

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



PROYECTO FIN DE CARRERA

**APLICACIÓN DE PROBLEMAS RESUELTOS DE CIRCUITOS
DIGITALES COMBINACIONALES BAJO ANDROID**

Pedro Madrigal Marina

Febrero 2014

APLICACIÓN DE PROBLEMAS RESUELTOS DE CIRCUITOS DIGITALES COMBINACIONALES BAJO ANDROID

**AUTOR: Pedro Madrigal Marina
TUTOR: Federico García Salzmann
PONENTE: Eduardo Boemo Scalvinoni**

**Digital System Laboratory
Dpto. Tecnología Electrónica y de Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Febrero 2014**

Resumen

Este proyecto consiste en la realización de una aplicación para dispositivos Android. En ella se desarrollará una guía de problemas y un tutorial sobre circuitos digitales combinacionales. Los problemas han sido elegidos de la Guía de la asignatura Circuitos Electrónicos Digitales de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid (EPS UAM).

El principal objetivo de la aplicación es ofrecer a los alumnos una guía de ejercicios de síntesis de circuitos, adaptados para ser resueltos en una pantalla de tamaño normal de un teléfono móvil. Además, permite corregir los ejercicios automáticamente cuando el usuario lo desee, ofrece ayudas para la resolución de los mismos y permite compartir las soluciones con otros usuarios.

La aplicación se ha desarrollado en inglés y se encuentra publicada en Google Play.

Palabras clave

Circuitos combinacionales, Android, tableta, móvil, puertas lógicas, multiplexor, decodificador, codificador, memoria ROM, BCD 7-segmentos, PAL, Karnaugh.

Abstract

This project consists in the realization of an Android application. It will develop an exercise guide and a tutorial about combinational circuits. The exercises have been chosen of the subject Circuitos Electrónicos Digitales of the Escuela Politécnica Superior, Universidad Autónoma de Madrid.

The main objective, is to offer a guide for the students with exercises of circuit synthesis, adapted to be solved in a normal screen of a smartphone. Besides, users can check the solution when they want to, also there are tips for helping in the exercise resolution and is possible to share solutions with others users.

Application is posted on Google Play and is written in English.

Key words

Combinational circuits, Android, tablet, Smartphone, logic gates, multiplexer, decoder, encoder, ROM memory, BCD 7-segments, PAL, Karnaugh.

Agradecimientos

Sin duda, los agradecimientos más especiales son para mis padres, y mi hermana Ana. Por sus consejos, por su ayuda, por sus broncas y por tantas cosas que han hecho posible que llegue hasta aquí.

A mi novia, Rocío. Por su apoyo incondicional, su compañía y comprensión, por hacer que todo este tiempo haya sido tan especial y por todo lo que nos queda por delante.

A Eduardo, por brindarme la oportunidad de realizar este proyecto y por su ayuda durante todo este periodo.

A Pablo, un gran compañero durante todos estos años y también a todo el grupo de amigos que hemos hecho aquí en la escuela y que espero que dure por muchos años.

ÍNDICE DE CONTENIDOS

1	INTRODUCCIÓN	1
1.1	Motivación	1
1.2	Objetivos	1
1.3	Organización de la memoria	3
2	ESTADO DEL ARTE	4
2.1	Dispositivos móviles inteligentes.....	4
2.2	Sistemas operativos para dispositivos móviles inteligentes	6
2.2.1	Android	6
2.2.2	IOS (iPhone OS).....	8
2.2.3	Windows Phone.....	9
2.2.4	Otros	9
2.3	Aplicaciones similares	10
3	DISEÑO	11
3.1	Finalidad del proyecto	11
3.2	Justificación de requisitos	11
3.2.1	Sistema operativo.....	11
3.2.2	Versiones de Android válidas	15
3.2.3	Tipos de dispositivos válidos	16
3.2.4	Tipos de pantalla válidos	16
3.2.5	Idioma	19
3.2.6	Resumen de requisitos	19
3.3	Limitaciones	20
3.4	Funcionalidades de la aplicación	20
3.4.1	Tutorial.....	20
3.4.2	Ejercicios	21
3.4.3	Resetear.....	21
3.4.4	Consejos y ayudas para la resolución de ejercicios.....	21
3.4.5	Comprobación de soluciones	22
3.4.6	Guardar soluciones.....	22
3.4.7	Cargar o borrar soluciones guardadas.....	22
3.4.8	Enviar soluciones guardadas	23
3.4.9	Breve manual de la aplicación	23
3.4.10	Botón Android: Menú	23

3.5	Aspectos docentes	24
4	DESARROLLO	34
4.1	Primeros pasos.....	34
4.2	Herramientas necesarias para el desarrollo del proyecto.....	34
4.3	Esquema general de la aplicación.....	35
4.4	Conceptos clave	36
4.4.1	Actividad	36
4.4.2	Ciclo de vida de una actividad	36
4.4.3	Vista	37
4.5	Menús de la aplicación	37
4.6	Gestión del botón Android: Menú.....	37
4.7	El fichero manifiesto	38
4.8	Diseño y desarrollo de ejercicios tipo tabla.....	38
4.8.1	Vista	38
4.8.2	Actividad	40
4.9	Adaptación a diferentes pantallas	41
4.9.1	Adaptación de vistas en XML.....	41
4.9.2	Adaptación de vistas en la actividad	41
4.10	Tutorial.....	44
4.11	Diseño y desarrollo de ejercicios de conexiones en circuitos	45
4.12	Consejos para la resolución de ejercicios	47
4.13	Gestión de datos de la aplicación	48
4.13.1	Método de preferencias	48
4.13.2	Guardar soluciones.....	48
4.13.3	Cargar o borrar soluciones	49
4.14	Sistema de comprobación de ejercicios	50
4.14.1	Comprobación en ejercicios tipo tabla.....	50
4.14.2	Comprobación en ejercicios de conexiones	50
4.15	Envío de soluciones.....	52
4.16	Diagrama de clases de la aplicación	54
4.16.1	Descripción de clases.....	55
4.16.2	Descripción de métodos principales	56
5	INTEGRACIÓN, PRUEBAS Y RESULTADOS.....	57
5.1	Primera publicación en Google Play Store.....	57
5.2	Publicaciones posteriores	58

5.3	Estadísticas proporcionadas por Google Play Store	58
5.4	Cuestionario de la aplicación	62
6	CONCLUSIONES Y TRABAJO FUTURO	64
6.1	Conclusiones	64
6.2	Trabajo futuro	65
	REFERENCIAS	67
	ANEXOS.....	- 1 -
	PRESUPUESTO.....	- 1 -
	PLIEGO DE CONDICIONES	- 2 -

ÍNDICE DE FIGURAS

Figura 1-1: Guía de problemas actualizada	2
Figura 2-1: IBM Simon, primer “smartphone”	4
Figura 2-2: “Smartphone” actual.....	5
Figura 2-3: Arquitectura Android	8
Figura 3-1: Evolución cuota de mercado SO para móviles.....	13
Figura 3-2: Evolución cuota de mercado SO tabletas.....	13
Figura 3-3: Porcentaje de tabletas vs móviles.....	14
Figura 3-4: IOS y Android en tabletas y móviles.....	14
Figura 3-5: Cuota de dispositivos activos por versión de Android	15
Figura 3-6: Diagonal de pantalla.....	16
Figura 3-7: Tamaños de pantalla en pulgadas.....	17
Figura 3-8: Distribución por tamaño de pantalla	17
Figura 3-9: Distribución por densidad de pantalla	18
Figura 4-1: Esquema general de la aplicación	35
Figura 4-2: Vista ejercicio tipo tabla	39
Figura 4-3: Adaptación de vistas en la actividad	42
Figura 4-4: Tutorial	44
Figura 4-5: Regiones de toque para conexiones	46
Figura 4-6: Ayuda a la resolución de ejercicios	47
Figura 4-7: Agrupación de datos para comprobación de resultados	51
Figura 4-8: Intercambio de conexiones en las entradas	52
Figura 4-9: Enviar archivos	53
Figura 4-10: Diagrama de clases.....	54
Figura 5-1: Evolución de instalaciones totales hasta el 11-02-2014	59
Figura 5-2: Evolución de instalaciones activas hasta el 11-02-2014	59
Figura 5-3: Instalaciones totales por país con dos o más descargas a 11-02-2014	60
Figura 5-4: Instalaciones totales por versión del sistema operativo a 11-02-2014	61
Figura 5-5: Instalaciones actuales por versión de la aplicación a 11-02-2014.....	61

ÍNDICE DE TABLAS

Tabla 2-1: Versiones de Android	7
Tabla 3-1: Versiones de Android instaladas	15
Tabla 3-2: Distribución tamaños de pantalla	17
Tabla 3-3: Distribución por densidad de pantalla	18
Tabla 3-4: Densidades y tamaños de pantalla.....	19
Tabla 3-5: Problemas tipo 1.....	25
Tabla 3-6: Problemas tipo 2.....	26
Tabla 3-7: Problemas tipo 3.....	27
Tabla 3-8: Problemas tipo 4.....	28
Tabla 3-9: Problemas tipo 5.....	29
Tabla 3-10: Problemas tipo 6.....	30
Tabla 3-11: Problemas tipo 7.....	31
Tabla 3-12: Problemas tipo 8.....	32
Tabla 3-13: Problemas tipo 9.....	32
Tabla 3-14: Problemas tipo 10.....	33
Tabla 4-1: Descripción de clases.....	55
Tabla 4-2: Descripción de métodos	56
Tabla 5-1: Versiones de la aplicación	58
Tabla 5-2: Cuestionario.....	62

1 INTRODUCCIÓN

1.1 Motivación

En los últimos años se han desarrollado dispositivos móviles inteligentes con características cercanas a las de algunos ordenadores personales. El uso de estos dispositivos se ha extendido enormemente, de tal manera que prácticamente todo el mundo dispone de uno de ellos.

Este proyecto surge con la idea de acercar las ventajas de los dispositivos inteligentes a los estudios universitarios, en particular los relacionados con electrónica combinacional.

La manera de llevar este proyecto a cabo, es desarrollando una aplicación prototipo (para la plataforma Android), en la que se pretende poner a disposición de los alumnos contenidos teóricos y prácticos.

Entre las ventajas que ofrecen estos dispositivos podemos destacar:

- Comodidad de uso.
- Sustitución del papel.
- Tener la información disponible con nosotros en todo momento.
- Comprobaciones y ayudas interactivas en los ejercicios, que no están disponibles en las guías de problemas en papel.
- Compartición de resultados con otros usuarios.
- Posibilidad de estudiar en cualquier sitio (autobús, metro, etc.).
- Utilización sin necesidad de internet.

1.2 Objetivos

El objetivo de este proyecto es desarrollar una aplicación Android destinada a los alumnos de la asignatura Circuitos Electrónicos Digitales de la Escuela Politécnica Superior (Universidad Autónoma de Madrid). Se ofrecerá una guía de doce problemas de circuitos digitales combinacionales, y también un tutorial sobre esta temática.

El desarrollo también tiene por objetivo fijar los siguientes parámetros:

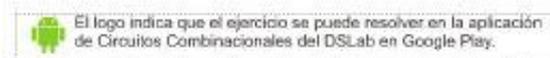
- Relación de aspecto de figuras.
- Tamaños de letras.
- Coordenadas de conexión.
- Separación mínima de áreas activas.
- Gestión interna y externa de datos.

Una vez fijados estos parámetros se utilizarán en el resto de las aplicaciones que se desarrollarán en el DSLab, entre ellas:

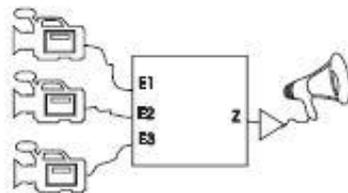
- FSM Mealy.
- FSM Moore.
- Circuitos MOS.
- Circuitos Aritméticos.

Estas aplicaciones deben complementar a las guías de problemas de Circuitos Electrónicos Digitales, las cuales indicarán qué ejercicios están disponibles en el teléfono, tal como se muestra en el siguiente ejemplo:

CIRCUITOS ELECTRÓNICOS DIGITALES
ESCUELA POLITÉCNICA SUPERIOR – UNIVERSIDAD AUTÓNOMA DE MADRID
Guía de Problemas nº1:
Tema: Circuitos Combinacionales
Hoja: MUXs y DEMUXs



1. Con puertas AND, OR e INV diseñe un: a) MUX 2-1, b) MUX 4-1, y c) MUX 8.
2. Un sistema de seguridad incluye 3 circuitos de monitorización idénticos que apuntan a una misma zona, cuyas salidas son E1, E2 y E3. Estas se ponen independientemente a "1", si detectan movimiento. Para evitar falsas alarmas, se pide diseñar un circuito de alarma con votación, cuya salida Z sea "1" cuando 2 o más de las señales E1, E2 y E3 se activen. Diseñe el circuito con: a) un Mux 8-1; b) un Mux 4-1 y c) un Mux 2-1 y puertas adicionales



3. Diseñar con MUX 2-1 los siguientes circuitos: a) MUX 4-1, b) MUX 8-1 y c) MUX 16-1.
4. Con un MUX 4-1 diseñar una puerta a) XOR, b) AND y c) OR.
5. Se quiere detectar la paridad de un mensaje de tres bits (b2, b1 y b0) de tal forma que cuando una línea de control C, esté a '0', se indique paridad par y cuando C = '1', se indique paridad impar. Sintetice el circuito utilizando el multiplexor 4-1 y el mínimo número de puertas de la figura. Indique las conexiones con un punto ó una X. NOTAS: Paridad par: cuando la cantidad de "1" de una palabra es un número par (por ejemplo, el número 3 es impar – no es divisible por 2 - pero tiene paridad par: 2 unos). Se considera que el cero es par.



Figura 1-1: Guía de problemas actualizada

Desde el punto de vista educativo del proyecto en sí, el objetivo es adquirir los siguientes conocimientos:

- Programación orientada a objetos: JAVA.
- Lenguaje XML.
- Estructura de aplicaciones Android.
- Conocimiento de la API Android.
- Herramientas de diseño gráfico.

Es importante reseñar que el plan de estudios de Ingeniería de Telecomunicación de la EPS no cuenta con ninguna asignatura obligatoria sobre JAVA.

La aplicación desarrollada ofrecerá la posibilidad de resolver problemas y comprobar sus soluciones. Además, se podrán guardar, cargar y enviar por correo electrónico las respuestas introducidas por el alumno.

El tutorial y los ejercicios tratan los siguientes conceptos:

- Puertas lógicas.
- Multiplexores.
- Memoria ROM.
- BCD 7-segmentos
- Tablas binarias.
- Mapas de Karnaugh.
- Decodificadores.
- Codificadores.
- PALs.

En principio, se enfatizarán los problemas relacionados con la síntesis de circuitos. Se estudiarán alternativas para adaptarlos a teléfonos móviles y limitar la libertad de diseño a cambio de conseguir simplicidad.

La aplicación debe estar disponible en Google Play Store, para que cualquier usuario de Android pueda descargarla.

1.3 Organización de la memoria

- Capítulo 1: Introducción.
- Capítulo 2: Estado del arte.
- Capítulo 3: Diseño del proyecto.
- Capítulo 4: Desarrollo del proyecto.
- Capítulo 5: Integración, pruebas y resultados.
- Capítulo 6: Conclusiones y trabajo futuro.

2 ESTADO DEL ARTE

En este apartado se va a realizar un pequeño repaso de la evolución de los dispositivos móviles inteligentes, y los sistemas operativos móviles más importantes. Además se realizará un estudio de aplicaciones educativas similares a la realizada en el proyecto.

2.1 Dispositivos móviles inteligentes

Generalmente se define a un teléfono móvil inteligente o “smartphone”, como un dispositivo que aúna las características de los teléfonos móviles convencionales y de las PDAs. Más adelante se fueron añadiendo muchas más funcionalidades.

