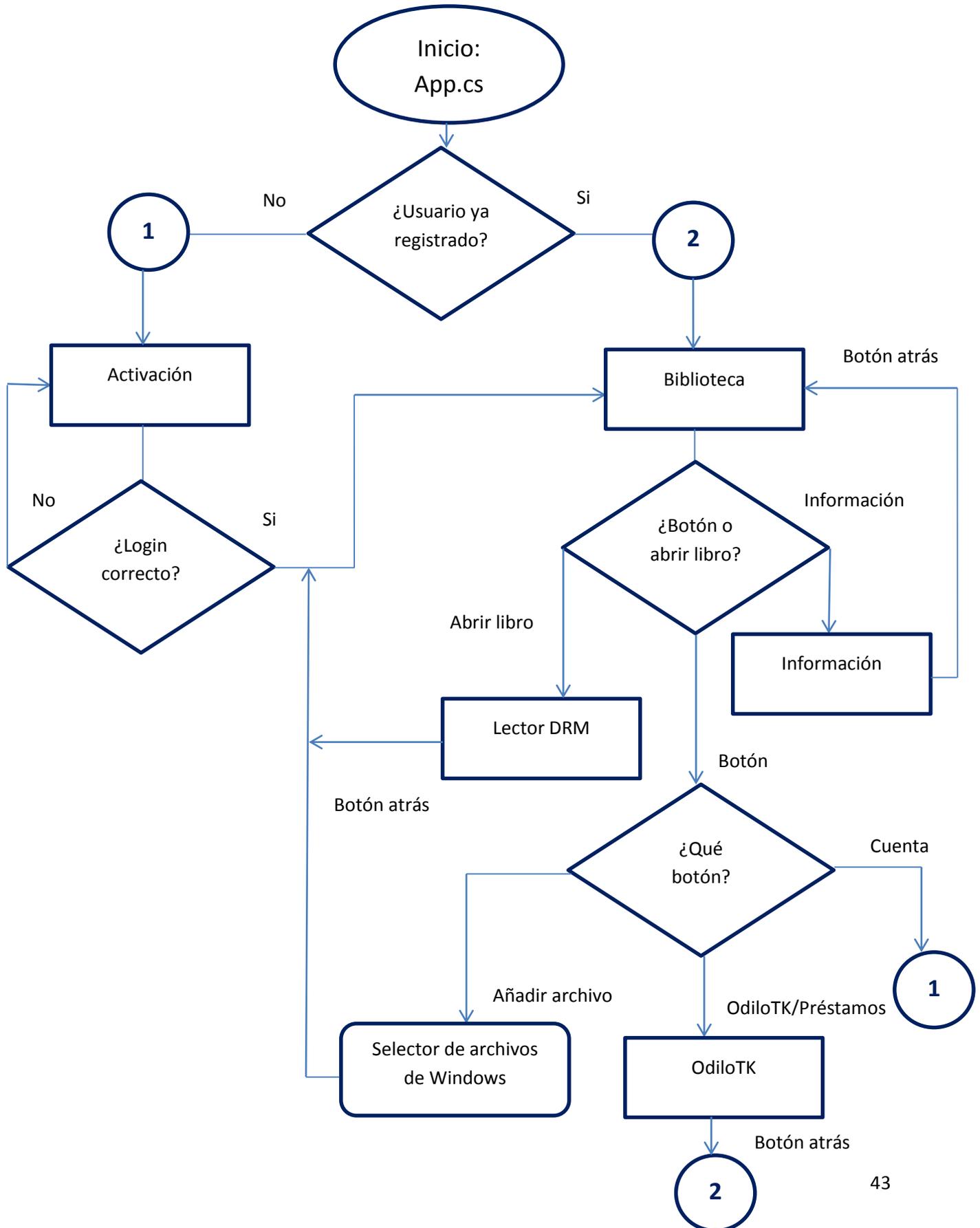
<https://dev.windows.com/es-es/develop/multilingual-app-toolkit/>



Editor del kit de herramientas multilingües



### Diagrama de flujo de la aplicación





## Páginas de la aplicación

### Activación

Activación

Usuario  
JoséLuis

Contraseña  
●●●●●●●●

Biblioteca  
The Library Network

Desactivar

Activar



Es la primera página de la aplicación, pero ésta solo aparece mientras el usuario no ha iniciado sesión, es decir, la primera vez. Volverá a aparecer si el usuario desactiva la aplicación.

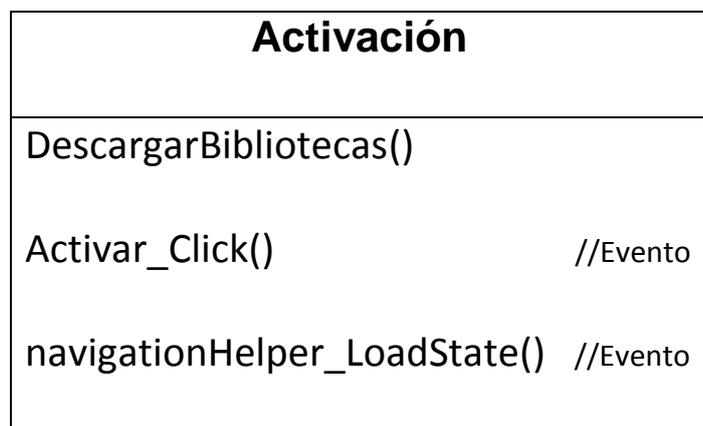
El usuario tiene que elegir su biblioteca de la lista desplegable e introducir sus credenciales para poder activar la aplicación, ya que es necesario para poder utilizarla. Una vez pulsado el botón activar, si las credenciales eran correctas, se navega a la página “Biblioteca”, pudiendo volver a “Activación” pulsando el botón “Cuenta”.

La lista de bibliotecas disponible se obtiene mediante la conexión a un servicio web REST de la empresa Odilo. Por supuesto, se necesita disponer de conexión a internet para esto y para proceder con la activación. De no ser así, se avisará al usuario y no se podrá realizar la activación hasta tener conexión.

Si el usuario había activado anteriormente la aplicación en este o en otro dispositivo Windows, automáticamente aparecerán los datos de la cuenta, por lo que el usuario solamente tendrá que pulsar sobre el botón “Activar”.

Por tanto, después de autenticar al usuario, se guarda tanto en local como en remoto (remoteSettings: usa OneDrive) los datos introducidos.

**A continuación podemos ver el diagrama de clases UML simple:**

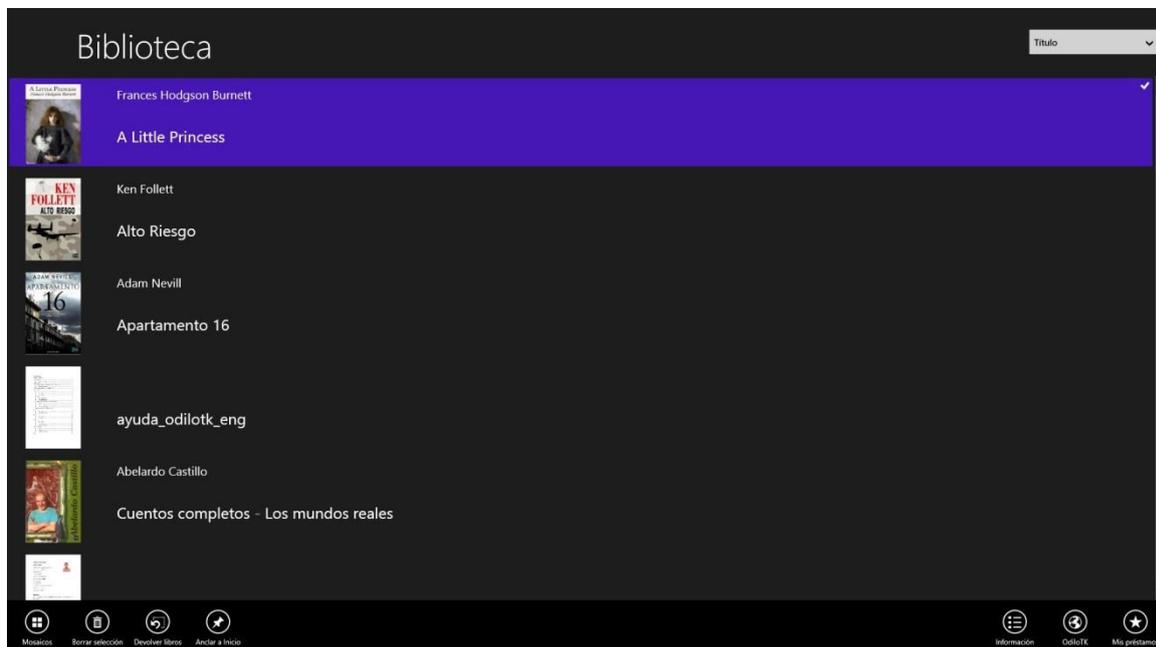


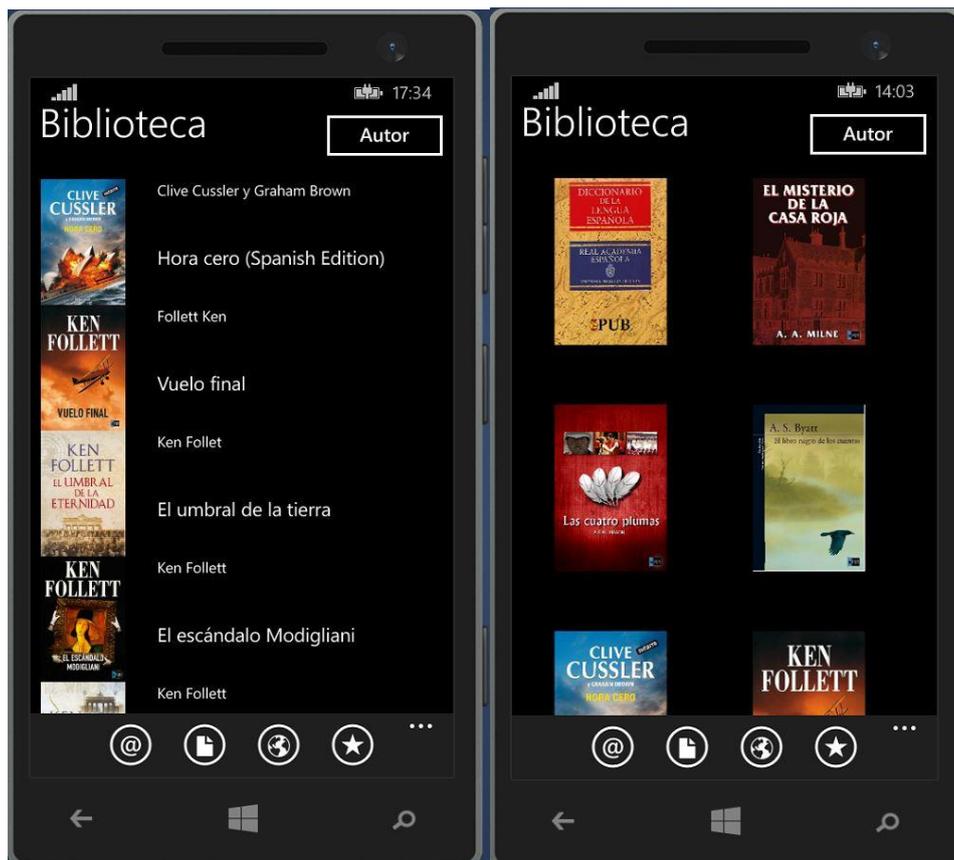
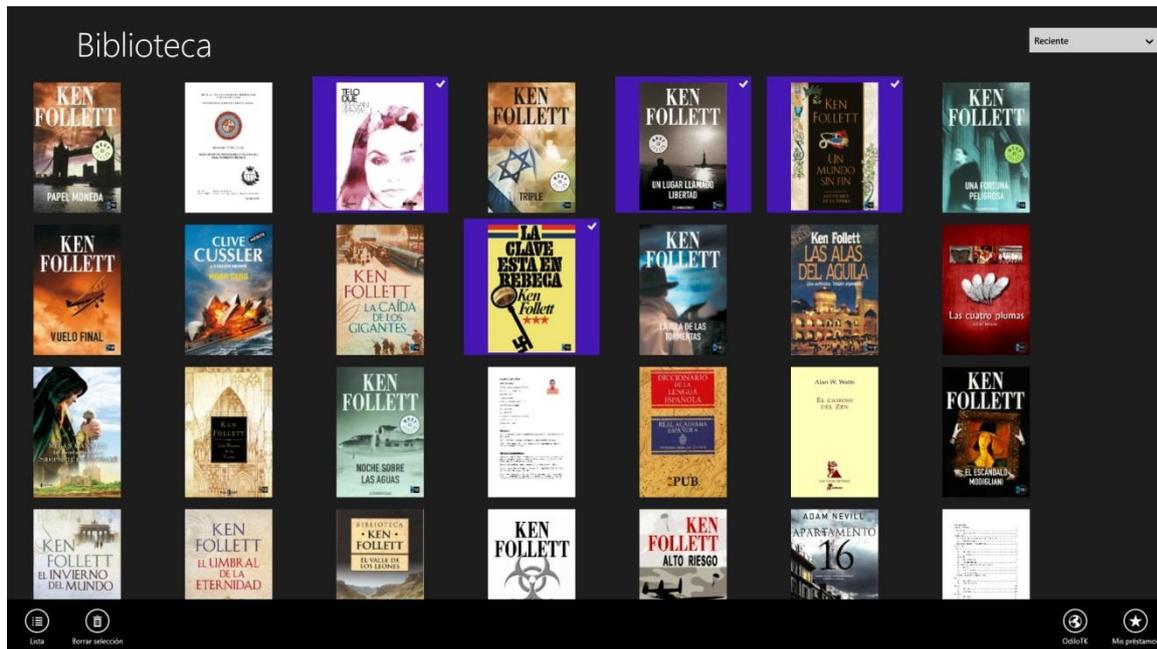
- *DescargarBibliotecas*: método que se encarga de llamar al servicio web REST de Odilo para descargar la lista de bibliotecas con su información en json, tratar la información y rellenar la lista desplegable (ComboBox).
- *Activar\_Click*: método que gestiona tanto la activación como la desactivación de la aplicación. Para ello comprueba si hay conexión a internet y comprueba que los datos sean válidos y guarda estos datos tanto el local como en remoto.



- *navigationHelper\_LoadState*: este método se ejecuta automáticamente cuando se carga la página, como el constructor, sirviendo para cargar el estado deseado en la página. Se encarga de rellenar los tres datos del usuario, tanto de local (si la cuenta está activada) como de remoto (si está desactivada) y de guardar los parámetros de navegación, por ej. La lista de libros a abrir pasada desde el explorador de archivos de Windows. También se realizan algunos ajustes en la interfaz gráfica, si se está ejecutando en un teléfono.

## Biblioteca





Esta es la Página Principal En esta página tenemos una lista de los libros que hemos añadido a nuestra aplicación (formato ePub o PDF). En las imágenes anteriores vemos el



modo “lista” y en la página siguiente se muestra el modo “mosaicos”. Para intercambiar entre las dos vistas, pulsamos el primer botón que podemos ver en las imágenes.

