

**Mi Primera Aplicación En Android**  
**MAEOCS**  
**Un Ejemplo Nada Más**

Carlos I. Gaitán M., Edward H. Jiménez M.

9 de mayo de 2013



# Índice general

<b>I</b>	<b>El mundo móvil y Android</b>	<b>1</b>
<b>1.</b>	<b>¿Por qué una aplicación móvil?</b>	<b>3</b>
1.1.	El mundo móvil . . . . .	4
1.2.	La historia del mundo móvil . . . . .	6
1.3.	Los grandes de hoy en día . . . . .	7
1.4.	El mercado móvil . . . . .	10
<b>2.</b>	<b>¿Por Qué Una Aplicación en Android?</b>	<b>11</b>
2.1.	La historia de Android . . . . .	12
2.2.	Ventajas y desventajas de Android . . . . .	12
2.3.	Android en el mundo . . . . .	12
2.4.	Android en el mercado móvil . . . . .	12
<b>3.</b>	<b>Consejos útiles</b>	<b>13</b>
3.1.	Proyectos móviles . . . . .	13
3.2.	Wireframes . . . . .	13

<b>II</b>	<b>Conociendo Android</b>	<b>15</b>
<b>4.</b>	<b>Preparándolo todo</b>	<b>17</b>
4.1.	El SDK de Android . . . . .	18
<b>5.</b>	<b>Mi primera actividad</b>	<b>23</b>
5.1.	Creando un proyecto Android . . . . .	24
5.2.	Hola mundo y el simulador de Android . . . . .	29
5.3.	Botones, textos y otros . . . . .	31
<b>6.</b>	<b>Varias actividades una sola aplicación</b>	<b>33</b>
6.1.	Intent y la secuencia de actividades . . . . .	33
6.2.	Layout orden en el caos . . . . .	33
6.3.	Vistas y más vistas . . . . .	33
6.4.	Fechas y contactos . . . . .	34
6.5.	Tabs, menús y preferencias . . . . .	34
<b>7.</b>	<b>Código avanzado</b>	<b>35</b>
7.1.	SQLite y archivos planos . . . . .	35
7.2.	Cambios de orientación . . . . .	35
7.3.	Estilos e imagen . . . . .	35
7.4.	Permisos y conexiones . . . . .	35
<b>8.</b>	<b>Sensores</b>	<b>37</b>
8.1.	Rotación y brújulas . . . . .	37
8.2.	GPS . . . . .	37

<i>ÍNDICE GENERAL</i>	v
-----------------------	---

8.3. Archivos . . . . .	37
-------------------------	----

<b>III Interacción con Android</b>	<b>39</b>
------------------------------------	-----------

<b>9. Java y Android</b>	<b>41</b>
--------------------------	-----------

9.1. Aplicaciones Java de Escritorio . . . . .	41
--	----

9.2. Tipos de datos compartidos . . . . .	42
---	----

9.3. Exportación y conexión . . . . .	43
---------------------------------------	----



## Parte I

# El mundo móvil y Android





## Capítulo 1

# ¿Por qué una aplicación móvil?

## 1.1. El mundo móvil

Hoy en día es muy común escuchar acerca de los dispositivos móviles y las aplicaciones que existen para estos, pero cuantas veces nos hemos detenido a pensar por un momento como en realidad funciona este mundo (que no es tan nuevo como muchos pensarían) y que tan fácil es entrar en su mercado, creando aplicaciones móviles o juegos para venderlos en las famosas App Store. A lo largo de este capítulo expondremos los beneficios de entrar al mundo móvil y sus ventajas competitivas frente al mercado que viene, para ello empezaremos con enumerar algunas de sus ventajas:

- La cifra de usuarios móviles en el mundo esta al rededor del 85 % de la población mundial.
- Esta constantemente conectado.
- Si juntamos el tiempo de juego de todos los usuarios de Angry Birds se suman 205,343 años.
- Siempre esta cerca de la persona.
- Tiene canales de pago exclusivos y seguros (App Store).
- Contiene la información personal de acceso sencillo.
- Captura el concepto social del consumismo.

Los dispositivos móviles siempre han sido un instrumento que representa portabilidad, eficacia, conectividad, privacidad y agilidad para las personas que hacen uso de sus múltiples capacidades creando así un publico bastante grande que hace uso de estos dispositivos ya sea para jugar, trabajar o simplemente comunicarse. Sabia usted que hoy en día existen más líneas telefónicas móviles que fijas y las personas que poseen un dispositivo móvil pasan más del 80 % del tiempo con este al lado, sabia que en Colombia el número de líneas móviles 1.2 es más de seis veces el número de líneas fijas 1.1; esto, sin duda, nos hace pensar acerca del increíble cubrimiento que puede llegar a tener una aplicación como lo puede ser un gestor de correo, un gestor de actividades, una agenda virtual, etc.

