

**UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
UNAN-MANAGUA
RECINTO UNIVERSITARIO RUBEN DARIO
FACULTAD DE CIENCIAS E INGENIERIAS
DEPARTAMENTO DE TECNOLOGIA**



**SEMINARIO DE GRADUACION PARA OPTAR AL TITULO DE INGENIERO
ELECTRONICO**

Tema:

Desarrollo de aplicación de Geolocalización con teléfonos celulares con sistema operativo Android utilizando el simulador SDK 1.6 y Eclipse 3.7.

Integrantes:

Br. Guillermo Lenin Castellón Jiménez

Br. Roberto Carlos Velásquez Céspedes

Tutor:

MSc. Álvaro Segovia

Asesor Tecnológico:

MSc. Elim Campos

Managua, Julio de 2012

INDICE

1. INTRODUCCIÓN	3
2. ANTECEDENTES	5
3. JUSTIFICACIÓN	7
4. PLANTEAMIENTO DEL PROBLEMA	8
5. HIPÓTESIS	9
6. OBJETIVOS:	10
6.1. OBJETIVO GENERAL:	10
6.2. OBJETIVOS ESPECÍFICOS:	10
7. MARCO TEÓRICO	11
HARDWARE DE TELÉFONOS CON SISTEMA OPERATIVO ANDROID	11
7.1. PRIMER DISPOSITIVO CON ANDROID	11
7.2. DISPOSITIVOS ACTUALES CON ANDROID	11
7.3. MOTOROLA DROID 2 GLOBAL	12
7.4. TECNOLOGÍAS DE CONECTIVIDAD	15
7.4.1. GSM/EDGE	16
7.4.2. IDEN	16
7.4.3. CDMA	16
7.4.4. EV-DO	17
7.4.5. UMTS	17
7.4.6. BLUETOOTH	17
7.4.7. WIFI	18
7.4.8. TETHERING	18
7.4.9. WIMAX	18
8. SISTEMA OPERATIVO ANDROID	18
8.1. GENERALIDADES DEL SISTEMA OPERATIVO ANDROID	18
8.2. HERRAMIENTAS DE DESARROLLO DE APLICACIONES ANDROID	22
8.2.1. ECLIPSE 3.7	22
8.2.2. ESTRUCTURA DE UN PROYECTO ANDROID	24
8.2.3. CARPETA /SRC/	24
8.2.4. CARPETA /RES/	24

8.2.5. CARPETA /GEN/	26
8.2.6. CARPETA /ASSETS/	26
8.2.7. FICHERO ANDROIDMANIFEST.XML	26
8.3. COMPONENTES DE UNA APLICACIÓN ANDROID	27
8.3.1. ACTIVITY.....	27
8.3.2. VIEW	27
8.3.3. SERVICE	27
8.3.4. CONTENT PROVIDER	27
8.3.5. BROADCAST RECEIVER.....	27
8.3.6. WIDGET.....	28
8.3.7. INTENT	28
9. DESARROLLO	28
9.1. PREPARACIÓN:.....	28
9.2. CREACIÓN DEL DIAGRAMA DE FLUJO DE LA APLICACIÓN	34
9.3. PROGRAMACIÓN.....	36
9.4. REQUERIMIENTOS MÍNIMOS DE HARDWARE Y SOFTWARE PARA CORRER LA APLICACIÓN “UBICACIONUNAN”	41
9.5. MANUAL DE UBICACIONUNAN	42
10. CONCLUSIONES	43
11. RECOMENDACIONES:	44
12. GLOSARIO DE TÉRMINOS	45
13. BIBLIOGRAFÍA:	48

1. Introducción

El desarrollo tecnológico que han tenido los celulares en los últimos años ha sido muy amplio, estos dispositivos han sido dotados de diferentes características funcionales, tales como la reproducción multimedia, la conexión a Internet mediante diferentes tecnologías, grabación de audio y video, incorporación de cámaras digitales, entre otras. Esto nos hace que nos planteemos que los celulares no siguen siendo solamente una herramienta para la comunicación sino que realmente el uso que se le da actualmente a estos dispositivos es muy amplio.

Estos avances en estas terminales móviles han obligado a que se desarrollen sistemas operativos para estos dispositivos móviles y puedan ser usados de manera más eficiente.

Android es un sistema operativo que está basado en Linux, es implementado en diferentes dispositivos móviles como teléfonos inteligentes (smartphones), tablets, GoogleTV, entre otros. El 5 de noviembre de 2007 se anunció la formación de la Open Handset Alliance (OHA) que decidió crear Android con el objetivo de desarrollar estándares abiertos para dispositivos móviles.

Android se ha desarrollado muy rápido desde su lanzamiento, disputándose actualmente los primeros lugares en presencia de teléfonos que utilicen el sistema operativo, junto a Windows Mobile, Apple iOS, BlackBerry OS y Symbian OS de Nokia. Estando presente en cerca de la tercera parte de los teléfonos móviles mundialmente.

El sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic.

El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

La mayor parte de teléfonos que utilizan Android poseen microprocesadores potentes que pueden realizar millones de operaciones por segundo, teniendo velocidades de los 500 Mhz en adelante teniendo incluso los dispositivos más avanzados velocidades más allá de 1GHz con doble núcleo.

Casi todos los modelos disponibles poseen conexión a Internet mediante Wi-Fi, además de conexión 3G para internet móvil de alta velocidad, así como también conexión mediante Bluetooth para compartir datos con otros dispositivos.

En varios modelos se incorporan también sensores de movimiento, localización mediante GPS, acelerómetros, entre otros; estas características abren la posibilidad para el desarrollo de infinidad de aplicaciones donde las limitaciones serían la imaginación y habilidad del programador, tomando en cuenta las prestaciones del teléfono a usar.

Cabe destacar que el teléfono celular se remonta a los inicios de la Segunda Guerra Mundial, donde ya se veía que era necesaria la comunicación a distancia, es por eso que la compañía Motorola creó un equipo llamado HandieTalkie H12-16, que es un equipo que permite el contacto con las tropas vía ondas de radio que en ese tiempo no superaban más de 600 Khz. Fue sólo cuestión de tiempo para que las dos tecnologías de Tesla y Marconi se unieran y dieran a la luz la comunicación mediante radio-telefonos, Los países europeos no se quedaron atrás e introdujeron en Escandinavia un sistema similar a AMPS (Advanced Mobile PhoneSystem). En 1985 se comenzaron a perfeccionar y amoldar las características de este nuevo sistema revolucionario ya que permitía comunicarse a distancia. Fue así que en los años 1980 se llegó a crear un equipo que ocupaba recursos similares a los HandieTalkie pero que iba destinado a personas que por lo general eran grandes empresarios y debían estar comunicados, es ahí donde se crea el teléfono móvil y marca un hito en la historia de los componentes inalámbricos ya que con este equipo podría hablar a cualquier hora y en cualquier lugar.

Estos teléfonos de ahora cuentan con mucha más capacidad de hardware, microprocesadores mayores, memorias RAM más grandes, características que anteriormente no poseían los teléfonos celulares. La tecnología sigue avanzando grandemente y el mundo de los celulares no se podía quedar atrás.

En resumen, hace una década aproximadamente los teléfonos celulares se caracterizaban sólo por llamar, pero ha sido tanta la evolución que ya podemos hablar de equipos Multimedia que puede llamar y ejecutar aplicaciones, jugar juegos 3D, reproducir vídeos, ver televisión y muchas cosas más. Debemos tener conciencia y prepararnos para lo que se viene más adelante y pensar que el telefonocelular ya no es tan sólo para hablar.

2. Antecedentes

La telefonía móvil inició en el siglo pasado durante la segunda guerra mundial y fue utilizada con fines militares y la transmisión analógica en AM con frecuencias que estaban en rangos cercanos a 600KHz, luego muchos tiempo después se empezó a darle uso comercial hasta principios de la década de 1980 cuando se lanzó el Motorola DynaTAC 8000X que ya trabajaba con FM pero la transmisión seguía siendo analógica, la tecnología usada en estos teléfonos fue principalmente la AMPS (Advanced Mobile Phone System) esta fue la primera generación de telefonía celular (1G) estas tecnologías no permitía la transmisión de otros tipos de datos aparte de la voz.

En los 90 se desarrolló la 2G que se basó principalmente en las tecnologías CDMA (Code Division Multiple Access) y GSM (Global System Mobile) ya se empiezan a usar mejores implementaciones en el transporte de los datos al hacer las transmisiones digitalmente lo que permite que se puedan acarrearse datos como mensajes de texto (SMS) y transmisión de datos a muy baja velocidad, otro avance importante que se debe recalcar es que ya se implementó protocolos para la encriptación de los datos cosa que no se había realizado antes. Esta fue la primera tecnología de telefonía móvil que se introdujo en nuestro país y la empresa de telecomunicaciones Bellsouth la que la implementó, a finales de los años 90.

Posteriormente se implementó el sistema 2.5G el cuya principal mejora fue el incorporar el GPRS (General Packet Radio Service) que permite la transmisión de datos a mayor velocidad y el envío de MMS (mensajes multimedia); en algunos equipos de esta generación se incluyó también EDGE (Enhanced Data Rates for GSM Evolution).

La 3G tiene aún una mejor capacidad para la transmisión de datos lo que posibilita el desarrollo de las aplicaciones multimedia en línea como por ejemplo las video llamadas, la tecnología que utiliza se llama UMTS (Universal Mobile Telecommunications System) esta tecnología se implementa en nuestro país pero solamente para el acarreo de datos para internet móvil por medio de módems USB es utilizada por ambas operadoras; no se utiliza en telefonía móvil debido a la infraestructura de red de nuestro país que es GSM por lo que UMTS utiliza redes CDMA.

Los teléfonos con Android aprovechan al máximo la conectividad a Internet por medio de sus diferentes aplicaciones, siendo algunas tan sencillas como mantenerse informado del clima de nuestro país hasta aplicaciones utilizando GPS y Google Maps para ubicar direcciones concretas entre otras.

En la actualidad las aplicaciones para teléfonos móviles se han vuelto muy importantes para los usuarios de estos dispositivos sobre todo para acortar distancias utilizando los distintos tipos de conexión a internet ya sea por Wifi o por internet móvil de alta velocidad (GPRS,EDGE).

En 2002 Enitel introduce la tecnología GSM 100 digital con servicio de voz y SMS, luego en 2004 Enitel lanza el GPRS, que permitía a sus usuarios acceder a la navegación WEB, WAP y mensajes de multimedia (SMS). La velocidad de transferencia de datos era de 56 Kbbps. Durante el año 2006 Enitel mejora el GPRS al sustituirlo por la red EDGE. Con esta red la velocidad para transmitir los datos llega hasta los 384 Kbbps y en 2008 Claro Enitel anuncia que Nicaragua cuenta ya con tecnología 3G.

Resaltamos que en nuestra universidad se han realizado estudios previos sobre telefonía convencional como redes PBX, pero no se han realizado investigaciones ni desarrollo de proyectos sobre telefonía móvil, esto nos ha motivado grandemente a llevar a cabo este proyecto.

3. Justificación

El sistema operativo Android actualmente lleva un crecimiento muy acelerado debido a la gran cantidad de teléfonos inteligentes de distintas marcas que han salido al mercado que traen instalado este sistema operativo, también tiene presencia en otros dispositivos como tablets, reproductores multimedia, entre otros.

Otro dato importante por señalar es la tienda Google Play donde uno tiene al alcance miles de aplicaciones para este sistema, en la misma web uno puede subir y compartir las aplicaciones que uno haya diseñado, pudiendo venderla teniendo un margen de ganancia del 70% del precio al que sea vendida dicha aplicación.

Nuestra idea principal es diseñar una aplicación que aproveche las características que traen dispositivos que trabaja con el sistema operativo Android que beneficiara directamente a la comunidad educativa de la UNAN-Managua por lo tanto teniendo en cuenta que nuestro recinto tiene una extensión muy amplia, saber nuestra ubicación exacta es en ocasiones algo complicado; y además tener la ubicación de diferentes puntos de referencia es muy importante para desplazarnos más eficientemente. Dado esta afirmación el poder ubicarse rápidamente y exactamente en nuestra universidad es muy importante así como poder tener la ubicación de los diferentes pabellones y auditorios esto nos permite ahorrar tiempo y esfuerzo. Por lo tanto nuestra idea es aprovechar las capacidades de los dispositivos con Android para realizar esto, por consiguiente diseñaremos una aplicación que permita obtener nuestra localización y también que nos muestre diferentes puntos de interés en nuestra universidad por medio de la visualización de un mapa.

