



UNIVERSIDAD PARTICULAR DE CHICLAYO



FAUI - UDCH

**FACULTAD DE ARQUITECTURA Y URBANISMO E INGENIERIAS
ESCUELA PROFESIONAL DE INGENIERIA INFORMATICA Y DE SISTEMAS**

TRABAJO DE SUFICIENCIA PROFESIONAL

**“IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA EL
SEGUIMIENTO DE PERSONAS MAYORES DE EDAD UTILIZANDO
GEOLOCALIZACIÓN DESDE UN DISPOSITIVO ANDROID CONECTADO
A UN SERVIDOR WEB CON MYSQL”**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:
INGENIERO INFORMÁTICO Y DE SISTEMAS**

AUTOR:

Bach. Maza Montes Jose Waldir

Chiclayo, mayo de 2017

SUSTENTACIÓN DE TRABAJO DE SUFICIENCIA PROFESIONAL

TITULO:

"IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA EL SEGUIMIENTO DE PERSONAS MAYORES DE EDAD UTILIZANDO GEOLOCALIZACIÓN DESDE UN DISPOSITIVO ANDROID CONECTADO A UN SERVIDOR WEB CON MYSQL"

Presentado como requisito para optar el **Título Profesional de INGENIERO INFORMÁTICO Y DE SISTEMAS**, sustentado por:

Jose Waldir Maza Montes
Bachiller en Ingeniería Informática
y de Sistemas

Aprobado por los siguientes Miembros de Jurado:

Ing. Cristian Quesada Machado
PRESIDENTE

Ing. Eduardo Arrascue Becerra
SECRETARIO

Ing. Luis Aguilar Fernández
VOCAL

Fecha de Sustentación: Chiclayo, 19 de mayo de 2017

DEDICATORIA

El presente se lo dedico a mi dios quien supo guiarme por el buen camino, darme fuerzas para seguir adelante y no desmayar en los problemas que se presentaban, enseñándome a encarar las actividades sin perder nunca la dignidad ni desfallecer en el intento.

A mi familia quienes por ellos soy lo que soy. Para mis padres por su apoyo consejos, comprensión, amor, ayuda en los momentos difíciles, y por ayudarme con los recursos necesarios para estudiar. Me han dado todo lo que soy como persona, mis valores, mis principios, mi carácter, mi empeño, mi perseverancia, mi coraje para conseguir mis objetivos.

AGRADECIMIENTO

A mi familia porque siempre me ayudaron a lograr mis objetivos y metas propuestas, así como también ayudarme a no desmayar en situaciones adversas.

El agradecimiento total a la universidad de Chiclayo que me dio la bienvenida al mundo como tal, las oportunidades que me ha brindado son incomparables, y antes de todo esto ni pensaba que fuera posible que algún día siquiera me topara con alguna de ellas.

A la escuela de ingeniería informática de sistemas por siempre impartirme conocimientos actualizados que nos hacen profesionales competitivos y sobre todo humanos en la sociedad.

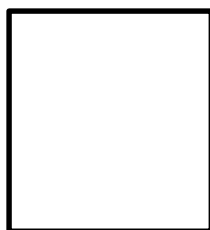
DECLARATORIA DE AUTENTICIDAD

Yo, José Waldir Maza Montes, Bachiller en Ingeniería Informática y de Sistemas de la Universidad Particular de Chiclayo, identificado con DNI N° 48479996, con el trabajo de suficiencia profesional titulado "Implementación de una aplicación web para el seguimiento de personas mayores de edad, utilizando geolocalización desde un dispositivo Android conectado a un servidor web con Mysql".

declaro bajo juramento que:

- 1) El trabajo de suficiencia profesional es de mi autoría que no ha sido previamente presentado para ningún grado o título profesional.
- 2) He respetado las normas internacionales de citas y referencias para las fuentes consultadas. Por tanto, el trabajo de suficiencia profesional no ha sido plagiado ni total ni parcialmente.
- 3) Los datos presentados en los resultados son reales, no han sido falsificados, ni duplicados, ni copiados.

Por lo expuesto, mediante la presente declaración cedo derecho de propiedad intelectual correspondiente a este trabajo, a la Universidad Particular de Chiclayo, según lo establecido por la ley de propiedad intelectual, por su reglamento y normatividad vigente.



Huella

José Waldir Maza Montes
Bachiller en Ingeniería Informática
y de Sistemas

Chiclayo, mayo de 2017

PRESENTACIÓN

El presente trabajo de investigación denominado: **"Implementación de una aplicación web para el seguimiento de personas mayores de edad, utilizando geolocalización desde un dispositivo Android conectado a un servidor web con Mysql"**, busca brindar apoyo tecnológico a aquellas personas que en algún momento necesiten conocer la ubicación exacta de una persona, para ello se hace uso de la nueva tecnología de sistemas operativos para dispositivos móviles "Android"; así como también el sistema de gestión de base de datos relacional "MySQL" y por supuesto el servidor de aplicaciones de mapa "google Maps".

Para la elaboración de este proyecto, se ha adoptado dos tipos de investigación: La descriptiva y aplicada. Descriptiva, pues con este tipo de investigación se pretende encontrar distintos factores que ayuden a determinar el número de casos que existe sobre la pérdida de personas mayores de edad, especialmente las que padecen de enfermedades mentales. Para ello, se realizó una encuesta para saber la frecuencia en que se llegan a perder. Y Aplicada, porque se utilizará tecnología Android para la geolocalización de las personas a seguir, por lo general los mayores de edad.

El presente trabajo se ha estructurado en ocho capítulos, con sus referencias bibliográficas y anexos.

Capítulo I: Contiene una breve introducción sobre el tema de la nueva tecnología que se está dando en los dispositivos móviles, la realidad problemática que presenta la sociedad con la perdida de personas, en su gran mayoría, de la tercera edad; así como también la formulación del problema con su respectiva hipótesis y, por último, el objetivo general del proyecto con sus objetivos específicos.

Capítulo II: Describe las bases teóricas del proyecto, lo cual hace referencia a los antecedentes de la investigación, el marco teórico que hace una

descripción de la tecnología android; así como también de los servicios web que existen en internet. Finalmente se cierra este capítulo con una definición de términos, los cuales se encontraron en la parte teórica.

Capítulo III: Se presenta el marco metodológico, en el cual se especifican las variables que intervienen en la investigación, con su respectiva operacionalización, así como la metodología empleada. Se incluyen los tipos de estudios realizados para determinar el número de casos que existe sobre la pérdida de personas mayores de edad, y la frecuencia con que estos se dan en la sociedad. Posteriormente, se describe el diseño que tiene la investigación. La población, muestra y muestreo que intervienen en el proceso de la investigación. Las técnicas e instrumentos de recolección de datos que se utilizaron para el desarrollo del proyecto. El método de análisis de datos y finalmente los aspectos éticos.

Capítulo IV: Corresponde a los resultados obtenidos en el análisis de los datos realizados a las personas que participaron de la encuesta. Puesto que 80 de los 100 encuestados que formaron parte de la muestra eran de la provincia de Chiclayo.

Capítulo V: Hace referencia a la discusión que se obtiene de la investigación, la cual nos permite verificar la hipótesis planteada sobre la mejora de la seguridad de la red

Capítulos VI y VII: Se definen las conclusiones y recomendaciones del presente trabajo de investigación.

En el capítulo VIII, se presenta la propuesta de Implementación de la aplicación web, las funciones aplicadas, tablas implementadas, así como la instalación de la aplicación que se implementó para lograr el objetivo principal.

Y por último los capítulos IX Referencias Bibliográficas y X Anexos.

ÍNDICE GENERAL

DEDICATORIA.....	3
AGRADECIMIENTO.....	4
DECLARATORIA DE AUTENTICIDAD	5
PRESENTACIÓN	6
ÍNDICE GENERAL	8
ÍNDICE DE FIGURAS	11
INDICE DE TABLAS	12
Resumen.....	13
Abstract	14
I. INTRODUCCIÓN	15
1.1 Realidad Problemática.....	17
1.2 Formulación del problema	17
1.3 Hipótesis	17
1.4 Objetivos	17
1.5 Limitaciones de la Investigación	18
1.6 Importancia	18
1.7 Justificación.....	18
II. MARCO TEÓRICO	19
2.1. Antecedentes de la Investigación.....	20
2.1.1 Antecedentes Locales	20
2.1.1 Antecedentes Internacionales	21
2.2 Marco Teorico	26
2.2.1 Android.....	26
2.2.2 Historia de Google Inc.	44
2.2.3 Origen del buscador Google	44
2.2.4 Otros productos de Google Inc.	47
2.2.5 Críticas a Google Inc.	50
2.2.6 MySQL	51
2.2.7 Dispositivo móvil.....	52

2.2.8 PDA 's	53
2.2.9 Smartphone.....	54
2.2.10 El Sistema de Posicionamiento Global (GPS)	55
2.2.11 Integración con telefonía móvil	57
2.2.12 Entorno de Desarrollo con Android Studio 2.1	58
2.3 Definición de Terminos	60
III. MARCO METODOLÓGICO	69
3.1 Variables.....	70
3.1.1 Variable Independiente:.....	70
3.1.2 Variable Dependiente:	70
3.2 Operacionalización de la variable.....	70
Tabla 3 Operacionalizacion de la Variable Dependiente	70
3.3 Metodología.....	72
3.4 Tipos de Estudio	74
3.5 Diseño.....	74
3.6 Población muestra y muestreo	74
3.6.1 Población	74
3.6.2 Muestra	75
3.6.3 Muestreo	76
3.7 Técnicas e instrumentos de recolección de datos.....	76
3.7.1 Técnicas.....	76
3.7.2 Instrumentos	77
3.8 Método de análisis de datos	77
3.9 Aspectos éticos	77
IV. RESULTADOS.....	78
V. DISCUSIÓN	81
VI. CONCLUSIONES	83
VII.RECOMENDACIONES	85
VIII. PROPUESTA	87
8.1 Generalidades	88
8.2 Análisis.....	88
8.3 Diseño.....	89
8.4 Implementación de la Aplicación.....	90

8.5 Costo y presupuesto	107
8.5.1 Bienes y servicios.....	107
8.5.2 Financiamiento	108
IX. REFERENCIAS BIBLIOGRÁFICAS	109
X. ANEXOS	112

ÍNDICE DE FIGURAS

FIGURA 1 Descargar de Android SDK.....	38
FIGURA 2 Selección de área de trabajo	39
FIGURA 3 Ejecución del Android SDK Manager.....	39
FIGURA 4 Paquetes del Android SDK Manager	40
FIGURA 5 Paquetes de instalación del Android SDK Manager	43
FIGURA 6 Extras del Android SDK Manager	44
FIGURA 7 Ejemplos de dispositivos móviles.	54
FIGURA 8 Entorno de Desarrollo Android Studio.....	58
FIGURA 9 Componentes del Android Studio	59
FIGURA 10 Detalle de los componentes del Android Studio	60
FIGURA 11 Tiempo de Actualización Aplicación.....	89
FIGURA 12 Confirmación del objetivo localizado.....	89
FIGURA 13 Google Maps.....	90
FIGURA 14 Acceso a la aplicación	104
FIGURA 15 Descarga de la aplicación.....	104
FIGURA 16 Aplicación descargada.....	105
FIGURA 17 Configuración de permisos de instalación	105
FIGURA 18 Fin de instalación.	106

INDICE DE TABLAS

Tabla 1 Versiones de las plataformas Android	32
Tabla 2 Versiones de las plataformas Android	33
Tabla 3 Operacionalizacion de la Variable Dependiente	70
Tabla 4 Operacionalizacion de la Variable Dependiente	71
Tabla 5 Tabla Resultados.....	79
Tabla 6 Tabla Resultados.....	80
Tabla 7 Tabla de Geolocalización	103
Tabla 8 Tabla de índices	103
Tabla 9 Tabla bienes y servicios	107
Tabla 10 Tabla otros	107
Tabla 11 Tabla equipo de desarrollo	108
Tabla 12 Tabla resumen del presupuesto	108

Resumen

El presente trabajo de investigación es denominado: **"Implementación de una aplicación web para el seguimiento de personas mayores de edad, utilizando geolocalización desde un dispositivo Android conectado a un servidor web con Mysql"**, el cual es de suma importancia para aquellas familias usuarias de dispositivos móviles que deseen realizar un seguimiento a uno de sus integrantes que por encontrarse en edad avanzada se llegan a perder, y de esta manera conocer la ubicación exacta de la persona.

Asimismo, la información que se obtiene por parte de las autoridades policiales acerca de la ubicación de la persona que se perdió no es precisa, lo cual ocasiona pérdida de tiempo y malestar en los familiares que de alguna manera u otra sienten angustia al no saber nada de su familiar.

Con la implementación de una aplicación web, utilizando geolocalización desde un dispositivo móvil se realizará el seguimiento de personas mayores de edad, permitiendo a la vez saber con exactitud el paradero de la persona, y por lo consiguiente obtener una búsqueda rápida de dicha persona.

PALABRAS CLAVES:

Aplicación web, geolocalización y android.

Abstract

The present work of investigation is denominated: "Implementation of a web application for the follow-up of people of legal age, using geolocation from an Android device connected to a web server with Mysql", which is a sum of importance for the user families of Mobile devices that wish to track one of their members who, due to their advanced age, becomes lost, and in this way know the exact location of the person.

Also, the information that can be obtained by the police authorities about the location of the person who was lost is not accurate, which occasionally was the loss of time and the type of life in which others from your relative.

With the implementation of a web application, using geolocation from a mobile device, you can track people of legal age, allowing the next time, with the whereabouts of the person, and why get a quick search of that person.

KEYWORDS:

Web application, geolocation and android.

CAPITULO

I. INTRODUCCIÓN

INTRODUCCIÓN

Hoy en día la tecnología está avanzando de forma desmesurada, donde cada vez la sociedad se vuelve más dependiente de ella, permitiendo a las familias tener facilidades para la protección de las personas mayores, ya que existen casos en que adultos mayores se llegan a perder o sentir mal.

El presente trabajo se centra principalmente en el desarrollo de una aplicación Android que está relacionada básicamente en una APK que se instala en un Smartphone con GPS activado y que, con conexión a internet, en cada cierto intervalo de tiempo, envía al servidor web la ubicación de una persona. Los datos emitidos se toman y mostrará su ubicación a través del google maps en tiempo real.

Asimismo, se destaca la importancia de esta aplicación web, debido a que el manejo es sencillo y además porque utiliza un dispositivo que hoy en día cualquier persona de distinto nivel socioeconómico del país lo puede adquirir.

Este proyecto busca despertar el interés de otros investigadores que desarrollen aplicaciones de interés social, profesional e individual y el cual pueda ser distribuido en forma gratuita o pagando un derecho en la tienda virtual conocida como play store.

Por ello, se ha propuesto la realización del presente trabajo cuyo objetivo principal es implementar una aplicación web que permita obtener el seguimiento de las personas mayores de edad, por intermedio de las facilidades que brinda google maps, el cual es completamente gratuito, pero enlazándolo con una aplicación sencilla, económica que permitirá la ubicación exacta de una persona en cualquier lugar del mundo siempre y cuando esté conectado a internet.

1.1 Realidad Problemática

En la actualidad son muchas las familias que por medidas de seguridad y protección brindan a las personas adultas un teléfono móvil para conocer su estado, necesidades o saber la ubicación de ellos. Sin embargo, en muchos casos las personas denominadas adultos mayores tienen dificultad para indicar en el momento indicado su ubicación exacta y sobre todo en algunos casos tienen dificultad para explicar verbal u oralmente con precisión dicha ubicación.

1.2 Formulación del problema

¿De qué manera la implementación de una aplicación web utilizando geolocalización desde un dispositivo Android permitirá el seguimiento de personas mayores?

1.3 Hipótesis

Implementando una aplicación web utilizando geolocalización desde un dispositivo Android conectado a un servidor web con mysql permitirá la localización de personas mayores de edad

1.4 Objetivos

1.4.1 Objetivo General

Implementación de una aplicación web para el seguimiento de personas mayores de edad utilizando geolocalización desde un dispositivo Android conectado a un servidor web con mysql.

1.4.2 Objetivo Específicos

- a) Determinar las acciones específicas para el desarrollo de la aplicación en el Android Studio.
- b) Implementar el servidor web para el almacenamiento de los datos en mysql.

- c) Implementar la aplicación en plataforma Android que permita el retorno de los datos posicionales de una persona.
- d) Utilizar una aplicación que permita visualizar los datos capturados de ubicación en forma gráfica.

1.5 Limitaciones de la Investigación

Se empleará como dispositivo móvil el Smartphone, porque es en la actualidad el artículo que puede ser adquirido por personas de cualquier nivel socio económico.

1.6 Importancia

Permitirá saber exactamente donde se encuentra en un momento determinado las personas adultas que requieren atención, cuidado y seguimiento por problemas de salud, edad y responsabilidad.

1.7 Justificación

a. A nivel Personal.

Investigar sobre las funciones del google respecto a la visualización de datos.

b. A nivel Académico.

Contribuir con la universidad en el cumplimiento y elaboración de una tesis relacionada con la investigación tecnológica

c. A nivel Económico.

El desarrollar aplicaciones web en plataforma android permite aprovechar los beneficios de las tiendas de ventas virtuales como Play Store

CAPITULO

II. MARCO TEÓRICO

2.1. Antecedentes de la Investigación

2.1.1 Antecedentes Locales

Título: “Diseño e Implementación de un sistema de geolocalización en interiores para la plataforma Andorid via la red Enterprise WLAN de la PUCP”

Año: 2016

Lugar: Lima, Perú

Autor: Gáloc Vilca, Javier Eduardo

Resumen:

El trabajo realizado en la presente tesis consiste en el desarrollo e implementación de una aplicación en Android que sea capaz de geolocalizar a un usuario a través de su dispositivo móvil dentro de los pabellones del campus PUCP. La geolocalización es realizada a través de técnicas de localización y algoritmos matemáticos de estimación. En el primer capítulo se da un vistazo a la evolución de la geolocalización y las plataformas tecnológicas móviles; también se ven algunos ejemplos de aplicaciones móviles actuales que utilizan la geolocalización en interiores (indoor); por ejemplo: Descubre PUCP y su problemática, que es vital para el objetivo principal de este proyecto. El segundo capítulo aborda la parte de ingeniería: se describen y evalúan las técnicas de localización indoor más comunes con sus pros y contras; haciendo énfasis en el método fingerprinting y el estimador Maximum Likelihood. También se describen las tecnologías de radiofrecuencias más usadas en los métodos de localización, haciendo énfasis en la tecnología WiFi y las principales características relativas a la propagación de su señal. En el tercer capítulo se lleva a cabo el análisis de la solución a implementar, se elige el diseño del sistema y se detallan los pasos que se siguen para su implementación. Con el análisis y el diseño hechos, se procede a la implementación de la solución que incluye la integración entre Android y

Matlab. Finalmente, en el cuarto capítulo se evalúa el desempeño de la solución implementada, se examinan los resultados de las pruebas en campo para hallar el error y se analiza cómo disminuirlo.

2.1.1 Antecedentes Internacionales

Título: “Diseño e Implementación de una Aplicación para Dispositivos Android en el Marco del Proyecto Pequeñas y Pequeños Científicos de la Universidad Politécnica Salesiana”

Año: 2012

Lugar: Cuenca, Ecuador

Autor: Quisi Peralta, D. F

Resumen:

Se ha obtenido un gran conocimiento sobre desarrollo en dispositivos móviles basados en Android, así como su arquitectura, características, componentes y funcionamiento. Cabe resaltar que el desarrollo en Android es muy parecido al desarrollo en lenguaje Java, ya que incluye algunos APIs de este lenguaje, por ello el aprendizaje y el desarrollo se realizó de forma óptima.

Dentro del desarrollo del presente proyecto se investigaron varios procesos de conversión de archivos de formato en flash (SWF) a imágenes animadas (GIF), el principal problema es que el archivo generado era 7 veces mayor que el original.

De igual forma, se probó realizar conversiones a videos y el resultado fue similar, teniendo archivos 5 veces mayores que el original. Por último, se realizaron conversiones a formato HTML5, siendo la que mejor resultados dio, para ello se utilizó herramientas de conversión como Swiffy de Google que aun está en versión de prueba y retorna buenos resultados en tamaño de espacio en disco, pero con información encriptada, por lo que no se puede modificar. En consecuencia, ninguna de las técnicas antes expuestas cumplía con los requerimientos mínimos para la ejecución de la

aplicación en los dispositivos móviles, ya que los mismos tienen menor capacidad en hardware.

Una de las principales razones por la que se desarrolló la aplicación en Android, es su alto nivel de utilización en plataformas celulares y tablets, en consecuencia, se permite que se pueda distribuir, modificar y estudiar sin limitaciones, ya que es un sistema completamente libre y a la vez el desarrollador puede decidir sobre su aplicación, si desea publicarla comercialmente o gratuitamente y los lugares en donde se distribuirá.

Se realizó una completa investigación de desarrollo para dispositivos móviles, encontrándose que se pueden implementar aplicaciones utilizando diferentes tecnologías web como HTML5, CSS3 y JavaScript y con ayuda de frameworks como JQuery Mobile, Sencha, PhoneGap, etc. Estas tecnologías se tomaron como primer punto de partida, con el objetivo de facilitar el desarrollo de aplicaciones y hacerlas multiplataforma, por ello se decidió desarrollar el módulo del mundo eléctrico en esta tecnología, pero durante su desarrollo se produjeron grandes limitaciones, específicamente al momento de representar la acción de movimiento de los personajes, lo cual no cumplió con las expectativas del equipo de Pequeñas y Pequeños Científicos. En virtud de ello, esta tecnología aún sigue en un proceso de desarrollo, por lo que los dispositivos actuales no contienen las características necesarias para una perfecta ejecución de las mismas ni en hardware ni en software, pero esta brecha se está reduciendo por motivos de avance y actualizaciones tecnológicas.

Título: “GPSLoc. Localización y Relaciones Sociales en el ámbito de los Teléfonos Inteligentes”.

Año: 2011

Lugar: Valencia

Autor: Sáez Sáez, P.

Resumen:

En esta tesina del Master de Ingeniería del Software, Métodos Formales y Sistemas de Información se ha presentado la aplicación GPSLoc, sus funcionalidades, su arquitectura, las tecnologías utilizadas, los problemas encontrados durante el desarrollo y las posibles mejoras de futuro. GPSLoc es una aplicación joven pero potente, que ofrece la posibilidad de mantener localizadas a otras personas, según cierto tipo de criterios, ya sean amigos, familiares, empleados de una empresa o tengan cualquier otro tipo de relación con el usuario de la aplicación. Para conseguir esto, la aplicación implementa tres tipos de permisos de visibilidad: permiso total (posición del usuario visible a cualquier hora del día y cualquier día de la semana), permiso por horas (posición del usuario visible solo durante un rango de horas y, si así se indica, fines de semana incluidos) y permiso por días (posición del usuario visible solo durante un rango de días). GPSLoc ofrece una amplia gama de funcionalidades atractivas para el usuario, desde las más comunes como registrarse, log-in, mostrar a los amigos en un mapa, seleccionar una foto de perfil visible por los demás usuarios, llamar por teléfono a un usuario o borrar permisos, hasta una nueva funcionalidad innovadora, poder observar las últimas posiciones en las que ha estado una persona, y no solo su última posición. La arquitectura escogida para la aplicación ha sido una arquitectura de 3 capas común, con la particularidad de que, debido a limitaciones de Android, ha sido necesaria la utilización de un puente intermediario para conectar la capa de lógica y la capa de persistencia. Este puente está construido mediante las tecnologías PHP y JSON, las cuales realizan una sencilla y eficiente comunicación, consistente en dos secciones principales: la parte ejecutada en el servidor remoto y la parte Android. La parte del servidor remoto consiste en ficheros PHP que se conectan a la base de datos, realizan una consulta y devuelven los resultados codificados en formato JSON (la codificación a este formato permite obtener y enviar los datos de forma segura, fluida y rápida). La parte

Android se conecta con los PHP para obtener los datos y transforma la respuesta JSON a tipos de dato estándar, para poder trabajar con ellos. Además, para optimizar la velocidad de la aplicación se ha optado por incorporar una BD Local SQLite, la cual almacena en el dispositivo móvil los amigos sobre los que tiene permiso el usuario y el propio permiso que tiene sobre los mismos. No se ha optado por almacenar en local las posiciones, ya que la naturaleza de esta aplicación conlleva poder visualizar la posición de los usuarios en tiempo real, y para ello, lo más adecuado es descargar las posiciones cada vez que se muestre al amigo o amigos en un mapa, y no cargar posiciones antiguas almacenadas en local. Sin duda, una de las características más cuidadas y trabajadas en GPSLoc es la ejecución en segundo plano de las funcionalidades más importantes de la aplicación. Tal y cómo se ha comentado anteriormente, Android paraliza las actividades de las aplicaciones que quedan en segundo plano, y las va borrando aleatoriamente si hay escasez de memoria. Para solventar este inconveniente han sido implementados tres services, clases especiales de Android que continúan su ejecución en segundo plano y no son cerradas por el sistema operativo. Un service se dedica a captar la posición GPS del usuario y enviarla al servidor remoto, otro service se encarga de mantener la base de datos local actualizada regularmente y el último service se encarga de comprobar periódicamente si se han recibido invitaciones nuevas y, si es así, avisa al usuario mediante la barra de estado de Android con una notificación. La elección de Android como sistema operativo destino para la aplicación ha resultado ser un acierto, en mi opinión personal. Cuando tomé las riendas de la tesina, en septiembre de 2010, Android era un sistema operativo nuevo, en constante evolución, y hoy es ya una realidad, habiendo desbancado a Symbian en cuota de mercado. Además, aparecen nuevas versiones del sistema operativo constantemente, cada una de ellas con nuevas mejoras cada vez más importantes, y conservando la retro compatibilidad con las aplicaciones diseñadas para versiones anteriores del sistema operativo, lo

que asegura a los desarrolladores una gran amplitud de clientes potenciales. Finalmente cabe destacar que, gracias al desarrollo de la aplicación GPSLoc, he adquirido una valiosa auto-formación en programación para el sistema operativo Android, pero no solo eso, sino también experiencia en PHP, XML, SQL y servidores remotos. Durante todos estos meses, he ido encontrando obstáculos y barreras que inicialmente no podía superar, como la inclusión de fotos en los perfiles de usuario, o la conexión con el servidor remoto. Gracias al esfuerzo dedicado para superar las dificultades y poder implementar todas las funcionalidades de la aplicación, he adquirido una experiencia muy valiosa de cara al futuro.