El siguiente botón “añadir archivo”, nos lanza un selector de archivos del sistema operativo que nos permite seleccionar libros de nuestro disco duro o almacenamiento en la nube, para añadirlos a la aplicación (copiándolos).

A continuación tenemos el botón “eliminar libro” que nos permite borrar uno o varios libros seleccionados de nuestra aplicación.

El siguiente botón “información”, nos llevará a la página que contiene la información del libro seleccionado.

El botón “OdiloTK” navegará a la página **OdiloTK**, detallada anteriormente.

Cuando no tenemos seleccionado ningún libro, como se puede ver en la imagen anterior, aparece un botón “lectura” que nos permite abrir directamente el último libro leído por el usuario.

También existe una caja de selección en la esquina superior derecha, que nos permite ordenar la lista de libros por tres criterios diferentes. Estos son: autor, título y más reciente.

**A continuación podemos ver el diagrama de clases UML simple:**

<b>Biblioteca</b>	
navigationHelper_LoadState()	//Evento
OdiloTK_Click()	//Evento
Listar_Click()	//Evento
OrdenarLibros()	
BotonAgregar_Click()	//Evento
ListaLibros_SelectionChanged()	//Evento
ListaLibros_DoubleTapped()	//Evento



BotonEliminar_Click()	//Evento
BotonInfo_Click()	//Evento
BotonLeer_Click()	//Evento
GuardarArchivos()	
Ordenado_SelectionChanged()	//Evento
Ordenado2_Click()	//Evento
CambiarBotonAnclar()	
BotonAnclar_Click()	//Evento
RejillaLista_Holding()	//Evento
LeerXml()	
EliminarLista()	//Evento
BotonActivacion_Click()	//Evento
BotonSincronizar_Click()	//Evento

- *navigationHelper\_LoadState*: este método se ejecuta automáticamente cuando se carga la página. Se encarga de recuperar los libros desde la base de datos SQLite, también de comprobar cómo se ha activado la aplicación, para realizar la acción que corresponda según el caso, como añadir los libros a la biblioteca o abrir un libro con el lector DRM. Para ello llama a la función “GuardarArchivos”
- *OdiloTK\_Click*: este evento obtiene una cookie conectándose a un servicio web de Odilo, además se ejecuta tanto si se pulsa el botón de “OdiloTK” como el de “Mis



préstamos”, navega hacia la página “OdiloTK” y le pasando como parámetro la opción seleccionada.

- *Listar\_Click*: evento que se ejecuta cuando se pulsa sobre el botón “Mosaicos/Lista” que cambiará según la el estado. Sirve para realizar el cambio de vista de la lista de libros, entre las opciones: lista y mosaicos. Para ello utiliza tipos de plantillas definidas en XAML, usando unas distintas para teléfono.
- *OrdenarLibros*: función ejecutada para realizar la ordenación de la lista de libros (no modifica la base de datos), se ejecuta al cargar la página y al modificar la lista desplegable donde se indica el tipo de ordenación: autor, título y más reciente.
- *BotonAgregar\_Click*: Este evento se ejecuta al pulsar el botón “Añadir archivo”, se llama al selector de archivos de Windows y si se ha seleccionado algo, se procede a incluirlos en la lista de libros y a abrir el último, llamando a “GuardarArchivos”.
- *ListaLibros\_SelectionChanged*: evento invocado al modificar la selección de la lista de libros (ListView). Se encarga de modificar los botones que son visibles en la barra de comandos de debajo de la pantalla, según la selección. La lista permite selección múltiple; por último, este evento no hace nada si la aplicación se ejecuta en un teléfono.
- *ListaLibros\_DoubleTapped*: evento que se invoca al hacer doble clic sobre la lista de libros. Se comprueba que se haya seleccionado un libro y si el libro está descargado o solo hemos añadido su información (del archivo ACSM). Si es necesario, se descarga el libro y posteriormente se abre, navegando a la página “LectorDRM” y actualizando su fecha de última apertura en la base de datos.
- *BotonEliminar\_Click*: método que se encarga de borrar los libros seleccionados por el usuario, tanto los archivos como la información almacenada en la base de datos, también elimina los tiles secundarios de esos libros (en caso de existir). Si no se selecciona ningún libro, se preguntará al usuario si desea eliminar todos los libros y en caso afirmativo, se llamará al método “EliminarLista” que realizará esa función.
- *BotonInfo\_Click*: cuando se llama a este método, se navega hasta la información del libro seleccionado. Puede llamarse mediante el botón o en caso del teléfono, mediante la opción del menú flotante que aparece al mantener pulsado un libro de la lista.



- *BotonLeer\_Click*: método invocado al pulsar sobre el botón “Leer ahora”. Lo que hace es navegar hasta la página “LectorDRM” pasándole el último libro leído como parámetro, para que se abra su contenido directamente.
- *GuardarArchivos*: Esta función es la más importante de toda la biblioteca, recibe como parámetro una lista de libros (IStorageItem) y devuelve una cadena con el título del último libro guardado. Realiza un tratamiento diferente según el tipo de archivo de cada libro: ePub, PDF o acsm. Se encarga de obtener la información de los libros y su portada, además de guardar esta información en la base de datos SQLite y de copiar cada libro y su portada a la memoria local de la aplicación (local folder).
  - En caso de abrir un fichero .acsm, que contiene los metadatos del libro, se realiza la descarga del libro ePub y su portada correspondiente. Si no se dispone de conexión a internet, se añade solo la información disponible a la BBDD.
  - Si se trata de un fichero PDF, se utiliza como portada la primera página del archivo, utilizando para ello la API para el tratamiento de PDFs en Windows 8.1. No está disponible en Windows Phone, donde se muestra una imagen por defecto.
  - Cuando el archivo es un ePub directamente, se descomprime para extraer su archivo de metadatos, se realiza un análisis detallado de este archivo, para saber la imagen que se utiliza como portada y descomprimirla.
- *Ordenado\_SelectionChanged*: Este evento se ejecuta cuando el usuario modifica la selección de la lista desplegable (ComboBox) y se está ejecutando en un dispositivo distinto al teléfono. Se guarda en la memoria local la selección del usuario y se llama a la función “OrdenarLibros” para que ordene la lista de libros, pasándole como parámetro el modo de ordenación elegido
- *Ordenado2\_Click*: este evento se invoca al pulsar el botón para elegir el modo de ordenación cuando la aplicación se ejecuta en un teléfono. Este botón sustituye a la lista desplegable y permite seleccionar las mismas opciones: autor, título y más reciente. Cada vez que se pulsa el botón, se cambia a la opción siguiente de entre las tres. Equivale al método “Ordenado\_SelectionChanged” descrito anteriormente.
- *CambiarBotonAnclar*: mediante esta función, se realiza la comprobación y el cambio entre el texto y el icono del botón/opción anclar/desanclar, según si el libro seleccionado ya está anclado a inicio o no.



- *BotonAnclar\_Click*: este evento sirve tanto como para desanclar de inicio un libro como para anclarlo. Cuando se selecciona el libro u se ancla, se crea en el menú inicio un “tile” secundario con la portada del libro en cuestión. En la versión Windows, se pide confirmación para crear el tile secundario y para eliminarlo. Se llama a la función “CambiarBotonAnclar” descrita anteriormente.
- *RejillaLista\_Holding*: método que solamente tiene efecto en Windows Phone. Se invoca cuando se mantiene pulsado un libro de la lista, crea un menú flotante que contiene algunos elementos que en la versión de Windows están como botones en la barra de comandos. Estos elementos son: información, anclar/desanclar, eliminar y devolver libro. Al seleccionar una de estas opciones, se invoca el mismo código que invocan los botones correspondientes en la versión Windows.
- *LeerXml*: esta función se invoca para leer cada atributo de un xml, en nuestro caso, del archivo de metadatos que contiene la información del libro. Si el atributo correspondiente no existe, devuelve una cadena vacía para identificarlo.
- *EliminarLista*: este evento se invoca cuando no hay ningún libro seleccionado y se pulsa sobre el botón “Eliminar todo” y se acepta la confirmación que aparece mediante un cuadro de diálogo. Se realiza un borrado de la base de datos, de los ficheros almacenados en la memoria local, de los tiles secundarios y de las claves almacenadas también en la memoria local.
- *BotonActivacion\_Click*: Cuando se hace clic sobre el botón “Cuenta” se lanza este evento, que hace que naveguemos a la página “Activación”, descrita anteriormente, pasándole un parámetro para que sepa que venimos de la página “Biblioteca”.
- *BotonSincronizar\_Click*: evento que se ejecuta al hacer clic sobre el botón “Sincronizar” de la barra de comandos. Se llama a un servicio web REST de la aplicación web OdiloTK que nos devuelve una lista en Json con los préstamos que tiene el usuario. Por lo tanto, se realiza un cruce entre esta lista y la lista de libros almacenada en local, para comprobar si hay que actualizar la lista debido a cambios. En caso afirmativo, se procede a la descarga de los libros nuevos y su inclusión en la base de datos, abriendo el último de ellos en el lector DRM.



## Información

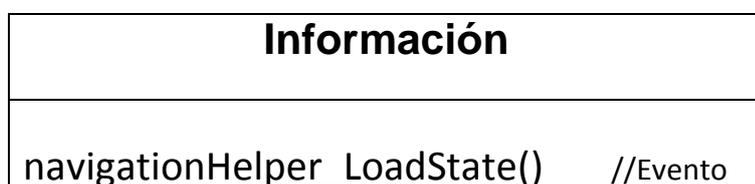


En esta página se muestra la diferente información del libro seleccionado, como el título, autor, formato, ubicación, descripción, etc.

Esta información está guardada en la BBDD local SQLite y se obtiene del propio libro ePub o de su fichero .ACSM si está protegido con DRM.

Dispone solamente de un botón “atrás” en la esquina superior izquierda, que navega hacia la página “Biblioteca” descrita anteriormente. (En teléfono se usa el botón físico)

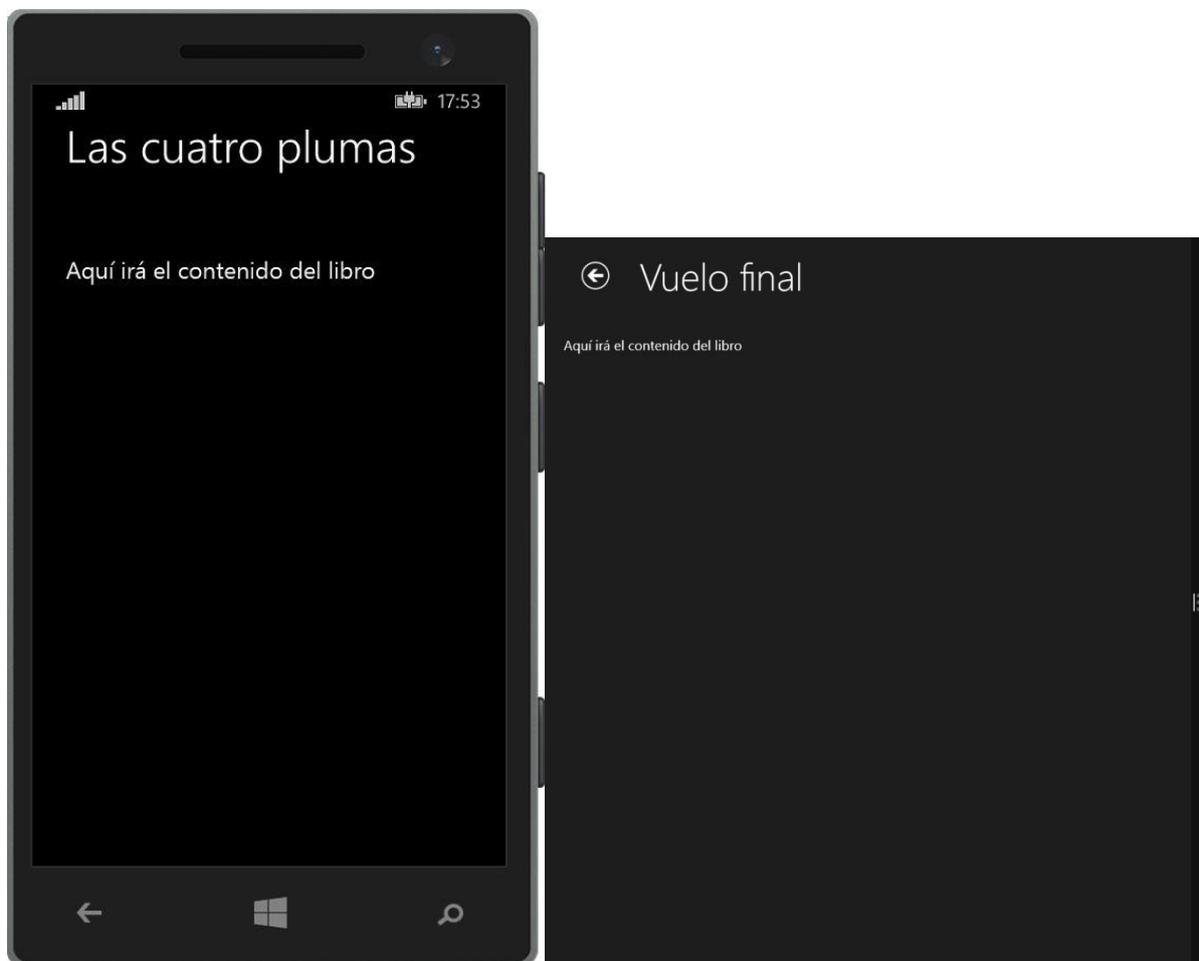
**A continuación podemos ver el diagrama de clases UML simple:**





- *navigationHelper\_LoadState*: este método se ejecuta automáticamente cuando se carga la página, como el constructor, sirviendo para cargar el estado deseado en la página. Se encarga de rellenar la información del libro que recibe como parámetro desde “Biblioteca”, estos datos están guardados en la BBDD local SQLite. También se realizan algunos ajustes en la interfaz gráfica, si se está ejecutando en un teléfono.