4. Planteamiento del problema

El Recinto Universitario RubénDarío de la Unan-Managua ocupa un área bastante amplio por lo tanto esto provoca que ubicarse se dificulte y se vuelva un problema muy engorroso el encontrar el lugar hacia el cual nos dirigimos.

Siendo uno de los recintos universitarios mas grandes y extensos del país con un área de 110 manzanas o un equivalente a 768680 m², el ubicarse suele ser difícil, incluso para un alumno o profesor regular de la universidad puede volverse un problema. Cabe resaltar que la universidad no cuenta con mapas para ubicarse correctamente en los pabellones y utilizarlos como referencia, por lo tanto nosotros usaremos la tecnología brindada por los teléfonos inteligentes con Android para solucionar este problema de una manera sencilla, práctica y eficiente.

Los teléfonos inteligentes con Android poseen sistema de localización geográfica mediante GPS y al ser Android un sistema operativo desarrollado por Google uno tiene acceso a los mapas de Google como programador. Queremos crear una aplicación que utilice estas ventajas y permita obtener la ubicación del usuario y brindarle puntos de referencia para ubicarse correctamente en el recinto.

5. Hipótesis

Tomando en cuenta los problemas que hemos expuesto anteriormente, nuestro objetivo es brindar un aporte para solucionar este problema de manera sencilla y práctica.

La alternativa que hemos desarrollado consiste en programar una aplicación para teléfonos con sistema operativo Android que realizará las siguientes funciones:

- Desplegar un mapa en pantalla con una interfaz amigable y gráfica que sea sencilla de utilizar.
- Obtener la ubicación en tiempo real del usuario.
- Mostrar al usuario la ubicación de los pabellones mediante el uso de marcas gráficas en el mapa desplegado.

Teniendo en cuenta estas características la aplicación permitirá que se pueda dar una solución práctica y sencilla a los alumnos, profesores y demás visitantes que tenga problemas a la hora de ubicarse en el centro de estudios.

Es importante tomar en cuenta que para utilizar la aplicación el usuario no es necesario que tenga conocimientos amplios o grandes en cuanto a programación o informática pues deseamos que nuestra aplicación pueda ser utilizada por todo tipo de personas.

Un punto a favor de esta aplicación es que podría ser utilizada como mapa de referencia en cualquier otra ubicación, lo que permite que su uso no solo se limite a nuestro recinto universitario.

6. Objetivos:

6.1. Objetivo General:

Analizar los requisitos de hardware de los teléfonos celulares con Sistema Operativo Android y desarrollar una aplicación utilizando el simulador SDK 1.6 y eclipse 3.7.

6.2. Objetivos Específicos:

- Caracterizar el hardware del teléfono celular Motorola Droid 2 Global con sistema operativo Android.
- Describir de manera práctica y fácil la herramienta simuladora.
- Desarrollar una aplicación de geolocalización en las instalaciones del Recinto Universitario Rubén Darío de la UNAN- Managua para el sistema operativo Android.
- Redactar un Manual de Usuario para el uso adecuado de la aplicación.

7. Marco Teórico

Hardware de Teléfonos con sistema operativo Android

7.1. Primer dispositivo con Android

El primer dispositivo con sistema operativo Android fue el HTC Dream, salió a la venta a finales de 2008 para ser exacto el 22 de octubre de ese año. Como la mayoría de los teléfonos celulares con Android disponía de pantalla táctil con una resolución 480 por 320 píxeles, procesador Qualcomm ARM 11 de 528 Mhz, una memoria RAM(Random Access Memory) de 192 MB y ROM(ReadOnlyMemory) de 256 MB, diseño slider con teclado QWERTY(Teclado completo), cámara de 3.15MP (mega píxeles) con autofocus, geolocalización mediante GPS, acelerómetro y memoria expandible mediante microSD. Este teléfono traía el sistema Android 1.6 Donut de fábrica. En la figura 1 está el primer teléfono que tuvo Android como sistema operativo



el HTC Dream.

Fig. 1: HTC Dream primer teléfono con Android. Fuente: Pagina Web de HTC

7.2. Dispositivos Actuales con Android

Con el paso del tiempo se han ido diseñando nuevos teléfonos y otros dispositivos tales como Google TV, Tablets, reproductores multimedia y más recientemente en relojes de pulsera que traen el sistema operativo Android, cada vez mas empresas de desarrollo de equipos móviles están usando Android en sus productos. Cabe mencionar empresas como Motorola, HTC, SonyErickson, Samsung, Alcatel, LG, Acer, entre otros.

Actualmente unos de los teléfonos más avanzados en el mercado son los Samsung de la rama Galaxy S II, contando entre sus modelos los siguientes Samsung Galaxy S II, Epic 4G, Samsung Galaxy S II Skyrocket. En la figura 2 podemos apreciar un Samsung Galaxy S II.



Fig. 2: Samsung Galaxy S II, uno de los modelos actuales más avanzados. Fuente: Página Web de Samsung

Estos modelos traen todos procesadores de doble núcleo que van desde 1.2 GHz hasta 1.5 GHz, poseen cámaras traseras de 8 Megapíxeles y frontales de 2 Megapíxeles, ubicación con GPS, acelerómetro, pantalla táctil AMOLED, compatibilidad con redes CDMA, GSM, UMTS y 4G, lo que permite completa funcionalidad en varios países con diferentes tipos de conexiones, además de conexión con redes Wi-Fi para conectividad con internet de alta velocidad, estos teléfonos trabajan con el sistema operativo Android 2.3 denominado Gingerbread, pero pueden ser actualizados a la versión 3 HoneyComb o la versión más nueva la 4 Ice CreamSandwich.

7.3. Motorola Droid 2 Global

Hemos decidido trabajar con un teléfono Motorola Droid 2 Global el cual posee las siguientes características: tarjeta madre OMAP3640 con procesador ARM-Cortex A8 a 1.2 Ghz y GPU(GraphicsProcessorUnit) PowerVR SGX530, memoria RAM de 512 MB, memoria interna de 8 GB, soporte de hasta 32 GB para memorias expandibles microSD, cámara de 5 MP con autofocus, diseño slider teclado QWERTY, conexión WIFI, Bluetooth, soporte de 4 bandas GSM, UMTS, CDMA, geolocalización con GPS, acelerómetro, sensor de proximidad, sistema operativo Android2.3(Gingerbread). En la figura 3 mostramos el Motorola Droid 2 Global



Fig. 3: Motorola Droid 2 Global Fuente: Página Web de Motorola

- **Placa base:**

La placa base que posee este teléfono es de la tercera generación de la línea de tarjetas madres de dispositivos móviles de Texas Instruments llamada OMAP 3640 que trae consigo un procesador de 1.2 Ghz ARM-Cortex A8 y un GPU PowerVR SGX530 que permite tomar fotos de 5 MP y video en HD(High Definition), esta gama de tarjetas utiliza tecnología de semiconductores de 45 nm (nanómetros) y además también poseen el sistema IVA2 (Imaging, Video, Audio) para mejor procesamiento multimedia. En la figura 4 exponemos el diagrama de bloques de la placa base del Motorola Droid 2 Global, la OMAP 3640.

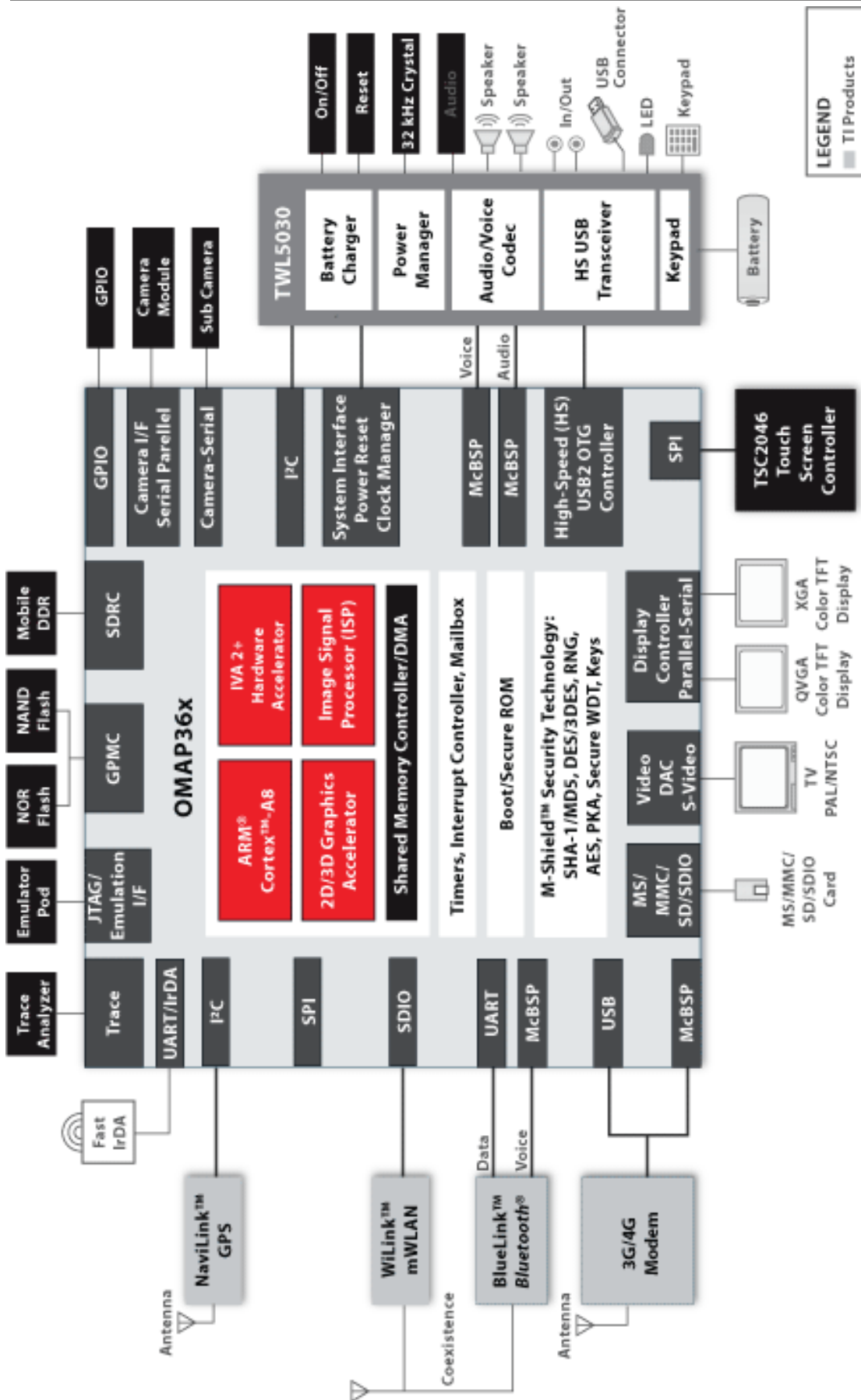


Fig. 4: Diagrama de bloques de la placa base OMAP 3640 para Motorola Droid 2 Global. Fuente: Página Web de Texas Instruments

- **Memoria RAM:**

La memoria RAM o memoria de acceso aleatorio permite que se carguen los recursos necesarios para correr una aplicación. El teléfono que utilizaremos trae una memoria de 512 MB lo que permite que se puedan correr varias aplicaciones a la vez.

- **Display:**

Este dispositivo cuenta con un display táctil con una resolución de tipo WVGA (Wide Video GraphicsArray) de 854 * 480 pixeles que utiliza tecnología TFT capacitiva este tipo de pantallas suelen traer una capa de oxido de indio y estaño que conduce corriente eléctrica a través del sensor, lo que genera un campo eléctrico en el sensor, que sirve como referencia así que cuando este campo es perturbado los sensores del display envían la información al controlador gráfico del dispositivo y este resuelve dicha señal transformándola en ordenes que entienda el sistema operativo del dispositivo, el campo eléctrico es perturbado por los dedos porque nuestro cuerpo también posee un campo eléctrico. Aquí abajo en la figura 5 podemos apreciar un pequeño diagrama que explica cómo funciona esta tecnología.