Título: “Implementación De Una Aplicación Móvil En Sistema Android Para La Geolocalización De Lugares Y Aulas Con Tecnologías De Mapas Y Realidad Aumentada”

Año: 2015

Lugar: Guayaquil.

Autor: Suárez Cruz, S. G.

Resumen:

El crecimiento del uso de las tecnologías multimedia en los diferentes aspectos de la vida cotidiana ha hecho que herramientas como la realidad aumentada (RA) sea usada como un medio para mejorar la colaboración entre un entorno real y uno virtual. Los campos de aplicación de la RA son diversos, uno de ellos y uno de los más utilizados es la geo localización de lugares. Debido a esto, el objetivo del presente proyecto es realizar un análisis exhaustivo de la viabilidad de aplicar RA y los mapas en tecnologías móviles, a través del desarrollo de una aplicación móvil en el sistema operativo Android para la facultad FIEC, para identificar los beneficios del uso de la RA en aplicaciones de este tipo. Durante la etapa de desarrollo se procedió a implementar un modo Mapa dentro de la aplicación para rutas internas usando los servicios de Google Maps y un

visualizador para consultar información sobre las edificaciones en 3D de la FIEC. Con la finalidad de mejorar la experiencia de los usuarios durante el uso de la aplicación se agregó como información adicional detalles de los diferentes horarios de las aulas disponibles en los edificios de FIEC; dicha información está integrada con el Centro de Servicios Informáticos (CSI), esto les permite obtener en tiempo real sus horarios y la ubicación de sus clases. Finalmente, en la etapa de prueba se realizó una encuesta de campo con el objetivo de medir la utilidad de la geo localización en este tipo de aplicaciones. Además, la retroalimentación recibida a través del estudio de campo, puede ser utilizada en estudios futuros con la finalidad de mejorar la experiencia del usuario en aplicaciones de este tipo.

2.2 Marco Teorico

2.2.1 Android

Antes de sumergirse plenamente en el universo Android, es preciso presentar la plataforma.

Vamos a descubrir aquí los orígenes y la historia de la plataforma Android, sus distintas versiones y su distribución actual y pasada en los dispositivos Android. Descubriremos, por último, la arquitectura de la plataforma Android sobre la que se ejecutan las aplicaciones.

▪ Open Handset Alliance™

Comprada por Google en 2005, Android era inicialmente una startup (empresa joven) que desarrollaba un sistema operativo para dispositivos móviles.

Desde entonces, varios rumores anunciaban la aparición de un teléfono Google llamado Gphone. No obstante Google preparaba, de hecho, mucho más que eso.

El 5 de noviembre de 2007 se anunció la creación del OHA (Open

Handset Alliance): un consorcio creado por iniciativa de Google que reunía en sus inicios a una treintena de empresas. La mayoría de empresas eran operadores móviles, fabricantes de dispositivos, empresas industriales y fabricantes de software. El rol del OHA era favorecer la innovación en los dispositivos móviles proporcionando una plataforma realmente abierta y completa.

El mismo día, el OHA presenta oficialmente Android: la primera plataforma completa y abierta para dispositivos móviles. Esta plataforma incluía un sistema operativo, el middleware (las aplicaciones intermedias), una interfaz de usuario y aplicaciones de referencia.

Algunos días más tarde, el 12 de noviembre de 2007, el OHA anuncia la aparición del primer SDK (Software Development Kit o kit de desarrollo de software) Android que permite a los desarrolladores crear sus propias aplicaciones para la plataforma Android.

A principios de 2014, el OHA agrupa a más de ochenta empresas miembros.

▪ **Historia**

El 21 de octubre de 2008, Google y el OHA anuncian la publicación del código fuente de la plataforma Android en open source, bajo licencia Apache 2.0.

Desde entonces era posible descargar el código fuente del sistema Android, compilarlo, instalarlo y ejecutarlo. También era posible contribuir a su evolución. Todo el código fuente y la información se encuentran en el sitio web Android Open Source Project: <http://source.android.com/>

En noviembre de 2008 apareció el Android Market, la tienda de aplicaciones de Google que permite a los desarrolladores y fabricantes de software poner a disposición de cualquier cliente de Android sus aplicaciones. En marzo de 2012, bajo la óptica de unificar sus servicios, Google renombra su tienda de aplicaciones como "Play Store". Esta

tienda está disponible directamente en los dispositivos Android a través de la aplicación Play Store, y también en el sitio de Internet en la siguiente dirección <https://play.google.com/store>.

Inicialmente la versión web sólo permitía consultar algunas aplicaciones de referencia. Más adelante se han incluido numerosas evoluciones, y en lo sucesivo Play Store permite consultar todas las aplicaciones disponibles, además de presentarle al usuario conectado la lista de aplicaciones instaladas en sus distintos dispositivos. También es posible instalar una aplicación sobre un dispositivo desde la aplicación web, desde un ordenador.

El primer Smartphone Android apareció en octubre de 2008 en los Estados Unidos y en marzo de 2009 en España. Fue el HTC Dream G1. Su particularidad consistía en poseer un teclado físico deslizante. Su pantalla ocupaba 8 cm de diagonal y poseía una resolución de 320 x 480 píxeles. Contaba con 192 MB de memoria RAM, 256 MB de ROM, una cámara digital sin flash de 3,1 mega- píxeles, Wi-Fi y 3G.

Más adelante, muchos fabricantes han sacado al mercado un número impresionante de dispositivos que funcionan con Android. A principios de 2014, la empresa Samsung, líder del mercado en dispositivos Android en dicha fecha, poseía más de veinte modelos diferentes de Smartphone y once modelos de tabletas. Según las estadísticas presentadas por Play Store existirían, actualmente, más de cinco mil dispositivos diferentes (contando todas las variaciones de cada fabricante) capaces de conectarse a Play Store.

Además de los Smartphone y las tabletas (la primera tableta apareció a principios de 2011), existen dispositivos intermedios llamados "tabléfonos" (phablet en inglés) en los catálogos de los fabricantes. Estos dispositivos, cuyo nombre es la contracción entre teléfono inteligente (Smartphone, en inglés) y tableta (Tablet, en inglés), disponen de pantallas de gran dimensión (típicamente 6,5 pulgadas, unos 16 cm), están equipados con

una tarjeta GSM (para utilizar la red del teléfono móvil) y, en general, pueden utilizarse con el dedo o con un lápiz táctil.

Los últimos dispositivos están ahora equipados con la tecnología NFC (Near Field Communication) que les permite comunicarse en el corto alcance, como por ejemplo para realizar un pago sin contacto. Estudiamos la tecnología NFC para Android más adelante en este libro.

Se ha recorrido mucho camino tras la aparición de la plataforma Android. Cada semana supone otra tanda de nuevos Smartphone y tabletas.

¡Y sigue creciendo!

▪ **Versiones de Android**

Igual que con el imperio romano, Android no se ha construido en un día. Es por ello que la progresión en el número de funcionalidades es similar a la de su cuota de mercado, simplemente impresionante.

Estas funcionalidades, mejoras y correcciones de bugs han ido apareciendo con el tiempo en las sucesivas versiones.

Veamos a continuación cada una de estas versiones desde la aparición oficial de Android y su reparto actual en el conjunto de dispositivos Android.

▪ **Cronología de versiones**

En septiembre de 2008 apareció la primera versión de la plataforma Android, la versión 1.0. A continuación apareció la versión 1.1 en febrero de 2009; versión que se integró en el primer Smartphone Android comercializado en España en marzo de 2009.

Algunos meses más tarde, Android pasó directamente a la versión 1.5, versión mayor que apareció en abril de 2009. Le seguirían tres revisiones de esta versión: la 1.5r2, la 1.5r3 y la 1.5r4.

La versión 1.5 aportaba nuevas funcionalidades tales como la detección

de la rotación mediante un acelerómetro, la grabación de vídeo y la integración de los App Widgets (véase el capítulo Funcionalidades avanzadas - App Widget).

La versión 1.6 apareció en septiembre de 2009. Esta versión integró en particular el soporte de numerosas resoluciones y tamaños de pantalla distintos. También se integró la síntesis vocal. A continuación, aparecieron dos revisiones, la 1.6r2 y la 1.6r3.

En octubre de 2009 apareció la versión mayor 2.0, aportando el soporte de HTML5 y de Exchange.

En diciembre de 2009 apareció la 2.0.1 y, en enero de 2010, la 2.1 que integraba los fondos de pantalla dinámicos. A continuación, apareció la revisión 2.1r2.

Estas últimas versiones fueron rápidamente reemplazadas por la versión 2.2 que incluía la funcionalidad de hotspot (punto de acceso Wi-Fi) y supuso una mejora neta del rendimiento de 2 a 5 veces más rápido que la versión 2.1. Esta versión permite a su vez la instalación de aplicaciones en un almacenamiento externo como una tarjeta de memoria extraíble. Todavía apareció una revisión: la 2.2r2.

A finales de 2010 apareció la versión 2.3 que incluyó el NFC, soporte a varias cámaras, una interfaz

mejorada, mejor gestión de la energía, gestión de nuevos sensores, una mejora importante en el rendimiento, de la máquina virtual Dalvik, un teclado virtual más rápido con la funcionalidad copiar/pegar mejorada, soporte para VoIP (Voice over Internet Protocol) y SIP (Session Initiation Protocol).

La versión 2.3 sólo estuvo disponible en el smartphone Nexus S de Google.

Fue preciso esperar a la versión 2.3.3 para que los demás teléfonos, el

Nexus One entre otros, pudieran situarse al mismo nivel.

En febrero de 2011 apareció la versión 3.0. Esta versión marcó una etapa importante en el mundo Android. Estaba especialmente diseñada para tabletas táctiles con el objetivo de aprovechar sus pantallas de grandes dimensiones. Además de la integración de una nueva interfaz de usuario rica en nuevos componentes que incorporaba un nuevo tema gráfico, la versión 3.0 aportó el soporte de la arquitectura de procesadores de múltiples núcleos.

El año 2011 se publicaron varias versiones 2.3.x para smartphones, así como las versiones 3.1 (mayo) y 3.2 (julio).

En octubre de 2011 se publicó la versión 4.0, que unificaba los sistemas operativos para smartphones y tabletas. Las versiones 4.1 y 4.2 se publicaron, respectivamente, en julio de 2012 y octubre de 2012. La versión 4.3 apareció en julio de 2013, tras varias versiones 4.2.x.

La última versión mayor (a principios de enero de 2014), Android 4.4 Kitkat, se anunció el 3 de septiembre de 2013.

A estas versiones del sistema Android le corresponden niveles de interfaz de programación o simplemente API (Application Programming Interface) del SDK.

A continuación, aparece una tabla que muestra esta correspondencia.

VERSION DE LA PLATAFORMA ANDROID	FECHA	NOMBRE	NIVEL DE API
1.0	09/2008		1
1.1	02/2009		2
1.5	04/2009	Cupcake	3
1.5r2	05/2009		
1.5r3	07/2009		
1.5r4	05/2010		
1.6	09/2009	Donut	4
1.6r2	12/2009		
1.6r3	05/2010		
2.0	10/2009	Éclair	5
2.0.1	12/2009		6
2.1	01/2010		7
2.1r2	05/2010		
2.2	05/2010	Froyo	8
2.2r2	07/2010		
2.3	12/2010	Gingerbread	9
2.3.3	02/2011		10
3.0	02/2011	Honeycomb	11
3.1	05/2011	Honeycomb	12
3.2	07/2011		13
3.2.1	08/2011		13
3.2.2	09/2011		13
4.0.1	10/2011	Ice Cream Sandwich	15
4.0.2	10/2011		14

Tabla 1 Versiones de las plataformas Android

VERSION DE LA PLATAFORMA ANDROID	FECHA	NOMBRE	NIVEL DE API
4.0.3	12/2011		15
3.2.4	12/2011	Honeycomb	13
3.2.6	02/2012		13
4.0.4	03/2012	Ice Cream Sandwich	15
4.1	07/2012	Jelly Bean	16
4.1.1	07/2012		16
4.1.2	10/2012		16
4.2	10/2012		17
4.2.1	11/2012		17
4.2.2	02/2013		17
4.3	07/2013		18
4.3.1	10/2013		18
4.4	10/2013	KitKat	19

Tabla 2 Versiones de las plataformas Android

El sistema Android asegura a las aplicaciones una compatibilidad ascendente. Esto significa que una aplicación pensada para funcionar sobre una versión mínima de Android 1.6 (API 4) funcionará automáticamente sobre todas las versiones de Android 1.6 (API 4) y superiores y, por tanto, por ejemplo, sobre la versión 4.0.1 (API 14).

Reparto de las distribuciones Android

Cada quince días, el sitio web Android (<http://developer.android.com>) proporciona el reparto de las versiones de Android en los sistemas que han accedido a Play Store en las dos últimas semanas. También aparecen los datos históricos que permiten seguir la evolución en este reparto.

Esta información está disponible en la siguiente dirección:
<http://developer.android.com/resources/dashboard/platform-versions.html>

Estos datos ayudan al desarrollador a la hora de escoger, con toda la información disponible, la versión mínima sobre la que deberá ejecutar su aplicación. Gracias a la compatibilidad ascendente de Android, esta aplicación funcionará automáticamente en las versiones superiores.

Cuanto menor sea la versión mínima de Android requerida por la aplicación, mayor será el número de dispositivos, y en consecuencia el número de usuarios y de compradores potenciales, que podrán ejecutar la aplicación. En contrapartida, la aplicación dispondrá de menos APIs de SDK que las versiones más recientes. Es, por tanto, tarea del desarrollador encontrar el mejor compromiso entre las funcionalidades requeridas del SDK y la extensión del público objetivo.

En enero de 2014, la versión más extendida de Android es Android Jelly Bean que, contando todas las versiones secundarias, equipa al 60% del parque Android, 21% de los dispositivos ejecutan todavía Android Gingerbread.

Esta profusión de versiones diferentes, así como la cantidad de dispositivos con características diferentes (tamaño de pantalla, resolución de pantalla, memoria, etc.), resumido bajo el término "fragmentación", es el principal problema para los desarrolladores de Android. Veremos, a lo largo de este libro, cómo gestionar de la mejor forma esta fragmentación sin restringir las funcionalidades y el diseño de nuestras aplicaciones.

▪ Arquitectura

Tras la versión Jelly Bean, el sistema Android está basado en el núcleo de Linux 3.0, y las primeras versiones estaban basadas en el núcleo 2.6. Este núcleo tiene en cuenta la gestión de las capas inferiores, tales como los procesos, la gestión de la memoria, los permisos de usuario y la capa de hardware.

Sobre este núcleo, se sitúa la capa de bibliotecas principales del sistema proporcionadas por los fabricantes. Éstas, de bajo nivel, están escritas en C y/o C++. Proporcionan los servicios esenciales tales como la gestión de la visualización 2D y 3D, un motor de base de datos SQLite, la reproducción y la grabación de audio y vídeo, un motor de navegador web.

Las funcionalidades ofrecidas por estas bibliotecas las recoge y utiliza la capa superior bajo la forma de bibliotecas Java. Éstas proporcionan bibliotecas y componentes reutilizables específicos a dominios particulares. Encontramos, por ejemplo, bibliotecas de acceso a bases de datos, de telefonía, de localización geográfica, de comunicación sin contacto de corto alcance.

Por último, la capa de más alto nivel es la de las aplicaciones. Estas aplicaciones son las que se incluyen por defecto, tales como la aplicación de inicio (llamada a menudo escritorio), la aplicación que permite ejecutar otras aplicaciones, el navegador web, la aplicación de telefonía... Pero también son las aplicaciones específicas creadas por desarrolladores, ¡de los que usted formará parte muy pronto!

Si bien es posible desarrollar aplicaciones basadas en código C y/o C++ mediante el NDK (Native Development Kit), sobre todo para mejorar el rendimiento, no se abordará este tema en este libro. Utilizaremos únicamente las API Java que proporciona el SDK y que bastan, en la gran mayoría de casos, para crear todo tipo de aplicaciones estándar. Si se tratara de aplicaciones con un gran consumo de recursos gráficos tales

como juegos 2D o 3D, el escenario sería distinto.

Por defecto, cada aplicación se ejecuta en una máquina virtual Java alojada en un proceso Linux dedicado. Esta máquina virtual es específica a la plataforma Android y está especializada para entornos embebidos. Se le llama máquina virtual Dalvik.

Entorno de desarrollo

Incluso aunque es posible desarrollar completamente una aplicación mediante un editor de texto básico y una línea de comandos para realizar la compilación, resulta mucho más cómodo utilizar un entorno de desarrollo, que le facilitará la escritura, la compilación y la depuración de sus aplicaciones.

Existen varios entornos de desarrollo, cada uno con sus ventajas e inconvenientes. Veremos, en este capítulo, cómo instalar dos de ellos: Eclipse/ADT (Android Development Tools) y Android Studio.

A lo largo del libro, utilizaremos como referencia el entorno de desarrollo Eclipse, al que habremos agregado el plug-in ADT.

Veremos en el siguiente capítulo cómo instalar y configurar un emulador de Android, herramienta esencial para el desarrollador.

1. Requisitos previos

Para poder desarrollar aplicaciones Android, es preciso asegurarse de que el puesto de desarrollo es compatible con los criterios requeridos.

Sistemas operativos soportados:

- Windows XP (32 bits), Windows Vista (32 o 64 bits), Windows 7 (32 o 64 bits), Windows 10.
- Mac OS X 10.5.8 o superior (x86 únicamente).
- Linux (librería GNU C (glibc) 2.11 o superior; las distribuciones de 64 bits deben ser capaces de ejecutar aplicaciones de 32 bits).

Se requiere un mínimo de 200 MB de espacio en disco sólo para la instalación del SDK, de sus herramientas y de alguna versión de la plataforma Android.

Se requiere el JDK (Java Development Kit) de Java 6 o superior. Si el puesto de desarrollo todavía no está disponible, es posible descargar el JDK de la plataforma Java SE (Java Platform, Standard Edition) en la siguiente dirección:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

2. Eclipse y el Plug-in ADT

Eclipse es un entorno de desarrollo bien conocido en el universo Java. Gratuito, dispone de una importante colección de plug-in, y está indicado para el desarrollo de aplicaciones Android. Para ello, es necesario agregar el plug-in ADT, que integra numerosas funcionalidades específicas del desarrollo Android, que van desde el asistente para la creación de un proyecto Android hasta la exportación de la aplicación final.

Google provee una versión "Bundle" de Eclipse, que integra de manera nativa el plug-in ADT, sin requerir una instalación o configuración particular. Esta versión es la que vamos a instalar y utilizaremos a lo largo de este libro.

La descarga del bundle puede llevarse a cabo desde la siguiente dirección:<http://developer.android.com/sdk/index.html>

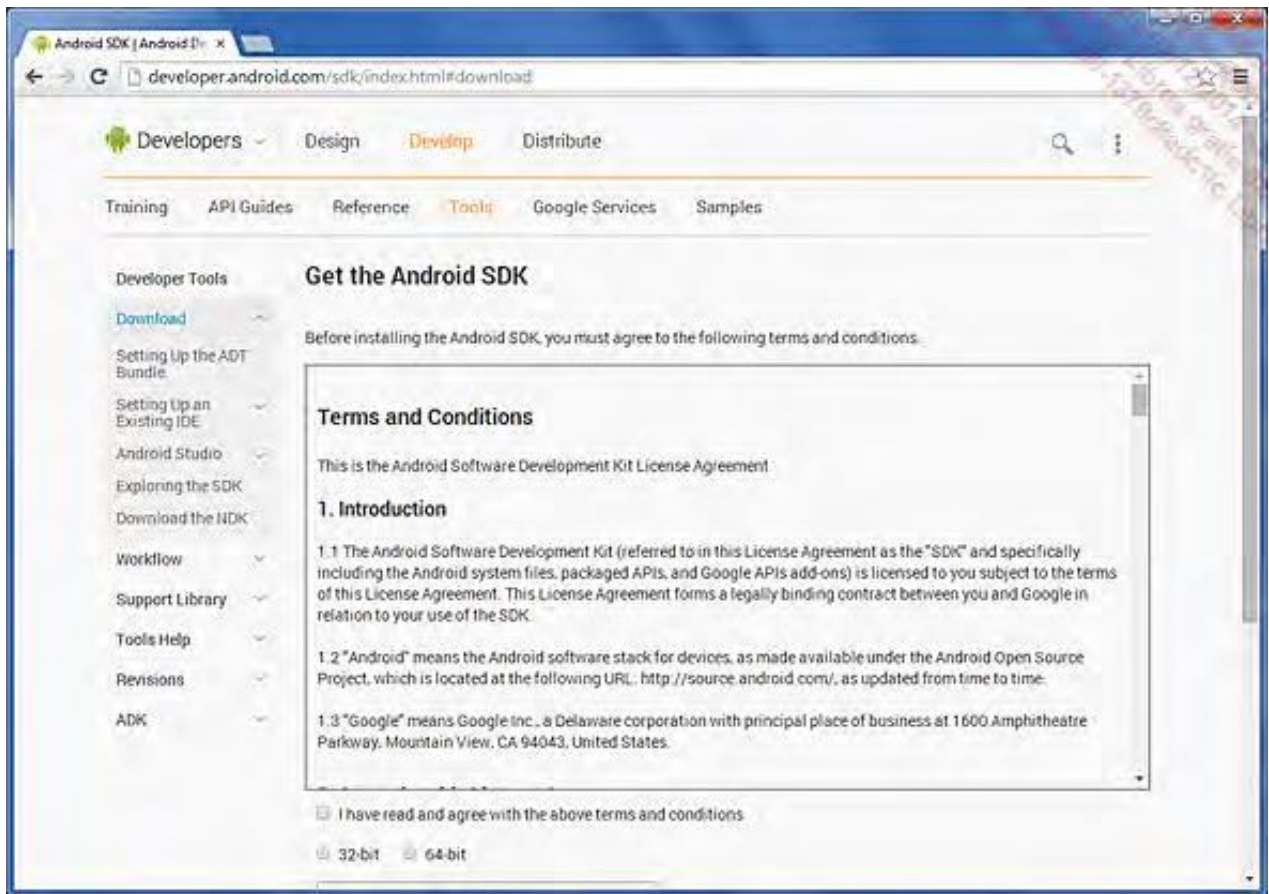


FIGURA 1 Descargar de Android SDK

Una vez descargado, basta con descomprimir el archivo distribuido en forma de archivo comprimido en una carpeta de su elección: recomendamos seleccionar una carpeta cuya ruta de acceso no incluya espacios.

La descompresión produce dos carpetas, "eclipse" y "sdk", así como un archivo ejecutable, "SDK Manager". Basta, ahora, con ubicarse en la carpeta "eclipse" y ejecutar la aplicación.

Tras la visualización de la pantalla de inicio "ADT", se abre un cuadro de diálogo que le invita a seleccionar la carpeta de trabajo en la que se almacenarán sus proyectos (workspace). Seleccione la carpeta de su elección, donde los proyectos se crearán como subcarpetas.

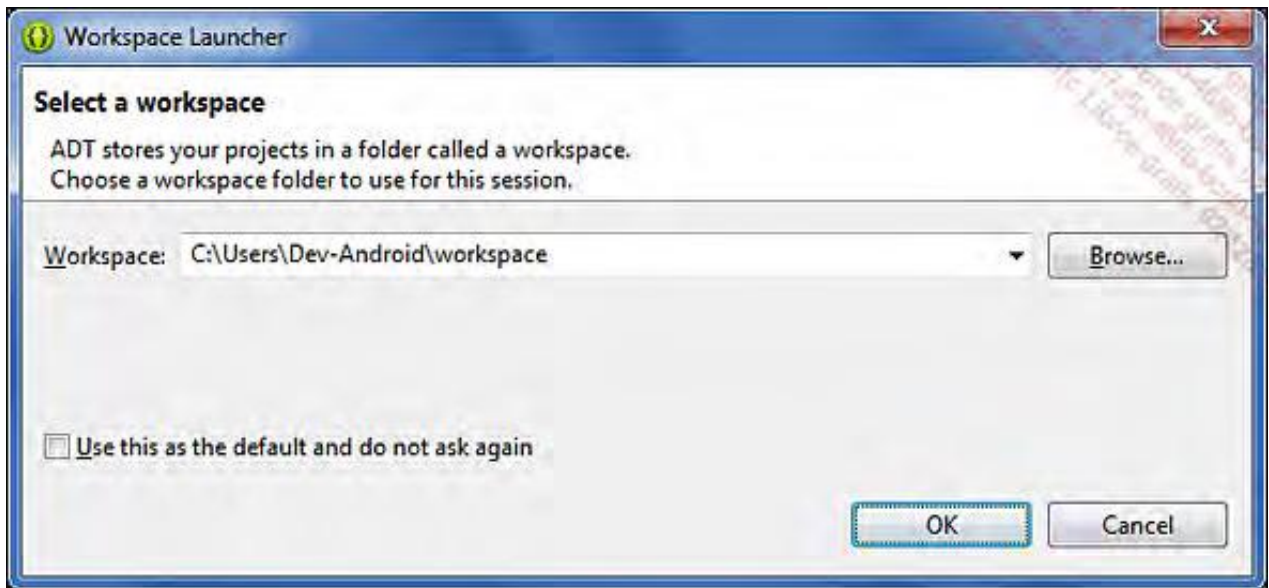


FIGURA 2 Selección de área de trabajo

Haga clic en O K. Se abre el espacio de trabajo por defecto de Eclipse.