IBM Simon, sacado a la venta en 1994, es considerado el primer “Smartphone” de la historia. Su pantalla era táctil y no hacía falta un puntero para utilizarla. Incorporaba servicios de voz y datos, por lo que funcionaba como teléfono, asistente personal e incluso como máquina de fax. Contaba con calendario, libreta de direcciones, reloj mundial, calculadora, bloc de notas, correo electrónico y juegos.



Figura 2-1: IBM Simon, primer “smartphone”

**Imagen obtenida de:*

http://commons.wikimedia.org/wiki/File:IBM_Simon_in_charging_station.png

Posteriormente aparecieron más dispositivos de este tipo, principalmente de las marcas PALM y NOKIA. Ninguno de estos dispositivos tuvo verdadera repercusión y no triunfaron en el mercado aun siendo un grandísimo avance tecnológico.

El verdadero estallido de los “smartphones” viene en 2007 con el lanzamiento del iPhone de Apple. Vendieron millones de unidades de este dispositivo. Sin duda, ofrecía la mejor experiencia en internet desde un teléfono móvil hasta la fecha.

Paralelamente, y también en 2007, Google presentaba el sistema operativo Android. Sin hacer mucho ruido en sus comienzos, ahora se alza como el sistema operativo móvil más extendido en el mundo.

Desde 2007, el crecimiento de los dispositivos móviles inteligentes ha crecido de manera desmesurada, es raro encontrar a alguien que no cuente con uno de estos hoy en día. Además, sus características (hardware y software) han mejorado tanto, que en muchos casos permiten sustituir a los ordenadores personales.



Figura 2-2: “Smartphone” actual

**Imagen obtenida de:*

http://images.samsung.com/is/image/samsung/es_SM-N9005ZKEPHE_000216751_combination3_black?§Download-Source§

Paralelamente a los “smartphones”, se desarrollaron también las tabletas, dispositivos con pantallas más grandes que tenían características y funcionalidades similares a las de los teléfonos inteligentes. El auge de las tabletas comenzó aproximadamente en 2010 con la llegada del iPad, también de la marca Apple.

2.2 Sistemas operativos para dispositivos móviles inteligentes

En este apartado se van a explicar los principales sistemas operativos para móviles, haciendo mayor hincapié en Android, ya que es el elegido para desarrollar el proyecto.

2.2.1 Android

Historia

El sistema operativo Android fue desarrollado inicialmente por Android Inc., una pequeña empresa de Palo Alto, California, que fue fundada en 2003. En julio de 2005 fue comprada por Google.

Dos años después de que Google se hiciera con Android se creó la Open Handset Alliance. Un consorcio de varias compañías importantes que pretendía desarrollar estándares abiertos para dispositivos móviles. Más adelante estrenó su primer producto, Android.

En octubre de 2008 se lanzó el HTC Dream, el primer dispositivo con Android que apareció en el mercado, contaba con la versión 1.0 del sistema operativo.

Características destacadas del sistema operativo:

- Conectividad.
- Mensajería.
- Almacenamiento.
- Navegador web.
- Soporte multimedia.
- Soporte para streaming.
- Multitarea.
- Soporte para Hardware adicional.
- Tethering.

Desde su aparición en el mercado, Android no ha parado de extenderse y de desarrollarse. Actualmente es el sistema operativo para “smartphone” más usado. Además, cuenta con una gran comunidad de desarrolladores. En muchas ocasiones, la comunidad saca actualizaciones para muchos teléfonos, cuando las propias marcas de dispositivos han dejado de actualizar. Esto permite a los usuarios alargar la vida útil de sus dispositivos sin que estos queden obsoletos.

Android cuenta con una tienda en la que ya hay cerca de un millón de aplicaciones disponibles.

A continuación se muestra una tabla con las principales actualizaciones del sistema. Desde 2008 hasta ahora se han desarrollado diecinueve versiones con distinta interfaz de programación de aplicaciones (API), donde se han ido añadiendo gran cantidad de funcionalidades.

Versión de Android	Nombre de la versión	Fecha de distribución	API
-	Beta	5-11-2007	-
1.0	Apple Pie	23-09-2008	1
1.1	Banana Bread	9-02-2009	2
1.5	Cupcake	30-4-2009	3
1.6	Donut	15-09-2009	4
2.0	Eclair	26-10-2009	5
2.0.1	Eclair	3-12-2009	6
2.1	Eclair	12-01-2010	7
2.2	Froyo	20-05-2010	8
2.3	Gingerbread	6-12-2010	9
2.3.3	Gingerbread	9-02-2011	10
3.0	Honeycomb	22-02-2011	11
3.1	Honeycomb	10-05-2011	12
3.2	Honeycomb	15-07-2011	13
4.0	Ice Cream Sandwich	19-10-2011	14
4.0.3	Ice Cream Sandwich	16-12-2011	15
4.1	Jelly Bean	9-07-2012	16
4.2	Jelly Bean	13-11-2012	17
4.3	Jelly Bean	24-07-2013	18
4.4	Kit Kat	31-10-2013	19

Tabla 2-1: Versiones de Android

Arquitectura

La arquitectura de Android está formada por cuatro capas, todas ellas de software libre:

1. **Núcleo de Linux:** Android utiliza la versión 2.6 del núcleo de Linux. El núcleo se encarga de gestionar todos los recursos del sistema como gestión de memoria, gestión de procesos, seguridad, pila de protocolos de red, soporte de drivers. Se puede decir que el núcleo funciona como una capa de abstracción entre el hardware y las demás capas de la arquitectura.
2. **Bibliotecas nativas:** Esta capa está formada por un conjunto de bibliotecas escritas en el lenguaje del sistema operativo, C/C++, y por "Android runtime". Este último a su vez, está compuesto por una serie de bibliotecas base en JAVA que proporcionan las principales funcionalidades de este lenguaje, además incluye la máquina virtual Dalvik, la cual está basada en registros, lo que hace que la ejecución sea más rápida. Dalvik se encarga de ejecutar las clases ya compiladas en el formato ".dex" (Dalvik Executable), el cual está optimizado para ahorrar memoria.

3. Entorno de aplicación: Esta capa está formada por todas las clases y servicios que utilizan las aplicaciones para realizar sus funciones. Estas clases utilizan las bibliotecas y la máquina virtual de la capa anterior.
4. Aplicaciones: Este nivel está compuesto por un conjunto de aplicaciones, algunas nativas, escritas en C o C++ y otras de tipo administradas, desarrolladas en JAVA.

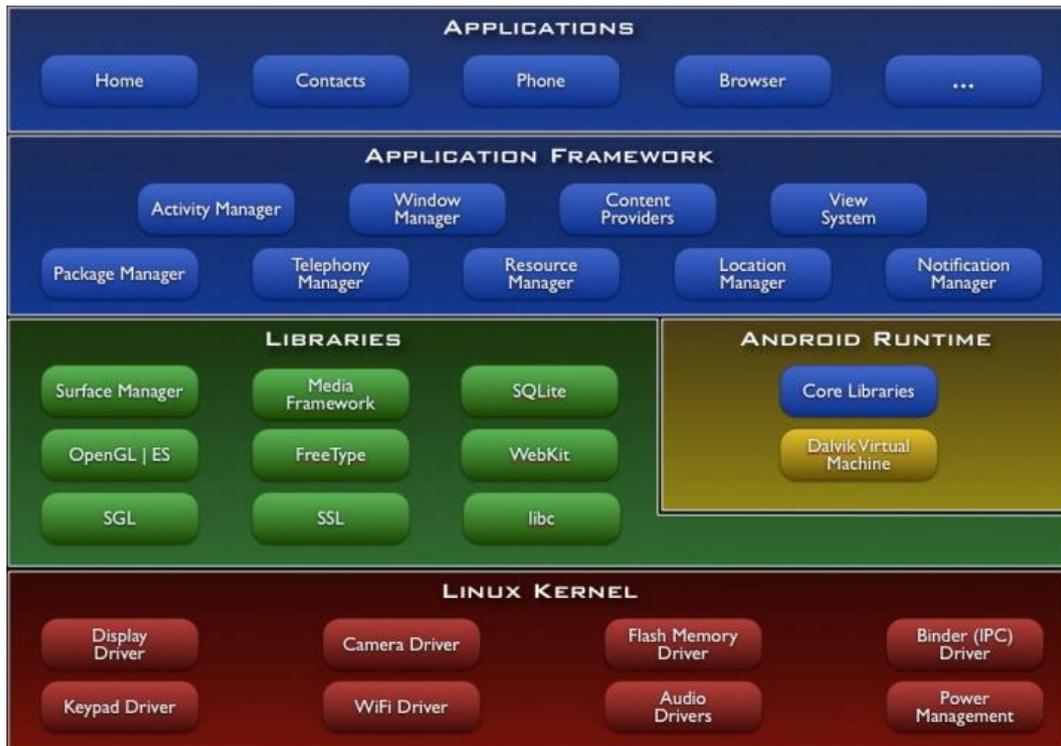


Figura 2-3: Arquitectura Android

*Imagen obtenida de:

<http://developer.android.com/images/system-architecture.jpg>

2.2.2 IOS (iPhone OS)

Sistema operativo móvil cerrado, desarrollado por Apple Inc., y solo disponible para dispositivos de su marca (iPhone, iPad, iPod, etc).

Este sistema fue una verdadera revolución, proponiendo una interfaz y una experiencia de usuario única hasta su aparición. Después de Android, es el sistema para dispositivos móviles inteligentes más extendido. El primer teléfono con este sistema fue el iPhone, lanzado en 2007.

Actualmente ya está disponible la séptima versión, IOS 7.

Sus aplicaciones se desarrollan con el programa XCode, usando el lenguaje Objective-C, basado en C y orientado a objetos.

2.2.3 Windows Phone

Sistema operativo cerrado desarrollado por Microsoft. La primera versión final, Windows Phone 7, se lanzó en octubre de 2010. Llegó con bastante retraso frente a sus competidores y está muy por detrás de ellos en cuanto a cuota de mercado. Aun así sigue creciendo poco a poco.

Actualmente se comercializa la segunda versión de este sistema, Windows Phone 8, del cual se han lanzado tres actualizaciones, la última en octubre de 2013.

Para el desarrollo de sus aplicaciones es necesario el uso del lenguaje XAML, con el que se diseña la interfaz de usuario, y C# o Visual Basic .NET, para la lógica de la aplicación.

2.2.4 Otros

Todos los sistemas siguientes tienen menor cuota de mercado en “smartphones” que los anteriores.

Symbian OS

Quizás uno de los sistemas más usados en todo el mundo. Fue desarrollado por la alianza de varias empresas de telefonía, como Samsung, Siemens, Nokia, Motorola, etc. Pero tras la llegada de los móviles inteligentes, el número de dispositivos con este sistema comenzó a caer en picado.

BlackBerry OS

Sistema desarrollado por RIM para dispositivos BlackBerry. Actualmente está en decadencia, pero hace unos años estuvo bastante extendido.

Bada

Anunciado en noviembre de 2009 y desarrollado por Samsung. No ha gozado de gran aceptación al no tener soporte para aplicaciones muy demandadas como por ejemplo WhatsApp.

Firefox OS

Está basado en HTML5 con núcleo Linux, de código abierto y desarrollado por Mozilla Corporation con el apoyo de otras empresas como Telefónica.

Ubuntu Touch

Otro sistema basado en Linux, en este caso desarrollado por Canonical Ltd, quien también desarrolla la versión para escritorio Ubuntu.

Tizen

El desarrollo de este sistema operativo está dirigido por Intel y Samsung. Está pensado principalmente para “Smartphones” y tabletas, pero también aparecerá dentro de poco en televisores.

2.3 Aplicaciones similares

Cuando se realiza un nuevo proyecto, es interesante ver si el mercado ofrece algo parecido, por lo que vamos a investigar si hay algo similar en las principales plataformas de aplicaciones de dispositivos móviles.

Cada uno de los tres sistemas operativos móviles principales tiene una tienda oficial de aplicaciones. Google Play Store es la de Android, App Store la de IOS y Windows Phone Store la de Windows Phone.

Vamos a centrar la búsqueda en las aplicaciones educativas que ofrezcan funcionalidades parecidas a las de este proyecto.

En cualquiera de las tres tiendas anteriores se puede encontrar gran cantidad de aplicaciones educativas de muchas índoles diferentes. Realizando búsquedas por las palabras clave de nuestra aplicación, encontramos distintas aplicaciones. La mayoría trata de forma teórica algún punto de electrónica, y en ningún caso se ha encontrado una aplicación para la resolución de ejercicios de circuitos combinacionales.

La aplicación que más parecido tiene con la nuestra puede ser “LogicMaker”, desarrollada para Android. En ella se pueden diseñar distintos circuitos combinacionales, con puertas lógicas y demás, y realizar comprobaciones en ellos. Pero no propone ejercicios, teoría o compartición de datos como hacemos nosotros.

Aunque en esta aplicación no se busca el éxito comercial, es interesante que no se haya encontrado ninguna parecida, ya que es posible que ofrezcamos algo novedoso. En cualquier lugar, el principal objetivo es ayudar a los alumnos de esta escuela con el aprendizaje de circuitos combinacionales.

3 DISEÑO

En este apartado se describen las principales características del proyecto y los diferentes requisitos y limitaciones para la realización del mismo. Además se explican el conjunto de funcionalidades disponibles en la aplicación y las diferentes elecciones tomadas.

3.1 Finalidad del proyecto

El desarrollo de los dispositivos móviles en los últimos años es de sobra conocido por todos, y cada vez es más común que estos incluyan funcionalidades típicas de otros elementos como pueden ser un libro electrónico o incluso como sustituto de papel y bolígrafo.

El objetivo de este proyecto es realizar una aplicación útil y sencilla de manejar, que en este caso permita a los estudiantes realizar diferentes ejercicios de electrónica combinacional desde la comodidad de su dispositivo móvil, ya sea tableta o “smartphone”. Se va a desarrollar en Android, sistema operativo móvil elegido para este proyecto. Más adelante se detallarán las causas de esta elección.

En principio la aplicación está pensada para los alumnos de Ingeniería de Telecomunicación de la EPS UAM, pero es válida para cualquier usuario que quiera comprobar sus conocimientos en el nivel y la temática que se ofrece.

La aplicación consta de un tutorial, y una serie de 12 ejercicios sobre circuitos combinatoriales. Se ofrecen ayudas para la resolución de los ejercicios, además de comprobaciones de los resultados introducidos por los usuarios. Se detallarán más adelante todas las funcionalidades disponibles.

3.2 Justificación de requisitos

3.2.1 Sistema operativo

Se ha elegido Android como sistema operativo para desarrollar la aplicación por diversas razones. A continuación se comparará con sus principales rivales.

En primer lugar, todas las herramientas de desarrollo necesarias son libres, es decir, no supone ningún coste adquirir las herramientas, las cuales se pueden descargar desde la página oficial de desarrolladores:

<http://developer.android.com/sdk/index.html>

Las herramientas están disponibles para los principales sistemas operativos de PC (Windows, MAC, LINUX). En el caso de IOS, sólo es posible desarrollar con dispositivos de la marca Apple, y en el caso de Windows Phone, con dispositivos que tengan Windows 7 o superior.

En el caso de Android se pueden desarrollar aplicaciones y probarlas en cualquier dispositivo sin tener necesariamente una cuenta de desarrollador. En IOS, sin cuenta de desarrollador, sólo es posible probar la aplicación en el simulador de "Xcode" (su herramienta de desarrollo). En Windows Phone solo es posible habilitar un dispositivo para probar tus aplicaciones.

Si por algún motivo nos planteamos publicar la aplicación en las tiendas oficiales de aplicaciones, Android también es la más barata, hay que abonar 25 \$, en cambio, en IOS son 99 € y en Windows Phone 75 €, además en estas dos últimas el pago es anual, mientras que en Android sólo se requiere un único pago. En Windows Phone han cambiado los costes hace poco, ahora son 19 \$, eso sí, anuales.