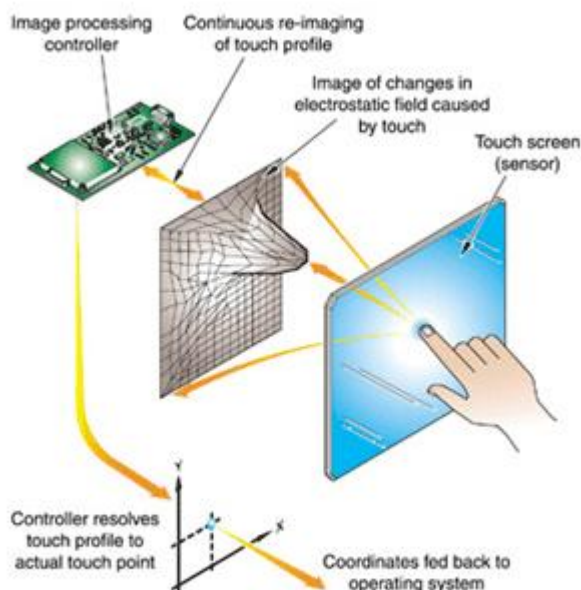


Fig. 5: Diagrama explicativo de cómo funciona un display capacitivo

- **Cámara:**

El Motorola Droid 2 Global cuenta con una cámara integrada de 5 MP y sistema Dual-Led Flash y grabación de video en HD a 720p. Trae los siguientes modos de imagen: “una toma”, “panorama”, “Autoretrato” y “Varias tomas”.

Se puede configurar para que tome fotos con resolución a 1 MP, a 2 MP, a 3MP y a 5 MP y para tomar video con calidad QVGA a 320 * 240 pixeles, CIF a 352 * 288 pixeles, VGA a 640 *480 pixeles y HD a 720 * 480 pixeles.

- **Batería:**

La batería que utiliza el móvil con el que estamos trabajando es un batería de Ion de Litio en Polímero, este tipo de batería funciona a partir de sales de litio que permite la reacción química reversible que lugar el flujo de electrones entre el ánodo y cátodo. Cada celda de este tipo de baterías proporciona 3.7 voltios. Son ideales para el uso en dispositivos móviles debido a su tamaño y peso reducidos, además de su bajo costo de producción.

- **Acelerómetro:**

El acelerómetro permite medir la aceleración y las fuerzas inducidas sobre el mismo, es decir que detecta el movimiento. Esto permite el desarrollo de ciertas aplicaciones que detectan el movimiento a través de este.

- **GPS:**

Es un sistema asistido por satélite que permite ubicar el dispositivo en cualquier parte del mundo, esta función nos será muy útil en nuestro proyecto pues permitirá que nos ubiquemos en el punto exacto en que estemos en la universidad.

7.4. Tecnologías de Conectividad

Las tecnologías móviles hoy en día son muy importantes para las empresas, ya que los mantiene conectados todo el tiempo, permite la transferencia inalámbrica de datos y así poder tener un mayor control y productividad.

Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, y WiMAX.

7.4.1. GSM/EDGE

Acrónimo para Enhanced Data rates for GSM Evolution, (Tasas de Datos Realizadas para la evolución de GSM) o también conocida como EGPRS (Enhanced GPRS), que fue la primera que hicimos hincapié anteriormente y podemos decir de ella que es una tecnología de la telefonía móvil celular, que actúa como puente entre las redes 2G y 2.5G. EDGE se considera una evolución del GPRS (General Packet Radio Service). Esta tecnología funciona con redes TDMA (La cual estas redes no son más que accesos múltiple por división de tiempo), es decir una denominación genérica para un tipo de tecnología de telefonía móvil digital, incluido el servicio GSM.

Cabe destacar que EDGE funciona con cualquier GSM que tenga implementado GPRS, donde el operador debe implementar las actualizaciones necesarias, además no todos los teléfonos móviles soportan esta tecnología. Es importante recalcar que también EDGE, o EGPRS, puede ser usado en cualquier transferencia de datos basada en conmutación por paquetes (PacketSwitched), como lo es la conexión a Internet. Los beneficios de EDGE sobre GPRS se pueden ver en las aplicaciones que requieren una velocidad de transferencia de datos, o ancho de banda alta, como video y otros servicios multimedia.

7.4.2. IDEN

IDEN Red Mejorada Digital Integrada (Integrated Digital Enhanced Network) es una tecnología inalámbrica desarrollada por Motorola en 1994, proporciona a los usuarios múltiples servicios en un único e integrado sistema de comunicaciones móviles.

Este sistema opera en bandas que van desde 800Mhz y 1500Ghz utilizando redes TDMA (Time Division Multiple Access), poseen 30 canales, 6 por cada usuario, cada canal utiliza 25 Khz y se modula mediante el formato M16-QAM de Motorola para sus transmisión y para sus digitalización VSELP.

7.4.3. CDMA

Otra tecnología de conectividad que se utiliza en Android es CDMA (Code División Multiple Access) que es una técnica digital de acceso múltiple por división de códigos, usado en las comunicaciones móviles según el estándar US (IS 95) en el intervalo de frecuencias entre los 800 y los 1.900 MHz. Donde los sistemas IS-95 dividen el espectro en portadoras de 1.25 MHz. Cuando se trabaja en comunicación de datos se necesita repartir muchos usuarios en un solo medio de transmisión esto se logra mediante la multiplexación para evitar

la interferencia, en la tecnología CDMA la multiplexación se logra dando a cada dispositivo transmisor un código único de señalización, por lo tanto la conexión entre 2 dispositivos se logra mediante el uso del código propio de cada aparato.

7.4.4. EV-DO

Esta tecnología se basa en un estándar de telecomunicaciones para la transmisión inalámbrica de datos a través de redes de telefonía celular evolucionadas desde IS-95 (CDMAOne). Es un estándar del grupo 3GPP2 que pertenece a la familia CDMA2000 y ha sido adoptado por muchos proveedores a nivel mundial, sobre todo en el continente americano, particularmente por aquellos que ya contaban con redes IS-95/CDMAOne (en competencia con las redes GSM). Este sistema utiliza la multiplexación por CDMA y por TDMA que permite maximizar la cantidad de información que se puede transmitir. Esta tecnología se desarrollo con el fin de optimizar la transmisión de datos a través de redes móviles.

7.4.5. UMTS

La tecnología UMTS (Universal Mobile TelecommunicationsSystem) se comenzó a emplear en la llamada tercera generación de telefonía móvil, la cual permite disponer de banda ancha y transmitir un volumen de datos importante por la red. Esta red puede ser usada no solamente para dispositivos también puede usarse en computadoras que tenga tarjetas que permitan conexión con este servicio, dado que la red de datos es independiente de la red telefónica a pesar que utilicen el mismo medio para transmitirse, debido a que los datos son encaminados en dependencia de su señalización. Este tipo de red permite que se pueda utilizar el dispositivo móvil ya no solamente para la comunicación telefónica sino también para comunicación multimedia de alta calidad.

7.4.6. BLUETOOTH

Bluetooth es una tecnología usada para WPAN (Wireless Personal Area Network) para transmisión de datos y voz, por radiofrecuencia en un área corta utilizando la ISM (Industrial, Scientific and Medical) de 2.4 GHz. Han surgido varias versiones de esta tecnología siendo la más nueva la 4 que permite transmisión de datos a 24 Mbit/s. El Motorola Droid 2 Global posee la versión 2.1 EDR que permite la transmisión a 3 Mbit/s.

7.4.7. WIFI

De igual manera otra tecnología de conectividad ya muy conocida es Wi-Fi que se creó para ser utilizada en redes locales inalámbricas pero es frecuente que en la actualidad también se utiliza para acceder a Internet, Wi-Fi es una marca de la Wi-Fi Alliance (anteriormente la Wireless Ethernet Compatibility Alliance), que la organización comercial que prueba y certifica que los equipos cumplen los estándares IEEE 802.11x. Wi-Fi en pocas palabras es un conjunto de estándares para redes inalámbricas basado en las especificaciones IEEE 802.11.

7.4.8. TETHERING

Android también trae consigo una novedosa tecnología como lo es TETHERING, este servicio es un proceso de conexión de un dispositivo no-móvil con un dispositivo móvil (teléfono celular, PDA, Netbook) con el fin de brindar al dispositivo no-móvil acceso a Internet mediante una conexión tethered (enlazada). Dicho de una forma más simple, el tethering consiste en usar un dispositivo móvil con acceso a internet como módem para conectar otro dispositivo que no posee acceso directo a internet.

7.4.9. WIMAX

Dentro de este tipo de tecnología la última que mencionamos es la tecnología WIMAX (WorldwideInteroperabilityforMicrowave Access) que significa Interoperabilidad Mundial para Acceso por Microondas, es la marca que certifica que un producto está conforme con los estándares de acceso inalámbrico IEEE 802.16. Dichos estándares son los que permiten conexiones de velocidades similares al ADSL o al cablemódem, sin cables, y hasta una distancia de 50-60 km. Es de gran importancia mencionar que este nuevo estándar será compatible con otros anteriores, como el de Wi-Fi (IEEE 802.11) para llegar a obtener un mayor uso.

8. Sistema Operativo Android

8.1. Generalidades del Sistema Operativo Android

Hemos decidido utilizar Android por ser un sistema operativo que está basado en software libre bajo el kernel de Linux 2.6 y posee cerca de la mitad del mercado de dispositivos móviles por sistemas operativos en el mundo como se puede ver en la figura 6 más abajo, esto gracias a la gran cantidad de dispositivos móviles que lo traen integrado y a la disminución en los costos de estos, contando además con cerca de 250000 aplicaciones disponibles y

siempre desarrollándose nuevas. En la figura 6 podemos apreciar el gran auge que están teniendo los dispositivos que traen el sistema operativo Android.

Worldwide smartphone market, by operating system, by 2011 global sales according to Canalys			
Operating System	Shipments 2011 (millions)	Market share 2011	Annual growth
Android	237.7	48.8%	244%
iOS	93.1	19.1%	96%
Symbian	80.1	16.4%	-29.1%
BlackBerry	51.4	10.5%	5.0%
Bada	13.2	2.7%	183.1%
Windows Phone	6.8	1.4%	-43.3%
Others	5.4	1.1%	14.4%
Total	487.7	100%	62.7%
Source: Canalys (Feb 2011)			via: mobiThinking

Fig. 6: Ventas de dispositivos por sistema operativo en 2011 Fuente: Página Web mobiThinking

Esta plataforma es de código abierto, basado en Linux, permite el desarrollo de aplicaciones por terceros y su sistema operativo (Android) se gestiona mediante el lenguaje de programación java y los dispositivos móviles se controlan por medio de bibliotecas creadas adaptadas por Google y su sistema operativo Android en el teléfono está dividido en particiones de la memoria nand, esto lo dejamos ver en la figura de abajo, en ella se pueden ver las diferentes particiones que tiene el móvil y la relación que tiene la partición system con el directorio (system del móvil y la partición userdata con el directorio data del terminal)

Las particiones más importantes son:

- La partición de Recovery, aquí se encuentra el sistema de recuperación del móvil.
- La partición Boot, en esta parte se encuentra el Kernel y el sistema de arranque del terminal.
- La partición System, que es la partición que contiene el sistema de Android.

- La partición Userdata, esta parte contiene la configuración del usuario y de las aplicaciones.

En la siguiente ilustración podemos apreciar el ordenamiento de las particiones y del sistema de archivos.

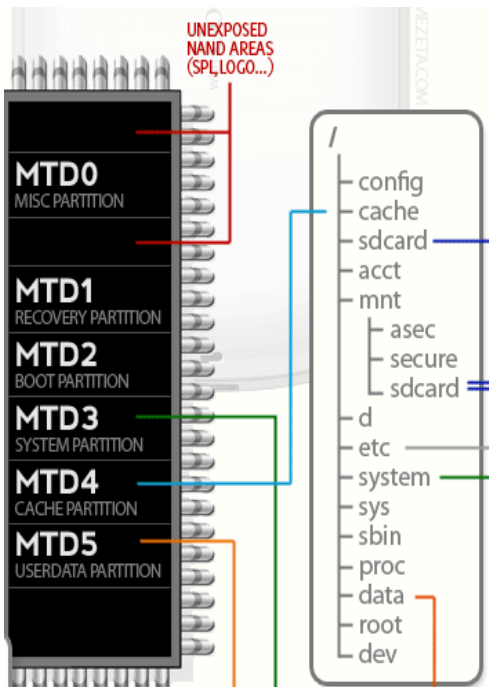


Fig. 7: Diagrama de ordenamiento de particiones y carpetas del sistema en Android
Fuente: Página Web <http://www.elandroidlibre.com>

Cabe mencionar que la memoria Nand en Android usan un túnel de inyección para la escritura y para el borrado un túnel de soldado y estas memorias basadas en NAND tienen, además de la evidente base en otro tipo de puertas, un coste bastante inferior, unas diez veces de más resistencia a las operaciones pero sólo permiten acceso secuencial (más orientado a dispositivos de almacenamiento masivo), frente a las memorias flash basadas en NOR que permiten lectura de acceso aleatorio. Sin embargo, han sido las NAND las que han permitido la expansión de este tipo de memoria, ya que el mecanismo de borrado es más sencillo (aunque también se borre por bloques) lo que ha proporcionado una base más rentable para la creación de dispositivos de tipo tarjeta de memoria. Las populares memorias USB o también llamadas Pendrives, utilizan memorias flash de tipo NAND.