### LectorDRM

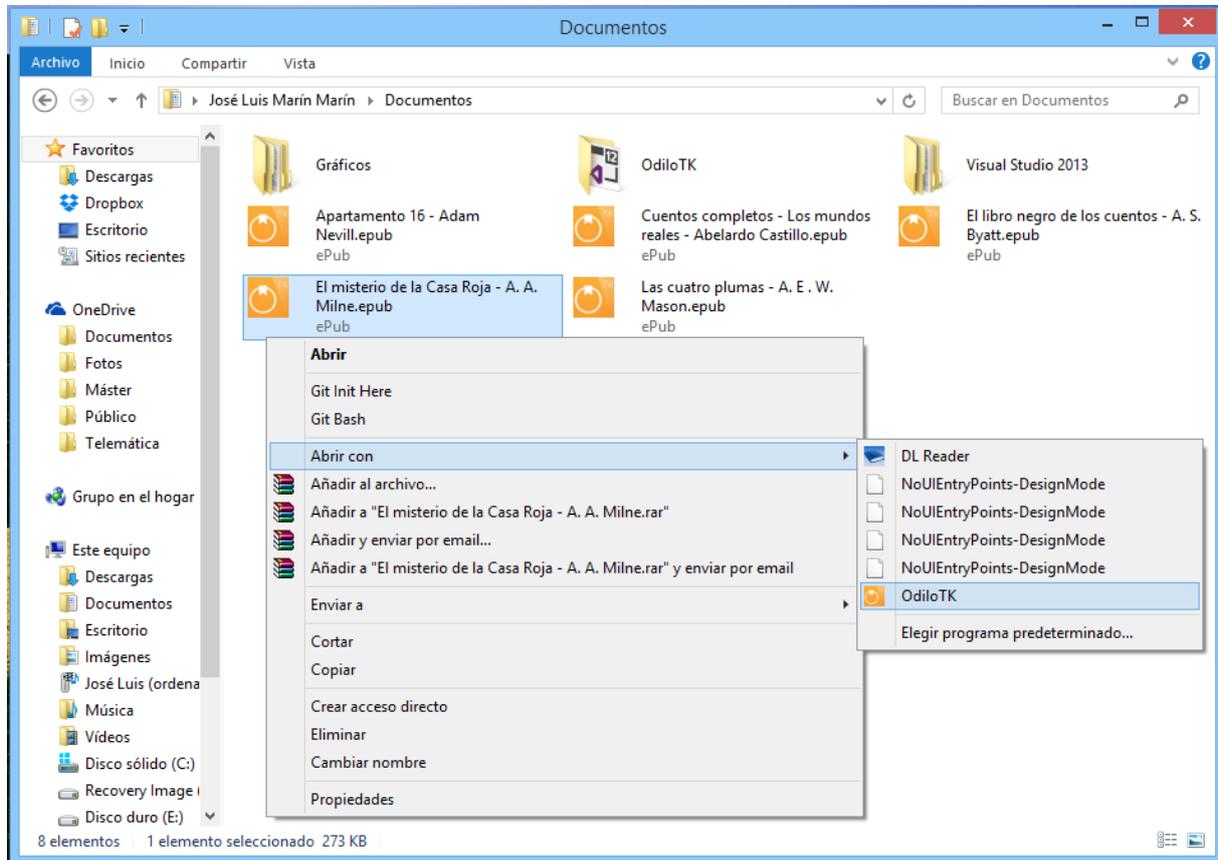


En esta página se integrará el lector digital de libros que usa la tecnología DRM de Adobe. Este lector es propiedad de Odilo y tiene los derechos de Adobe.

Podemos ver que dispone del título del libro que estamos leyendo y de un botón “atrás” en la esquina superior izquierda, que navega hacia la página “Biblioteca” descrita anteriormente. (En teléfono usa el botón físico)



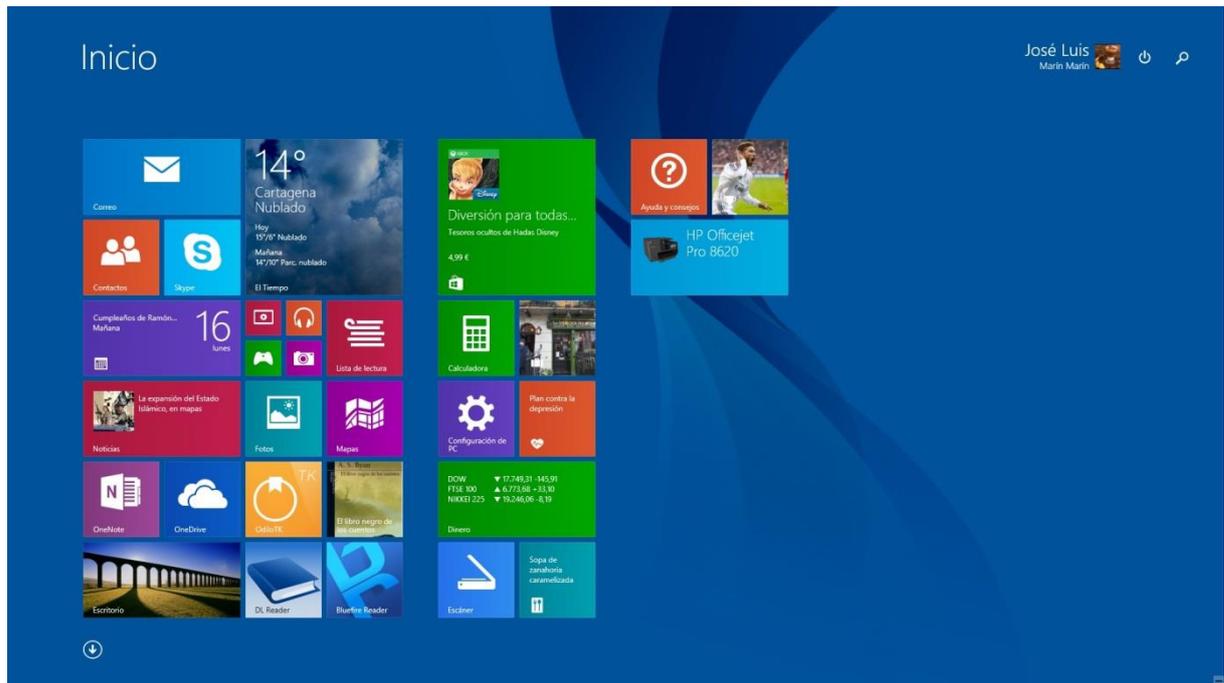
A esta página se llega automáticamente si se abre un libro con la aplicación, desde el explorador de archivos de Windows, como podemos apreciar en la figura siguiente.



Explorador de archivos de Windows

La aplicación nos permite abrir uno o varios libros a la vez desde el explorador de archivos de Windows. Se incluirán todos en la biblioteca y se abrirá la página “LectorDRM” con el último seleccionado.

También nos permite abrir un libro directamente desde el “tile” o azulejo del menú de inicio de Windows 8.1 como podemos ver en las siguientes imágenes:



Inicio Windows 8.1



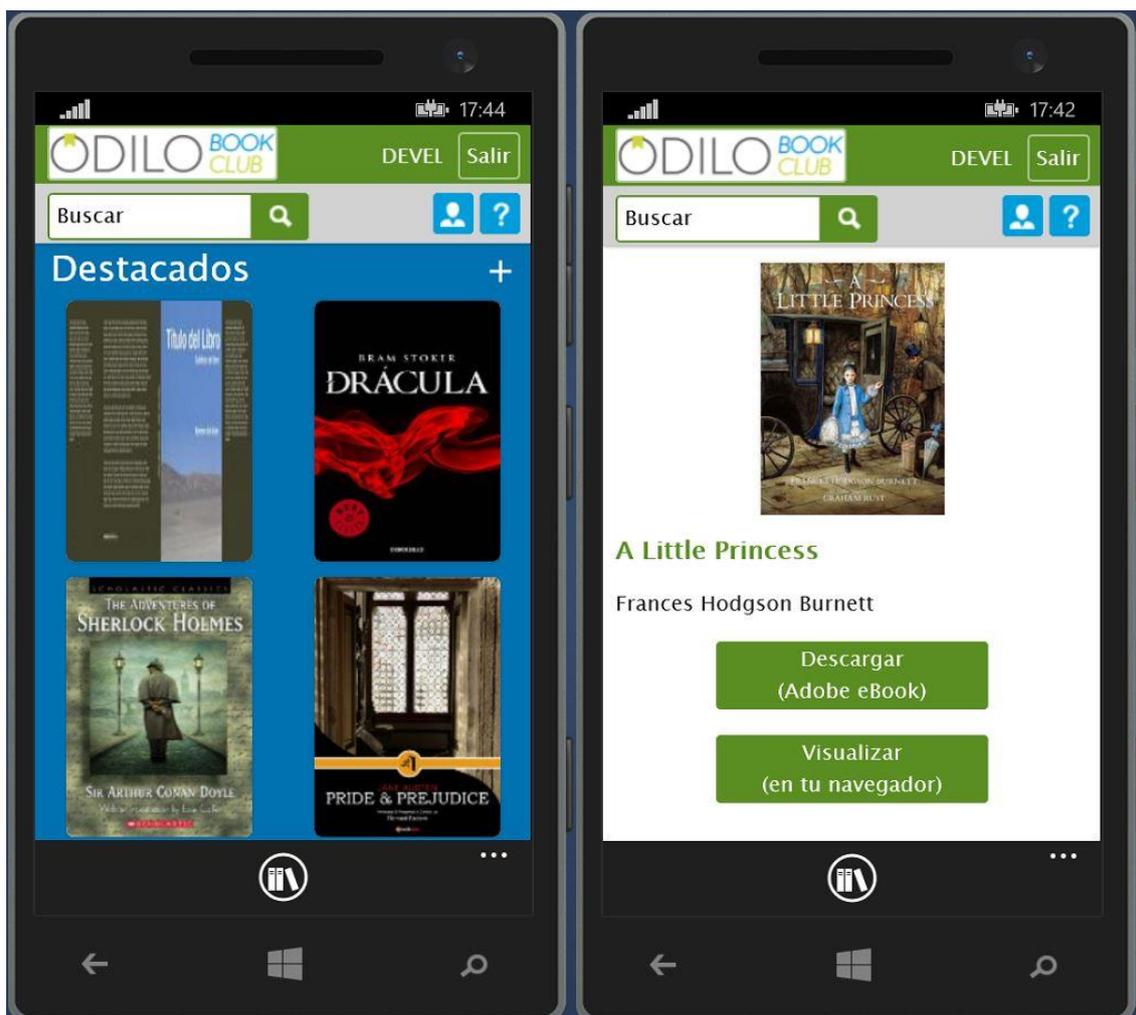


A continuación podemos ver el diagrama de clases UML simple:

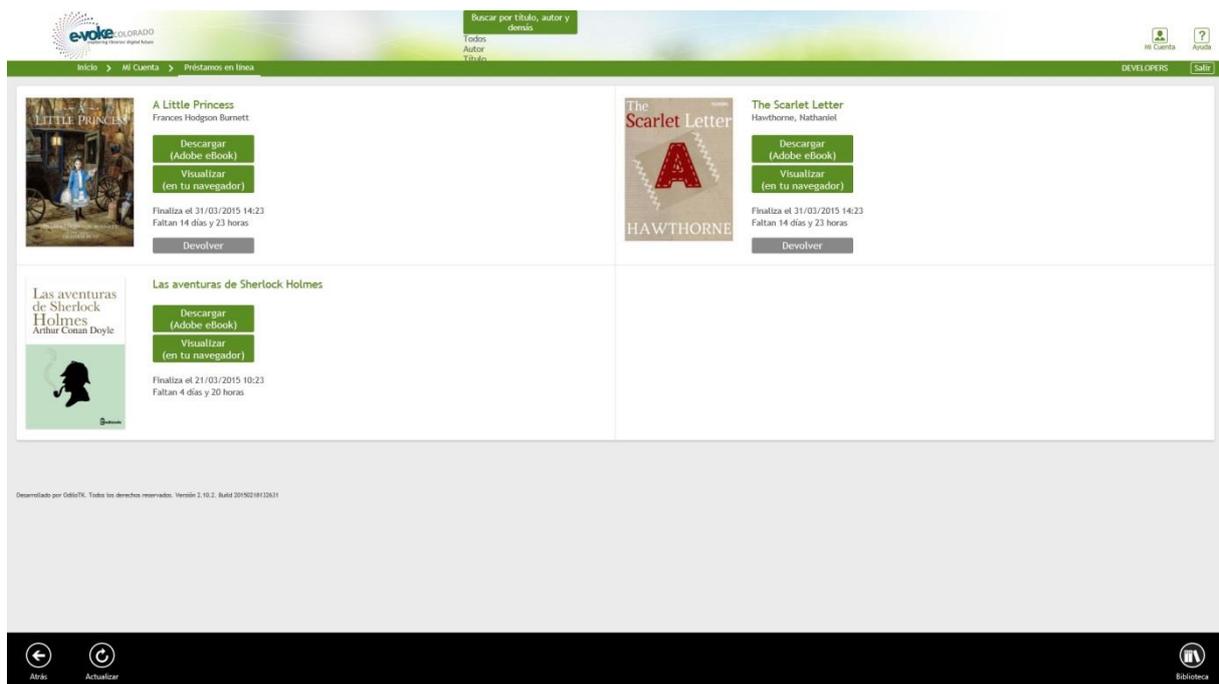
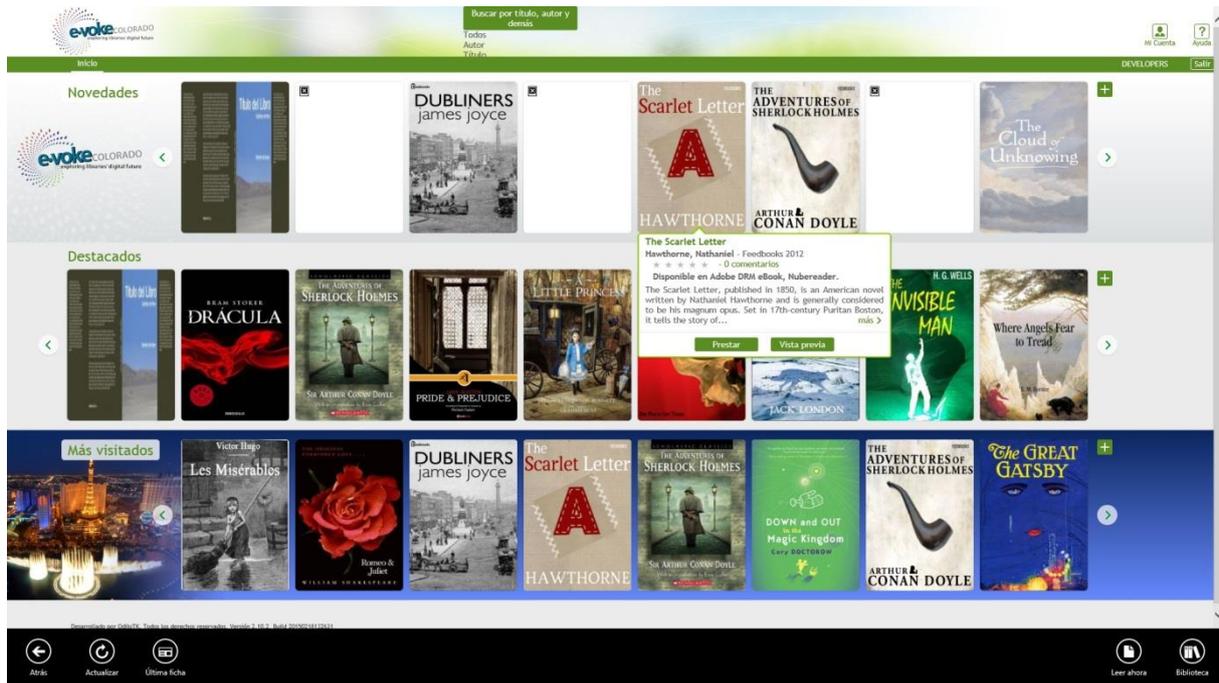


- *navigationHelper\_LoadState*: este método se ejecuta automáticamente cuando se carga la página, como el constructor, sirviendo para cargar el estado deseado en la página. Se encarga de obtener la información del libro que recibe como parámetro desde “Biblioteca”, estos datos están guardados en la BBDD local SQLite. También se realizan algunos ajustes en la interfaz gráfica, si se está ejecutando en un teléfono.

## OdiloTK



Solución integral para pc, tablet y smartphone dirigida a la gestión y lectura de libros electrónicos implementado en C# y XAML - UPCT



**Préstamos del usuario**

Es la página de la aplicación que incorpora un contenedor web (Webview) en el que se ha cargado la página (aplicación) web de la OdiloTK, desarrollada por Odilo.