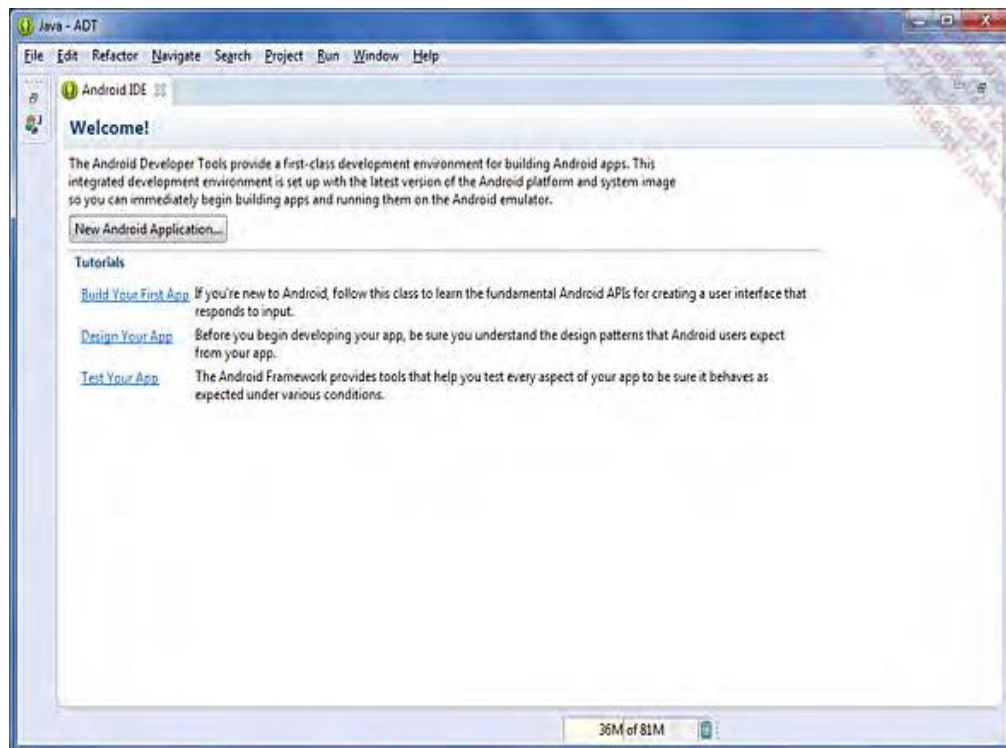


FIGURA 3 Ejecución del Android SDK Manager

Antes de crear un primer proyecto, vamos a ver cómo instalar varias versiones de Android, lo que es indispensable para poder probar una aplicación en diferentes configuraciones. Para ello, basta con ejecutar la aplicación Android SDK Manager desde la carpeta de instalación del bundle (también es posible ejecutar el SDK Manager a través de Eclipse, en el menú Window - Android SDK Manager, o mediante la barra de herramientas de Eclipse).

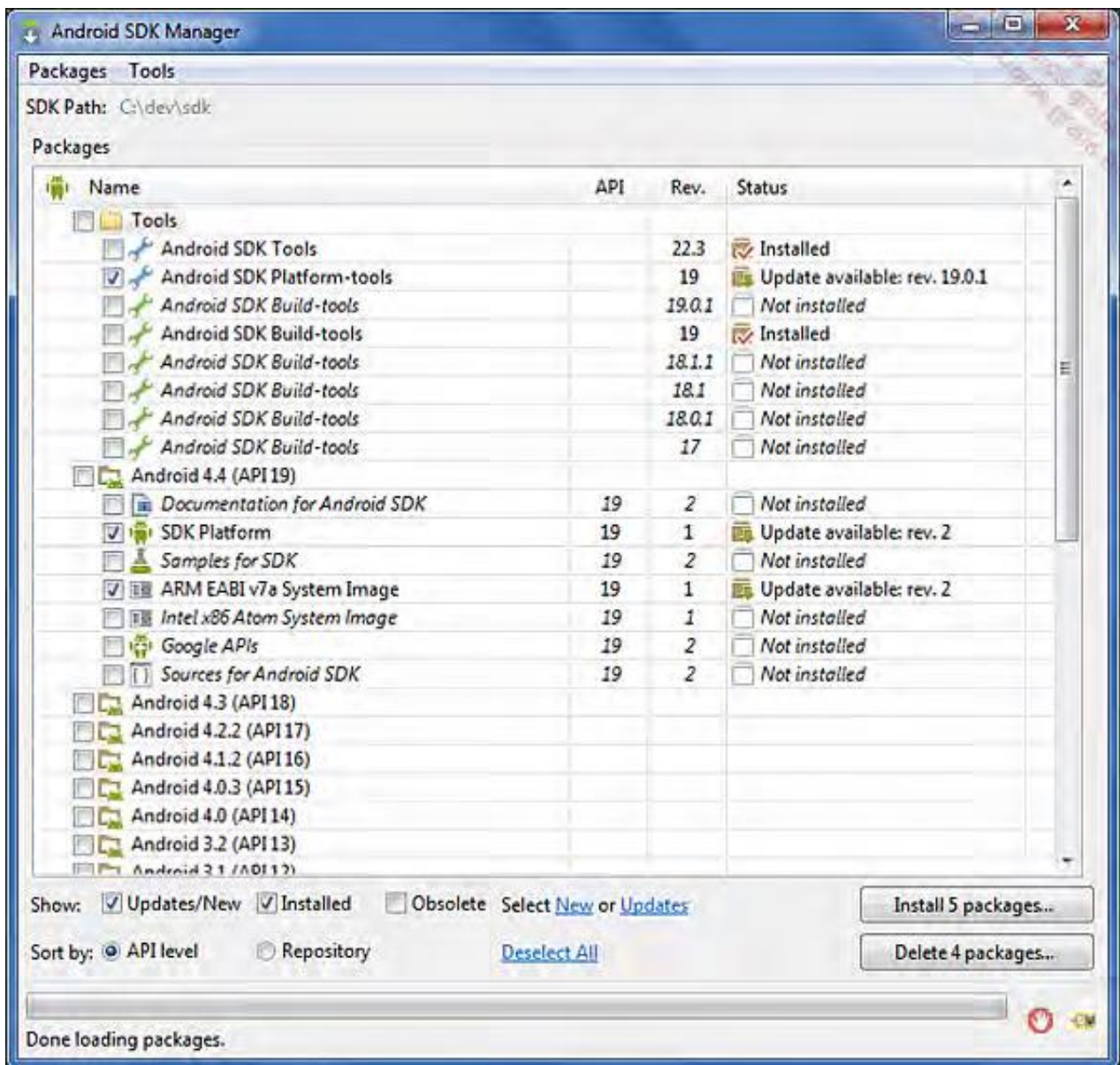


FIGURA 4 Paquetes del Android SDK Manager

La aplicación Android SDK Manager se conecta directamente con los servidores de Google para descargar la lista de paquetes disponibles. La pantalla de la aplicación muestra, a continuación, una lista de paquetes, organizados según la versión de Android afectada. Para instalar un paquete, o un elemento de un paquete, basta con marcar la opción correspondiente.

De arriba a abajo, los paquetes disponibles contienen:

- Android SDK Tools: la última versión de las herramientas dependientes de la plataforma Android.
- Documentation for Android SDK: la última versión de la documentación del SDK Android. Se trata de una copia local de la documentación disponible en línea.
- SDK Platform Android...: SDK de varias plataformas Android desde la versión 1.5 (API 3), cada una en su última versión. Observe que ciertas versiones, como la 2.0 (API 5) y la 2.0.1 (API 6), no están disponibles y han sido reemplazadas por la versión 2.1 (API 7).
- Samples for SDK API proyectos Android de ejemplo correspondientes a distintas versiones del SDK.

La categoría Extra contiene módulos complementarios y opcionales que permiten extender las funcionalidades del entorno de desarrollo o de las aplicaciones Android. Destacaremos los siguientes componentes:

- Google Play services: módulos que proporcionan funcionalidades suplementarias ofrecidas por la empresa Google tales como los mapas geográficos de Google Maps (véase el capítulo Sensores y geolocalización - Google Maps), la gestión de cuentas de tipo Gmail, interacciones con Google+, etc.
- Google Play Licensing Library: librería que permite proteger las aplicaciones de pago mediante la verificación en línea de la licencia

(véase el capítulo Funcionalidades avanzadas - Proteger las aplicaciones de pago). Esta funcionalidad implica que la aplicación esté publicada en la Play Store.

- Google Play Billing Library: librería que permite comprar contenido digital o funciones suplementarias, desde una aplicación. Este servicio se llama In-app Billing (pago integrado o in-app). Esta funcionalidad implica que la aplicación esté publicada en Google Play Store. Estudiaremos esta librería en el capítulo Funcionalidades avanzadas.
- Android Support Library: librería que ofrece ciertas funcionalidades aparecidas con la versión 4.x a las aplicaciones destinadas a versiones anteriores. Utilizaremos esta librería en el capítulo Componentes principales de la aplicación.

También es posible seleccionar la documentación del SDK para descargarla sobre el puesto local. De este modo, la consulta de esta documentación podrá realizarse directamente en local, mejor y más rápido que en línea. Ocurre lo mismo con los ejemplos. Es posible instalar y desinstalar cada uno de estos componentes en cualquier momento, a excepción de las herramientas Android SDK Tools que permiten, entre otros, administrar la instalación de los componentes y deben estar por tanto siempre presentes.

Se desaconseja instalar todos los paquetes disponibles: ralentizará considerablemente el inicio de la aplicación - cada paquete se carga durante el arranque de Eclipse. Es preferible seleccionar los paquetes correspondientes a las versiones principales de Android, e instalar más adelante un paquete concreto si fuera necesario. Recomendamos instalar las versiones Android 4.4 (API 19). Android 4.1 (API 16) así como la versión 2.3.3 (API 1)

- Marque las opciones correspondientes a los paquetes deseados.

- Una vez seleccionados los componentes, haga clic en el botón Install x packages.0).

Aparece una nueva ventana detallando los componentes seleccionados.

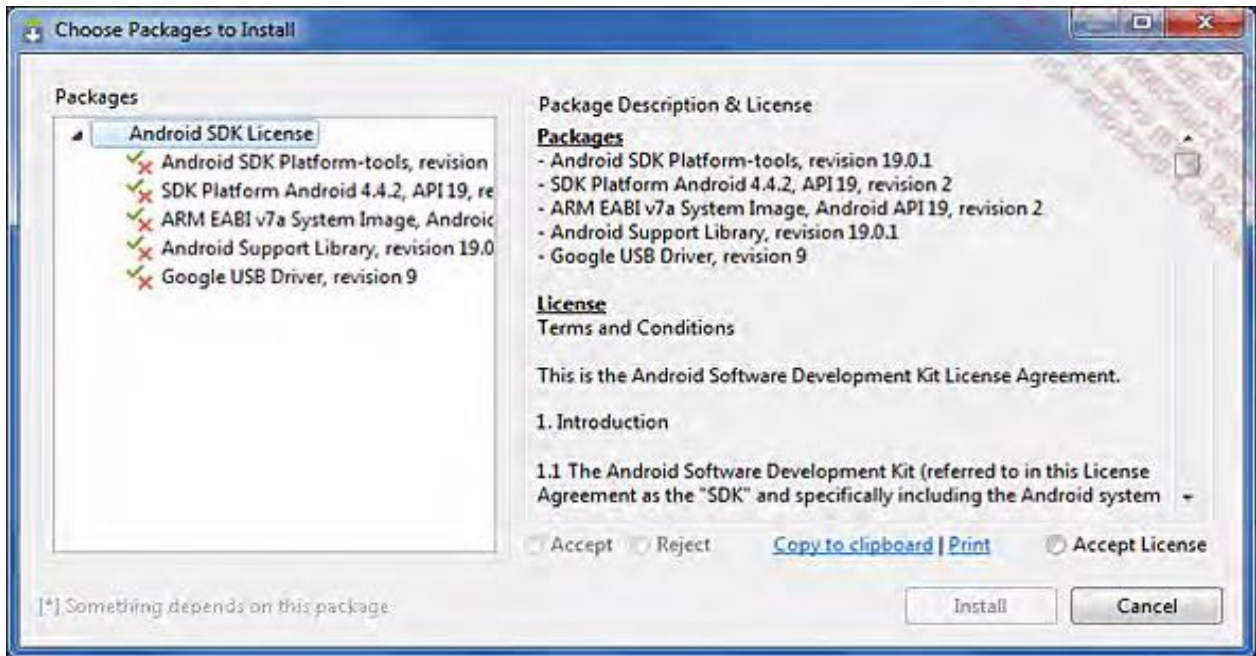


FIGURA 5 Paquetes de instalación del Android SDK Manager

- Marque la opción Accept License y haga clic en el botón Install para ejecutar la descarga y la instalación de los componentes. Dependiendo de su conexión a Internet y del número de paquetes seleccionados, esta operación puede resultar algo larga.
- Una vez finalizada la instalación, estos componentes deben aparecer en la lista de componentes instalados.

Para verificarlo, seleccione únicamente Installed en la opción Show de la ventana de Android SDK Manager para mostrar la lista de componentes instalados.

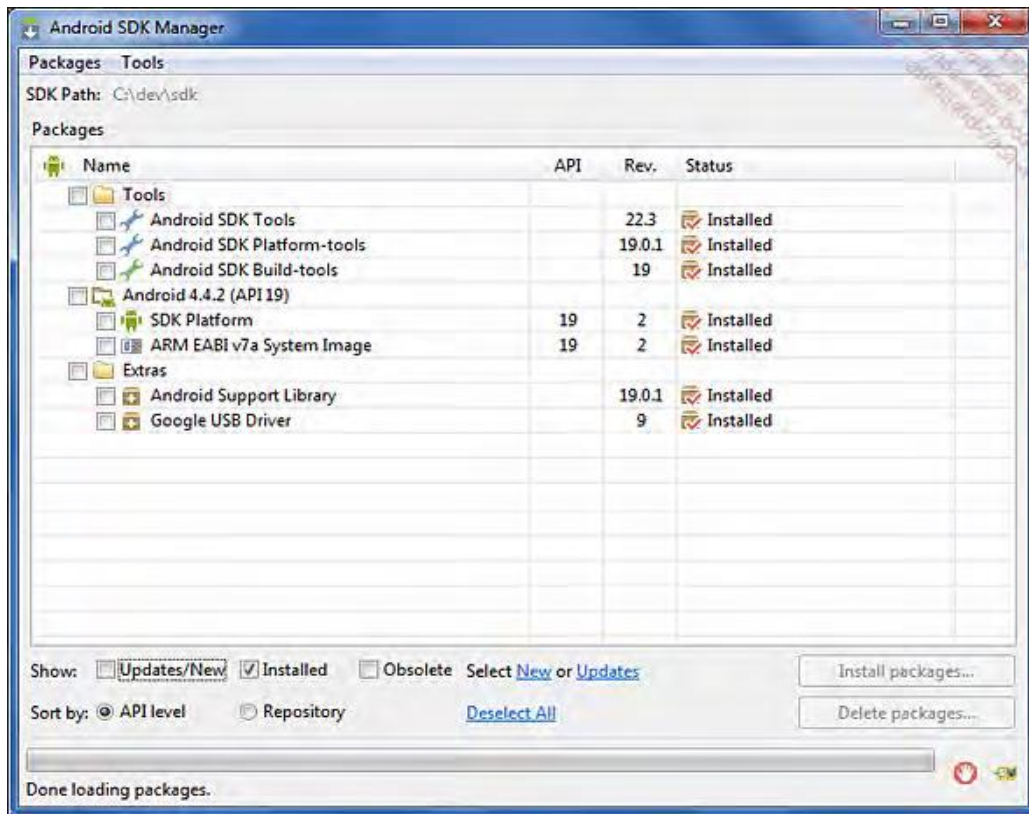


FIGURA 6 Extras del Android SDK Manager

Cierre a continuación la ventana.

¡Listo! El entorno de desarrollo Eclipse está preparado.

2.2.2 Historia de Google Inc.

El nacimiento y desarrollo de Google Inc. es una de esas tantas historias que parecen cumplir a rajatabla los estereotipos del “sueño americano”: una simple pero brillante idea que, con esfuerzo, talento y dedicación, se convierte desde cero en una realidad de éxito.

2.2.3 Origen del buscador Google

En 1995 dos estudiantes de la Universidad de Standford, Larry Page y Sergey Brin, descubren su común interés en averiguar alguna forma de poder conocer el nivel de importancia de una determinada página

web. En concreto, Larry Page trabajaba en una idea según la cual una página tenía mayor o menor importancia en la red en función del número de páginas que la enlazaran a través de links. La idea derivaba del mundo universitario en donde Page se movía, en el que la relevancia de un artículo académico venía dada por las veces que era citado a su vez en el trabajo de otros investigadores.

De este modo, Page creó un primer prototipo llamado BackRub, que rastreaba los sitios web y almacenaba los enlaces hallados en ellas en una base de datos. Cuando comenzó a realizar pruebas con BackRub, Page consiguió determinar cuántos enlaces había entre las páginas, pero no hallaba la forma de clasificarlas en importancia, ya que la compleja telaraña de enlaces que se formaba implicaba un considerable cálculo recursivo. Es en este punto donde entra Sergey Brin, licenciado en Informática y Ciencias Matemáticas, que andaba buscando un tema interesante para su tesis doctoral. Brin siempre había destacado por su habilidad matemática y se puso a trabajar junto con Page en elaborar un algoritmo que pudiera clasificar la importancia de los enlaces.

Fruto de esta colaboración nació PageRank, la evolución del anterior proyecto de Page, que además de buscar y almacenar enlaces era capaz de clasificar la importancia de una determinada página en función de éstos. Pronto se percataron de que PageRank no sólo buscaba enlaces, los almacenaba y clasificaba, sino que también permitía realizar búsquedas entre todos esos enlaces.

De hecho, los resultados obtenidos eran mejores que aquellos dados por los buscadores más populares de la época, como por ejemplo Altavista. Además, PageRank era totalmente escalable: a mayor número de enlaces en su base de datos, mejor y más precisas eran las búsquedas que realizaba en ellos.

Viendo los buenos resultados, Page y Brin bautizaron de nuevo su creación con el definitivo nombre de Google. El extraño nombre proviene del término científico “googol”, que denota al número 10 elevado a 100. En 1997, publicaron su buscador en un servidor web de la Universidad de Stanford, convirtiéndose en muy poco tiempo en una de las herramientas web favoritas de los alumnos de esta universidad. Este éxito animó a sus creadores a continuar desarrollándolo y ampliándolo, comprando discos duros y CPU de segunda mano a sus propios compañeros e instalando sus servidores en la misma habitación de Sergey Brin. En pocos meses, Google comenzó a saturar la red de la Universidad de Stanford, una de las de mayor capacidad del mundo en aquella época, debido al tráfico de consultas que recibía. Es entonces cuando deciden adquirir el dominio “google.com”.

Las necesidades de hardware y espacio de Google no dejaban de crecer, hasta el punto de que Larry Page y Sergi Brian se ven incapaces de financiarlo, a pesar de las labores de intermediación de la propia universidad para vender el proyecto. Finalmente, en 1998 y tras una charla informal de 10 minutos con Andy Bechtolsheim (cofundador de Sun Microsystems y vicepresidente de Cysco Systems), éste les extiende un cheque por valor de 100.000 dólares para que paguen “sus discos duros y a los dichosos abogados, para los temas legales” (sic). Para poder hacer efectivo los cobros se ven obligados a fundar sobre la marcha Google Inc., ya que fue el nombre que escribió Bechtolsheim en el cheque.

Desde entonces, Google no ha parado de indexar páginas y se convirtió de forma gradual en el buscador consultado por la abrumadora mayoría de los usuarios de Internet. Larry Page y Sergi Brian no invirtieron ni un solo dólar en publicidad: la boca a boca y sus resultados fueron suficientes para darlo a conocer al mundo. Actualmente, tiene una media de 200 millones de consultas al día.

En 2008, Google Inc. recibió el Premio Príncipe de Asturias de Comunicación y Humanidades, por "favorecer el acceso generalizado al conocimiento" y su "contribución decisiva al progreso de los pueblos, por encima de fronteras ideológicas, económicas, lingüísticas o raciales".

2.2.4 Otros productos de Google Inc.

Sin lugar a dudas, el producto estrella de Google Inc. es su potente buscador web. Hoy en día el uso de Internet sería muy diferente sin la posibilidad de realizar búsquedas con la precisión y rapidez que Google puede ofrecer.

Sin embargo, tras el espectacular crecimiento de Google y su primer servicio comercial, AdWords (que permitía asociar publicidad a determinadas búsquedas), Google Inc. no ha dejado de innovar. En pocos años ha ido creando nuevos servicios que, en la mayoría de los casos, han adquirido tal popularidad entre la gente, que para muchos usuarios sería difícil imaginar el uso cotidiano de Internet si no existieran. En todo ellos, Google Inc. parece querer imprimir la misma filosofía de empresa: un servicio web, de acceso libre, gratuito en su mayor parte, con una interfaz minimalista pero funcional, y orientado siempre a mejorar la experiencia del usuario y a la interconexión con otros servicios en red.

Entre los productos más populares a través de la Web que actualmente ofrece Google, podemos encontrar los siguientes:

2.2.4.1 Gmail

Probablemente, el servicio más utilizado de Google Inc. junto con su buscador. Gmail es un completo servicio de correo electrónico lanzado en 2004, que pronto destacó por la gran capacidad de almacenamiento (actualmente más de 7 GB), tamaño de archivos adjuntos, facilidad para realizar búsquedas entre el historial, su formato de etiquetado y clasificación de mensajes, configuración de idiomas y la sencilla interfaz que ofrecía.

2.2.4.2 Google Maps

Google Maps es un servicio de localización a través de la Web, que ofrece una vista aérea de casi cualquier lugar del mundo. Mediante este servicio, el usuario puede cambiar entre el punto de vista de una fotografía real de satélite o el de mapa. Además, permite navegar libremente por el plano, alejando y acercando la imagen hasta un nivel de precisión considerable. Ofrece también un servicio de mejor ruta entre dos puntos y de localización exacta mediante la introducción de una dirección cualquiera del globo. Recientemente ha sido mejorado con Google Street View, que añade una nueva perspectiva panorámica de 360° desde pie de calle.

2.2.4.3 Google Calendar

Google Calendar es un servicio web que ofrece una completa agenda y planificación de horarios. Cada usuario dispone de calendarios mensuales o semanales, diferenciados por colores según la actividad, con la posibilidad de detallar cada día de la semana y de programar alertas dirigidas a un teléfono móvil o un correo electrónico. Los usuarios pueden compartir entre sí calendarios y actividades.

2.2.4.4 Google Documents

Representa uno de los proyectos más ambiciosos que actualmente desarrolla Google Inc. El objetivo es crear una completa plataforma de herramientas ofimáticas utilizando únicamente la Web, sin necesidad de instalar ningún software. Pensado para poder competir algún día con el extendido Microsoft Office, Google Documents contempla actualmente un procesador de texto, hojas de cálculo y presentaciones.

2.2.4.5 Google Chrome

Google Chrome es el navegador web desarrollado por Google Inc. Es un proyecto de software libre y está basado en componentes de otros

navegadores de código abierto, como WebKit y Mozilla. Su principal objetivo es ofrecer una navegación más rápida, estable y segura, sumado a una interfaz simple y eficiente. La versión beta de Google Chrome fue lanzada en septiembre de 2008 y aunque recibió una buena acogida, todavía presentaba algunas limitaciones que no le permiten competir directamente con Explorer de Microsoft o con Mozilla Firefox.

2.2.4.6 Youtube

Aunque no es un producto desarrollado por Google Inc., fue adquirido por éste en 2006. Youtube es un sitio web que permite a los usuarios subir y visualizar toda clase de vídeos. Además, los vídeos pueden recibir comentarios, votaciones, y los usuarios pueden crear sus propios canales de distribución. Su popularidad es tan grande que actualmente está entre las páginas más visitadas de todo Internet y se ha convertido en un auténtico fenómeno sociológico. Recientemente, ha debido suprimir gran parte de sus contenidos debido a la vulneración de los derechos de autor.

2.2.4.7 Picasa

Picasa es otra plataforma software no creada inicialmente por Google Inc., pero sí adquirida en 2004. Este software permite organizar, editar y buscar las fotografías de un usuario. También se encuentra en desarrollo una nueva funcionalidad que permita reconocer imágenes para poder asociarlas a determinadas palabras o claves de búsqueda. A diferencia de otros servicios, Picasa sí necesita ser instalado en el equipo del usuario, aunque tiene algunas funcionalidades web.

2.2.4.8 Blogger

Adquirido por Google Inc. en 2003, Blogger es actualmente uno de los portales de blogs más populares en todo el mundo. Su facilidad de uso

permite al usuario registrarse, crear un blog e introducir nuevas entradas en tan sólo unos pequeños pasos.

2.2.5 Críticas a Google Inc.

A lo largo de su corta pero fulgurante trayectoria, Google Inc. ha ido cosechando numerosos éxitos que la han convertido en la empresa de referencia dentro de las tecnologías de Internet. Para muchos, Google Inc. es esa empresa cuyos directivos visten vaqueros y camiseta, apenas rondan los 35 años y convierten en oro todo lo que tocan, generando grandes beneficios. Sin embargo, en los últimos años se han empezado a oír voces que critican algunos de sus comportamientos, y que cuestionan su popular eslogan "*Don't be evil*" ("No seas malvado").

Quizás el más sonado y que peor repercusión tuvo para la imagen de Google Inc. fue su incursión en China. Para que el gobierno chino permitiera operar a la compañía en su territorio, Google Inc. se vio obligado a aceptar y colaborar con las férreas normas de censura de aquel país, bloqueando el acceso a multitud de páginas y servicios que eran considerados maliciosos para el pueblo chino o contrarios a los intereses de su régimen político.

Por otro lado, el nacimiento de Gmail no estuvo tampoco exento de críticas. El hecho de que los robots de la compañía recorriesen los mensajes de los usuarios en busca de palabras a las que asociar anuncios de publicidad, o las condiciones de uso que advertían del mantenimiento de mensajes aún después de haber sido borrados, puso en alerta a muchas asociaciones de consumidores que acusaban a Google Inc. de vulnerar la intimidad de las personas.

Además, Google Inc. ha recibido acusaciones de atentar contra la propiedad intelectual. En 2008 fue finalmente absuelto por reproducir en los resultados de sus búsquedas parte de los contenidos de algunas páginas web, mientras que ese mismo año fue requerido judicialmente a

proporcionar los datos de los usuarios de Youtube al gigante mediático Viacom, para que así éste pudiera presentar una demanda por utilizar parte de sus producciones sin autorización.

2.2.6 MySQL

Es un sistema de gestión (o manejador) de bases de datos. Una Base de Datos es una colección estructurada de datos. Puede ser, desde una simple lista de artículos a las inmensas cantidades de información en una red corporativa.

MySQL es un software de código abierto. Esto quiere decir que es accesible para cualquiera, para usarlo o modificarlo. Se puede descargar MySQL desde Internet y usarlo sin pagar nada, de esta manera cualquiera puede inclinarse a estudiar el código fuente y cambiarlo para adecuarlo a sus necesidades.

MySQL usa el GPL (GNU General Public License) para definir que se puede o no hacer con el software en diferentes situaciones.