Al ser Android una plataforma libre, cuenta con una gran comunidad de desarrolladores y cientos de páginas web que proporcionan información, ayuda y sugerencias para el desarrollo de aplicaciones. Además de esto, en la página oficial hay una documentación muy completa de todas las herramientas, clases, métodos, vistas, etc.

En cuanto al número de usuarios potenciales, Android es el sistema que más ha crecido en los últimos años, y actualmente es el que tiene una cuota de mercado mayor entre los sistemas operativos para móviles y tabletas. Con esto aseguramos llegar a una mayor cantidad de usuarios.

A continuación se muestran algunas gráficas mostrando datos reales acerca de este último punto.

Primero se mostrará, la evolución de la cuota de mercado de los principales sistemas operativos móviles. Como se puede apreciar, desde hace prácticamente un par de años, Android es el sistema móvil predominante, aumentando su ventaja frente al resto con el paso del tiempo.

IOS se mantiene en una cuota cercana al 25 %, mientras que BlackBerry OS y sobre todo Symbian OS se han desplomado en este periodo.

Por último, Windows Phone, que ha sido el más tardío en aparecer en el mercado, muestra un muy ligero crecimiento.

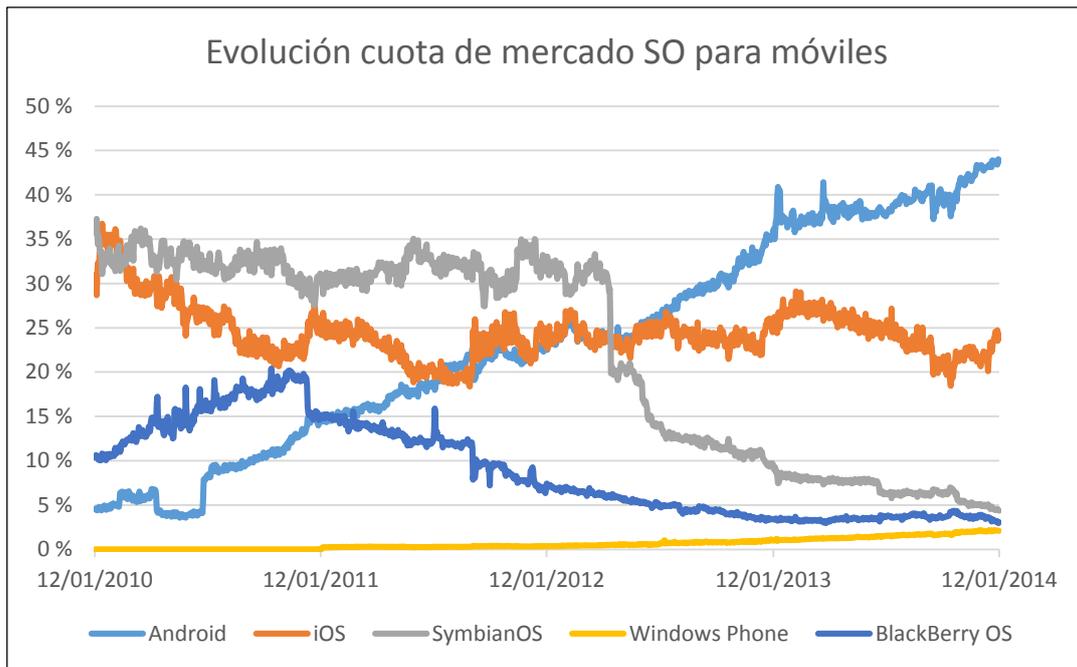


Figura 3-1: Evolución cuota de mercado SO para móviles

Ahora en el caso de las tabletas, se puede observar el dominio absoluto de IOS. Aunque Android tiene una cuota considerable, aún le queda mucho terreno para alcanzar a su gran rival.

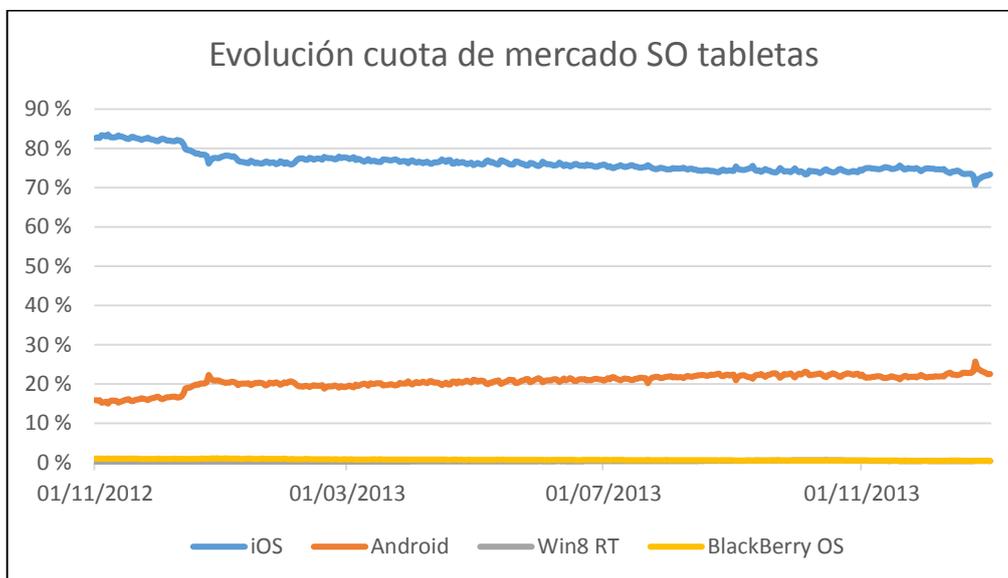


Figura 3-2: Evolución cuota de mercado SO tabletas

Viendo las dos figuras anteriores pueden surgir dudas en la elección del sistema operativo entre IOS y Android.

En la siguiente figura se muestra el porcentaje de móviles y tabletas con respecto al total de ambas:

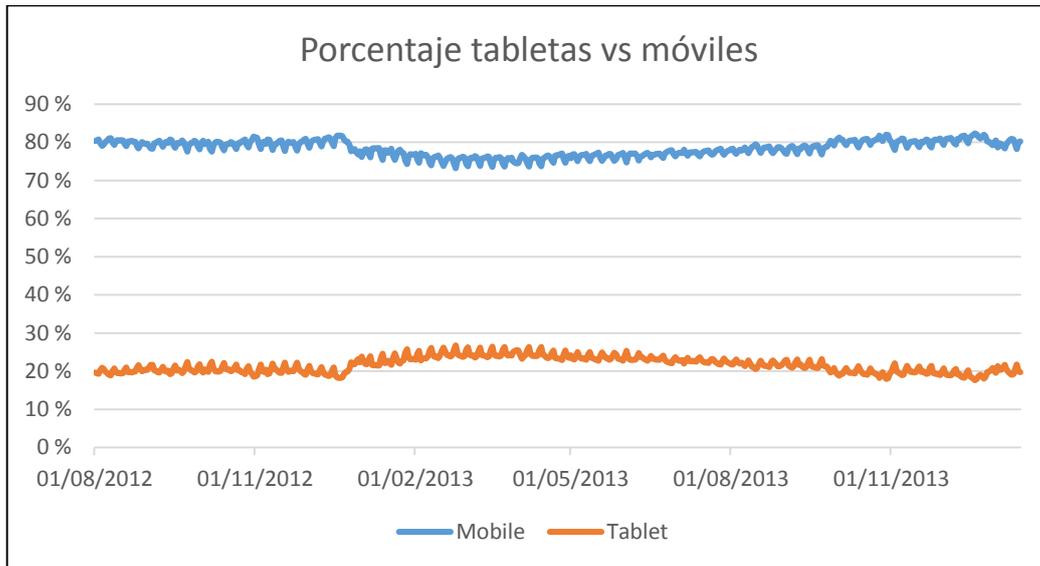


Figura 3-3: Porcentaje de tabletas vs móviles

La conclusión que se puede sacar de aquí, es que hay muchos más móviles que tabletas, aproximadamente por cada tableta hay cuatro móviles.

Para terminar con este apartado se comparan los dos sistemas más extendidos, IOS y Android, en términos de dispositivos totales:

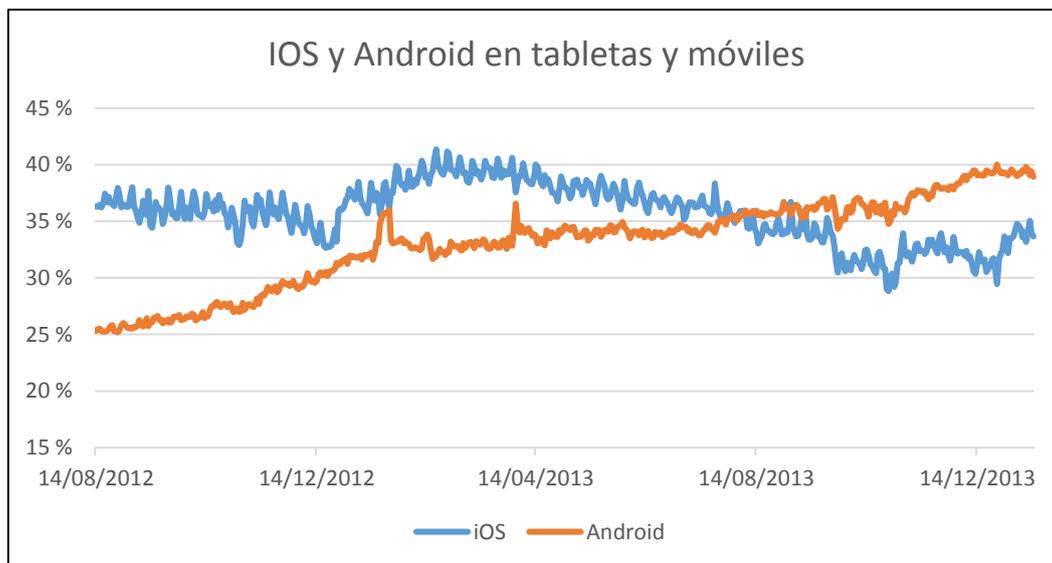


Figura 3-4: IOS y Android en tabletas y móviles

Aunque IOS tiene mucha más cuota de mercado en tabletas, la ventaja de Android en teléfonos móviles hace que sume más dispositivos en total.

Por consiguiente y después de todas las evaluaciones anteriores, se ha elegido Android como sistema operativo para el desarrollo del proyecto.

**Todos los datos utilizados en este apartado se han extraído de la web:
<http://gs.statcounter.com/>*

3.2.2 Versiones de Android válidas

Para elegir las versiones de Android que pueden ejecutar la aplicación, se tuvieron en cuenta los siguientes datos.

Versión de Android	Nombre de la versión	Fecha de distribución	Cuota
2.2	Froyo	20-05-2010	1.3 %
2.3	Gingerbread	6-12-2010	21.2 %
3.0	Honeycomb	22-02-2011	0.1 %
4.0	Ice Cream Sandwich	19-10-2011	16.9 %
4.1, 4.2, 4.3	Jelly Bean	9-07-2012, 13-11-2012, 24-07-2013	59.1 %
4.4	Kit Kat	31-10-2013	1.4 %

Tabla 3-1: Versiones de Android instaladas

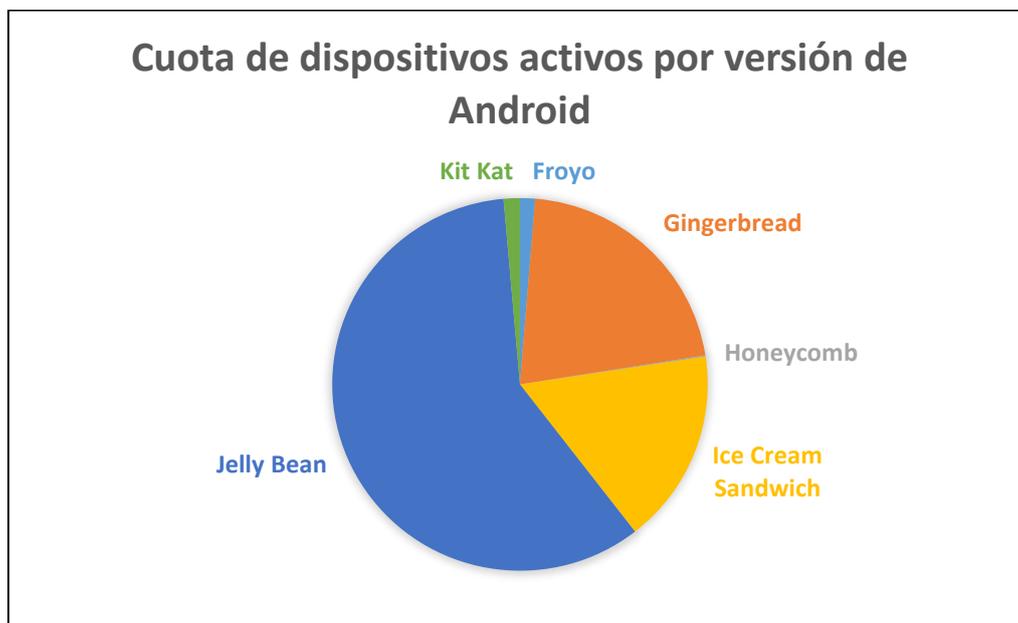


Figura 3-5: Cuota de dispositivos activos por versión de Android

Estos datos muestran la distribución de versiones de Android instaladas en dispositivos activos en la primera semana de Enero de 2014.

En este caso, el punto principal es elegir la mínima versión de Android en la que se podrá ejecutar la aplicación, ya que la última siempre es recomendable que esté disponible.

Aunque "Froyo", tiene un porcentaje bajo de dispositivos activos, se decidió que fuera la versión más antigua disponible para la aplicación, ya que no añadía ninguna limitación significativa.

**Todos los datos utilizados en este apartado se han extraído de la web oficial de desarrolladores:*

<http://developer.android.com/about/dashboards/index.html>

3.2.3 Tipos de dispositivos válidos

Se valoró la opción de realizar el proyecto sólo para tabletas, pero al no estar estas tan extendidas, (ver *Figura 3-3*) decidimos hacer el proyecto compatible con tabletas y móviles.

3.2.4 Tipos de pantalla válidos

En Android se hacen varias distinciones para las pantallas de sus dispositivos.

En primer lugar, diferenciamos las pantallas por su tamaño, indicado por un valor en pulgadas. Este dato se refiere a la longitud de la diagonal de la pantalla del dispositivo.



Figura 3-6: Diagonal de pantalla

Ahora bien, como hay una infinidad de dispositivos con diferentes tamaños de pantalla, Android los agrupa en cuatro escalones.

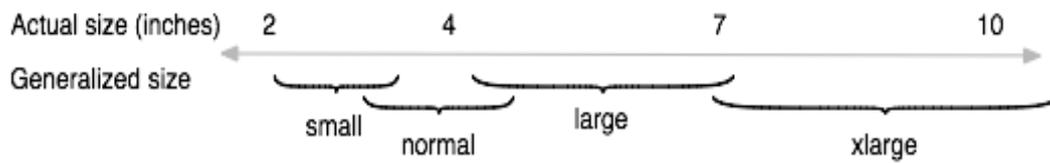


Figura 3-7: Tamaños de pantalla en pulgadas

La distribución de dispositivos de cada grupo se muestra en la siguiente tabla y posterior gráfico.