Además incorpora una barra inferior, como se muestra en la figura, e incorpora una serie de botones que permiten al usuario interactuar con la página web y el resto de la aplicación de forma sencilla.

Dispone del botón “atrás”, para navegar hacia atrás, de la misma forma que se haría en un navegador web como Internet Explorer. Además, este botón navegará a la biblioteca cuando no hay historial de páginas anteriores.

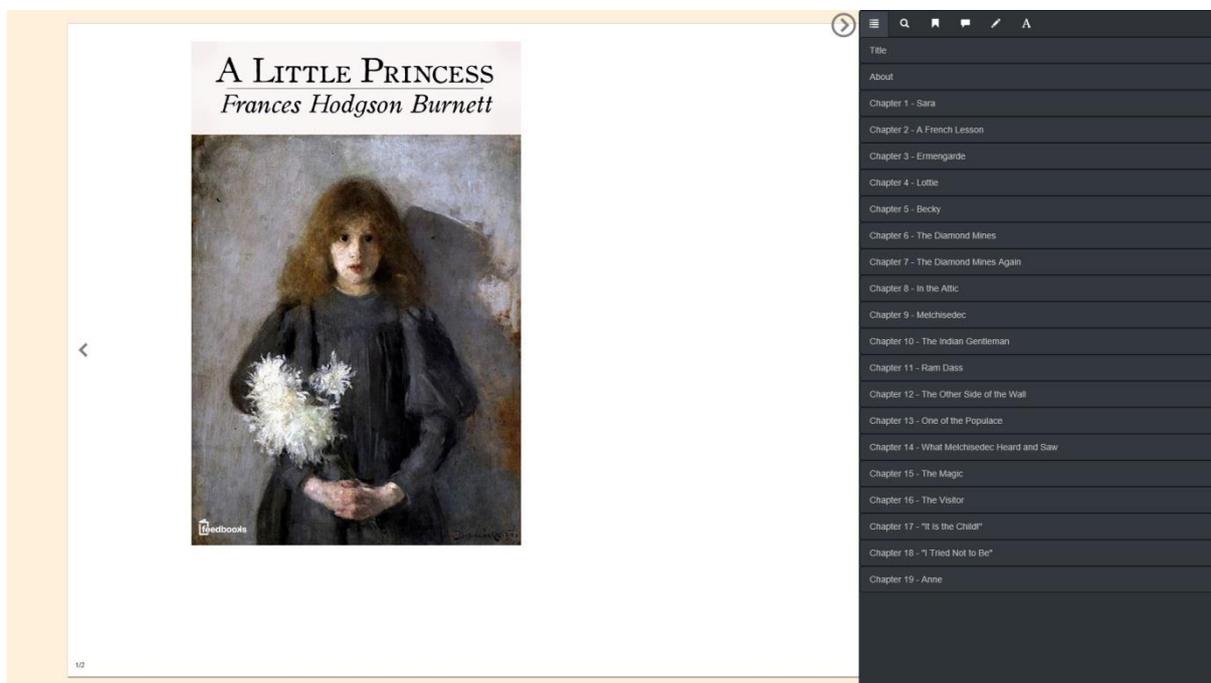
El botón “Última ficha”, navega directamente a la ficha del último libro visitado por el usuario.

El botón “Biblioteca” navega hacia atrás, es decir a la página “Biblioteca” directamente.

Por último, el botón “Actualizar” refresca la página dentro del contenedor Web (web view).

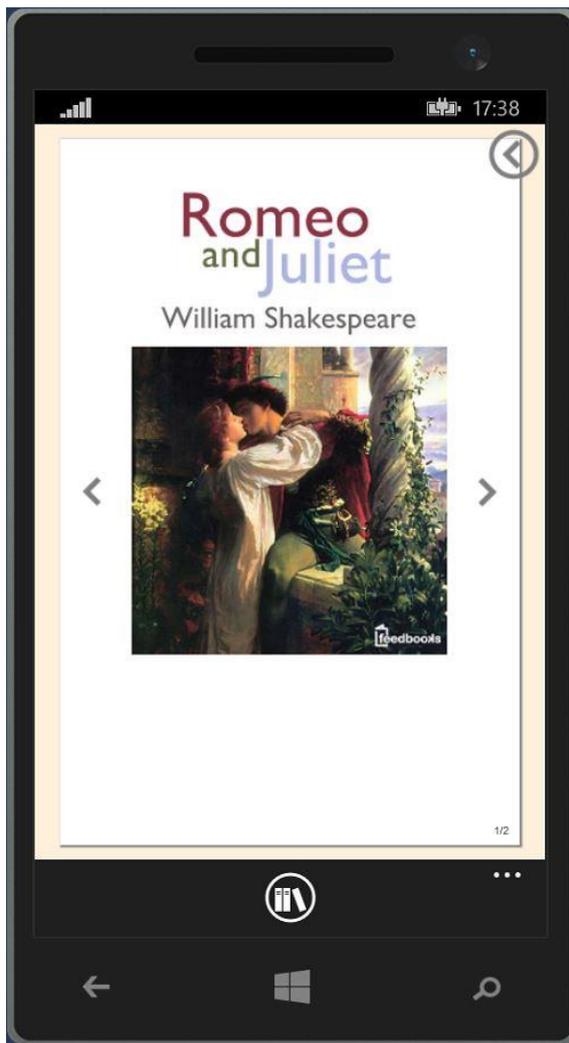
Como se puede ver en las figuras anteriores, podemos navegar a esta página de dos modos diferentes: “OdiloTK” y “Préstanos”, dependiendo del botón que hayamos pulsado en la página “Biblioteca”. Esto hará que la página mostrada en el contenedor web sea diferente.

A continuación vamos a describir el lector web “Nubereader” desarrollado por Odilo y que está incluido en la aplicación. Podemos verlo en las siguientes imágenes:



Lector Nubereader

# Solución integral para pc, tablet y smartphone dirigida a la gestión y lectura de libros electrónicos implementado en C# y XAML - UPCT





Para abrir un libro con el lector “Nubereader” hemos de ir a los préstamos del usuario y hacer clic en el botón “Visualizar”. Por el contrario, si hacemos clic en el botón “Descargar”, se descarga el libro, se añade a la biblioteca y se abre automáticamente con el lector DRM interno.

Cuando se abre un libro con “Nubereader”, automáticamente se guarda de forma que se pueda abrir el último libro abierto mediante el botón “Leer ahora” en la página “OdiloTK”. Además cuando no se ejecuta en el teléfono, se muestra el lector a pantalla completa.

**A continuación podemos ver el diagrama de clases UML simple:**

<b>OdiloTK</b>	
navigationHelper_LoadState()	//Evento
explorador_NavigationStarting()	//Evento
Biblio_Click()	//Evento
explorador_NavigationCompleted()	//Evento
botonAtras_Click()	//Evento
navegar()	
Page_Loaded()	//Evento
botonFicha_Click()	//Evento
detectaEmergente()	
explorador_ScriptNotify()	//Evento
botonLectura_Click()	//Evento



<code>explorador_UnviewableContentIdentified()</code>	<code>//Evento</code>
<code>BotonActualizar_Click()</code>	<code>//Evento</code>

- *navigationHelper\_LoadState*: este método se ejecuta automáticamente cuando se carga la página, como el constructor, sirviendo para cargar el estado deseado en la página. Se encarga de navegar a la página del “catálogo” o los “préstamos”, según se le pase como parámetro desde “Biblioteca”.
- *Explorador\_NavigationStarting*: Se encarga de hacer un seguimiento del historial de páginas navegadas, además de guardar el último libro abierto (por Nubereader) y establecer/quitar la pantalla completa para la lectura de los libros.
- *Biblio\_Click*: Realiza una navegación atrás entre páginas, es decir vuelve a la página “Biblioteca”.
- *explorador\_NavigationCompleted*: Se encarga de llamar a la función “detectaEmergente” que detallaremos a continuación y que ejecuta código JavaScript sobre el contenedor web.
- *botonAtras\_Click*: Realiza la navegación hacia atrás en el histórico de páginas visitadas en el contenedor web; finalmente navega hacia atrás, es decir, a la página “Biblioteca”.
- *navegar*: esta función es la encargada de realizar la navegación del contenedor web, a una página pasada como parámetro.
- *Page\_Loaded*: este método se ejecuta al cargar la página, establece el tamaño del contener web (web view) correctamente.
- *botonFicha\_Click*: este método navega hacia la última ficha del libro visitada por el usuario.
- *detectaEmergente*: función que se ejecuta periódicamente y sirve para detectar si hay enlaces al lector que se habrán externamente y modificarlos. Inyecta código JavaScript sobre el contenedor web. Si realiza alguna modificación, notifica a la aplicación, que será recogida por el método “explorador\_ScriptNotify” que veremos a continuación.
- *explorador\_ScriptNotify*: método que recoge la notificación enviada por el código JavaScript de “detectaEmergente”, se encarga de comprobar si estamos en una

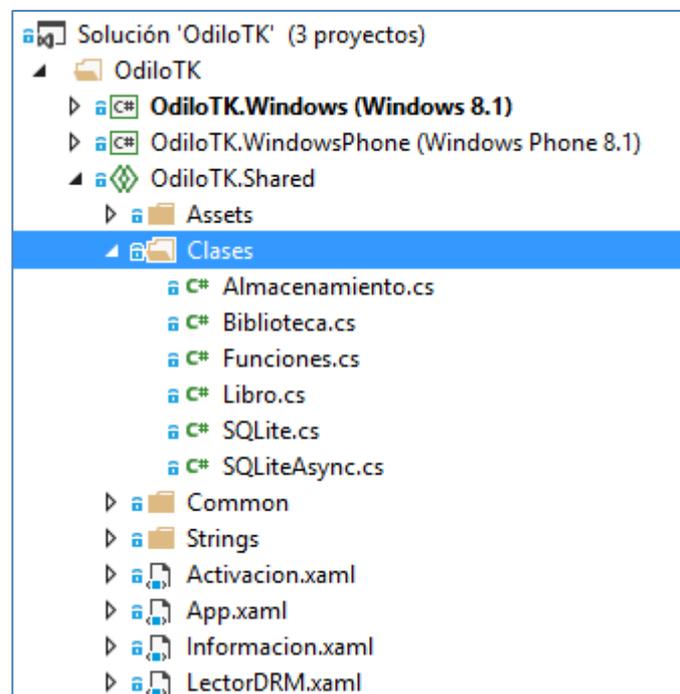


ficha de algún libro y capturar el ID de este. De este modo, el usuario puede navegar hasta esta última ficha mediante el botón “Ultima ficha”.

- *botonLectura\_Click*: Evento que realiza la navegación al último libro abierto, en el lector Nubereader.
- *Explorador\_UnviewableContentIdentified*: Este evento se produce cuando se descarga un libro, hace que se pueda descargar correctamente, nos lleva a la Biblioteca, que lo abrirá mediante el lector DRM.
- *BotonActualizar\_Click*: comprueba si hay conexión a internet y actualiza la página contenida en el web view.

## Clases de la aplicación

Se han creado varias clases que incluyen funciones estáticas que son útiles para el desarrollo de nuestro proyecto y que pueden ser llamadas desde cualquier parte de la aplicación. Las podemos ver en la siguiente figura:



Lista de clases



## Almacenamiento

Contiene los métodos para manejar el acceso al almacenamiento aislado, es decir, para guardar y consultar información tanto en pares clave/valor como en ficheros de texto. Contiene la versión de estos métodos con las nuevas API de Windows RT.

Utiliza principalmente el espacio de nombres “Windows.Storage” con las clases “Windows.Storage.StorageFile” y “Windows.Storage.ApplicationDataContainer”.

Contiene algunas funciones como “GuardarClaveAlmacenAislado” y “CargarClaveAlmacenAislado” que se utilizan frecuentemente.

## Biblioteca

Esta clase se utiliza para representar a una Biblioteca que implante el sistema de Odilo y a la cual nos podemos conectar mediante nuestra aplicación. La lista de bibliotecas disponibles nos la devuelve un servicio web REST de Odilo.

## Funciones

Clase que contiene una serie de funciones estáticas que pueden llamarse desde el resto del código de la aplicación. Algunas de estas funciones son:

- *MensajeAviso*: sirve para mostrar un cuadro de diálogo determinado con ciertos parámetros.
- *redimensionarImagen*: función que permite modificar el tamaño de una imagen, pasándole como parámetro el nuevo alto y ancho en píxeles.
- *IniciarSesion*: realiza el inicio de sesión en la aplicación web OdiloTK con los credenciales de nuestra cuenta. Para ello hace una petición a un servicio web REST de Odilo que nos devuelve una cookie con los datos de la sesión.
- *MostrarFecha* y *FormatearFecha*: funciones que se encargan de formatear las fechas en el formato deseado, según el país e idioma del usuario, así como de pasar una fecha en “string” a “DateTime”.
- *QuitarEtiquetas*: sirve para eliminar etiquetas HTML que a veces están presentes dentro de la descripción de los libros.
- *DescargarArchivo*: se utiliza para descargar archivos de internet, en nuestro caso la usamos para descargar ficheros “ePub”, “.acsm” y “.jpg”. Los ficheros descargados se guardan en el directorio local (LocalFolder).



- *DescomprimirArchivo*: método que utilizamos para descomprimir un libro en formato ePub y extraer tanto el archivo de metadatos como la imagen de portada.
- *hayInternet*: esta función la utilizamos para comprobar el estado de la conexión a internet, por cualquiera de los medios (Wi-fi, datos móviles, cable...)

## Libro

Esta clase contiene la definición para la tabla Libros, de la BBDD local SQLite que contiene nuestro dispositivo en el almacenamiento aislado.

El diseño de la tabla se ha realizado manualmente sobre esta clase, como puede apreciarse en la siguiente imagen.

```
using System;
using System.Collections.Generic;
using System.Text;
using SQLite;
using System.ComponentModel;

namespace OdiloTK
{
    public class Libro
    {
        [PrimaryKey, AutoIncrement, Column("Id")]
        public int Id { get; set; }

        [Column("NombreLibro")]
        public string NombreLibro { get; set; }

        [Column("NombreArchivo")]
        public string NombreArchivo { get; set; }

        [Column("Autor")]
        public string Autor { get; set; }

        [Column("Caratula")]
        public string Caratula { get; set; }

        [Column("Formato")]
        public string Formato { get; set; }
    }
}
```

Diseño de la tabla Libro

Hemos creado una única tabla de “Libros” que contiene los diferentes campos que pueden verse a continuación, en la herramienta de diseño de SQLite para Visual Studio, que se instala como una extensión disponible en la página de descargas de SQLite para .NET:

<http://system.data.sqlite.org/index.html/doc/trunk/www/downloads.wiki>

Esta herramienta también permite diseñar la tabla de una forma gráfica y visual.