MySQL es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños, aunque esta rapidez es a costa de no implementar ciertos aspectos del SQL.

Principales características:

- Velocidad y robustez.
- Multiproceso, es decir puede usar varias CPU, si éstas están disponibles.
- Puede trabajar en distintas plataformas y sistemas operativos distintos.
- Sistema de contraseñas y privilegios muy flexible y seguro.
- Todas las palabras viajan encriptadas en la red.
- Los clientes usan TCP o UNIX Socket para conectarse al servidor.
- El servidor soporta mensajes de error en distintos idiomas.

- Todos los comandos tienen -help o - para las ayudas.

2.2.7 Dispositivo móvil.

Un dispositivo móvil es un aparato de pequeño tamaño, con algunas capacidades de procesamiento, alimentación autónoma, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales.

Dado el variado número de niveles de funcionalidad asociado con dispositivos móviles, en el 2005, T38 y DuPont Global Mobility Innovation Team propusieron los siguientes estándares para la definición de dispositivos móviles:

2.2.7.1 Dispositivo Móvil de Datos Limitados (Limited Data Mobile Device):

Dispositivos que tienen una pantalla pequeña, principalmente basada en pantalla de tipo texto con servicios de datos generalmente limitados a SMS y acceso WAP. Un típico ejemplo de este tipo de dispositivos son los teléfonos móviles

2.2.7.2 Dispositivo Móvil de Datos Básicos (Basic Data Mobile Device):

Dispositivos que tienen una pantalla de mediano tamaño, (entre 120 x 120 y 240 x 240 pixels), menu o navegación basada en íconos por medio de una "rueda" o cursor, y que ofrecen acceso a e-mail, lista de direcciones, SMS, y un navegador web básico. Un típico ejemplo de este tipo de dispositivos son las BlackBerrys y los Teléfonos Inteligentes

2.2.7.3 Dispositivo Móvil de Datos Mejorados (Enhanced Data Mobile Device):

Dispositivos que tienen pantallas de medianas a grandes (por encima de los 240 x 120 pixels), navegación de tipo stylus, y que ofrecen las mismas características que el "Dispositivo Móvil de Datos Básicos" (Basic Data Mobile Devices) más aplicaciones nativas y aplicaciones

corporativas usuales, en versión móvil. Este tipo de dispositivos incluyen el sistema operativo como Windows Mobile, Android, iPhone OS.

2.2.8 PDA 's

Una PDA (del inglés Personal Digital Assistant (asistente digital personal)), es una computadora de mano originalmente diseñada como agenda electrónica con un sistema de reconocimiento de escritura. Hoy día (2011) estos dispositivos pueden realizar muchas de las funciones de una computadora de escritorio, pero con la ventaja de ser portátil.

La primera mención formal del término y concepto de PDA es de 1992 cuando la compañía Appel presento el Apple Newton, un aparato que pese a contar con las características de una PDA moderna (pantalla sensible al tacto, conexión a una computadora para sincronización, interfaz de usuario especialmente diseñada para el tipo de máquina, conectividad a redes vía módem y reconocimiento de escritura) fue un fracaso para la compañía debido a los diferentes problemas: la tecnología estaba poco desarrollada y el reconocimiento de escritura en la versión original era bastante impreciso, entre otros. Aun así, el Apple Newton sentó las bases de las PDA's modernas.

Inicialmente los PDAs incluían aplicaciones estrictamente relacionadas con su función como agenda electrónica, es decir, se reducían a calendario, lista de contactos, bloc de notas y recordatorios. Con el paso de tiempo han ido evolucionando hasta los dispositivos actuales que ofertan un rango mucho más extendido de aplicaciones, como juegos, acceso al correo electrónico o la posibilidad de ver películas, crear documentos, navegar por Internet o reproducir archivos de audio.

Cabe destacar la aparición de la empresa Palm, Inc. en 1995 que propició una etapa de crecimiento y desarrollo de las PDA's

Las características de la PDA moderna son pantalla sensible al tacto, conexión a una computadora para sincronización, ranura para tarjeta de memoria, y al menos Infrarrojo, Bluetooth o WiFi.

Las PDAs de hoy en día traen multitud de comunicaciones inalámbricas (Bluetooth, WiFi, IrDA, GPS,) que los hace tremendamente atractivos.

La llegada de los teléfonos inteligentes o Comunicadores (híbridos entre PDA y teléfono móvil) supuso para el mercado, por un lado, la entrada de nuevos competidores y, por otro, la incorporación a éste de usuarios avanzados de móviles. Hoy en día la mayoría de los PDAs son smartphones.



FIGURA 7 Ejemplos de dispositivos móviles.

2.2.9 Smartphone

El teléfono inteligente (Smartphone en inglés) es un término comercial para denominar a un teléfono móvil que ofrece más funciones que un teléfono común.

Casi todos los teléfonos inteligentes son móviles que soportan completamente un cliente de correo electrónico con la funcionalidad completa de un organizador personal. Una característica importante de

casi todos los teléfonos inteligentes es que permiten la instalación de programas para incrementar el procesamiento de datos y la conectividad. Estas aplicaciones pueden ser desarrolladas por el fabricante del dispositivo, por el operador o por un tercero. El término "Inteligente" hace referencia a cualquier interfaz, como un teclado QWERTY en miniatura, una pantalla táctil (lo más habitual, denominándose en este caso "teléfono móvil táctil"), o simplemente el sistema operativo móvil que posee, diferenciando su uso mediante una exclusiva disposición del menú, teclas, atajos, etc.

Entre otras características comunes está la función multitarea, el acceso a Internet, los programas de agenda, a una cámara digital integrada, administración de contactos, y algunos programas de navegación, así como poder trabajar con documentos en diferentes formatos.

Todos los Smartphone deben contar con un S.O. que los provea de características similares a las de un computador. Aunque esta característica les hace ser vulnerables a virus y ataques al SO, tal como sucede en la actualidad con los equipos portátiles o de escritorio.

2.2.10 El Sistema de Posicionamiento Global (GPS)

El GPS (Global Positioning System: sistema de posicionamiento global) o NAVSTAR-GPS es un sistema global de navegación por satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave, con una precisión hasta de centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. El sistema fue desarrollado, instalado y actualmente operado por el Departamento de Defensa de los Estados Unidos de América.

Este sistema está constituido por tres segmentos:

- 1. El segmento espacial:** que consiste en una constelación nominal formada por 24 satélites en órbita sobre el globo, a 20.200 km, que

transmiten señales unidireccionales que proporcionan la posición y la hora de cada satélite del GPS.

2. **El segmento de control:** que está formado por estaciones de seguimiento y control distribuidas por todo el mundo a fin de mantener los satélites en la órbita apropiada mediante maniobras de mando y ajustar los relojes satelitales. Esas estaciones también realizan el seguimiento de los satélites del GPS, cargan información de navegación actualizada y garantizan el funcionamiento adecuado de la constelación de satélites.
3. **El segmento del usuario:** consiste en el equipo receptor del GPS que recibe las señales de los satélites del GPS y las procesa para calcular la posición tridimensional y la hora precisa.

Aplicaciones

Civiles:

- Navegador GPS de pantalla táctil de un vehículo con información sobre la ruta, así como las distancias y tiempos de llegada al punto de destino.
- Navegación terrestre (y peatonal), marítima y aérea. Bastantes automóviles lo incorporan en la actualidad, siendo de especial utilidad para encontrar direcciones o indicar la situación a la grúa.
- Teléfonos móviles
- Localización agrícola (agricultura de precisión), ganadera y de fauna. Salvamento y rescate.
- Deporte, acampada y ocio.
- Para localización de enfermos, discapacitados y menores.
- Aplicaciones científicas en trabajos de campo (ver geomática).
- Geocaching, actividad deportiva consistente en buscar "tesoros" escondidos por otros usuarios.
- Para rastreo y recuperación de vehículos, Navegación deportiva.

- Deportes aéreos: parapente, ala delta, planeadores, etc.
- Sistemas de gestión y seguridad de flotas.

Militares:

- Navegación terrestre, aérea y marítima.
- Guiado de misiles y proyectiles de diverso tipo: Búsqueda y rescate.
- Reconocimiento y cartografía.
- Detección de detonaciones nucleares.

2.2.11 Integración con telefonía móvil

Algunos móviles pueden vincularse a un receptor GPS diseñado a tal efecto. Suelen ser módulos independientes del teléfono que se comunican vía inalámbrica bluetooth, o implementados en el mismo terminal móvil, y que le proporcionan los datos de posicionamiento que son interpretados por un programa de navegación. Esta aplicación del GPS está particularmente extendida en los teléfonos móviles que operan con el sistema operativo Symbian OS, Android y PDAs con el sistema operativo Windows Mobile.

2.2.12 Entorno de Desarrollo con Android Studio 2.1

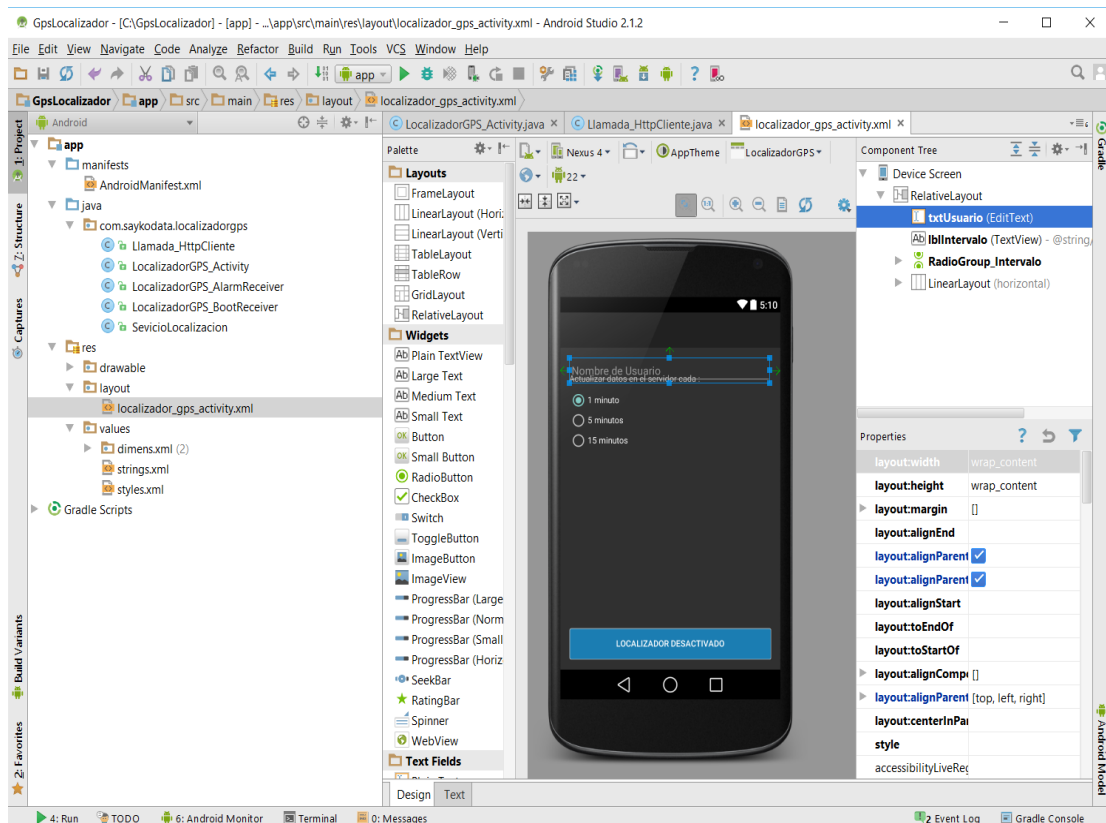


FIGURA 8 Entorno de Desarrollo Android Studio

Hay 4 diferentes tipos de componentes:

Activity: Representa una pantalla independiente con una interfaz de usuario. A pesar de que nuestra aplicación dispondrá de múltiples pantallas interconectadas entre sí, nosotros deberemos generarlas individual e independientemente (pudiendo pasar datos entre ellas, en caso de ser necesario).

Service: Es un componente que corre de fondo para hacer operaciones de larga duración o trabajo en procesos remotos. Contrario a la actividad, no dispone de interfaz gráfica.

Content Provider: Este componente nos permite gestionar un conjunto de datos de la aplicación para compartir. Los Contactos son el ejemplo

perfecto para este componente: datos que podemos compartir entre diferentes aplicaciones. Pero podemos crear nuestro propio conjunto de datos.

Broadcast Receiver: El cuarto de los componentes nos permite responder a anuncios broadcast del sistema. Un buen ejemplo es si queremos gestionar cuando tengamos el aviso de batería baja (el cual enviará un mensaje broadcast), aunque podemos diseñar nuestros propios mensajes.

Un aspecto interesante de diseño de Android es que una aplicación A podría abrir un componente de una aplicación B. El ejemplo ideal es cuando queremos usar la cámara en nuestra app, podemos hacer una Activity con la cámara, o abrir el componente de la cámara que viene ya instalada por defecto en el sistema operativo.

Para ello utilizamos un mensaje llamado Intent, el cual también sirve para activar 3 de los 4 componentes de una app (todos excepto el Content Provider).



FIGURA 9 Componentes del Android Studio

¿Cómo sabe nuestra aplicación qué componentes tiene disponibles?

Para ello, existe el archivo **AndroidManifest.xml**. Este archivo será el encargado de comunicarle al sistema operativo:

- los componentes que dispone la aplicación (actividades, servicios, etc)
- los permisos necesarios que podrá utilizar la aplicación (cámara, GPS, etc)
- la versión de Android mínima necesaria
- el hardware y software requerido y/o usado
- las librerías externas que utiliza (como Google Maps...)

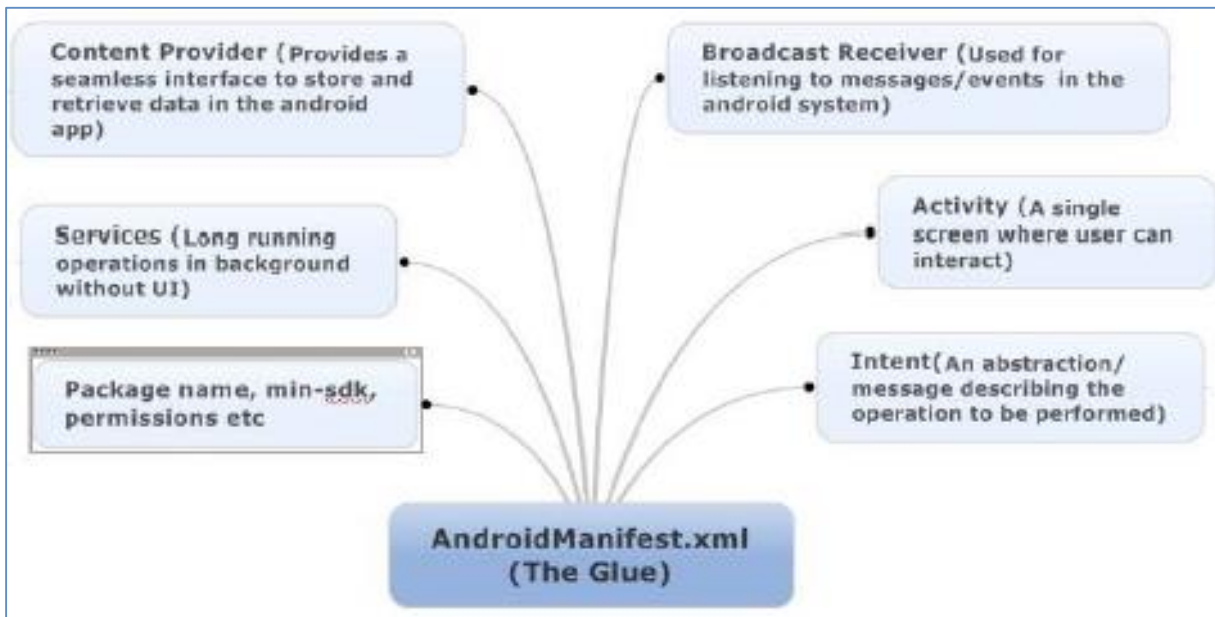


FIGURA 10 Detalle de los componentes del Android Studio

2.3 Definición de Terminos

a. Acelerómetro: instrumento destinado a medir aceleraciones. Frecuentemente se utiliza para conocer la velocidad y desplazamiento de un cuerpo en uno o varios ejes dimensionales.

b. Add-on: ver plug-in.

c. AIDL: siglas de Android Interface Definition Language, en español Lenguaje para la Definición de Interfaces en Android. Constituye un lenguaje

de sintaxis muy básica que permite describir interfaces que pueden ser utilizadas de forma remota.

- d. **API:** siglas de Application Programming Interface, en español Interfaz de Programación de Aplicaciones. Consiste en un conjunto de llamadas que ofrecen acceso a funciones y procedimientos, representando una capa de abstracción para el desarrollador.
- e. **Background:** representa un proceso que se ejecuta con pocos recursos, que no requiere interacción directa con el usuario y que existe sin el conocimiento de este.
- f. **Biblioteca:** agrupación de código y datos que proporcionan servicios a programas independientes, pasando a formar parte de éstos. Permiten la distribución de funcionalidades y la construcción modular. También conocido como librería, por su similitud con el inglés library.
- g. **Bluetooth:** protocolo que permite la transmisión de datos entre dispositivos, más o menos próximos y alineados, mediante un enlace de radiofrecuencia. Está especialmente diseñado para dispositivos de bajo consumo y coste.
- h. **Bytecode:** código intermedio, más abstracto que el código máquina, y que necesita de un mediador o máquina virtual para poder ser transformado y ejecutado en un hardware local.
- i. **Callback:** se denomina así a la relación que existe entre dos procesos cuando el origen de la comunicación es a su vez llamado o invocado por el proceso destino.
- j. **Checkbox:** elemento de interfaz de usuario que permite hacer una selección múltiple en un conjunto de opciones.
- k. **Códec:** abreviatura de Compresor-Decompresor, en español Compresor- Descompresor. Representa un algoritmo software o componente hardware que permite transformar un flujo de datos en una señal interpretable.
- l. **Controlador:** programa que permite al sistema operativo interactuar con un

periférico, abstrayéndolo y propocionando una interfaz para usarlo. También conocido como driver.

- m. CORBA:** siglas de Common Object Request Broker Architecture, en español Arquitectura Común de Intermediarios en Peticiones a Objetos. Define un estándar mediante el cual elementos distribuidos programados en distintos lenguajes pueden intercambiar datos y objetos.
- n. CPU:** siglas de Central Processing Unit, en español Unidad Central de Procesamiento. Elemento de hardware que controla el funcionamiento de un computador y lleva a cabo sus funciones de procesamiento de instrucciones y datos.
- o. CVM:** siglas de Compact Virtual Machine. Representa, junto a KVM, una de las máquinas virtuales de Java disponibles en Java ME.
- p. Dalvik:** nombre de la máquina virtual utilizada por el sistema operativo Android. Dalvik está específicamente adaptada a las características de rendimiento de un dispositivo móvil y trabaja con ficheros de extensión “.dex”, obtenidos desde el bytecode de Java.
- q. Datagrama:** fragmento de información enviado por una red bajo un determinado protocolo de comunicación, generalmente formado por una cabecera y datos.
- r. Dispositivo móvil:** aparato electrónico que es de reducido tamaño, cuenta con cierta capacidad tanto para la computación como para el almacenamiento de datos y cuenta con elementos de E/S básicos, por ejemplo, pantalla y/o teclado.
- s. Driver:** ver controlador.
- t. DTD:** siglas de Document Type Definition, en español Definición de Tipo de Documento. Permite definir el formato que ha de tener un determinado documento XML.
- u. E/S:** abreviatura de Entrada/Salida. Un elemento de E/S es aquél que permite la comunicación entre un sistema de procesamiento de datos y una entidad externa a él (un usuario humano u otro sistema de procesamiento). Entrada se considera toda aquella información que es

recibida por un sistema, mientras que salida es aquella información que es enviada por el mismo.

v. Ebook: dispositivo electrónico que permite la visualización de documentos, como por ejemplo libros, en formato digital a través de una pantalla.

w. EDGE: siglas en inglés de Enhanced Data rates for GSM of Evolution, en español Tasas de Datos Mejoradas para la evolución de GSM. Es una tecnología para telefonía móvil que representa un puente entre la segunda y tercera generación de estos dispositivos.

x. Fragmentación: fenómeno que describe una situación en la que una misma tecnología ha evolucionado de forma que se hace incompatible entre sí.

y. Framework: término con el que se define un amplio conjunto de elementos que permite desarrollar y organizar software utilizando un determinado lenguaje, sistema o tecnología. Habitualmente incluye bibliotecas, programas de desarrollo o manuales.

z. Googol: término con el que se designa al número 10 elevado a 100.

aa. GPS: siglas de Global Positioning System, en español Sistema de Posicionamiento Global. Es un sistema global de navegación que, mediante satélites, permite ubicar un objeto en la superficie terrestre con una precisión que va desde varios metros a centímetros.

bb. GSM: siglas de Groupe Spécial Mobile, más conocido como Sistema Global para las Comunicaciones Móviles, es el estándar más extendido para las comunicaciones con telefonía móvil. Permite llamadas, navegación por Internet o envío de SMS.

cc. GPRS: siglas de General Packet Radio Service, en español Servicio General de Paquetes vía Radio. es una extensión del estándar GSM que permite mejorar sus prestaciones originales, como el envío de datos, uso de correo electrónico, navegación web o el aumento de las tasas de

transferencia de datos.

dd. GUI: siglas de Graphical User Interface, en español Interfaz Gráfica de Usuario. Representa la parte del software que, mediante un contexto o lenguaje principalmente visual y simbólico, permite al usuario utilizar una aplicación.

ee. Hilo: en sistemas operativos, un hilo constituye cada uno de los flujos de ejecución en el que puede ser dividido un proceso. Todos los hilos de un proceso comparten espacio en memoria, archivos abiertos, variables globales, semáforos, etc. Permiten la ejecución concurrente de varias tareas. También llamado thread.

ff. HTTP: siglas de HyperText Transfer Protocol, en español Protocolo de Transferencia de Hipertexto. Constituye el protocolo utilizado para la transmisión de documentos a través de la Web entre un cliente y servidor.

gg. Interfaz: en computación, una interfaz se refiere a una abstracción que un determinado elemento ofrece de sí mismo al exterior, facilitando de esta forma su acceso y uso por otros elementos de hardware o software.

hh. JAD: siglas de Java Application Descriptor, en español Descriptor de Aplicación Java. Archivo que acompaña a una aplicación Java ME y que ofrece información general y de despliegue sobre la misma.

ii. JAR: acrónimo de Java ARchive, en español Archivo Java. Representa una agrupación de varios ficheros Java y se usa generalmente para la distribución conjunta de clases y metadatos.

jj. Java ME: Java Micro Edition, edición de Java especialmente dirigida a los dispositivos móviles.

kk. Java Micro Edition: ver Java ME.

ll. JSR: siglas de Java Specification Request, en español Solicitud de Especificación Java. Representa un documento formal que describe las especificaciones y tecnologías propuestas para ser añadidas oficialmente a la plataforma Java.

mm. JVM: Siglas de Java Virtual Machine, en español Máquina Virtual de Java.

Constituye un elemento software de la tecnología Java, encargado de transformar el código intermedio universal o bytecode en código máquina específico del hardware donde está instalado.

nn. Kernel: parte fundamental de un sistema operativo, responsable de facilitar acceso seguro al hardware, gestionar recursos y hacer llamadas al sistema. También conocido como núcleo.

oo. KVM: siglas de Kilobyte Virtual Machine. Representa, junto a CVM, una de las máquinas virtuales de Java disponibles en Java ME.

pp. Laptop: ordenador portátil. También llamado notebook.

qq. Latitud: distancia angular entre el ecuador y un punto de la superficie del planeta. Se mide en grados entre 0 y 90.

rr. Librería: ver biblioteca.

ss. Listener: objeto que está a la espera de determinado evento.

tt. Longitud: distancia angular entre el meridiano y un punto de la superficie del planeta. Se mide en grados entre 0 y 360.

uu. Máquina virtual: representa un software que emula el comportamiento de una determinada arquitectura o que permite adaptar un código fuente a las características de la máquina nativa.

vv. Middleware: capa de abstracción software que posibilita el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.

ww.MMS: siglas de Multimedia Messaging System o en español Sistema de Mensajería Multimedia. Es un estándar de mensajería que le permite a los teléfonos móviles enviar y recibir contenidos multimedia, incorporando sonido, video, f o t o s o cualquier otro contenido disponible en el futuro.

xx. MOAP: siglas de Mobile Oriented Applications Platform, una plataforma software basada en Symbian para los teléfonos del fabricante japonés FOMA.

yy. MP3: formato de compresión de audio digital cuyo nombre completo es MPEG-1 Audio Layer 3. También utilizado, por extensión, para nombrar al dispositivo móvil reproductor de audio digital en el que se almacena,

organiza o reproduce archivos de audio digital.

zz. Núcleo: ver kernel.

aaa. OHA: siglas de Open Handset Alliance, un conglomerado de empresas de sectores tecnológicos lideradas por Google que promueve la innovación y desarrollo de dispositivos móviles y sus aplicaciones. Su primera contribución es el sistema operativo Android.

bbb. PDA: siglas de Personal Digital Assistant, en español Asistente Digital Personal. Dispositivo móvil utilizado como organizador personal, que cuenta generalmente con pantalla táctil, agenda, calendario, conectividad Wi-Fi, y aplicaciones ofimáticas, entre otros

ccc. Plug-in: componente de software que se relaciona y ejecuta con otro para aportarle una función nueva y generalmente muy específica.

ddd. Proceso: un proceso es un programa en ejecución, y representa la unidad de procesamiento básica gestionada por el sistema operativo.

eee. Radiobutton: elemento de interfaz de usuario que permite seleccionar un único elemento dentro un conjunto definido de valores.

fff. AM: siglas de Random Access Memory, o en español Memoria de Acceso Aleatorio. Componente de memoria volátil, con palabras de acceso individual mediante una dirección de memoria específica y con rápida ejecución de operaciones de lectura y escritura.

ggg. SAX: siglas de Simple API for XML, en español API Simple para XML, representa una conocida API para Java que facilita el procesamiento de documentos XML.

hhh. SDK: siglas de Software Development Kit, en español Kit de Desarrollo de Software. Constituye un conjunto de herramientas que permiten a un desarrollador crear aplicaciones para una determinada plataforma o lenguaje.

iii. Servlet: elemento de la tecnología Java, que extiende la funcionalidad de un servidor Web, aceptando y procesando peticiones.

jjj. Sistema operativo: programa cuya finalidad principal es simplificar el manejo y explotación de un elemento con capacidad computacional,

gestionando sus recursos, ofreciendo servicios a las demás aplicaciones y ejecutando mandatos del usuario.

kkk. Smartphone: dispositivo móvil que representa una evolución de los teléfonos móviles, con la inclusión de pantalla táctil, teclado, conexión Wi-Fi, aplicaciones de usuario como navegador web o cliente de correo, entre otros.