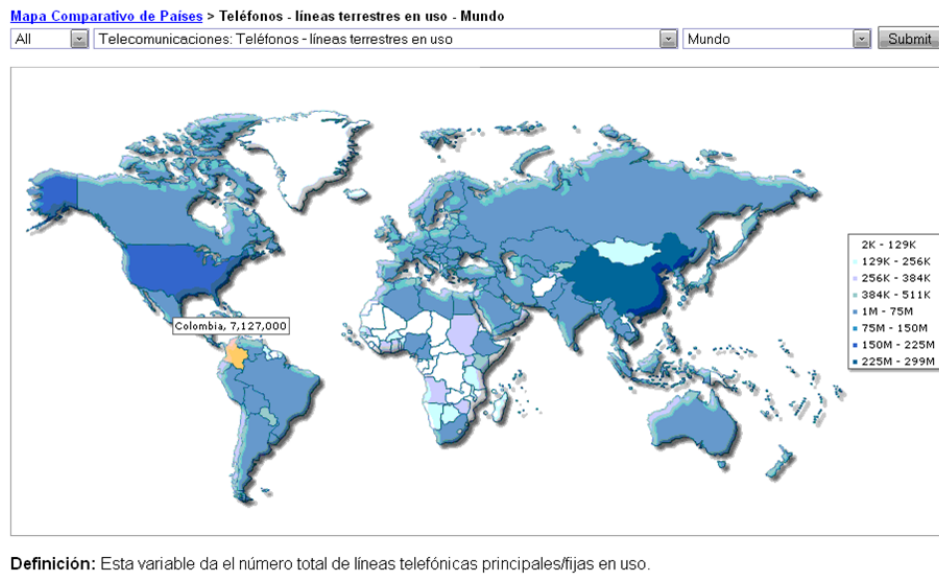


Figura 1.1: Estadísticas líneas fijas en Colombia

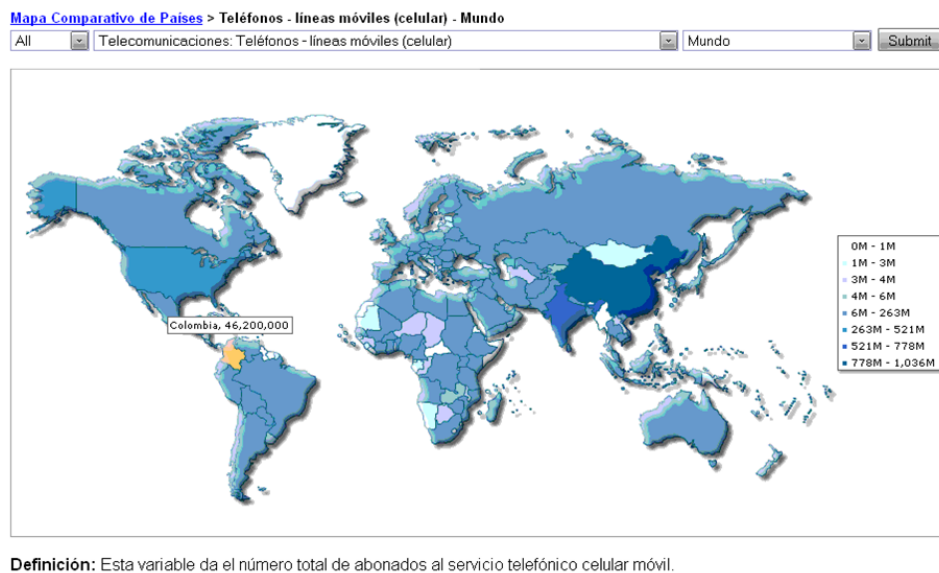


Figura 1.2: Estadísticas líneas móviles en Colombia

## 1.2. La historia del mundo móvil

Esta sección queremos iniciarla con la frase *"para saber a donde vas tienes que saber de donde vienes"*, en el mundo móvil si que es importante conocer de donde viene no solo la tecnología sino también las aplicaciones que queremos llegar a diseñar por eso en este capítulo profundizaremos un poco de lo que fue e incluso aun sigue siendo.

Muchas personas suelen pensar que la movilidad comienza y termina con los celulares pero la verdad es que detrás de los celulares viene todo un mundo de dispositivos como los asistentes digitales (PDA), el beeper o las agendas electrónicas que fueron en realidad los que fusionados con los celulares dieron nacimiento a los smartphone.



Figura 1.3: Cuadro de dispositivos móviles

Hagamos una pequeña reseña histórica de los dispositivos móviles; como todo avance tecnológico comienza con una necesidad o en este caso dos: la capacidad de permanecer conectado y poder organizar grandes cantidades de información a cualquier hora del día en el lugar donde que me encuentre (la casa, el trabajo, la calle, el bus, etc.), normalmente este tipo de necesidades las imponían compañías e importantes empresarios a las empresas de telecomunicaciones las cuales respondieron no a los empresarios sino a las fuerzas armadas de la segunda guerra mundial con el Handie Talkie, años después comenzó la creación de los primeros teléfonos móviles para el público con pesos aproximados a un kilogramo y con un precio promedio de cuatro mil dolares, con el tiempo hubo respuestas más baratas para los empresarios pequeños como lo es el beeper y mucho después atendiendo a la segunda necesidad nacen las agendas electrónicas las cuales serían rápidamente remplazadas por los PDA que al final terminarían fusionándose con los teléfonos móviles para suplir las dos necesidades como una sola.

Lo más importante para destacar no es el avance ni los cambios tecnológicos que ha tenido el mundo móvil, por el contrario lo que vale la pena evaluar es lo genérico y convencional que se han mantenido sus aplicaciones básicas como lo son el calendario, la agenda de contactos, la calculadora, la agenda

de tareas y claro no pueden faltar los juegos, si hay algún cambio que atribuir a lo mencionado anteriormente es sobre la usabilidad y calidad de imagen lo cual quiere decir de una u otra forma que si se crean las mismas aplicaciones más amigables y con mejor calidad de imagen sin duda alguna remplazarán a las que vienen pre-instaladas en el dispositivo.

### 1.3. Los grandes de hoy en día

Vale la pena aclarar desde un inicio que con grandes no nos referimos ni al tamaño de la compañía ni a la tecnología en los dispositivos o sistemas operativos, con "grandes" hacemos referencia a los más usados en el mercado hoy en día y para ello nos apoyamos en dos gráficas: mercado de los sistemas operativos en el mundo 1.5 y en Colombia 1.4.