Por otra parte trae incorporado consigo almacenamiento SQLite, que es una base de datos liviana, usada para propósitos de almacenamiento de datos.

Tiene dos tipos de mensajería SMS y MMS, incluyendo mensajería de texto y ahora la Android Cloud toDeviceMessaging Framework (C2DM) es parte del servicio de PushMessaging de Android.

El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, el cual no es más que una plataforma para aplicaciones que funciona como base para el navegador web Safari, Google Chrome, Epiphany, Maxthon, Midori, Qupzilla entre otros. Está basado originalmente en el motor de renderizado KHTML del navegador web del proyecto KDE, Konqueror, emparejado con el motor JavaScript V8 de Google Chrome. El navegador obtiene una puntuación de 93/100 en el test Acid3. Soporte de Java Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma.

El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados.

El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner. Soporte multimedia Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, OggVorbis, WAV, JPEG, PNG, GIF y BMP. Soporte para streaming, RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video>tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft SmoothStreaming con el port de Silverlight a Android. Adobe Flash HTTP DynamicStreaming estará disponible mediante una actualización de Adobe Flash Player. Soporte para hardware adicional Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, termómetro, aceleración 2D y 3D. Entorno de desarrollo Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno de desarrollo integrado es Eclipse (actualmente 3.4, 3.5 o 3.6) usando el plugin de Herramientas de Desarrollo de Android.

Google Play es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.

Android tiene soporte nativo para pantallas multi-táctiles que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel (posiblemente para evitar infringir patentes de otras compañías). Más tarde, Google publicó una actualización para el NexusOne y el Motorola Droid que activa el soporte para pantallas multi-táctiles de forma nativa.

Bluetooth el soporte para A2DF y AVRCP fue agregado en la versión 1.5; el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0; y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2

Android soporta videollamada a través de Google Talk desde su versión HoneyComb.

Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj, a diferencia de otros sistemas de la competencia en la que la multitarea es congelada.

8.2. Herramientas de desarrollo de aplicaciones Android

Para el desarrollo de las aplicaciones en teléfonos con Android se utiliza en conjunto varias herramientas, como son el SDK(Software Development Kit) oficial que provee Google a través de la página <http://developer.android.com/index.html>, en esta página además de este software podemos encontrar tutoriales y mucha información básica para el desarrollo de aplicaciones; el otro componente importante para el desarrollo de aplicaciones en Android es Eclipse, una herramienta para diseño de software basado en Java.

8.2.1. Eclipse 3.7

Eclipse nos brinda una interfaz grafica de sencillo uso en conjunto con el plugin oficial del SDK de Android el ADT (AndroidDevelopment Tools). Esto es muy útil para programadores, pues permite acciones más intuitivas y rápidas.

Para empezar primero debemos realizar las siguientes acciones:

- Paso 1. Descarga e instalación de Eclipse.

Si aún no tienes instalado Eclipse, puedes descargar la última versión (3.7 en el momento de escribir estas líneas) desde este enlace. Recomiendo descargar por ejemplo la versión “Eclipse IDE for Java Developers”. La

instalación consiste simplemente en descomprimir el ZIP en la ubicación deseada.

- Paso 2. Descargar el SDK de Android.

El SDK de la plataforma Android se puede descargar desde este enlace (la última versión disponible en el momento de escribir este tema es la r18). Una vez descargado, de nuevo bastará con descomprimir el zip en cualquier ubicación.

- Paso 3. Descargar el plugin Android para Eclipse.

Google pone a disposición de los desarrolladores un plugin para Eclipse llamado Android Development Tools (ADT) que facilita en gran medida el desarrollo de aplicaciones para la plataforma. Podéis descargarlo mediante las opciones de actualización de Eclipse, accediendo al menú “Help / Install new software...” e indicando la URL de descarga: <https://dl-ssl.google.com/android/eclipse/> Se debe seleccionar e instalar el paquete completo Developer Tools, formado por Android DDMS y Android Development Tools.

- Paso 4. Configurar el plugin ADT.

En la ventana de configuración de Eclipse, se debe acceder a la sección de Android e indicar la ruta en la que se ha instalado el SDK (paso 2).

- Paso 5. Descargar los targets necesarios.

Además del SDK de Android comentado en el paso 2, también debemos descargar los llamados SDK Targets de Android, que no son más que las librerías necesarias para desarrollar en cada una de las versiones concretas de Android. Así, si queremos desarrollar por ejemplo para Android 1.6 tendremos que descargar su target correspondiente. Para ello, desde Eclipse debemos acceder al menú “Window / Android SDK and AVD Manager“, y en la sección Available Packages seleccionar e instalar todos los paquetes deseados.

- Paso 6. Configurar un AVD.

A la hora de probar y depurar aplicaciones Android no tendremos que hacerlo necesariamente sobre un dispositivo físico, sino que podremos configurar un emulador o dispositivo virtual (Android Virtual Device, o AVD) donde poder realizar fácilmente estas tareas. Para ello, volveremos a acceder al AVD Manager, y en la sección Virtual Devices podremos añadir tantos AVD

como se necesiten (por ejemplo, configurados para distintas versiones de Android). Para configurar el AVD tan sólo tendremos que indicar un nombre descriptivo, el target de Android que utilizará, y las características de hardware del dispositivo virtual, como por ejemplo su resolución de pantalla, el tamaño de la tarjeta SD, o la disponibilidad de GPS.

8.2.2. Estructura de un proyecto Android

Cuando se crea un nuevo proyecto Android en Eclipse se genera automáticamente la estructura de carpetas necesaria para poder generar posteriormente la aplicación. Esta estructura será común a cualquier aplicación, independientemente de su tamaño y complejidad. En la figura 8 podemos ver la estructura básica de las carpetas en un proyecto de aplicación básico de Android.

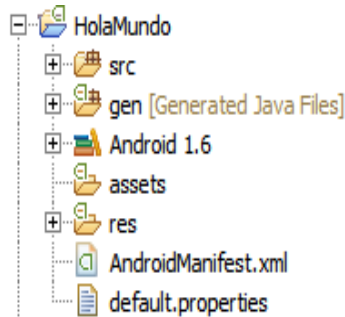


Fig. 8: Imagen de elementos creados inicialmente para un nuevo proyecto Android.[HolaMundo]. Fuente: Captura Propia

8.2.3. Carpeta /src/

Esta Carpeta contiene todo el código fuente de la aplicación a realizar, como el código de la interfaz gráfica, clases auxiliares, etc. Cabe destacar que Eclipse creará por nosotros el código básico de la pantalla (Activity) principal de la aplicación, siempre bajo la estructura del paquete java definido. En la figura 9 vemos el ordenamiento de esta carpeta.



Fig. 9: Recursos básicos /src/. Fuente: Captura Propia

8.2.4. Carpeta /res/

Esta carpeta a diferencia de la anterior es la encargada contener todos los ficheros de recursos necesarios para el proyecto: imágenes, vídeos, cadenas de texto, etc. Los diferentes tipos de recursos de deberán distribuir entre las siguientes carpetas:

Carpeta	Descripción
/res/drawable/	Contienen las imágenes de la aplicación. Para utilizar diferentes recursos dependiendo de la resolución del dispositivo se suele dividir en varias subcarpetas: <ul style="list-style-type: none"> ➤ /drawable-ldpi ➤ /drawable-mdpi ➤ /drawable-hdpi
/res/layout/	Contienen los ficheros de definición de las diferentes pantallas de la interfaz gráfica. Para definir distintos layouts dependiendo de la orientación del dispositivo se puede dividir en dos subcarpetas: <ul style="list-style-type: none"> ➤ /layout ➤ /layout-land
/res/anim/	Contiene la definición de las animaciones utilizadas por la aplicación.
/res/menu/	Contiene la definición de los menús de la aplicación.
/res/values/	Contiene otros recursos de la aplicación como por ejemplo cadenas de texto (strings.xml), estilos (styles.xml), colores (colors.xml), etc.
/res/xml/	Contiene los ficheros XML utilizados por la aplicación.
/res/raw/	Contiene recursos adicionales, normalmente en formato distinto a XML, que no se incluyan en el resto de carpetas de recursos.

Y aquí abajo en la figura 10 tenemos una captura de pantalla de la estructura de la carpeta /res/.

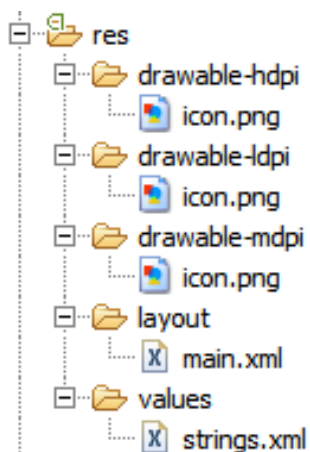


Fig. 10: Recursos básicos /res/ Fuente: Captura Propia

8.2.5. Carpeta /gen/

Contiene una serie de elementos de código generados automáticamente al compilar el proyecto. Cada vez que generamos nuestro proyecto, la maquinaria de compilación de Android genera por nosotros una serie de ficheros fuente en java dirigidos al control de los recursos de la aplicación. En la siguiente figura vemos como se ordenan los archivos generados automáticamente en la carpeta /gen/.

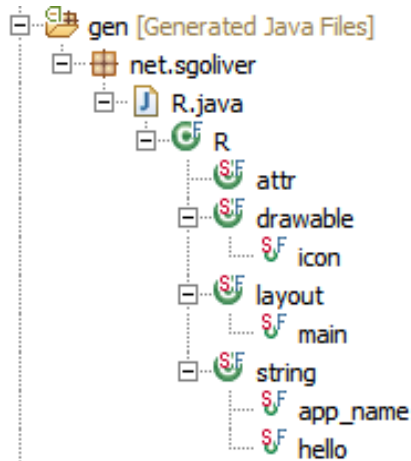


Fig. 11: Recursos básicos /gen/ Fuente: Captura Propia

8.2.6. Carpeta /assets/

Esta es la carpeta que contiene todos los demás ficheros auxiliares necesarios para la aplicación (y que se incluirán en su propio paquete), como por ejemplo ficheros de configuración, de datos, etc. La diferencia entre los recursos incluidos en la carpeta /res/raw/ y los incluidos en la carpeta /assets/ es que para los primeros se generará un ID en la clase R y se deberá acceder a ellos con los diferentes métodos de acceso a recursos. Para los segundos sin embargo no se generarán ID y se podrá acceder a ellos por su ruta como a cualquier otro fichero del sistema. Es importante mencionar que se usará uno u otro según las necesidades de nuestra aplicación.

8.2.7. Fichero AndroidManifest.xml

Esta parte de los elementos creados inicialmente para un nuevo proyecto Android, contiene la definición en XML de los aspectos principales de la aplicación, como por ejemplo su identificación (nombre, versión, icono, ...), sus componentes (pantallas, mensajes, ...), o los permisos necesarios para su ejecución.

8.3. Componentes de una aplicación Android

8.3.1. Activity

Las actividades representan el componente principal de la interfaz gráfica de una aplicación Android. Se puede pensar en una actividad como el elemento análogo a una ventana en cualquier otro lenguaje visual.

8.3.2. View

Los objetos view son los componentes básicos con los que se construye la interfaz gráfica de la aplicación, por ejemplo a los controles de Java o .NET. De inicio, android pone a nuestra disposición una gran cantidad de controles básicos, como cuadros de texto, botones, listas desplegables o imágenes, aunque también existe la posibilidad de extender la funcionalidad de estos controles básicos o crear nuestros propios controles personalizados.

8.3.3. Service

Los servicios son componentes sin interfaz gráfica que se ejecutan en segundo plano. Son exactamente iguales a los servicios presentes en cualquier otro sistema operativo. Los servicios pueden realizar cualquier tipo de acciones, por ejemplo actualizar datos, lanzar notificaciones, o incluso mostrar elementos visuales (p.ej. activities) si se necesita en algún momento la interacción con el usuario.

8.3.4. Content Provider

Un content provider es el mecanismo que se ha definido en Android para compartir datos entre aplicaciones. Mediante estos componentes es posible compartir determinados datos de nuestra aplicación sin mostrar detalles sobre su almacenamiento interno, su estructura, o su implementación. De la misma forma, nuestra aplicación podrá acceder a los datos de otra a través de los content provider que se hayan definido.