III. SMS: siglas de Short Message Service, en español Servicio de Mensajes Cortos, es un estándar de la telefonía móvil que permite enviar mensaje de texto con un número de caracteres limitado.

mmm. Socket: abstracción software, identificada por una dirección IP, un protocolo y un puerto, que permite la comunicación de dos programas, generalmente situados en computadores distintos.

nnn. Teclado QWERTY: teclado cuya distribución de letras es la más común hoy día en ordenadores y otros elementos de computación. Toma su nombre de sus 5 primeras letras alfabéticas: Q, W, E, R, T, e Y.

ooo. Thread: ver hilo.

ppp. UIQ: siglas de User Interface Quartz, una plataforma software basada en Symbian usada en algunos teléfonos de los fabricantes Sony Ericsson y Motorola.

qqq. UMTS: siglas de Universal Mobile Telecommunications System, en español Sistema Universal de Telecomunicaciones Móviles. Constituye en estándar de comunicación para dispositivos de tercera generación o 3G, que ofrece capacidades multimedia y conexiones de alta velocidad en Internet.

rrr. AP: siglas de Wireless Application Protocol, en español o Protocolo de Aplicaciones inalámbricas. Es un estándar para aplicaciones que utilizan las comunicaciones inalámbricas, como el acceso a servicios de Internet desde un teléfono móvil.

sss. Widget: componente gráfico utilizado en interfaces de usuario, con el cual el usuario puede interactuar, como por ejemplo cajas de texto, botones, ventanas, etc.

ttt. Wi-Fi: acrónimo de Wíreless Fidelity, estándar de envío de datos que utiliza ondas de radio en lugar de cables.

Uuu. XML: siglas de Extensible Markup Language, en español Lenguaje de Marcado Extensible. Representa un lenguaje estándar que, mediante el uso de etiquetas y atributos, permite expresar e intercambiar fácilmente estructuras de datos.

vvv. XMPP: siglas de Extensible Messaging and Presence Protocol, en español Protocolo Extensible de Mensajería y Presencia. Es un protocolo basado en XML que se utiliza en servicios de mensajería instantánea.

CAPITULO

III. MARCO METODOLÓGICO

3.1 Variables

3.1.1 Variable Independiente:

Aplicación web

3.1.2 Variable Dependiente:

Seguimiento de personas mayores de edad

3.2 Operacionalización de la variable

Título: "IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA EL SEGUIMIENTO DE PERSONAS MAYORES DE EDAD UTILIZANDO GEOLOCALIZACIÓN DESDE UN DISPOSITIVO ANDROID CONECTADO A UN SERVIDOR WEB CON MYSQL"

VARIABLE	DIMENSIONES	INDICADORES
Variable Independiente: Aplicación Web	Cognitivo	• ¿Sabe el significado de aplicaciones web y móviles? • ¿Han brindado charlas en su localidad sobre nuevas tecnologías? • ¿Tiene algún conocimiento de dispositivo Android?
	Tecnología	• ¿Tiene internet en su casa? • ¿Cuenta con dispositivos móviles en su hogar? • ¿Todos los integrantes de su hogar cuentan con dispositivos móviles? • ¿Ha utilizado nuevas aplicaciones en sus equipos móviles? • ¿Se ha propuesto alguna vez adquirir nuevas aplicaciones para su celular?
	Flexibilidad	• ¿Cree usted que el uso de nuevas aplicaciones móviles ha mejorado su vida cotidiana? • ¿El funcionamiento de estos equipos móviles es bueno?

Tabla 3 Operacionalización de la Variable Dependiente

VARIABLE	DIMENSIONES	INDICADORES
Variable Dependiente: Seguimiento de personas mayores de edad	Cognitivo	• ¿Sabe el significado de geolocalización? • ¿Ha recibido charlas sobre temas de geolocalización?
	Actividad Humana	• ¿Con qué frecuencia su familiar realiza salidas fuera de casa? • ¿Ha tenido la oportunidad de acudir a la policía para realizar la búsqueda de un familiar? • ¿Le sirvió de ayuda la policía de su localidad?
	Salud	• ¿Existen en su hogar adultos mayores que presentan problemas de salud mental? • ¿Alguna vez se ha llegado a perder algún familiar mayor? • ¿Has realizado algún mecanismo para saber el paradero de ese familiar? • ¿Le ayudó bastante el mecanismo realizado?
	Seguridad	• ¿Existe protección para el adulto mayor? • ¿Cuenta con medidas de protección para el adulto mayor?

Tabla 4 Operacionalización de la Variable Dependiente

3.3 Metodología

3.3.1 Desarrollo de la metodología Mobile-D para aplicaciones móviles.

Hoy en día, el sector de los dispositivos móviles ha evolucionado de manera muy creciente en pocos años. Esto ha permitido crear un nuevo mercado, bastante amplio para todos los públicos y un abanico enorme de posibles soluciones eficaces que nos harían la vida más cómoda. Muy pocas personas son los que, a día de hoy, no tienen un dispositivo móvil, ya sea un smartphone o una tablet.

El desarrollo de aplicaciones móviles no se aleja mucho con respecto a los desarrollos de cualquier tipo de software normal, ya que nos encontramos con los mismos problemas a la hora de realizarlo. Unos factores en los que sí varía mucho es en el hardware donde serán implantados dichos desarrollos, ya que están en constante evolución y el usuario suele cambiar de dispositivo en poco tiempo, a diferencia con los ordenadores personales que se aguantan mucho más, a pesar de que el coste es menor.

A la hora de elegir una metodología a seguir para realizar un desarrollo de una aplicación móvil, nos podemos encontrar varias, como por ejemplo SCRUM, Extreme Programming, etc. Pero me ha llamado la atención la metodología ágil **Mobile-D**, que se ajusta mucho a este tipo de desarrollos.

Esta metodología está basada en diversas tecnologías como Rational Unified Process, Extreme Programming y Crystal Methodologies, y su finalidad es intentar obtener pequeños ciclos de desarrollo de forma rápida en dispositivos pequeños.

Para el desarrollo de este proyecto se utilizará la metodología **Mobile-D**, compuesta por cinco fases, las cuales se describen a continuación:

1. Fase de Exploración

Esta fase es la encargada de la planificación y deducción de requisitos del proyecto, donde se tendrá la visión completa del alcance del proyecto y también todas las funcionalidades del producto.

2. Fase de inicialización

La fase de inicialización es la implicada en conseguir el éxito en las próximas fases del proyecto, donde se preparará y verificará todo el desarrollo y todos los recursos que se necesitarían. Esta fase se divide en cuatro etapas: la puesta en marcha del proyecto, la planificación inicial, el día de prueba y día de salida.

3. Fase de producción

En la fase de producción, se vuelve a repetir la programación de los tres días, iterativamente hasta montar (implementar) las funcionalidades que se desean. Aquí usamos el desarrollo dirigido por pruebas (*TDD*), para verificar el correcto funcionamiento de los desarrollos.

4. Fase de estabilización

Se llevarán a cabo las últimas acciones de integración donde se verificará el completo funcionamiento del sistema en conjunto. De toda la metodología, esta es la fase más importante de todas ya que es la que nos asegura la estabilización del desarrollo. También se puede incluir en esta fase, toda la producción de documentación.

5. Fase de pruebas

Es la fase encargada del testeo de la aplicación una vez terminada. Se deben realizar todas las pruebas necesarias para tener una versión estable y final. En esta fase, si nos encontramos con algún tipo de error, se debe proceder a su arreglo, pero nunca se han de realizar desarrollos nuevos de última hora, ya que nos haría romper todo el ciclo.

3.4 Tipos de Estudio

La investigación realizada es de tipo descriptiva y aplicada.

Descriptiva, porque se analizarán los datos recogidos a través de los instrumentos de recopilación que luego serán procesados mediante gráficos estadísticos, los cuales ayudarán a determinar el nivel de protección que existe hacia la persona mayor cuando se encuentre fuera de casa y que posteriormente serán explicados en función de las variables (dependiente e independiente).

Y Aplicada, porque se aprovechará de la tecnología Android para utilizar geolocalización, y de esta manera saber la ubicación de la persona.

3.5 Diseño

Transversal de tipo descriptivo y aplicado, pues se investigaron hechos o fenómenos en un determinado tiempo; además la información de la cual se ha tomado en cuenta fue de las variables correspondientes.

Por lo que en esta investigación se propone en brindar la ubicación exacta de la persona mayor de edad, mediante el seguimiento realizado desde el dispositivo móvil.

3.6 Población muestra y muestreo

3.6.1 Población

Se tomará como población a los habitantes del Pueblo Joven Ricardo Palma del distrito de Chiclayo y que sean propietarios de un dispositivo móvil, puesto que son el mayor número de personas con las posibilidades de adquirir la aplicación y hacer uso de ella, para ello se tomó los datos estadísticos de un estudio realizado en el año 2007 por el Instituto Nacional de Estadística e Informática (INEI), en el área de Tecnologías de la Información y comunicaciones (TIC's), referente a las personas que utilizan teléfonos inteligentes y en donde se pudo determinar la población de 1500 usuarios.

3.6.2 Muestra

Por estar relacionado al tema de personas mayores de edad, únicamente se consideró a aquellos adultos mayores que por motivo de salud son los que más ayuda necesitan, cuando éstos salen fuera de casa, pues se da el caso de que estos no recuerdan donde se encuentran o se lleguen a sentir mal.

El tamaño de la muestra se obtiene aplicando la siguiente fórmula:

$$n = \frac{Z^2 * P * Q * N}{(N - 1) * e^2 + (Z^2 * P * Q)}$$

Donde:

N = Población

e = 0.05 (máximo de error permisible)

Z = 1.96 (valor tabla) 95%

P = 0.5 (proporción de la población)

Q = 0.5 (1-P)

Para nuestro estudio tenemos:

Donde:

N = 1500

e = 0.05

Z = 1.96

P = 0.5

Q = 0.5

$$\begin{aligned} n &= \frac{1.96^2 * 0.5 * 0.5 * 1500}{(1500 - 1) * 0.05^2 + (1.96^2 * 0.5 * 0.5)} \\ n &= \frac{1440.6}{4.7079} = 305.99 \end{aligned}$$

n = 306

La muestra a considerar para el estudio es de 306 usuarios, que son adultos mayores de edad que disponen de un dispositivo móvil.

3.6.3 Muestreo

Para el muestreo se toman los usuarios propietarios de dispositivos móviles que forman parte de la muestra y que son los indicados para el cumplimiento del objetivo principal.

3.7 Técnicas e instrumentos de recolección de datos

3.7.1 Técnicas

La encuesta

Es una técnica que permite obtener información a través de una serie de preguntas estratégicas que determinan el grado de importancia del proyecto de investigación. Estas preguntas darán los datos estadísticos necesarios, para analizar cuál es la viabilidad del proyecto.

Con esta técnica se obtuvo información relevante y necesaria referente al tema de la investigación, la cual ayudara a sustentarlo y justificarlo.

La Entrevista

Técnica que realiza un procedimiento similar a la encuesta con la diferencia que las preguntas se desarrollan de forma oral obteniendo las respuestas de igual forma.

Se utiliza esta técnica, porque permite conocer información de determinados hechos a través de opiniones de diferentes personas que poseen dispositivos móviles; además se pretendió recopilar datos válidos y confiables referentes a los problemas que presentan los adultos mayores.

Técnica de la investigación Bibliográfica

Con esta técnica se sustentará la base teórica de la investigación, mediante consultas a: fuentes bibliográficas confiables, libros, artículos científicos, base de datos científicas entre otras

3.7.2 Instrumentos

Se selecciona el cuestionario y la entrevista como instrumentos de recolección de datos

Cuestionario

Nos permitió observar las carencias de seguridad en los accesos a la información de las distintas áreas, así como la interferencia en los tipos de datos voz y video

La entrevista

Fue aplicada directamente a las personas mayores de edad propietarias de un dispositivo móvil.

3.8 Método de análisis de datos

Los resultados cuantitativos permitieron el análisis cualitativo de los resultados referidos a la situación actual en que viven los adultos mayores, con respecto al problema de su ubicación que tienen cuando se encuentran fuera de casa, por lo que nos permitió discutir los resultados obtenidos.

3.9 Aspectos éticos

Como investigador me comprometo a respetar la veracidad de los resultados, la confiabilidad de los datos brindados por la población del Pueblo Joven Ricardo Palma y la protección de la identidad de los individuos que participan en el estudio.

CAPITULO
IV. RESULTADOS

RESULTADOS

Para comprobar que la aplicación se ejecuta correctamente en varios dispositivos, se realizó una prueba con la ayuda de la herramienta online apkudo, la cual permite obtener una lista de dispositivos móviles en los cuales una aplicación funciona sin ningún contratiempo gracias a la ayuda de Mono Running, una herramienta utilizada para determinar la compatibilidad de la aplicación con los diferentes tipos de versiones de Android y modelos de celulares. En la tabla 5 se presentan los dispositivos en los que la aplicación funcionó adecuadamente. Apartir de la versión 4.4 en adelante la aplicación funciona correctamente.

MARCA	NOMBRE	MODELO	VERSION DE S.O ANDROID	RESULTADOS
HTC	G2	PC10100	2.3.4	Fallido
	Nexus One	PB99100	2.3.6	Fallido
	One V	T-320e	4.0.3	Fallido
	One X LTE	PJ83100	4.0.3	Fallido
	One S	HTC One S	4.0.4	Fallido
	One X	S720E	4.0.4	Fallido
	Sense Home		4.4	Correcto
	Desire	526	5.1	Correcto
LG	Optimus L3	E400	2.3.6	Fallido
	Viper 4G LTE	LS-840	2.3.7	Fallido
	Optimus Elite	LG-LS696	2.3.7	Fallido
	L5	E610	4.0.3	Fallido
	Intuition	VS950 4G	4.1.2	Fallido
	G2	VS980	4.4.2	Correcto
	Nexus 4	E960	4.4.4	Correcto

Tabla 5 Tabla Resultados

MARCA	NOMBRE	MODELO	VERSION DE S.O ANDROID	RESULTADOS
MOTOROLA	Optimus	LGMS695	2.3.6	Fallido
	Droid Razr Maxx HD	XT926	4.4.2	Correcto
	Moto X	1ST Gen	5.1	Correcto
SAMSUNG	Galaxy Pocket	GT-S5300	2.3.6	Fallido
	Nexus S	SPH-D720	2.3.6.	Fallido
	Galaxy Express	SGH-I437	4.0.4	Fallido
	Galaxy SIII	SGH-I747	4.1.1	Fallido
	Galaxy SII	GT-I9100	4.1.2	Fallido
	Nexus S	GT-I2090T	4.1.2	Fallido
	Galaxy Note II	SGH-I317	4.3	Fallido
	Galaxy 5		4.5.1	Correcto
	Galaxy 7		4.5.3	Correcto

Tabla 6 Tabla Resultados

CAPITULO
V. DISCUSIÓN

DISCUSIÓN

En el siguiente apartado se pretende mostrar un análisis acerca de los resultados obtenidos con el desarrollo del presente trabajo de titulación.

5.1 Desarrollo de la propuesta alternativa

Actualmente, existe un gran problema cuando los adultos mayores por motivos de distracción salen fuera de casa y la mayoría de ellos no logran dar con exactitud el sitio donde se encuentran; existiendo una gran ausencia de personas y de vehículos que causan zozobra en ellos por lo que deciden ir en busca de algún tipo de ayuda.

En consecuencia, los adultos mayores se encuentran con una serie de problemas para dar con su ubicación y al mismo tiempo provocando angustia y pérdida de tiempo a sus familiares que muchas veces realizan búsquedas, a través de los medios de comunicación y redes sociales, las cuales la mayoría de ellas no son exitosas. Ante ello, se ha propuesto la realización del presente trabajo de titulación como una posible solución ante esta problemática.

Objetivo Específico 1: Implementar el módulo de Geolocalización para la visualización de personas

Éste objetivo fue cumplido con ayuda de la recopilación de la información y de la fase de codificación.

Para conseguirlo se hizo uso de los datos obtenidos desde el web Service, específicamente de la latitud y longitud con el objetivo de agregar cada lugar Para desarrollar la función de la visualización de una ubicación se hizo uso de place autocomplete de google con el fin de ofrecer al usuario el servicio de autocompletado automático de una dirección; aunque también se creó la opción de que pudiese hacerlo mediante una pulsación larga dentro del mapa. Ingresando a la URL <http://181.65.182.202/otros/gpstrack/> podemos seleccionar en la parte inferior el usuario rastreado y así nos visualizará el recorrido y su posición actual.

CAPITULO

VI. CONCLUSIONES

CONCLUSIONES

Se determinó las acciones específicas de los requerimientos lo cual permitió desarrollar con mayor precisión la aplicación en Android Studio.

Se implementó el acceso web a los datos lo cual permite mayor seguridad y confiabilidad de acceso desde cualquier punto de interacción con la aplicación y el propio gestor de la base de datos MySQL.

Se implementó la aplicación en la plataforma Android lo cual permitió obtener los datos posicionales de una persona en forma sencilla y veraz.

Se utilizó la herramienta Google Maps lo cual permitió vincular los datos obtenidos por la aplicación de tal forma que posibilitó la visualización de las imágenes en tiempo real.

CAPITULO

VII. RECOMENDACIONES

RECOMENDACIONES

Motivar a los futuros egresados a implementar aplicaciones complementarias que permitan ampliar el rendimiento de la aplicación actual.

Planificar los pagos de alojamiento para evitar rupturas del mismo y por tal la continuidad de la aplicación.

Invitar a los desarrolladores de aplicaciones Android a extender la limitación de la aplicación y permitir su ejecución desde cualquier dispositivo móvil.

Desarrollar aplicaciones innovadoras que permitan aprovechar los recursos que ofrece Google en el internet.

CAPITULO
VIII. PROPUESTA

8.1 Generalidades

En este proyecto se implementará una aplicación web que con el desarrollo de una aplicación Android que está relacionada básicamente en una APK que se instala en un Smartphone con GPS activado y que, mediante conexión a internet, en cada cierto tiempo, envía al servidor web la ubicación de una persona. Posteriormente los datos emitidos se toman y en tiempo real mostrará su ubicación a través del google maps.

8.2 Análisis

- a) Se utilizará tecnología Android para la complementación del proyecto, haciendo uso del dispositivo móvil, el cual permitirá obtener la ubicación de la persona, que por algún motivo u otro no recuerda el lugar donde se encuentra, o simplemente desconoce ese sitio.
- b) Para mayor seguridad al proceso de geolocalización se utilizará siempre internet para el envío de datos al servidor web y de esta manera mostrar ubicación en tiempo real, la cual será visualizada desde el google maps.
- c) Al utilizar un dispositivo móvil smartphone, permite tener facilidades en cuanto a los nuevos avances que se están presentando hoy en día en la sociedad

8.3 Diseño

8.3.1 Actividad (formulario) localizador_gps_activity.xml:

Es la actividad principal de la aplicación Android y permite indicar el usuario y cada que intervalo de tiempo se debe enviar los datos de ubicación al servidor. Se debe pulsar el botón inferior para iniciar el envío.

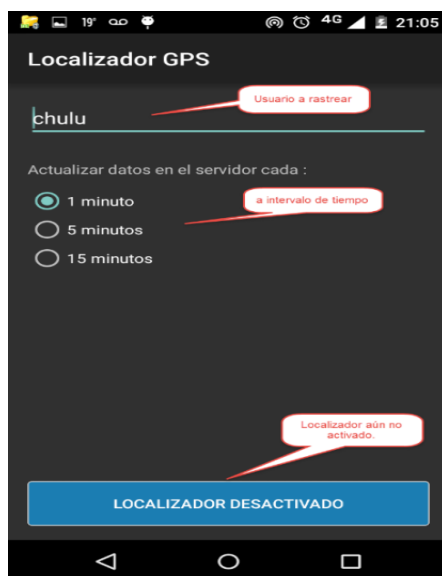
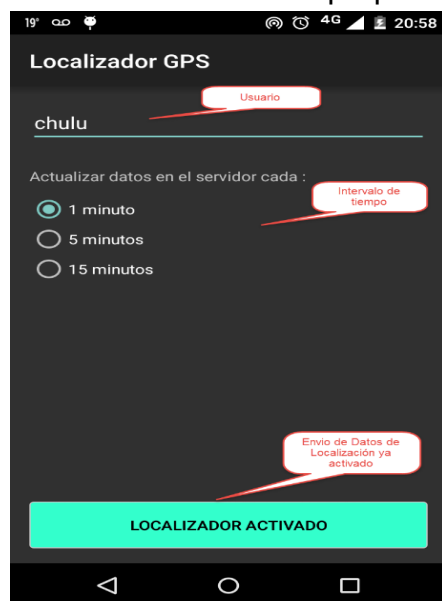


FIGURA 11 Tiempo de Actualización Aplicación

Fuente: Elaboración propia



Para detener el envío de datos se debe pulsar, nuevamente, el botón inferior.

FIGURA 12 Confirmación del objetivo localizado

Fuente: Elaboración propia

8.4 Implementación de la Aplicación

8.4.1 Página web utilizando Google Maps

Ingresando a la URL <http://181.65.182.202/otros/gpstrack/> podemos seleccionar en la parte inferior el usuario rastreado y así nos visualizará el recorrido y su posición actual.

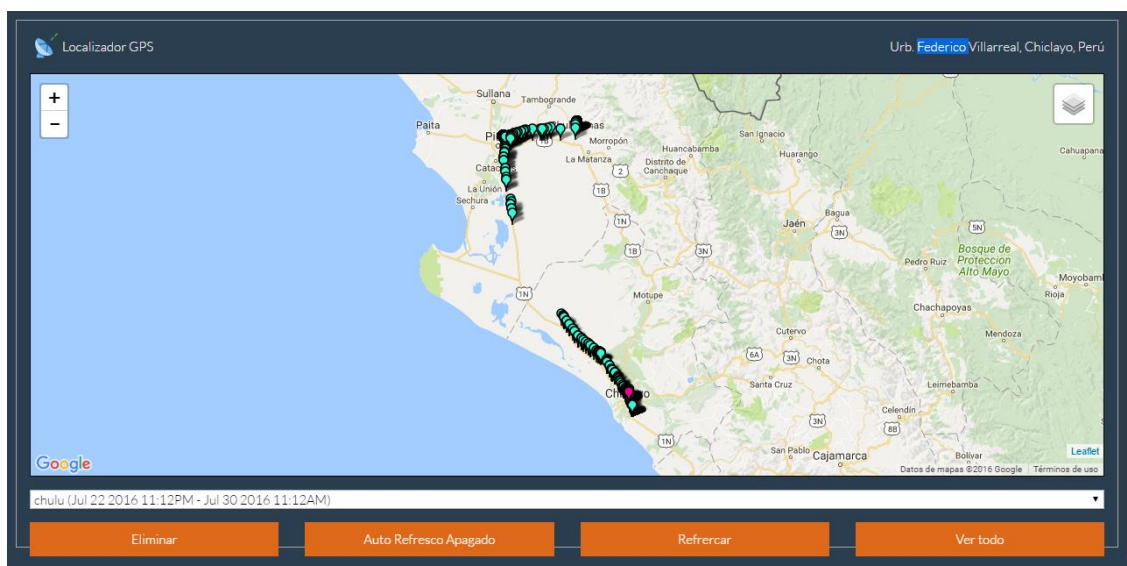


FIGURA 13 Google Maps
Fuente: Elaboración propia

8.4.2 Clase LocalizadorGPS_Activity perteneciente a la actividad principal de la aplicación:

Esta clase pertenece a la actividad principal de la aplicación y, por lo tanto, se ejecuta al momento de ejecutarse y mostrarse en pantalla.

8.4.2.1 Funciones aplicadas.

Entre las principales funciones tenemos:

Función de Inicio (OnCreate):

Básicamente las primeras acciones realizadas en esta función de inicio son:

- Referenciar a variables los controles de la actividad (activity).
- Establecer en las variables compartidas **actualmente_localizando** y **PrimeraVezObteniendoPosicion** sus valores iniciales de false y true, respectivamente.

- Definir la función a ejecutar al seleccionar una opción de la lista: GrabarIntervalo
- Definir la función a ejecutar al pulsar el botón para iniciar o detener rastreo: Rastreando_Ubicacion.

```

private static final String TAG =
    "LocalizadorGPS_Activity";
private String directorio_servidorweb;
private static EditText txtUsuario;
private static EditText txtWebsite;
private static Button BotonActivacion;
private boolean actualmente_localizando;
private RadioGroup intervaloRadioGroup;
private int intervaloEnMinutos = 1;
private AlarmManager RelogManager;
private Intent LocalizadorGPSIntent;
private PendingIntent IntentPendiente;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.localizador_gps_activity);
    directorio_servidorweb =
    getString(R.string.default_upload_website);
    txtUsuario = (EditText)findViewById(R.id.txtUsuario);
    intervaloRadioGroup =
    (RadioGroup)findViewById(R.id.RadioGroup_Intervalo);
    BotonActivacion =
    (Button)findViewById(R.id.BotonInicioFinRastreo);

    txtUsuario.setImeOptions(EditorInfo.IME_ACTION_DONE)
    ;
    SharedPreferences VariablesGlobales =

```

```

this.getSharedPreferences("com.saykodata.localizador_
gps_prefs", Context.MODE_PRIVATE);
    actualmente_localizando =
VariablesGlobales.getBoolean("actualmente_localizando
", false);

```

```

boolean primeraVezLlamandoAplicacion =
VariablesGlobales.getBoolean("PrimeraVezObteniendoP
osicion", true);

```

```

if (primeraVezLlamandoAplicacion) {
    SharedPreferences.Editor editor =
VariablesGlobales.edit();

```

```

editor.putBoolean("PrimeraVezObteniendoPosicion",
false);
    editor.putString("appID",
UUID.randomUUID().toString());
    editor.apply();
}

```

```

intervaloRadioGroup.setOnCheckedChangeListener(
    new RadioGroup.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(RadioGroup
radioGroup, int i) {
            GrabarIntervalo();
        }
    });

```

```

        BotonActivacion.setOnClickListener(new
View.OnClickListener() {
    public void onClick(View view) {
        AdministrarRastreo (view);
    }
});
}

```

Función GrabarConfiguracionUsuario:

Esta función se ejecuta al seleccionar una opción de la lista de intervalos (cada que tiempo se debe obtener la posición del dispositivo GPS).

