

**ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN**

UNIVERSIDAD DE MÁLAGA



TRABAJO FIN DE MÁSTER

*Estudio de viabilidad y diseño de una herramienta software
para monitorización de tráfico IP en Windows Phone.*

**MÁSTER EN TELEMÁTICA Y
REDES DE TELECOMUNICACIÓN**

MÁLAGA, 2011

Dayron Agüero Jiménez

**ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN**

UNIVERSIDAD DE MÁLAGA

Titulación: Máster en Telemática y Redes de Telecomunicación

Reunido el tribunal examinador en el día de la fecha, constituido por:

D./D^a. _____

D./D^a. _____

D./D^a. _____

para juzgar el Trabajo Fin de Máster titulado:

**Estudio de viabilidad y diseño de una herramienta software
para monitorización de tráfico IP en Windows Phone.**

del alumno/a D./D^a. *Dayron Agüero Jiménez*

dirigido por D./D^a. *Pedro Merino Gómez*

ACORDÓ POR _____ OTORGAR LA
CALIFICACIÓN DE _____

Y, para que conste, se extiende firmada por los componentes del tribunal, la presente diligencia

Málaga, a _____ de _____ de _____

El/La Presidente/a

El/La Vocal

El/La Secretario/a

Fdo.: _____ Fdo.: _____ Fdo.: _____

**ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN**

UNIVERSIDAD DE MÁLAGA

**Estudio de viabilidad y diseño de una herramienta software para
monitorización de tráfico IP en Windows Phone.**

REALIZADO POR:

Dayron Agüero Jiménez

DIRIGIDO POR:

Dr. Pedro Merino Gómez

DEPARTAMENTO DE: *Lenguajes y Ciencias de la Computación.*

TITULACIÓN: **Máster en Telemática y Redes de Telecomunicación**

PALABRAS CLAVE:

Internet Protocol, redes móviles, dispositivos móviles, tráfico IP, protocolos, Windows Phone.

RESUMEN:

Sin lugar a duda el Internet Protocol (IP), Ethernet y las tecnologías ópticas dominarán en el futuro el escenario de las redes. Unido a ello, la convergencia de las redes fijas a redes móviles, es uno de los campos con más futuro en el sector de las telecomunicaciones, debido al fuerte auge que han tenido los dispositivos inalámbricos en los últimos años. De allí que surja la necesidad de llevar a cabo estudios del comportamiento de los distintos protocolos basados en IP para optimizar su uso en escenarios móviles.

Este trabajo, estudia el posible diseño de una herramienta que tiene como objetivo principal, el permitir capturar todo el tráfico IP sobre las diferentes tecnologías de acceso presentes en los dispositivos móviles, que utilizan Windows Phone como sistema operativo. Con el fin de alcanzar los mejores resultados, se propone que la herramienta se ejecute en los propios terminales, para conseguir una percepción real del rendimiento ofrecido por los distintos protocolos.

Málaga, diciembre del 2011

Dedicatoria

A mis abuelos por el apoyo y la confianza que me han dado y por ser mi fuente de inspiración.

A mi tía Lydia por quererme tanto y para que sepa que yo la adoro aunque no se lo diga.

A mi mamá por preocuparse tanto por mí y por ser mi paño lágrimas.

A mi papá por todo el apoyo que me brindó.

A mi tía tata por estar siempre tan cerca, aunque está lejos.

A Melanie Sofía para que se porte bien y siga siendo nuestra alegría.

A mis hermanos: Circe, Melissa, Darianna, Raulin y Marcos por tener la dicha de contar con ellos.

A mi hermano Robin por sus locos, pero sabios consejo.

Dayron

AGRADECIMIENTOS

A mi tutor que me ayudó en todo momento.

A la Asociación Universitaria Iberoamericana de Postgrado por brindarme la oportunidad.

A mis compis de aula en la Universidad de Málaga.

A mis familiares, por quererme y apoyarme siempre.

A mi novia por estar a mi lado y apoyarme hasta en los momentos más duros.

A mis amigos y amigas, por su preocupación y por estar siempre allí cuando los necesito.

A mis suegros y cuñada por el apoyo brindado.

A todas aquellas personas que de una forma u otra formaron parte de mi desarrollo profesional.

A todos Gracias.

RESUMEN

Sin lugar a duda el Internet Protocol (IP), Ethernet y las tecnologías ópticas dominarán en el futuro el escenario de las redes. Unido a ello, la convergencia de las redes fijas a redes móviles, es uno de los campos con más futuro en el sector de las telecomunicaciones, debido al fuerte auge que han tenido los dispositivos inalámbricos en los últimos años. De allí que surja la necesidad de llevar a cabo estudios del comportamiento de los distintos protocolos basados en IP para optimizar su uso en escenarios móviles.

Este trabajo, estudia el posible diseño de una herramienta que tiene como objetivo principal, el permitir capturar todo el tráfico IP sobre las diferentes tecnologías de acceso presentes en los dispositivos móviles, que utilizan Windows Phone como sistema operativo. Con el fin de alcanzar los mejores resultados, se propone que la herramienta se ejecute en los propios terminales, para conseguir una percepción real del rendimiento ofrecido por los distintos protocolos.

ABSTRACT

Without a doubt, the Internet Protocol (IP), Ethernet and optical technologies will rule the futures tage of the networks. Together with this, the convergence of fixed and mobile networks is one of the most promising fields in the telecommunications sector due to the strong boom that wireless device shave had in recent years. Hence the need a rises to under takes studies of the behavior of the various IP-based protocols to optimize their use in mobile stages.

This work studies the design of a tool which primary objective is to capture all IP traffic on different access technologies present on mobile devices that use Windows Phone operating system. In order to achieve the best results, we propose that this tool is run on the terminals themselves, to get a real sense of performance offered by the different protocols.

CONTENIDOS

| | |
|--|------------|
| CONTENIDOS | I |
| LISTA DE FIGURAS | III |
| LISTA DE TABLAS | V |
| CAPÍTULO 1: INTRODUCCIÓN | 1 |
| CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA | 5 |
| 2.1 LA RED TELEFÓNICA..... | 6 |
| 2.2 TELEFONÍA MÓVIL..... | 7 |
| 2.3 ANTECEDENTES Y ACTUALIDAD..... | 7 |
| 2.3.1 Primera generación (1G)..... | 8 |
| 2.3.2 Segunda generación (2G)..... | 8 |
| 2.3.3 Generación (2.5 G)..... | 9 |
| 2.3.4 Tercera generación (3G)..... | 9 |
| 2.3.5 Generación (3.5 G)..... | 9 |
| 2.3.6 Cuarta generación (4G)..... | 10 |
| 2.4 ESTRUCTURA DE LA RED TELEFÓNICA CELULAR..... | 10 |
| 2.5 CARACTERÍSTICAS Y LIMITACIONES DE LOS TELÉFONOS MÓVILES..... | 11 |
| 2.6 CAPTURA Y ANÁLISIS DE TRÁFICO..... | 12 |
| 2.7 ANALIZADORES DE PROTOCOLOS..... | 13 |
| 2.7.1 Wireshark..... | 14 |
| 2.8 PUNTO DE PARTIDA..... | 15 |
| CAPÍTULO 3: WINDOWS PHONE | 19 |
| 3.1 MODELO DE HARDWARE..... | 20 |
| 3.2 MODELO DE SOFTWARE..... | 20 |
| 3.2.1 Modelo de Aplicación..... | 21 |
| 3.2.2 Modelo de UI..... | 21 |
| 3.2.3 Integración con la nube..... | 22 |
| 3.2.4 Application Runtime..... | 22 |
| 3.3 RECURSOS PARA DESARROLLADORES..... | 23 |
| 3.4 FRAMEWORKS PARA DESARROLLAR EN WINDOWS PHONE 7..... | 24 |
| 3.4.1 Silverlight..... | 24 |
| 3.5 TERMINOLOGÍA..... | 25 |
| 3.5.1 Protocolos..... | 26 |
| 3.5.2 Tecnologías y Herramientas..... | 27 |
| 3.5.3 Servicios..... | 28 |
| 3.5.4 Patrones..... | 29 |
| 3.5.5 Diseños y controles..... | 29 |
| 3.6 PROPUESTA DE MARCO DE TRABAJO..... | 31 |
| CAPÍTULO 4: PROPUESTA DE SOLUCIÓN | 33 |
| 4.1 CONSIDERACIONES BÁSICAS..... | 33 |
| 4.2 PROPÓSITOS DE LA HERRAMIENTA..... | 34 |
| 4.3 DISEÑO DE LA HERRAMIENTA..... | 36 |
| 4.3.1 Requisitos de la herramienta..... | 36 |
| 4.3.1.1 Requerimientos de facilidad de uso..... | 36 |
| 4.3.1.2 Requerimientos no funcionales..... | 37 |
| 4.3.1.3 Requerimientos de desarrollo del proceso..... | 38 |
| 4.3.2 Casos de Uso del Sistema..... | 38 |
| 4.3.3 Patrón de diseño..... | 39 |
| 4.3.4 Interfaz de usuario..... | 40 |
| 4.4 ANALIZANDO LA VIABILIDAD..... | 43 |
| 4.4.1 Obtención de información de la interfaz de red..... | 43 |
| 4.4.2 Obteniendo el tráfico IP..... | 44 |
| 4.4.3 Servicios de geolocalización..... | 45 |

| | |
|---|-----------|
| 4.4.4 Ping | 46 |
| CAPÍTULO 5: CONCLUSIONES..... | 47 |
| BIBLIOGRAFÍA..... | 49 |
| APÉNDICE A: DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA (CUS)..... | 51 |

LISTA DE FIGURAS

| | |
|--|-----------|
| Figura 2.1: Ejemplo de Red Celular..... | 11 |
| Figura 2.2: Información proporcionada por TestelDroid dividida en tres niveles. | 17 |
| Figura 3.1: Modelo de software de Windows Phone 7. | 21 |
| Figura 3.2: Esquema del Application Runtime de Windows Phone. | 23 |
| Figura 3.3: Soft input panel. | 27 |
| Figura 3.4: Model-View-ViewModel. | 29 |
| Figura 3.5: Pivot control..... | 30 |
| Figura 3.6: Panorama de control..... | 31 |
| Figura 4.1: Captura de todo el tráfico IP. | 35 |
| Figura 4.2: Exportación y análisis del tráfico capturado..... | 36 |
| Figura 4.3: Estructura de clases por capas de la herramienta aplicando el MVVM. | 39 |
| Figura 4.4: Pantalla de inicio..... | 41 |
| Figura 4.5: Interfaz para hacer PING. | 41 |
| Figura 4.6: Interfaz para Geolocalización. | 42 |
| Figura 4.7: Interfaz de información. de red..... | 42 |
| <i>Figura 4.8: Interfaz para captura de tráfico IP.</i> | <i>42</i> |

LISTA DE TABLAS

| | |
|--|----|
| Tabla 2.1: Resumen de las características proporcionadas por la herramienta TestelDroid..... | 16 |
| Tabla 3.1: Plataformas y navegadores que soportan Silverlight 4. | 25 |
| Tabla 4.1: Requisitos de facilidad de uso. | 36 |
| Tabla 4.2: Requisitos no funcionales..... | 37 |
| Tabla 4.3: Requisitos de desarrollo del proceso. | 38 |
| Tabla A.1: Descripción del CUS Mostrar ayuda. | 51 |
| Tabla A.2: Descripción del CUS obtener información de la interfaz de red..... | 51 |
| Tabla A.3: Descripción del CUS hacer ping. | 52 |
| Tabla A.4: Descripción del CUS obtener información de localización. | 52 |
| Tabla A.5: Descripción del CUS capturar tráfico. | 52 |

ACRÓNIMOS

| <i>Acrónimo</i> | <i>Expansión del acrónimo</i> |
|-----------------|--|
| 3GPP | <i>3rd Generation Partnership Project.</i> |
| AJAX | <i>Asynchronous JavaScript y XML.</i> |
| AMPS | <i>Advanced Mobile Phone System.</i> |
| API | <i>Application Programming Interface.</i> |
| CDMA | <i>Code Division Multiple Access.</i> |
| CID | <i>Cell Identification.</i> |
| CPU | <i>Central Processing Unit.</i> |
| DL | <i>Downlink (enlace descendente).</i> |
| EDGE | <i>Enhanced Data Rates for Global Evolution.</i> |
| EMS | <i>Enhanced Messaging Service.</i> |
| FDMA | <i>Frequency Division Multiple Access.</i> |
| GPRS | <i>General Packet Radio System.</i> |
| GPS | <i>Global Positioning System.</i> |
| GSM | <i>Global System for Mobile Communications.</i> |
| HSCSD | <i>High Speed Circuit Switched.</i> |
| HTTP | <i>Hypertext Transfer Protocol.</i> |
| ICMP | <i>Internet Control Message Protocol.</i> |
| IDE | <i>Integrated Development Environment.</i> |
| IDS | <i>Intrusion Detection System.</i> |
| IP | <i>Internet Protocol.</i> |
| LAC | <i>Location Area Code.</i> |
| LTE | <i>Long Term Evolution.</i> |
| M-WiMax | <i>Mobile Worldwide Interoperability for Microwave Access.</i> |
| MIMO | <i>Multiple-input Multiple-output.</i> |
| MMS | <i>Multimedia Messaging System.</i> |
| MPNS | <i>Microsoft Push Notification System.</i> |
| MVVM | <i>Model-View-ViewModel.</i> |
| NAT | <i>Network Address Translation.</i> |
| NGN | <i>Next-Generation Networks (Redes de Nueva Generación).</i> |
| OData | <i>Open Data Protocol.</i> |
| OFDMA | <i>Orthogonal Frequency Division Multiple Access.</i> |
| PCS | <i>Personal Communication Services.</i> |
| PDA | <i>Personal Digital Assistant.</i> |
| PDC | <i>Personal Digital Communications.</i> |
| PSC | <i>Primary Scrambling Code.</i> |
| QoS | <i>Quality of Service (Calidad de servicios).</i> |
| RAT | <i>Radio Access Technology.</i> |
| REST | <i>Representational State Transfer.</i> |
| RIA | <i>Rich Internet Applications.</i> |
| RSCP | <i>Received Signal Code Power.</i> |
| RSS | <i>Really Simple Syndication.</i> |
| RSSI | <i>Radio Signal Strength Indicator.</i> |
| SC-FDMA | <i>Single Carrier Frequency Division Multiple Access.</i> |
| SMS | <i>Short Message Service.</i> |

| | |
|------|--|
| SO | <i>Operating System (sistema operativo).</i> |
| SOAP | <i>Simple Object Access Protocol.</i> |
| SSL | <i>Secure Sockets Layer.</i> |
| TDD | <i>Test-Driven Development.</i> |
| TLS | <i>Transport Layer Security.</i> |
| UL | <i>Uplink (enlace ascendente).</i> |
| UMTS | <i>Universal Mobile Telecommunications System.</i> |
| WP7 | <i>Windows Phone 7.</i> |
| WPF | <i>Windows Presentation Foundation.</i> |
| XAML | <i>Extensible Application Markup Language.</i> |
| XML | <i>Extensible Markup Language.</i> |

CAPÍTULO 1: INTRODUCCIÓN.

La introducción de tecnologías inalámbricas podría revolucionar el uso de las tecnologías de información. En efecto, varios expertos en el tema (1), opinan que esta revolución puede llegar a alcanzar una significación equivalente a la adopción de Internet por el gran público. Parte de este gran público ha podido experimentar las ventajas y enormes posibilidades que brindan estas nuevas tecnologías, esto, unido a sus atractivos y populares precios ha permitido la introducción paulatina de las redes inalámbricas en el mercado.

En los últimos años, las tecnologías inalámbricas se han desarrollado grandemente y han ganado en importancia. Ventajas como la del costo de instalación de los recursos, acceso físico a las instalaciones, menor dificultad de expansión, flexibilidad y movilidad hace que en la actualidad las tecnologías inalámbricas sean muy utilizadas en diferentes ámbitos y para dar cumplimiento a diferentes objetivos.

Dentro de este cúmulo de tecnologías inalámbricas sin dudas la telefonía móvil ha alcanzado un desarrollo superior, o al menos el impacto social causado ha sido de una magnitud superior. Las comunicaciones móviles contribuyen a la eficiencia de las compañías, tanto en logística y marketing, como en las comunicaciones internas, más allá de esto, el teléfono móvil ha probado ser un instrumento valioso para la pequeña empresa y sus dueños.

De todas las ventajas que ofrece la telefonía móvil, la disponibilidad de servicios y la movilidad sobresalen por encima de las demás. Las grandes empresas proveedoras de servicios compiten por lograr garantizar la disponibilidad de un conjunto de servicios de buena calidad, accesibles a todos los usuarios, a un precio asequible y sin falsear la competencia (2). Unido a ello la movilidad comienza a ser un factor clave en la vida diaria de la sociedad.

Ahora bien, a la hora de elegir un móvil, un factor realmente importante es el sistema operativo (SO), y si se piensa en SO a la hora de elegir un ordenador. ¿Por qué no se hace así cuando compramos un móvil, PDA o dispositivo móvil en general? Existen multitud de opciones, encontrándose dentro de las más extendidas Symbian, BlackBerry OS, Android, iPhone OS y Windows Mobile.

El Sistema Operativo Windows Mobile de la familia Windows, desarrollado por Microsoft, a pesar de llevar el nombre Windows, no es un sistema derivado ni una versión recortada del mismo, sino un nuevo sistema diseñado específicamente para

dispositivos móviles. Esto no quiere decir que no presente un ambiente similar a los Sistemas Operativos desarrollados para ordenadores por Microsoft, permitiendo ello una mayor familiarización del usuario con el dispositivo en cuestión. Windows Phone, anteriormente llamado Windows Mobile es un sistema operativo móvil compacto desarrollado por Microsoft, diseñado para su uso en teléfonos inteligentes comúnmente conocidos como Smartphones.