Tamaño de pantalla	Distribución
Small	8,20%
Normal	78%
Large	8,40%
Xlarge	5,40%

Tabla 3-2: Distribución tamaños de pantalla

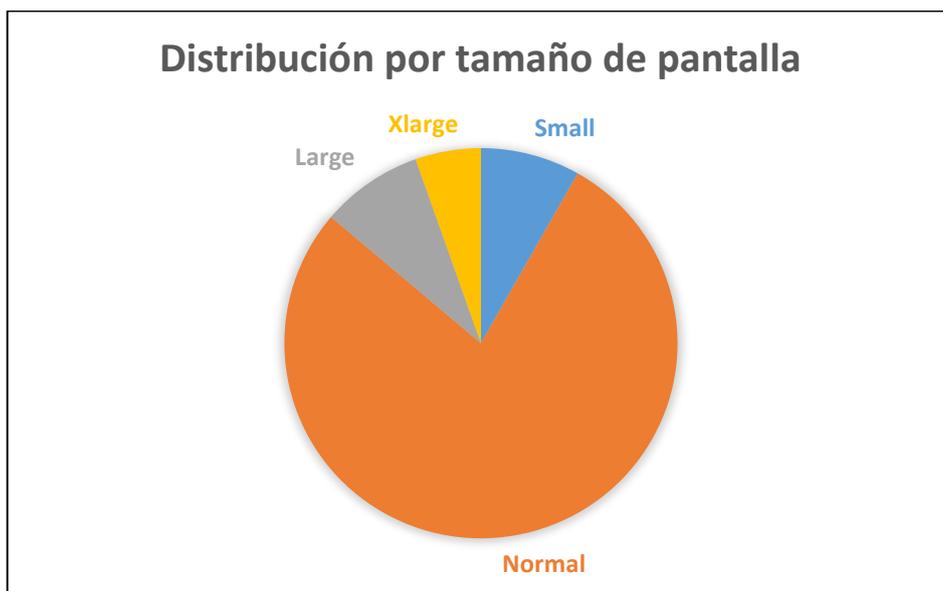


Figura 3-8: Distribución por tamaño de pantalla

Queda claro el predominio de dispositivos con pantallas de tamaño “normal”, por lo que se ha incluido este tamaño de pantalla entre los disponibles para el uso de la aplicación.

También se incluyen los tamaños “large” y “xlarge”, ya que al tener un mayor tamaño permiten que el estudio y la visibilidad de la aplicación sea mejor.

Por último, aunque quizá la experiencia de usuario no sea tan buena, se ha decidido que el tamaño “small” también esté entre los disponibles para usar la aplicación, y así poder llegar a un número mayor de dispositivos.

En segundo lugar, diferenciamos las pantallas por su densidad. Esta se mide en dpi (“dots per inch”), puntos por pulgada.

Estos son los distintos grupos de densidades que distingue Android, de menor a mayor densidad:

Densidad de pantalla	Distribución
ldpi	9,30%
mdpi	23,40%
tvdpi	1,70%
hdpi	34%
xhdpi	21%
xxhdpi	10,60%

Tabla 3-3: Distribución por densidad de pantalla

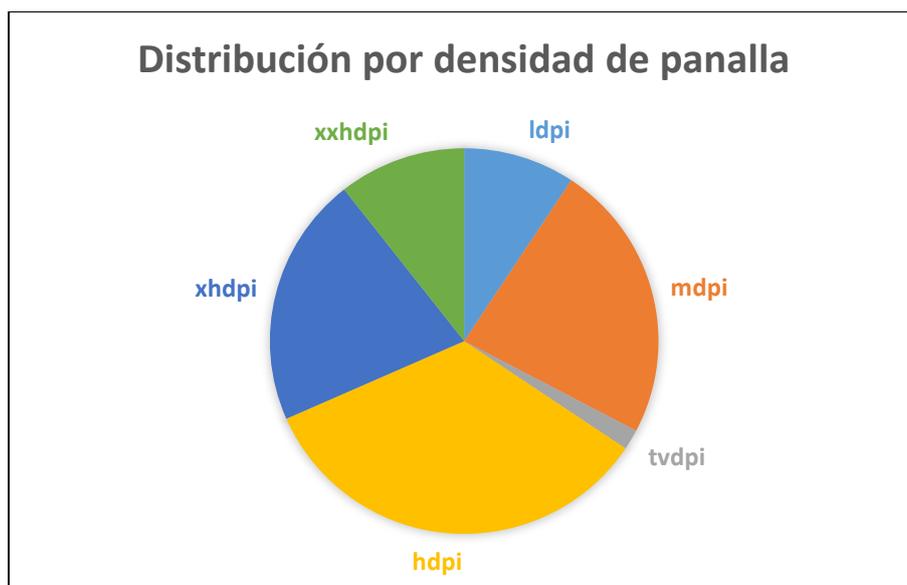


Figura 3-9: Distribución por densidad de pantalla

Cuanto mayor sea la densidad de las pantallas, con mayor detalle mostrarán estas el contenido. Además hay que tener en cuenta que para el mismo escalón de tamaño, nos podemos encontrar en diferentes grupos de densidades de pantalla.

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi
Small	8,2 %					
Normal	0,1 %	14,2 %		33,0 %	20,1 %	10,6 %
Large	0,9 %	4,5 %	1,7 %	0,6 %	0,7 %	
Xlarge	0,1 %	4,7 %		0,4 %	0,2 %	

Tabla 3-4: Densidades y tamaños de pantalla

El principal problema en este apartado, viene con las densidades “ldpi”, que en su mayoría corresponden a las pantallas de tamaño “small”. Al ser pantallas pequeñas y además con baja densidad, hace más complicada la visualización de imágenes o vistas con cierto grado de detalle. Se ha aceptado este tipo de pantalla por la misma razón anterior, llevar la aplicación a un mayor número de dispositivos.

En este caso, se ha decidido que todas las densidades sean válidas para el uso de la aplicación. No se considera “tvdpi”, al estar destinada a televisores.

**Todos los datos utilizados en este apartado se han extraído de la web oficial de desarrolladores: <http://developer.android.com/about/dashboards/index.html>*

3.2.5 Idioma

El idioma elegido para la aplicación es el inglés, y en particular el de Estados Unidos.

La razón principal es que el inglés, es el idioma más extendido.

La ficha en la publicación en Google Play Store se ha realizado tanto en inglés como en castellano.

3.2.6 Resumen de requisitos

Sistema operativo:

- Android.

Versiones del sistema operativo:

- 2.2 hasta 4.4.

Tipos de dispositivo:

- Móviles.
- Tablet.

Tipos de pantalla:

- Por tamaño: "Small", "Normal", "Large", y "Xlarge".
- Por densidad: "ldpi", "mdpi", "hdpi", "xhdpi", "xxhdpi".

Idioma de la aplicación:

- Inglés de Estados Unidos.

3.3 Limitaciones

Las principales limitaciones vienen dadas por el tipo de ejercicio y por el tutorial:

- Los circuitos, deben tener un tamaño relativamente pequeño para poder contemplarlo en la pantalla de cualquier dispositivo sin la necesidad de hacer zoom.
- El número de conexiones en los ejercicios no será mayor de 100-140, ya que esto haría que los puntos de conexión estuvieran demasiado cerca, y fuera demasiado complejo acertar en las pulsaciones.
- Las imágenes usadas en los ejercicios y en el tutorial, tienen una densidad máxima de 300 ppi ("pixels per inch"), ya que el uso de imágenes con mayor densidad aumentaría mucho el tamaño final de la aplicación.

3.4 Funcionalidades de la aplicación

En este apartado se van a explicar de forma resumida las diferentes funciones que se pueden llevar a cabo con la aplicación desarrollada.

3.4.1 Tutorial

El tutorial consiste en una sucesión de imágenes que explican diferentes conceptos a cerca de circuitos combinatoriales y que posteriormente serán útiles para la resolución de los ejercicios.

El texto o imágenes se muestran en pantalla completa y es posible pasar páginas hacia adelante o hacia atrás, pulsando en la parte derecha o izquierda de la pantalla respectivamente.

Para salir de este modo, simplemente hay que pulsar el botón "back" de Android.

El tutorial ha sido desarrollado por el profesor Eduardo Boemo.

3.4.2 Ejercicios

En la aplicación hay disponibles doce ejercicios sobre circuitos combinacionales. Cuando se accede a cada uno de ellos, se muestra el enunciado del mismo, y las opciones de empezar a resolver y cargar o borrar soluciones guardadas.

En algunos ejercicios se puede pedir rellenar una o varias tablas y en otros se mostrará un circuito en el que habrá que completar diferentes conexiones.

En esta pantalla de resolución de ejercicio, se pueden realizar diferentes acciones adicionales:

- Comprobar si solución introducida es correcta.
- Guardar la solución actual en un fichero.
- Mostrar una tabla o esquema si el ejercicio lo requiere.
- En algunos casos se ofrecen consejos para ayudar a la resolución del ejercicio.
- Resetear los datos introducidos.

3.4.3 Resetear

Esta función está disponible en todos los ejercicios y permite borrar los datos introducidos por el usuario, dejando el ejercicio en blanco. Antes de eliminarlos definitivamente, se consulta al usuario si está seguro de la eliminación, ya que estos datos no se podrán recuperar posteriormente.

3.4.4 Consejos y ayudas para la resolución de ejercicios

Esta opción sólo está disponible para determinados ejercicios.

Se puede acceder a esta funcionalidad desde la pantalla de resolución de ejercicios, pulsado el botón "Tips".

Una vez dentro de la pantalla de consejos, podemos ir visualizándolos uno a uno.

Algunos consejos se muestran en forma de alerta referidos a la lógica del problema, otros mostrando diferentes conexiones en los circuitos.

El objetivo de este apartado es ofrecer unas pequeñas pistas, para que el usuario pueda realizar el ejercicio y así poder aprender los conceptos que el problema requiere.

3.4.5 Comprobación de soluciones

En la pantalla de resolución de ejercicio, está disponible la opción de comprobación. Accediendo a ella, se realiza una comprobación de la solución actual introducida por el usuario.

Posteriormente, se indica en pantalla si esta solución es correcta o no.

La corrección se puede llevar a cabo en cualquier momento de la resolución. En algunos ejercicios de tablas se muestran correcciones más detalladas, indicándose el error por filas o por zonas.

3.4.6 Guardar soluciones

Se podrá guardar la solución introducida por el usuario en cualquier momento.

La primera vez que se guarda un archivo, se crea un directorio en el dispositivo donde se guardarán todos los ficheros.

A continuación se mostrará una alerta en la que hay que escribir el nombre con el que se quiere guardar el archivo. Al nombre del archivo se le añade una cabecera que aparece en la alerta y que sirve para distinguirlo de otros ejercicios.

Si por algún motivo el nombre del archivo elegido ya existe, se indicará, y se dará la opción de sobrescribirlo o de elegir otro nombre diferente para guardarlo.

En la cabecera antes mencionada, aparece una distinción del número de ejercicio que se está realizando y además, las letras "EM" o "IM", que nos indican si el archivo va a ser guardado en la memoria externa, o en la memoria interna.

3.4.7 Cargar o borrar soluciones guardadas

Cuando se pulsa en cualquier ejercicio, se pasa a la pantalla de enunciado del mismo. Aquí se puede iniciar la resolución del ejercicio o se puede ir a la opción de cargar o borrar ficheros. Al acceder a esta última, se muestran todos los ficheros guardados para este ejercicio.

Si queremos borrar algún fichero, debemos seleccionarlo y pulsar "delete". Se pueden eliminar varios ficheros a la vez. Antes de eliminarlo definitivamente, se consulta al usuario si está seguro de la eliminación, ya que estos datos no se podrán recuperar.

En cambio, si lo que queremos es cargar algún fichero, sólo debemos seleccionar uno, y pulsar "Load". A continuación se pasa a la pantalla de resolución de ejercicio, donde se cargará la solución que hay en este fichero, sobrescribiendo los datos que hubiera antes.

3.4.8 Enviar soluciones guardadas

A partir del menú principal de la aplicación, se pueden enviar soluciones por correo electrónico.

Se muestra un listado con todos los archivos guardados de todos los ejercicios. Se pueden seleccionar todos los archivos que se quiera, es decir, no es necesario enviarlos de uno en uno.

Cuando ya hemos seleccionado los ejercicios que se quieren enviar, se muestran los gestores de correo disponibles en el dispositivo Android.

Una vez seleccionado un gestor de correo, el usuario sólo debe introducir el correo del destinatario, y automáticamente se rellena el campo de asunto, se adjuntan los archivos seleccionados y se rellena el cuerpo del correo con las indicaciones necesarias para que el destinatario pueda utilizar los ficheros recibidos.

Estas indicaciones consisten en:

- Descargar los archivos en el dispositivo Android.
- Utilizar un gestor de archivos para acceder a ellos.
- Copiar los archivos en la carpeta de nuestra aplicación "CombinationalCircuits".

Si el destinatario realiza estos pasos correctamente, cuando abra nuestra aplicación, ya tendrá todos los ficheros cargados en ella.

Este es un método muy cómodo para que diferentes usuarios puedan compartir sus soluciones de una manera sencilla.

3.4.9 Breve manual de la aplicación

Es posible acceder a un breve manual de uso de la aplicación, en el que se explica cómo utilizarla de una forma gráfica y fácil.

Se puede acceder al manual a partir del menú principal, o a partir del botón de Android, "Menú".

3.4.10 Botón Android: Menú

El uso del botón "Menú" está activo en todas las pantallas de la aplicación, exceptuando las pantallas a las que se llega a partir de él.

Al pulsar este botón, se puede acceder a dos opciones:

- Help: Contiene el manual de usuario visto en el punto 3.4.9.
- About: Contiene el "Acerca de" de la aplicación.

3.5 Aspectos docentes

Los problemas de circuitos digitales se pueden dividir en dos ramas según su naturaleza:

Análisis: Un circuito completo es presentado al estudiante. Se debe descubrir su funcionalidad, extrayendo la tabla de verdad o el diagrama de estados. Es el tipo de problema que se presenta en ingeniería inversa.

Este tipo de problema también es útil para verificar si un estudiante tiene conocimientos básicos sobre un determinado tema. Por ejemplo, en el caso de la aplicación de este proyecto, se ha utilizado para que el estudiante verifique si conoce la simplificación gráfica por Veitch-Karnaugh o la función de bloques estándares como codificadores y decodificadores.

Síntesis: El estudiante se enfrenta a una especificación en forma de texto, gráfico o tabla y a partir de éstos, debe diseñar y construir el circuito final. Es el tipo de problema más habitual en ingeniería y el más utilizado en los exámenes de Circuitos Electrónicos Digitales en la EPS. De los 12 problemas propuestos en esta guía, 10 son de síntesis.

Un problema de la síntesis, es el dibujo final del circuito. Existen herramientas EDA (“Electronic Design Automation”) profesionales para ordenadores, que requieren monitores de al menos 24 pulgadas para poder aprovechar la potencia de las mismas. No es recomendable dibujar un circuito en un teléfono, aunque sea pequeño (en la actualidad existe una aplicación Android llamada “Logic Maker” que hace esto, pero su funcionamiento no es muy bueno).

De acuerdo con lo anterior, en este proyecto se ha optado por:

- Reducir el tamaño del problema en todos aquellos casos en los que el concepto a enseñar no depende de la complejidad del circuito medida en puertas. Hay problemas, como el de diseñar un árbol de reloj, que básicamente depende del tamaño del circuito. Pero en general, es posible concebir un circuito mínimo para la enseñanza de cada tema.
- Crear una estructura de circuito donde se restringe el grado de libertad. Por ejemplo, se pueden hacer conexiones pero no cambiar tipos de puertas, o dar a elegir entre varias opciones de un subconjunto de puertas.

Bajo las dos premisas anteriores es posible diseñar un conjunto de ejercicios que pueden ser resueltos en la pantalla de un teléfono móvil.

En términos pedagógicos, se ha optado por dar ayudas parciales para orientar al estudiante, antes de permitir que vea la solución definitiva. Las ayudas pretenden animar al estudiante a acabar el ejercicio. La inclusión de la respuesta completa se ha realizado de manera experimental y pueden desactivarse en versiones futuras. Para ello

se ha preparado una encuesta de uso entre los estudiantes de Circuitos Electrónicos Digitales de la EPS.