8.3.5. Broadcast Receiver

Un broadcast receiver es un componente destinado a detectar y reaccionar ante determinados mensajes o eventos globales generados por el sistema (por ejemplo: "Batería baja", "SMS recibido", "Tarjeta SD insertada", ...) o por otras aplicaciones (cualquier aplicación puede generar mensajes (intents, en terminología Android) broadcast, es decir, no dirigidos a una aplicación concreta sino a cualquiera que quiera escucharlo).

8.3.6. Widget

Los widgets son elementos visuales, normalmente interactivos, que pueden mostrarse en la pantalla principal (home screen) del dispositivo Android y recibir actualizaciones periódicas. Permiten mostrar información de la aplicación al usuario directamente sobre la pantalla principal.

8.3.7. Intent

Un intent es el elemento básico de comunicación entre los distintos componentes Android que hemos descrito anteriormente. Se pueden entender como los mensajes o peticiones que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones. Mediante un intent se puede mostrar una actividad desde cualquier otra, iniciar un servicio, enviar un mensaje broadcast, iniciar otra aplicación, etc.

9. Desarrollo

9.1. Preparación:

- **Descargar Eclipse:**

El primer paso a seguir en este proceso es descargar e instalar la versión más nueva de eclipse. Esta se encuentra en la página oficial: <http://www.eclipse.org/downloads/>

Una vez que instalamos Eclipse procedemos a descargar e instalar el plugin del SDK de Android desde la siguiente dirección: <http://developer.android.com/sdk/index.html>

Una vez que tenemos listo esto ya podemos empezar a diseñar la aplicación. La pantalla principal de Eclipse se mira como la fig. 12 que tenemos acá abajo.

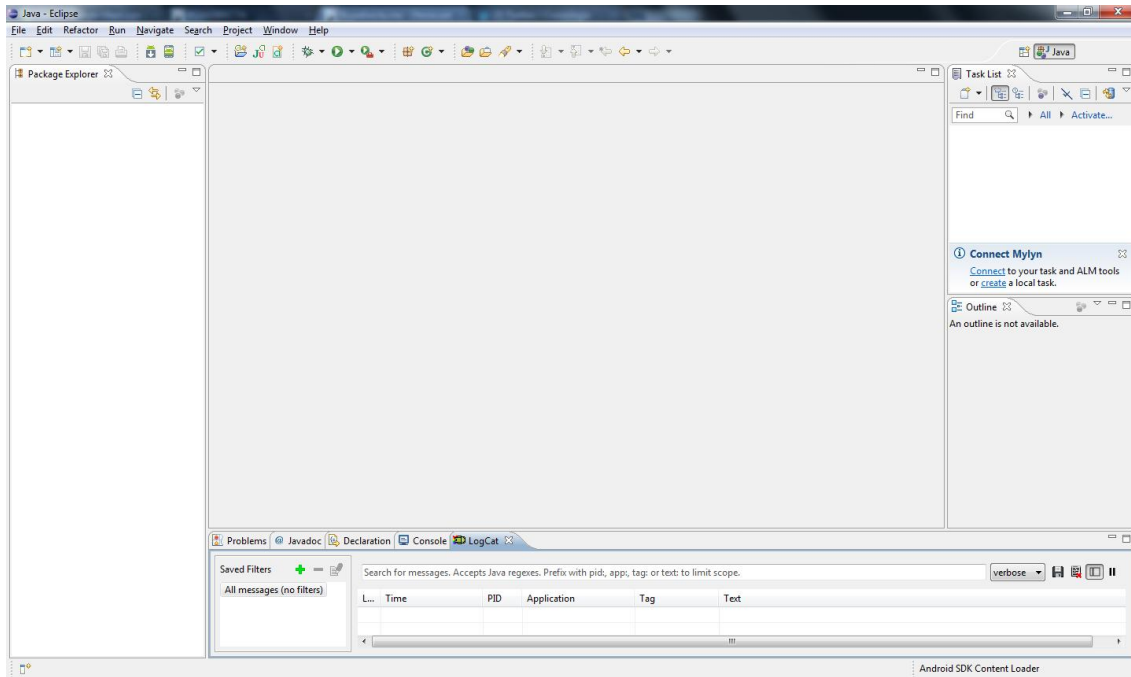


Fig. 12 Pantalla principal de Eclipse Fuente: Captura Propia

Primero debemos crear un nuevo proyecto de Android desde el menú File al acceder se nos presentará la ventana en la cual debemos poner el nombre de la aplicación como se muestra en la figura 13.

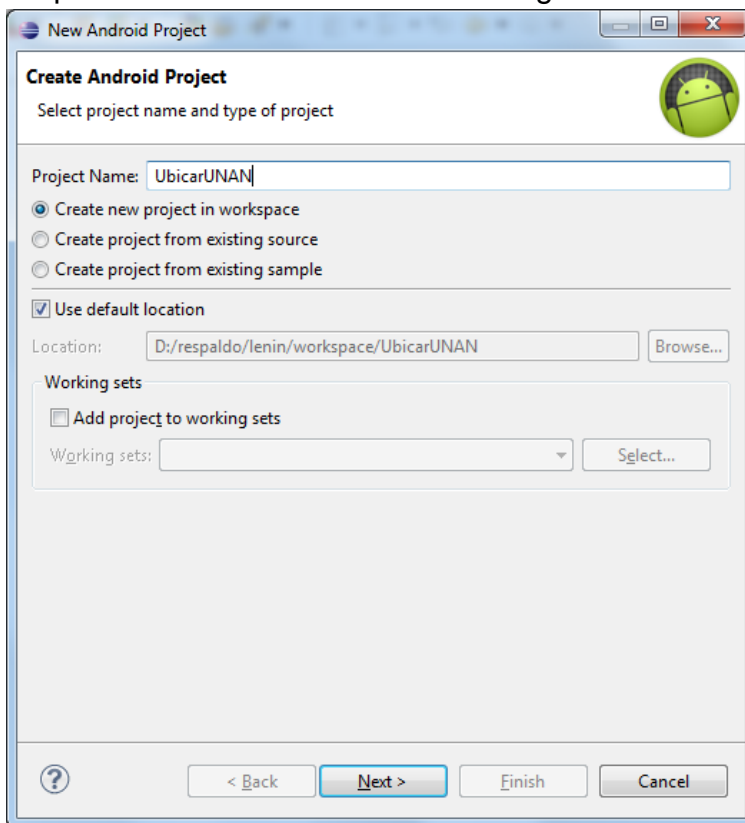


Fig. 13 Nombrar nuestra aplicación para Android. Fuente: Captura Propia

Una vez que hemos puesto el nombre a nuestra aplicación debemos escoger el API con el cual vamos a trabajar, debido a que usaremos una aplicación en la que utilizaremos mapas hemos escogido trabajar con la API de nivel 7 la cual corresponde a la versión 2.1 de Android o también llamado Eclair, en vista que buscamos a la vez compatibilidad con varios dispositivos; cabe resaltar que la mayoría de los dispositivos con Android en nuestro país traen esta versión o posterior por lo tanto nuestra aplicación será compatible con la mayoría de los dispositivos. En la figura 14 que está ubicada en la siguiente página apreciamos el punto en que se escoge la versión o API de Android que usaremos en este caso Google API bajo plataforma Android 2.1 o Eclair.

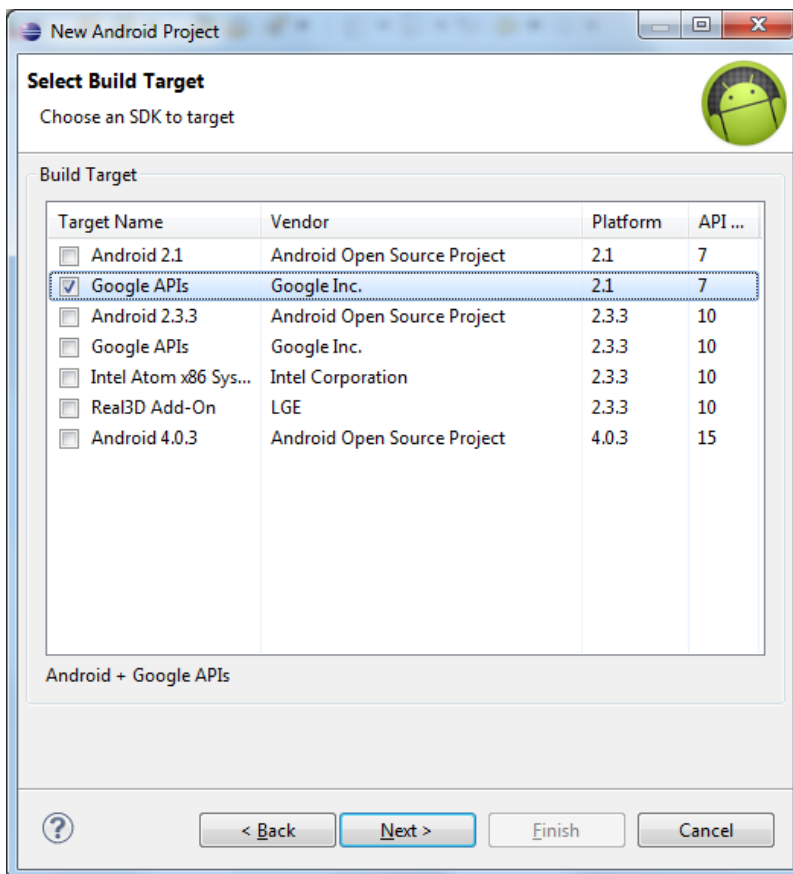


Fig.14 Escogiendo API correspondiente a la aplicación. Fuente: Captura Propia

Finalmente para tener listo el proyecto debemos escoger el nombre del paquete el cual es "com.mapa.unan" y crear la actividad principal de la aplicación que es "UbicarUNANActivity"; tal y como podemos apreciar en la

siguiente captura de la figura 15. Tras concluir esto ya podemos empezar con la programación de la aplicación.

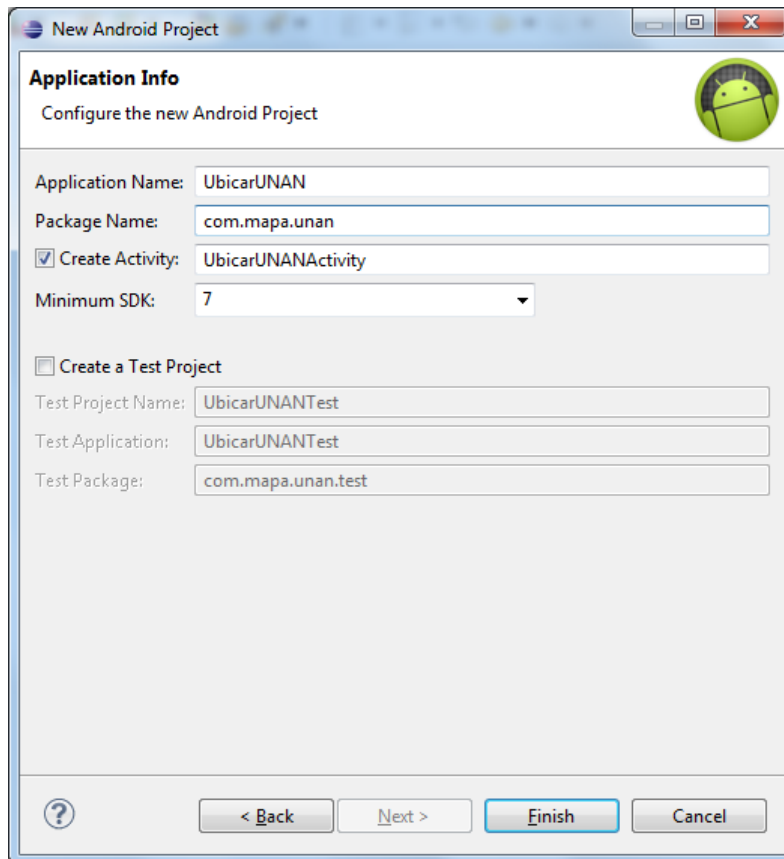
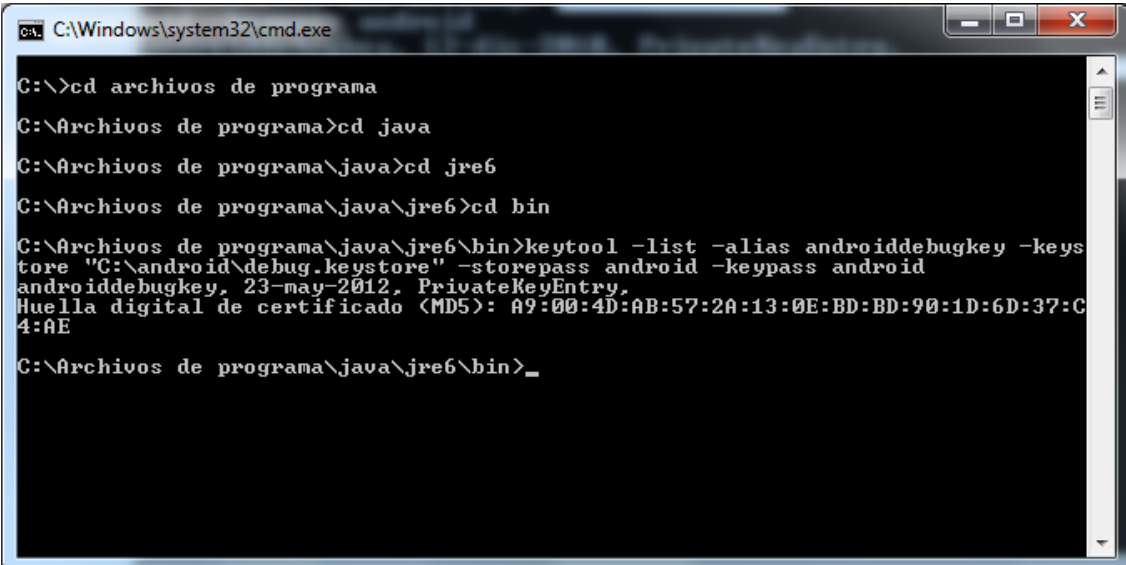


Fig. 15 Aquí ponemos el nombre del paquete y el nombre de la actividad principal.
Fuente: Captura Propia

- **Obtener una Keystore para poder utilizar el API de Google Maps**

Debido a que vamos a trabajar con el API de Google Maps debemos conseguir una Keystore de Google que nos servirá para poder simular y una vez terminada la aplicación empaquetar y montarla en un dispositivo físico.