Teniendo en cuenta las versiones anteriores que Microsoft proporciona para desarrollar las soluciones orientadas a dispositivos móviles, vale la pena destacar que Windows Phone 7 es una revolución pues está basado en un nuevo enfoque para el mercado de la telefonía móvil. Microsoft Windows Phone 7 no sólo representa una ruptura con el pasado de Windows Mobile sino que también se diferencia de otros teléfonos inteligentes que existen actualmente en el mercado. Dentro del conjunto de novedades presentadas, es la interface de usuario, denominada “Metro”, basada en mosaicos dinámicos, la que tuvo la mayor acogida, pues muestra de forma sencilla y efectiva información útil al usuario. Además propone un nuevo concepto llamado HUB (concentrador), un lugar donde centralizar acciones y agrupar aplicaciones por la actividad a la que se destinan.

A pocos meses del lanzamiento de Windows Phone 7, surge entonces su primera actualización en Marzo de 2011 llamada Windows Phone 7,5 “Mango”, que incluye mejoras en el rendimiento general así como la inclusión de copiar y pegar, el motor de Internet Explorer 9, con su soporte para HTML5, multitarea en aplicaciones de terceros y otras ventajas.

Pese a todos los avances obtenidos en materia de software y hardware en la telefonía móvil, no se está exento hoy en día de problemas de seguridad, producto del medio hostil por el que viaja la información. Un medio cambiante, donde las interferencias creadas por la naturaleza o el propio hombre influyen negativamente en la calidad de la comunicación.

Según un informe de Georgia Tech (Instituto Tecnológico de Georgia). *“Los diversos intentos de fraude y estafa que actualmente sufrimos a través de las redes de ordenadores, entre otros tipos de ataques, se trasladarán a las redes de telefonía móvil y alcanzarán una especial virulencia”*

Lo expresado anteriormente corrobora la idea del especialista César Gutiérrez cuando opina: *“La voluminosa cantidad de información manejada por medio de los teléfonos móviles, especialmente los datos financieros y otros datos importantes, sumado a las dificultades para implementar en estos dispositivos software anti-virus, los hace especialmente vulnerables y los convierte en inminente objetivo de la ciberdelincuencia”* (3).

Con la finalidad de brindar a los investigadores herramientas que le ayuden a obtener elementos que le permitan crear nuevas soluciones tanto para posibles problemas de seguridad como para hacer un uso óptimo de los recursos de cada dispositivo y las redes, surgen herramientas de capturas y análisis de tráfico. Debe tenerse en cuenta que el análisis de tráfico puede determinar el tipo de información que circula y el impacto que puede llegar a tener sobre la red y los dispositivos.

Paralelamente al desarrollo de nuevos servicios orientados específicamente al entorno móvil, es evidente la tendencia de migrar a los terminales móviles las aplicaciones que tradicionalmente han formado parte de los ordenadores de escritorio, motivado principalmente por la necesidad que tienen los usuarios de estar comunicados en todo

momento. Sin embargo estas aplicaciones han sido diseñadas y probadas para funcionar en un entorno que difiere mucho del presente en los sistemas de comunicaciones móviles. Es por ello que no siempre pueden ser utilizadas herramientas de captura y análisis de tráfico empleadas en el desarrollo y depuración de aplicaciones de comunicación en redes fijas como Wireshark, IP sniffer y Nmap, las cuales se encuentran dentro de las utilizadas en la actualidad.

El objetivo general del presente trabajo es el estudio de viabilidad para diseñar una herramienta de captura de tráfico para dispositivos móviles que utilicen Windows Phone 7 como sistema operativo. Dicha herramienta permitirá hacer capturas de tramas IP entrantes y salientes desde y hacia el dispositivo, para posteriormente analizar esta captura en analizadores de protocolos.

Concretándose los objetivos específicos en:

- Realizar un estudio preliminar de la tecnología Windows Phone.
- Realizar un estudio de herramientas con propósitos similares.
- Proponer una interfaz de usuario amigable con un alto grado de usabilidad.
- Diseñar una aplicación que permita la captura de tráfico.

El trabajo está compuesto por cinco capítulos. Tras el primer capítulo de introducción general, el segundo contiene la fundamentación teórica del trabajo donde se pueden encontrar algunos conceptos, una reseña del surgimiento y evolución de la telefonía móvil, así como un pequeño análisis del estado del arte. En el capítulo tres se hace un estudio profundo de las potencialidades de Windows Phone, de donde surgirá la propuesta de diseño que será el contenido del capítulo cuatro. Finalmente en el quinto capítulo se dan las conclusiones del trabajo y se dejan claras posibles líneas de investigación para futuras investigaciones.

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA.

Desde hace ya varios años se ha venido manejando la idea de lograr una red integral que logre unir en una sola todos los servicios. Sin embargo y pese a varias propuestas, hasta hoy no se ha conseguido desarrollar una red global en la cual la información proveniente de una conversación telefónica viaje indistintamente junto con la información de datos o video. Una de las principales razones por la que no ha tenido éxito esta idea es porque el trato que debe darse a la información procedente de una conversación telefónica es muy diferente a la que se debe dar a una transmisión de datos. Es decir, que mientras las comunicaciones de datos en la red emplean protocolos diseñados para la conmutación de paquetes, las comunicaciones de voz emplean protocolos de conmutación de circuitos.

Las redes y sistemas de nueva generación (NGN) pudieran ser la solución, ya que permiten suministrar servicios innovadores, mejorar la atención al cliente y adaptarse más rápidamente a las nuevas Tecnologías de la Información y Comunicación (TIC). Estas redes tendrán que solucionar temas como:

- La movilidad de las redes inalámbricas.
- La fiabilidad de la red pública.
- La seguridad de las líneas privadas.
- La capacidad de las redes ópticas.
- La flexibilidad de IP y de MPLS como mecanismo de transporte de datos estándar para la integración de servicios de datos, voz y video.

Existen numerosas definiciones de NGN, sin embargo, por su validez internacional, se considera la definición dada por el Grupo de Estudio 13 del Sector de Normalización de la Unión Internacional de Telecomunicaciones (UIT –T), que define una NGN como:

“Red basada en paquetes que permite prestar servicios de telecomunicación y en la que se pueden utilizar múltiples tecnologías de transporte de banda ancha propiciadas por la QoS (Quality of Service), y en la que las funciones relacionadas con los servicios son independientes de las tecnologías subyacentes relacionadas con el transporte. Permite a los usuarios el acceso sin trabas a redes y a proveedores de servicios y/o servicios de su elección. Se soporta movilidad generalizada que permitirá la prestación coherente y ubicua de servicios a los usuarios”.

Esta definición sugiere entre otras cosas que tanto las funciones referentes a los servicios, como al transporte, se pueden ofrecer separadamente.

No cabe duda que Internet Protocol (IP), Ethernet y las tecnologías ópticas dominarán en el futuro el escenario de las Redes de Nueva Generación (NGN) durante algún tiempo. Para el acceso a banda ancha por cable, el bucle de abonado digital todavía puede contar con algunos años de vida, sin perder de vista que la fibra óptica será la tecnología dominante. Para el acceso de banda ancha móvil, tanto LTE como M-WiMax parecen prometedores. Las inversiones existentes y el conocimiento de la tecnología pueden muy bien determinar la dirección futura y la evolución. Como siempre, los operadores hacen esfuerzos considerables para reducir los gastos resultantes de la implementación de tecnologías de redes en competencia y cortesía (4).

La gestión de la seguridad en estas nuevas redes sería un elemento importante a tener en cuenta, ello implicaría la gestión de la identificación de usuarios, autenticación, autorización, certificación, etc. De lo contrario, será muy difícil obtener los servicios y mantener la continuidad del servicio cuando el usuario (o sesión) se mueve de una red de acceso a otra, o se desplace de un proveedor de servicios a otro. En este sentido sin duda alguna, la captura y análisis de tráfico tendrán un papel protagónico.

2.1 La red telefónica.

Los sistemas telefónicos han sufrido cambios significativos desde sus inicios hasta nuestros días, aun cuando los avances en cuestión de telefonía han sido enormes, desde que se comercializó el primer equipo telefónico, siempre el mecanismo empleado para el flujo de información ha sido la conmutación de circuitos. No fue hasta finales de los 90 y principios de la siguiente década cuando se concibió por primera vez una red telefónica en la que en vez de emplear conmutación de circuitos se emplea conmutación de paquetes por medio del uso de protocolos destinados a redes de datos como IP (5).

De la red telefónica se pueden destacar tres elementos fundamentales, cada uno de estos elementos tiene funciones determinadas dentro de la red telefónica, por ejemplo:

- El terminal telefónico es el encargado de proporcionar la interfaz adecuada para lograr la transmisión de información vocal entre usuarios distantes. Para realizar su misión dispone, en la parte de recepción de voz, de un dispositivo encargado de la transformación en electricidad de las ondas sonoras llamado micrófono. Del lado emisor, el encargado de realizar la función inversa es el auricular. Las tecnologías que se emplearon en la realización práctica de los terminales telefónicos fueron un reflejo de los avances en el campo de la electrónica.
- Las centrales de conmutación son las encargadas de proporcionar la selectividad necesaria en una llamada telefónica automática. Mediante ésta el usuario del servicio logra entablar una conversación con la persona que desea. Sin embargo, en el procedimiento que permite que el destinatario, al descolgar su terminal telefónico, pueda intercambiar información con el que origina la llamada es necesaria la concurrencia de dos funciones, estas son la señalización y la transmisión.

La señalización es el conjunto de informaciones elaboradas por el usuario emisor de la red telefónica de una parte y por los elementos integrantes de la propia red por otra. Estas hacen posible mediante su análisis e interpretación que la central de conmutación ponga en contacto físico al usuario emisor con el receptor. Además tiene otras funciones

no menos importantes dentro de las que se pueden citar la identificación del servicio del destino, la tarificación de la llamada y de los servicios en ella involucrados.

En las centrales de conmutación telefónica se realizan otra serie de funciones que no son esenciales en la comunicación telefónica, pero que igualmente son vitales para el correcto funcionamiento de la red. Así, por ejemplo, en una central de conmutación se realizan funciones de chequeo periódico de la red y de sus distintos elementos integrantes. Estas tareas tienen como finalidad la verificación del estado de la red y la toma de datos que permiten la elaboración de estadísticas sobre el tráfico cursado, averías ocurridas o distribuciones de servicios ofrecidos.

Las técnicas de conmutación utilizadas en las centrales telefónicas han sufrido una profunda evolución, de forma paralela al desarrollo de la tecnología electrónica. Así, desde los primeros conmutadores mecánicos se ha pasado a los actuales sistemas electrónicos de conmutación. Estos permiten además, toda una serie de operaciones extras como son el encaminamiento alternativo de las llamadas (enviar una llamada a través de una ruta u otra dependiendo de las condiciones de las líneas, del tráfico, etc.), tarificación detallada, entre otras.

2.2 Telefonía móvil.

La telefonía móvil o telefonía celular, se encuentra constituida básicamente por dos partes significativas: la red de comunicaciones o red de telefonía móvil y los terminales o dispositivos móviles que permiten el acceso a los recursos y servicios de dicha red.

El teléfono móvil es un dispositivo electrónico e inalámbrico que provee el acceso a la red de telefonía celular o móvil. Dicho teléfono es comúnmente conocido como celular en la mayoría de los países latinoamericanos ya que el servicio funciona mediante una red de celdas, donde cada antena brinda cobertura a un área determinada.

Los teléfonos celulares han revolucionado el área de las comunicaciones, redefiniendo como se perciben las comunicaciones. Tradicionalmente, estos dispositivos se mantuvieron fuera del alcance de la mayoría de los consumidores debido a los altos costos que poseían. Producto de ello, las grandes compañías proveedoras de servicios en estos campos, invirtieron tiempo y recursos para desarrollar nuevos sistemas de mayor capacidad que implicaran menores costos de comercialización. Ello ha traído aparejado que los sistemas celulares se hayan beneficiado de estas investigaciones y han comenzado a desarrollarse como productos de consumo masivo en todo el planeta.

Las comunicaciones móviles han superado las limitaciones que presentaba el teléfono fijo convencional, al no estar limitado a un mismo ámbito o lugar determinado como la vivienda o el lugar de trabajo. De todos modos, las comunicaciones móviles no son algo tan nuevo como nos puede parecer, ya que lleva mucho tiempo en funcionamiento para enlaces con barcos en altamar o aviones.

2.3 Antecedentes y Actualidad.

La comunicación inalámbrica tiene sus raíces en la invención del radio por Nikolai Tesla en la década de 1880, aunque formalmente fue presentado en 1894 por un joven italiano llamado Guglielmo Marconi.

El 1947 marcó pautas importantes en la historia de la industria de la telefonía celular. En ese año los científicos desarrollaron las ideas que permitirían el uso de teléfonos móviles utilizando "células" que identificaran un usuario en cualquier punto desde

donde se efectuara la llamada. Sin embargo, la limitada tecnología con la que se contaba en ese momento obligó a desarrollos posteriores.

En 1949 las personas que comenzaron a sentir la necesidad de comunicación móvil tenían que confiar en el uso de radio-teléfonos en sus autos. En el sistema radio-telefónico, existía sólo una antena central por cada ciudad y unos pocos canales disponibles para transmitir.

Esta antena central significaba que el teléfono en el vehículo requería una antena poderosa, lo suficientemente poderosa para transmitir a 50 ó 60 km de distancia. A esto se sumaba el hecho, que muchas personas no podían usar los radios-teléfonos al unísono, debido simplemente a que no existían suficientes canales para conectar.

Aparecieron entonces en 1964, la aplicación de los sistemas selectores de canales automáticos, estos fueron colocados en servicio para los sistemas de telefonía móvil. Ello eliminó la necesidad de la operación oprimir-para-hablar (push-to-talk), permitiendo a los clientes marcar directamente sus llamadas, sin la ayuda de una operadora.

La creciente demanda en el espectro de frecuencia de telefonía móvil saturado, impulsó a buscar un modo de proporcionar una eficiencia del espectro de frecuencia mayor en 1971. En este año, la compañía AT&T (American Telephone and Telegraph) hizo una propuesta sobre la posibilidad técnica de proporcionar respuesta a lo anterior. Se comenzaba a delinear el principio de la radio celular. En este mismo año en Finlandia se lanza la primera red pública exitosa de telefonía móvil, llamada la red ARP. Dicha red es vista como la Generación 0 (0G).

En 1973 el Dr. Martin Cooper quien es considerado el inventor del primer teléfono portátil, siendo gerente general de sistemas de Motorola realizó una llamada a sus competidores de AT&T desde su teléfono celular, transformándose en la primera persona en realizar una llamada utilizando un teléfono móvil. Es por este motivo que es reconocido como “el padre de la telefonía celular”.

Con el transcurso de los años la telefonía móvil ha ido evolucionando pasando desde la 1G a la 4G, en los siguientes epígrafes se hablará un poco de cada una de ellas.

2.3.1 Primera generación (1G).

La 1G de la telefonía móvil hizo su aparición en 1979 y se caracterizó por ser analógica y estrictamente para voz. La calidad de los enlaces era muy baja, tenían baja velocidad (2400 bauds). En cuanto a la transferencia entre celdas, era muy imprecisa ya que contaba con una baja capacidad (Basada en FDMA, Frequency Division Multiple Access) y la seguridad no existía. La tecnología predominante de esta generación es AMPS (Advanced Mobile Phone System).

2.3.2 Segunda generación (2G).

La 2G no arribó hasta 1990 y a diferencia de la primera generación, se caracterizó por ser digital. El sistema 2G utiliza protocolos de codificación más sofisticados, los cuales son empleados en los sistemas de telefonía celular actual. Las tecnologías predominantes son: GSM (Global System por Mobile Communications); IS-136 (conocido también como TIA/EIA136 o ANSI-136) y CDMA (Code Division Multiple Access) y PDC (Personal Digital Communications), este último utilizado en Japón.

Los protocolos empleados en los sistemas 2G soportan velocidades de información más altas por voz, pero limitados en comunicación de datos. Se pueden ofrecer servicios auxiliares, como datos, fax y SMS (Short Message Service). La mayoría de los protocolos de 2G ofrecen diferentes niveles de inscripción. En Estados Unidos y en otros países se le conoce a 2G como PCS (Personal Communication Services).

2.3.3 Generación (2.5 G).

La generación 2.5 surge bajo la necesidad de inclusión de nuevos servicios tales como:

- EMS: Servicio de mensajería mejorado, permitiendo la inclusión de melodías e iconos dentro del mensaje basándose en los sms; un EMS equivale a 3 o 4 sms.
- MMS (Sistema de Mensajería Multimedia): Mensajes que permiten la inserción de imágenes, sonidos, videos y texto.

Los cuales necesitarían una mayor velocidad de transferencia de datos, esto se hizo realidad con tecnologías, como: GPRS (General Packet Radio System), HSCSD (High Speed Circuit Switched), EDGE (Enhanced Data Rates for Global Evolution), entre otras.

2.3.4 Tercera generación (3G).

La 3G se caracteriza por contener a la convergencia de voz y datos con acceso inalámbrico a Internet; en otras palabras, es apta para aplicaciones multimedia y altas transmisiones de datos. Los protocolos empleados en los sistemas 3G soportan altas velocidades de información y están enfocados para aplicaciones más allá de la voz, audio (mp3), video en movimiento, videoconferencia y acceso rápido a Internet, sólo por nombrar algunos. El desarrollo tecnológico permitió un sistema totalmente nuevo: UMTS (Universal Mobile Telecommunications System).

UMTS utiliza la tecnología CDMA, lo cual le hace alcanzar velocidades realmente elevadas (de 144 Kbps hasta 7.2 Mbps, según las condiciones del terreno). UMTS ha sido un éxito total en el campo tecnológico pero no ha triunfado excesivamente en el aspecto comercial. Se esperaba que fuera un bombazo de ventas como GSM pero realmente no ha resultado ser así, porque al parecer, la mayoría de los usuarios se sienten satisfechos con la transmisión de voz y la transferencia de datos por GPRS y EDGE.