A continuación se resumen los problemas seleccionados y los objetivos docentes de los mismos.

Concepto: Firmware.
Tipo de Problema: Síntesis.
Problema nº: 1 y 2.
Descripción: En un circuito ya diseñado, el estudiante debe llenar el contenido de una memoria. El circuito se adapta simplemente cambiando dicho contenido. En el caso del Problema 2, el mismo hardware muestra que puede servir para manejar tanto displays ánodo común, como cátodo común.
<p>Figura:</p> <p>The diagram illustrates the connection between a ROM and a common anode 7-segment display. On the left, the ROM chip has address lines labeled A0 through A9, data lines labeled D0 through D7, and an output enable pin labeled OE. On the right, a common anode 7-segment display is shown with its segments connected to data lines D0 through D6. The common anode of the display is connected to data line D7. The OE pin is connected to a ground symbol.</p>

Tabla 3-5: Problemas tipo 1

Concepto: Diseño jerárquico.
Problema nº: 3.
Tipo de Problema: Síntesis.
Descripción: Construcción de multiplexores de más entradas a partir de multiplexores simples. Para el caso del problema, se hace un MUX 4-1 con 3 MUX 2-1. El estudiante debe conectar y agrupar las señales de control de los multiplexores simples.
<p>Figura:</p> $F(D, C, B, A) = DC + DB + CA$ <p>The diagram shows a 4x6 grid of inputs: D, C, B, A, Vcc, and GND. Three 2-to-1 multiplexers are used. The first multiplexer has inputs D and C, with control signal A. Its output is connected to the '0' input of the second multiplexer. The second multiplexer has inputs B and the output of the first multiplexer, with control signal C. Its output is connected to the '1' input of the third multiplexer. The third multiplexer has inputs Vcc and GND, with control signal A. Its output is the final function F.</p>

Tabla 3-6: Problemas tipo 2

Concepto: Multiplexor como LUT (*Look-up table*).

Problema nº: 3, 4, 5 y 9.

Tipo de Problema: Síntesis.

Descripción: Utilización de un multiplexor para realizar una función lógica, en lugar de la aplicación natural de estos circuitos en comunicaciones TDM (“time división multiplexing”).

Figura:

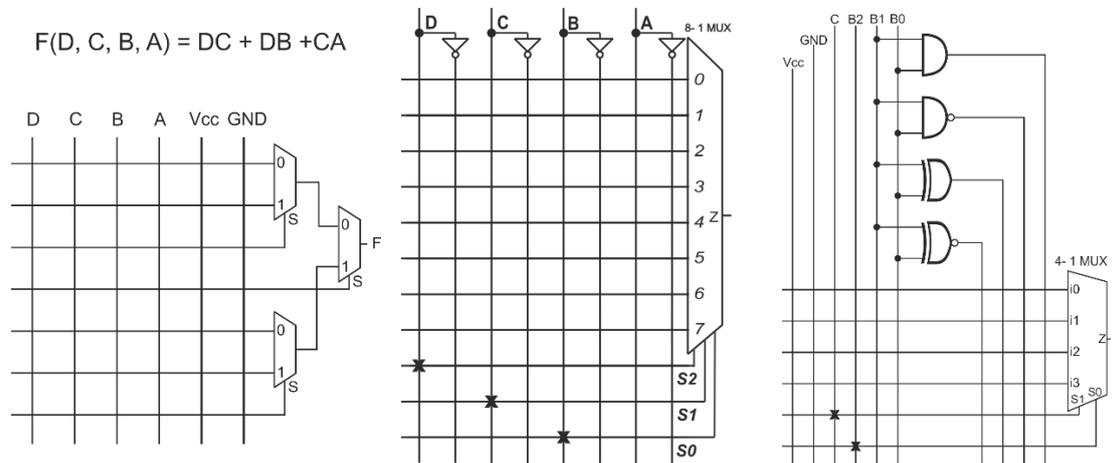


Tabla 3-7: Problemas tipo 3

Concepto: Señales activo bajo.
Problema nº: 6 y 8.
Tipo de Problema: Síntesis.
Descripción: Utilización de un multiplexor para realizar una función lógica. Sin embargo, este circuito tiene la opción de elegir la salida invertida (tal como ocurre en un multiplexor TTL tipo SN54/74LS151). En tal caso, la función a sintetizar debe ser la inversa a la que se desea obtener. No es necesario ningún cambio en el hardware. En el problema 8 se repite el concepto pero a partir de una PAL.
<p>Figura:</p> <p>The figure illustrates the implementation of the function $F(DCBA) = \text{SUM}(0, 2, 5, 8, 10, 12, 14)$ using an 8-to-1 MUX. On the left, a truth table is shown with inputs D, C, B, A, Vcc, and GND, and outputs I0 through I7. The function is implemented by selecting the minterms S0, S1, and S2, which are marked with asterisks. On the right, the logic circuit is shown with three inverters for inputs A2, A1, and A0, and three 3-input AND gates for outputs Z1, Z0, and M.</p>

Tabla 3-8: Problemas tipo 4

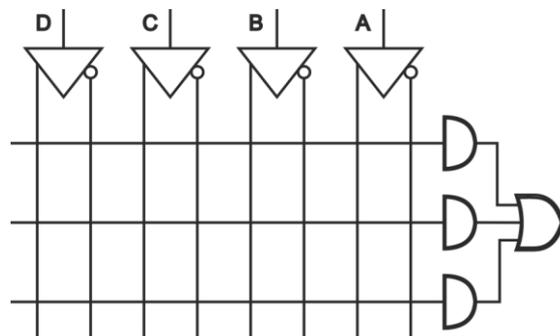
Concepto: Simplificación por Veitch-Karnaugh.

Problema nº: 7.

Tipo de Problema: Metodología, Síntesis.

Descripción: A partir de un mapa K, se debe realizar una simplificación del mismo y mapear el resultado en una PAL.

Figura:



	BA			
DC	00	01	11	10
00	0	1	1	0
01	0	0	1	1
11	1	0	0	1
10	0	1	1	0

Tabla 3-9: Problemas tipo 5

Concepto: Estructura de una PAL.

Problema nº: 7 y 8.

Tipo de Problema: Síntesis.

Descripción: Mapear el circuito en una PAL con salidas activo bajo. El estudiante debe realizar las conexiones utilizando la convención de estos dispositivos.

Figura:

The figure illustrates the implementation of a 2-bit multiplier in a PAL. It includes a truth table, a logic diagram, and a PAL schematic.

DC \ BA	00	01	11	10
00	0	1	1	0
01	0	0	1	1
11	1	0	0	1
10	0	1	1	0

The logic diagram shows four inputs (D, C, B, A) each passing through an inverter. The outputs of these inverters are connected to the inputs of three AND gates. The outputs of these AND gates are connected to the inputs of three OR gates, which produce the outputs Z1, Z0, and M.

The PAL schematic shows three AND gates (A2, A1, A0) and three OR gates (Z1, Z0, M). The connections are as follows:

- AND gate A2: Inputs D and C.
- AND gate A1: Inputs C and B.
- AND gate A0: Inputs B and A.
- OR gate Z1: Inputs from AND gates A2 and A1.
- OR gate Z0: Inputs from AND gates A1 and A0.
- OR gate M: Inputs from AND gates A2 and A0.

Tabla 3-10: Problemas tipo 6

Concepto: LUTs con MUX de menor tamaño.
Problema nº: 9.
Tipo de Problema: Metodología, Síntesis.
Descripción: Utilización de un multiplexor para realizar una función lógica que tiene más variables que entradas de selección del MUX. En tal caso, es necesario dividir la tabla y utilizar puertas adicionales.
Figura:

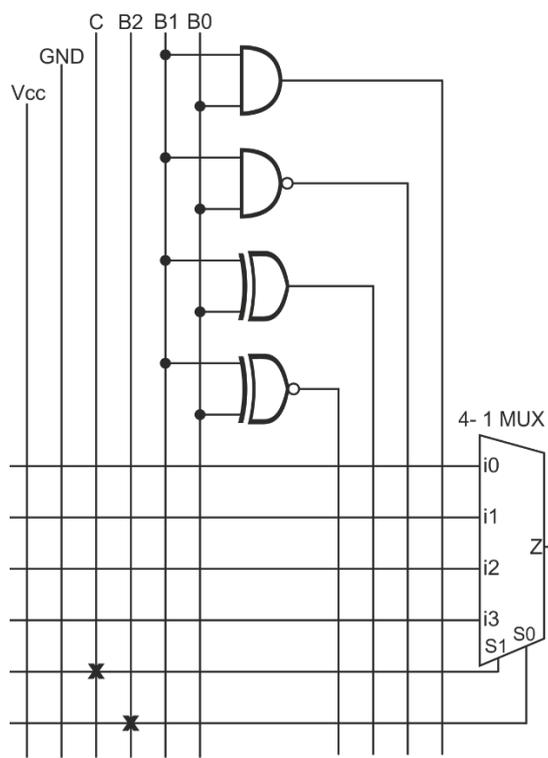


Tabla 3-11: Problemas tipo 7

Concepto: Codificadores.
Problema nº: 10.
Tipo de Problema: Análisis, Conocimientos básicos.
Descripción: Analizar un circuito ya diseñado y obtener su salida para una tabla de entrada de datos.
Figura:
<p style="text-align: center;"> priority encoder 4-2 (S3 with higher priority) </p> <p style="text-align: center;"> decoder 2-4 </p>

Tabla 3-12: Problemas tipo 8

Concepto: Código Gray.
Problema nº: 11.
Tipo de Problema: Conocimientos básicos.
Descripción: Verificar que se ha comprendido la idea del Código Gray.

Tabla 3-13: Problemas tipo 9

Concepto: LUTs a partir de Decodificadores.
Problema nº: 12.
Tipo de Problema: Síntesis.
Descripción: Utilización de decodificadores y puertas adicionales para generar una función lógica.
Figura:

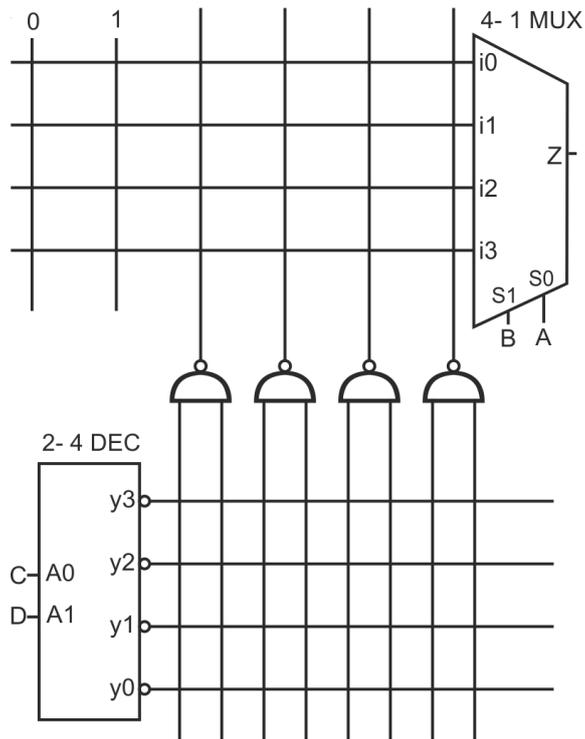


Tabla 3-14: Problemas tipo 10

4 DESARROLLO

En este apartado se va a describir el proyecto de una manera más detallada:

- Pasos previos.
- Herramientas necesarias.
- Esquema general de la aplicación.
- Conceptos clave.
- Partes importantes de la aplicación.
- Problemas y soluciones.

4.1 Primeros pasos

Antes de iniciar el desarrollo en sí de la aplicación, se han realizado una serie de tareas para poder llevar a cabo el proyecto.

- Revisión de algunos libros de JAVA.
- Estudio del ciclo de vida de una aplicación.
- Estudio genérico de la API de Android, para conocer sus características y funcionalidades principales.
- Realización de pequeñas aplicaciones previas para familiarizarse con el entorno y el otro lenguaje principal de Android, XML.

4.2 Herramientas necesarias para el desarrollo del proyecto

- Ordenador personal.
- SDK de Android.
- Simulador ADT Android.
- Eclipse.
- Diferentes dispositivos Android.
- Herramienta de diseño gráfico para las imágenes de los circuitos.

4.3 Esquema general de la aplicación

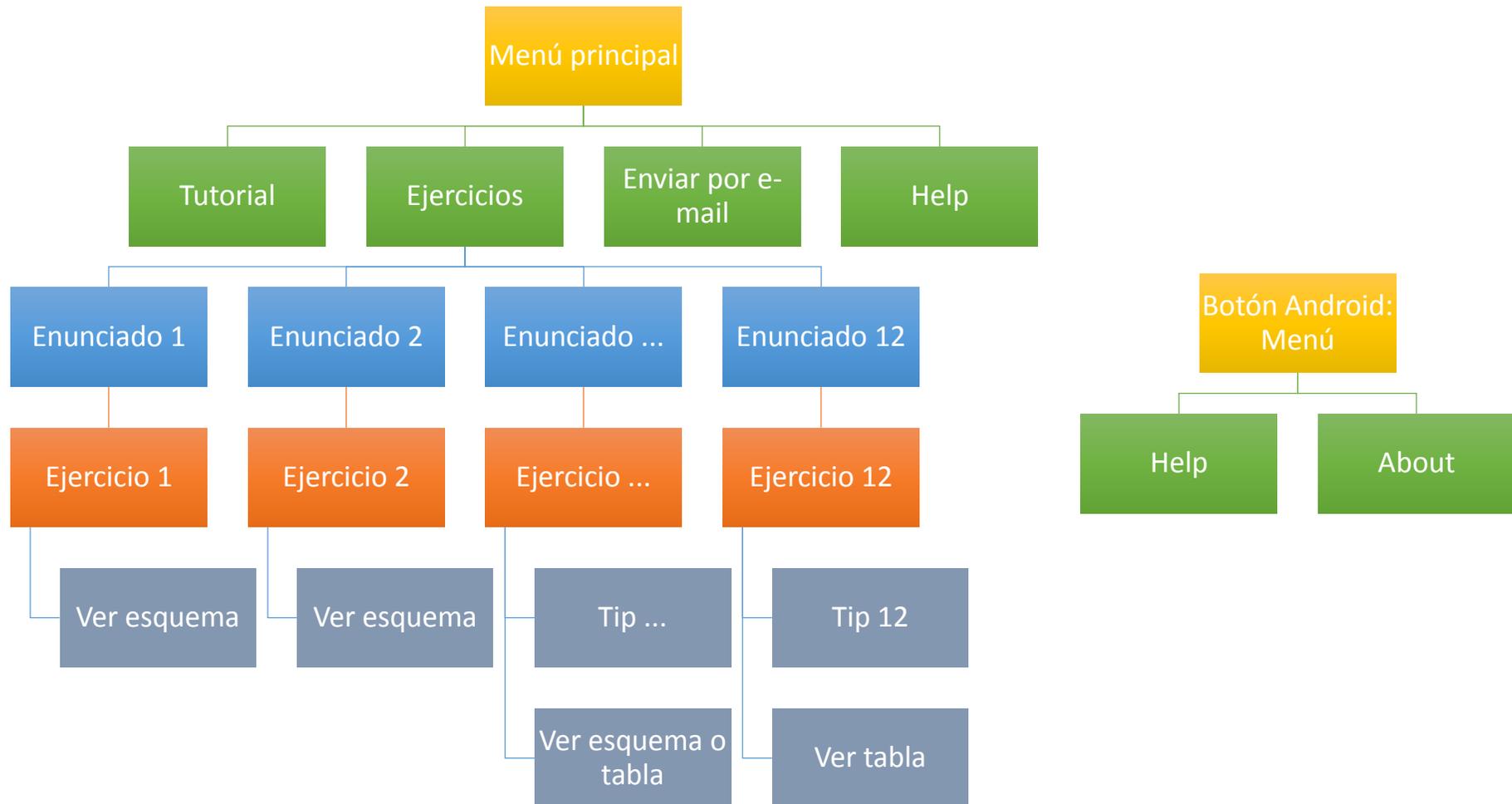


Figura 4-1: Esquema general de la aplicación

4.4 Conceptos clave

Para comprender mejor los siguientes apartados, se van a exponer brevemente una serie de conceptos que se van a utilizar posteriormente.