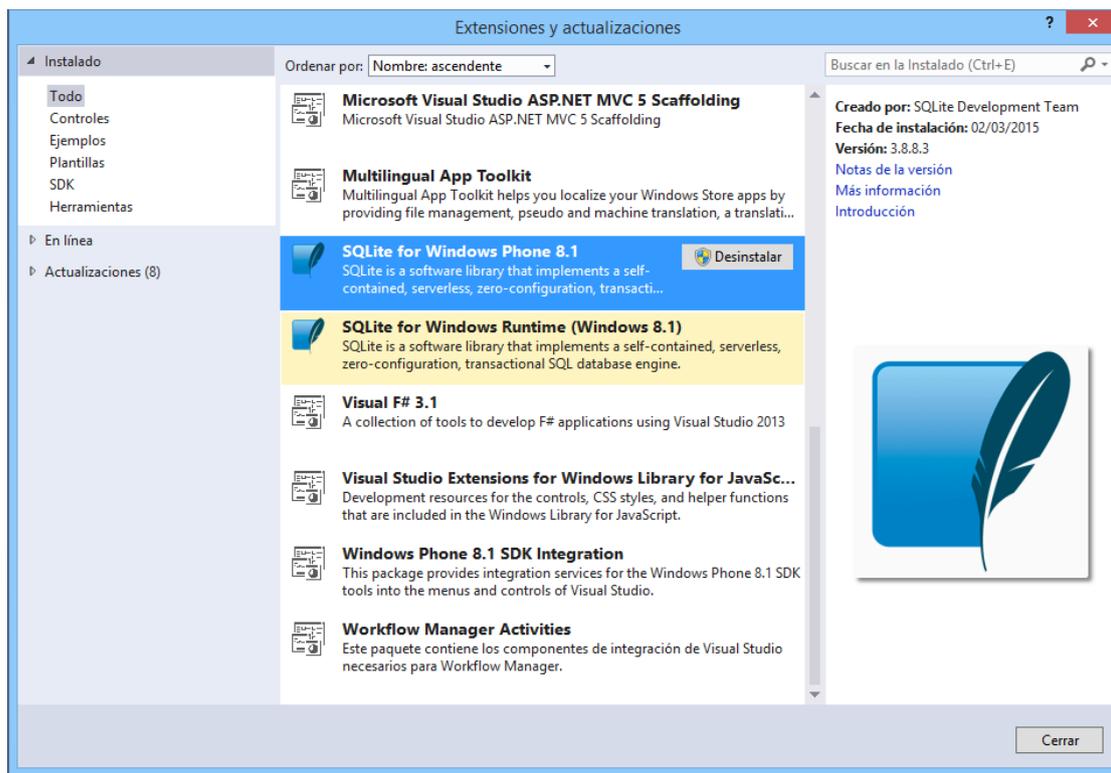


	Column Name	Data Type	Allow Nulls
▶ 🔑	Id	integer	<input type="checkbox"/>
	NombreLibro	varchar	<input checked="" type="checkbox"/>
	NombreArchivo	varchar	<input checked="" type="checkbox"/>
	Autor	varchar	<input checked="" type="checkbox"/>
	Caratula	varchar	<input checked="" type="checkbox"/>
	Formato	varchar	<input checked="" type="checkbox"/>
	Ruta	varchar	<input checked="" type="checkbox"/>
	Descripcion	varchar	<input checked="" type="checkbox"/>
	Idioma	varchar	<input checked="" type="checkbox"/>
	FechaPublicacion	datetime	<input checked="" type="checkbox"/>
	FechaApertura	datetime	<input checked="" type="checkbox"/>
	Descargado	integer	<input checked="" type="checkbox"/>
	Protegido	integer	<input checked="" type="checkbox"/>
	Distribuidor	varchar	<input checked="" type="checkbox"/>
	URLOperador	varchar	<input checked="" type="checkbox"/>
	Caducidad	datetime	<input checked="" type="checkbox"/>
	Transaccion	varchar	<input checked="" type="checkbox"/>
	Recurso	varchar	<input checked="" type="checkbox"/>
	Editor	varchar	<input checked="" type="checkbox"/>
	Identificador	varchar	<input checked="" type="checkbox"/>
	URLMiniatura	varchar	<input checked="" type="checkbox"/>
	Fuente	varchar	<input checked="" type="checkbox"/>
	HMAC	varchar	<input checked="" type="checkbox"/>

Tabla Libro

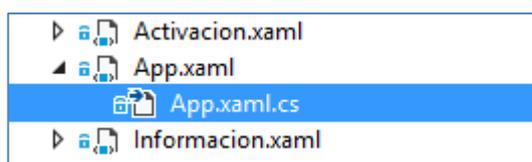
## SQLite y SQLiteAsync

Estas clases forman parte del motor de base de datos SQLite y se añaden automáticamente al proyecto al realizar la instalación de las extensiones para Visual Studio: “SQLite for Windos Phone 8.1” y “SQLite for Windows Runtime (Windows 8.1)”. Podemos verlas a continuación:



Extensiones de SQLite para Visual Studio

## App



App.cs

La clase “App.cs” está presente en toda aplicación Windows/Windows Phone y es la que se encarga de gestionar el inicio, la transición entre estados, la salida y los posibles errores de la aplicación.

En nuestro caso, le hemos añadido una serie de funciones:

- Decidimos el flujo inicial de la aplicación, para ello comprobamos si el usuario ya se ha registrado o no mediante una clave en el almacenamiento aislado. Dependiendo si está registrado o no, se navegará a “Activación” o “Biblioteca”.



- Hemos realizado la implementación de varios métodos, los cuales se diferencian en el punto de entrada a la aplicación, es decir, según se active la aplicación saltará un evento u otro.
  
- Estos métodos son:
  - *OnLaunched*: método principal de la clase, que se encarga de la activación cuando el usuario utiliza el tile principal o uno de los secundarios.
  - *OnFileActivated*: este evento se produce cuando el usuario abre uno o varios ficheros desde el explorador de archivos del sistema Windows.
  - *OnActivated*: evento que se lanza cuando la activación es realizada por otros medios distintos a los citados anteriormente. En nuestro caso, solamente se ejecuta en la versión de Windows Phone y hace el tratamiento en caso de que la aplicación sea activada mediante voz (Cortana) o mediante el selector de archivos, que previamente se ha tenido que abrir desde la página “Biblioteca”.

## ***Asistente personal Cortana***

Cortana es un nuevo asistente personal para smartphone con la tecnología de Bing e inspirado en los asistentes personales de la vida real. Cortana aprende aquello que te importa y usa esos conocimientos para anticiparse a tus necesidades.

Con el paso del tiempo, Cortana aprende cuáles son tus intereses, preferencias y rutinas. Te hace preguntas para averiguar qué cosas te importan. Cortana combina información relevante con el conocimiento de tu programación para ofrecer sugerencias proactivas que te puedan ser de utilidad. Los recordatorios basados en personas de Cortana te ayudan a recordar los datos importantes de las personas que más te importan.

Comenzó estando disponible en Windows Phone 8.1, luego en la Microsoft Band y, posteriormente en Windows 10, en todos los dispositivos que incluyan al mismo como sistema operativo.



Cortana reconoce el lenguaje natural (sin necesidad de utilizar comandos concretos) y utiliza Bing, Yelp y Foursquare como bases de datos. Sustituye a la búsqueda integrada de Bing en Windows Phone 8.1 y se activa pulsando el botón de búsqueda. A partir de la actualización a Lumia Denim, Cortana podrá ser activada diciendo "Ey Cortana"

Como Cortana es específica de cada país, se ha adaptado a los patrones cotidianos de cada región del usuario, su voz al idioma en concreto, la cultura y el habla. Por ejemplo, la versión británica de Cortana habla con un acento británico y utiliza modismos británicos, mientras que la versión china, conocido como Xiao Na, habla chino mandarín y tiene un icono con una cara y dos ojos.

También trata de satisfacer las necesidades según su uso, asimismo podrá visualizar información relevante sobre una gran diversidad temática, como por ejemplo, equipos deportivos locales, negocios, series de televisión y las bolsas de valores.



Inicio de Cortana mediante el botón de búsqueda

## Integración de aplicaciones con Cortana

Usa los comandos de voz con Cortana para iniciar una aplicación y especificar una acción o un comando que quieras que se ejecute. Por ejemplo, un usuario podría pulsar el botón Inicio y decir "Contoso Widgets, mostrar más vendidos" para iniciar una aplicación llamada "Contoso Widgets" e ir a la página "más vendidos" de la aplicación.

Cortana permite a los usuarios acceder a algunas de las funciones integradas del teléfono mediante comandos de voz. Estos son los comandos de voz del sistema, aunque pueden variar según el idioma de voz que se haya establecido para usarse en la función de voz del teléfono. A causa del requisito de certificación de voz, debes tener en cuenta estos comandos de voz del sistema a la hora de elegir un nombre para la aplicación o de escoger el elemento opcional CommandPrefix que se va a usar para la aplicación.



A continuación detallamos los pasos básicos para agregar funciones de comandos de voz e integrar Cortana en tu aplicación mediante entradas de voz o de teclado:

1. Crea un archivo VCD (Definición de comando de voz). Se trata de un documento XML en el que se definen todos los comandos de voz que el usuario puede decir para iniciar acciones o invocar comandos al activar la aplicación.
2. Registra los conjuntos de comandos en el archivo VCD con la función de voz del teléfono.
3. Administra el comando de activación por voz, la navegación dentro de la aplicación y la ejecución del comando.

Para más información acerca de la implementación, podemos dirigirnos al siguiente enlace:

<https://msdn.microsoft.com/library/windows/apps/xaml/dn630430.aspx/>

A continuación podemos ver un ejemplo de fichero VCD:





```
<VoiceCommands xmlns="http://schemas.microsoft.com/voicecommands/1.1">

  <CommandSet xml:lang="en-us" Name="examplevcd">
    <CommandPrefix>Recognize</CommandPrefix>
    <Example>Show default speech recognition</Example>

    <Command Name="showRecognition">
      <Example>Show default recognition</Example>
      <ListenFor>[Show] {options} {person}</ListenFor>
      <Feedback>Showing {options} {person}</Feedback>
      <Navigate/>
    </Command>

    <PhraseList Label="options">
      <Item>default recognition</Item>
      <Item>web search</Item>
      <Item>list constraint</Item>
      <Item>file constraint</Item>
    </PhraseList>

    <PhraseTopic Label="person" Scenario="Search">
      <Subject>Person Names</Subject>
    </PhraseTopic>

  </CommandSet>

  <!-- Other CommandSets for other languages -->

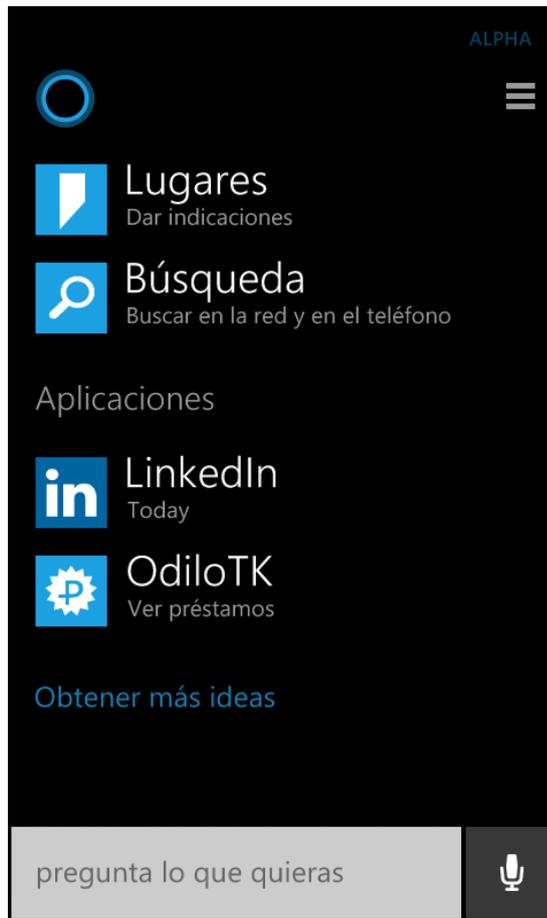
</VoiceCommands>
```

Ejemplo de fichero VCD (definición de comandos de voz)

## Funcionamiento de OdiloTK con Cortana

Hemos integrado la aplicación con Cortana, de forma que pueda activarse mediante comandos de voz. Para ello, hemos creado un archivo de definición de comandos de voz (VCD) que contiene los conjuntos de comandos para español e inglés, ambos idiomas admitidos por la aplicación.

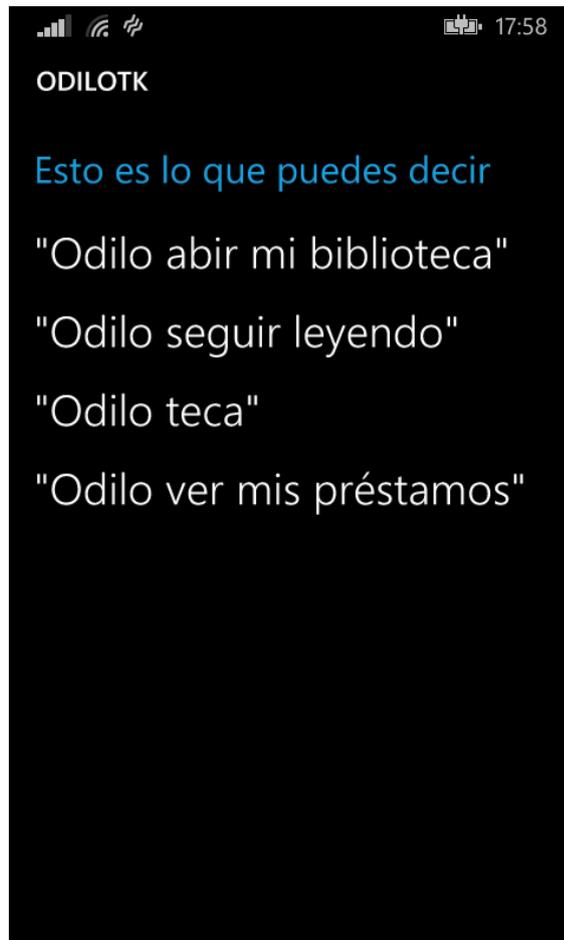
A continuación hemos registrado este archivo VCD en el código de la aplicación, de forma que al preguntarle a Cortana “¿qué puedo decir?” nos indica que la aplicación OdiloTK permite comandos de voz, como podemos ver a continuación:



Aplicaciones integradas con Cortana

Si hacemos clic en OdiloTK veremos lo que se muestra en la siguiente figura, es decir, una lista con ejemplos de lo que se puede decir para cada comando de voz. En este caso, la aplicación tiene registrados cuatro comandos de voz y muestra un ejemplo para invocar cada uno de ellos, como podemos ver a continuación.

Pero a cada uno de esos comandos, le hemos asignado más posibles formas de decirlo y activarlo, por ejemplo, para ir a la OdiloTK se puede decir “Odilo muestra mi catálogo”.