Lo primero que debemos hacer es buscar localizar el archivo que contiene la keystore que trae por defecto y una vez localizada debemos usar la aplicación java llamada keytool para obtener la etiqueta MD5 que trae la cual es única para cada instalación de Eclipse. En la figura 16 vemos como se obtiene la etiqueta desde consola de comandos.



```
C:\Windows\system32\cmd.exe
C:\>cd archivos de programa
C:\Archivos de programa>cd java
C:\Archivos de programa\java>cd jre6
C:\Archivos de programa\java\jre6>cd bin
C:\Archivos de programa\java\jre6\bin>keytool -list -alias androiddebugkey -keystore "C:\android\debug.keystore" -storepass android -keypass android
androiddebugkey, 23-may-2012, PrivateKeyEntry,
Huella digital de certificado (MD5): A9:00:4D:AB:57:2A:13:0E:BD:BD:90:1D:6D:37:C4:AE
C:\Archivos de programa\java\jre6\bin>_
```

Fig. 16 Utilizando keytool para obtener la huella digital única MD5. Fuente: Captura propia

Tras tener lista la etiqueta o huella digital tenemos que conectarnos utilizando nuestra cuenta de Google y acceder a la siguiente dirección:

<http://code.google.com/android/maps-api-signup.html> Una vez allí debemos aceptar las clausulas del uso de los mapas de Google y poner nuestra etiqueta MD5 cuando y obtenemos nuestra llave nueva. En la figura 17 se observa nuestra llave.



API de Google Maps
[Página principal de Google Code](#) > [API de Google Maps](#) > Suscripción al API de Google Maps

Gracias por suscribirte a la clave del API de Android Maps.

Tu clave es:

0UoYVocYrKLYcn1GxkUAFNspq1BBUL9PLnGp40A

Esta clave es válida para todas las aplicaciones firmadas con el certificado cuya huella dactilar sea:

A9:00:4D:AB:57:2A:13:0E:BD:BD:90:1D:6D:37:C4:AE

Incluimos un diseño xml de ejemplo para que puedas iniciarte por los senderos de la creación de mapas:

```
<com.google.android.maps.MapView
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:apiKey="0UoYVocYrKLYcn1GxkUAFNspq1BBUL9PLnGp40A"
 />
```

Consulta la [documentación del API](#) para obtener más información.

©2008 Google - [Política de privacidad](#) - [Ofertas de empleo](#) - [Condiciones de uso](#) - [Acerca de Google](#)

Fig. 17 Llave y ejemplo de código que nos proporciona Google para utilizar mapas en nuestras aplicaciones.

Cuando ya tenemos nuestra llave somos libres para utilizarla en todas las aplicaciones que queramos sean para motivos educativos, así como en prácticas o para aplicaciones comerciales.

9.2. Creación del diagrama de flujo de la aplicación

Tomando en cuenta las actividades que necesita realizar nuestra aplicación, se debe crear el diagrama de flujo el cual servirá como referencia para la subsecuente programación de la aplicación. En el diagrama debemos reflejar cómo se comportará la aplicación. En la figura 18 mostramos el diagrama de flujo que hemos diseñado para la aplicación.

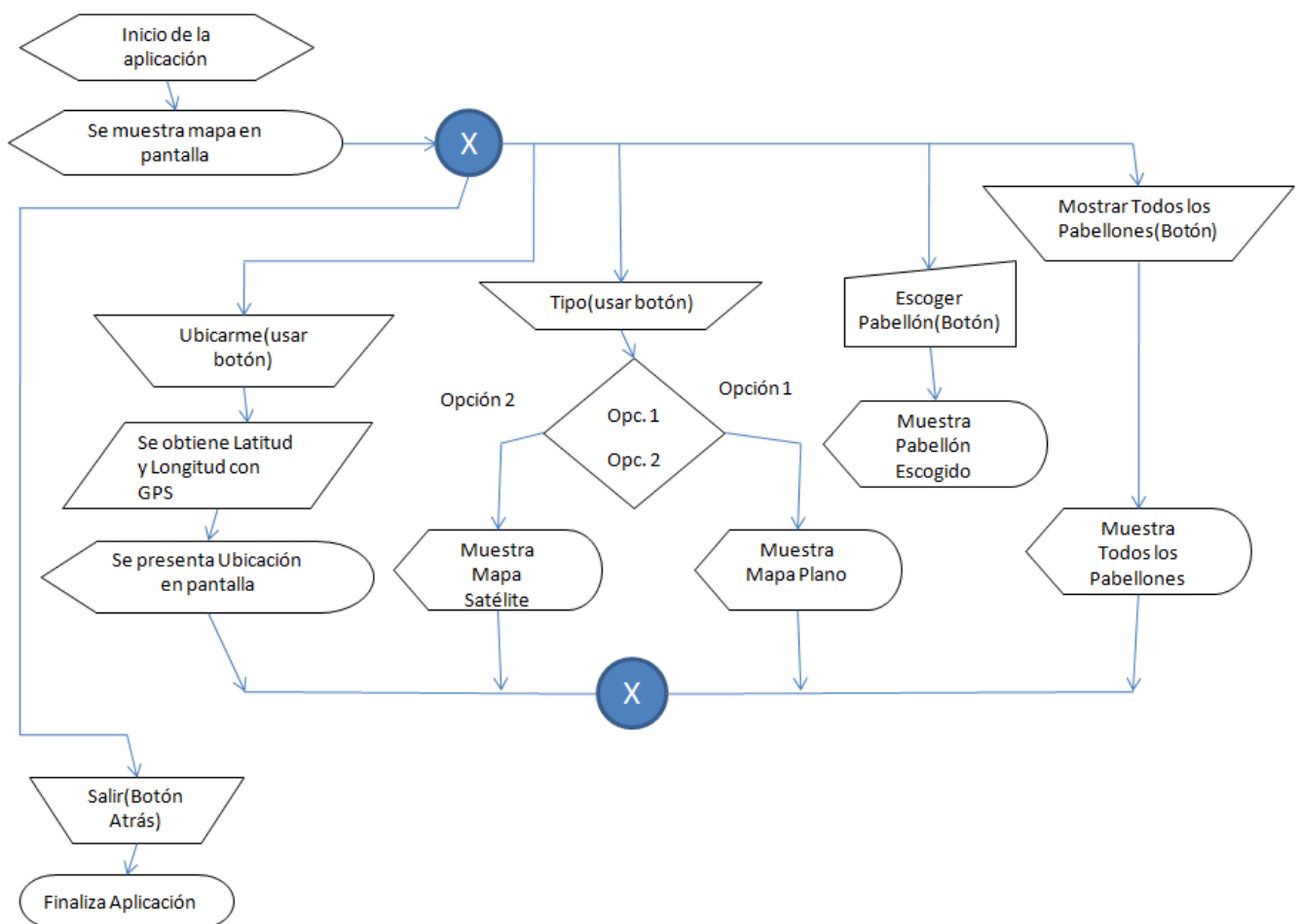


Fig. 18 Diagrama de flujo de la aplicación. Fuente: Propio

Tras analizar las acciones que debe realiza la aplicación que estamos diseñando hemos llegado a la conclusión de que serán necesarios 4 botones de acción principales para obtener los resultados esperados.

Debido a que nuestra aplicación se desarrolla en un lenguaje de programación basada en objetos, esto quiere decir que la forma en que se trabajará será cambiando los valores de las variables asociadas a los objetos, mediante eventos.

Desde que la aplicación arranca mostrará un mapa con una ubicación global por defecto, este es el primer evento que tendrá nuestra aplicación.

Siguiendo la lógica del funcionamiento de nuestra aplicación un punto importante a resaltar será que los demás eventos se pueden realizar secuencialmente sin importar el orden. Una de las acciones que necesitamos que realice nuestra aplicación es ubicarnos en el mapa para esto programaremos un botón que utilice el localizador GPS del teléfono, este botón se llamará “Ubicarme”.

Otra de las acciones que debe hacer la aplicación de geolocalización es permitir cambiar el tipo de mapa que se despliega en pantalla, hemos decidido que muestre un mapa plano y un mapa de satélite, para que se lleve a cabo esta acción se procederá a programar un botón llamado “Tipo”, al presionar este botón se cambiará entre ambos tipos de vista.

Nuestra aplicación debe mostrar puntos concretos que permitirán al usuario llegar a dichos puntos o tenerlos como referencia. Para esto programaremos 2 botones uno que permitirá mostrar todos los pabellones del recinto así como los auditorios, este botón se llamará “Mostrar Pabellones” y finalmente también programaremos un spinner o lista desplegable que permitirá solamente mostrar un pabellón a la vez.

Para finalizar nuestra aplicación bastará con accionar sobre el botón “Atrás” de nuestro dispositivo Android.

9.3. Programación

Una vez que hemos completado los pasos anteriores nos toca hacer la interfaz gráfica y la programación de la aplicación.

Empezamos creando la interfaz grafica abriendo la carpeta Layout que se crea automáticamente al crear el proyecto. Utilizaremos 2 botones, uno para ubicarnos en cualquier punto de la unan y el otro para mostrar todos los pabellones, y también un control spiner para mostrar uno a uno el pabellón deseado; por lo último lo más importante el mapa, el cual se llama control mapView. Como se muestra en la figura 18 se vería la interfaz gráfica de la aplicación.