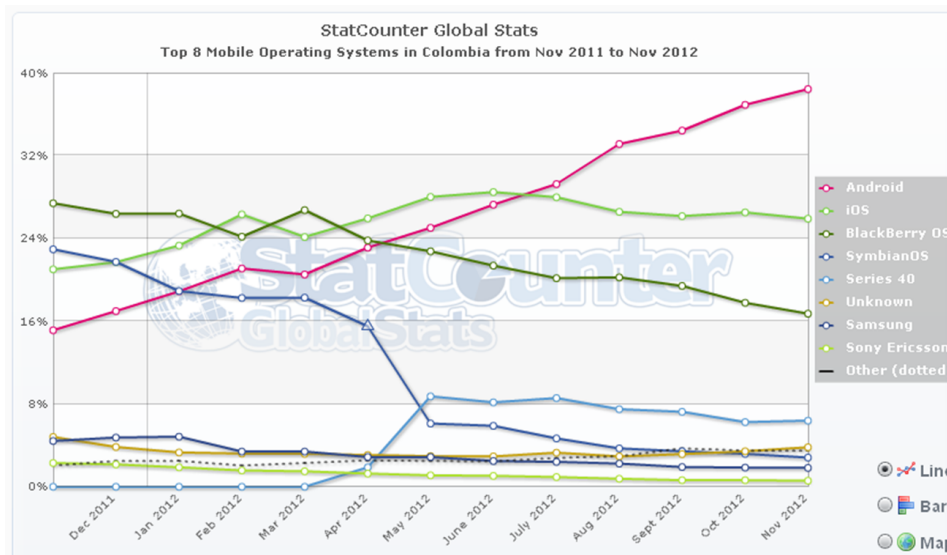


Figura 1.4: Estadísticas sistemas operativos móviles en Colombia

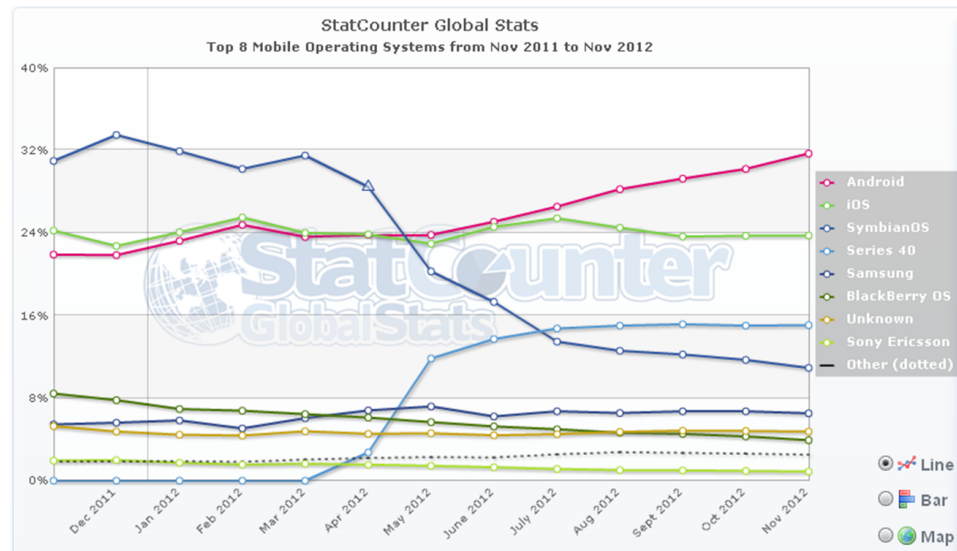


Figura 1.5: Estadísticas sistemas operativos móviles en el mundo

Como se puede apreciar en las gráficas vale la pena hablar de cuatro grandes:

- **Symbian:** Es un sistema operativo móvil que si bien hoy en día ya no esta en el mercado hasta hace poco fue uno de los más usados por los smartphone, fue creado en convenio por muchas compañías como Nokia, Sony Mobile Communications, Psion, Samsung, Siemens, Arima, Benq, Fujitsu, Lenovo, LG, Motorola, Mitsubishi Electric, Panasonic, Sharp, etc. Durante todo su ciclo fue siendo comprado lentamente por Nokia hasta que este se apodero totalmente de el, con el tiempo y siendo Nokia una de las empresas más imponentes en celulares Symbian gano un gran publico el cual lentamente fue desapareciendo en el tiempo, todo el mundo sabe que Symbian marco el inicio de los sistemas operativos para smartphone pero en el 2012 lo único que marco fue el final de su era, muchos atribuyen su desaparición a la falta de avances y otros más a su falta de publicidad como sistema operativo, una ultima teoría bastante valida se atribuye a la falta de aplicaciones móviles que pudieran competir con las de su competencia.
- **Blackberry:** BlackBerry OS es un sistema operativo móvil desarrollado por Research In Motion para sus dispositivos BlackBerry. El sistema

operativo de Blackberry fue creado para uso empresarial y así fue que se hizo famoso sin embargo cuando salio al mercado para el publico en general fue bien recibido por sus dos grandes ventajas: el servicio de mensajería y su impresionante seguridad la cual traía del mercado empresarial del cual había aprendido tanto, actualmente Blackberry como sistema operativo y como teléfono móvil esta pasando por una crisis pues se ha visto arrinconado frente a su competencia por razones muy similares a las de Symbian pero la razón más grande que lo ha venido dejando a bajo en comparación con Android y IOS es la constante queja de muchos usuarios frente a la usabilidad y lo rígido del sistema.