Ejemplo comandos de voz

Aquí podemos ver un ejemplo de código para el registro del conjunto de comandos de voz del teléfono, en el método OnNavigatedTo (paso 2).

```
protected override async void OnNavigatedTo(NavigationEventArgs e)
{
    if (e.NavigationMode == NavigationMode.New)
    {
        var storageFile = await Windows.Storage.StorageFile.GetFileFromApplicationUriAsync(new Uri("ms-appx:///ContosoWidgets.xml"));
        await Windows.Media.SpeechRecognition.VoiceCommandManager.InstallCommandSetsFromStorageFileAsync(storageFile);
    }

    // TODO: If your application contains multiple pages, ensure that you are
    // handling the hardware Back button by registering for the
    // Windows.Phone.UI.Input.HardwareButtons.BackPressed event.
    // If you are using the NavigationHelper provided by some templates,
    // this event is handled for you.
}
```

Ejemplo de registro de archivo VCD

A continuación, hemos seguido los siguientes pasos para administrar la activación y ejecutar los comandos de voz, es decir, una vez iniciada la aplicación e instalados los



conjuntos de comandos, se especifica cómo responde la aplicación a activaciones posteriores mediante comandos de voz:

1. Comprobar que la aplicación se haya activado mediante un comando de voz. Anula el evento *Application.OnActivated* y comprueba si *IActivatedEventArgs.Kind* es *VoiceCommand*.
2. Determinar el nombre del comando y qué se ha dicho. Obtén una referencia a un objeto *VoiceCommandActivatedEventArgs* desde la interfaz *IActivatedEventArgs* y consulta la propiedad *Result* de un objeto *SpeechRecognitionResult*.

De forma similar a como se puede ver en la posterior figura:

```
protected override void OnActivated(IActivatedEventArgs args)
{
    // Was the app activated by a voice command?
    if (args.Kind == Windows.ApplicationModel.Activation.ActivationKind.VoiceCommand)
    {
        var commandArgs = args as Windows.ApplicationModel.Activation.VoiceCommandActivatedEventArgs;
        Windows.Media.SpeechRecognition.SpeechRecognitionResult speechRecognitionResult = commandArgs.Result;

        // If so, get the name of the voice command, the actual text spoken, and the value of Command/Navigate@Target.
        string voiceCommandName = speechRecognitionResult.RulePath[0];
        string textSpoken = speechRecognitionResult.Text;
        string navigationTarget = speechRecognitionResult.SemanticInterpretation.Properties["NavigationTarget"][0];

        switch (voiceCommandName)
        {
            case "showWidgets":
                if (textSpoken.Contains("today's specials"))
                {
                    // Code to show today's specials.
                    // To do this, refactor OnLaunched into a method that you can call from both OnLaunched and OnActivated.
                    // The navigation logic will resemble this: rootFrame.Navigate(typeof(<today' specials page class>), e.Arguments);
                }
                else if (textSpoken.Contains("best sellers"))
                {
                    // Code to show the best sellers.
                }
                break;

            // Cases for other voice commands.

            default:
                // There is no match for the voice command name.
                break;
        }
    }
}
```

Activación por comandos de voz



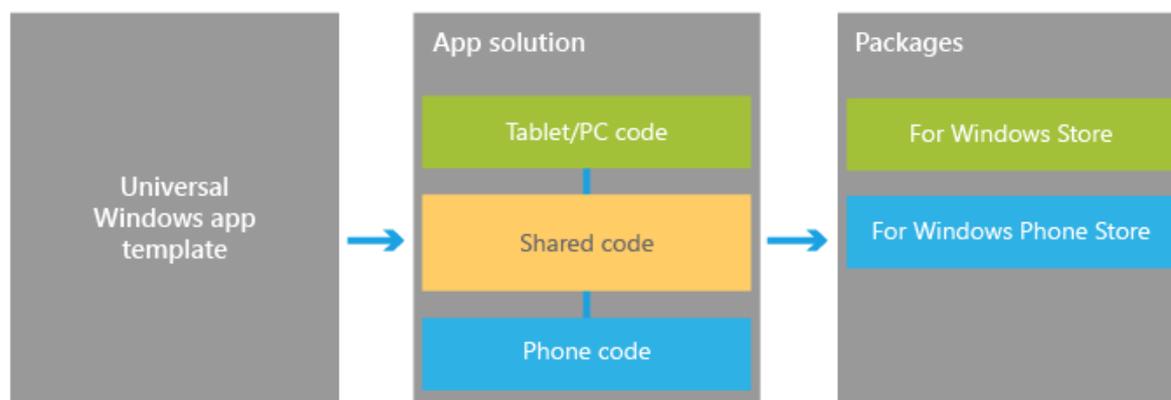
## Capítulo VIII – Aplicaciones universales

### *Introducción.*

Desarrolla aplicaciones una sola vez para todos los dispositivos de Windows con un tiempo de ejecución de Windows unificado y herramientas de VS que te permiten hacer compatibles con un dispositivo experiencias únicas en XAML, HTML y Direct X, y compartir el código que las hace compatibles en todos los dispositivos con C++, C# o JavaScript.

Cuando hayas terminado, podrás producir los paquetes de la aplicación que enviarás a la Tienda Windows y a la Tienda de Windows Phone mediante una única acción para que tu aplicación llegue a los usuarios de cualquier dispositivo de Windows.

En cuanto a la experiencia de la Tienda, los usuarios disfrutarán de una identidad de la aplicación compartida a través de la Tienda Windows y la Tienda de Windows Phone. La identidad compartida supone que, si alguien compra tu aplicación en la Tienda Windows, también podrá instalarla en un dispositivo desde la Tienda de Windows Phone con la misma cuenta Microsoft sin tener que volver a comprar la aplicación. También existe la opción de que incluya aspectos como las compras desde la aplicación.



Aplicaciones Universales en Windows 8.1

La creación de una nueva solución de Visual Studio mediante una plantilla de aplicación universal conllevará la creación de una solución de Visual Studio que incluya proyectos independientes para diseñar y presentar experiencias únicas en tabletas, PC y dispositivos con Windows Phone, así como proyectos compartidos que fomenten la reutilización del código que sea relevante para ambos.



## ***Aplicaciones universales en Windows 10***

La plataforma de Aplicaciones universales es una nueva plataforma o modelo de funcionamiento que Microsoft está implementando en Windows 10, el cual nos permitirá instalar una misma aplicación tanto desde la tienda de aplicaciones móviles como la tienda de aplicaciones de escritorio, sin importar que estos equipos sean diferentes. Esto en conjunto con el todo en uno de la tienda de Windows 10 puede ser de mucho beneficio para los usuarios.



Una de las opciones que ofrece es que los desarrolladores pueden optar por permitir que los usuarios puedan disfrutar de sus creaciones en distintos dispositivos con un solo pago. Este sistema multiplataforma tendrá su máximo exponente en la Tienda Única, un lugar desde el cual poder descargar las aplicaciones en el dispositivo que tengamos.

¿Por qué son tan importantes? Porque pone al alcance de los desarrolladores la oportunidad de que mediante la programación de un solo código de una App, puedan ofertar sus productos en varias plataformas, es decir, aumentan potencialmente el número de usuarios a los que llegar y consecuentemente las oportunidades de ingresos.

Éstas están pensadas para que los desarrolladores puedan crear de una sola vez aplicaciones que corran en cualquier dispositivo Windows, o en desarrollo, aplicaciones que puedan compartir casi todo su código fuente para que puedan ejecutar en dispositivos Windows.

Tal fin se logra gracias a que el compilador que Microsoft pone a disposición de los mismos utilizando Visual Studio, genera diferentes versiones de la aplicación para que se adapte al dispositivo donde vaya a ser usado, solamente habría que realizar adaptaciones de la interfaz de usuario y reducir considerablemente el esfuerzo de tener una aplicación disponible en varias plataformas



## Capítulo IX – Diseño de la aplicación

### *Introducción a la interfaz gráfica de usuario*

La interfaz gráfica de usuario (en idioma inglés **Graphical User Interface, GUI**) es un tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Habitualmente las acciones se realizan mediante manipulación directa para facilitar la interacción del usuario con la computadora.

En el contexto del proceso de interacción persona - ordenador, la *interfaz gráfica de usuario* es el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático.

Surge como evolución de la línea de comandos de los primeros sistemas operativos y es pieza fundamental en un entorno gráfico. Como ejemplo de interfaz gráfica de usuario podemos citar el escritorio o 'desktop' del sistema operativo Windows y el entorno X-Window de Linux y también Aqua de Mac OS X.



Escritorio

### *Interfaz gráfica en .NET (Windows Forms)*

Un formulario Windows representa la conocida ventana utilizada en los Sistemas Operativos de tipo Windows.



Por otra parte, un control es aquel elemento situado dentro de una ventana o formulario y que permite al usuario de la aplicación Windows interactuar con la misma, para introducir datos o recuperar información.

Dentro de .NET, las ventanas clásicas Windows reciben la denominación de Windows Forms o WinForms.

–NET Framework ofrece soporte extensivo para el desarrollo de aplicaciones para el sistema operativo Windows.

–El componente principal es Windows Forms = WinForms.

–La palabra “form” es prestada de Visual Basic y es analogía de desarrollo rápido de aplicaciones (RAD).

–La IDE de VS2005 provee un diseñador para la creación de WinForms.

–WinForms permite crear la interfaz gráfica (GUI) de las aplicaciones para Windows. (Windows Forms Designer).

–Utiliza el namespace: System.Windows.Forms.

–Permite crear aplicaciones conectadas a base de datos: Cliente/Servidor o Distribuidas.

## **Componentes de las aplicaciones Windows:**

–Formularios (Forms)

–Controles (Controls)

–Componentes (Components)

–Cuadros de diálogo (Dialog boxes)

## **System.Windows.Forms**

Es el espacio de nombres que contiene todos los tipos del entorno, a través de los cuales podremos desarrollar aplicaciones compuestas por los formularios Windows, junto a los correspondientes controles que permiten interactuar al usuario con la aplicación desarrollada.



El conjunto de clases, estructuras, enumeraciones,..., del espacio *System.Windows.Forms* permiten la creación de aplicaciones del tipo Windows.

## La Clase Form

Es la clase que contiene todos los elementos necesarios para la creación y manipulación de formularios (ventanas).

Tras instanciar un objeto de esta clase, dispondremos en nuestro programa de un formulario o ventana que podremos configurar según nuestras necesidades (Propiedades del formulario), y el cual podrá a su vez contener todo tipo de controles (cajas de texto, botones, ListBox, . . .) de forma que permita la interacción necesaria con el usuario.

### Ejemplo muy simple de ventanas hechas con .NET, WinForms:



Aplicación Winforms



## ***Interfaz metro en Windows.***

### **Características de una Aplicación Metro**

Una de las diferencias más importantes entre las Aplicaciones Metro y las aplicaciones clásicas a las que estábamos acostumbrados en versiones anteriores de Windows es la coherencia que existe entre todas ellas.

Microsoft está haciendo un gran hincapié en asegurar que existe una continuidad en sus principios, un lenguaje común en su diseño (Metro) y una homogeneidad en su uso que ayude al usuario a saltar de una a otra sin fricciones, o a empezar a usar una nueva aplicación sin apenas curva de aprendizaje. De entre las características que definen una aplicación estas son las más importantes:

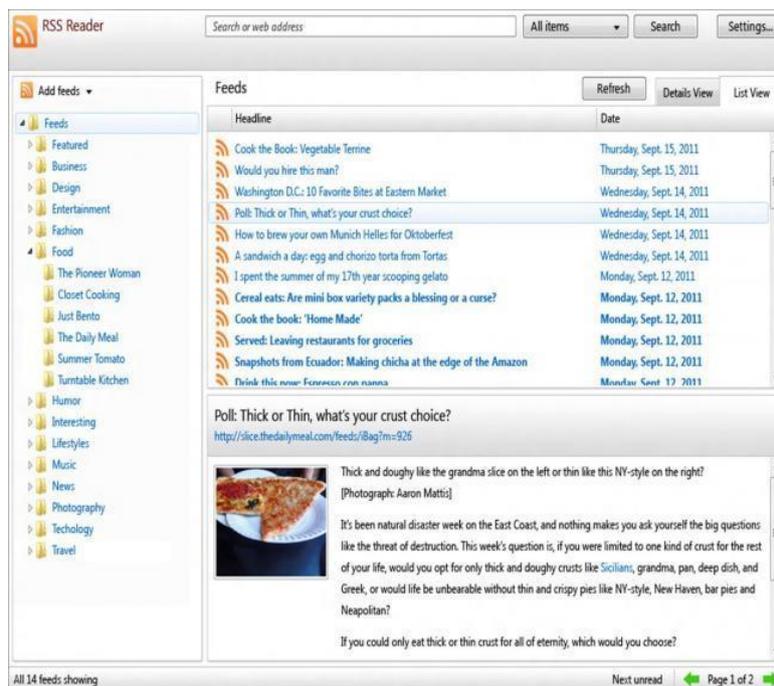
#### **Contenido antes que “cromo”**

Viene a representar toda la parafernalia a la que las aplicaciones tradicionales recurren para presentar la información en pantalla: bordes, sombras, transparencias, menús, etc. Adornos, en definitiva, que nos distraen de lo que realmente importa: el contenido. Precisamente en esto se centran las Aplicaciones Metro, en eliminar todos estos adornos y centrarse en mostrar contenido y hacer que brille, puesto que ayuda a construir experiencias inmersas para el usuario.

Sin embargo, estos adornos visuales tienen una función en muchos escenarios que, en caso de eliminarlos, debe ser sustituida por otros mecanismos. En concreto, gracias al “cromo” podemos definir navegación, disposición de elementos en pantalla e interacción. Para estos tres grupos las Aplicaciones Metro ofrecen una alternativa mejor:

- **Layout:** entre otros recursos, se utilizará espacio en blanco para separar elementos; y tipografías y jerarquías para agrupar y relacionar información.
- **Navegación:** la navegación se realizará con el propio contenido, en lugar de utilizar adornos específicos. Además se podrá recurrir a navegación jerárquica o plana en función del tipo de aplicación.
- **Interacción:** se permitirá al usuario manipular el contenido directamente, además de desplazar a una barra de la aplicación aquellos comandos que no se necesitan en el uso habitual de la misma. Los “Charms” de Windows 8 también permitirán ofrecer un sustituto de funcionalidad habitual como búsqueda u opciones de usuario.

En definitiva, el objetivo es pasar de aplicaciones como la que se puede ver en la siguiente imagen, a aplicaciones metro como se puede ver en la posterior:



Aplicaciones Win32 y estilo Metro

## Táctil primero, sin olvidar ratón y teclado

No es ningún secreto que las interfaces táctiles están de moda entre los usuarios. Por supuesto las Aplicaciones Metro tienen muy en cuenta esto e incluyen un soporte de primera clase para este tipo de interacciones. Entre otras características, se ha diseñado un nuevo lenguaje táctil específico para Windows 8 basado en varios principios:

- Simple y consistente, evitando cualquier gesto que pudiera resultar complejo.
- Gestos que puedan realizarse rápidamente.
- Conjunto limitado de gestos.
- Limitación de los gestos que se basan en tiempo.



## Anexo – Documentación técnica

El más importante de los detalles es como se ha comentado anteriormente que la aplicación requiere de la versión de Windows 8.1 o superior para poder instalarse, de hecho, no aparecerá en la tienda si no se cumple este requisito.

La aplicación está subida a la tienda de Windows Phone pero en modo Beta, esto quiere decir que solo está disponible para los usuarios invitados a probarla. Por tanto, la aplicación contiene iconos, pantalla de carga, etc que **podrían ser modificados en su versión final lanzada al público, así como la inclusión de nuevas funciones** o mejora de las existentes. Una vez subida la versión final a la tienda de Windows Phone en modo “normal” (ahora está en beta) se podrá descargar sin ningún problema.