4.4.1 Actividad

Las aplicaciones están formadas por una o varias actividades.

Normalmente cada pantalla se desarrolla como una actividad (en un fichero JAVA). Una actividad es una subclase de la clase "Activity", la cual se usa para interactuar con el usuario y realizar las acciones implementadas.

Por tanto, cada vez que aparezca una nueva pantalla (normalmente tras pulsar un botón), lo más probable es que sea una actividad nueva. Más adelante se verán casos de este tipo.

4.4.2 Ciclo de vida de una actividad

Las actividades pasan por una serie de estados desde que se crean hasta que se finalizan:

- Activa: La actividad está en primer plano en pantalla.
- Pausa: La actividad sigue visible, pero hay algo delante de ella, por ejemplo una alerta.
- Parada: La actividad está guardada en la pila, con toda su información. Es posible que el sistema la elimine si necesita memoria.

Android se encarga de gestionar estos estados, pero es recomendable que nosotros lo hagamos en nuestra aplicación.

Para evitar problemas con las actividades, lo que vamos a hacer es no dejar ninguna actividad en la pila, es decir, sólo tendremos la que esté activa. Para realizar esto, hay que hacer varias modificaciones:

- Cada vez que lancemos una nueva actividad, finalizaremos la actividad actual, por lo que esta no se guardará en la pila.
- Gestionaremos el botón de Android "back": Este, por defecto activa la última actividad guardada en pila. En nuestro caso, se cerraría la aplicación ya que no habría ninguna actividad disponible. Por lo tanto, lo que hacemos es que al pulsar el botón "back", se lance la actividad a la que debería ir, además de cerrar la actual. Así, siempre tendremos controlado lo que va a pasar.

Hay una excepción para esto. Cuando estamos en alguna de las opciones que presenta el botón menú ("Help" y "About"), permitimos que se guarde en pila la actividad desde

la que se pulsa este botón, y así cuando pulsemos “back” (esta vez no lo gestionaremos nosotros), volveremos a donde estábamos.

4.4.3 Vista

Es todo lo que se muestra en pantalla. Botones, imágenes, texto, etc. Normalmente se diseña en un fichero aparte (fichero en lenguaje XML) dentro de la carpeta “layout” del proyecto. También se puede hacer desde un fichero JAVA, usando la clase “View”.

4.5 Menús de la aplicación

La aplicación está organizada en menús, a partir de ellos se puede acceder a todas las funcionalidades de la aplicación.

Los menús presentan una estructura muy simple, un conjunto de botones agrupados que al pulsarlos ejecutan una acción programada, normalmente lanzar una nueva actividad.

Tenemos disponibles tres menús:

- Menú principal: Desde donde se puede ver el tutorial, ir al menú de ejercicios, enviar ficheros (esto se hace desde la misma actividad del menú) e ir al menú de ayuda.
- Menú de ejercicios: A partir de él se puede acceder a cualquier ejercicio.
- Menú de ayuda: Muestra las distintas opciones de ayuda de la aplicación.

Los dos primeros simplemente lanzan una actividad por cada botón (excepto la de envío de ficheros), mientras que el menú de ayuda sólo lanza una actividad para todas las opciones. Esto se ha realizado así ya que lo único que hace la segunda actividad, es mostrar vistas según la opción elegida en el menú ayuda.

4.6 Gestión del botón Android: Menú

En las primeras versiones de Android, este botón siempre aparecía de forma física en los dispositivos, pero actualmente cada vez son menos los que lo incluyen. En cualquier caso, si no se dispone de él, se puede añadir de manera sencilla.

Para esta aplicación se ha implementado este botón, estando disponible en todas las pantallas.

Al pulsar el botón, se ofrecen dos opciones:

- La ayuda de la aplicación (“Help”).
- El “acerca de” (“About”).

Desde dentro de estas dos opciones, no se ha habilitado el botón menú.

4.7 El fichero manifiesto

Este fichero contiene información importante de la aplicación:

- Nombre del paquete JAVA.
- Memoria donde se descargará por defecto la aplicación. En este caso se ha dejado en modo automático, siendo el sistema el encargado de colocarla donde crea conveniente. También es posible cambiar el lugar de instalación a través del gestor de aplicaciones de Android.
- Permisos que necesita la aplicación para ejecutarse. Sólo se necesitan permisos de escritura en la memoria externa.
- Versión mínima y máxima de Android en la que se podrá ejecutar.
- Declaración de todas las actividades de la aplicación, y algunas características de ellas, como por ejemplo la orientación.
- Versión del código.

4.8 Diseño y desarrollo de ejercicios tipo tabla

Para llevar a cabo este ejercicio, hay que desarrollar dos partes completamente diferentes. La vista de la actividad y la propia actividad.

4.8.1 Vista

En este caso estará diseñada en lenguaje XML.

En primer lugar, hay que definir el tipo de vista principal, que será lineal y con orientación vertical, así los elementos sucesivos aparecen unos debajo de otros. Dentro de esta vista se añadirá un elemento tabla y dentro de este, varios de tipo fila hasta completar la tabla deseada.

Dentro de la vista de tipo fila, añadiremos los elementos de texto y botones necesarios para completar el ejercicio.

Se han usado dos tipos de botones:

- “Button”: Utilizado para activar las funcionalidades del ejercicio.

- “ToggleButton”: Utilizado para rellenar la tabla, ya que funciona como un biestable. Por defecto está a “0” y si pulsamos se pone a “1”, si se sigue pulsando seguirá cambiando entre los dos estados posibles.

A continuación se muestra la vista de un ejercicio de este tipo, explicando brevemente la colocación de los elementos que la forman:

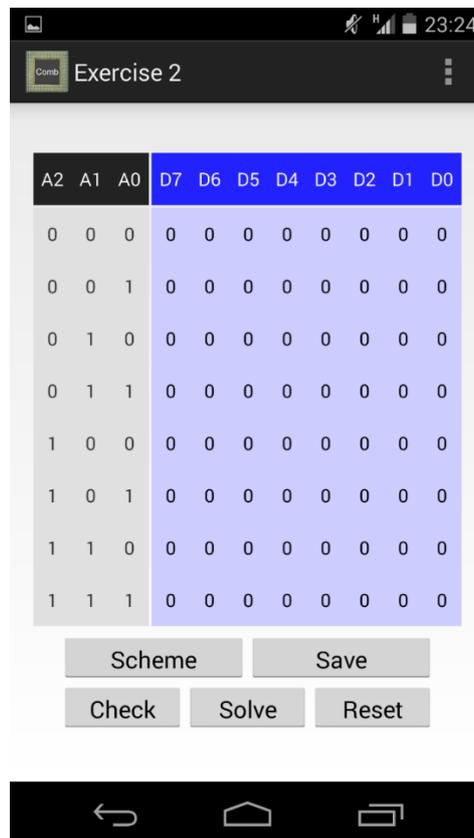


Figura 4-2: Vista ejercicio tipo tabla

Se han definido once filas en la vista:

- La primera es de texto, con el encabezado de la tabla.
- La ocho siguientes, representan los datos del ejercicio. En estas filas, los elementos en gris son texto, mientras que los de azul claro son botones de tipo “ToggleButton”.
- Las dos últimas filas se utilizan para colocar los botones de funcionalidades del ejercicio. Estos son de tipo “Button”.

4.8.2 Actividad

En la actividad del ejercicio hay que manejar los datos que va a ir introduciendo el usuario. Para ello se crean dos vectores, uno de tipo "ToggleButton" y otro "Boolean", su tamaño será del número de "ToggleButton" definidos en la vista.

Cada "ToggleButton" de la vista tiene un identificador. Este se usará en el código de la actividad para asignar un "ToggleButton" a cada elemento del primer vector creado.

Hay que llevar especial cuidado al manejar este vector, ya que si modificamos sus datos internamente, en la vista también se cambiarán. Para que esto no ocurra, se crea el segundo vector, con él se trabajará durante la actividad sin miedo a modificar la vista. Se ha elegido de tipo "Boolean", ya que es un tipo de dato similar al de los "ToggleButton" y es más fácil trabajar con él. Por ejemplo, para modificar uno de los "ToggleButton", hay que utilizar el método "setChecked()", y en el caso de datos "Boolean" se haría con una simple asignación.

Resumiendo:

- Primer vector: Contiene todos los "ToggleButton" definidos en la vista, si los modificamos, también modificamos la vista.
- Segundo vector: Contiene los datos del primer vector en un formato más cómodo de manejar.

Ahora se crea una función para que cada vez que el usuario modifique un valor en la tabla, este quede guardado. Además, estos datos se guardarán también en el archivo de preferencias, más adelante se comentará esto con más detalle.

Ya están disponibles todos los datos en la actividad, por lo que se pueden ejecutar las funcionalidades que ofrece este tipo de ejercicios:

- Resetear: Usando el método "setChecked()", se pondrán todos los "ToggleButton" de la vista a "0". También se guardarán los datos en el vector "Boolean" y en preferencias.
- Solución: Se muestra la solución en la vista. Se realiza de la misma forma que resetear, pero poniendo los valores correctos del problema.
- Esquema: Se muestra el esquema del enunciado.
- Guardar: Se guardan los datos actuales del ejercicio en un fichero. Usando datos tipo "Boolean".
- Comprobar: Se realizan comprobaciones totales y parciales del ejercicio. Usando datos tipo "Boolean".

Más adelante se profundizará más en las dos últimas funcionalidades.

4.9 Adaptación a diferentes pantallas

Para que la aplicación se pueda utilizar de forma adecuada en dispositivos con tamaños y densidades diferentes, es necesario adaptar las vistas.

Se han utilizado dos tipos de vistas que se explican a continuación.

4.9.1 Adaptación de vistas en XML

Todas las vistas de la aplicación son de este tipo, exceptuando las de resolución de ejercicios de conexiones y las del tutorial.

Todas las vistas son elaboradas mediante el lenguaje XML, como se había comentado antes.

Para comprobar la visualización de este tipo de vistas en diferentes terminales, el kit de desarrollador ofrece una herramienta bastante útil, el “Graphical layout”. Con ella, se pueden definir pantallas de cualquier tamaño y resolución y muestra gráficamente como quedaría la vista en las pantallas que elijamos.

Es complicado hacer una vista, que se vea bien en pantallas grandes y pequeñas. Por esto, Android permite definir carpetas de vistas, para cada escalón de tamaño de pantalla o de densidad.

En esta aplicación se han definido vistas para los cuatro tamaños de pantalla que maneja Android, es decir, se ha creado cada vista cuatro veces, guardando cada una en las carpetas correspondientes a sus tamaños:

- “layout-small”.
- “layout” (para el tamaño normal).
- “layout-large”.
- “layout-xlarge”.

Con esto se consigue visualizar la aplicación correctamente en todo tipo de dispositivos.

4.9.2 Adaptación de vistas en la actividad

Se han realizado este tipo de vistas en los ejercicios de conexiones y en el tutorial.

En este caso la vista y la adaptación se realizan en la actividad.

Dentro de la clase “Activity”, se crea una nueva de tipo “View” donde se creará la vista.

Para poder elaborar la vista, se usará el método “OnDraw”, para el que se define un lienzo, disponible con la clase “canvas”.

A través del lienzo, se pueden dibujar líneas, imágenes (con la función “DrawBitmap”), texto, en definitiva, casi cualquier cosa.

Para introducir algo, primero hay que indicar donde se quiere dibujar. Android nos permite hacer esto mediante coordenadas. Cada pantalla, tiene un número diferente de coordenadas acorde a su resolución, por lo que primero hay que hallar la resolución del dispositivo.

Usando un objeto “DisplayMetrics” se obtiene la resolución del dispositivo donde se está ejecutando la aplicación, y con este dato ya sabemos los límites de coordenadas que podemos usar.

Ahora se procede a adaptar la vista a todas las pantallas de una manera automática. Teniendo la resolución, sabemos el número total de píxeles en el eje X y en el eje Y de cada pantalla. Las coordenadas se darán principalmente en función del ancho en píxeles de cada pantalla, ya que esta, siempre es más estrecha que alta.

A continuación se muestra una vista adaptada, la cual se va a explicar con un poco más de detalle.

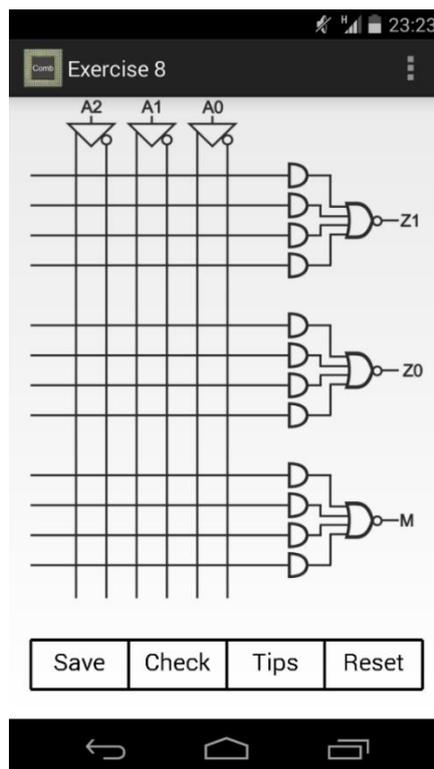


Figura 4-3: Adaptación de vistas en la actividad

La pantalla anterior tiene una resolución de 1080x1920 píxeles:

- Ancho = 1080 píxeles.
- Alto = 1920 píxeles.

La relación alto/ancho es, 1.778. Cuanto más cercano sea este valor a 1, más cuadrada será la pantalla.

Es importante conocer la relación alto/ancho de la pantalla del dispositivo, ya que debemos insertar imágenes en la vista manteniendo también su relación alto/ancho, y si esta relación es mayor en la imagen, esta ocupará más que toda la pantalla, a no ser que se muestre reducida. También hay que tener en cuenta la barra de estado de Android y la del título de la actividad, que hacen que la superficie disponible para la vista tenga una relación alto/ancho un poco menor que la de la pantalla. Además, debajo de la imagen se dibujan los botones de funcionalidades de los ejercicios, esto hace que la vista completa (imagen más botones) tenga una relación alto/ancho mayor que la de la imagen sola.

En definitiva, para que la vista ocupe aproximadamente toda la pantalla, hay que tener en cuenta la relación alto/ancho de la imagen que se inserta, el tamaño de los botones que van debajo y la barra de estado Android y la de título de la actividad.

Para que se vea todo de una forma óptima, se diseñarán las imágenes con una relación alto/ancho entre 1.4 y 1.6, ya que la mayoría de dispositivos tienen pantallas con relaciones alto/ancho mayores.

Ahora que están claras todas las medidas de diseño de este tipo de vista, se pasa a insertar los circuitos en forma de mapa de bits escalado, siempre manteniendo sus proporciones. En esta vista los botones están dibujados con líneas, en el punto de desarrollo de ejercicios de conexiones se explicará en más detalle este tipo de botones.

Tanto la imagen como los botones tienen un margen del 5 % con los lados de la pantalla, para que la visualización sea correcta.

Tras realizar la adaptación se observaron problemas en pantallas más cuadradas de lo habitual. Para estos casos se optó por desarrollar una medida extra, consiste en aumentar los márgenes de la vista, consiguiendo una reducción de la imagen y de los botones, esta medida se ha tomado para dos escalas de relación alto/ancho. La primera para relaciones de pantalla menores de 1.4, donde los márgenes son de un 15% a cada lado y la segunda entre 1.4 y 1.52, siendo los márgenes del 10 %.