- Android: Es un sistema operativo móvil basado en Linux, que junto con un conjunto de aplicaciones da una oferta libre para el mundo móvil. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. En sus inicios Android fue una revolución para aquellos que siempre han buscado el software libre y gratuito siendo así el sistema operativo con más aplicaciones gratis en su App Store, a diferencia de muchos sistemas operativos móviles no esta ligado a una marca de teléfono en específico sin embargo la tabla de dispositivos que soporta no es tan grande como muchos desearíamos, es un sistema operativo que a cada día tiene más usuarios y esto se debe a su interfaz fácilmente adaptable a los gustos de los usuarios y también a su gran cantidad de aplicaciones no solo para su App Store sino también en diferentes paginas o sitios web.
- IOS: Es un sistema operativo móvil de Apple. Originalmente desarrollado para el iPhone, siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV. Una gran desventaja y fuerte de IOS que le ha costado muchos usuarios es lo cerrado de su sistema y el poco entusiasmo o facilidades que da a los programadores para desarrollar para su plataforma; provocando así que solo se pueda usar en dispositivos Apple con aplicaciones en su mayoría desarrolladas por empresas colocando a IOS como la plataforma con las aplicaciones más costosos en el mercado móvil.

## 1.4. El mercado móvil

Aunque no profundizaremos mucho en esta sección si nos parece muy importante que haya una clara orientación de como es el mercado al que vamos a entrar y como pueden vender sus aplicaciones.

- **Aplicativos con costo:** Esta opción hace referencia a cobrar directamente la aplicación en la App Store, en este caso hay que tener mucho cuidado pues el precio que pongamos puede hacer a nuestra aplicación un gran éxito o un gran fracaso para ello se debe considerar tres cosas:
  - **Costo promedio de la tienda:** Costo al que están la mayoría de las aplicaciones del mismo estilo o similares a la que se intenta vender.
  - **Costo/periodo de la suscripción a la App Store:** Toda tienda cobra un periodo de suscripción para poder subir las aplicaciones a la tienda, dicho periodo varia entre 3, 6 y 12 meses.
  - **Porcentaje de ganancia real sobre la aplicación:** Adicional al costo de suscripción la tienda también impone un porcentaje de ganancia sobre nuestras aplicaciones.
- **Aplicativos sin costo:** Esta opción hace referencia a dejar gratis nuestra aplicación en la App Store, en este caso hay dos opciones si nuestro interés es hacer algo de dinero:
  - **Publicidad:** Esta opción consiste en activar opciones publicitarias para nuestra aplicación permitiendo a la tienda insertar contenidos adicionales los cuales nos pagaran dependiendo de la cantidad de descargas y usos de la aplicación, para usar esta opción debemos tener claro que la usaremos desde el inicio de la creación de nuestra aplicación.
  - **Adicionales:** Una estrategia muy común en las aplicaciones móviles es liberar el core o aplicación base completamente gratis y cobrar adicionales, un ejemplo claro son los juegos donde un arma mejor o personalizar a tu personaje puede costarte unos cuantos centavos. Para esta opción las tiendas disponen de sistemas especiales que debes integrar a tu aplicación desde que comienzas a desarrollarla.



## Capítulo 2

# ¿Por Qué Una Aplicación en Android?

## **2.1. La historia de Android**

blablablabla

## **2.2. Ventajas y desventajas de Android**

blablablabla

## **2.3. Android en el mundo**

blablablabla

## **2.4. Android en el mercado móvil**

blablablabla

## Capítulo 3

# Consejos útiles

blablablabla

### 3.1. Proyectos móviles

blablablabla

### 3.2. Wireframes

blablablabla



## Parte II

# Conociendo Android



## Capítulo 4

# Preparándolo todo

## 4.1. El SDK de Android

Es importante aclarar que desde este punto el libro tomara una orientación de guía, ayudando a aclarar desde como instalar todo lo que necesitamos para instalar el ambiente para desarrollar hasta como crear cierto tipo de aplicaciones, para que la guía sea lo más útil debe prestar especial atención a las áreas resaltadas o sombreadas de las imágenes.

Conseguir e instalar el SDK:

1. Lo primero es ir a la pagina de Android y descargar todo lo que necesitamos (<http://developer.android.com/sdk/index.html>).

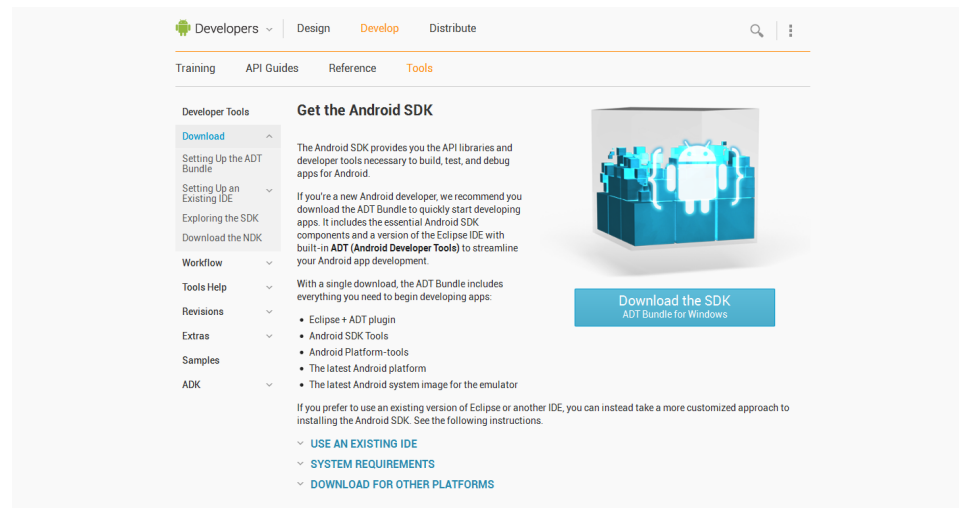


Figura 4.1: Pagina principal del SDK Android 2012



2. Lea detenidamente los términos de la licencia y a continuación seleccione aceptar los términos y la casilla siguiente de acuerdo a la arquitectura de su sistema operativo, al dar clic en el botón de descargas comenzara a descargarse un paquete comprimido que contiene el SDK y un eclipse listo para empezar a desarrollar.