En la tienda de Windows no se pueden subir aplicaciones en modo beta, esto es debido a que se permite directamente crear un paquete para distribuir la aplicación e instalarlo directamente sin pasar por la tienda. Para ello los usuarios que vayan a probar la aplicación, necesitan de una cuenta de Microsoft e instalar un certificado de desarrollador gratuito que pueden renovar. Hemos realizado pruebas para comprobar el correcto funcionamiento de la aplicación en Windows, mediante este tipo de distribución.

Como se ha comentado, la aplicación funciona independientemente del modelo del fabricante, está ligada al sistema operativo. Además soporta dispositivos con tipos de resoluciones muy diferentes, ya que se puede instalar desde en dispositivos móviles de pocas pulgadas hasta en grandes pantallas de ordenador. En todas ellas la aplicación funcionará correctamente, tanto en orientación vertical como horizontal.

Un usuario puede registrarse con su cuenta en varios dispositivos sin ningún problema y cambiar a otra cuenta/biblioteca Odilo, si dispone de varias. Es más, **cuando el usuario inicia sesión, se guardan sus credenciales en la nube** (remote settings), de forma que si posteriormente instala la aplicación en otro dispositivo o en el mismo, **se recuperan los credenciales que uso por última vez**, siendo esto un añadido muy útil.

La lista de APIS/servicios que usa la aplicación se encuentra recogida en la lista de capacidades que el usuario puede ver en la descripción de la aplicación de la tienda y en ciertos casos el usuario tendrá que aceptar que permite el acceso a esas APIS al instalar la aplicación. En nuestro caso, la lista de capacidades de OdiloTK se puede ver en la siguiente imagen, pero puede variar en el futuro si se incluyen actualizaciones con nuevas funciones.



ODILOTK

# detalles relacio

Información

Última actualización: 29/04/2015  
Versión actual: 2015.429.1256.4507  
Tamaño: 2 MB  
Categoría: libros y referencia

Editor

Nombre: JLMM

[Más de JLMM](#)

Notas

Idiomas admitidos: English; español;  
Requiere: internet connection; microphone;

[¿Qué es esto?](#)

ver    compartir    ...

Tienda - capacidades



## Conclusión

Tras el desarrollo de nuestro proyecto, me he dado cuenta de la gran cantidad de ventajas y algunas desventajas de utilizar la plataforma .NET de Microsoft, que desde mi punto de vista, son muchas más las ventajas y consideramos que este lenguaje C# nos ofrece una gran cantidad de herramientas y capacidades para el desarrollo de la aplicación, tanto como otras muchas que podrían usarse en diversas aplicaciones de mayor o menor índole e importancia. Gracias a las API que .NET nos ofrece para usar la mayoría de recursos, seríamos capaces de implementar una aplicación de casi cualquier tipo y que cumpla las necesidades deseadas en función de la demanda.

En nuestro caso concreto, la posibilidad de integrar las aplicaciones creando una única aplicación universal, acceder a la agenda y al almacenamiento aislado del teléfono, usar internet (conexión Datos/Wifi), la funcionalidad del contenedor “Web view”... han hecho posible cumplir los objetivos propuestos que deseábamos resolver mediante el desarrollo de nuestro proyecto. Combinando todas estas posibilidades, se podrían resolver infinidad de problemas, siendo la capacidad de la creación e aplicaciones universales y el uso de comandos de voz mediante Cortana, una gran ventaja que podría ir mucho más allá de los objetivos de nuestra aplicación.

He podido comprobar que el uso de la plataforma .NET para el desarrollo de aplicaciones para Windows 8.1 es un entorno ameno, con un montón de librerías y soporte que nos resulta muy útil a la hora de realizar ciertas tareas, cada vez es más amplio el conjunto de programadores que se deciden por Windows para desarrollar sus aplicaciones. Esto se debe principalmente a que es una plataforma relativamente nueva (Windows RT), muy potente e intuitiva, unido a que el número de dispositivos en el mercado no para de crecer. Además la integración con diversos servicios en red tales como Windows Azure, Onedrive, redes sociales, Cortana, etc. es tal que se abre ante nosotros un nuevo mundo de posibilidades.

Pienso que la aplicación aquí desarrollada es muy útil para gestionar y leer tus libros favoritos mediante cualquier dispositivo con sistema operativo Windows, por medio de una interfaz sencilla y fácil de manejar por cualquier usuario. Es una aplicación segura según los estándares de Microsoft y creada en un nuevo mercado potencial en continuo crecimiento.

Por otro lado, una ventaja muy importante de Windows y otros sistemas operativos en la actualidad es la existencia de la “tienda”, lugar donde los usuarios pueden buscar las aplicaciones que necesiten, de forma que cualquier aplicación subida puede ser descargada de una manera muy sencilla para el usuario, además de leer comentarios de otros usuarios e incluso calificar y opinar acerca de la experiencia de uso.

Por parte del programador, es muy sencillo realizar un seguimiento del nº de descargas, de las opiniones de los usuarios y los informes de cuelgues producidos, para darse cuenta de qué se puede mejorar en la aplicación. Cuando se añade o se modifica alguna función, tan solo basta con subirla a la tienda en modo “actualización” y a los usuarios que la tengan les aparecerá automáticamente que hay una actualización disponible, pudiendo especificar que funcione o cambios incorpora la actualización, en las notas de la descarga.



## Manual de usuario

### Instalación (ejemplo para Nokia)

Para poder instalar la aplicación es necesario tener un dispositivo móvil con el sistema operativo Windows 8.1 o superior y conexión a internet.

Debe entrar a la tienda “Windows Phone Store” o “Windows Store” y seleccionar el botón “buscar” de la barra de tareas que se muestra en la imagen siguiente, debe buscar “**Odilo**”. También puede llegar a la aplicación si entra en el apartado “aplicaciones” y la categoría “libros y referencia”, donde aparecerá entre todas las demás.



Tienda - OdiloTK

Una vez encontrada la aplicación, le aparecerá una pantalla como se muestra en la imagen. Debe seleccionar la opción de “instalar” y el proceso de descarga e instalación comenzará y se realizará de forma automática.



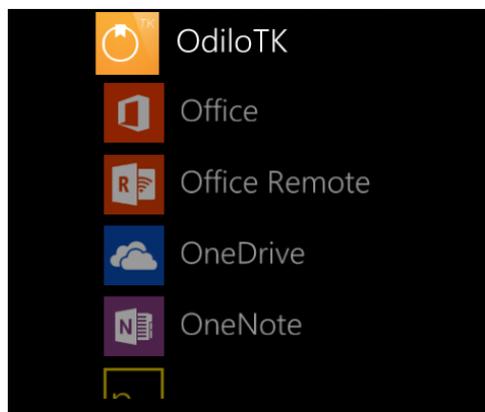
## ***Desinstalación (ejemplo para Nokia)***

Para realizar la desinstalación de la aplicación, debe de dirigirse a la lista de aplicaciones de su teléfono (deslizamiento hacia la derecha en el escritorio de tiles) y mantener pulsado sobre la aplicación “OdiloTK” de la lista, lo que desplegará el menú que podemos ver a continuación, debe seleccionar “desinstalar” y se le pedirá confirmación.

Una vez realizado esto, se procederá a la desinstalación de forma automática. Esto conllevará la liberación de los recursos usados por la aplicación.



- anclar a inicio
- calificar y opinar
- compartir
- desinstalar



Desinstalar aplicación

## ***Requisitos mínimos***

El requisito principal para poder instalar la aplicación es disponer de un dispositivo con sistema operativo Windows 8.1 o superior. Para consultar la versión de su dispositivo Windows Phone diríjase a la lista de aplicaciones y seleccione “configuración”, después seleccione “información” y podrá ver el software instalado en su teléfono.



Nota: si su versión de Windows es inferior a la 8.1, la aplicación “OdiloTK” no le aparecerá en la tienda.

Por otro lado, en este enlace puede consultar la lista de dispositivos del mercado que actualmente soportan Windows Phone:

<http://www.windowsphone.com/es-ES/phones>

Además la parte de ordenadores y tabletas, también es necesario disponer de Windows 8.1, tanto para procesadores Intel como para ARM. Esto no supone ningún problema ya que todos estos dispositivos soportan la actualización gratuita a Windows 8.1 o superior.

## ***Funcionamiento de la aplicación.***

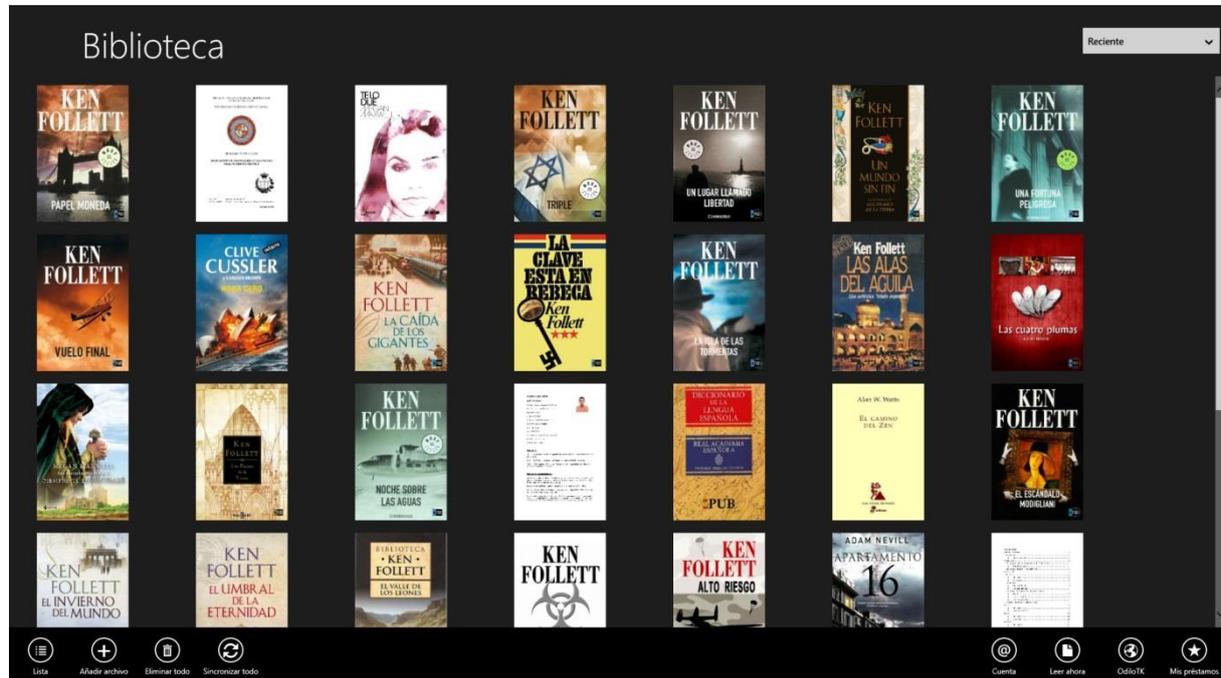
1) Lo primero de todo para que la aplicación comience a funcionar es registrarse correctamente con la cuenta de su biblioteca (posteriormente puede modificarla si desea).



Activación correcta

Nota: A esta página puede volver, haciendo clic sobre el botón “Cuenta” de la página “Biblioteca” y por ejemplo desactivar su cuenta mediante el botón “Desactivar” de la barra de comandos.

2) Aparecerá en la página Biblioteca, si ya tiene libros prestados en su cuenta, puede comenzar por pulsar el botón sincronizar todo, que le descarga sus libros y le abrirá el último de ellos.



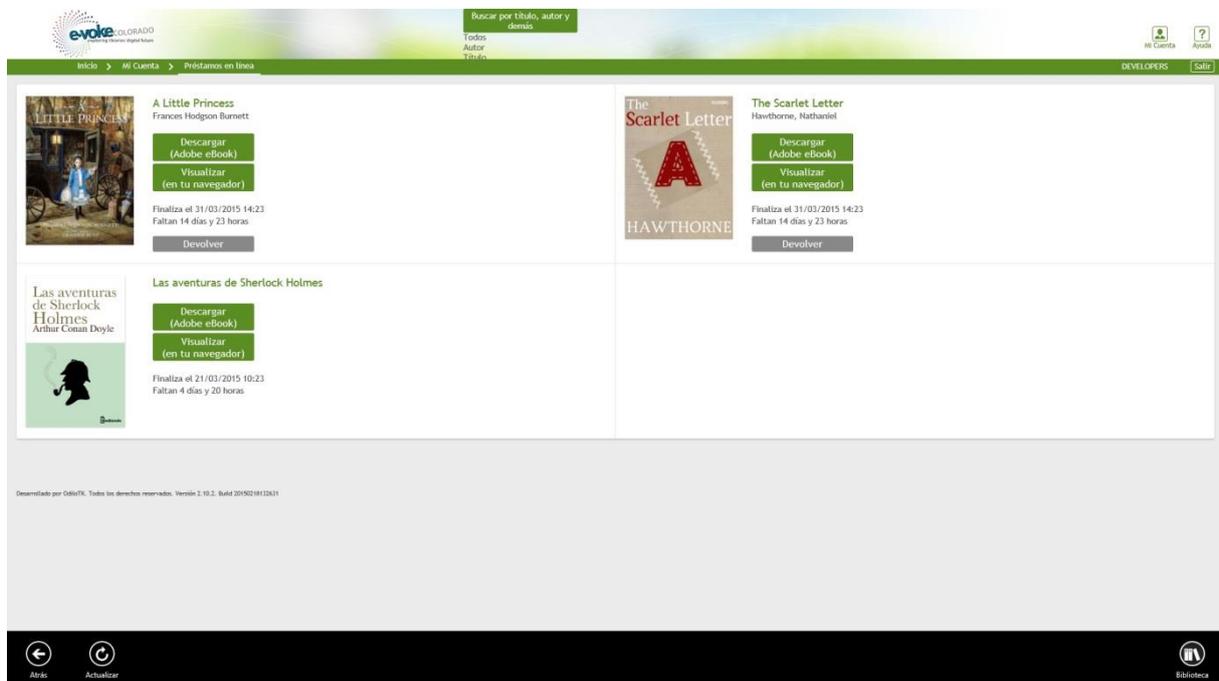
3) Si desea realizar devolver algún libro de los que dispone, solamente necesitará seleccionarlo y pulsar sobre la opción “devolver libro”.

