**DESARROLLO DE UNA APLICACION PARA**

**UNA INTERFAZ MULTITACTIL EN**

**DISPOSITIVOS MOVILES**

**AGENDAVIAJE.**

**Romaní Flores Rogelio**

***Universidad Nacional José María Arguedas***

**RESUMEN**

*El artículo presenta las herramientas y técnicas de desarrollo de una aplicación móvil, las definiciones del software para dispositivos móviles y una propuesta de método de trabajo para el desarrollo de aplicaciones para móviles siguiendo el patrón arquitectónico MVVM.*

*El objetivo del proyecto será enseñar a crear una aplicación para quien desee adentrarse en la programación de aplicación en Windows Phone. Para ello, la aplicación que desarrollaremos a modo de ejemplo será una agenda de viajes. A la par que se desarrolla la aplicación daremos a conocer el sistema operativo. Al ser todavía un sistema operativo minoritario pretendemos detallar en este proyecto cómo es el entorno de programación, y a la par que vamos creando nuestra aplicación, explicar las distintas funciones que tenemos disponibles para su programación.*

*La aplicación se desarrolló con metodología orientada a objetos y muestra interfaces familiares basada en XAML, también siguiendo el patrón modelo, vista, vistamodelo (MVVM).*

**Palabras Clave**

aplicaciones para móviles “apps”; Windows Phone 8.1; desarrollo; publicación en el mercado de aplicaciones.

**INTRODUCCION**

El desarrollo de aplicaciones móviles hace referencia al proceso de escribir aplicaciones para smartphones. Los teléfonos ejecutan un sistema operativo como plataforma para proporcionar servicios a las aplicaciones. Como resultado, el desarrollo en dispositivos móviles es parecido al desarrollo de escritorio: pero con matices. Los teléfonos no tienen la memoria ni el rendimiento de los pc’s. Para simplificar el desarrollo, muchos desarrolladores inicialmente ejecutan, depuran y prueban sus aplicaciones en una versión de software del teléfono, también conocido como emulador. Este permite la creación de la aplicación en un entorno estándar y conocido, y puede ser implementado en un teléfono cuando está listo para su uso.

La combinación de Windows 8 y Windows Phone 8 se ha convertido en la tecnología software más avanzada existente para móviles, tablets y PC, proporcionando funcionalidades nunca vistas hasta el momento y, por primera vez en la historia, permite a los móviles compartir el mismo corazón software (“Shared Windows Core”) que un ordenador, ofreciendo finalmente a los desarrolladores el anhelado desarrollo multi-dispositivo: móvil, tablet y ordenador.

Arquitectura de Windows Phone 8:

Uno de los grandes problemas a los que Microsoft debió hacer frente en la época de Windows Mobile fue la fragmentación de hardware y software existente en el ecosistema e inconsistencia del sistema a través de distintos dispositivos. Aprendiendo de sus errores, en Windows Phone 8 nos encontraremos un modelo de hardware y software claro y conciso que ayudara a evitar la repetición de esta situación tratando de no atarse a un solo dispositivo.

Microsoft requiere que todo teléfono que desee ejecutar Windows Phone 8 disponga de unas características mínimas, para asegurar la consistencia de todos los usuarios del sistema. A partir de estas características los fabricantes de software son libres de ampliarlas en algunos casos y están obligados a cumplirlas con exactitud en otros.

Hasta el momento, para Windows Phone solo existían dos tipos de frameworks de desarrollo: Los desarrollos bajo Microsoft Silverlight o los desarrollos bajo la tecnología de XNA. En ambos se podían utilizar los mismos lenguajes de programación, C# o VB.Net. En Windows Phone 8 estos frameworks desaparecen. Para desarrollar aplicaciones usaremos WinRT, tanto en C# con en VB.Net, en lo referente a Silverlight. En cuanto a XNA utilizaremos C++ nativo y DirectX para el desarrollo de aplicaciones que hagan un uso intensivo de gráficos.

**Métodos, herramientas técnicas**

**Patrón Arquitectónico MVVM**

MVVM es un patrón de arquitectura de software. Al igual que muchos patrones, promueve la reutilización de código a través de separaciones de preocupaciones y modularidad.

Una aplicación MVVM se divide en tres partes principales:

**Modelo**

La capa modelo incluye todo el código que implementa la lógica principal de la aplicación y define los tipos requeridos para modelar el dominio de la aplicación. Esta capa es completamente independiente de las capas de vista y modelo de vista.

**Vista**

La capa vista es la presentación de datos, la vista gestiona entrada (pulsaciones de teclas, los movimientos del ratón, toque gestos, etc.), que en última instancia manipula propiedades del modelo.

La vista en MVVM contiene comportamientos, eventos y datos-bindings que finalmente requieren conocimiento de la modelo y el modelo de vista subyacente. Si bien estos eventos y comportamientos podrían asignarse a propiedades, llamadas a métodos y comandos, la vista sigue siendo responsable de su manejo de eventos propios y no se enciende está completamente sobre el modelo de vista.

**VistaModelo**

La capa de vistamodelo proporciona destinos de enlace de datos para la vista. En muchos casos, el modelo de vista expone el modelo directamente o proporciona miembros que encapsulan miembros de modelo específicos.