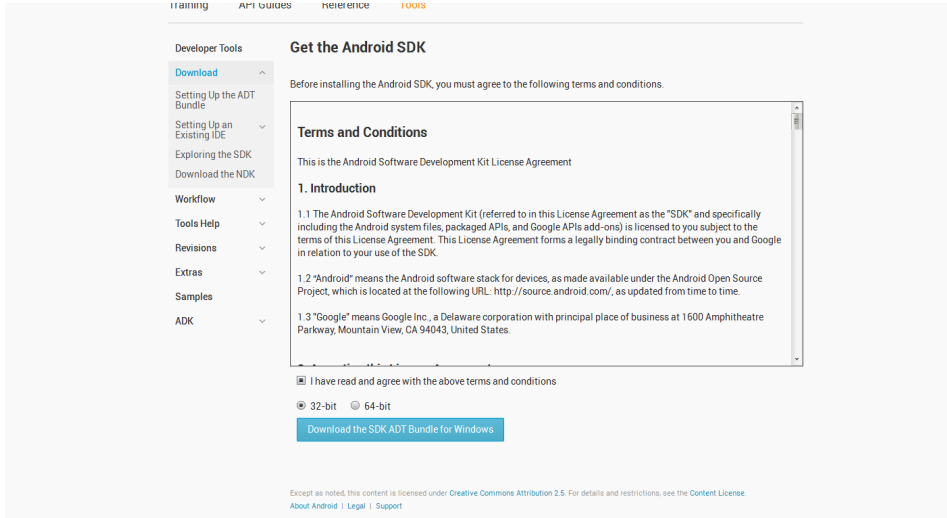
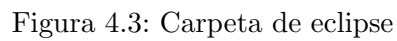


Figura 4.2: Términos de contrato

- 
- The screenshot shows the Eclipse IDE workspace with a project named 'eclipse'. The project structure includes the following folders and files:
- configuration** (Folder)
  - droptins** (Folder)
  - features** (Folder)
  - p2** (Folder)
  - plugins** (Folder)
  - readme** (Folder)
  - Workspace** (Folder)
  - eclipseproduct** (Eclipse Product File, 1 KB)
  - artifacts** (XML Document, 72 KB)
  - notice** (Plain Text HTML Document, 9 KB)
  - Configuration Settings** (1 KB)
  - Configuration** (Folder)
  - ep-v10** (PreTeXt HTML Document, 17 KB)
  - eclipseec** (Folder)
  - eclipse** (Eclipse IDE icon)



4. En la barra de arriba aparece un icono de Android con una flecha hacia abajo, este icono simboliza el SDK Manager el cual sera nuestro gestor de paquetes y versiones de Android, a continuación dar clic en el SDK Manager.

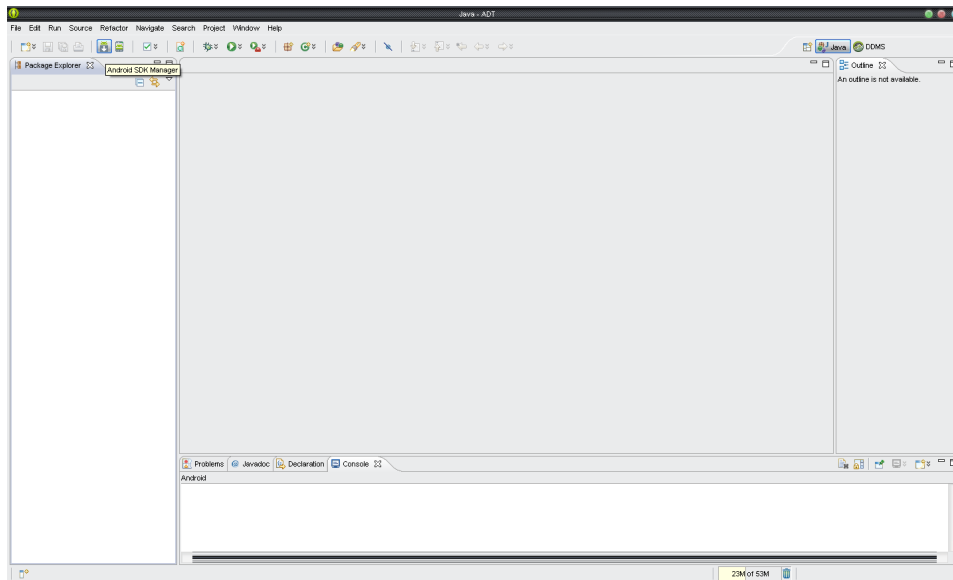


Figura 4.4: Entorno de eclipse para Android

5. Veremos una lista de paquetes separados por las versiones existentes de Android, en esta parte trabajaremos con la versión 2.3.3 por lo tanto sugerimos seleccionar los mismos paquetes que aparecen en la imagen, al dar clic en instalar debemos estar consientes de que estos paquetes pueden llegar a ser bastante pesados y que dependiendo de nuestra conexión a Internet y otros factores la instalación puede tomar horas.