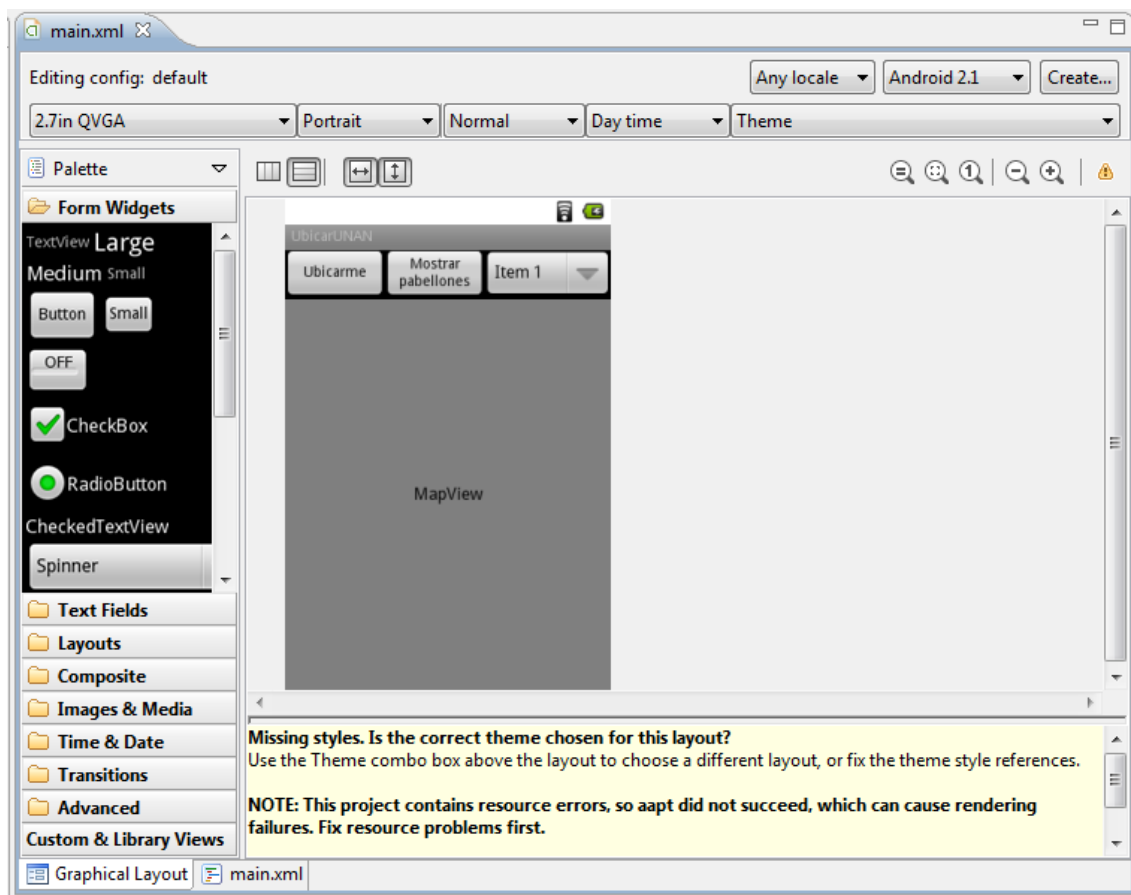


Fig. 18 Interfaz gráfica de la aplicación

Cuando ya tenemos listo la interfaz grafica procedemos a incluir el código fuente que permitirá el funcionamiento de la aplicación. Entonces debemos abrir el fichero .java de la actividad principal el cual se encuentra en la carpeta SRC a la cual podemos acceder desde el explorador de paquetes.

Primero declaramos las clases y variables a usar en la figura 19 se aprecia.

```

package com.ubicacion.unan;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapController;
import com.google.android.maps.MapView;

import android.content.Context;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;

public class UbicacionUnan extends MapActivity {
    /** Called when the activity is first created. */
    private MapView mapa = null;
    private Button btntipo = null;
    private Button btnUbicar = null;
    private Button btnPabellones = null;
    private Spinner Spinner=null;
    private MapController controlmapa=null;
    
```

Fig. 19 Declaración de variables y clases que se usaran en la aplicación. Fuente: Captura Propia

Ahora declaramos las acciones que permiten la visualización del mapa y el código para obtener nuestra ubicación tal y como se muestra en la figura 20.

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    mapa = (MapView)findViewById(R.id.mapaunan);
    btntipo = (Button)findViewById(R.id.button3);
    btnUbicar = (Button)findViewById(R.id.button2);
    btnPabellones = (Button)findViewById(R.id.button1);
    Spinner = (Spinner)findViewById(R.id.spinner1);
    controlmapa = mapa.getController();

    mapa.setBuiltInZoomControls(true);

    btnUbicar.setOnClickListener(new OnClickListener(){

        public void onClick(View arg0) {

            // TODO Auto-generated method stub
            GeoPoint loc = mapa.getMapCenter();
            int latitud= loc.getLatitudeE6();
            int longitud= loc.getLongitudeE6();

            controlmapa.setZoom(18);
            Context contexto=mapa.getContext();
            String msg ="Lat:"+latitud/1E6 + " "+"Lon:"+longitud/1E6;
            Toast toast1=Toast.makeText(contexto, msg, Toast.LENGTH_SHORT);
            toast1.show();

        }

    });
}

```

Fig. 20 Código para control de mapa y ubicación. Fuente: Captura Propia

Ahora debemos agregar el código para cambiar el tipo de vista y obtener los puntos de los pabellones en la figura 21 de abajo podemos ver dicho código.

```
btntipo.setOnClickListener(new OnClickListener() {
public void onClick(View arg0) {
    if(mapa.isSatellite())
        mapa.setSatellite(false);
    else
        mapa.setSatellite(true);
}
});

}
public boolean onTap(GeoPoint point1, MapView mapa)
{
    int latitud1=point1.getLatitudeE6();
    int longitud1=point1.getLongitudeE6();

    Context contexto1 = mapa.getContext();
    String msg1 = "Lat: " + latitud1/1E6 + " - " + "Lon: " + longitud1/1E6;

    Toast toast2 = Toast.makeText(contexto1, msg1, Toast.LENGTH_SHORT);
    toast2.show();