El modelo de vista también puede definir miembros para realzar un seguimiento de los datos que son relevantes para la interfaz de un usuario, pero no para el modelo, como el orden de visualización de una lista de elementos.

Estas clases que al instanciarse van a servir de contexto para hacer binding a nuestra vista.

**Orientada a objetos**

La programación orientada a objetos modela el mundo real, cualquier cosa del mundo puede ser modelada como un objeto. Así un objeto tiene propiedades (un estado) y un comportamiento.

Los objetos se definen utilizando clases, una clase es similar a una plantilla para construir objetos

**Visual Studio 2013**

Es la interface de desarrollo exclusiva para programadores, responsables de la lógica de negocio, acceso a datos y todo lo referente a este rol. De forma nativa ya tenemos las plantillas necesarias para crear aplicaciones en WP8, controles, controles personalizados etc. (Firtman & Natale, 2010)

**SDK**

El SDK de Windows Phone proporciona las herramientas que son necesarias para desarrollar Apps y juegos para Windows Phone 8 y Windows Phone 7.x.

El paquete completo contiene las siguientes herramientas:

* Visual Studio 2013 edition para Windows Phone.
* Blend para Visual Studio 2013.
* Windows Phone Developer Registration tool.
* Windows Phone Connect tool.
* Emulators de Windows Phone 7.1 y 8.0
* Windows Phone Application Analysis tool.
* Simulation Dashboard para Windows Phone

**StarUml**

StarUML es una herramienta para el modelamiento de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Arquitecture), que en un principio era un producto comercial y que hace cerca de un año paso de ser un proyecto comercial (anteriormente llamado plastic) a uno de licencia abierta.

**C#**

C# (leído en inglés "C Sharp" y en español "C Almohadilla") es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET.

Sus principales creadores son Scott Wiltamuth y Anders Hejlsberg, éste último también conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi. Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET.

Por esta razón, se suele decir que C# es el lenguaje nativo de .NET. (Días Concha, 2011).

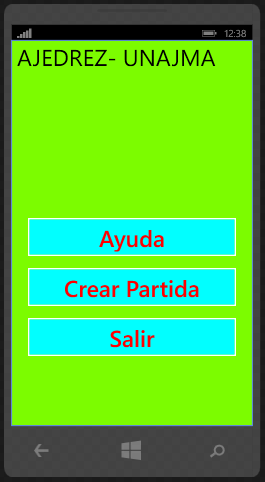
**XAML**

Significa Lenguaje Extensible de Formato para Aplicaciones (del inglés, Extensible Application Markup Languaje), cuya pronunciación es algo así como “zammel”. Este lenguaje declarativo basado en XML nos permite definir la interface de usuario de las aplicaciones, utilizando el concepto de CodeBehind, en donde podremos codificar la lógica de negocio que se une mediante clases parciales, logrando así separar las tareas para diseñadores y para programadores. (Firtman & Natale, 2010).