2.3.5 Generación (3.5 G).

LTE (Long Term Evolution) es estándar de la norma 3GPP (3rd Generation Partnership Project). Definido como una evolución de la norma 3GPP UMTS (3G). Lo novedoso de LTE es la interfaz radioeléctrica basada en OFDMA (Orthogonal Frequency Division Multiple Access) para el enlace descendente (DL) y SC-FDMA (Single Carrier Frequency Division Multiple Access) para el enlace ascendente (UL). La modulación elegida por el estándar 3GPP hace que las diferentes tecnologías de antenas MIMO (Multiple-input Multiple-output) tengan una mayor facilidad de implementación.

Esta tecnología inalámbrica de telecomunicaciones proporciona una mayor simplificación del acceso a la radio y la red de paquetes primaria con el objetivo de apoyar una mayor eficiencia del espectro y la reducción de la latencia de las interfaces de aire. Las generaciones 2G y 3G están basadas en técnicas de Conmutación de Circuito (CS) para la voz, mientras que LTE propone la técnica de Conmutación por

Paquetes IP (PS) para todo el tráfico. Utiliza de una sola red núcleo y transporte IP en la red de acceso a radio hasta las estaciones base NodeB. En términos de soporte de arquitectura y evolución, LTE reduce la complejidad de la red IP de manera significativa, para ello se logra disminuir la latencia extremo a extremo y el caudal puede ser mejorado al mismo tiempo, reduciendo así los costes de la red. (4)

Todo lo anteriormente expuesto da lugar a la obtención de las siguientes prestaciones:

- Tasa de pico del enlace DL de hasta 326,5 Mbit/s para 4x4 antenas y 172,8 Mbit/s para 2x2 antenas. Espectro de frecuencia 20 MHz.
- Tasa de pico del enlace UL de hasta 86,5 Mbit/s. Espectro de frecuencia 20 MHz.

2.3.6 Cuarta generación (4G).

Como con las mejoras previas de tecnología obtenidas en cada generación, LTE-Advanced busca mejoras en tres áreas:

- Tasas de datos de pico y promedio.
- Eficiencia espectral.
- Latencia en el plano de usuario y de control.

Los nuevos requerimientos de pico de tasa de datos sólo pueden alcanzarse con dos pasos. Primero, mejorando las capacidades MIMO en descendente (hasta MIMO 8x8 de único usuario) y permitiendo MIMO en el ascendente (hasta MIMO 4x4 de único usuario). Segundo, usando agregación de banda o portadora. LTE-Advanced permite la agregación de hasta cinco portadoras de hasta 20 MHz cada una para obtener un ancho de banda de transmisión de hasta 100 MHz. Para admitir compatibilidad hacia atrás cada una de estas portadoras se puede configurar como 3GPP Release 8. El objetivo en estos sistemas 4G es alcanzar tasas de transmisión más altas de hasta 1 Gbit por segundo.

Hablar de sistemas de comunicaciones móviles celulares de cuarta generación, es hablar de una serie de plataformas tecnológicas que en conjunto rompen el paradigma de la evolución lineal que se venía dando hasta los sistemas de tercera generación (3G). Los cambios tecnológicos a partir de estos sistemas se han dado de manera vertiginosa, no permitiendo ello que la tecnología 3G llegue a su etapa de maduración.

2.4 Estructura de la red telefónica celular.

La telefonía móvil celular se basa en un sistema de áreas de transmisión, a las que se conoce con el nombre de células. Estas abarcan áreas comprendidas entre 1,5 y 5 km, dentro de las cuales existen una o varias estaciones repetidoras. Dichas estaciones trabajan con un determinado rango de frecuencias, que debe ser diferente en las células circundantes.

Al dividir el territorio en células, se obtiene lo que se conoce como sistema celular. Cada célula es atendida por una estación de radio o estación base que restringe su zona de cobertura. Esto propicia que el espectro radioeléctrico pueda ser reutilizado en cada nueva célula, de esta manera, se puede aumentar el número de usuarios, al no requerirse una frecuencia exclusiva para cada uno de ellos, ver figura 2.1. También es válido señalar, que se utiliza una estructura hexagonal porque en el mallado que se forma, la relación entre el perímetro y la superficie es mínima, lo que disminuye el número de

traspasos (handovers) que se producen en los desplazamientos. En la práctica, se trata de evitar que existan zonas de sombra, debido a las irregularidades del terreno o la ciudad.

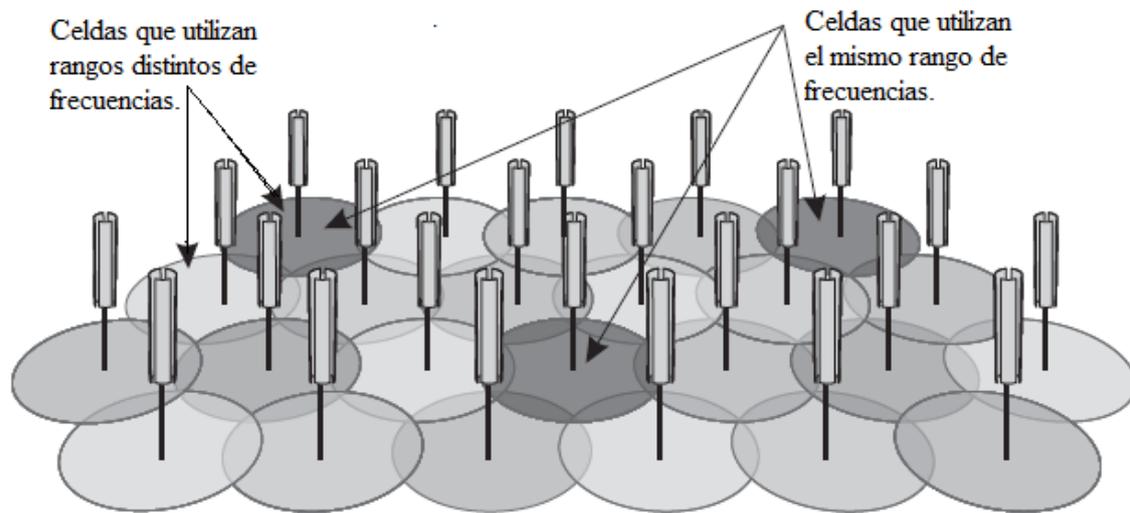


Figura 2.1: Ejemplo de Red Celular.

La estación base es uno de los elementos más importantes de una red de comunicaciones móviles, ya que se trata del equipamiento fijo distribuido geográficamente para cubrir el área a la que se pretende prestar el servicio. Otra cuestión de importancia es que la estación base es el elemento que controla la comunicación vía radio entre el terminal de usuario y la red, constituyendo por tanto el acceso de los usuarios a la misma. Una estación base se compone de:

- Antena o antenas: emisora(s) y receptora(s) de las señales de radio.
- Equipos electrónicos: sirven para establecer y mantener la comunicación.
- Equipos electrónicos exteriores (radio-enlaces) o interiores (fibra óptica o par de cobre) para la conexión de la estación base con la red de conmutación.
- Baterías: sirven para garantizar el funcionamiento del sistema en caso de cortes del fluido eléctrico.
- Sistema de refrigeración: permiten el correcto funcionamiento de la estación en épocas de calor.

Durante una comunicación, el usuario se conecta a través de la estación base más cercana que a su vez, está comunicada con el resto de las células y con los demás sistemas de telefonía (móvil o fija). Este hecho garantiza la comunicación en movimiento, es decir, cuando el usuario está en movimiento y atraviesa distintas células de cobertura.

2.5 Características y limitaciones de los teléfonos móviles.

Las propiedades físicas de todos los teléfonos móviles como plataforma para crear aplicaciones cliente poseen ciertas limitaciones, si se tienen en cuenta los tipos de aplicación que se puede construir. Por ejemplo, el dispositivo es poco probable que tenga acceso a grandes cantidades de memoria y almacenamiento, lo que impide la creación de aplicaciones que almacenen grandes volúmenes de datos. Además, la

conectividad a la red puede interrumpirse a ratos, lo cual unido a la falta de ancho de banda disponible, implica que las aplicaciones deben ser capaces de dejar de funcionar sin conexión o ser capaces de manejar las interrupciones y volver a sincronizar cuando la conexión esté disponible.

A continuación se muestran algunas de las diferencias existentes entre los teléfonos móviles y los ordenadores comunes:

- El poder de procesamiento, memoria y almacenamiento: Los dispositivos móviles suelen utilizar CPUs con un nivel inferior energía, tienen una memoria muy limitada y tienen mucho menos espacio de almacenamiento disponible.
- Tamaño de la pantalla: Los dispositivos móviles suelen tener una pantalla pequeña. En Windows Phone 7 por ejemplo, la resolución de pantalla máxima es de 800 x 480 píxeles. La pantalla también se puede girar para que coincida con la orientación del dispositivo.
- Consumo de energía: La batería es más pequeña en dispositivos móviles y esto implica que la aplicación debe ejercer el consumo mínimo posible de la misma. Técnicas tales como la minimización del uso de los servicios de localización y la reducción al mínimo de la actividad de la red, pueden colaborar en ello.
- Ejecución de aplicaciones: Los dispositivos móviles suelen permitir que sólo una aplicación se ejecute en primer plano, con el fin de minimizar el consumo de energía y maximizar el rendimiento. Aplicaciones de telefonía móvil deben estar diseñadas para mantener su estado cuando se interrumpe o termina con el sistema operativo.
- Conectividad: Los dispositivos móviles a menudo experimentan períodos de conectividad limitada o nula ya que el usuario se mueve de un lugar a otro. El ancho de banda disponible y el costo de la transferencia de datos pueden variar considerablemente, dependiendo del tipo de conexión que está disponible en cada lugar y en las distintas regiones geográficas.
- Interfaz de usuario: Las aplicaciones para dispositivos móviles suelen ser operadas por medio de una pantalla táctil. La ubicación, el espaciado y tipos de controles que son adecuados para una interfaz basada en el contacto difieren de las de una aplicación de escritorio o portátil que se basa en un ratón y un teclado.

2.6 Captura y análisis de tráfico.

En telefonía o en general en telecomunicaciones se denomina ingeniería o gestión de tráfico a diferentes funciones necesarias para planificar, diseñar, proyectar, dimensionar, desarrollar y supervisar redes de telecomunicaciones en condiciones óptimas, de acuerdo a la demanda de servicios, márgenes de beneficios de la explotación, calidad de la prestación y entorno regulatorio y comercial.

Las tecnologías de transmisión de datos y voz a través de redes de telefonía móviles son el eje central del funcionamiento de las principales empresas del sector. Un excelente desempeño de la red trae como consecuencia un aumento de la productividad informática. El ingreso de nuevos equipos a la red, la existencia de protocolos innecesarios, la mala configuración de equipos activos, el mal empleo de las interfaces o

dispositivos móviles que no respondan a las prestaciones requeridas pueden causar la decadencia del desempeño de la red. Por medio de pruebas, captura de paquetes, análisis de flujo de información y verificación de la configuración de equipos activos de red, se puede ofrecer una solución óptima para depurar y optimizar el funcionamiento de la red.

Analizar el tráfico que circula por una red o un determinado dispositivo que forme parte de ella trae muchas ventajas consigo, sobre todo a los administradores, técnicos y desarrolladores que se encuentren a cargo de tareas de monitorización de redes y desarrollo de aplicaciones en estos campos. Teniendo la información suficiente, se pueden detectar todo tipo de anomalías, ataques y problemas a los que los administradores se enfrentan diariamente. Es sabido que las redes que no son bien mantenidas o tienen vulnerabilidades fuertes en su estructura son blanco fácil para los atacantes, donde podrán entrar y hacer daño fácilmente. Pero conocer el tráfico que normalmente circula por la red y saberlo analizar, brinda un escudo bastante seguro para protegerse de cualquier problema que se pueda presentar. Las herramientas de Análisis del Tráfico de Red proporcionan funcionalidades que permiten un análisis intuitivo y una mayor comprensión de los protocolos de red utilizados en la mayoría de las redes.

Teniendo en cuenta que la calidad de servicio (QoS) es el rendimiento de extremo a extremo de los servicios electrónicos tal como lo percibe el usuario final. Una red o un determinado dispositivo debe garantizar un cierto nivel de calidad de servicio, para un nivel de tráfico que sigue un conjunto especificado de parámetros. De allí que la implementación de políticas de calidad de servicio basadas en rigurosos análisis del tráfico existente pueda dar buenos dividendos teniendo en cuenta los siguientes puntos:

- Asignar ancho de banda en forma diferenciada.
- Evitar y/o administrar la congestión en la red.
- Manejar prioridades de acuerdo al tipo de tráfico.
- Modelar el tráfico de la red.

Estas circunstancias han motivado el estudio de las propiedades estadísticas del tráfico que se generan, determinado por el comportamiento de los usuarios y por las acciones de los protocolos implicados en las comunicaciones entre los diferentes elementos de la red. Resulta igualmente importante la realización de análisis previos que ayuden a determinar el impacto que tendrán los nuevos servicios en el futuro, además de tomar las precauciones necesarias para evitar los posibles problemas que puedan aparecer, lo que provocaría el consecuente descontento de los usuarios y la mala imagen que se proyecta del servicio. Estos análisis previos, si son llevados a cabo con la seriedad y la exactitud necesaria, permitirían predecir el comportamiento del sistema sin la necesidad de desplegar grandes infraestructuras, evitando con ello, los elevados costes de desarrollo que conlleva.

2.7 Analizadores de Protocolos.

En innumerables ocasiones encontramos que dentro de una red ocurren problemas bien sea de tráfico, ataques de virus u otros, estos problemas son difíciles de detectar sin la ayuda de herramientas que permitan ver el tráfico que está circulando por toda la red o algún sector en específico. Muchas de estas herramientas son software que se instalan sobre una computadora o cualquier otro terminal, sin dar por descontado la existencia de otros dispositivos, que se encuentren especializados en este tipo de tareas, los cuales permiten ver mucho más.

Sería importante hacer énfasis en el gran sentido de ética y la responsabilidad que implica el uso de estas herramientas, debido a que como permiten visualizar los paquetes que circulan por la red, estos paquetes pueden contener información personal de cada usuario o información sensible de la empresa que no debiera ser de conocimiento público. En cualquier caso, nunca este tipo de tecnologías debieran ser utilizadas como una herramienta de espionaje. Por seguridad, los analizadores de protocolos requieren permisos de administrador del sistema donde se ejecuten, para poder realizar las capturas del tráfico.

Los analizadores de protocolos o de red, también conocidos vulgarmente como “sniffers” son herramientas de gran ayuda para los administradores de las redes, debido a que permiten el análisis detallado de muchos factores del comportamiento de las mismas. Estas aplicaciones permiten capturar una copia de los paquetes que circulan por la red para su análisis posterior. Los más avanzados incluyen una interfaz gráfica capaz de mostrar los campos de los protocolos de comunicación de los distintos niveles, obtener estadísticas de utilización y facilitar considerablemente el posterior análisis de los datos capturados. De este modo se facilita la detección de problemas, así como la depuración del software de red durante su fase de elaboración. Por ejemplo, un administrador de red que detecte que las prestaciones de la red son bajas puede utilizar uno de estos analizadores para detectar los segmentos de la red, los protocolos y las máquinas que están generando más tráfico, de esta forma poder llevar a cabo las acciones necesarias, o bien, verificar el correcto funcionamiento de los diferentes dispositivos de red (hosts, servidores, routers, cortafuegos, NAT, etc).

A la hora de elegir un analizador de protocolos se puede encontrar una abundante oferta, tanto de productos comerciales como de software de libre distribución. A continuación nos centraremos Wireshark, como herramienta para el análisis de protocolos.

2.7.1 Wireshark.

Ethereal es una herramienta gráfica utilizada por los profesionales y/o administradores de la red para identificar y analizar el tipo de tráfico en un momento determinado. Ethereal permite analizar los paquetes de datos en una red activa como también desde un archivo de lectura previamente generado, un caso particular es generar un archivo con la herramienta en línea de comandos TCPDUMP y luego analizarlo con Ethereal. A partir del año 2006 Ethereal es conocido como WireShark y hoy en día está categorizado como uno de los cinco sniffer más utilizados.

Algunas de las características de WireShark (6) son las siguientes:

- Disponible para UNIX, LINUX, Windows y Mac OS.
- Se provee bajo la licencia GPL.
- Captura los paquetes directamente desde una interfaz de red.
- Permite obtener detalladamente la información del protocolo utilizado en el paquete capturado.
- Es compatible con más de 480 protocolos.
- Cuenta con la capacidad de importar/exportar los paquetes capturados desde/hacia otros programas.
- Filtra los paquetes que cumplan con un criterio definido previamente.
- Permite obtener estadísticas.

- Sus funciones gráficas son muy poderosas ya que identifica mediante el uso de colores los paquetes que cumplen con los filtros establecidos.

Es importante tener presente que WireShark no es un IDS (Intrusion Detection System) ya que no es capaz de generar una alerta cuando se presentan casos anómalos en la red. Sin embargo, permite analizar y solventar comportamientos anómalos en el tráfico de la red.

2.8 Punto de partida.

Hasta donde conocen los autores, no hay herramientas de uso general para la captura de tráfico IP sobre un dispositivo móvil que utilice Windows Phone como sistema operativo. En la actualidad son múltiples las herramientas que permiten hacer captura y análisis de tráfico y ejemplo de ellas son las citadas en el epígrafe 2.7, pero se debe resaltar que en su gran mayoría todas estas han sido concebidas para trabajar sobre ordenadores, los cuales poseen prestaciones superiores a las que se pueden encontrar en los dispositivos móviles actuales.

Esta investigación tiene como punto de partida dos herramientas creadas con propósitos similares. La primera de ellas un analizador de protocolo para dispositivos móviles que utilicen Symbia como sistema operativo. Dicha herramienta denominada SymPA, fue diseñada y desarrollada por la Dra. Almudena Días y el Dr. Pedro Merino (7) y fue creada para cubrir necesidades de captura de tráfico IP en el contexto de terminales que soportan el sistema operativo Symbian OS. La segunda es TestelDroid, una herramienta para el monitoreo de parámetros y tráfico en los dispositivos basados en Android. La misma fue diseñada y desarrollada por Andrés Álvarez, la Dra. Almudena Días y el Dr. Pedro Merino.