    return false;
}
```

Fig 21. Código para cambiar el tipo de vista y para obtener puntos. Fuente: Captura Propia

Finalmente se debe editar el archivo AndroidManifest.xml que contiene los permisos que utiliza cada aplicación para funcionar correctamente aquí abajo en la figura 22 vemos los permisos que hemos incluido en nuestra aplicación.

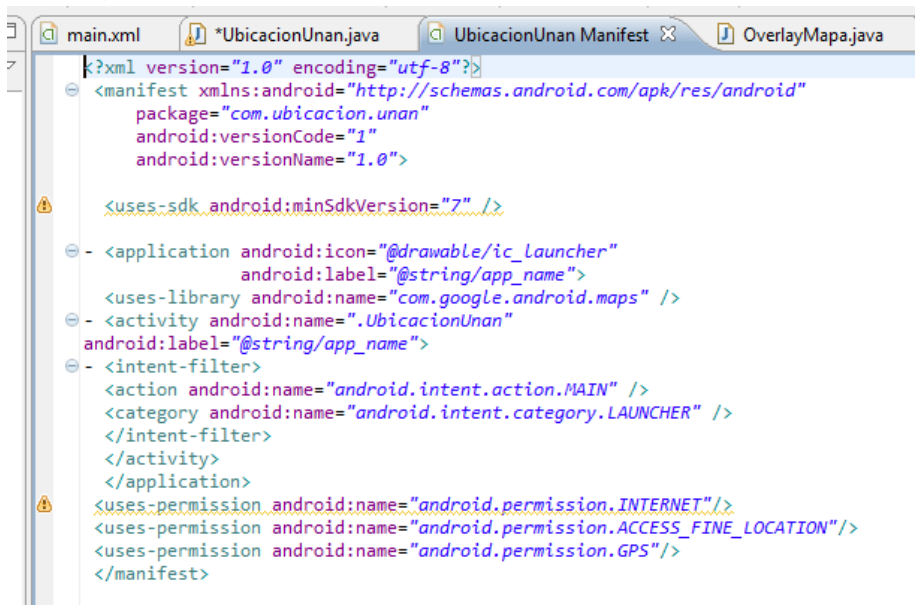


Fig. 22 Fichero AndroidManifest.xml con los permisos que necesitamos para nuestra aplicación

Finalmente para probar nuestra aplicación debemos crear un dispositivo virtual Android en la figura 23 vemos los parámetros que hemos incluido para que funcione correctamente nuestra aplicación.

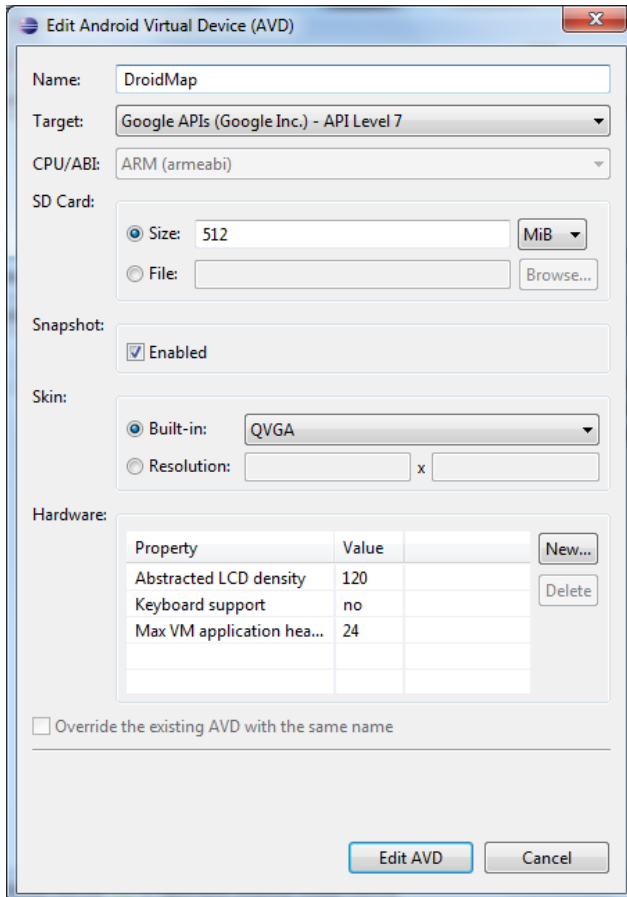


Fig. 23 parámetros del dispositivo Android Virtual para nuestra aplicación. Fuente: Captura Fuente Propia

Ya configurado nuestro dispositivo Android lo iniciamos y debido a que usamos la API 7 de Google por su facilidad de uso para mapas y uso extendido en varios dispositivos con Android, el cual corresponde a la versión 2.1 de Android o Eclair en la figura 24 se ve nuestra unidad virtual de Android

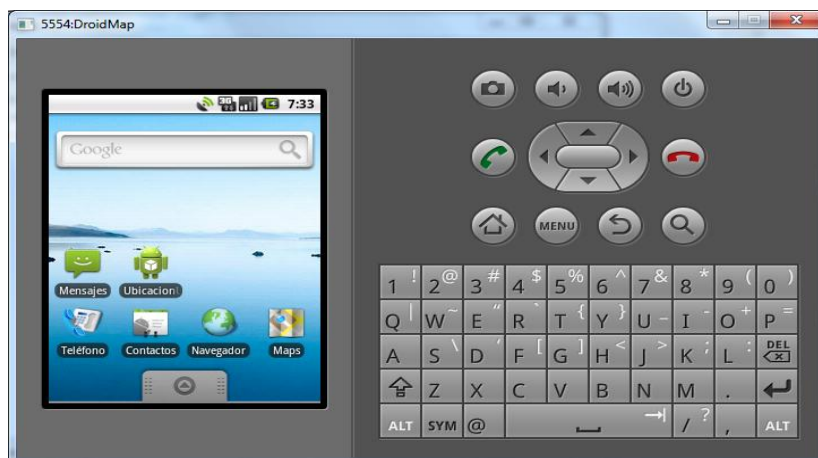


Fig. 24 Dispositivo virtual Android para simulación de aplicaciones. Fuente: Captura Propia

Finalmente ya podemos probar nuestra aplicación android en nuestro dispositivo virtual. En la figura 25 podemos ver funcionando nuestra aplicación “UbicacionUnan”

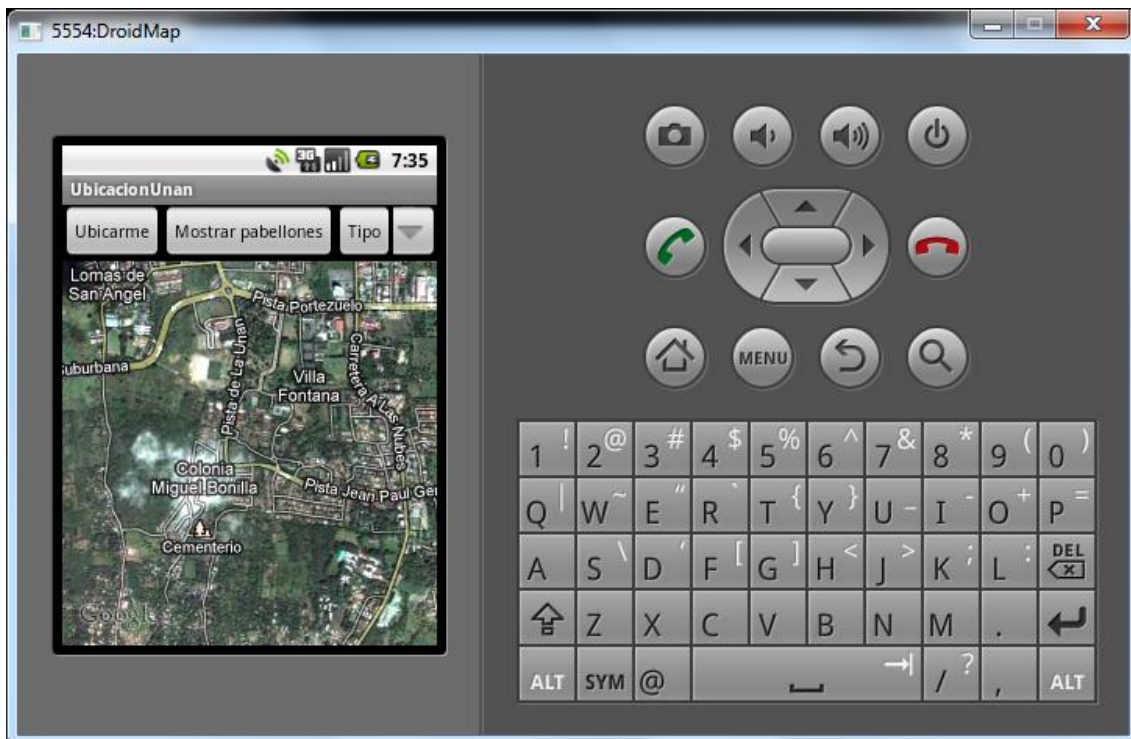


Fig. 25 Aplicación “UbicacionUnan” corriendo en el dispositivo Android Virtual Fuente: Captura Propia

9.4. Requerimientos mínimos de hardware y software para correr la aplicación “UbicacionUnan”

Tras concluir con la programación, las simulaciones y las pruebas en diferentes dispositivos como por ejemplo en Motorola Droid Pro, LG P509, Samsung Galaxy Pro; hemos diseñado la siguiente tabla que permite saber los requerimientos mínimos de hardware y software necesarios para que la aplicación corra eficientemente.

Característica	Valor Requerido
Procesador	600Mhz
Memoria Ram	256 MB
GPS	Si
Conexión a internet	Conexión 3G mínimo o WIFI
Cámara	No es necesario
Acelerómetro	Si(para rotar automáticamente le mapa al voltear el dispositivo)
Tamaño Display	240*320 pixeles
Espacio para instalación (memoria para Aplicación)	172 KB
Memoria Micro-SD	Si(no importa la capacidad la aplicación solo necesita 172 KB)
SO	Android 2.1(Eclair)

Estos son los requerimientos mínimos que deberá poseer el dispositivo en que se instale la aplicación “UbicacionUNAN” casi todos los teléfonos con Android cumplen estos requerimientos lo cual es muy conveniente para que la aplicación se pueda difundir en varios dispositivos.

9.5. Manual de UbicacionUNAN

La aplicación “UbicacionUnan” ha sido diseñada para trabajar en teléfonos con Android desde la versión 2.1 denominada Eclair, puede correr en todas las versiones posteriores. Los terminales con Android poseen sistema GPS y conexión a Internet mediante WIFI o plan de datos, que serán los recursos necesarios para que pueda ejecutarse correctamente la aplicación.

Primeramente deberemos copiar el archivo UbicarUnan.apk en la memoria microSD del dispositivo.

Posteriormente debemos utilizar un gestor de archivos (en teléfonos con Android 2.3 ya traen su propio gestor) previamente instalado en el dispositivo (como Astro File Manager) Android para correr el instalador de la aplicación.

Cuando localizamos el archivo .apk solo lo abrimos y aceptamos los permisos que requiere la aplicación (permiso de conexión a internet y acceso al GPS) y listo tendremos nuestra aplicación instalada.

Una vez que hayamos instalado la aplicación en nuestro terminal Android, procederemos a buscarla en la lista de aplicaciones instaladas y la abrimos con simplemente tocar el icono.

La aplicación trae 3 botones y 1 spinner.

- **Botón Ubicarme:** Cuando uno presiona este botón obtiene la ubicación actual del dispositivo y sus coordenadas.
- **Botón Mostrar Pabellones:** Muestra la ubicación de todos los pabellones y auditorios de la universidad.
- **Botón Tipo:** Permite cambiar la vista que usaremos para visualizar el mapa, las 2 posibles opciones son mapa con imagen de satélite y mapa tipo plano.
- **Spinner:** El uso de este control permite que escojamos un pabellón para poder visualizarlo en el mapa.

Entre opciones podemos modificar el zoom del mapa acercando o alejando nuestros dedos sobre un punto de la pantalla del móvil o utilizar el pequeño botón con el + o – para realizar estos cambios.

Para salir de la aplicación solo debemos utilizar el botón “Back” del teléfono.

10. Conclusiones

Tras finalizar este trabajo hemos llegado a las siguientes conclusiones en base a los objetivos planteados:

Nuestro principal objetivo era desarrollar una aplicación aprovechando las capacidades de los dispositivos con Android, específicamente el uso del GPS y de la conexión a Internet, que son capacidades que poseen casi todos los dispositivos con Android, tomamos en como referencia las capacidades del Motorola Droid 2 Global para esto.

Hemos diseñado la aplicación para que funcione en los dispositivos a partir de la versión 2.1 de Android denominada Eclair, cabe señalar que la versión más difundida es la 2.3 conocida como GingerBread la que trae el Motorola Droid 2 Global y otros dispositivos en los cuales fue probada la aplicación, pero también nos intereso que nuestra aplicación se pudiera ejecutar con modelos anteriores y sirviera en más dispositivos.

El uso del software de desarrollo Eclipse permitió que la programación de nuestra aplicación fuese más sencilla recalando que no teníamos conocimientos para programar en Java. Lo cual demuestra que el potencial para diseñar otras aplicaciones en Android utilizando esta herramienta es muy amplio y abre muchas posibilidades para programadores independientes.

Puesto que nuestra aplicación sería usada por diferentes usuarios hemos creado un pequeño manual donde se detallan los pasos para la instalación y uso, en un lenguaje sencillo y práctico, perfectamente entendible para cualquier usuario promedio de Android.

11. Recomendaciones:

Tras la conclusión de este pequeño proyecto y tomando en cuentas nuestras experiencias nuestras recomendaciones para el diseño de aplicaciones en teléfonos celulares son las siguientes:

- Conocer el hardware de los diferentes dispositivos en los cuales se instalará la aplicación para saber cuáles serían las diferentes limitaciones de la aplicación a desarrollar.
- Documentarse sobre el lenguaje de programación que se utilizará y sobre la interfaz o herramienta de desarrollo.
- Estar atentos a las actualizaciones que se le realicen al software de desarrollo para aprovechar las diferentes mejoras y correcciones que se realicen.
- Probar la aplicación en diferentes dispositivos, para ver su comportamiento en pantallas de diferentes tamaños o medidas; para revisar el rendimiento en dependencia del procesador y la cantidad de memoria Ram.

12. Glosario de términos

ADT: Siglas de Android Development Tools, en español esto significa herramientas de desarrollo Android, son un conjunto de aplicaciones que vienen con el plugin para Eclipse.

Amoled: Es un dispositivo de display Oled de matriz activa que consiste en un conjunto de píxeles OLED que se depositan o integran en una serie de transistores de película fina (TFT) para formar una matriz de píxeles, que se iluminan cuando han sido activados eléctricamente, controlados por los interruptores que regulan el flujo de corriente que se dirige a cada uno de los píxeles. El TFT continuamente regula la corriente que fluye por cada uno de los píxeles, para así caracterizar el píxel con el nivel de brillo que mostrará.

API: Por sus siglas en inglés es interfaz de programación de aplicaciones. Una interfaz de programación representa la capacidad de comunicación entre componentes de software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla. De esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio. Las APIs asimismo son abstractas: el software que proporciona una cierta API generalmente es llamado la implementación de esa API.

AMPS: Es el Sistema Telefónico Móvil Avanzado o AMPS (del inglés *Advanced Mobile Phone System*) es un sistema de telefonía móvil de primera generación (1G, voz analógica) desarrollado por los laboratorios Bell. Se implementó por primera vez en 1982 en Estados Unidos. Se llegó a implantar también en Inglaterra y en Japón, con los nombres TACS y MCS-L1 respectivamente.

AVD: Es Android Virtual Device, es un emulador de dispositivos Android que trae el SDK de Android para Eclipse, mediante este emulador podemos probar nuestras aplicaciones, creando dispositivos con características específicas.

Eclair: Es la versión 2.1 de Android. Trajo varias mejoras como: Velocidad de hardware optimizada, soporte para más tamaños de pantalla y resoluciones, interfaz de usuario renovada, nuevo interfaz de usuario en el navegador y soporte para HTML5, nuevas listas de contactos, una mejor relación de contraste para los fondos, mejoras en Google Maps 3.1.2, soporte para Microsoft Exchange, soporte integrado de flash para la cámara, zoom digital, MotionEvent mejorado para captura de eventos multi-touch, teclado virtual mejorado Bluetooth 2.1 fondos de pantalla animados.

Framework: Llamado también *infraestructura digital*, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de *software* concretos, con base a la cual otro proyecto de *software* puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Gingerbread: Es la versión 2.3 de Android actualmente es la más difundida en el mundo entre los dispositivos con este sistema operativo.

GPU: La unidad de procesamiento gráfico o GPU (acrónimo del inglés *graphicsprocessingunit*) es un coprocesador dedicado al procesamiento de gráficos u operaciones de coma flotante, para aligerar la carga de trabajo del procesador central en aplicaciones como los videojuegos y o aplicaciones 3D interactivas. De esta forma, mientras gran parte de lo relacionado con los gráficos se procesa en la GPU, la unidad central de procesamiento (CPU) puede dedicarse a otro tipo de cálculos (como la inteligencia artificial o los cálculos mecánicos en el caso de los videojuegos)

Kernel: En informática, un núcleo o kernel (de la raíz germánica *Kern*, núcleo, hueso) es un software que constituye la parte más importante del sistema operativo. Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. Como hay muchos programas y el acceso al hardware es limitado, también se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo, lo que se conoce como multiplexado. Acceder al hardware directamente puede ser realmente complejo, por lo que los núcleos suelen implementar una serie de abstracciones del hardware. Esto permite esconder la complejidad, y proporciona una interfaz limpia y uniforme al hardware subyacente, lo que facilita su uso al programador.

Keystore: El keystore es la base de datos de claves privadas y sus certificados asociados. La entrada keystore sirve para indicar dónde consultar las claves públicas de las entidades firmantes especificadas en las entradas grant del fichero. La entrada keystore debe aparecer en el fichero si cualquier entrada grant especifica el alias de una entidad que firma.

Plugin: También se lo conoce como plug-in (del inglés "enchufable"), add-on (agregado), complemento, conector o extensión es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

SDK: Un kit de desarrollo de software o SDK (siglas en inglés de *software development kit*) es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto, por ejemplo ciertos paquetes de software, frameworks, plataformas de hardware, computadoras, videoconsolas, sistemas operativos, etc.

Spinner: Las listas desplegadas en Android se llaman Spinner. Funcionan de forma similar al de cualquier control de este tipo, el usuario selecciona la lista, se muestra una especie de lista emergente al usuario con todas las opciones disponibles y al seleccionarse una de ellas ésta queda fijada en el control.

Streaming: Es la distribución de multimedia a través de una red de computadoras de manera que el usuario consume el producto al mismo tiempo que se descarga. La palabra *streaming* se refiere a que se trata de una corriente continua (sin interrupción). Este tipo de tecnología funciona mediante un búfer de datos que va almacenando lo que se va descargando para luego mostrarse al usuario. Esto se contrapone al mecanismo de descarga de archivos, que requiere que el usuario descargue los archivos por completo para poder acceder a ellos.

13. Bibliografía:

Jorge Carballo Franquis, David D. Harjani. *Desarrollo de Aplicaciones Móviles en Android* Universidad de la Laguna. Formato: PDF

Federico Fernández, Mariana Fernández. *Tecnologías de la Información y las comunicaciones* Formato: PDF

Salvador Gómez Oliver *Curso Programación Android*. Versión 2.0 Formato: PDF

Adrian Catalán. *Curso Android: Desarrollo de Aplicaciones Móviles Maestros del Web*. Versión: 1.0. Formato: PDF

Miguel Francisco Núñez Burguera *Creación de Sistemas Android Personalizados* Formato: PDF

Mónica Batanero La Rotta *Aplicación GeoBuddies para Android* Universidad Politécnica de Madrid. Formato: PDF

Josep Prieto Blázquez, Robert Ramírez Vique, Julián David Morillo Pozo, Marc Domingo Prieto. *Tecnología y Desarrollo en dispositivos móviles* Universitat Oberta de Catalunya. Formato: PDF

Valinda Sequeira Calero, Astralia Cruz Picón. *Investigar es fácil Manual de Investigación*. Unan Managua, Managua, Nicaragua, 1994.

Web oficial para desarrolladores de Aplicaciones en Android: <<http://developer.android.com/index.html>>