Acciones realizadas en la función:

1. Verificar que nombre de usuario no esté vacío o contenga espacios en blanco entre los caracteres.
2. Actualizar la variable global compartida **intervaloEnMinutos** de acuerdo a la opción seleccionada.
3. Definir la variable global compartida **userName** de acuerdo a lo ingresado en la caja de texto.

```

private boolean GrabarConfiguracionUsuario() {
    if (CamposTextoEstanVacios_o_con_Espacios()) {
        return false;
    }
    SharedPreferences VariablesGlobales =
this.getSharedPreferences("com.saykodata.localizador_
gps_prefs", Context.MODE_PRIVATE);
    SharedPreferences.Editor editor =
VariablesGlobales.edit();

    switch

```

```

(intervaloRadioGroup.getCheckedRadioButtonId()) {
    case R.id.i1:
        editor.putInt("intervaloEnMinutos", 1);
        break;
    case R.id.i5:
        editor.putInt("intervaloEnMinutos", 5);
        break;
    case R.id.i15:
        editor.putInt("intervaloEnMinutos", 15);
        break;
}
editor.putString("userName",
txtUsuario.getText().toString().trim());
editor.apply();
return true;
}

```

Función AdministrarRastreo:

Esta función se ejecuta al pulsar el botón para iniciar o detener el rastreo de la posición indicada en el dispositivo GPS.

Acciones realizadas en la función:

1. Si no está disponible el servicio de google de geolocalización, salimos de la función.
2. Si actualmente ya se está enviando al servidor los datos de localización, detenemos el reloj y establecemos las variables globales a sus datos iniciales.
3. Sino, activar reloj y establecer las variables globales.
4. Cambiar el color y su texto de acuerdo sea el estado actual ("Iniciar" o "detener").

```

protected void AdministrarRastreo(View v) {
    SharedPreferences VariablesGlobales =
this.getSharedPreferences("com.saykodata.localizador_
gps_prefs", Context.MODE_PRIVATE);
    SharedPreferences.Editor editor =
VariablesGlobales.edit();

    if (!VerificaSiGooglePlayDisponible()) {
        return;
    }
    if (actualmente_localizando) {
        DesactivarReloj();
        actualmente_localizando = false;
        editor.putBoolean("actualmente_localizando",
false);
        editor.putString("sessionId", "");
    } else {
        ActivarReloj();
        actualmente_localizando = true;
        editor.putBoolean("actualmente_localizando", true);
        editor.putFloat("totalDistanceInMeters", 0f);
        editor.putBoolean("firstTimeGettingPosition", true);
        editor.putString("sessionId",
UUID.randomUUID().toString());
    }

    editor.apply();
    EstablecerEstado_BotonActivacion();
}

```

Función **ActivarReloj**:

Esta función es invocada desde la función AdministrarRastreo.

Acciones realizadas en la función:

1. Obtener el entorno (context) original de la actividad para
 - a. Obtener el servicio de alarma (reloj) del sistema.
 - b. Instanciar la clase (Intent) a ejecutar en cada intervalo de tiempo.
 - c. E indicar que se ejecute cada vez que Android reciba una notificación de la alarma.
2. Obtener de las variables globales, cada que intervalo de tiempo ejecutar la alarma (reloj).
3. Activar la alarma (reloj).

```
private void ActivarReloj() {  
    Context context = getBaseContext();  
    RelogManager =  
(AlarmManager)context.getSystemService(Context.ALARM  
_SERVICE);  
    LocalizadorGPSIntent = new Intent(context,  
LocalizadorGPS_AlarmReceiver.class);  
    IntentPendiente = PendingIntent.getBroadcast(context,  
0, LocalizadorGPSIntent, 0);  
  
    SharedPreferences VariablesGlobales =  
this.getSharedPreferences("com.saykodata.localizador_  
gps_prefs", Context.MODE_PRIVATE);  
    intervaloEnMinutos =  
VariablesGlobales.getInt("intervaloEnMinutos", 1);  
  
    RelogManager.setRepeating(AlarmManager.ELAPSED_R  
EALTIME_WAKEUP,
```



```

        SystemClock.elapsedRealtime(),
        intervaloEnMinutos * 60000, // 60000 = 1 minuto
        IntentPendiente);
    }

```

Función DesactivarReloj:

Esta función es invocada desde la función AdministrarRastreo.

Acciones realizadas en la función:

1. Obtener el entorno (context) original de la actividad para
 - a. Instanciar la clase (Intent) ejecutada en cada intervalo de tiempo.
 - b. Obtener la referencia al PendingIntent.
2. Obtener el administrador de Alarma creado previamente.
3. Cancelar dicha alarma.

```

private void DesactivarReloj() {
    Context context = getBaseContext();
    Intent LocalizadorGPS_Intent = new Intent(context,
        LocalizadorGPS_AlarmReceiver.class);
    PendingIntent pendingIntent =
        PendingIntent.getBroadcast(context, 0,
        LocalizadorGPS_Intent, 0);
    AlarmManager alarmManager =
        (AlarmManager)context.getSystemService(Context.ALARM
        _SERVICE);
    alarmManager.cancel(pendingIntent);
}

```

Clase ServicioLocalizacion:

Esta clase (servicio) es invocada cada vez que el sistema de alarma (reloj) informa que ya transcurrieron los minutos indicados por el

usuario. El objetivo de esta función es activar el servicio de localización (GPS) del dispositivo, obtener las coordenadas actuales (longitud, latitud, etc) y enviarlos al servidor web utilizando el método GET del protocolo HTTP.

Entre las principales funciones tenemos:

Función **onStartCommand**:

Esta función es invocada automáticamente cada vez que se invoca al servicio y permite iniciar el servicio de localización de Android.

```
public int onStartCommand(Intent intent, int flags, int
startId) {
    // Si estamos actualmente intentado obtener la ubicacion
    y el Administrador de Alarma se ha invocado nuevamente ,
    // no hay necesidad de procesar una nueva solicitud de
    ubicacion actual.

    if (!actualmente_procesando_localizacion) {
        actualmente_procesando_localizacion = true;
        Iniciar_ServicioLocalizacion();
    }

    return START_NOT_STICKY;
}
```

Función **Iniciar_ServicioLocalizacion**:

Esta función es invocada por la función de inicio de servicio: onStartCommand y permite, si está disponible, activar el servicio de localización del sistema operativo Android.

```
private void Iniciar_ServicioLocalizacion() {
    if
    (GooglePlayServicesUtil.isGooglePlayServicesAvailable(this
s) == ConnectionResult.SUCCESS) {
```

```

        googleApiClient = new GoogleApiClient.Builder(this)
            .addApi(LocationServices.API)
            .addConnectionCallbacks(this)
            .addOnConnectionFailedListener(this)
            .build();

        if (!googleApiClient.isConnected() ||
!googleApiClient.isConnecting()) {
            googleApiClient.connect();
        }
    } else {
        // No se pudo conectar a los servicios de Google Play
    }
}
}

```

Función onLocationChanged:

Esta función es invocada cada vez que se ha cambiado de posición y permite enviar los datos de la nueva ubicación al servidor web.

```

public void onLocationChanged(Location location) {
    if (location != null) {
        if (location.getAccuracy() < 500.0f) {
            Detener_Actualizaciones_Ubicacion ();
            EnviarDatos_Localizacion_WebSite(location);
        }
    }
}
}

```

Función EnviarDatos_Localizacion_WebSite:

Esta función es invocada desde el evento onLocationChanged y permite enviar los datos al servidor web.

protected void

```
EnviarDatos_Localizacion_WebSite(Location location) {  
    DateFormat dateFormat = new  
SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  
    dateFormat.setTimeZone(TimeZone.getDefault());  
    Date date = new Date(location.getTime());  
    SharedPreferences VariablesGlobales =  
this.getSharedPreferences("com.saykodata.localizador_  
gps_prefs", Context.MODE_PRIVATE);  
    SharedPreferences.Editor editor =  
VariablesGlobales.edit();  
    boolean primeravez_obteniendo_posicion =  
VariablesGlobales.getBoolean("PrimeraVezObteniendoP  
osicion", true);  
    if (primeravez_obteniendo_posicion) {  
  
editor.putBoolean("PrimeraVezObteniendoPosicion",  
false);  
    } else {  
        Location previousLocation = new Location("");  
  
previousLocation.setLatitude(VariablesGlobales.getFloat("LatitudPrevia", 0f));  
  
previousLocation.setLongitude(VariablesGlobales.getFloat  
("LongitudPrevia", 0f));  
    }  
}
```

```

        editor.putFloat("LatitudPrevia",
(float)location.getLatitude());
        editor.putFloat("LongitudPrevia",
(float)location.getLongitude());
        editor.apply();

        final RequestParams ParametrosRequest = new
RequestParams();
        ParametrosRequest.put("latitud",
Double.toString(location.getLatitude()));
        ParametrosRequest.put("longitud",
Double.toString(location.getLongitude()));
        ParametrosRequest.put("username",
VariablesGlobales.getString("userName", ""));
        ParametrosRequest.put("phonenummer",
VariablesGlobales.getString("applD", ""));
        ParametrosRequest.put("sessionid",
VariablesGlobales.getString("sessionID", ""));

        final String uploadWebsite =
VariablesGlobales.getString("directorio_servidorweb",
directorio_servidorweb);

        Llamada_HttpCliente.get(uploadWebsite,
ParametrosRequest, new AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode,
cz.msebera.android.httpclient.Header[] headers, byte[]
responseBody) {
                Llamada_HttpCliente.debugLoopJ(TAG, Envio

```

```

Datos Servidor -Exitoso", uploadWebsite,
ParametrosRequest,
responseBody, headers, statusCode, null);
    stopSelf();
}
@Override
public void onFailure(int statusCode,
cz.msebera.android.httpclient.Header[] headers, byte[]
errorResponse, Throwable e) {
    Llamada_HttpCliente.debugLoopJ(TAG, "Envio
Datos Error", uploadWebsite, ParametrosRequest,
errorResponse, headers, statusCode, e);
    stopSelf();
}
});
}

```

8.4.3 Tablas implementadas

8.4.3.1 Tabla Gps_localizaciones

Columna	Tipo	Nulo	Predeterminado
Gps_id (Primaria)	int(10)	No	
fecharegistro	Timestamp	No	CURRENT_TIMESTAMP
Latitud	decimal(10,7)	No	0.0000000
Longitud	decimal(10,7)	No	0.0000000
id_dispositivo	varchar(50)	No	
Usuario	varchar(50)	No	
sesionID	varchar(50)	No	
Dirección	int(10)	No	0
Precisión	int(10)	No	0
Data_extra	varchar(255)	No	
tipoevento	varchar(50)	No	Android

Tabla 7 Tabla de Geolocalización

Fuente: Elaboración propia

8.4.3.2 Índices

Nombre de la clave	Tipo	Único	Columna	Nulo	Comentario
PRIMARY	BTREE	Sí	Gps_id	No	
Idx_sesionID	BTREE	No	sesionID	No	
Idx_id_dispositivo	BTREE	No	id_dispositivo	No	
Idx_usuario	BTREE	No	usuario	No	

Tabla 8 Tabla de índices

Fuente: Elaboración propia

8.4.4 Instalación de la aplicación Android

A. Descargar de la dirección web indicada la aplicación “localizadorGPS.apk”:



FIGURA 14 Acceso a la aplicación

Fuente: Elaboración propia.

A. Para ello escribimos en el navegador web la siguiente dirección:
<http://181.65.182.202/descargas/localizadorGPS.apk>

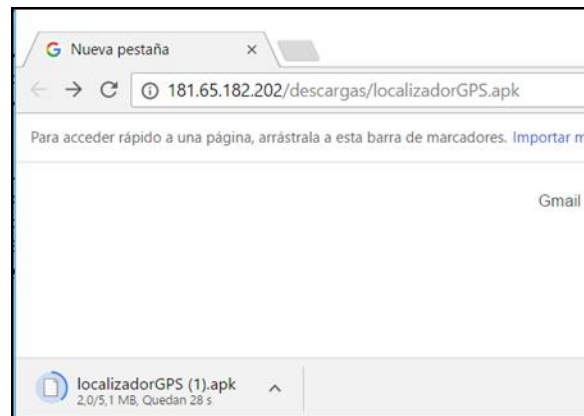


FIGURA 15 Descarga de la aplicación

Fuente: Elaboración propia

A. Luego de ello, ubicamos el archivo descargado en nuestra carpeta del dispositivo movil:

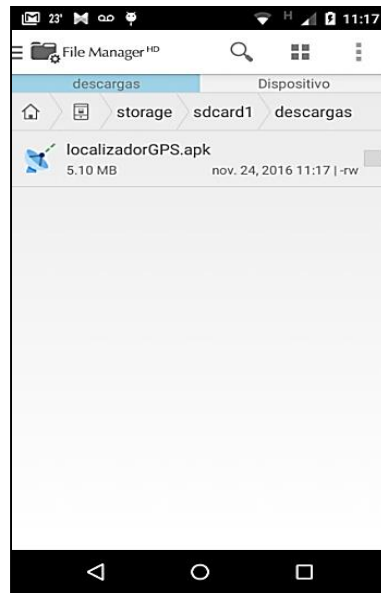


FIGURA 16 Aplicación descargada

Fuente: Elaboración propia

A. Pulsamos sobre el instalador para ejecutarlo, con lo cual aparece la interfaz con los requerimientos y permisos que se necesitan para que se ejecute la aplicación:



FIGURA 17 Configuración de permisos de instalación

Fuente: Elaboración propia

Una vez instalado, pulsamos sobre el botón “Abrir”:

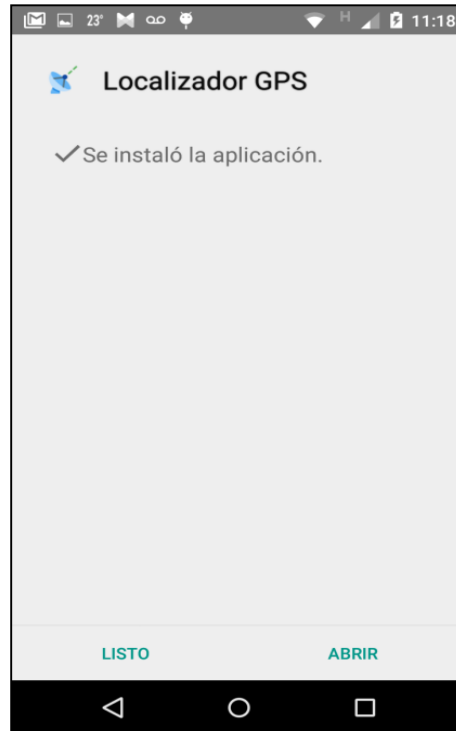


FIGURA 18 Fin de instalación.

Fuente: Elaboración propia

8.5 Costo y presupuesto

8.5.1 Bienes y servicios

DESCRIPCION	CANTIDAD	COSTO UNITARIO	SUBTOTAL
BIENES			
Papel bond	50	0.05	2.50
Lapiceros	2	0.50	1.00
Libros	1	85.00	85.00
SERVICIOS			
Fotocopiado	200	0.10	20.00
Escaneado	10	0.50	5.00
Impresión	250	0.10	46.50
Anillados	1	3.50	3.50
TOTAL			S/ 154.50

Tabla 9 Tabla bienes y servicios

Fuente: Elaboración propia

Otros

DESCRIPCIÓN	CANTIDAD	COSTO UNITARIO	SUBTOTAL
Internet			30.00
Pasajes	—	—	60.00
Llamadas	—	—	25.00
TOTAL			S/. 115.00

Tabla 10 Tabla otros

Fuente: Elaboración propia

8.5.2 Financiamiento

En el presente proyecto respecto a su diseño, simulación y presentación, se obtuvieron los costos autofinanciados por los integrantes del proyecto.

8.5.2.1 Presupuesto

- **Equipo de desarrollo**

ESPECIALISTA	Nº HORAS	COSTO/HORA	TOTAL
Diseñador	45	30.00	1350.00
Programador	30	20.00	600.00
TOTAL			1950.00

Tabla 11 Tabla equipo de desarrollo

Fuente: Elaboración propia

Resumen del Presupuesto:

DESCRIPCIÓN	SUBTOTAL
Equipo de desarrollo	1950.00
Bienes y servicios	154.50
Otros	115.00
TOTAL	2219.50

Tabla 12 Tabla resumen del presupuesto

Fuente: Elaboración propia

CAPITULO

IX. REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRAFICAS

Anddev.org, comunidad de desarrolladores de Android (en inglés). Último acceso en septiembre de 2016.

<http://www.anddev.org/>

Android Community, entrevista a Jason Chen, desarrollador de Android. Último acceso en octubre de 2016

<http://androidcommunity.com/jason-chen-answers-questions-about-android-20080603/>

Android Developers Blog, blog de los desarrolladores de Android. Último acceso en septiembre de 2016.

<http://android-developers.blogspot.com/2008/08/some-information-on-apis-removed-in.html>

Android, el sistema operativo de Google. Último acceso, Octubre de 2016.

<http://code.google.com/intl/es-ES/android/index.html>

Descripción de la arquitectura de Android, por Google. Último acceso en septiembre de 2016.

<http://es.youtube.com/watch?v=QBGfUs9mQYY>

Descripción del ciclo de vida de una aplicación Android, por Google. Último acceso en septiembre de 2016

<http://es.youtube.com/watch?v=fL6gSd4ugSI>

El Mundo, edición digital. Noticia de la sección El Navegante, publicada el 5 de noviembre de 2007. Último acceso en septiembre de 2016

<http://www.elmundo.es/navegante/2007/11/05/tecnologia/1194284462.html>

Guía de instalación del SDK de Android. Web del proyecto Android. Último acceso en septiembre de 2016

<http://code.google.com/android/intro/installing.html>

MyGeoPosition, servicio de obtención de coordenadas. Última visita en octubre de 2016.

<http://www.mygeoposition.com>

“Motorola se cambia a Android”, entrada publicada en el blog Gizmóvil sobre tecnologías y dispositivos móviles el 19 de diciembre de 2008. Última visita en noviembre de 2016.

<http://gizmovil.com/2008/12/motorola-se-cambia-a-android/>

Página oficial del HTC G1, primer dispositivo comercial con Android._
<http://www.htc.com/www/product/g1/overview.html>

SDK de Android. Web del proyecto Android. Último acceso en septiembre de 2008.
<http://code.google.com/android/download.html>

“Desarrollo de una aplicación móvil Android para la búsqueda de plazas disponibles en un parqueadero” 2005.
<http://dspace.unl.edu.ec/jspui/bitstream/123456789/11582/1/Chinchay%20Cuenca%2C%20Marjorie%20Juliana.pdf>

CAPITULO

X. ANEXOS

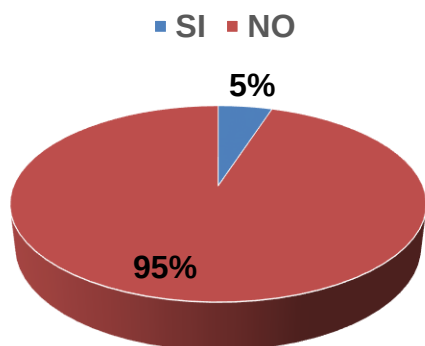
ANEXO N° 01: Medición de percepción sobre el seguimiento de personas mayores de edad en el Pueblo Joven “Ricardo Palma”

Esta medición fue realizada en base a encuestas que ayudarán a determinar los recursos que disponen los usuarios de dispositivos móviles para poder utilizar nuevas aplicaciones

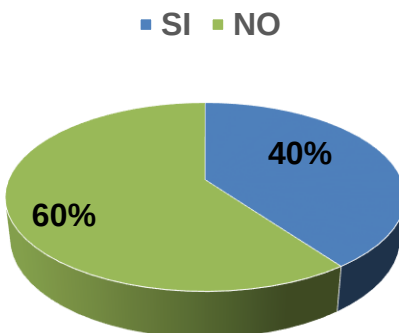
Para llevar a cabo esta encuesta, se le aplicó a muestra de 300 usuarios del Pueblo Joven “Ricardo Palma”, para determinar el nivel de protección que existe hacia el adulto mayor y al mismo tiempo saber el conocimiento que tienen sobre la existencia de nuevas aplicaciones en los dispositivos móviles.

Encuesta

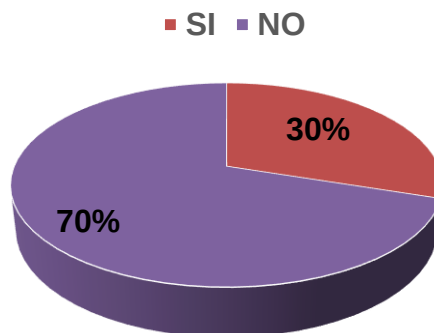
¿Ha utilizado nuevas aplicaciones en sus equipos móviles?



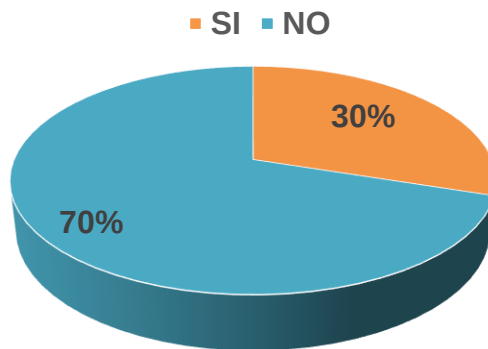
¿Se ha propuesto alguna vez adquirir nuevas aplicaciones para su celular?



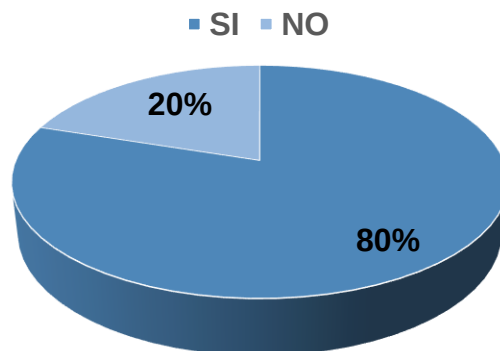
¿Presenta algún problema de salud mental?



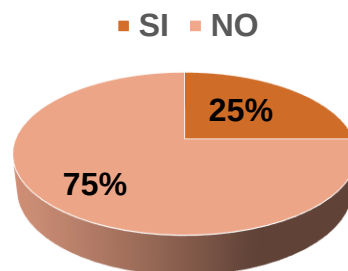
¿Alguna vez se ha llegado a perder cuando salió a otro lugar?



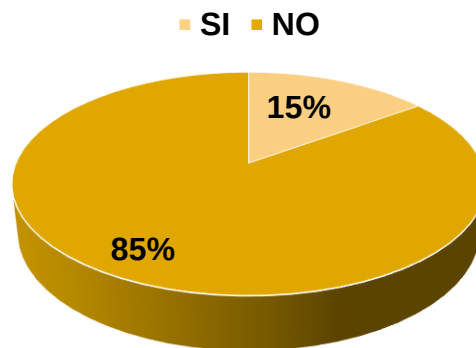
¿Han realizado algún mecanismo para saber sobre su paradero?



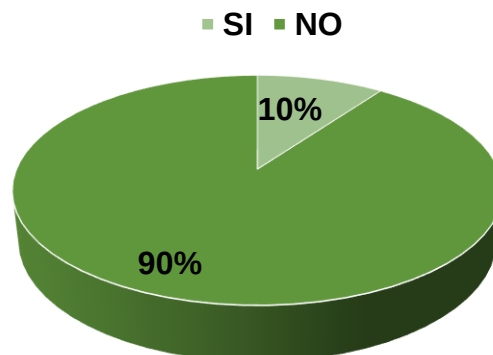
¿Le ayudó bastante el mecanismo realizado?



¿Cree que existe protección hacia usted?



¿Cuenta con medidas de protección fuera de casa?



ANEXO N° 02: Instalación de Android Studio

Google ha anunciado, tras la conferencia Google I/O 2013, la aparición de un entorno de desarrollo específico para Android, llamado Android Studio. Este entorno de desarrollo, basado en el IDE IntelliJ IDEA, de la empresa JetBrains, está disponible de manera gratuita, aunque a día de hoy se ofrece en versión "Early Access". Por este motivo, es preferible ser cauto y meticuloso antes de iniciar un proyecto de producción (incluso aunque la tendencia parece mejorar, las primeras actualizaciones de Android Studio han presentado problemas en el pasado).

A fecha 1 de febrero de 2014, la versión más reciente es la v 0.4.2. El vínculo para descargar el archivo de instalación es el siguiente: <http://developer.android.com/sdk/installing/studio.html>

Comience leyendo y aceptando la licencia de uso (Terms and Conditions) y,

A continuación, descargue la versión propuesta para su sistema operativo.

La instalación de Android Studio es muy sencilla:

Inicie la ejecución del archivo descargado y, a continuación, siga el asistente de instalación: seleccione la carpeta de instalación que desee y haga clic en Install.

La instalación termina y se abre un cuadro de diálogo que le pregunta si desea iniciar la aplicación.



Seleccione la carpeta de destino y haga clic en Next.



La aplicación arranca y ofrece diversas opciones. Seleccione New Project.... ¡La instalación de Android Studio ha terminado!

Observe que, como ocurre con el entorno Eclipse, Android SDK Manager se instala al mismo tiempo que la aplicación. Su funcionalidad Es idéntica a la versión que hemos estudiado antes en este capítulo.