SymPA es un analizador de protocolos que se ejecuta en el terminal y que permite capturar el tráfico TCP/IP entrante sin alterar el comportamiento de otras aplicaciones ni el resto de la funcionalidad del terminal. En su diseño se han tenido en cuenta principalmente los siguientes objetivos:

- Capturar todos los paquetes IP entrantes, asegurando que no se deja de analizar tráfico por problemas de saturación.
- Llevar a cabo la gestión eficiente de los recursos, de acuerdo a las limitaciones de potencia de procesamiento y duración de la batería de los dispositivos móviles.
- Incorporar funciones básicas de generación de tráfico, como el test de conectividad mediante ping.
- Obtener información referente a la interfaz de red.
- Obtener información referente a rastreo GPS y localización.
- Obtener información referente monitoreo de baterías.
- Proporcionar una interfaz que permita procesar la información capturada y exportarla a otros entornos.
- Analizar los problemas de seguridad en dispositivos móviles.
- Depurar los protocolos de red en las redes celulares.
- Depurar los nuevos protocolos para nuevos servicios móviles.

- Detectar errores en las implementaciones de protocolo de red para dispositivos móviles.
- Detectar conductas irregulares de los protocolos tradicionales de móviles entornos.

Otra característica que diferencia a SymPA es que permite capturar tráfico generado por aplicaciones reales de forma que no es necesario que sea la propia herramienta la que origine el tráfico. Esto la hace adecuada para ayudar en el desarrollo de nuevas aplicaciones. Por otro lado las medidas se llevan a cabo en condiciones de carga de red reales, lo que asegura la objetividad de los resultados obtenidos.

Por su parte TestelDroid surge con el fin de caracterizar el comportamiento de las comunicaciones celulares, en dispositivos que utilicen Android (8). La información obtenida se pueden registrar y exportar para su posterior análisis con otras herramientas, tales como Wireshark. La Tabla 2.1 ofrece un resumen de las características proporcionadas por la herramienta.

Tabla 2.1: Resumen de las características proporcionadas por la herramienta TestelDroid.

| Indicador | Información obtenible. |
|---------------------------|---|
| Red. | Operador. RAT (Tecnología de acceso radio). CID (identificación de células). LAC (Código de Ubicación). RSSI (Indicador de fuerza de la señal radio). PSC (código primario de codificación). |
| Células vecinas. | PSC. RSSI. RSCP (Código de potencia de la señal recibida). RAT (Tecnología de acceso radio). |
| Batería. | Nivel de la batería. Temperatura (°C). Voltaje (mV). Corriente (mA). |
| GPS. | Longitud. Latitud. Altitud. Velocidad. |
| Tráfico IP. | Formato Pcap. Marcas de tiempo de llegada. Modo promiscuo. |
| Prueba de conectividad. | Ping. Puertos abiertos. |
| Prueba de tráfico activo. | Servidor-cliente de móvil a móvil Transferencia de autogenerados. Bit rate monitoreo. Velocidad de transferencia promedio. |

La información proporcionada por TestelDroid se recoge en tres niveles diferentes, como se muestra en la figura 2.2.

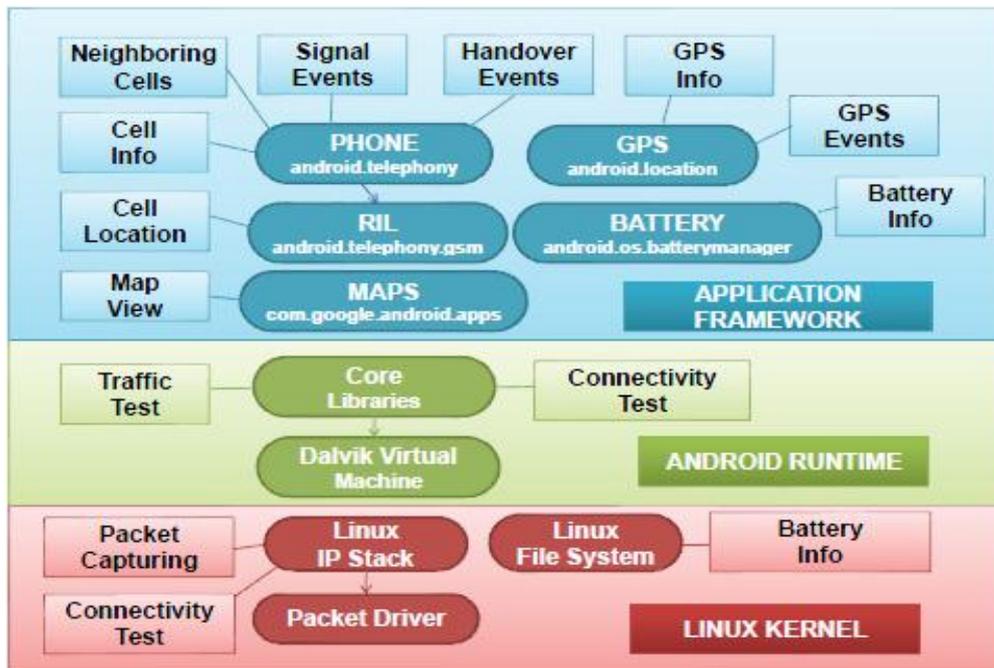


Figura 2.2: Información proporcionada por TestelDroid dividida en tres niveles.

A nivel aplicación del framework (zona azul), las API de Android proporciona información referente a: la celda actual, las celdas vecinos, de ubicación GPS, entre otras. Android a nivel de tiempo de ejecución (área verde) proporciona parte de las pruebas de conectividad (comprueba si un socket puede ser abierto en un host/puerto especificado) y pruebas de tráfico (transferencia de un archivo en un modelo cliente / servidor). Por último, algunas tareas se realizan directamente en el kernel de Linux (zona roja), como la captura de paquetes, las pruebas de conectividad (ping) y la información de consumo de batería.

CAPÍTULO 3: WINDOWS PHONE

Windows Phone 7 es un sistema operativo móvil desarrollado por Microsoft, como sucesor de la plataforma Windows Mobile. Con Windows Phone 7 Microsoft ofrece una nueva interfaz de usuario, integra varios servicios en el sistema operativo y planea un estricto control del hardware que implementará el sistema operativo, evitando la fragmentación con la evolución del sistema.

El 15 de Febrero de 2010, durante el Mobile World Congress celebrado en Barcelona, Microsoft presentó al mundo su nuevo sistema operativo para smartphones: Windows Phone 7. Su desarrollo comenzó en 2008 tras la reorganización del equipo de Windows Mobile y la cancelación del proyecto “Photon”, la versión del sistema en la que Microsoft trabajaba desde 2004 con pocos resultados. Originalmente se planteó su salida al mercado en 2009, pero una serie de retrasos obligaron a preparar una versión de transición de Windows Mobile (WM 6.5) y retrasar el lanzamiento de Windows Phone 7 hasta Octubre de 2010 en Europa, Noviembre de 2010 en USA y 2011 para Asia.

Dentro del conjunto de novedades presentadas se puede citar la interface de usuario, denominada “Metro”, basada en mosaicos dinámicos, fue la que tuvo la mayor acogida, pues muestra de forma sencilla y efectiva información útil al usuario. Además propone un nuevo concepto llamado HUB (concentrador), un lugar donde centralizar acciones y agrupar aplicaciones por la actividad a la que se destinan, de esta manera se puede encontrar el HUB de Imágenes, Zune, Office o Xbox Live, cada uno de los cuales dará acceso a tareas específicas como música, documentos o juegos respectivamente.

Un año después de su presentación, de nuevo en el marco del Mobile World Congress, Windows Phone 7, con apenas unos meses de vida, presenta importantes novedades. A su primera actualización de Marzo de 2011, que incluye mejoras en el rendimiento general así como la inclusión de copiar y pegar, se une una nueva actualización, llamada “Mango” que incluye el motor de Internet Explorer 9, con su soporte para HTML5, Multitarea en aplicaciones de terceros e integración con Xbox 360 y Kinect. En Mayo de 2011 Microsoft ha puesto a disposición de los desarrolladores la primera versión beta de las nuevas herramientas de desarrollo de la versión “Mango” de Windows Phone.

Hace solo unos días que Microsoft puso en marcha su programa gradual de actualización a la nueva versión de Windows Phone 7.5 “Mango”, un proceso automático para los propietarios de móviles basados en esta plataforma. Esta nueva actualización, no se encuentra disponible para todos los usuarios aún. En esta primera

etapa se ha puesto a disposición solo de un grupo de usuarios con el objetivo de recolectar información sobre posibles fallos en dichas actualizaciones. Aunque es válido señalar que hasta el momento los resultados han sido positivos, lo que indica que la actualización estará disponible para el resto de los usuarios muy pronto.

Sería bueno señalar que “Mango” incluye esperadísimas funcionalidades como la multitarea, el acceso a la cámara por parte de aplicaciones para desarrolladores, imprescindible para que florezcan aplicaciones de éxito como las que se han visto en otras plataformas, también aplicaciones de Office, mejoras en el motor Javascript de Internet Explorer, un HUB de juegos renovado, integración con Twitter y agrupación de las conversaciones por hilos entre otras muchas.

3.1 Modelo de Hardware.

Como fabricante del sistema, Microsoft requiere que todo teléfono que desee ejecutar Windows Phone 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. En este modelo, conseguimos aunar las ventajas del modelo iPhone, donde todos los usuarios obtienen la misma experiencia de uso y los desarrolladores saben que la aplicación funcionará de forma idéntica en todos los dispositivos Windows Phone 7, y las del modelo Windows Mobile / Android, al no estar encerrados en un solo hardware, existen distintos dispositivos de varios fabricantes, todos con unas características mínimas comunes pero con suficientes diferencias como para sentir que se puede escoger el terminal que más se ajuste a las diversas exigencias personales.

Con este modelo en mente, Microsoft creó especificaciones mínimas iniciales, llamadas chasis 1, por las que todo fabricante que deseara crear terminales para el nuevo sistema debe guiarse, estas son (9):

- **Procesador:** ARMv7 Cortex/Scorpion a 1Ghz.
- **Procesador gráfico:** Soporte hardware completo de DirectX9.
- **Memoria:** 256 RAM/8GB ROM.
- **Sensores:** A-GPS, Acelerómetro, brújula, iluminación, proximidad.
- **Cámara:** 5mpx con flash y botón físico de disparo.
- **Multimedia:** Aceleración de audio y video por hardware.
- **Pantalla:** Capacitiva, resolución: 800x480.
- **Botones físicos:** Inicio, Buscar, Atrás.

Microsoft ha anunciado que en el futuro aparecerán las especificaciones para chasis 2 y chasis 3, que soportarán otros tipos de teléfonos y mantendrán una relación de unidad con el chasis 1, compartiendo un mínimo denominador común que permita obtener la misma experiencia de usuario independientemente del dispositivo hardware que se utilice.

3.2 Modelo de Software

Windows Phone está basado en Windows CE 6.0 R3, un sistema más moderno y avanzado que Windows CE 5.2 el cual era utilizado en los Windows Mobile 6.

Windows CE 6.0 supuso un rediseño total del sistema con respecto a su anterior versión. Ahora, la Shell y la plataforma de aplicaciones residen en memoria de usuario, mientras que el Kernel, los drivers, el sistema de archivos, la red, el sistema de gráficos y el sistema de actualizaciones residen en el espacio de Kernel.

Sobre toda la infraestructura provista por Windows CE 6.0 R3 se levanta el modelo de software de Windows Phone 7 (Figura 3.1). En este sentido al igual que con el hardware, Microsoft ha pretendido crear un modelo conciso, pensado para facilitar el uso del dispositivo y aportar cohesión a la interface de usuario.

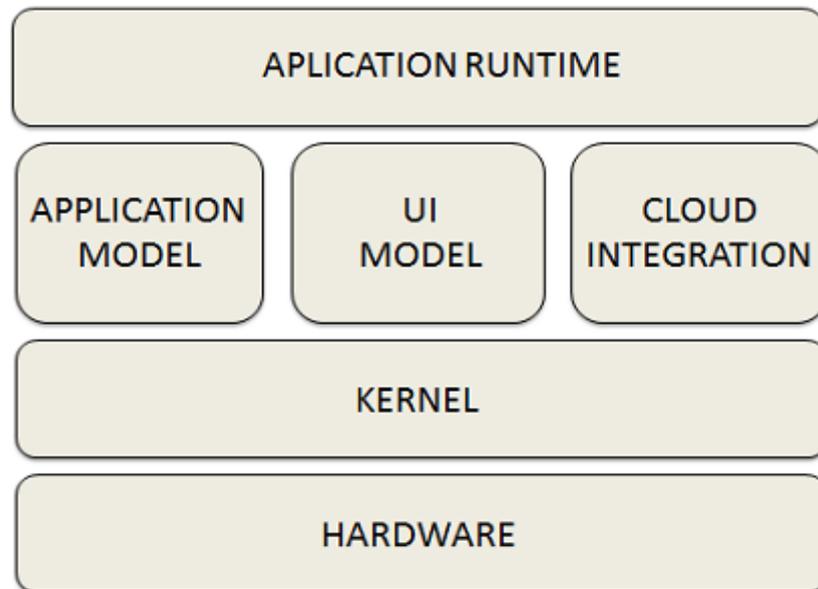


Figura 3.1: Modelo de software de Windows Phone 7.

3.2.1 Modelo de Aplicación.

En Windows Phone las aplicaciones se despliegan en forma de paquete XAP, básicamente es un archivo comprimido dentro del cual se pueden encontrar los ensamblados y recursos originales de nuestra aplicación. La única forma de instalar una aplicación en Windows Phone es mediante la tienda oficial de Microsoft, el Marketplace, en la cual debemos registrarnos como desarrolladores para poder vender nuestras aplicaciones.

Para garantizar la seguridad del sistema y evitar la piratería, el malware o virus, a cada aplicación se le asigna un ID único y un certificado de seguridad emitido cuando nuestra aplicación es aprobada en el Marketplace de Windows Phone. Este certificado puede instalarse en modo demo, por lo que solo con comprobar el modo de este certificado desde nuestra aplicación podremos saber si nos encontramos en modo de prueba o modo completo y restringir o cambiar la forma en que nuestra aplicación se comporta, evitando el desarrollo de versiones “lite” (programa ligero, que no es completo si no una parte del original) como en otras plataformas.

3.2.2 Modelo de UI.

El modelo de interface de usuario de Windows Phone se compone de elementos, páginas y sesiones. Un elemento, es todo control que se muestra al usuario; una página, es una agrupación lógica de elementos y una sesión, es el conjunto de interacciones que

realiza un usuario sobre nuestra aplicación e incluso puede involucrar a otras aplicaciones.

Si se pretende por ejemplo realizar una aplicación que necesite del usuario una foto, se tiene un botón (elemento) en la página que muestra el HUB de imágenes y una vez que el usuario ha seleccionado una, vuelve a nuestra aplicación y se muestra la imagen escogida. Este conjunto de acciones se engloba dentro de una sesión.

3.2.3 Integración con la nube.

Con el lanzamiento de Windows Phone 7, Microsoft ha proporcionado a los desarrolladores una gran cantidad de servicios basados en la nube. Estos servicios pueden ser aprovechados por los desarrolladores para sus propias aplicaciones.

Al hacer referencia a la nube, se puede definir como el conjunto de servidores, redes y puertas de enlace que proporcionan un entorno de computación que se entrega a través de internet como servicio de pago por uso.

Windows Phone nace con una clara integración con la nube. Por defecto tenemos integración con servicios como Exchange, Google Mail, Hotmail, Xbox Live, Skydrive, Facebook o Bing. En la versión actual del kit de desarrollo no existen APIs que permitan a nuestras aplicaciones acceder a estos servicios directamente, pero se espera que aparezcan en próximas actualizaciones.

Recientemente Microsoft ha liberado el Windows Phone Cloud Services SDK, un kit de desarrollo que dotará de servicios exclusivos en la nube las aplicaciones para Windows Phone 7. Dentro de los servicios más importantes se encuentran el reconocimiento OCR, voz a texto y una serie de APIs para intercomunicar dispositivos sin necesidad de tener una dirección IP fija en ellos.

3.2.4 Application Runtime

Hasta ahora se ha mostrado una visión general de los modelos de hardware y software en los que se basa Windows Phone, en esta sección se examinará la parte que más nos involucra: el Application Runtime. A continuación aparecen algunas de las interrogantes que se estarán respondiendo. ¿Dónde y cómo se ejecutarán las aplicaciones? ¿Qué limitaciones se encontrarán? ¿Cuáles frameworks se tendrán al alcance para desarrollar?

En Windows Phone se pueden encontrar dos frameworks claramente diferenciados: Silverlight para Windows Phone y XNA para Windows Phone.

- Silverlight para Windows Phone es un framework basado en Silverlight que nos permitirá crear aplicaciones multimedia ricas que se ejecuten de forma nativa en Windows Phone, con una interface creada en XAML (Véase epígrafe 3.4.2).
- XNA para Windows Phone a su vez se basa en XNA para Windows, es una solución Multiscreen (Xbox, Windows, Windows Phone) 2D y 3D para crear juegos con calidad profesional en nuestro dispositivo (Véase epígrafe 3.4.1).

Estos frameworks se ejecutan sobre un sandbox (aislamiento de procesos) de .NET que les facilita el acceso al hardware, sensores, almacenamiento, localización, etc. Esto quiere decir que las aplicaciones nunca tendrán acceso nativo al sistema y siempre se ejecutarán aisladas del sistema y entre ellas mismas, no pudiendo compartir espacio de

almacenamiento ni ningún otro tipo de información a no ser que se usen servicios externos en la nube para ello.

La figura 3.2 muestra un esquema de cómo está estructurado el Application Runtime de Windows Phone (10).