Más adelante se probó el funcionamiento en dispositivos sin teclas físicas, como por ejemplo el de la figura anterior. En ellos, se integran los botones en la pantalla, ocupando la parte más inferior de la vista y por consiguiente dificultando en algunos casos la visión de parte de los botones.

Después de la observación anterior, se definió la vista completa de tipo "Scroll". De esta forma se permite realizar desplazamientos en la pantalla en caso de no obtener una visión completa de imagen más botones. Con esto se solucionaría también la medida extra explicada anteriormente, aunque se optó por dejarla, ya que permitía visualizar mejor los circuitos.

Se ha explicado el proceso de adaptación de pantalla centrado en los ejercicios, para el caso del tutorial, la adaptación también es de este tipo, aunque más sencilla, sólo será necesario insertar las imágenes sin ningún botón.

4.10 Tutorial

El tutorial consiste en una serie de imágenes que se van mostrando en la misma actividad. La idea de esta funcionalidad es la de simular un pequeño libro electrónico.

Para llevarlo a cabo se crea una vista de tipo JAVA, como la explicada anteriormente.

Se registran los identificadores de todas las imágenes del tutorial, cargando en la vista solamente una de ellas. La imagen es elegida según marque un contador que se va moviendo en función del avance o retroceso de las páginas del tutorial.

Para poder movernos por las páginas, se han definido dos regiones de toque. Una para retroceder, que contiene el primer 30% de la pantalla empezando por la izquierda y otra para avanzar, que contiene el último 30% de esta.

A continuación se muestra un esquema de la situación anterior:

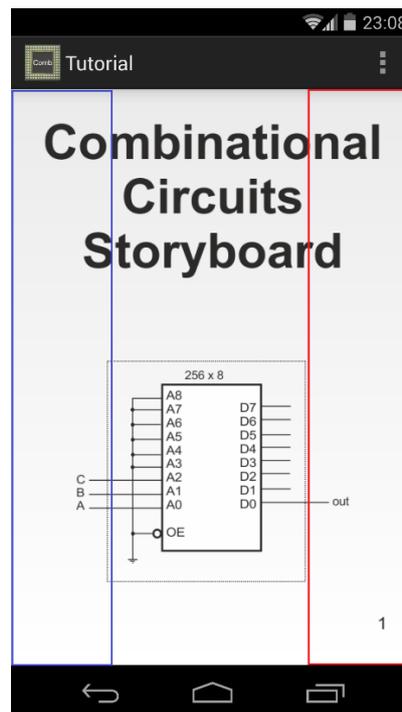


Figura 4-4: Tutorial

Si se pulsa en la región recuadrada en rojo, se avanzará una página, y si se hace en la región azul, se retrocederá.

Cuando se llega al final del tutorial, si seguimos avanzando continúa por la primera página.

4.11 Diseño y desarrollo de ejercicios de conexiones en circuitos

En primer lugar se han diseñado los circuitos de manera que las conexiones sean fáciles de localizar y estén separadas por distancias iguales o proporcionales.

La herramienta de diseño gráfico mide las distancias en centímetros y en la actividad del ejercicio nos movemos por la pantalla con píxeles. Por lo tanto, hay que hallar una relación entre píxeles y centímetros.

Sabemos el ancho en centímetros de la imagen, ya que la hemos diseñado nosotros, sabemos los píxeles que tiene el ancho de la pantalla, y sabemos el ancho en píxeles que va a ocupar la imagen (generalmente el 90 % del ancho en píxeles de la pantalla, ver apartado 4.9.2). Con eso podemos sacar una relación directa entre píxeles y centímetros:

$$\text{Relación_px_cm} = \frac{0.9 * \text{ancho_pantalla_px}}{\text{ancho_imagen_cm}}$$

Con la relación anterior, sabiendo la distancia entre conexiones en centímetros, hallamos la distancia entre conexiones en píxeles:

$$\text{Distancia_entre_conexiones_px} = \text{Distancia_entre_conexiones_cm} * \text{Relación_px_cm}$$

Con los datos anteriores, y una vez se localiza la primera conexión de arriba a la izquierda, nos vamos moviendo por la imagen mediante un bucle, guardando las coordenadas de todas las conexiones. Tendremos una coordenada en el eje X y otra en Y por cada conexión.

Ahora ya es posible dibujar conexiones sobre la imagen, el paso siguiente es poder reconocer las pulsaciones que hace el usuario.

Para esto se usará el método "onTouchEvent". Cada vez que se pulse la pantalla se entra en esta función y mediante un objeto de tipo "MotionEvent", podemos obtener las coordenadas de cada pulsación. Como es prácticamente imposible que se pulse el punto exacto de las conexiones, se han definido pequeñas áreas alrededor de estas. Quedan de la siguiente manera:

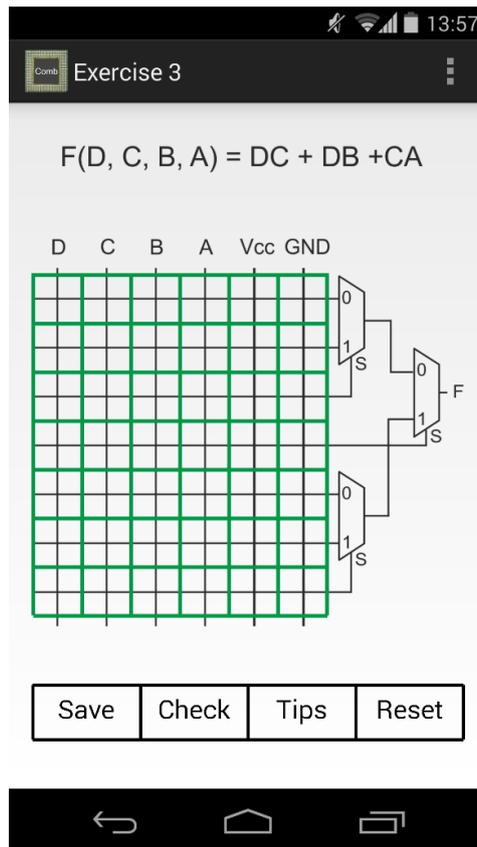


Figura 4-5: Regiones de toque para conexiones

Cada cuadrado resaltado en verde, representa un área activa de toque que se corresponde con la conexión que hay en el centro de la misma. Por tanto, cada vez que se pulse dentro de una de estas áreas, se activará o desactivará una conexión.

Las conexiones activas se marcarán con un punto azul. Para realizar esto, hay que usar la función “invalidate()” dentro de “onTouchEvent”, que pasa a ejecutar el método de dibujo “onDraw” donde ya se podrán dibujar las conexiones. A continuación se dibuja un círculo en las coordenadas de la pulsación correspondiente.

Cada vez que el usuario pulse una de las áreas activas, se guardan en un vector todos los estados de las conexiones actualizados y también en las preferencias, así, si se sale de la aplicación no se perderán los datos actuales.

Los datos del vector anterior, son los que se guardan también en fichero de soluciones, el cual se puede cargar o enviar para su uso en otros dispositivos.

Por otro lado, están los botones de funcionalidades de ejercicios. En este tipo de vistas se ha optado por dibujarlos y cada uno de ellos comprende un área de toque como las descritas anteriormente. Al pulsar uno de estos botones, se ejecutan las mismas acciones que para un botón de tipo “Button”.

4.12 Consejos para la resolución de ejercicios

Esta herramienta sólo está disponible para los ejercicios de conexiones. Se muestra de la misma forma que la vista de resolución del ejercicio, cambiando únicamente las funciones que se realizan al pulsar los botones.

Por cada ejercicio de conexiones hay una actividad de consejos. Sólo hay un ejercicio que no cumple esto, y la razón es que sólo dispone de un consejo de alerta y este se integra en la actividad de la resolución.

Se podrían haber integrado todos los consejos en las actividades de los ejercicios, pero esto sobrecargaba la vista. Al final se ha optado por hacerlo en actividades separadas, ya que también queda de una manera más clara.

En la actividad de consejo principalmente se muestran alertas con indicaciones teóricas o se dibuja alguna conexión utilizando la clase "canvas".

Al contrario que en el ejercicio, en las actividades de consejos no se pueden realizar conexiones. Las únicas regiones de toque son las de los botones.

En este tipo de actividad se muestra información organizada en varias partes, a las que se puede acceder a través de diferentes botones.

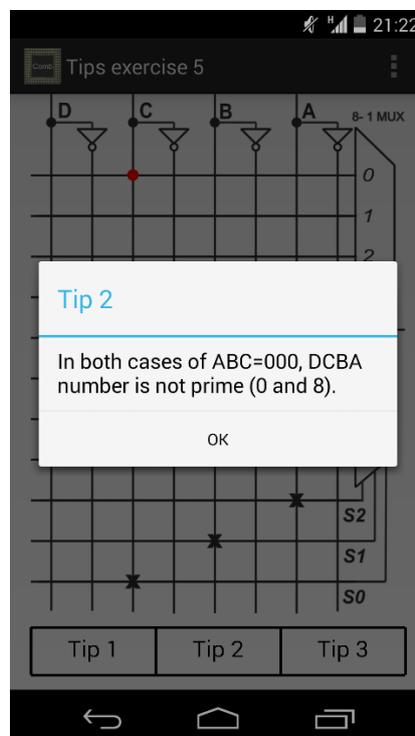


Figura 4-6: Ayuda a la resolución de ejercicios

4.13 Gestión de datos de la aplicación

En este apartado se detallará cómo guardan y cómo comparten datos las actividades. Además de cómo se realiza el manejo de ficheros de resultados.

4.13.1 Método de preferencias

El método elegido para gestionar los datos es el de preferencias.

Las preferencias consisten en uno o más archivos XML, en los que las actividades pueden guardar información.

Son muy útiles para compartir datos entre actividades, ya que todas tienen acceso a estos archivos.

Cada actividad que use preferencias, debe declarar un objeto de tipo "SharedPreferences" e indicar cuál es el nombre del archivo de preferencias.

Para introducir datos en el archivo de preferencias, es necesario declarar un editor de preferencias. En él se indicará el tipo de dato que se guarda y el valor de este.

Para cargar datos desde el archivo de preferencias hay que usar un método del objeto "SharedPreferences" declarado.

En esta aplicación se usan las preferencias en dos tipos de actividades:

- Actividades de enunciados, donde se guarda únicamente el dato número de ejercicio. Esta información se utilizará en la actividad que se encargará de cargar o borrar ficheros para poder mostrar los archivos del ejercicio deseado.
- Actividades de ejercicios, donde se guardan en preferencias todos los datos introducidos por el usuario. Lo primero que se hace al lanzar la actividad, es cargar las preferencias, por lo que aunque el usuario no guarde sus soluciones, si este no resetea, las seguirá teniendo cuando vuelva al ejercicio. Si no hubiera ningún dato guardado en las preferencias, se cargarían los datos por defecto de la actividad.

Por tanto, todos los datos que introduce el usuario están guardados en preferencias.

4.13.2 Guardar soluciones

Esta actividad se puede lanzar desde cualquier pantalla de resolución de ejercicios, y sirve para guardar un archivo de soluciones del usuario.

En ella están definidas todos los vectores que se usan para guardar los datos de cada ejercicio.

La vista de esta actividad se muestra en forma de diálogo. En este, se pide rellenar el nombre del archivo donde se va a guardar la información del ejercicio. Además de esto, al nombre del archivo se le añade una cabecera que contiene el número de ejercicio y un indicador del tipo de memoria donde se va a guardar. Si se va a guardar en memoria externa EM, y si fuera en memoria interna IM.

Cuando hay memoria externa disponible, el fichero de soluciones se guarda en la carpeta “CombinationalCircuits” en la raíz de esta memoria. La memoria externa no tiene porqué ser necesariamente una tarjeta SD, ya que casi todos los móviles de última generación habilitan una parte de la memoria interna para que esté disponible para el usuario, que es a la memoria que se le llama externa.

En caso de no tener memoria externa disponible en el dispositivo, se guardará el fichero en una zona de la memoria interna reservada para datos de la aplicación.

Para extraer todos los datos que se quieren guardar, se utiliza el método de preferencias comentado anteriormente. Se guardan los datos en un vector y posteriormente son escritos en un fichero de una forma específica, que más adelante permitirá leerlos fácilmente.

4.13.3 Cargar o borrar soluciones

A esta actividad se accede desde el enunciado de cualquier ejercicio. En ella se muestran todas las soluciones guardadas para el ejercicio en cuestión y se permite cargar o borrar cualquier fichero mostrado.

Cuando se seleccionan uno o más ficheros y se pulsa el botón de borrar, se muestra una alerta, si se acepta, los ficheros quedarán eliminados y no se podrán recuperar.

Para cargar un ejercicio, solamente hay que seleccionar un fichero. Al pulsar el botón de cargar, se lanzará la actividad del ejercicio con los datos del fichero cargados en esta.

La idea anterior se realiza de la siguiente manera. Primero hay que leer el fichero, almacenar sus datos en un vector y posteriormente guardar estos datos en las preferencias del ejercicio correspondiente. Como las actividades de cada ejercicio cargan las preferencias al iniciarse, los datos están disponibles automáticamente.

Al cargar cualquier solución de un ejercicio, se sobrescriben los datos introducidos en la última sesión del mismo. Esto es debido a que esos datos están guardados en el archivo de preferencias, y ahora se guardan ahí los datos que se han cargado del fichero de soluciones.

4.14 Sistema de comprobación de ejercicios

En este apartado se van a explicar los distintos métodos de comprobación de soluciones.

4.14.1 Comprobación en ejercicios tipo tabla

Para este tipo de ejercicios la corrección normalmente es más simple, ya que las tablas tienen solución única. Aun así, hay diferencias en las comprobaciones.

Para los ejercicios de memoria ROM la comprobación se realiza por filas. Los datos de cada fila se guardan en un vector y se comparan con el vector correcto. Posteriormente, se muestran por pantalla los números de filas incorrectas o el mensaje de acierto.

En el ejercicio de codificador y decodificador, se indican los números de tabla donde hay errores, si no los hay, se muestra el mensaje de acierto.

Para el problema del código Gray, sólo se indicará si la solución introducida es correcta o incorrecta, sin dar pistas de donde está el fallo.

4.14.2 Comprobación en ejercicios de conexiones

Para el caso de los ejercicios de conexiones la comprobación es más tediosa, ya que generalmente los ejercicios tienen múltiples soluciones y hay que abarcar todas.

Primero hay que agrupar los datos introducidos de una manera determinada para poder facilitar las comprobaciones de las diferentes soluciones posibles. Por ejemplo, siempre tenemos un vector de pulsaciones donde se registran todas las conexiones, según sea el circuito, habrá que descomponer este vector en otros más pequeños que contengan los datos de conexiones de diferentes zonas del circuito.

Vamos a ver un ejemplo, de un problema en concreto:

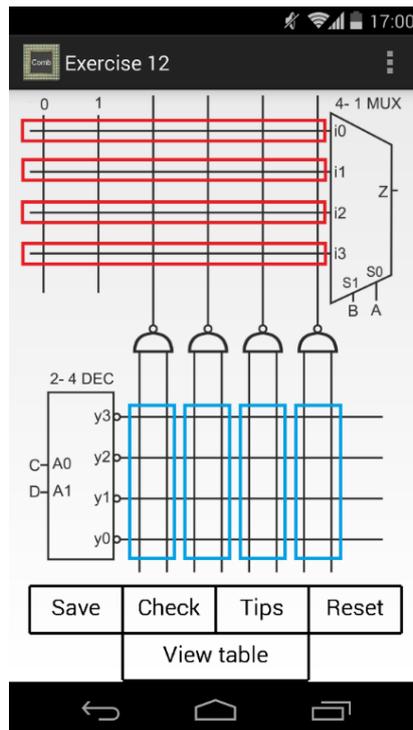


Figura 4-7: Agrupación de datos para comprobación de resultados

Para este circuito se han agrupado los datos de dos maneras diferentes:

1. La parte de arriba está dividida en cuatro vectores, cada uno con seis valores que representan cada una de las conexiones que hay dentro de su recuadro rojo. Cada vector se corresponde con una entrada del multiplexor.
2. La parte de abajo también está dividida en cuatro vectores resaltados en azul. Estos contienen las ocho conexiones que entran a las puertas NAND.