4) Si por el contrario, lo que desea es pedir prestados nuevos libros, puede hacer clic sobre el botón “OdiloTK” para dirigirse al catálogo de libros, donde podrá realizar búsquedas y obtener información detallada de cada uno de los diferentes libros que están disponibles



Catálogo de libros

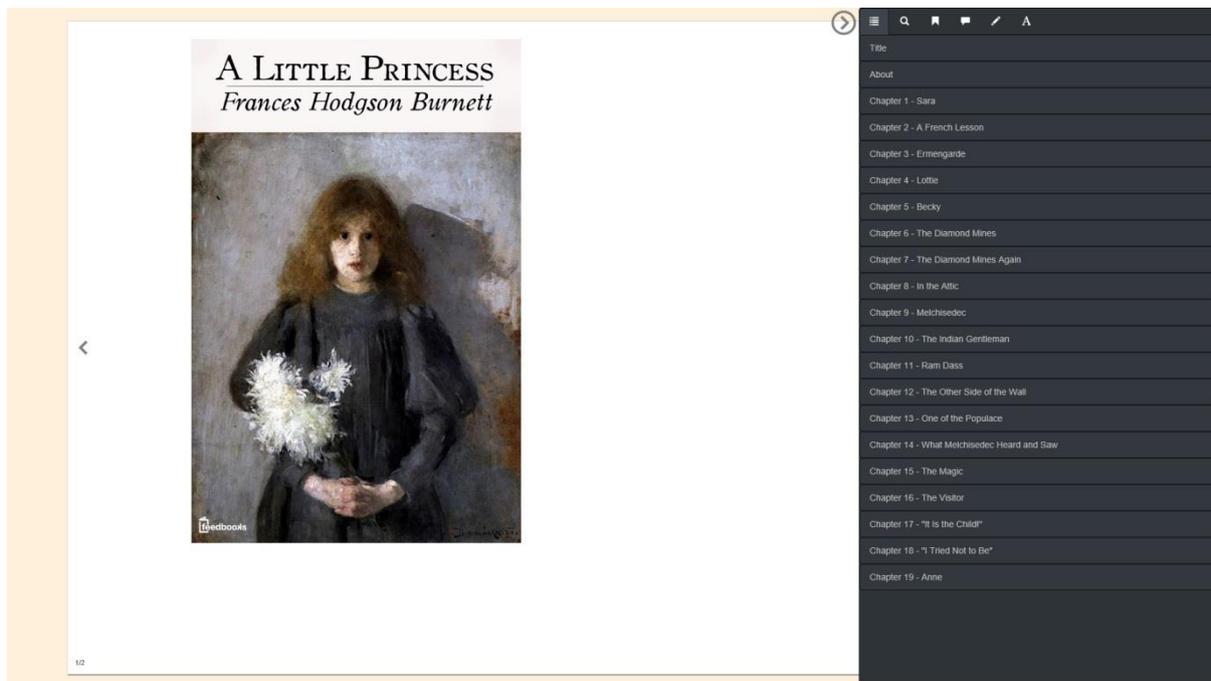
5) Una vez solicitados los préstamos, éstos aparecen en su página personal a donde puede dirigirse haciendo clic en el botón “Mis préstamos” de la página “Biblioteca”, o desde la navegación: Mi cuenta → Mis préstamos, por ejemplo desde el catálogo.



Página personal “mis préstamos”



6) Si desea leer alguno de los libros prestados tiene dos opciones, la primera de ellas es hacer clic sobre el botón “Visualizar (en tu navegador)” de la página “mis préstamos”. De esta forma, se procederá a la apertura de dicho libro mediante el lector “Nubereader”, como podemos ver a continuación:

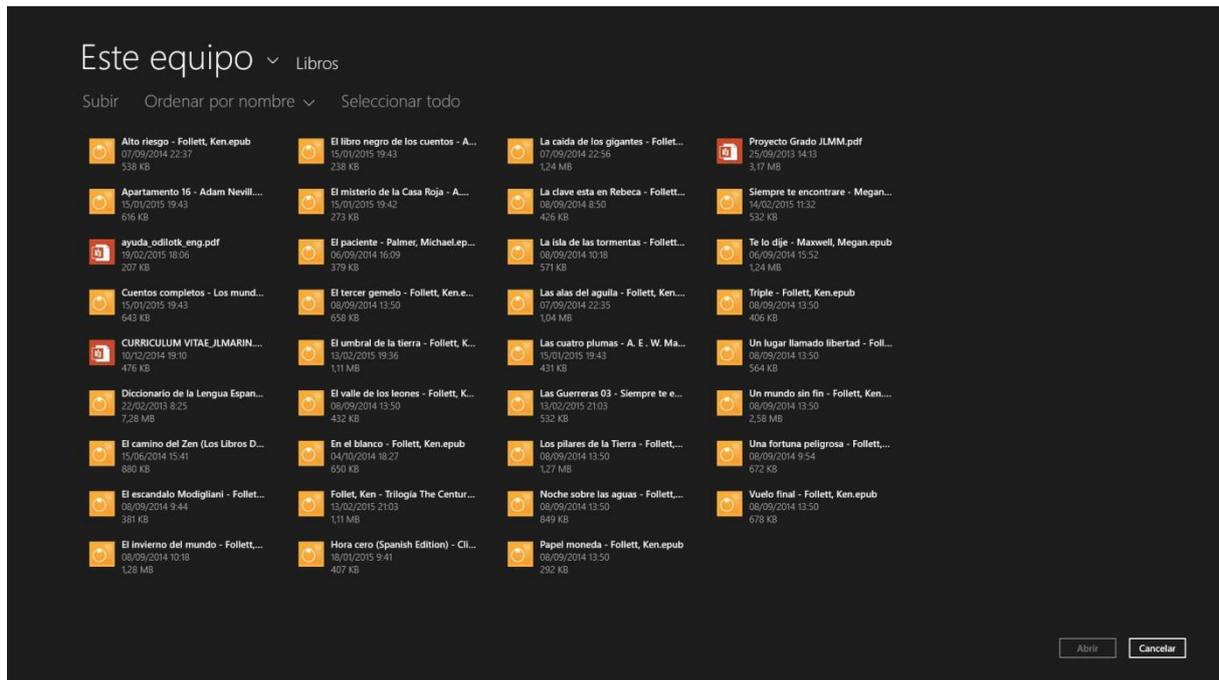


Lector Nubereader

7) La segunda opción de la que dispone es mediante el botón “Descargar (Adobe eBook)” en la página de “Mis préstamos”. Esto provocará la descarga del libro y su inclusión en la biblioteca de la aplicación, además de abrir el libro mediante el lector interno “LectorDRM”.

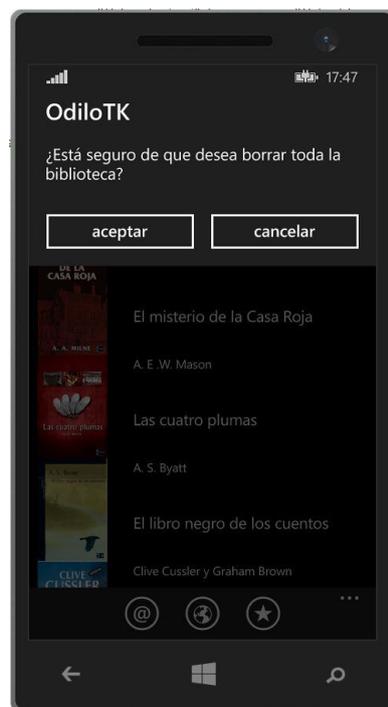
*Nota:* También puede realizarlo de forma equivalente si desde la página “Biblioteca” pulsa el botón “Sincronizar todo”.

8) También puede añadir libros, tanto protegidos por DRM (fichero .acsm), como públicos en formatos ePub y PDF. Para ello, desde la página “Biblioteca” haga clic en el botón “añadir archivo”, lo cual abrirá el selector de archivos del sistema Windows y le permitirá añadir tantos libros como desee, como podemos ver aquí:



Selector de archivos Windows – interfaz Metro

9) Por último en un momento dado, puede eliminar todos los libros de la biblioteca, si así lo desea. Para ello, solamente tiene que dirigirse a la página “Biblioteca” y sin seleccionar ningún libro, pulsar sobre el botón “Eliminar todo”. Esta acción le pedirá confirmación como vemos a continuación:



Confirmación – eliminar todo



*Nota:* Esta acción no realizará la devolución de los libros prestados, solamente los eliminará de la aplicación (localmente), pero le seguirán apareciendo en la página “Mis préstamos” donde los puede volver a descargar o leer.

*Nota:* Si lo que desea es devolver uno o varios libros, selecciónelos y pulse sobre el botón “Devolver libros” en la página “Biblioteca”. Si por el contrario pulsa el botón “Borrar selección”, solamente los eliminará de la aplicación

## ***Problemas y soluciones.***

### **1) No encuentro la aplicación en la tienda:**

Esto puede deberse a que no ha escrito correctamente el nombre de la aplicación “OdiloTK” (con escribir Odilo basta). Debe asegurarse de que dispone de conexión a internet mediante Wifi o datos móviles y que la versión de su sistema operativo es Windows 8.1 o superior.

### **2) No puedo activar la aplicación:**

Debe asegurarse que los datos introducidos de su cuenta, es decir sus credenciales, son correctos y que además ha seleccionado su biblioteca Odilo.

Si dispone de varias cuentas en una o más bibliotecas Odilo, asegúrese de que los datos introducidos corresponden a la biblioteca seleccionada.

Tenga en cuenta que los datos con los que iniciaría sesión en su biblioteca mediante el navegador web (ejemplo internet Explorer) son los mismos que debe introducir para activar la aplicación.

### **3) No puedo acceder al catálogo OdiloTK, a mis préstamos o al lector Nubereader:**

Esto puede ser debido a que no dispone de conexión a Internet, de forma que la aplicación le permite acceder a su biblioteca y leer (lectorDRM) o añadir libros desde local, pero no le permite realizar ninguna tarea que requiera acceso a internet. Simplemente active su conexión de datos móviles, Wifi o cable y automáticamente se le permitirá realizar estas acciones (NO es necesario reiniciar la aplicación).

### **4) He borrado la aplicación sin querer de mi dispositivo y no sé cómo recuperar los libros.**



No se preocupe, tan solo tiene que volver a instalar la aplicación. Automáticamente le aparecerán los datos de su cuenta y su biblioteca (la última cuenta que usó) y solamente tiene que hacer clic en el botón “Activar”.

Posteriormente solamente tiene que pulsar sobre el botón “sincronizar todo” de la página “Biblioteca” y ya dispondrá de sus libros prestados en la aplicación.

**5) No entiendo cómo se puede leer un libro directamente, sin tener que buscarlo en la aplicación.**

Para realizar esta acción, dispone de varias formas que se le facilitan:

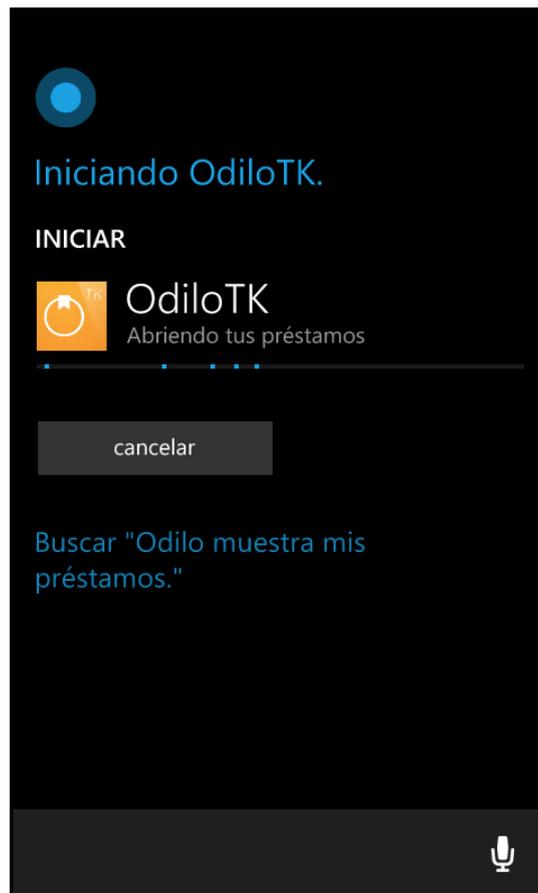
- Puede entrar a la aplicación y pulsar sobre el botón “Leer ahora” en la página “Biblioteca” para abrir el último libro leído mediante el lector “DRM” o en la página “OdiloTK” para abrir el último libro leído mediante el lector “Nubereader”.
- Puede acceder directamente a cualquier libro mediante el lector “DRM” si lo ancla a inicio mediante el botón “Anclar a inicio” de la página “Biblioteca”, lo que creará un tile secundario como puede verse a continuación:



Tile aplicación y tile secundario en inicio



- También puede utilizar Cortana, en caso de ejecutar la aplicación en un teléfono con Windows 8.1 Update 1 o superior. Si no dispone de Cortana en su teléfono, debe actualizarlo a la versión del sistema más reciente. Mantenga presionado el botón de búsqueda de su teléfono y diga por ejemplo: *Odilo seguir leyendo*. Se mostrará una imagen similar a la que podemos ver y se accederá directamente al último libro abierto por el lector DRM.



Cortana abriendo la página de "mis préstamos"



## **Glosario de términos utilizados.**

GPS – Global Position System

SMS – Short Message Service

WWW – World Wide Web

PDA – Personal Digital Assistant

API - Application Programming Interface

GUI - Graphical User Interface

HTTP - HyperText Transfer Protocol

IDE - Integrated Development Environment

GSM - Groupe Special Mobile, sistema global para las comunicaciones

SQL - Structured Query Language

PC – Personal Computer

XAML - eXtensible Application Markup Language

CLR - Common Language Runtime

WP – Windows Phone

BBDD – Base de datos

MPNS – Microsoft Phone Notification Service

CAD - Computer-Aided Design

SDK - Software Development Kit

LINQ - Language Integrated Query

MTP – Media Transfer Protocol

HTML - HyperText Markup Language



## Referencias

- [1] <http://www.microsoftvirtualacademy.com/>
- [2] <https://dev.windows.com/es-es>
- [3] <http://www.nokia.es/>
- [4] Josué Yeray Julián, Rafael Serna, Ibón Landa . Desarrollo en Windows 8 y Windows Phone 8 con XAML y C#, Editorial Krasis Press
- [5] [https://msdn.microsoft.com/es-es/library/windows/apps/xaml/Dn440594\(v=win.10\).aspx](https://msdn.microsoft.com/es-es/library/windows/apps/xaml/Dn440594(v=win.10).aspx)
- [6] <https://msdn.microsoft.com/library/windows/apps/dn609832.aspx/>
- [7] <http://channel9.msdn.com/series/C-Fundamentals-for-Absolute-Beginners/20>
- [8] <http://code.msdn.microsoft.com/wpapps/site/search>
- [9] <http://channel9.msdn.com/Series/Building-Apps-for-Windows-Phone-8-1>
- [10] <http://channel9.msdn.com/Series/Windows-Store-apps-for-Absolute-Beginners-with-C->
- [11] <https://dev.windows.com/es-es/develop/multilingual-app-toolkit/>
- [12] <https://javiersuarezruiz.wordpress.com/windows-8/>
- [13] [https://msdn.microsoft.com/es-es/library/windows/apps/xaml/Hh758325\(v=win.10\).aspx](https://msdn.microsoft.com/es-es/library/windows/apps/xaml/Hh758325(v=win.10).aspx)
- [14] <http://stackoverflow.com/questions/936028/linq-case-statement>
- [15] <https://msdn.microsoft.com/library/windows/apps/xaml/dn630430.aspx/>
- [16] <http://www.windowsphonebook.com/>
- [17] <https://msdn.microsoft.com/library/windows/apps/windows.ui.xaml.controls.webview.aspx/>
- [18] <https://msdn.microsoft.com/es-es/library/windows/apps/xaml/hh700362.aspx>
- [19] <http://stackoverflow.com/questions/12560412/winrt-apps-and-regional-settings-the-correct-way-to-format-dates-and-numbers-ba>
- [20] <https://msdn.microsoft.com/library/windows/apps/dn832613.aspx/>