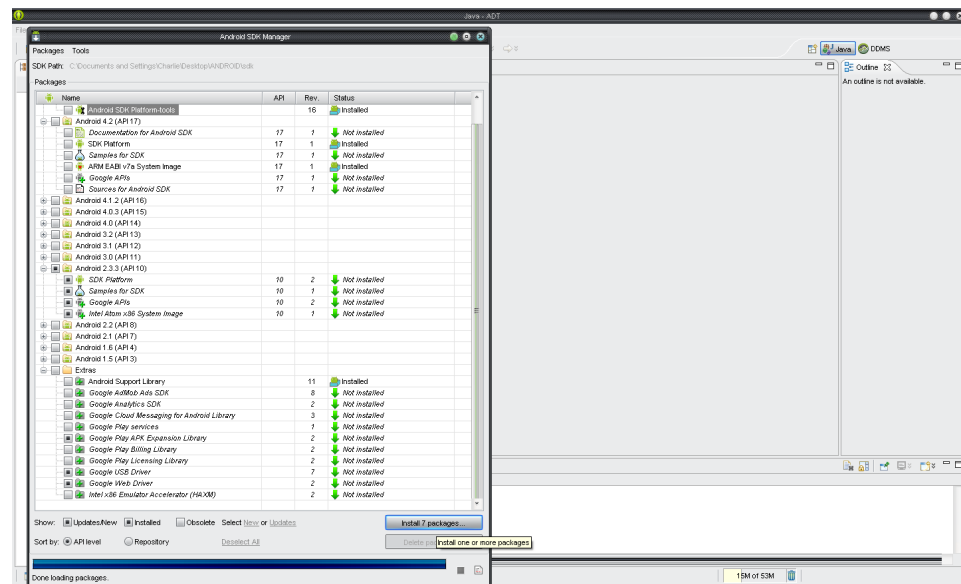


Figura 4.5: SDK Manager de Android

## Capítulo 5

### Mi primera actividad

## 5.1. Creando un proyecto Android

A partir de aquí evitaremos las imágenes de pantalla en lo posible pues más que una guía visual del entorno de eclipse para Android lo que buscamos es una guía conceptual del código y el desarrollo en Android.

1. Crear un proyecto Android es muy similar a como se hace para Java en eclipse, basta con hacer clic secundario en el área de exploración de proyectos y seleccionar la opción de un nuevo proyecto Android.

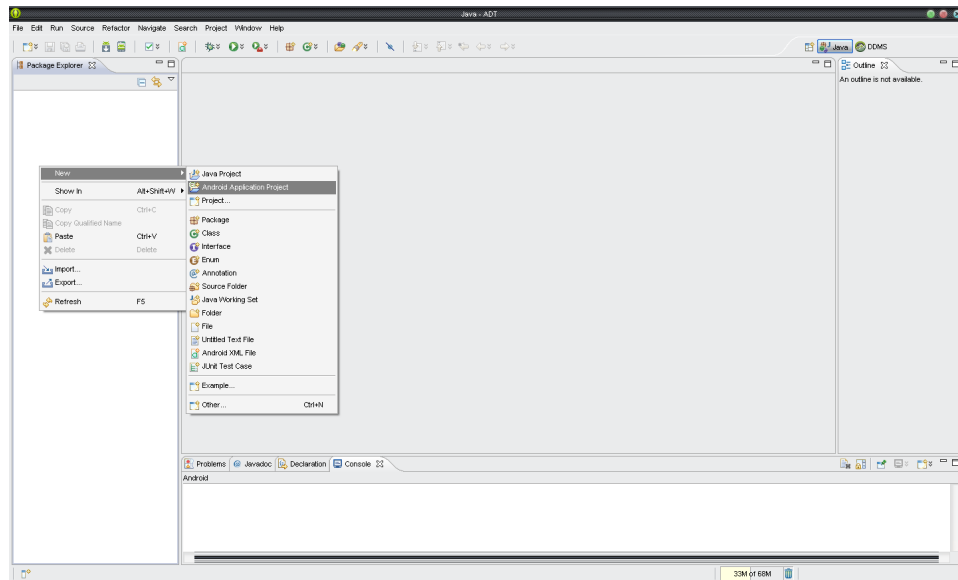


Figura 5.1: Creando un nuevo proyecto Android

2. Para el nuevo proyecto tenemos varias opciones la primera hace referencia al nombre que queremos aparezca en el dispositivo en la segunda el nombre del proyecto en eclipse, la tercera es para los paquetes sobre los que se desarrollara el proyecto y las ultimas a las versiones sobre las cuales se ejecutara el proyecto.

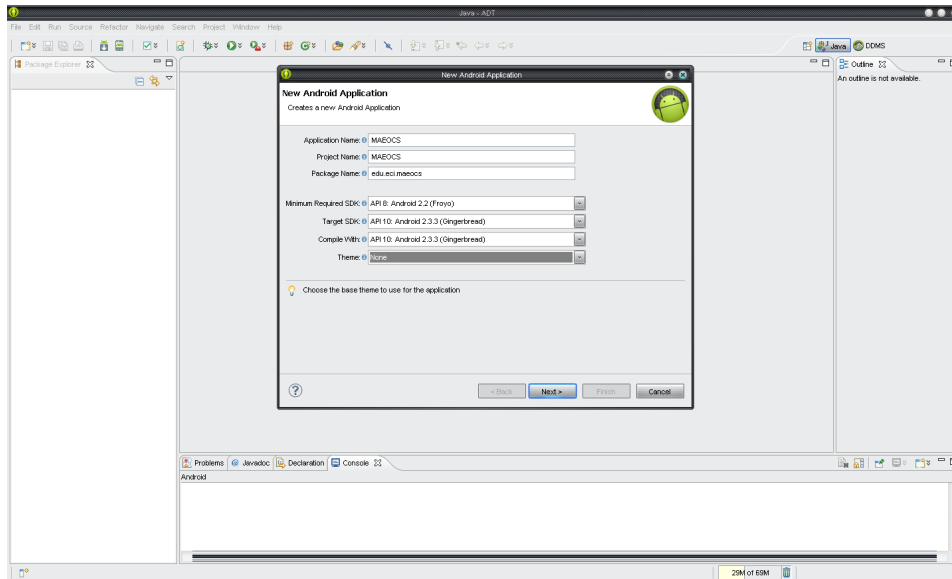


Figura 5.2: Ventana de nuevo proyecto

3. La siguiente parte es para seleccionar el icono que aparecerá en nuestra aplicación, Android nos provee con algunas opciones estándar, pero si diseñamos nuestros propios iconos podemos seleccionarlos en la pestaña de imágenes, en este ultimo caso debemos tener cuidado con los tamaños de la imagen para obtener la mejor calidad posible dependiendo del tamaño del dispositivo que vaya a usar nuestra aplicación.