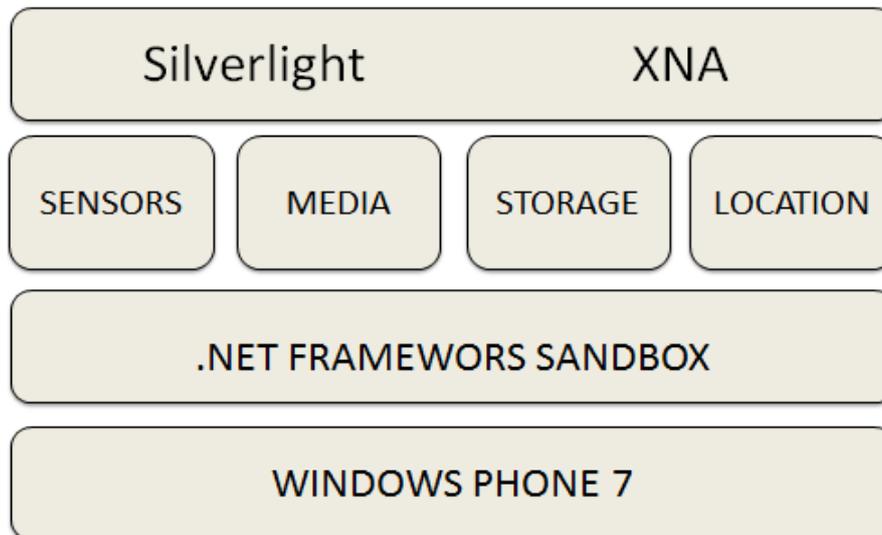


Figura 3.2: Esquema del Application Runtime de Windows Phone.

Con la nueva versión de las herramientas de desarrollo para Mango (Windows Phone 7.5) se abre a la posibilidad de tener aplicaciones mixtas, que usen Silverlight para la interface de usuario y XNA para gráficos 3D, creando una experiencia de usuario más rica y completa.

3.3 Recursos para desarrolladores.

La herramienta principal para el desarrollo de aplicaciones de Windows Phone 7 es el Visual Studio. Tanto la depuración y la codificación, como las pruebas se pueden realizar desde este entorno de desarrollo integrado (IDE). Las herramientas de Windows Phone Developer se pueden descargar e instalar junto a un emulador del teléfono. Este emulador permite desarrollar aplicaciones sin necesidad de un teléfono físico, pese a ello, es recomendable probar la aplicación en dispositivos físicos reales antes del despliegue.

Windows Phone Developer Tools incluye:

- Visual Studio 2010 Express para Windows Phone CTP.
- Recursos de Windows Phone Emulator.
- Silverlight 4 Tools para Visual Studio.
- XNA Game Studio 4.0.
- Microsoft Expression Blend para Windows Phone.

Requisitos del sistema (11):

- Sistemas operativos compatibles: Windows 7; Windows Vista.

- Windows Vista (x86 y x64) con Service Pack 2, todas las ediciones excepto Starter Edition.
- Windows 7 (x86 y x64), todas las ediciones excepto Starter Edition.
- Para la instalación se requieren 3 GB de espacio de disco disponible en la unidad del sistema.
- 2 GB de RAM.
- Windows Phone Emulator requiere una tarjeta gráfica con funcionalidad DirectX 10 y un controlador WDDM 1.1.
- Windows Phone Developer Tools es compatible con la versión final de Visual Studio 2010.

3.4 Frameworks para desarrollar en Windows Phone 7.

Microsoft Windows Phone 7 proporciona toda una plataforma informática para teléfonos móviles. Esta ha sido diseñada para que la programación de las aplicaciones, el intercambio y la reutilización de código, se lleve a cabo mucho más fácil, mediante el aprovechamiento de dos tecnologías comunes que ya son ampliamente utilizadas por los usuarios: Silverlight y XNA.

Aunque ambos cohabitan sobre el mismo framework de .NET su propósito es distinto, debido a que XNA fue concebido para desarrollar aplicaciones que hagan uso extensivo de las capacidades gráficas del dispositivo, por ejemplo los juegos, mientras que Silverlight está pensado para aplicaciones que utilicen de manera más general las capacidades multimedia, los gráficos, el audio, los sensores y las conexiones. Ello no significa que no se puede desarrollar un lector de RSS con XNA, ni es imposible desarrollar un juego con Silverlight, pero no son para lo que están pensados.

A parte de las diferencias en la dificultad de ejecución, XNA y Silverlight, aunque comparten en su base el mismo framework de .NET, soportan características muy diferentes del mismo, siempre pensando en que cada uno tenga acceso solo a lo necesario para aligerar su huella en memoria y su carga. A continuación la investigación se centrará en Silverlight, para lograr sus objetivos.

3.4.1 Silverlight.

Silverlight, que está ahora en su cuarta versión, se desarrolló originalmente para proporcionar a los desarrolladores una forma de crear sofisticadas aplicaciones de Internet, utilizando las herramientas y lenguajes estándar de .NET. Rich Internet Applications (RIA) se convirtió en el estándar de facto de estilo de aplicaciones para la Web 2.0. Estas aplicaciones típicamente renuncian al modelo tradicional donde los usuarios hacen clic en un enlace y toda la página se actualiza con nuevos datos. En cambio, las aplicaciones web basadas en RIA hacen uso de un estándar de programación llamado Asynchronous JavaScript y XML (AJAX) para actualizar solo las partes de la página web que realmente necesitan ser actualizadas en lugar de actualizar toda la página.

Tomando las ventajas del método de actualización de RIA se mejora la experiencia del usuario final, los usuarios no tienen que sentarse y esperar que una página entera sea cargar, cuando solo un área pequeña es en realidad la que va a cambiar. También se

reduce el ancho de banda real utilizado en el servidor web, ya que una cantidad mucho menor de datos se transmiten por el cable.

Silverlight se deriva de la tecnología Windows Presentation Foundation (WPF). WPF utiliza Extensible Application Markup Language (XAML) para definir la interfaz de usuario y utiliza en sus clases el código .NET framework para implementar la funcionalidad de la aplicación. Silverlight utiliza un conjunto reducido de WPF y .NET framework. Este fue diseñado para proporcionar una interfaz de usuario ligera, atractiva e interactiva, con amplias capacidades para aprovechar al máximo las características del dispositivo. Puede ser entregado a un navegador Web y ejecutarse en un servidor, o bien, se puede empaquetar en un archivo XAP y ejecutar de forma nativa en los dispositivos que lo soporten. Silverlight proporciona un entorno de desarrollo que es ideal para aplicaciones de negocio y aplicaciones de consumo que muestran, recopilan y procesan la información.

Silverlight es compatible con múltiples plataformas y con muchos de los navegadores web más populares. En la tabla 3.1 se enumeran algunos de los sistemas operativos y navegadores que soporta Silverlight 4.

Tabla 3.1: Plataformas y navegadores que soportan Silverlight 4.

| Sistemas operativos. | Navegadores |
|-------------------------|--|
| Windows 7. | Internet Explorer (6, 7, 8), Firefox 3, Safari (3, 4), Chrome 4. |
| Windows Vista. | Internet Explorer (6, 7, 8), Firefox 3, Safari (3, 4), Chrome 4. |
| Windows Server 2008. | Internet Explorer (7, 8), Firefox 3, Chrome 4. |
| Windows Server 2008 R2. | R2 Internet Explorer 8, Firefox 3, Chrome 4. |
| Windows XP SP2, SP3. | Internet Explorer (6, 7, 8), Firefox 3, Chrome 4. |
| Windows 2000 SP4. | Internet Explorer 6. |
| Mac OS 10.4.11 +. | Firefox 3, Safari (3, 4). |

El desarrollo de Silverlight para el teléfono de Windows 7 es similar a la plataforma Silverlight de desarrollo para la web. Aunque Windows Phone 7 utiliza una versión personalizada de Silverlight, la mayoría de las funcionalidades disponibles en la web también están disponibles en la versión móvil de la ejecución. Por tanto, para portar aplicaciones escritas, usando Silverlight para la web a Windows Phone no necesita mucho esfuerzo. De hecho, la mayoría del código se traslada muy bien.

Silverlight también proporciona un potente conjunto de controles que se pueden utilizar en sus aplicaciones para controlar la presentación de los datos, la entrada del usuario y mucho más. Además, la mayoría de estos controles han sido diseñados para hacer uso del nuevo estilo de la interfaz de usuario Metro, que es inherente a Windows Phone 7.

3.5 Terminología.

Windows Phone 7 ofrece todas las ventajas de una plataforma estándar, unido a la experiencia de desarrollo consistente en la plataforma de Microsoft para dispositivos de distintos fabricantes. No es un sustituto de Windows Mobile, que sigue ofreciendo una poderosa plataforma para una amplia gama de dispositivos y escenarios de aplicación. En su lugar, Windows Phone 7 incorpora un completo conjunto de características

necesarias para crear aplicaciones que satisfagan las necesidades de las empresas y los consumidores, para permitir a los desarrolladores crear fácilmente poderosas aplicaciones interactivas y atractivas, volviendo a utilizar sus habilidades y conocimientos de los actuales entornos de desarrollo modernos, como el Silverlight de Microsoft y las plataformas de desarrollo Microsoft XNA.

A continuación se muestran un conjunto de elementos que describen las tecnologías, modelos, marcos, aplicaciones, servicios y otros términos comúnmente asociados con el desarrollo para dispositivos móviles que utilicen Windows Phone 7 (WP7).

3.5.1 Protocolos.

Dentro de los protocolos más utilizados para el desarrollo de aplicaciones en WP7 se pueden encontrar:

- Secure Sockets Layer (SSL), un protocolo de cifrado para asegurar la comunicación entre clientes y servicios sobre una red TCP / IP. A menudo descrito como HTTPS, que funciona por defecto a través del puerto 443 en lugar del puerto 80. La aplicación más reciente es el Transport Layer Security (TLS).
- Transport Layer Security (TLS) es un protocolo de cifrado para asegurar la comunicación entre clientes y servicios sobre una red TCP / IP. Proporciona una mejor seguridad general de SSL y se utiliza a menudo para asegurar la comunicación de mensajes, así como las peticiones web y las respuestas.
- Simple Object Access Protocol (SOAP) es un protocolo que permite a los clientes ejecutar métodos en un servidor remoto para obtener información o realizar procesamiento. Es un formato basado en XML que empaqueta los datos en una arquitectura SOAP. Los mensajes SOAP transportan tanto información de payload (carga útil) como de metadatos, proporcionando seguridad y fiabilidad independientemente del transporte. Además pueden añadir capas de funcionalidad para la comunicación, tales como transacciones atómicas.
- Open Data Protocol (OData) es un protocolo de Internet para consultar y actualizar datos. OData se basa en tecnologías Web tales como HTTP, PubAtom y JavaScript Object Notation para facilitar el acceso a la información de una variedad de aplicaciones, servicios y tiendas. Permite tener acceso a un origen de datos de forma remota mediante protocolos de Internet estándar sobre HTTP.
- Atom Publishing es un protocolo de nivel de aplicación para la publicación y edición de recursos web mediante HTTP. Este presenta un formato basado en XML para los datos, el cual utiliza Transferencia de Estado Representacional (REST) y un estilo arquitectónico donde los nombres de las colecciones y las entidades se definen en la cadena de consulta de una solicitud.

Representational State Transfer (REST) es un estilo arquitectónico utilizado para exponer los datos de los servicios, que permite un acceso simple y la manipulación opcional de la información sin necesidad de que el servidor mantenga el estado entre las solicitudes.

3.5.2 Tecnologías y Herramientas.

Al hacer referencia a las tecnologías y herramientas que se pueden utilizar para desarrollar aplicaciones para WP7 sin dudas existen algunas que no se pueden dejar de citar, por ejemplo:

Un elemento a tener en cuenta es Prism, una biblioteca de utilidad gratuita del grupo de prácticas y patrones de Microsoft. Los componentes de esta biblioteca pueden ayudar a los desarrolladores a crear aplicaciones para WPF, Silverlight y Windows Phone que son más fáciles de mantener y actualizar a medida que cambian las necesidades. Prism ayuda a diseñar y construir aplicaciones utilizando componentes débilmente acoplados, que pueden evolucionar de forma independiente, pero que puede ser fácil y perfectamente integrado en el total de la aplicación.

Windows Phone 7 implementa la interacción a través del tacto y los gestos, entre el teléfono y las aplicaciones que se ejecutan en él. Se reconoce un máximo de cuatro puntos de contacto y los gestos tales como hacer clic, doble-clic, panear, chasquido, contraer, estirar y sostener contacto. Ello es reconocido como Touch Input.

En estrecha relación con lo expuesto anteriormente, el soft input panel (teclado de software) es un método de entrada por pantalla para dispositivos que no incluyen un teclado de hardware (Ver figura 3.3).



Figura 3.3: Soft input panel.

Al hacer alusión a las pulsaciones que pueden ser ejercidas sobre los dispositivos de WP7 surge AllKeys API, el cual permite a sus programas solicitar todas las pulsaciones de teclas para enviarlas directamente a la aplicación solicitante. Por lo general, algunos botones son interceptados por el sistema operativo para uso propio, pero los juegos y otras aplicaciones, puede que deseen acceder a estos botones para uso propio.

Otra herramienta muy útil es el comprobador de aplicación (AppVerifier), una herramienta de prueba de software que se utiliza para comprobar la estabilidad de la aplicación y detectar errores de programación comunes asociados con la gestión de memoria. AppVerifier puede detectar y localizar fugas de memoria, pérdidas de identificadores y daños en la pila.

También se puede encontrar el proximity sensor (sensor de proximidad), el mismo precisa la capacidad del dispositivo para detectar si el teléfono está cerca de un objeto,

como el cuerpo humano. El sistema operativo utiliza esta opción para cambiar el comportamiento cuando se utiliza el dispositivo en modo de teléfono.

Por otra parte Acelerómetro permite medir la capacidad de aceleración en los tres planos y la dirección de la fuerza de gravedad que indica la actitud del equipo. Se puede utilizar para detectar el movimiento, incluyendo gestos como el de sacudir el terminal.

Varias veces por segundo, el acelerómetro del teléfono envía informes de la dirección y la magnitud de la fuerza total que se aplica en el teléfono. Esta fuerza se expresa con tres valores X, Y y Z, donde X es la horizontal, Y es la vertical y Z es la perpendicular a la pantalla. La magnitud de cada valor es un multiplicador de g (la fuerza de gravedad en la superficie de la Tierra). Cada valor se limita a un rango de -2 a 2. Si el teléfono está en reposo sobre una mesa con la pantalla hacia arriba, los valores reportados para X e Y son casi cero y el valor de Z es aproximadamente -1.

Otra tecnología a tener en cuenta es el Tombstoning referente a la posibilidad de ejecutar aplicaciones en segundo plano. El proceso que se esté ejecutando actualmente debe detenerse y permitir que otra aplicación o funcionalidad del dispositivo (como una llamada telefónica entrante) pueda ejecutarse. La aplicación detenida debe guardar su estado de manera que pueda continuar desde donde se detuvo o finalizar sin pérdida de datos.

3.5.3 Servicios.

Algunos de los servicios más comúnmente utilizados en la plataforma WP7 son:

- Servicio de localización: Es un servicio de consulta telefónica que el teléfono utiliza para descubrir su ubicación geográfica sobre la base de una serie de factores. Estos factores pueden incluir la capacidad integrada de GPS, la triangulación de puntos de acceso Wi-Fi y de torres de señal telefónica.
- Microsoft Push Notification System (MPNS): Es un servicio que permite a los desarrolladores enviar notificaciones a los teléfonos de los usuarios y que se muestren incluso si la aplicación no se está ejecutando o actualizar una ficha en el menú de inicio del teléfono para indicar que hay nueva información disponible.
- Notificaciones: Son mensajes que se envían desde un servidor al teléfono, a través del servicio de notificación de Windows y pueden ser recibidos incluso cuando la aplicación no se está ejecutando. Las notificaciones pueden mostrar un mensaje en la parte superior de la pantalla o cambiar un mosaico del programa en la pantalla de inicio. Los usuarios deben registrarse para recibir notificaciones en su teléfono.
- Servicios en la nube: Servicios que se ejecutan en uno o más centros de datos remotos, con un hardware especialmente diseñado y un tiempo de ejecución virtual. Proporcionan alta disponibilidad y fiabilidad a través de varias instancias, es posible alcanzar buenos rendimientos y son escalables. En general, los servicios en la nube son factibles si se tiene en cuenta una relación costo-efecto. Siendo capaces de proporcionar acceso local y global a las aplicaciones y servicios sin necesidad de inversión en el costo de mantenimiento, los gastos generales de administración y tiempo de ejecución de una infraestructura de servidores en las instalaciones.

Uno de los servicios más importante a tener en cuenta es el de Windows Marketplace. El cual ofrece una solución de extremo a extremo para los usuarios finales con el objetivo de descubrir, comprar y descargar aplicaciones basadas en Windows para el escritorio y sus dispositivos móviles. Esto incluye una experiencia del usuario final en el dispositivo, un sitio web, así como un portal de autoservicio para que los desarrolladores presenten sus solicitudes para la inclusión en Windows Marketplace.

3.5.4 Patrones.

Model-View-ViewModel (MVVM) es un patrón de diseño especialmente adaptado para Windows Presentation Foundation (WPF) y Silverlight. El objetivo del patrón MVVM es separar de una forma efectiva la visualización (View) de nuestra pantalla, página o ventana, de la lógica de visualización que interactúa y responde ante el usuario y de toda la lógica de negocio.