Primer proyecto Android

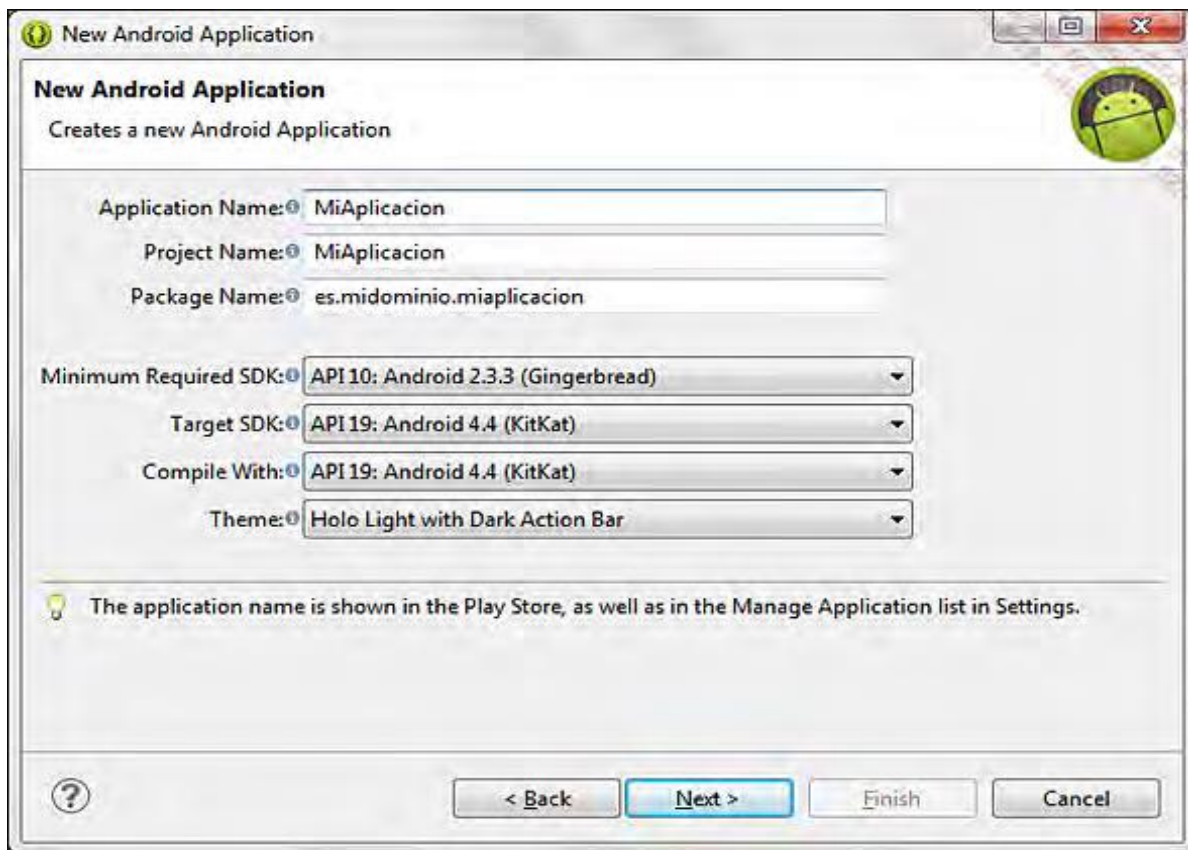
Con el entorno de desarrollo ya instalado y listo para su uso, vamos a crear sin esperar más nuestro primer proyecto Android bajo Eclipse. A continuación, lo ejecutaremos en el emulador de Android y, eventualmente, sobre un dispositivo Android.

A continuación, en la sección siguiente, descubriremos la estructura detallada de un proyecto Android.

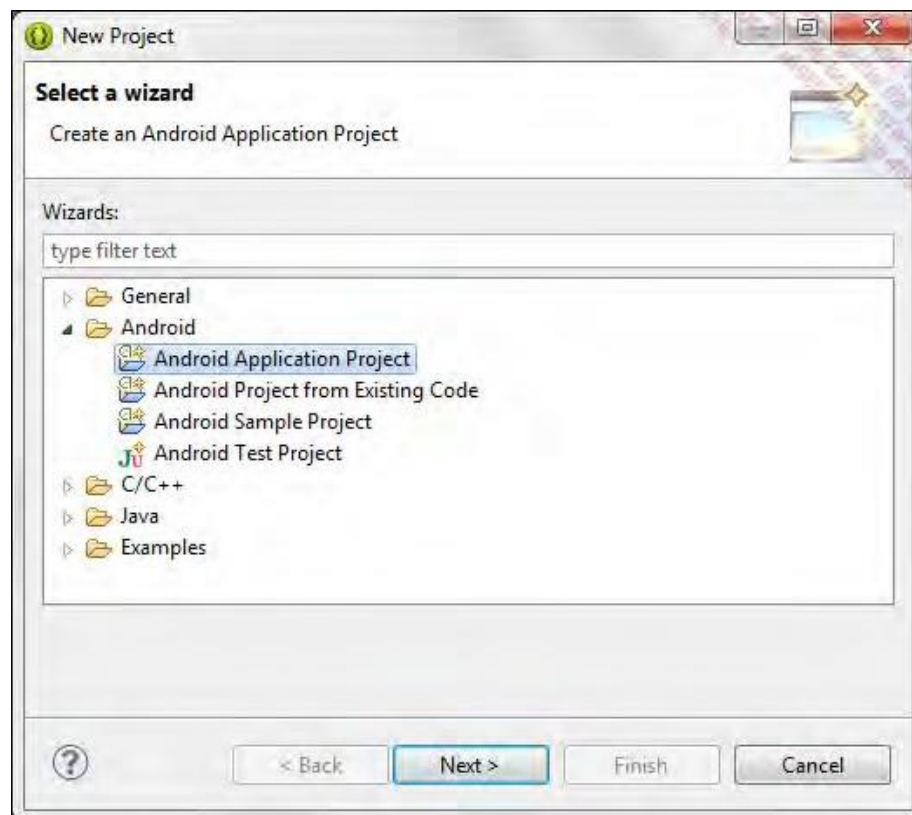
1. Creación del proyecto

En el menú general de Eclipse, seleccione File - New - Project.

Aparece la ventana New Project, que muestra los distintos asistentes para la creación de proyectos.



Haga clic en el botón Next.



Despliegue la carpeta Android, seleccione Android Application Project y haga clic en el botón Next.

- ✓ Aparece a continuación la ventana New Android Application Project del asistente de creación de proyectos Android.
- ✓ Es prApplication Name: nombre de la aplicación, tal y como aparecerá en la Play Store y en el título que se muestra en la pantalla del dispositivo. En este primer ejemplo, seleccionamos el nombre MiAplicacion.
- ✓ Project Name: nombre del proyecto Android y de la carpeta que contiene el proyecto. Por defecto, el asistente propone el mismo nombre que el de la aplicación.
- ✓ Package Name: nombre del paquete Java en el que se generará el código fuente. Éste sirve a su vez como identificador único de la aplicación para Play Store. No debe cambiar una vez publicada la aplicación en Play Store. Es preciso utilizar un formato estándar de nombre de dominio. En nuestro ejemplo, utilizaremos enciso informar numerosos campos. He aquí su correspondencia:

es.midominio.miaplicacion.

- ✓ Minimum Required SDK: versión mínima requerida para que un dispositivo pueda ejecutar la aplicación. Se recomienda seleccionar el nivel de API más bajo posible, en base a las restricciones impuestas, para poder alcanzar a una audiencia lo más amplia posible. En este primer ejemplo, vamos a seleccionar la versión 2.3.3 (API 10). La selección de una API mínima y sus consecuencias se discutirán en el capítulo Completar la interfaz de usuario.
- ✓ Target SDK: indica qué nivel máximo de API está comprobado para esta aplicación. Permite asegurar al sistema que no debe preocuparse de la compatibilidad de la aplicación para todos los dispositivos que dispongan de una versión del sistema inferior o igual a la indicada. En este ejemplo, seleccionaremos la versión 4.4 (API 18).
- ✓ Compile With: permite especificar la versión objetivo de la plataforma Android utilizada para compilar la aplicación. Se recomienda seleccionar el

mismo nivel de API que para Target SDK.

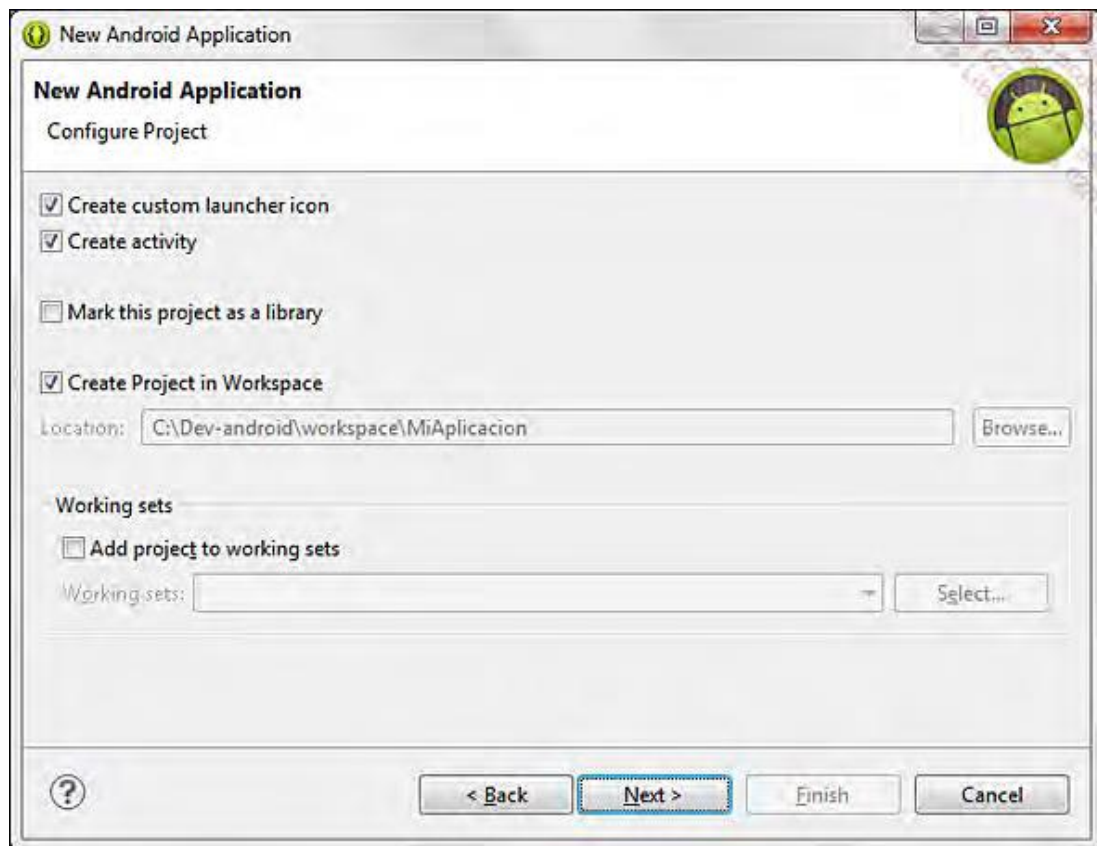
- ✓ Theme: aquí indicaremos qué tema base utilizaremos para el diseño de la aplicación (color, tipografía, etc.). Seleccionaremos el tema Holo Light with Dark Action Bar, que parece ser el más extendido.

Estas opciones pueden modificarse una vez creado el proyecto, bien sean específicas a Eclipse o a Android.

El asistente nos permite, a continuación, definir opciones relativas por un lado a elementos que se integrarán automáticamente en el proyecto y, por otro, a la configuración de Eclipse para este proyecto:

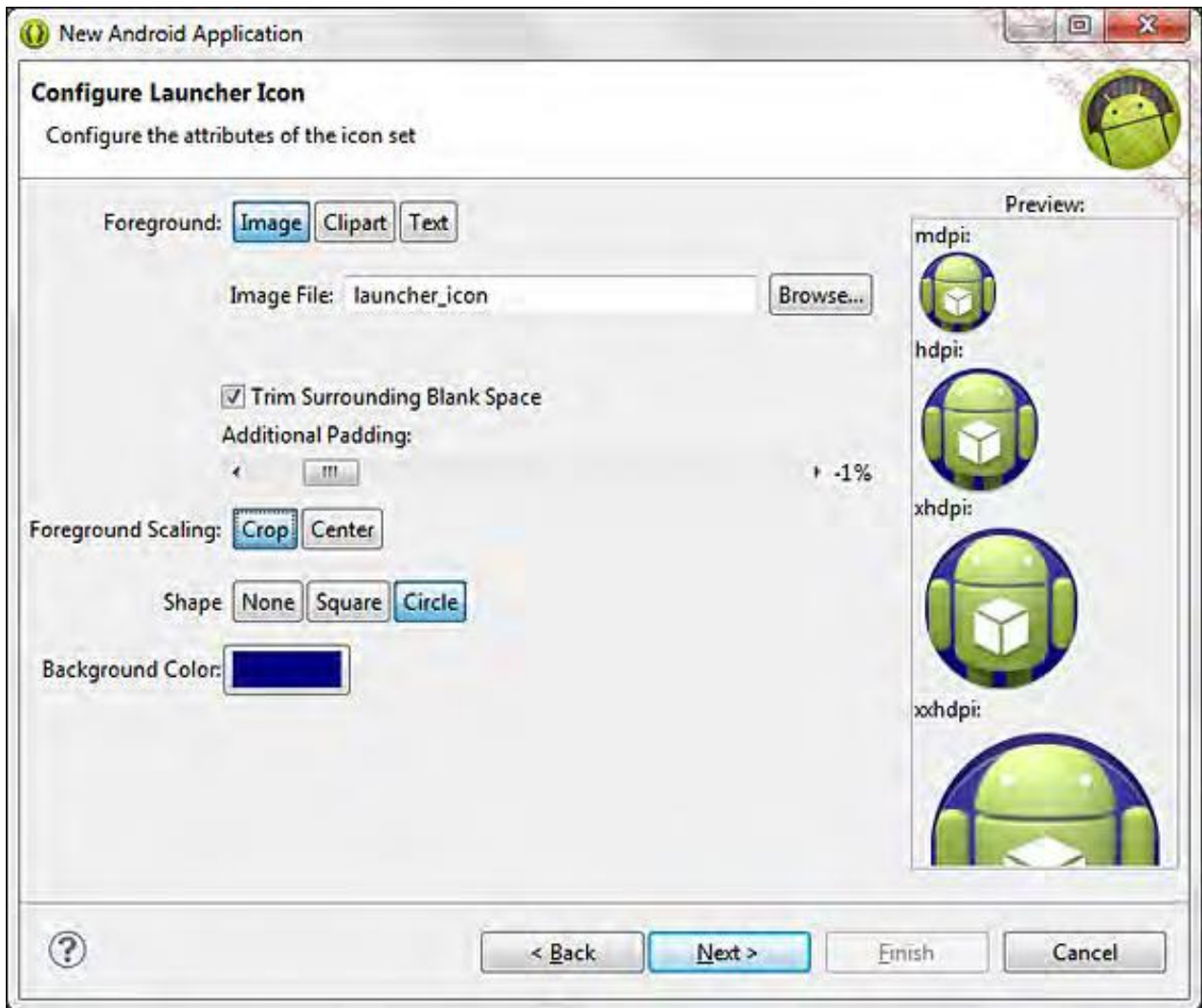
- Create custom launcher icon: permite indicar al asistente si debe presentarnos la ventana de creación de un icono de inicio de la aplicación. En un proyecto de producción su interés es bastante limitado, aunque para un proyecto de ejemplo, supone un ahorro de tiempo apreciable. Por esta vez, marcaremos la opción.
- Create activity: marcaremos esta opción, que permite crear automáticamente una primera pantalla (llamada activity en el universo Android) para nuestra aplicación; pantalla que servirá como punto de entrada de la aplicación.
- Mark this project as a library: permite indicar que el proyecto no es una aplicación directamente ejecutable, sino una librería que se utilizará en otras aplicaciones. Dejamos la opción desmarcada para crear una aplicación directamente ejecutable.
- Create Project in Workspace: esta opción afecta a la configuración de los proyectos en Eclipse. Cada proyecto se integra en un workspace (espacio de trabajo), materializado por una carpeta del dispositivo de almacenamiento correspondiente. El proyecto se crea en una subcarpeta de dicho workspace. Dejamos la opción marcada para mantener la ubicación por defecto propuesta por Eclipse.
- Add project to working sets: Eclipse permite definir working sets (grupos de trabajo), que permiten agrupar varios proyectos en un mismo grupo.

Esta funcionalidad resulta útil cuando se trabaja con un gran número de proyectos en el mismo workspace y se desea establecer subconjuntos de proyectos para trabajar con una mayor claridad. Dejamos esta opción desmarcada.



Haga clic en el botón Next para pasar al asistente de creación de icono.

Esta pantalla nos permite seleccionar una imagen, un clipart o texto como base de nuestro icono de inicio de la aplicación. Es posible redimensionar el elemento seleccionado, ajustar el color de fondo, etc. Dejaremos libertad de creación a este nivel. Observe la sección derecha de la ventana que muestra cómo se generan varias calidades de imagen (las cuales son necesarias) para adaptarse a las distintas configuraciones de pantalla



Una vez creado el icono, haga clic en el botón Next que permite especificar La actividad que se creará automáticamente.

El asistente le propone tres modelos de pantalla:

- Blank Activity: una pantalla vacía, con una barra de acción (action bar),

en el formato por defecto de Android.

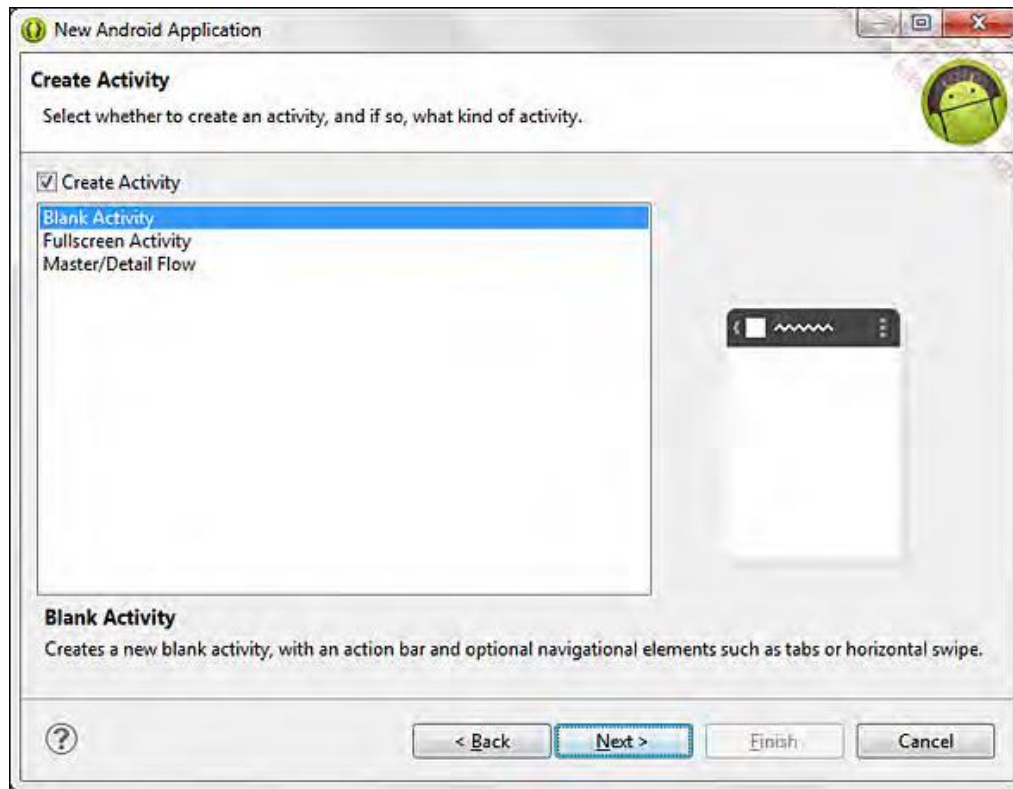
- Fullscreen Activity: una pantalla vacía, sin barra de acción, que ocupa la totalidad de la pantalla del terminal.

Master/ Detail Flow: una pantalla dividida en dos partes, la sección de la izquierda presentando una lista de elementos y la sección de la derecha una zona de detalle para el elemento seleccionado. El interés de dicha

configuración automática consiste en integrar a la vez los mecanismos de presentación de pantallas de gran dimensión (tabletas) y pantallas de smartphones.

También puede, adicionalmente, desmarcar la opción Create Activity para no crear esta primera actividad.

En el marco de nuestro primer ejemplo, seleccionaremos una actividad de tipo Blank Activity.



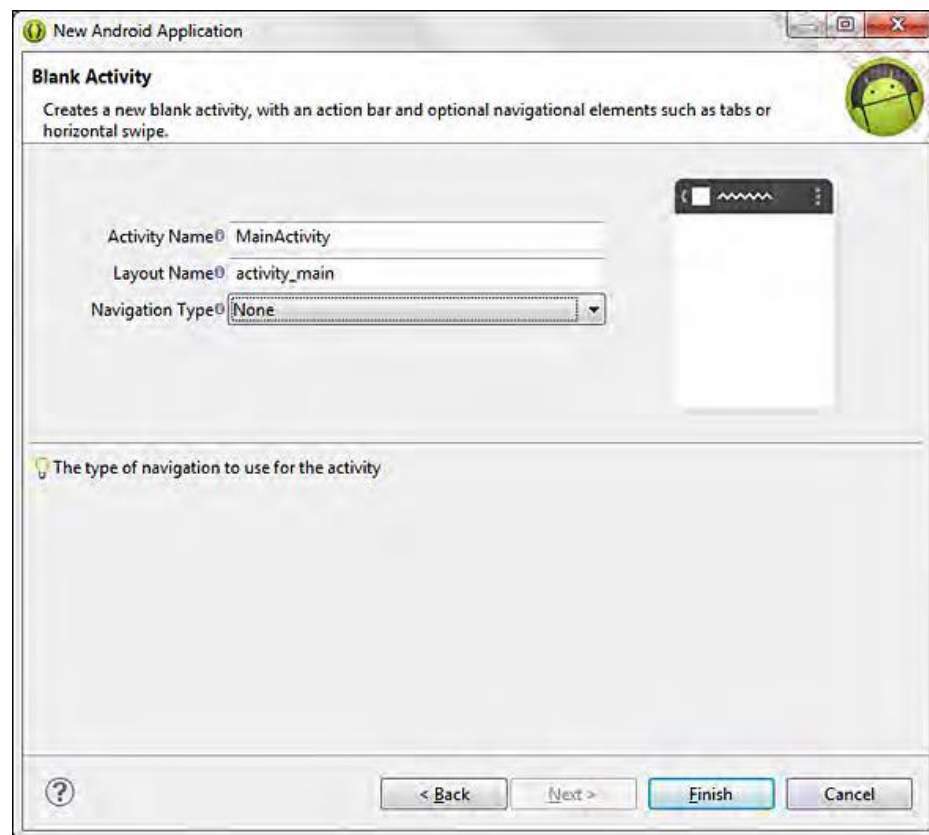
Haga clic en el botón Next para indicar el nombre de la actividad que se creará.

La pantalla nos presenta, a continuación, las opciones disponibles para la Creación de nuestra actividad.

- Activity Name: registrará, aquí, el nombre de su actividad. Una actividad es, desde el punto de vista de la programación, una clase java, de modo que aquí indicará el nombre de la clase. Dejaremos la propuesta por

defecto, MainActivity.

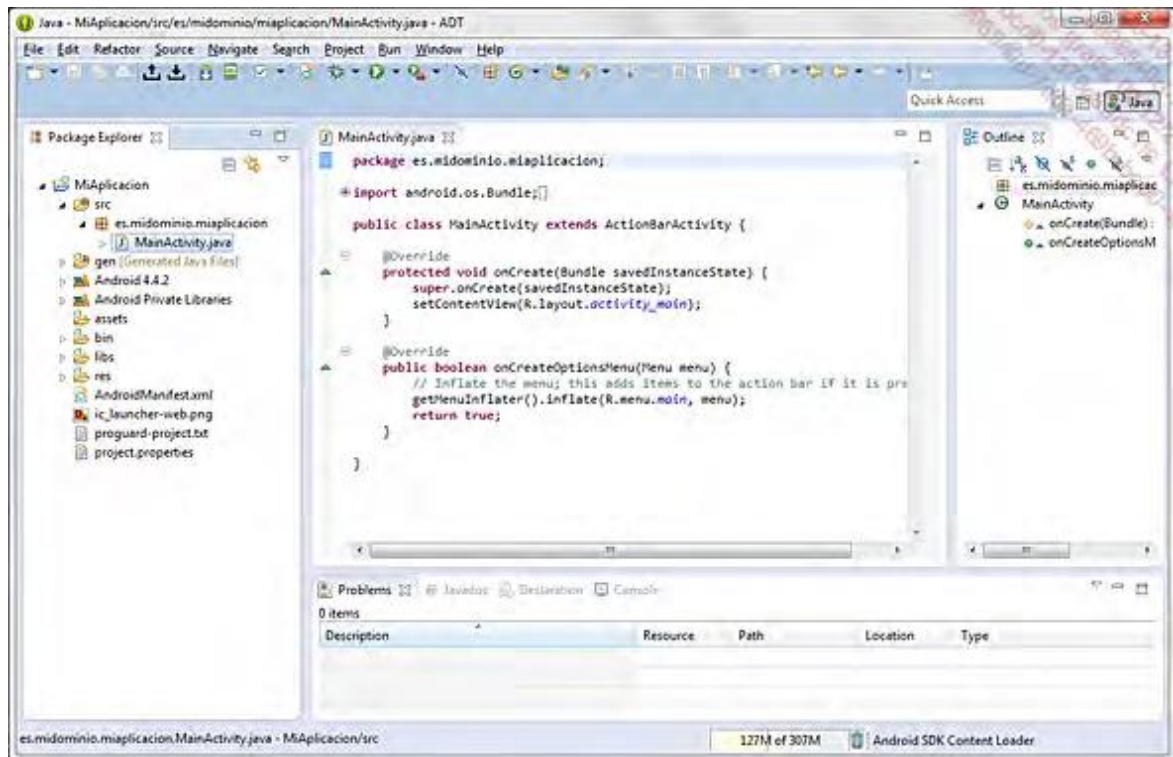
- **Layout Name:** en general, la composición de una pantalla - composición de distintos elementos - se define en un archivo XML, llamado layout. Los layouts se estudian con detalle a lo largo de este libro, de modo que no diremos mucho más en este punto y dejaremos el nombre propuesto por defecto, `activity_main`.
- **Navigation Type:** permite crear automáticamente el mecanismo de navegación que se implementará para la actividad. La lista desplegable permite, a continuación, seleccionar entre varias opciones: None (sin navegación prevista), Fixed Tabs + Swipe (sistema de pestañas con navegación por movimientos de barrido), Scrollable Tabs + Swipe (pestañas deslizantes con navegación por movimientos de barrido) o Dropdown (lista desplegable en la barra de acción, que hace las veces de menú). Seleccionaremos None, nuestra aplicación es una simple pantalla como primer ejercicio



Haga clic en el botón Finish para iniciar la creación del proyecto

Se nos propone crear un proyecto de prueba. No haremos nada de momento pues este tema se abordará en otro capítulo del libro (véase el capítulo Trazas, depuración y pruebas - Pruebas unitarias y funcionales).