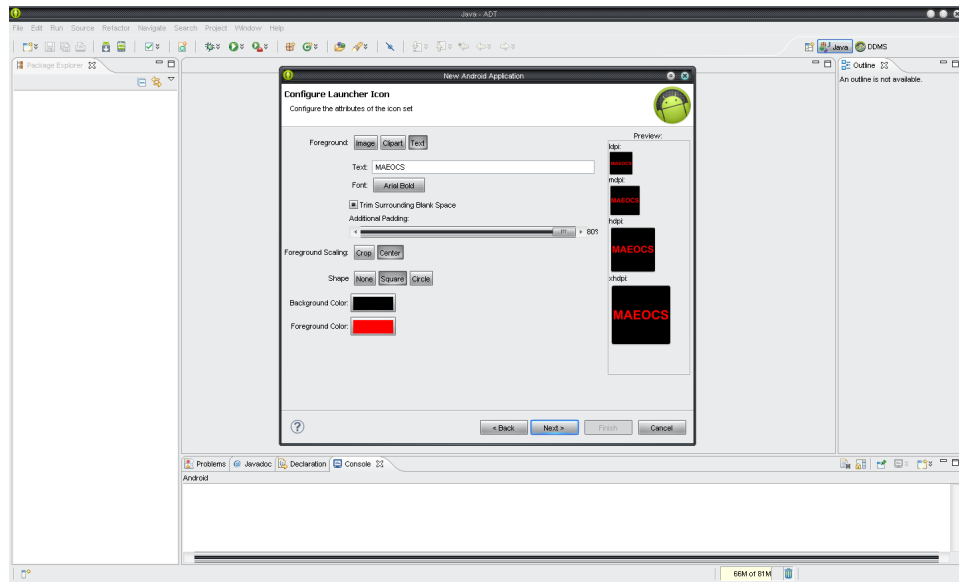


Figura 5.3: Seleccionando el icono de la aplicación



4. La siguiente ventana corresponde a la selección del tipo de actividad que vamos a crear, como trabajaremos en la versión 2.3.3 el tipo de actividad solo puede ser blanca ya que las otras opciones corresponden a versiones superiores.

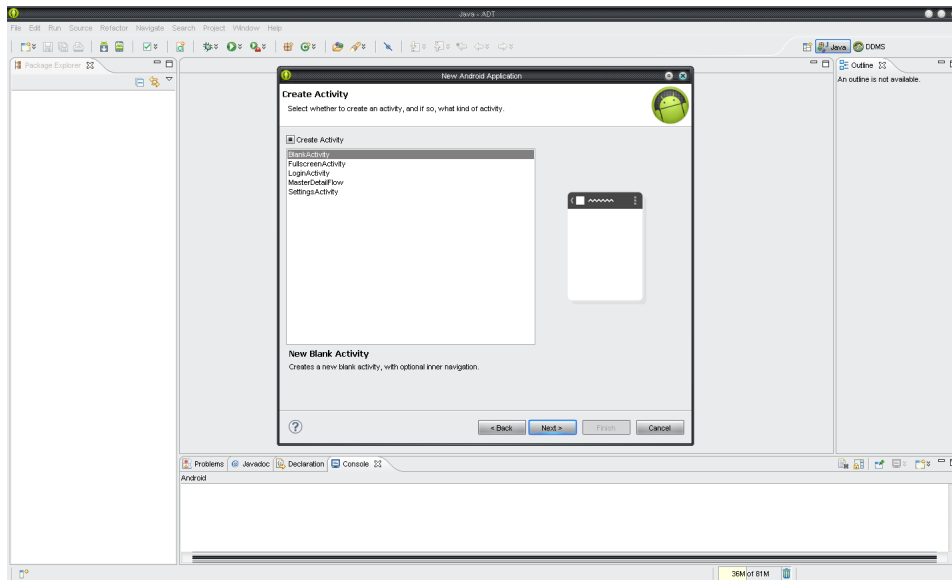


Figura 5.4: Términos de contrato

5. Por ultimo debemos nombrar la actividad y la interfaz principales que acabamos de crear, si deseamos que la actividad principal sea otra clase diferente debemos especificarlo más adelante en el manifiesto.

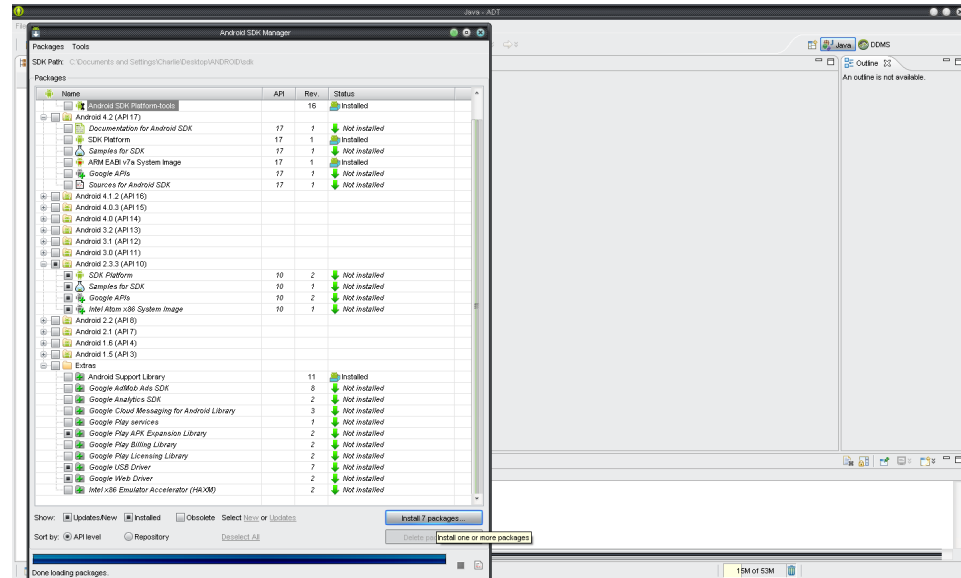


Figura 5.5: Términos de contrato

## 5.2. Hola mundo y el simulador de Android

Primero que nada un fragmento del código para el hola mundo, de ahora en adelante cada vez que hablemos de una actividad mostraremos dos partes del código el .xml correspondiente a la interfaz y el código Java correspondiente a la actividad, hay que tener especial cuidado cuando se programa para Android pues los nombres de los objetos de la interfaz como lo son los campos de texto y los botones deben ser muy dicentes pues estos se pueden llegar a confundir o refundir entre más grande sea la aplicación, la explicación es la siguiente:

La parte XML:

- **RelativeLayout**: Clase para representar un cuadro de contenido cuyas márgenes y elementos se definen por como se posicionan unos al lado de otros.
- **TextView**: Clase que sirve para representar un campo de texto, es el equivalente a Label en Java.
  - **id**: Identificador del cuadro de texto.
  - **layout width**: Ancho del cuadro de texto, algunas opciones son:
    - **match parent**: Forma automática de acomodar el tamaño usando como referencia el contenedor.
    - **wrap content**: Forma automática de acomodar el tamaño usando como referencia el contenido cercano al objeto.
    - **fill parent**: Forma automática de acomodar el tamaño llenando por completo el contenedor.
  - **layout height**: Alto del cuadro de texto.
  - **centerHorizontal**: Aclarar si se centra respecto al contenedor de forma horizontal.
  - **centerVertical**: Aclarar si se centra respecto al contenedor de forma vertical.
  - **text**: texto a mostrar, normalmente todas las cadenas se almacenan en @string el cual se encuentra configurado en res/values/strings.xml.

---

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/labelDeTexto"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="@string/hello_world" />

</RelativeLayout>
```

---

La parte JAVA:

- Bundle: Clase que sirve para transmitir tipos primarios de datos a través de las actividades.
- Activity: Clase que representa una actividad de Android, se puede interpretar como cada pantalla o servicio que se provee en la aplicación.
- Menu: Toda actividad trae un menú al cual se le puede agregar opciones modificando el xml o directamente en la clase.
- TextView: Clase que sirve para representar un campo de texto, es el equivalente a Label en Java.
- setContentView(): Método que define la interfaz que sera usada por esta actividad.
- findViewById(): Método que encuentra una vista u objeto como el cuadro de texto que definimos en el xml, R hace referencia a un codigo autogenerado donde se encuentran todos los recursos.

---

```
//HolaMundo.java
package com.example.ejemplo1;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.TextView;

public class MainActivity extends Activity {

    TextView labelDeTexto;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        labelDeTexto = (TextView) findViewById(R.id.labelDeTexto);
        labelDeTexto.setText("Hola Mundo");
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

---

### 5.3. Botones, textos y otros

blablablabla



## Capítulo 6

# Varias actividades una sola aplicación

blablabla

### 6.1. Intent y la secuencia de actividades

blablabla

### 6.2. Layout orden en el caos

blablabla

### 6.3. Vistas y más vistas

blablabla

## **6.4. Fechas y contactos**

blablabla

## **6.5. Tabs, menús y preferencias**

blablabla



## Capítulo 7

# Código avanzado

blablablabla

### 7.1. SQLite y archivos planos

blablablabla

### 7.2. Cambios de orientación

blablablabla

### 7.3. Estilos e imagen

blablablabla

### 7.4. Permisos y conexiones

blablablabla



## Capítulo 8

# Sensores

blablabla

### 8.1. Rotación y brújulas

blablabla

### 8.2. GPS

blablabla

### 8.3. Archivos

blablabla



## Parte III

# Interacción con Android



## Capítulo 9

# Java y Android

blablablabla

### 9.1. Aplicaciones Java de Escritorio

Java es un lenguaje orientado a objetos para el cual existen muchas librerías, entre las cuales se encuentran objetos de estructuras de datos controladores, etc. A la hora de hacer interfaces gráficas existen dos librerías, *AWT* y *SWING*, las cuales facilitan el desarrollo, pues cuentan con sets de paneles y objetos usados para implementar una interfaz interactiva.

Las librerías más usadas para el desarrollo de esta aplicación fueron:

---

```
import java.awt.Color;
import java.awt.Dimension;
import java.awt.GridLayout;
import java.awt.Image;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;
import java.util.Hashtable;
```

```
import java.util.Map.Entry;

import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JPanel;
```

---

## 9.2. Tipos de datos compartidos

A la hora de compartir los datos, por las similitudes de language entre android y java, se utilizó una clase intermedia con tipos de datos que pudieran ser utilizado por ambas aplicaciones.

---

```
public class Exportable implements Serializable{

    private static final long serialVersionUID =
        -3213930124871683109L;
    private byte[] image;
    private Hashtable<String, String> localsDictionary;
    private RoadsDirectory roads;
    private int width;
    private int height;
    private int xCells;
    private int yCells;

    public Exportable(byte[] image,Hashtable<String, String>
        localsDictionary,
        RoadsDirectory roads, int width, int height,
        int xCells, int yCells ){

        this.image = image;
        this.localsDictionary = localsDictionary;
        this.roads = roads;
        this.width = width;
        this.height = height;
        this.xCells = xCells;
        this.yCells = yCells;
    }
}
```

---



A la hora de trabajar con imagenes, android no cuenta con la libreria de *AWT* por lo que debe trabajar con un dato más primitivo, es allí donde se usa un arreglo de bytes para guardar una imagen, para luego ser transformada a un objeto que utilice Android.

### 9.3. Exportación y conexión

blablabla