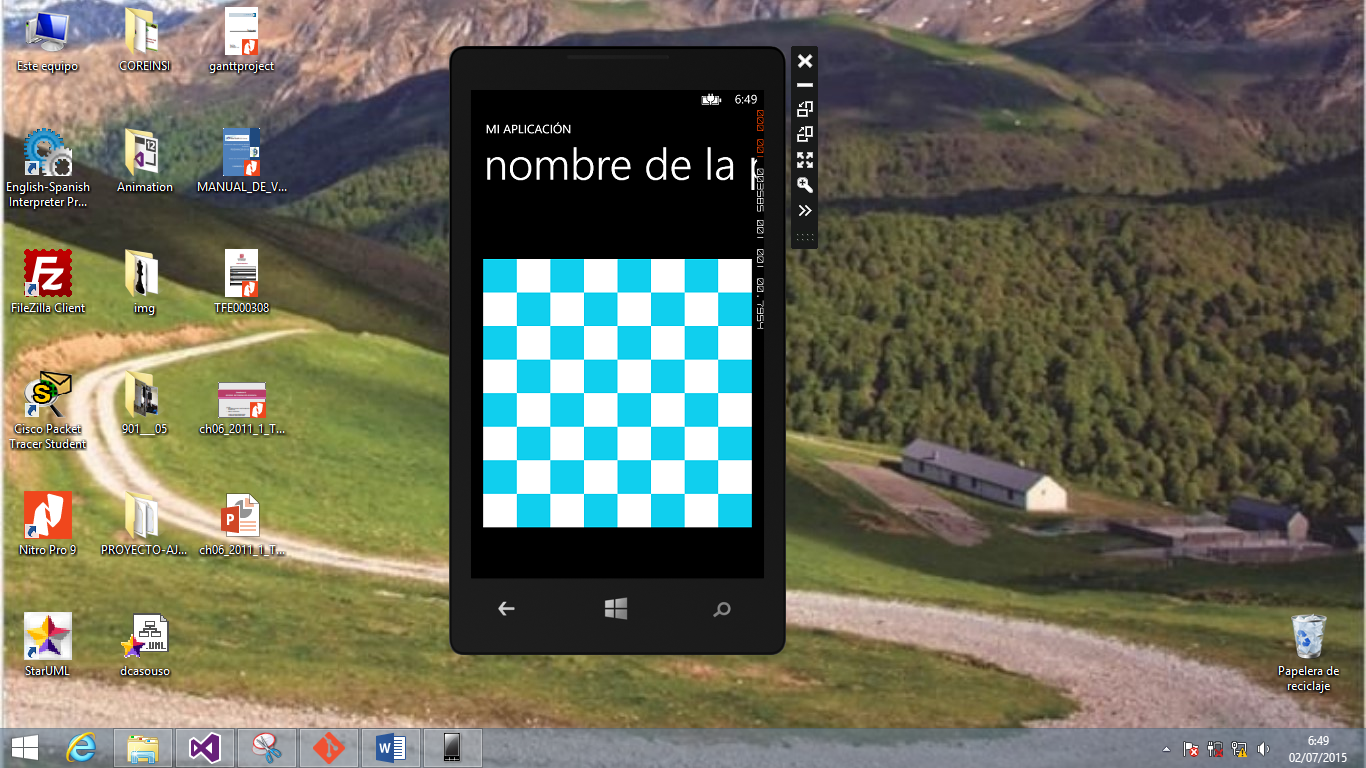
**Resultados**

En esta parte mostrare los resultados del avance. Porque no pude concluir con el objetivo trazado, pero los requerimientos y toda la estructura de la aplicación esta hecha.

Como primer resultado se obtuvo la página principal del aplicativo como se ve en la imagen.



En segundo resultado se diseña otra página que contiene al tablero del juego.





**Conclusiones**

Las aplicaciones móviles pueden ayudar a solventar los problemas de tipo particular o general de la sociedad, debido a sus características de movilidad y ubicuidad.

El hecho de poseer un teléfono Windows Phone me animó a elegir esta plataforma. Actualmente todo tipo de usuarios cuenta con teléfonos inteligentes con plataformas que soportan video juegos, apps, etc. Esto hace pensar que a los desarrolladores de aplicaciones para dispositivos móviles se les presenta un futuro bastante bueno. Cabe destacar que la comunidad de desarrolladores para Windows Mobile en concreto, que es la que yo he tratado más de cerca, es mucho más amplia ahora que cuando comencé a investigar para este proyecto. Esto indica que se encuentra en pleno crecimiento.

Al hacer balance sobre el trabajo realizado con este proyecto resulta destacable el hecho de haber adquirido conocimientos en este tipo de tecnologías, en concreto sobre el Microsoft .NET y aprender un nuevo lenguaje de programación, C#, que siempre es muy útil de cara al mercado laboral.

Al ser la primera vez que desarrollaba un proyecto de esta magnitud partiendo desde cero he podido comprobar las dificultades con las que tienen que enfrentarse los desarrolladores independientes. Esto hace valorar de otra forma cualquier aplicación que normalmente podemos considerar que no es muy buena, apreciando realmente todo el trabajo que hay detrás.

Otra de las conclusiones que he obtenido durante el desarrollo de la aplicación es que, al desarrollar una aplicación con MVVM (Modelo, Vista, VistaModelo). Me despierta el espíritu de investigar sobre algo desconocido por mí. De tal manera enriquece mi conocimiento de formación profesional.

Tal vez en cuanto al funcionamiento de la aplicación no pude concluir debido a que la naturaleza del lenguaje de programación para mí fue nueva. Con esto no quiero decir dejar de lado el desarrollo del juego seguiré investigando; pues esto es una experiencia de mi formación profesional.

**Agradecimientos**

Agradezco principalmente a mi madre y mis hermanos por ser ellos mi apoyo en mi desarrollo como profesional, a todos mis docentes que comparten sus conocimientos y a mis compañer@s, amig@s y demás personas que estuvieron presentes y aún continúan aportándome con su compañía, consejos y experiencia.

**Referencias**

* Josué Y., Rafael S., Ibón L. Desarrollo en Windows y Windows Phone con XAML y C#.
* Tom Archer (2001). A fondo C#. McGraw-Hill/INTERAMERICANA DE ESPAÑA.
* Jeff F., Brian P., Jasón B (2003). La biblia de C#, Editorial Anaya.
* Harvey M., Deitel y Paul J. (segunda edición) C# como programar.
* Nacho Cabanes. Visual C# 2010 (paso a paso).
* Shawn Wildermuth: “Windows Phone 7.5. Desarrollo de aplicaciones con Silverlight”. Ed.: Anaya
* Daniel Vaughan: ” Windows Phone 7.5 Unleashed”. Ed.: Sams
* Josué Yeray e Ibon Landa: “Windows Phone 7.5. Desarrollo de aplicaciones en Silverlight”. Ed.: Krasis Press.
* Charles Petzold: “Programming Windows Phone 7”. Ed.: Microsoft Press
* J. Ferguson, B. Petterson y J. Beres: “La biblia de C#”. Ed.: Anaya

**Contactos**

Celular: 995843378

Rogelio Romaní Flores

rromaflores@gmail.com