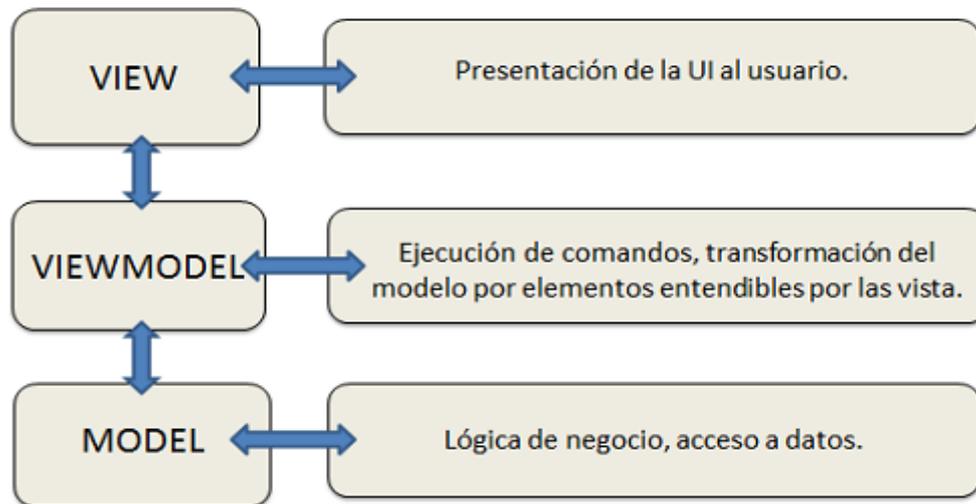


Figura 3.4: Model-View-ViewModel.

Como se puede apreciar en el esquema de la figura 3.4, en MVVM la vista (View) no conoce el modelo (Model) subyacente, solo se relaciona con la vista-modelo (ViewModel) que se encarga de comunicarse con el modelo para recuperar o hacer persistir información y expone esta información hacia la vista. Mientras que vista-modelo también es la encargada de realizar las acciones demandadas por la interface, de forma que la vista solo sea una definición de la interface sin lógica ni código y la vista-modelo contenga toda esta lógica totalmente aislada de la representación visual de la misma.

3.5.5 Diseños y controles.

Metro es el nombre del sistema de diseño de las interfaces de Windows Phone. Metro está diseñado para combinar los elementos visuales de forma armónica, funcional y atractiva, al mismo tiempo de lograr guiar con claridad a los usuarios finales a obtener los contenidos que desee. Es recomendable tratar de seguir el mismo estilo visual y el paradigma de interacción en su aplicación, que el que se utiliza en las aplicaciones incorporadas y la interfaz de Windows Phone 7 para asegurar que su aplicación es intuitiva y fácil de usar.

Microsoft ofrece una guía para el estilo de diseño Metro y la recomienda para crear una experiencia de interfaz de usuario más consistente y fluida, incluyendo descripciones de

la función de entrada estándar en el marco de interfaz de usuario, los elementos de interacción de software y los controles básicos del sistema. Si se desean vender aplicaciones a través de Windows Marketplace, se deben seguir estas pautas de diseño.

Otro elemento a tener en cuenta en el diseño de las aplicaciones es la barra de aplicación. Esta, hace referencia al área pequeña en la parte inferior de la pantalla que contiene botones para las funciones más utilizadas de la aplicación que se está ejecutando. Esta barra opcional muestra un conjunto de iconos por defecto, aunque el usuario también puede expandir para mostrar los subtítulos de texto de los botones. Una aplicación puede mostrar un máximo de cuatro botones en la barra de aplicaciones.

Con el objetivo de brindar atractivas posibilidades de diseño, Microsoft brinda una serie de controles que pueden ser utilizados por los desarrolladores en WP7, a continuación se hará alusión a algunos de ellos.

- **Pivot control:** Es un control que ofrece vistas individuales sobre conjuntos de datos relacionados. Por lo general, las vistas individuales están relacionadas, por ejemplo, vistas que muestran elementos nuevos, elementos favoritos y elementos vistos recientemente (Véase figura 3.5).



Figura 3.5: Pivot control.

Un pivote es básicamente una interfaz de usuario con pestañas en las que se puede deslizar horizontalmente o dando clic en una de las cabeceras para cambiar a una pestaña diferente. Este estilo de interfaz de usuario es comúnmente utilizado para el correo, el calendario y aplicaciones de configuración, pero también es utilizado por la mayoría de las otras aplicaciones integradas: Internet Explorer, mapas, Marketplace, música, videos, contactos e imágenes.

- **MultiScale Image control:** Es un control que se puede utilizar para mostrar una imagen de gran tamaño sin necesidad que esté completamente descargado en la memoria. Se utiliza una colección de múltiples sub-imágenes a diferentes resoluciones. El control proporciona métodos para desplazarse y hacer zoom sobre la imagen.
- **Panorama de control:** Es un control que ofrece una manera de ver los controles, datos y servicios mediante el uso de un lienzo largo horizontal que se extiende más allá de los bordes de la pantalla. Las áreas de este lienzo se ven como

pequeñas zonas individuales mediante movimientos laterales. Efectivamente, esto permite visualizar un segmento de un área mucho más grande. El usuario puede desplazarse por la ventana horizontalmente a través de la vista más grande para ver el contenido completo (Véase figura 3.6).



Figura 3.6: Panorama de control.

3.6 Propuesta de marco de trabajo.

En concreto, para el desarrollo de la herramienta, en la que se centra esta investigación, se propone tener en cuenta las siguientes recomendaciones:

- La utilización de Visual Studio 2010 Express para Windows Phone CTP como herramienta principal para el desarrollo.
- La utilización de Windows Phone Emulator para probar cada funcionalidad de la herramienta creada.
- La utilización de Microsoft Expression Blend para Windows Phone para el diseño de las interfaces de usuario.
- La utilización de Silverlight 4 Tools para Visual Studio para la implementación de toda la lógica de la aplicación.
- La utilización de la actualización de Windows Phone 7.0 como sistema operativo para desarrollar.
- La utilización de Silverlight como framework para el desarrollo de la herramienta.

A continuación en el capítulo 4 se detallará el diseño de herramienta y se llevará a cabo un análisis de las posibilidades actuales que ofrece la plataforma Windows Phone, para la obtención de las funcionalidades que requiere la herramienta.

CAPÍTULO 4: PROPUESTA DE SOLUCIÓN

Sin lugar a dudas detrás de cada software exitoso se encuentra de manera casi oculta, un no menos relevante, proceso de análisis y diseño del software. Téngase en cuenta que un buen diseño constituye a priori, el basamento de la buena implementación.

El diseño debe proporcionar una completa idea de lo que es el software, enfocado en el comportamiento desde el punto de vista de la implementación de la aplicación. Debe ser una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el software.

Por su parte el análisis de la viabilidad es el estudio que dispone el éxito o fracaso de un proyecto a partir de una serie de datos. Se trata de un recurso útil antes de la iniciación de una obra o del lanzamiento de un nuevo producto. De este modo, se minimiza el margen de error ya que gran parte las circunstancias vinculadas a los proyectos son estudiadas. Puede hablarse de viabilidad técnica para hacer referencia a aquello que atiende a las características tecnológicas y naturales involucradas en un proyecto.

A lo largo del capítulo cuatro, se llevará a cabo el diseño de una herramienta que tendrá como principal funcionalidad la captura de tráfico IP en dispositivos móviles que utilicen Windows Phone como sistema operativo. Además se concebirá un estudio de la viabilidad para llevar a cabo el desarrollo de la herramienta, basado en las posibilidades actuales de la plataforma de desarrollo que brinda Windows Phone.

4.1 Consideraciones básicas.

Las primeras decisiones tomadas fueron relacionadas con el tipo de aplicación que se va a construir y lo adecuado que pueda ser este tipo de aplicación para un dispositivo móvil con sistema operativo Windows Phone 7. También se consideró cómo poner en práctica la visión que se tiene, con un estilo y un patrón que coincidan con la interactividad de las aplicaciones incorporadas en el dispositivo y cómo se pueden utilizar los tipos de controles disponibles en el teléfono para crear una interfaz amigable. Es importante resaltar que en Windows Phone 7, la plataforma de desarrollo que se elija (Microsoft Silverlight o XNA) influirá en el diseño de la aplicación y los tipos de controles disponibles.

Antes de intentar diseñar y construir la aplicación en Windows Phone, se tomaron algunas decisiones básicas sobre el modo en que se utilizarán técnicas como test-driven

development (TDD), patrones de diseño, como el Modelo-Vista-ViewModel y la inyección de dependencia. Unido a ello fue necesario decidir si se reutilizarían componentes de terceros o si utilizarían frameworks que respondieran a las necesidades. También se previó cómo implementar el almacenamiento de datos en el dispositivo y si se aprovecharían capacidades del teléfono, tales como: la cámara y el Sistema de Posicionamiento Global (GPS). Unido a ello se tuvo en cuenta cómo se va a gestionar la conectividad a la red y ser consciente de cómo un dispositivo Windows Phone 7 puede desactivar y pasar a un segundo plano su aplicación.

A pesar de no tener que tomar todas estas decisiones antes de comenzar, tenerlas en cuenta desde el principio, ayudó a estabilizar el diseño y a reducir el número de cambios drásticos, una vez avanzado el diseño.

El diseño y la creación de aplicaciones para pequeños dispositivos móviles como el teléfono, requieren de la concientización de las limitaciones de los recursos inherentes a este tipo de dispositivos. Es necesario lograr que la aplicación se comporte como un "buen ciudadano" en el teléfono, mediante la optimización de rendimiento y el uso de recursos, proporcionando de esta manera, una mejor experiencia para los usuarios y evitando cualquier conducta que afecte negativamente a otras funciones y aplicaciones en el teléfono. Debiéndose prestar especial atención a la asignación y liberación de los recursos que se utilicen, minimizar el consumo de la batería y maximizar el rendimiento en los procesadores de baja potencia, que son los utilizados en los teléfonos móviles. También se manejó la idea de aprovechar al máximo el procesador de gráficos en el dispositivo.

Finalmente, se debe considerar cómo va a interactuar con los servicios remotos que proporcionan los datos que utiliza la aplicación. Puede interactuar con muchos tipos de servicios, incluyendo servicios en la nube como Windows Live y los servicios de localización y notificaciones que se han diseñado específicamente para integrarse con dispositivos Windows Phone 7. Técnicamente, el formato de los datos que se utilizan para implementar la aplicación no es relevante, aunque se debe tener en cuenta la interoperabilidad con otros dispositivos, la seguridad de la comunicación y si es necesario llevar a cabo sincronización de archivos o datos entre el dispositivo y el servidor. Sin embargo, es de vital importancia considerar la utilización de ancho de banda, el volumen de transferencia y los costos de conectividad.

4.2 Propósitos de la herramienta.

La herramienta que se propone tendrá un amplio espectro de aplicación, ya que podrá ser utilizada por los desarrolladores de aplicaciones para teléfonos móviles que utilicen WP7. La utilización de esta herramienta les permitirá hacer un uso óptimo de los recursos disponibles en los dispositivos y dimensionar de forma eficaz sus aplicaciones, sin dejar de lado el análisis de la seguridad implementada. De conjunto con lo anterior, permitirá a los usuarios comunes monitorizar su tráfico IP y hacer los análisis correspondientes en materia de seguridad y economía. Además, la herramienta pudiera aplicarse en entornos académicos facilitando el análisis de protocolos comúnmente utilizados en esta área.

En sentido general la herramienta posibilitará:

- Capturar todos los paquetes IP entrantes, evitando al mismo tiempo la información de sobrecarga.
- Obtener información de la celda.
- Proporcionar interfaces para el procesamiento de la información capturada y su exportación a otros entornos.
- Obtener información de geolocalización.
- Incorporar funciones básicas de generación de tráfico, como el test de conectividad mediante ping.
- Análisis de los problemas de seguridad en dispositivos móviles.
- Depuración de protocolos de red en las redes celulares.
- Depuración de los nuevos protocolos para nuevos servicios móviles.
- Detección de errores en las implementaciones de protocolo de red para dispositivos móviles.
- Detección de conductas irregulares de los protocolos tradicionales de entornos móviles.

Sería importante resaltar que esta herramienta posibilitará todo este conjunto funcionalidades, partiendo de la captura de tramas IP, desde una aplicación que se ejecutará en el propio dispositivo móvil en un segundo plano y sin sobrecargar la capacidad de procesamiento del mismo. Esta aplicación almacenará todo el tráfico IP que haya tenido lugar en un determinado espacio de tiempo en un fichero (véase figura 4.1), para su posterior exportación a otros sistemas en donde tendrá lugar el análisis de esta información (véase figura 4.2). Se propone entonces exportar la información resultante de la captura en un formato legible por el analizador de protocolos Wireshark.

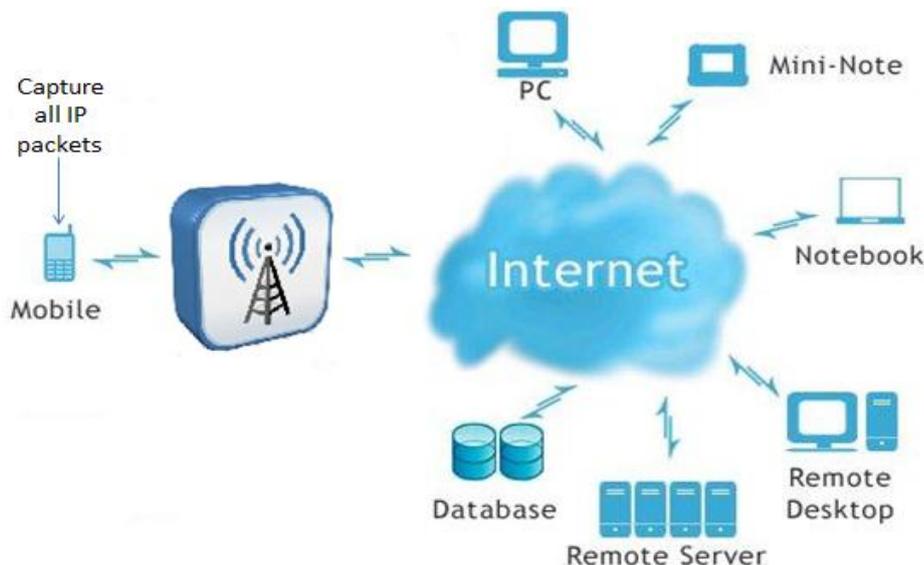


Figura 4.1: Captura de todo el tráfico IP.

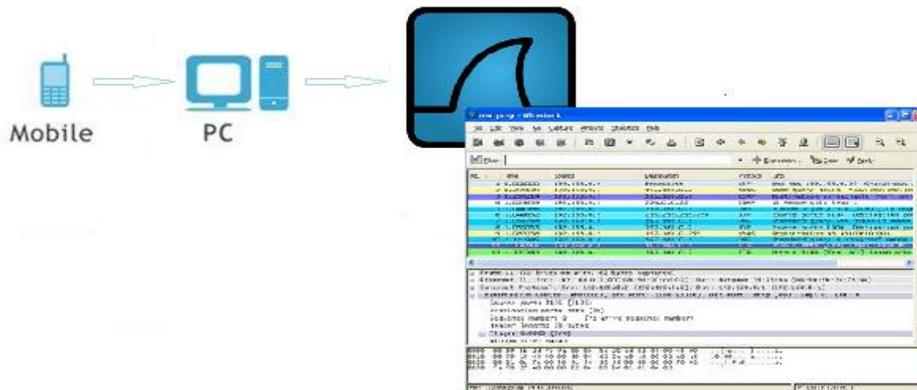


Figura 4.2: Exportación y análisis del tráfico capturado.

4.3 Diseño de la herramienta.

La nueva plataforma de Windows Phone 7 fue todo un reto para la realización de este trabajo y la falta de familiaridad siempre introduce algunos riesgos adicionales para el proyecto. En este sentido los primeros pasos fueron encaminados a entender las capacidades de la plataforma Windows Phone 7, para determinar una buena arquitectura y diseño de la aplicación a instalar en el cliente móvil, sin obviar las potencialidades de los nuevos elementos presentes en la nube.

A continuación se describirá detalladamente el proceso de diseño, así como todos los elementos tenidos en cuenta para ello.

4.3.1 Requisitos de la herramienta.

Windows Phone 7 ofrece toda una novedosa plataforma que brinda una gran cantidad de posibilidades para los desarrolladores y diseñadores. Se propone entonces, asegurarse que la herramienta hace el mejor uso posible de la plataforma y que el diseño de la misma coincida con las reglas establecidas. Unido a ello, se propone un comportamiento de "buen ciudadano" que permita que la aplicación haga un uso eficiente de los recursos en el dispositivo y no interfiera en otras funciones del dispositivo.

Se identificaron tres conjuntos de requerimientos para el diseño y desarrollo de la aplicación: los requerimientos de facilidad de uso, los requerimientos no funcionales y los requerimientos de desarrollo del proceso. A continuación se ampliará cada uno de ellos.

4.3.1.1 Requerimientos de facilidad de uso.

Los requerimientos de facilidad de uso han sido diseñados para asegurar que la experiencia del usuario con la herramienta, cumple con sus expectativas de aplicaciones para un dispositivo que utiliza Windows Phone. La siguiente tabla detalla los requerimientos de usabilidad.

Tabla 4.1: Requisitos de facilidad de uso.

| Requisito. | Descripción del requisito. |
|------------|---|
| | La aplicación debe utilizar: <ul style="list-style-type: none"> • Los gestos estándares de entrada para introducir |

| | |
|--|--|
| Ventajas de la apariencia y comportamiento. | <p>los datos y los colores estándar del sistema.</p> <ul style="list-style-type: none"> • Incluir iconos y gráficos diseñados para que coincida con el tema del teléfono. • Actualizar su aspecto para mezclarse con la luz normal del teléfono y los temas oscuros. |
| Controles estándares de Windows Phone. | La aplicación debe utilizar los controles estándares, incluyendo el Application Bar y los controles de Panorama, para que el usuario se sienta como en casa y minimizar la curva de aprendizaje. |
| Directrices de interfaz de usuario. | El botón Atrás debe desplazarse hacia atrás, de tal manera que coincida con las expectativas del usuario. La aplicación debe restaurar la interfaz de usuario a su estado anterior después que el usuario termina de contestar una llamada entrante. |
| Integración con los sensores en el teléfono. | La aplicación debe utilizar los servicios de localización en el teléfono para brindar su ubicación geográfica. |
| Cambios en la orientación de la pantalla. | La aplicación debe actualizar automáticamente la orientación de la pantalla cuando el usuario cambia la orientación del teléfono. |
| Resoluciones de pantalla estándares. | La aplicación debe mostrarse correctamente en las resoluciones de pantalla estándares para dispositivos Windows Phone. |

4.3.1.2 Requerimientos no funcionales.

Los requerimientos no funcionales describen los comportamientos esperados de la herramienta, incluyendo los comportamientos de "buen ciudadano" que se relacionan con la escasez de recursos en el dispositivo. La siguiente tabla detalla los requerimientos no funcionales.