Se genera el proyecto en el espacio de trabajo de Eclipse y se agrega a la vista Package Explorer (Explorador de paquetes). Se compila automáticamente.



La compilación del proyecto conlleva la creación de un archivo binario

MiAplicacion.apk en la carpeta **bin** del proyecto.

Se trata de un archivo de formato apk (Android Package), es decir, un archivo zip que contiene todo el proyecto: código compilado, datos y archivos de recursos.

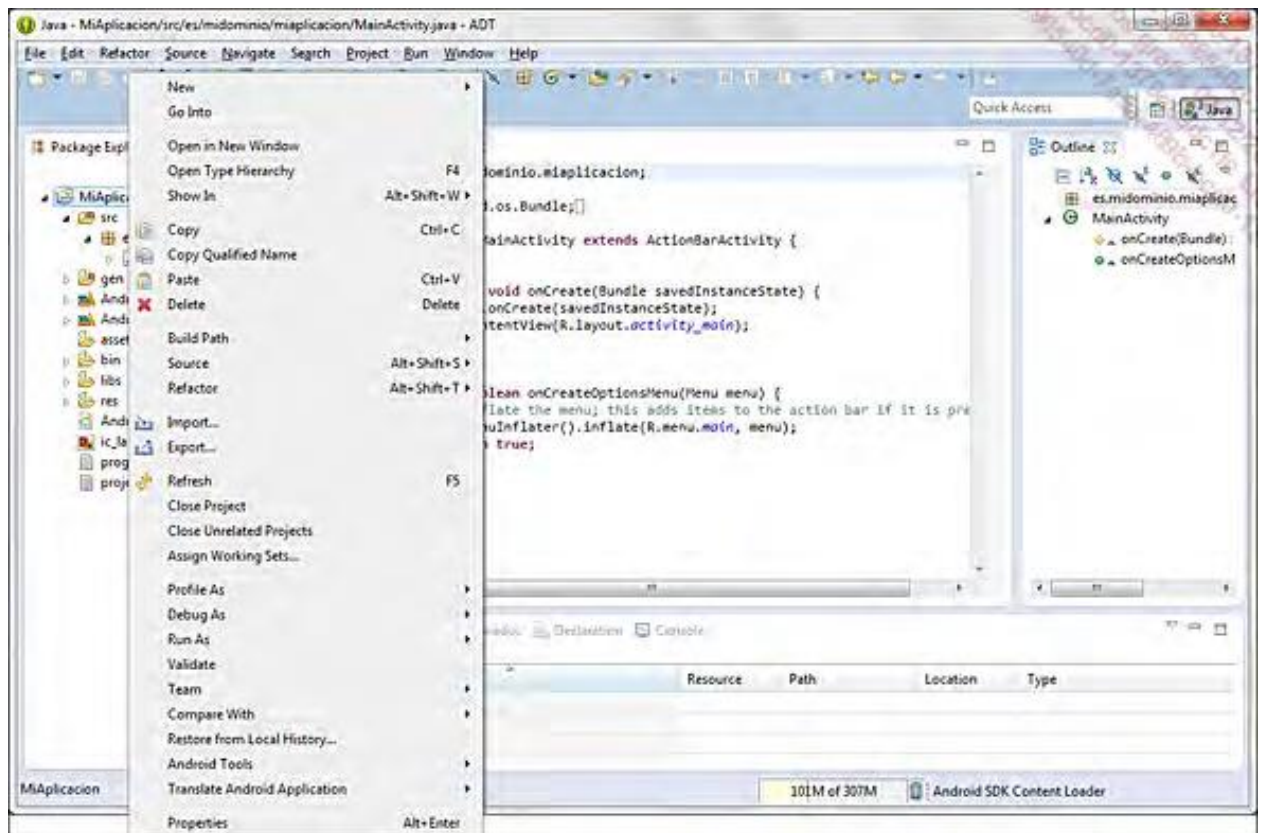
Debido a que este archivo contiene la aplicación, es habitual referirse a

este archivo usando directamente el término aplicación.

2. Ejecución de la aplicación

Ahora que el proyecto ha sido creado con éxito, y que la aplicación ha sido generada, sólo queda instalarla en un sistema Android y ejecutarla. Para ello:

En la vista Package Explorer de Eclipse, Haga clic con el botón derecho sobre el proyecto Mi Aplicación.



A continuación, haga clic en Run As - Android Application.

La vista Consola permite seguir el avance de carga de la aplicación y de su ejecución.

Si aparece en la consola un mensaje de error del tipo The connection to adb is down, and a severe error has occurred, reinicie Eclipse e inténtelo de nuevo.

Aparece una ventana de diálogo Android AVD Error informándonos de que no existe ningún AVD (Android Virtual Device), es decir una configuración del emulador Android o un dispositivo Android conectado que posea una versión igual o superior a la versión mínima indicada durante la creación del proyecto. Se nos propone crear la configuración de un dispositivo a emular. Haga clic en el botón No y cierre la ventana Android Device Chooser. Volveremos a ella más tarde.

Existen dos posibilidades a la hora de ejecutar la aplicación:

- Crear un AVD, un periférico Android virtual, y ejecutarlo sobre el emulador.
- Utilizar un dispositivo Android real, como un Smartphone o una tableta táctil, conectado al ordenador.

a. En el emulador Android

El SDK proporciona un emulador de periféricos Android. Este emulador permite emular numerosas configuraciones de hardware. Un AVD es una configuración de hardware concreta que utilizará el emulador.

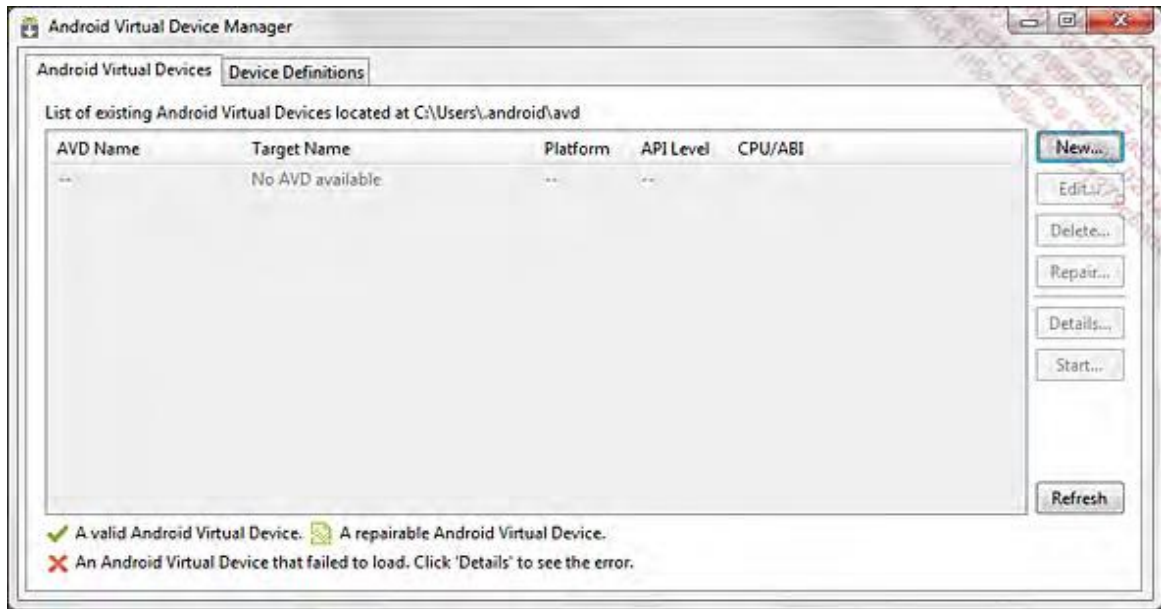
Veamos cómo se crea un AVD.

Ejecute la herramienta Android Virtual and Device Manager (AVD MANAGER)

Recuerde, la herramienta AVD Manager puede ejecutarse desde la barra de herramientas haciendo clic en su icono o seleccionado Window - Android Virtual Device Manager.

A continuación, se abre la herramienta Android Virtual Device Manager

indicando que no existe ningún AVD.



Haga clic en el botón New ... situado arriba a la derecha. Aparece la ventana Create new Android Virtual Device (AVD). Para crear el AVD es preciso informar numerosos campos:

- AVD Name: nombre del AVD. No debe contener caracteres especiales, ni espacios.
- Este nombre debe, a su vez, ser corto de cara a tener una mayor comodidad dado que se utiliza por línea de comandos o en una consola, y ser lo suficientemente explícito para caracterizar la configuración del AVD.
- Device: lista de dispositivos Android de configuración estándar. La lista de dispositivos Nexus está presente, así como un conjunto de configuraciones estándar de pantalla, desde QVGA (240 x 320, baja definición) hasta WXGA (1280 x 720, ultra alta definición) y más. La selección de un dispositivo Nexus permite partir de opciones que se corresponden con las especificaciones del dispositivo seleccionado.
- Target: lista que proporciona las plataformas disponibles descargadas e instaladas sobre el puesto de desarrollo. Recuerde, la versión de la API de la plataforma escogida deberá ser igual o superior a la versión mínima de la API especificada en la aplicación.
- CPU/ABI: lista que permite especificar qué tipo de procesador se

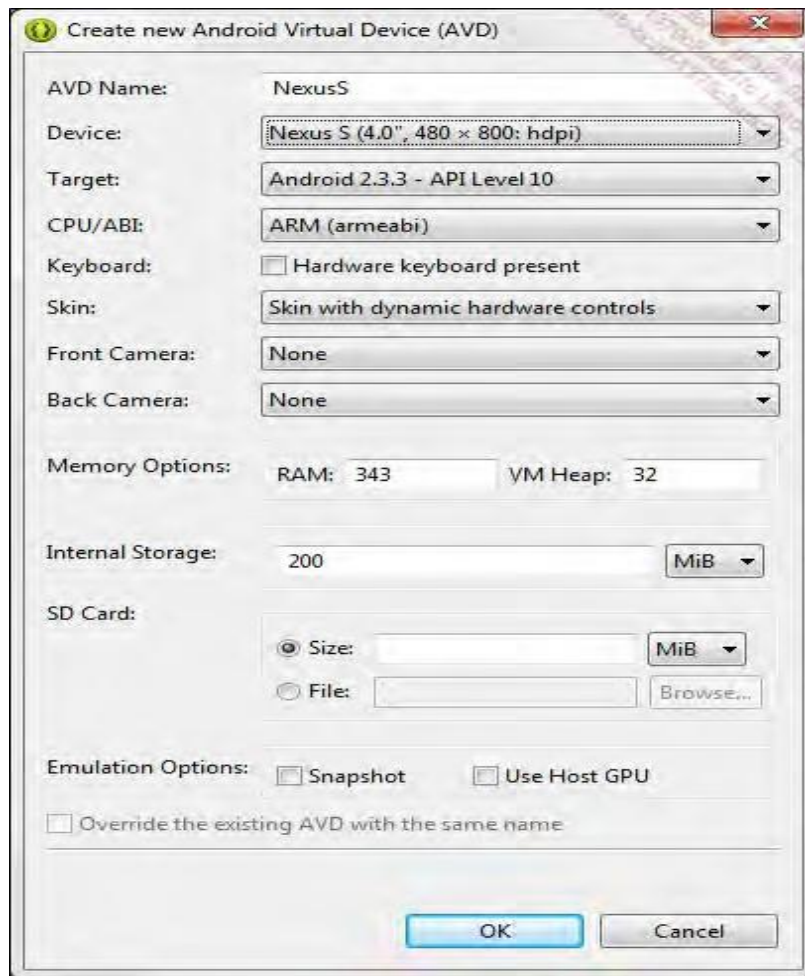
emulará. Aunque, en teoría, el desarrollador debería probar su aplicación en todas las configuraciones posibles, en el marco de una aplicación clásica el hecho de seleccionar ARM se admite, comúnmente, como lo más generalista, abarcando el mayor número de dispositivos Android.

- Keyboard: permite especificar si el dispositivo simulado dispone de un teclado físico o no. La gran mayoría de dispositivos Android no disponen de teclado, de modo que es preferible desmarcar esta opción.
- Skin: seleccionar esta opción indica al emulador que se desea presentar, además de la pantalla del dispositivo, los botones físicos habituales que se encuentran en un terminal Android: botones de volumen, botón Home (inicio), botón de menú y botón back (volver). Es preferible marcar esta opción, ¡pues de lo contrario será preciso utilizar atajos de teclado.
- Front Camera: permite especificar si el dispositivo emulado dispone de una cámara frontal (sobre la cara principal del dispositivo). Puede emular una cámara o bien utilizar la webcam de su equipo. La opción None permite obviar este aspecto de la emulación, en el caso de que la aplicación desarrollada no interactúe con la cámara del dispositivo.
- Back Camera: permite especificar si el dispositivo emulado dispone de una cámara sobre la cara trasera. De manera típica, los dispositivos Android disponen de una cámara sobre la cara trasera con una mejor resolución que la disponible en la cara principal.
- Memory Options: permite indicar la memoria disponible sobre el dispositivo. Los valores por defecto son los recomendados en la mayoría de casos.
- Internal Storage: permite especificar el espacio disponible para la aplicación. Sabiendo que una aplicación Android típica posee un tamaño muy inferior a 50 MB, el valor por defecto propuesto de 200 MB es más que suficiente.

- SD Card: permite emular la presencia de una tarjeta SD en el dispositivo virtual especificando un tamaño. Preste atención de no indicar un tamaño excesivamente grande dado que se creará un archivo del tamaño de esta tarjeta SD en el puesto de desarrollo. También es posible reutilizar una tarjeta SD virtual creada anteriormente indicando el nombre de esta tarjeta SD.
- Snapshot: cuando se selecciona esta opción, el emulador realiza una copia de seguridad de la RAM tras el arranque para, a continuación, restaurarla en el siguiente inicio, acelerando de este modo el arranque de la AVD.
- Use Host GPU: permite especificar que, si una aplicación utiliza comandos OpenGL, utilizará la GPU del ordenador host en lugar de emular una GPU. Esta opción es incompatible con la opción Snapshot.

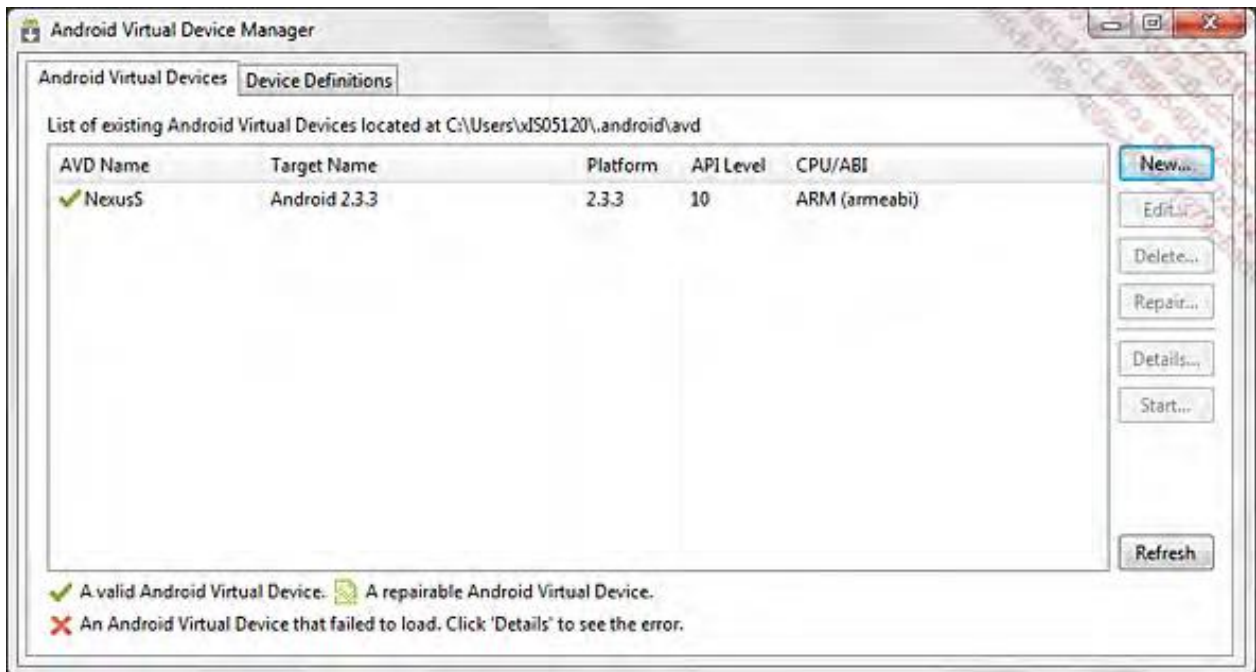
Para nuestro ejemplo, y como primer dispositivo Android, vamos a emular un smartphone Nexus S, que dispone de una pantalla de 480 x 800 en hdpi, Android 2.3.3 y 343 MB de RAM.

Tras la selección del modelo y una vez introducido un nombre para el terminal que se creará, haga clic en el botón OK.



A continuación, el AVD se agrega a la lista de AVD existentes.

Es posible crear tantos AVD como se quiera. Esto permite probar la misma aplicación sobre numerosas configuraciones de hardware y asegurarse así del buen funcionamiento de ésta sobre el mayor número de configuraciones diferentes.



Los botones Delete..., Repair... y Details... permiten respectivamente suprimir, reparar el AVD seleccionado actualmente y mostrar sus características.

El botón Start permite iniciar el emulador con el AVD seleccionado especificando los parámetros de inicio. Es posible modificar el tamaño de la ventana del emulador y borrar los datos de usuario del AVD como si el dispositivo saliera de fábrica. También es posible indicar que se quiere restaurar el AVD tras una copia de seguridad marcando la opción Launch from snapshot. La opción Save to snapshot permite realizar una copia de seguridad del AVD tras cada cierre de la ventana del emulador.

Cierre la ventana.

Vuelva a ejecutar la aplicación tal y como hemos visto anteriormente.

El emulador Android se ejecuta a continuación en el puesto de desarrollo.

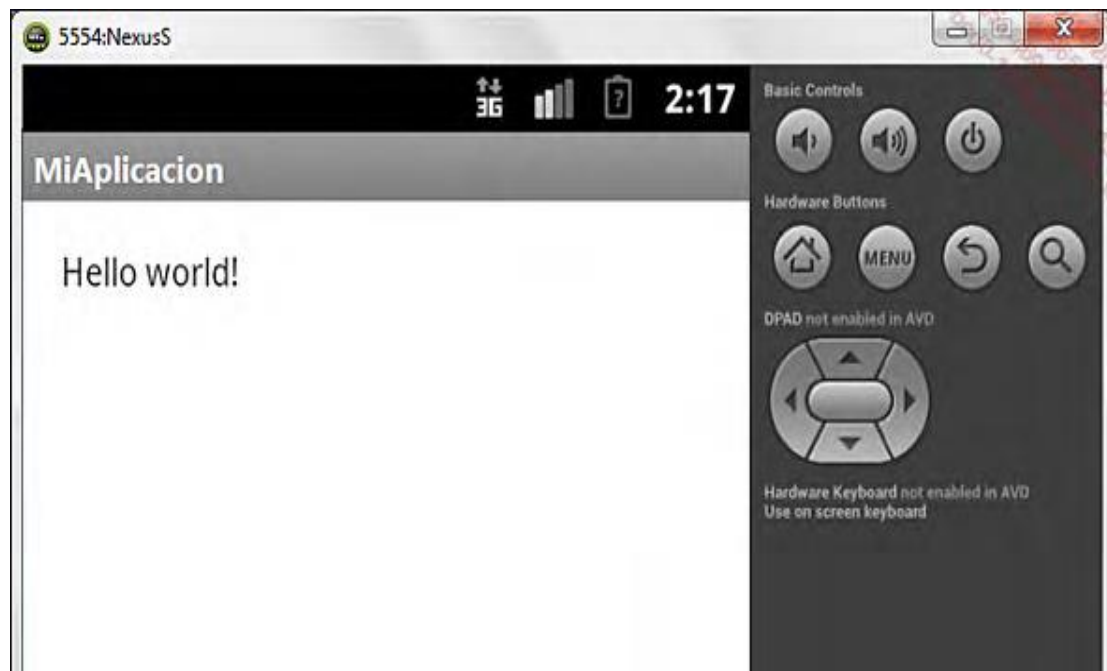
Espere el tiempo de carga del sistema. Esto puede llevar varios minutos según la potencia del puesto de desarrollo. A continuación, aparece la

pantalla de bienvenida o la pantalla de desbloqueo. A continuación, se carga y se ejecuta la aplicación automáticamente.

Si aparece la pantalla de desbloqueo, desbloquee para poder ver la aplicación.

La interfaz del emulador y la interfaz de la aplicación son distintas según la versión concreta de Android utilizada por el proyecto, la versión del AVD sobre la que se ejecuta la aplicación, así como las opciones especificadas durante la creación de la actividad.

Con el fin de tener una primera impresión del trabajo de desarrollo de aplicaciones en Android, las dos siguientes capturas de pantalla muestran el aspecto de la aplicación sobre dos dispositivos (emulados) diferentes: incluso para una aplicación tan simple como un proyecto por defecto, las diferencias son visibles.



He aquí la representación de la aplicación sobre el terminal en que acabamos de crear una emulación y, más abajo, el mismo terminal, pero equipado con la versión 4.4. de Android.



Existen numerosos atajos de teclado que permiten interactuar con el emulador. Por ejemplo, la tecla 7 del teclado numérico del ordenador permite cambiar la orientación de la pantalla en un sentido, y la tecla 9 en el otro sentido. Las distintas opciones de ejecución del emulador están disponibles en la siguiente dirección.

<http://developer.android.com/guide/developing/tools/emulator.html>

b. En un dispositivo Android

Si bien el emulador puede adaptarse a numerosas configuraciones de hardware, también es cierto que hay algunas que no puede emular de manera cómoda. Uno de los mayores defectos del emulador es su rendimiento, más bien mediocre. De este modo, el emulador mostrará rápidamente su debilidad si la aplicación que quiere probar requiere un

dispositivo potente. Además, probando una aplicación a través del emulador ésta puede parecer correcta mientras que una prueba real va a mostrar en seguida los defectos de la aplicación.

Un ejemplo muy sencillo es la presión táctil sobre la pantalla. En el emulador,

basta con hacer clic con el cursor del ratón, que es muy preciso a nivel de píxel. En un dispositivo real, el dedo del usuario es menos preciso que el cursor del ratón, hasta el punto de que, si la aplicación requiere hacer clic sobre una zona pequeña de la pantalla, será mucho más delicado si se hace con los dedos, mientras que no supondrá problema alguno utilizando el ratón en un emulador.

Otro aspecto que genera problemas es la rotación de la pantalla: si bien en un emulador el cambio en la orientación se controla completamente, sobre un dispositivo físico la rotación de la pantalla está vinculada directamente a la manipulación del usuario. El desarrollador, pensé a todas las pruebas que pueda hacer, no prevé la respuesta tras una rotación de pantalla que sobreviene en un momento imprevisto. El uso de un dispositivo real permite ajustar cierto realismo a las pruebas, ¡lo cual resulta saludable y ayuda a mejorar la calidad de la aplicación publicada!

Es por ello que es casi obligatorio ejecutar la aplicación en un dispositivo Android real para validar su buen funcionamiento, su reactividad y, sobretodo, su ergonomía. No obstante, dado que esto sólo validará la aplicación en un único tipo de hardware, lo ideal es combinar pruebas en un emulador y pruebas sobre dispositivos reales.

Para probar una aplicación sobre un dispositivo Android, es preciso activar la depuración USB sobre el dispositivo. Para ello, la primera etapa de la manipulación es distinta según el sistema Android utilizado, o según el fabricante - la mayoría de fabricantes y/o distribuidores tienen la (incómoda)

costumbre de agregar una capa propietaria por encima de Android.

En sistemas Android de versiones inferiores a la 3.0 (API 11), vaya a la pantalla de bienvenida si no se muestra directamente y presione a continuación la tecla Menú. En sistemas Android de versiones 3.0 (API 11) o superiores, haga clic sobre la hora en la parte inferior derecha y a continuación haga clic en la pequeña ventana que contiene la hora que acaba de aparecer.

Seleccione Configuración.

A continuación, se muestra la aplicación de configuración de los parámetros del dispositivo.

Seleccione Aplicaciones, a continuación, Desarrollo.

Active la opción Depuración USB y valide el mensaje de advertencia haciendo clic en el botón OK.

A continuación, hay que hacer que el puesto de desarrollo reconozca al dispositivo Android cuando se conecte mediante un cable USB. Siga las instrucciones que aparecen a continuación según el sistema operativo instalado sobre el puesto de desarrollo.

Si apareciesen problemas durante la instalación de los controladores o si el dispositivo no se detectase o reconociese, verifique que el cable USB que está utilizando es el entregado originalmente con el dispositivo Android y que está conectado directamente al puesto de desarrollo.

Windows 7 o superior

Es preciso instalar un controlador USB sobre el puesto de desarrollo para reconocer dispositivos Android.

- Descárguelo mediante la herramienta Android SDK and AVD Manager vista anteriormente. El paquete se llama Google USB Driver package y se encuentra en Third party Add-ons y a continuación Google Inc.
- Conecte el dispositivo Android al puesto de desarrollo utilizando un cable USB. Windows detecta automáticamente el dispositivo e intenta instalarlo.
- Abra el menú Inicio y seleccione Panel de control.
- Seleccione Hardware y sonido y a continuación administrador de dispositivos (bajo Dispositivos e impresoras).
- Haga clic con el botón derecho sobre la línea Periférico USB compuesto que muestra un pequeño icono de alerta en la lista y seleccione a continuación Actualizar software del controlador.
- Seleccione Buscar software del controlador en el equipo.
- Haga clic en el botón Examinar.
- Seleccione la carpeta `sdk\extras\google\usb_driver\` y haga clic en Siguiente.
- Por último, haga clic en el botón Instalar.
- Se efectuará la instalación del controlador.