Tabla 4.2: Requisitos no funcionales.

| Descripción del requisito. | Ejemplo. |
|---|---|
| La herramienta debe detectar cambio de interfaz de red. | La herramienta debe ser capaz de detectar automáticamente cambios en la interfaz de red, utilizando eventos para ello. |
| La herramienta debe brindar información de red. | <p>La herramienta debe ser capaz de ofrecer información de la interfaz de red, dentro de la que se puede citar:</p> <ul style="list-style-type: none"> • Operador. • RAT (Tecnología de acceso radio). • CID (identificación de células). • LAC (Código de Ubicación). • RSSI (Indicador de fuerza de la señal radio). • PSC (código primario de codificación). |
| La herramienta debe | La herramienta debe ser capaz de almacenar en un fichero |

| | |
|--|---|
| capturar todo el tráfico IP. | todo el tráfico IP, para que posteriormente dicho fichero pueda exportarse y analizarse. |
| La herramienta debe minimizar el uso de la batería. | Si el teléfono se está quedando sin recursos, se desactivará su solicitud. |
| La herramienta debe brindar pruebas de conectividad. | La herramienta debe mostrar resultados de conectividad mediante la utilización del Ping. |
| La herramienta debe brindar información de localización. | La herramienta debe mostrar resultados de la localización actual del dispositivo, utilizando para ello mapas y representación numérica de los resultados. |

4.3.1.3 Requerimientos de desarrollo del proceso.

Para la creación de la herramienta que se propone, también se identificaron una serie de requisitos que se relacionan con el propio proceso de desarrollo. La siguiente tabla detalla los requerimientos de desarrollo del proceso.

Tabla 4.3: Requisitos de desarrollo del proceso.

| Descripción del requisito. | Ejemplo. |
|---|--|
| La herramienta debe tener un código altamente comprobable. | La herramienta debe utilizar el patrón de diseño MVVM, en el cual, una de la ventaja más significativa para su uso, es la de hacer el código más comprobable. |
| La herramienta debe tener capacidad para soportar otras plataformas móviles en el futuro. | La herramienta debe hacer uso de estándares base, enfocados para interactuar, lo cual hace que sea más fácil desarrollar otros clientes para otras plataformas. |
| La herramienta debe tener un proceso de desarrollo eficiente. | La herramienta debe ser desarrollada con la utilización Silverlight, lo cual permitirá que los desarrolladores y diseñadores puedan trabajar en paralelo. Los diseñadores puedan crear prototipos y construir la interfaz de usuario mediante el software de diseño Microsoft Expression Blend, mientras que los desarrolladores se centren en la lógica de la aplicación. |
| Adaptación. | La herramienta debe adaptarse para trabajar con cualquiera de las nuevas capacidades de la plataforma Windows Phone en futuras versiones. |

4.3.2 Casos de Uso del Sistema.

El primer paso para definición de los casos de uso del sistema es definir los actores. Un actor representa el rol que asume una persona, sistema o entidad que interactúa con el software en construcción. En la herramienta que se diseña en este trabajo, el usuario sería el único actor del sistema.

Un caso de uso es una técnica muy usada para la captura de requisitos potenciales de un nuevo sistema. Estos suelen representarse en diagramas, visualizando el comportamiento de un sistema

mediante su interacción con los usuarios y/o otros sistemas. Para la implementación de esta herramienta se detectaron 5 casos de usos, estos son:

- Mostrar ayuda.
- Obtener información de la interfaz de red.
- Hacer Ping.
- Obtener información de localización.
- Capturar tráfico.

El apéndice A contiene la descripción detallada de los casos de uso anteriormente citados.

4.3.3 Patrón de diseño.

Si bien Silverlight es visto por muchos desarrolladores solamente como una tecnología para la web, en la práctica se ha convertido en una excelente plataforma para desarrollar cualquier tipo de aplicaciones. Al crear las aplicaciones Silverlight con el patrón de diseño Model-View-ViewModel (MVVM) se obtiene, además de las todas las características propias de Silverlight, ventajas adicionales como: facilidad de mantenimiento, mejor capacidad de pruebas y separación de la UI de la lógica subyacente.

Ahora bien, teniendo en cuenta lo anteriormente planteado se propone la utilización del patrón de diseño MVVM. La figura 4.3 muestra la distribución de clases por capas de la herramienta, aplicando el MVVM, téngase en cuenta que las relaciones entre los modelos y las vistas-modelos no se representan todas, sino que se incluyen los modelos dentro de un contenedor para que cada vista-modelo tenga acceso al modelo que necesite en cada momento.

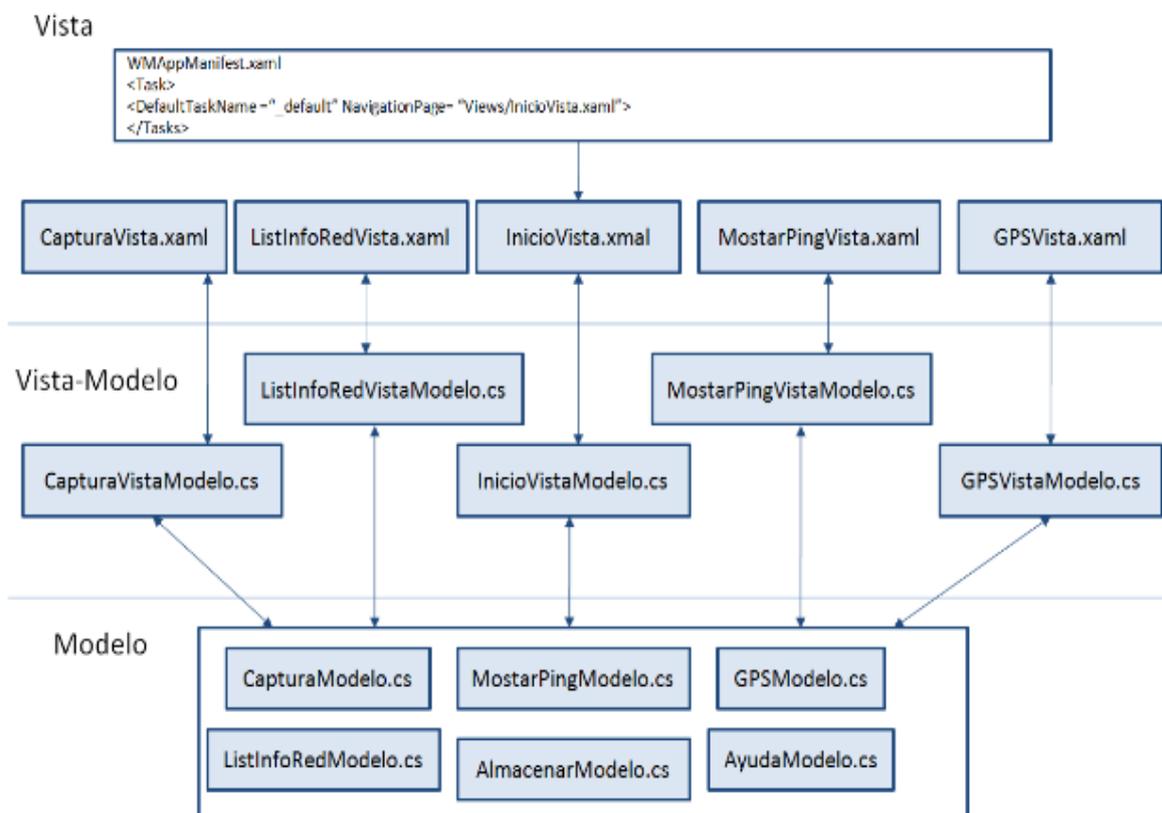


Figura 4.3: Estructura de clases por capas de la herramienta aplicando el MVVM.

Es importante resaltar que los modelos representan los datos o información con la que se trabaja. La clave para recordar el modelo es que contiene la información, pero no las acciones o servicios que la manipulan. No es responsable de darle forma para que se vea bien en la pantalla, o de obtener una lista de elementos de un servidor remoto.

Por su parte, la vista es la parte con la que comúnmente estamos más familiarizados y la que ve el usuario. Su papel es representar la información, tomándose a veces ciertas libertades con el fin de hacerla más clara o presentable. La vista se encarga de las teclas presionadas, los movimientos del ratón, los gestos en una pantalla táctil, entre otros, que eventualmente ejerce influencia en las propiedades del modelo.

Por último se encuentran la capa vista-modelo, nuestro controlador o presentador y pieza clave del trío. Esta capa introduce el concepto de separación de la presentación, es decir, mantiene al modelo separado y protegido de los minuciosos detalles de la vista. Esta separación permite que el modelo se limite a contener los datos, en vez de verse obligado a saber la forma en que se presenta y tener que hacer la conversión de formatos. De igual manera, la vista sólo tiene que presentar sus datos. El controlador trabaja como intermediario entre ambos, recibiendo información de la vista e insertándola en el modelo, o actuando con un servidor para obtener datos del modelo y luego convertirlos en propiedades que pueden ser usadas en la vista.

De forma complementaria a lo anteriormente planteado, se sugiere una estructura de carpeta para la organización del código y los recursos, similar a la que se propone a continuación:

- **Root:** En la carpeta raíz del proyecto, se encuentra el archivo `App.xaml` que cada proyecto Microsoft Silverlight debe incluir, este define algunos de los comportamientos de inicio de la aplicación. La carpeta raíz también contiene algunos archivos de imagen que todas las aplicaciones de Windows Phone 7 deben incluir.
- **Properties:** En esta carpeta, se encuentra el archivo `AssemblyInfo.cs` que contiene la configuración del proyecto que se usa al compilar la aplicación.
- **Resources:** Esta carpeta contiene varios archivos de imagen que utiliza la aplicación y algunas clases que realizan conversiones de tipos, que se usan en la interfaz de usuario.
- **Views:** Esta carpeta contiene los archivos XAML que definen las pantallas de la aplicación.
- **ViewModels:** Esta carpeta contiene los archivos de C# que implementan los modelos de vista.
- **Models:** Esta carpeta contiene los archivos de C# que implementan los modelos.
- **Themes:** Los archivos XAML en esta carpeta contienen las definiciones de estilo.

A continuación se mostrarán algunos de los diseños de interfaz de usuarios propuestos para el desarrollo de la herramienta.

4.3.4 Interfaz de usuario.

El diseño de interfaces de usuario es un campo sumamente amplio, de hecho, se debería contar con personas dedicadas y preparadas para realizar exclusivamente esta tarea.

Pese a ello esta responsabilidad a menudo recae sobre los propios programadores y nos damos a la tarea de armar la interfaz visual para el usuario.

Para el diseño de las diferentes interfaces de usuarios, se tuvo en cuenta básicamente los requisitos de facilidad de uso. Además se propone ambientar la vista inicial, de forma que sea atractiva para el usuario.

A continuación se muestran varias imágenes que ilustran la interfaz de usuario propuesta.



Figura 4.4: Pantalla de inicio.

En la pantalla de inicio se propone utilizar el componente Application Bar, el cual contendrá dos botones, el primero permitirá acceder a la ayuda, la cual contendrá información referente al manejo de la herramienta y el segundo permitirá salir de la herramienta. Unido a ello, al desplegar se podrán encontrar todas las funcionalidades que brinda la aplicación, (ver figura 4.4).

La figura 4.5 expone el diseño de interfaz de usuario propuesto para satisfacer la funcionalidad de hacer Ping. En la misma es posible observar la utilización de teclado de software para la introducción de la dirección IP y el mantenimiento de la barra de aplicaciones con las funcionalidades antes descritas.



Figura 4.5: Interfaz para hacer PING.

Otra de las interfaces a mostrar es la que brinda el servicio de geolocalización. La propuesta para ello se observa en la figura 4.6. Resulta interesante señalar que el usuario no solo tendrá acceso a sus coordenadas (longitud y latitud), sino que también podrá obtener un mapa con su posición actual.



Figura 4.6: Interfaz para Geolocalización.



Figura 4.7: Interfaz de información. de red.

Por su parte la figura 4.7 muestra la propuesta de interfaz de usuario que permitirá obtener información de la interfaz de red utilizada. Nótese que dicha información puede actualizarse cada vez que el usuario lo desee.

Por último la figura 4.8 muestra la interfaz de usuario propuesta para la captura del tráfico IP. Nótese que esta permitirá iniciar, pausar, detener y exportar la captura. Además de permitir que nuestra herramienta se ejecute en segundo plano al seleccionar la opción de minimizar.



Figura 4.8: Interfaz para captura de tráfico IP.

Para lograr que las interfaces de usuarios antes mencionadas no se distorsionen una vez el usuario cambie la posición de su teléfono, se propone la utilización de canvas. Ello

permitirá cuadrricular la parte editable de la pantalla del dispositivo, lo cual sería ventajoso para mantener una interfaz de usuario lo más organizada posible, donde no se superpongan elementos.

Lógicamente el diseño de la herramienta que se propone se regirá por la aplicación de patrones de diseño, los cuales establecerán la estructura interna del proyecto, así como la interacción entre sus distintas partes.

Una vez definidos tanto las interfaces de usuario como el patrón de diseño a utilizar, es momento de definir todo el conjunto de espacios de nombres, clases, métodos y propiedades que permitirán obtener las funcionalidades antes descritas.

4.4 Analizando la viabilidad.

En este epígrafe se llevara a cabo una descripción detallada de los espacios de nombres, clases, métodos y propiedades que se proponen para el logro de las funcionalidades previstas. Es importante señalar que aunque algunas de las propuestas que tienen lugar en este epígrafe, en la actualidad no son posibles de implementar, se espera que en próximas actualizaciones de Windows Phone ya tengan soporte.

4.4.1 Obtención de información de la interfaz de red.

Con el objetivo de brindar información acerca de la interfaz de red, se propone la utilización de la clase `NetworkInterfaceInfo` del espacio de nombres `Microsoft.Phone.Net.NetworkInformation`, la cual contiene toda la información disponible de una interfaz de red. Dicha clase engloba un conjunto de métodos y propiedades para el desarrollo de sus funcionalidades, para más información remitirse a:

<http://msdn.microsoft.com/es-ar/library/microsoft.phone.net.networkinformation.networkinterfaceinfo%28v=VS.92%29.aspx>.

A continuación se exponen las propiedades disponibles en dicha clase:

- `NetworkInterfaceInfo.Bandwidth`: obtiene la velocidad de la interfaz de red, la velocidad se expresa en kilobits por segundo.
- `NetworkInterfaceInfo.Characteristics`: obtiene las características de la interfaz de red. Esto incluye información de si la red permite roaming o no.
- `NetworkInterfaceInfo.Description`: obtiene una descripción de la interfaz de red.
- `NetworkInterfaceInfo.InterfaceName`: obtiene el nombre de la interfaz de red.
- `NetworkInterfaceInfo.InterfaceState`: obtiene el estado de la conexión de la interfaz de red, especificando cuando está conectada o desconectada.
- `NetworkInterfaceInfo.InterfaceType`: obtiene el tipo de la interfaz, incluye información referente a si la interfaz de red utiliza una red celular GSM, una red CDMA, u otro tipo de red.
- `NetworkInterfaceInfo.InterfaceSubtype`: obtiene información adicional sobre el tipo de interfaz de red. Esto incluye detalles sobre la tecnología de red de la interfaz de red, tales como la 3G celular.

La clase `NetworkInterfaceInfo` también brinda algunos tipos enumerados, los cuales especifican el rango de valores alcanzables por cada variable. Una variable de tipo enumerado proporciona una solución eficiente para definir un conjunto de constantes con nombre integral, que puede ser asignado a una variable. Por ejemplo, supongamos que se debe definir una variable cuyo valor represente los días de la semana. Dicha variable solo tendría siete valores significativos para almacenar y se definiría de tipo `enum` por ende. A continuación se citan alguna a variables de tipo `enum` que ofrece la clase `NetworkInterfaceInfo`:

- `NetworkInterfaceType`: especifica los tipos de interfaz de red, para más información:
<http://msdn.microsoft.com/en-us/library/microsoft.phone.net.networkinformation.networkinterfacetype%28v=VS.92%29.aspx>.
- `NetworkInterfaceSubType`: especifica la información adicional sobre el tipo de interfaz de red. El valor predeterminado es desconocido, para más información:
<http://msdn.microsoft.com/en-us/library/microsoft.phone.net.networkinformation.networkinterfacesubtype%28v=VS.92%29.aspx>.
- `NetworkSelectionCharacteristics`: especifica las características de interfaz de red preferida, para más información: <http://msdn.microsoft.com/en-us/library/microsoft.phone.net.networkinformation.networkselectioncharacteristics%28v=VS.92%29.aspx>.

Conjuntamente con lo anterior se propone la utilización de la clase `IPAddress` del espacio de nombres `System.Net`, la cual contiene la dirección del equipo en una red IP y se puede utilizar para representar una dirección IPv4 o IPv6, para más información dirigirse a: <http://msdn.microsoft.com/es-es/library/s128tyf6%28v=VS.95%29.aspx>.

4.4.2 Obteniendo el tráfico IP.

Antes de la obtención del tráfico IP es necesario tener en cuenta como llevar a cabo el almacenamiento del mismo, para ello se propone la utilización del espacio de nombres `System.IO`, el cual contiene tipos que permiten leer y escribir en los archivos y secuencias de datos, así como tipos que proporcionan compatibilidad básica con los archivos y directorios. Específicamente se propone la utilización de la clase `File`, la cual proporciona métodos estáticos para crear, copiar, eliminar, mover y abrir archivos. Además contribuye a la creación de objetos de la clase `FileStream`, para más información dirigirse a:

<http://msdn.microsoft.com/es-es/library/system.io.file%28v=VS.95%29.aspx>

Por su parte la clase `FileStream` permitirá leer, escribir, abrir y cerrar archivos en un sistema de archivos, así como manipular otros identificadores del sistema operativo relacionados con archivos, incluidos los de canalizaciones, entrada estándar y salida estándar. Permite además especificar que las operaciones de lectura y escritura sean sincrónicas o asincrónicas. Es importante resaltar que `FileStream` almacena en el búfer la entrada y la salida para obtener un mejor rendimiento.

Otra opción para el acceso al almacenamiento local es la utilización de las clases `IsolatedStorageFile` y `IsolatedStorageFileStream`, del espacio de nombres `System.IO.IsolatedStorage` el cual contiene tipos para crear y utilizar un sistema de

archivos virtual. La clase `IsolatedStorageFileStream` permite leer, escribir y crear archivos en el almacenamiento aislado. Las aplicaciones pueden utilizar el almacenamiento aislado para guardar los datos existentes en la zona aislada del sistema de archivos, sin tener que especificar una ruta en particular dentro del sistema de archivos.

Para la captura de todo el tráfico IP se propone la utilización de la clase `IPGlobalStatistics` del espacio de nombres `System.Net.NetworkInformation`, dicha clase es utilizada por los métodos `GetIPv4GlobalStatistics` y `GetIPv6GlobalStatistics` para devolver información de tráfico IP. Téngase en cuenta que tanto IPv4 como IPv6 son protocolos de la capa de red, la cual es responsable de enrutar paquetes de datos desde su origen hasta su destino. También sería importante resaltar que el objeto devuelto por estos métodos refleja las estadísticas a partir de la hora que se crea el objeto `IPGlobalStatistics`.

Unido a lo planteado anteriormente, se propone el empleo de algunas de las propiedades de la clase `IPGlobalStatistics` para mostrar algunas estadísticas del tráfico obtenido, estas son:

- `OutputPacketsDiscarded`: obtiene el número de paquetes del protocolo de Internet (IP) transmitidos que se han descartado.
- `OutputPacketsWithNoRoute`: obtiene el número de paquetes (IP) para los que el equipo local no pudo determinar una ruta a la dirección de destino.
- `ReceivedPackets`: obtiene el número de paquetes (IP) recibidos.
- `ReceivedPacketsDelivered`: obtiene el número de paquetes (IP) entregados.
- `ReceivedPacketsDiscarded`: obtiene el número de paquetes (IP) que han sido recibidos y descartados.
- `ReceivedPacketsForwarded`: obtiene el número de paquetes (IP) reenviados.

Sería importante señalar que aunque todas las clases y propiedades mencionadas en este epígrafe están disponibles en el framework de .NET, en la actualidad la clase `IPGlobalStatistics` no está soportada por Windows Phone. Ello no permite en estos momentos la obtención de todas las funcionalidades que ella brinda, lo que no descarta que en posteriores actualizaciones del SO pueda utilizarse.

4.4.3 Servicios de geolocalización.

Microsoft Windows Phone 7 incluye una API de ubicación geográfica fácil de usar, que le permite determinar la posición actual y el movimiento de un usuario, expresados en latitud y longitud (y a veces también en altitud). Una vez que obtiene acceso a estos datos, está preparado para crear características de reconocimiento de ubicación en la aplicación de Windows Phone 7. Téngase en cuenta que los dispositivos que soportan la plataforma WP7 incluyen hardware para posibilitar el acceso al servicio de GPS.

El espacio de nombre `System.Device.Location` proporciona acceso a las APIs de Windows Phone `ServiceLocation`, permitiendo el desarrollo de aplicaciones de localización. Este espacio de nombre contiene un conjunto de clases e interfaces muy útiles. Dentro de las interfaces se recomienda el uso de `IGeoPositionWatcher<T>`, dicha interfaz permite la resolución de las posiciones geográficas, implementado por la clase `GeoCoordinateWatcher`.

Para la obtención de la localización se propone la utilización de la clase `GeoCoordinateWatcher` de las APIs de Windows Phone 7 para recuperar la ubicación actual del teléfono. La clase `GeoCoordinateWatcher` utiliza el evento `PositionChanged` que indica que la latitud o longitud de la ubicación del teléfono ha cambiado, además utiliza el evento `StatusChanged` para notificar a la aplicación cuando se tiene una solución disponible en la ubicación del teléfono.

La clase `GeoCoordinateWatcher` contiene además un conjunto de propiedades y métodos para obtener otras funcionalidades orientadas al servicio de geolocalización, para más información dirigirse a:

<http://msdn.microsoft.com/en-us/library/system.device.location.geocoordinatewatcher.aspx>

Para la visualización de dichos resultados se propone la utilización de `Bing Silverlight Control`, el cual combina el poder de `Silverlight` y `Bing Maps` para proporcionar una experiencia mejorada de mapeo. Los desarrolladores pueden utilizar los mapas de `Bing Silverlight Control` para incorporar las últimas funciones de localización y búsqueda local en sus aplicaciones, para más información dirigirse a:

<http://msdn.microsoft.com/en-us/library/ee681884.aspx>

4.4.4 Ping.

El ping es una herramienta sumamente importante incorporada en casi todos los sistemas operativos, la cual permite conectarse a una red local o a internet. La misma permite determinar si hay conectividad entre dos nodos. La topología de red puede determinar el éxito de hacer ping para contactar a un host remoto. La presencia y la configuración de proxy, de `Network Address Translation (NAT)` o servidores de seguridad pueden evitar que ping tenga éxito.

La herramienta que se diseña en este trabajo, permite conocer si un equipo determinado es accesible a través de la red, partiendo de conocer su dirección IP. Para el logro de ello, se propone la utilización de la clase `Ping` del espacio de nombres `System.Net.NetworkInformation`. Esta clase proporciona una funcionalidad similar a la herramienta de línea de comandos `Ping.exe`. El método `Send (IPAddress)` intenta enviar un mensaje de eco `Internet Control Message Protocol (ICMP)` para el equipo que tiene la dirección IP especificada y recibir el correspondiente mensaje de respuesta de eco `ICMP` de ese equipo.

Además de todas las bondades anteriormente expuestas, la clase `Ping` contiene un conjunto de métodos, eventos y funcionalidades que se pueden utilizar para obtener otros resultados, para más información dirigirse a:

<http://msdn.microsoft.com/en-us/library/system.net.networkinformation.ping.aspx>.

Sería importante señalar que aunque todas las clases, métodos y propiedades mencionadas en este epígrafe están disponibles en el framework de `.NET`, en la actualidad la clase `Ping` no está soportada por `Windows Phone`. Ello no permite en estos momentos la obtención de todas las funcionalidades que ella brinda, lo que no descarta que en posteriores actualizaciones del SO pueda utilizarse.

CAPÍTULO 5: CONCLUSIONES

La investigación desarrollada permitió detectar la importancia de la captura y análisis del tráfico que circula a través de un teléfono móvil. Unido a ello el estudio realizado sobre la plataforma de desarrollo de Windows Phone permitió adquirir una idea general de todas las potencialidades de dicha plataforma y cuán lejos se puede llegar actualmente en la misma.

Inmersos en el proceso de diseño, se explotaron las ventajas que brindan las herramientas, lenguajes y metodologías escogidas.

Sería importante resaltar que en la actualidad, no es viable el desarrollo de una herramienta que permita capturar tramas IP desde el propio dispositivo, en teléfonos móviles que utilicen Windows Phone como sistema operativo, debido a que muchas de las clases, métodos y propiedades que brinda .NET para ello, aún no están soportadas en Windows Phone. Unido a ello tampoco es viable el desarrollo de otras funcionalidades que permitirán obtener información de la celda, hacer ping a otros equipos presentes en la red, por motivos similares. Aunque se tiene la seguridad, que en posteriores versiones del sistema operativo Windows Phone, todos los elementos antes mencionados estén soportados.

En la actualidad, del conjunto de funcionalidades propuestas inicialmente, solo son viables para implementar, las de obtención de información de la interfaz de red y la de localización. Ello hace que otras herramientas desarrolladas para otros sistemas operativos para dispositivos móviles (véase epígrafe 2.8), puedan ofrecer mayores utilidades.

Sería significativo destacar que durante el proceso de investigación se detectaron un conjunto de funcionalidades y servicios, las cuales se pretenden incluir en posteriores versiones de la herramienta. Dentro de los que se pueden citar:

1. El monitoreo de baterías.
2. Obtención de información de celdas vecinas.
3. Monitorizar el uso de la memoria RAM.
4. Obtención de información sobre las capacidades de red del teléfono del usuario.
5. Utilización de la plataforma Windows Azure (servicios en la nube).

Propiciando un seguimiento en la investigación, así como la aparición de nuevas versiones de la herramienta, que permitan al usuario adueñarse de un producto mucho más completo.

La utilización de esta herramienta posibilitará un mayor control sobre toda la información referente al tráfico existente y protocolos empleados, garantizando una mayor seguridad en los sistemas, un empleo óptimo de los protocolos y un uso eficiente de los recursos existente en el dispositivo móvil. La solución propuesta influirá en el aumento de la producción de software para dispositivos móviles, pues contribuirá a disminuir en gran medida el tiempo, trabajo y recursos destinados al desarrollo de herramientas de monitoreo de tráfico.

BIBLIOGRAFÍA

1. *Comunicaciones en redes inalámbricas*. **R.J, Bates**. New York : s.n., 1994.
2. **Directiva 2002/22/CE del Parlamento Europeo y del Consejo**. EUR-Lex. *EUR-Lex*. [En línea] 24 de Marzo de 2002. [Citado el: 12 de Octubre de 2011.] <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0022:es:NOT>.
3. **Gutiérrez, César**. Tendencia de las Telecomunicaciones. [En línea] 21 de Octubre de 2008. http://www.tendencias21.net/La-telefonía-móvil-se-convierte-en-objetivo-hacker_a2653.html.
4. **THOMAS PLEVYAK, IEEE Communications Society**. *NEXT GENERATION TELECOMMUNICATIONS NETWORKS, SERVICES, AND MANAGEMENT*. s.l. : JOHN WILEY & SONS, 2010.
5. **ORTA, MAURICIO JUAREZ**. *ANÁLISIS Y DISEÑO DE UNA RED TELEFÓNICA PARA LA UNAM*. Ciudad México : s.n., 2007.
6. Wireshark. [En línea] [Citado el: 3 de diciembre de 2011.] <http://www.wireshark.org/>.
7. **Almudena Díaz, Pedro Merino, F. Javier Rivas Tocado**. *SymPA: Un Analizador de Protocolos para Dispositivos*. Málaga : s.n., 2010.
8. **Alvarez, Andrés, y otros**. *Field measurements of mobile services with Android smartphones*. Málaga : s.n., 2011.
9. **Lecrenski, Nick, Watson, Karli y Fonseca, Robert**. *Beginning Windows Phone 7 Application Development*. Indianapolis : Wiley Publishing, 2011.
10. **Julián, Josué Yeray**. *Windows Phone 7.5 "Mango" Desarrollo Silverlight*. 2011.
11. **Microsoft**. Microsoft Download Center. [En línea] [Citado el: 3 de 12 de 2011.] <http://www.microsoft.com/downloads/es-es/details.aspx?FamilyID=04704acf-a63a-4f97-952c-8b51b34b00ce>.
12. **Randolph, Nick y Fairbairn, Christopher**. *Professional Windows Phone 7 Application Development*. Indianapolis : Wiley Publishing, Inc, 2011.
13. **Pentikousis, Kostas, y otros**. *Mobile Networks and Management*. Atenas : Springer, 2009.
14. **Lee, Henry y Chuvyrov, Eugene**. *Beginning Windows Phone 7 Development*. s.l. : Apress, 2010.
15. **McLaughlin, Jeff y LeDonne, Beverly**. *The Consumer's Guide to Wireless Phones and Service Plans*. s.l. : Syngress Publishing, 2002.
16. **Iniewski, Krzysztof**. *CONVERGENCE OF MOBILE AND STATIONARY NEXT-GENERATION NETWORKS*. New Jersey : JOHN WILEY & SONS, 2010.
17. **Betts, Dominic, y otros**. *Windows phone 7 developer guide*. 2010.
18. **Curwen, Peter**. *The Future of Mobile Communications*. 2002.

19. **Grant, August E. y Meadows, Jennifer H.** *Communication Technology Update and Fundamentals*. 2010.
20. *Plataforma Windows Azure para empresas*. **Kommalapati, Hanu**. Febrero, 2010.
21. **Kupper, Axel.** *Location-based Services Fundamentals and Operation*. Munich : John Wiley & Sons Ltd, 2005.
22. *INTERNET Y LA TELEFONÍA MÓVIL NUEVOS SOPORTES PARA DISTRIBUIR CONTENIDOS AUDIOVISUALES*. **Ruano, Soledad**. 68, Monterrey, México : s.n., 2009.
23. **Microsoft.** MSDN Library. [En línea] [Citado el: 2 de diciembre de 2011.] <http://msdn.microsoft.com/en-us/library/ms123401.aspx>.
24. *Experimental analysis of peer-to-peer streaming in cellular networks*. **Díaz, Almudena, Merino, Pedro y Panizo, Laura**. Málaga : s.n., 2007.
25. **Díaz, Almudena, Merino, Pedro y Rivas, F. Javier.** *Mobile Application Profiling for Connected Mobile Devices*. Málaga : s.n., 2010.
26. **González, Martín Espinoza.** *Análisis de tráfico en sistemas inalámbricos*. 2008.
27. **Petzold, Charles.** *Programming Windows Phone 7*. Washington : s.n., 2010.



APÉNDICE A: DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA (CUS).

Tabla A.1: Descripción del CUS Mostrar ayuda.

| | | |
|---|---|--|
| Caso de Uso: | Mostrar ayuda. | |
| Actores: | Usuario. | |
| Resumen: | El CUS se inicia cuando el usuario decide ver la ayuda de la herramienta. | |
| Flujo Normal de Eventos | | |
| Sección "General" | | |
| Acción del Actor | Respuesta del Sistema | |
| 1- El usuario se dirige al Application bar y selecciona la opción de mostrar ayuda. | 1.1- El sistema muestra la ayuda, ayudándose para ellos de controles panoramas. | |

Tabla A.2: Descripción del CUS obtener información de la interfaz de red.

| | | |
|--|--|--|
| Caso de Uso: | Obtener información de la interfaz de red. | |
| Actores: | Usuario. | |
| Resumen: | El CUS se inicia cuando el usuario decide obtener información de la interfaz de red. | |
| Flujo Normal de Eventos | | |
| Sección "General" | | |
| Acción del Actor | Respuesta del Sistema | |
| 1- El usuario se dirige al Application bar y selecciona la opción de información de red. | 1.1- El sistema muestra la información de red obtenida. | |
| 2- El usuario puede seleccionar la opción de Actualizar. | 2.1- El sistema actualiza y muestra la información de red obtenida. | |

Tabla A.3: Descripción del CUS hacer ping.

| | | |
|---|--|--|
| Caso de Uso: | Hacer Ping. | |
| Actores: | Usuario. | |
| Resumen: | El CUS se inicia cuando el usuario decide hacer ping. | |
| Flujo Normal de Eventos | | |
| Sección "General" | | |
| Acción del Actor | Respuesta del Sistema | |
| 1- El usuario se dirige al Application bar y selecciona la opción ping. | 1.1- El sistema solicita la dirección IP. | |
| 2- El usuario puede seleccionar la opción Hacer Ping. | 2.1- El sistema muestra la información obtenida, referente a conectividad. | |

Tabla A.4: Descripción del CUS obtener información de localización.

| | | |
|--|---|--|
| Caso de Uso: | Obtener información de localización. | |
| Actores: | Usuario. | |
| Resumen: | El CUS se inicia cuando el usuario decide obtener información de localización. | |
| Flujo Normal de Eventos | | |
| Sección "General" | | |
| Acción del Actor | Respuesta del Sistema | |
| 1- El usuario se dirige al Application bar y selecciona la opción geolocalización. | 1.1- El sistema muestra las siguientes opciones: a) Calcular. b) Mostrar Mapas. | |
| 2- El usuario puede seleccionar una de las dos opciones. | 2.1. a - Si selecciona calcular, el sistema muestra información de la latitud y la longitud. 2.1. b - Si selecciona Mostrar Mapas, el sistema muestra un mapa con su ubicación geográfica. | |

Tabla A.5: Descripción del CUS capturar tráfico.

| | | |
|---|---|--|
| Caso de Uso: | Capturar tráfico. | |
| Actores: | Usuario. | |
| Resumen: | El CUS se inicia cuando el usuario decide capturar el tráfico IP generado en su dispositivo móvil. | |
| Flujo Normal de Eventos | | |
| Sección "General" | | |
| Acción del Actor | Respuesta del Sistema | |
| 1- El usuario se dirige al Application bar y selecciona la opción capturar tráfico. | 1.1- El sistema solicita el nombre del fichero donde se almacenaran los datos y si desea adicionar o no la fecha actual al nombre de dicho fichero, unido a | |

| | |
|---|--|
| | <p>ello el sistema muestra las siguientes opciones:</p> <ul style="list-style-type: none">a) Iniciar Captura.b) Minimizar.c) Pausar.d) Detener Captura.e) Exportar. |
| <p>2- El usuario puede seleccionar una de las opciones.</p> | <p>2.1. a - Si selecciona <i>Iniciar Captura</i>, el sistema comienza a almacenar en un fichero las trazas del tráfico IP.</p> <p>2.1. b - Si selecciona <i>Minimizar</i>, el sistema se ejecutará en un segundo plano.</p> <p>2.1. c - Si selecciona <i>Pausar</i>, el sistema detendrá momentáneamente la captura de las tramas IP.</p> <p>2.1. d - Si selecciona <i>Detener Captura</i>, el sistema detendrá la captura de las tramas IP.</p> <p>2.1. e - Si selecciona <i>Exportar</i>, el sistema proveerá las interfaces necesarias para exportar el fichero con las trazas.</p> |