La primera entrada del multiplexor, “i0”, debe estar conectada a “0”. Las otras tres entradas se pueden conectar a cualquiera de las salidas de las puertas NAND. Por tanto, deberá quedar una puerta lógica sin conectar a ninguna entrada. Se deja total libertad de elección al usuario, es decir, puede utilizar las puertas que quiera.

Para comprobar lo anterior, se definen cinco vectores:

- El primero con la conexión a “0 ” habilitada, es decir, el primer valor del vector. Este sólo comprobará la primera entrada del multiplexor.
- Los otros cuatro, tendrán solo una pulsación habilitada, que se corresponderá con una de las cuatro puertas lógicas.

Una vez verificado que la primera entrada del multiplexor está a “0” y las otras tres conectadas cada una a una NAND diferente, pasamos a la parte de abajo.

Necesitaremos tres vectores de soluciones válidas, ya que solo se usan tres puertas NAND. En la puerta que no se usa, no importa qué conexiones estén habilitadas ya que no debe estar conectada a ninguna entrada del multiplexor, y si lo está, dará error en la comprobación de la parte de arriba.

Los tres vectores del párrafo anterior en realidad deben ser seis, ya que el usuario puede intercambiar las conexiones que entran a cada una de las entradas de cada puerta NAND y la solución seguiría siendo válida.

Veamos esto último en una figura aclarativa:

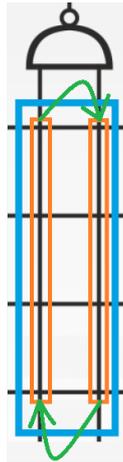


Figura 4-8: Intercambio de conexiones en las entradas

Para terminar, si las tres NAND, tienen cada una a su entrada el vector correcto, correspondiente con la entrada del multiplexor a la que están conectadas, la parte de abajo del circuito será también correcta.

4.15 Envío de soluciones

Esta opción está disponible en el menú principal de la aplicación.

Esta funcionalidad está implementada en la actividad del menú principal, ya que consiste en una alerta que muestra en una lista todos los ficheros de soluciones guardados por el usuario.

En este caso, se pueden seleccionar todos los ejercicios que se crea conveniente, y tras pulsar enviar, se mostrará una alerta con los gestores de correo disponibles.

Cuando se ha seleccionado el gestor de correo, la actividad hace lo siguiente:

- Rellena el asunto del correo.
- Adjunta todos los ficheros seleccionados anteriormente.
- Añade las indicaciones necesarias para que el receptor pueda utilizar estos ficheros.

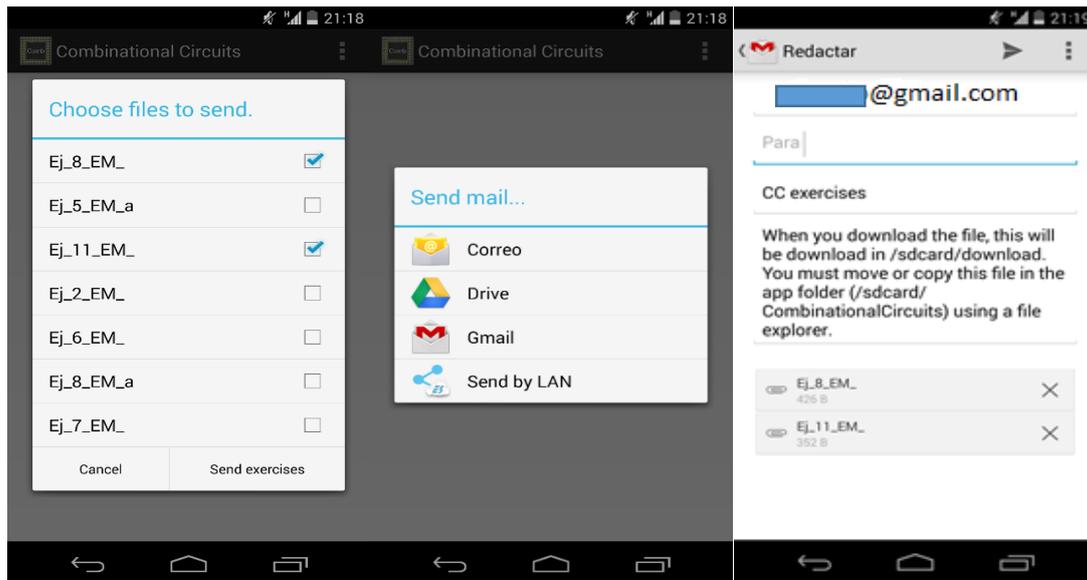


Figura 4-9: Enviar archivos

El usuario sólo deberá rellenar el campo de destinatario.

Las indicaciones para el receptor del correo son las siguientes:

- Descargar los ficheros adjuntos. Estos normalmente se guardan por defecto en el directorio “sdcard/download”.
- Debe tener instalada la aplicación y haber guardado alguna vez un fichero.
- Con un gestor de archivos, vaya al directorio anterior y copie o mueva los ficheros descargados a la carpeta de la aplicación, “CombinationalCircuits”.

Después de realizar estos pasos, cuando se vaya a la opción de cargar ejercicios, ya estarán disponibles estos ficheros de soluciones.

4.16 Diagrama de clases de la aplicación

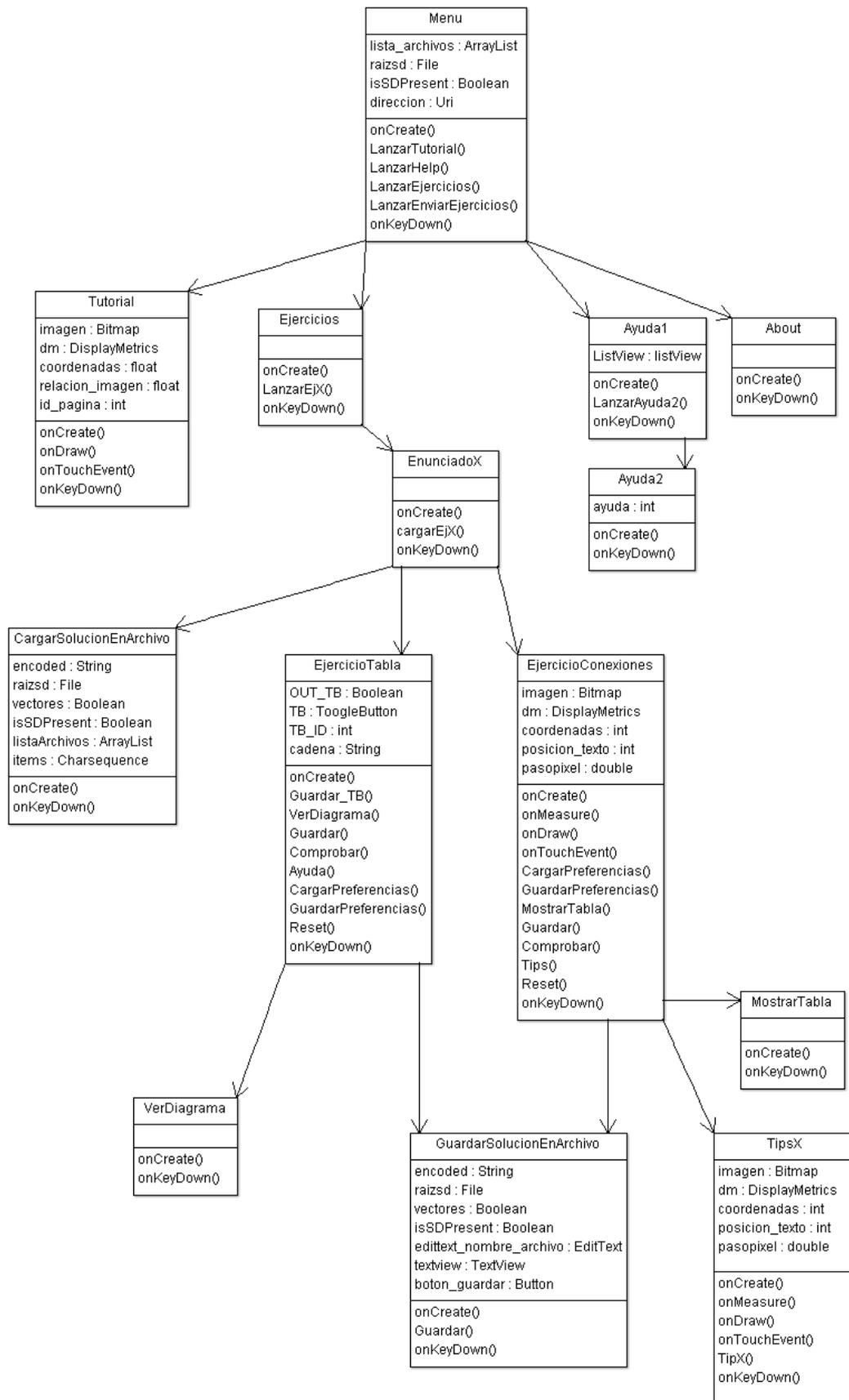


Figura 4-10: Diagrama de clases

4.16.1 Descripción de clases

Clases	Descripción
Menu	Ofrece el menú principal y en ella se desarrolla el envío de soluciones por correo.
Tutorial	Se encarga de gestionar el avance y retroceso de páginas del tutorial y de su correcta visualización.
Ejercicios	Menú de ejercicios. Desde aquí se lanzan los enunciados.
About	Muestra el acerca de de la aplicación.
Ayuda1	Menú con las ayudas disponibles.
Ayuda2	Se encarga de la visualización de la ayuda seleccionada.
EnunciadoX	Muestra el enunciado de los ejercicios y a partir de aquí, se permite acceder al ejercicio o a los ficheros de soluciones.
EjercicioTablaX	Clase para la resolución de ejercicios de tipo tabla, donde están implementadas todas las funcionalidades disponibles en este.
EjercicioConexionesX	Clase para la resolución de ejercicios de tipo conexión, donde están implementadas todas las funcionalidades disponibles en este.
CargarSolucionEnArchivo	Se encarga de cargar las soluciones guardadas en ficheros en la actividad del ejercicio correspondiente.
GuardarSoluciónEnArchivo	Se encarga de guardar el resultado introducido por el usuario en un fichero.
MostrarTabla	Muestra una tabla de utilidad para la resolución de ejercicios.
VerDiagrama	Muestra un diagrama de utilidad para la resolución de ejercicios.
TipsX	En ella están implementados los consejos, alertas y dibujos para ayudar al usuario a resolver determinados ejercicios.

Tabla 4-1: Descripción de clases

4.16.2 Descripción de métodos principales

Se van a comentar los principales métodos usados y desarrollados en la aplicación:

Método	Descripción
onCreate()	Es invocado cuando se inicia la actividad, y en él se desarrollan todas las opciones que tienen que estar disponibles en el inicio de la actividad actual. Por ejemplo, desde este método siempre se carga la vista de la actividad.
LanzarEnviarEjercicios()	Se encarga de mostrar los archivos disponibles en un mensaje tipo alerta y realizar el posterior envío por correo.
onDraw()	Método para diseñar la vista.
onTouchEvent()	Detecta eventos de toque en la pantalla del dispositivo móvil y realiza las acciones pertinentes según el área en el que hallen estas pulsaciones.
onMeasure()	Establece unas medidas determinadas para la vista cuando esta es de tipo "ScrollView".
CargarPreferencias()	Carga en el ejercicio los datos del fichero de preferencias.
GuardarPreferencias()	Guarda los datos del ejercicio en el fichero de preferencias.
Ayuda()	Carga la solución en ejercicios tipo tabla.
Guardar_TB()	Guarda las pulsaciones en los "ToggleButton".
Comprobar()	Realiza la comprobación de resultados.
Guardar()	Guarda los datos que se van introduciendo.
Reset()	Se encarga de resetear los datos del ejercicio actual.
TipX()	Realiza el consejo seleccionado.
onKeyDown()	Aquí se realiza la gestión del botón "back" para el salto entre actividades.

Tabla 4-2: Descripción de métodos

Los métodos del diagrama de clases que no son descritos en este apartado tienen la única función de lanzar una actividad determinada.

5 INTEGRACIÓN, PRUEBAS Y RESULTADOS

5.1 Primera publicación en Google Play Store

Cuando la aplicación estaba finalizada y su funcionamiento comprobado, se realizó la publicación de esta en la tienda de aplicaciones de Android.

Crear cuenta de desarrollador

En primer lugar, es necesario darse de alta como desarrollador de Android, para lo que hay que abonar 25\$, rellenar diferentes datos y elegir un nombre de desarrollador. En este caso la cuenta pertenece al laboratorio donde se ha desarrollado este proyecto, y esta será utilizada para publicar otros proyectos de este tipo, ya que no caduca. El nombre de desarrollador es “DSLlab UAM”.

Añadir una nueva aplicación

Para publicar una aplicación hay que realizar los siguientes pasos:

- Rellenar la ficha de la aplicación:
 - Título: Combinational Circuits.
 - Descripción del contenido y funcionalidades principales.
 - Idioma: Inglés de Estados Unidos. La ficha también se ha realizado en castellano.
 - Capturas de pantallas de diferente tamaño.
 - Icono de la aplicación en formato 500x500 píxeles.
 - Categoría: Educación.
 - Datos de contacto:
 - Correo: dslab.uam@gmail.com
 - Página web: http://arantxa.ii.uam.es/~euroform_dslab/android.htm
 - Política de privacidad.
- Precio y distribución: La aplicación se podrá adquirir de manera gratuita. Ha sido distribuida a todos los países disponibles, según indica la plataforma estos son 138 más resto del mundo.
- Subir el archivo “.apk”: Este es un fichero comprimido que contiene la aplicación y a partir de él se realiza la instalación de la misma.

La aplicación ya está disponible en el siguiente enlace o a través de la aplicación Google Play Store:

<https://play.google.com/store/apps/details?id=com.CombinationalCircuits>

5.2 Publicaciones posteriores

Aunque la aplicación había sido probada concienzudamente, siempre hay cosas que mejorar y pequeños errores que corregir, por lo que después de la primera subida, se realizaron distintas actualizaciones.

Para publicar una nueva actualización, es necesario aumentar el número de la versión de código en el fichero de manifiesto, si no, la plataforma de subida no aceptará el fichero “.apk”.

A continuación se muestra una tabla con todas las versiones:

Versión de código	Fecha de actualización	Estado actual
1	16-12-2013	No publicado
2	16-12-2013	No publicado
3	20-12-2013	No publicado
4	23-12-2013	No publicado
5	23-12-2013	No publicado
6	7-02-2014	No publicado
7	8-02-2014	En producción

Tabla 5-1: Versiones de la aplicación

Las versiones publicadas en 2013, corrigieron pequeños errores, sobre todo de visualización y de texto. En la versión 6, se añadió el tutorial sobre circuitos combinatoriales, y en la versión 7 se mejoró la visualización de este en algunos dispositivos en los que daba problemas.

La versión 7, publicada el 8-02-2014, es la versión final.

5.3 Estadísticas proporcionadas por Google Play Store

Cuando ya tenemos disponible nuestra aplicación en Google Play Store, la propia consola de desarrollador nos ofrece gran cantidad de información, y con actualizaciones diarias.

Instalaciones totales

En la siguiente figura, se van a mostrar las descargas totales que ha tenido la aplicación hasta el 11-02-2014.

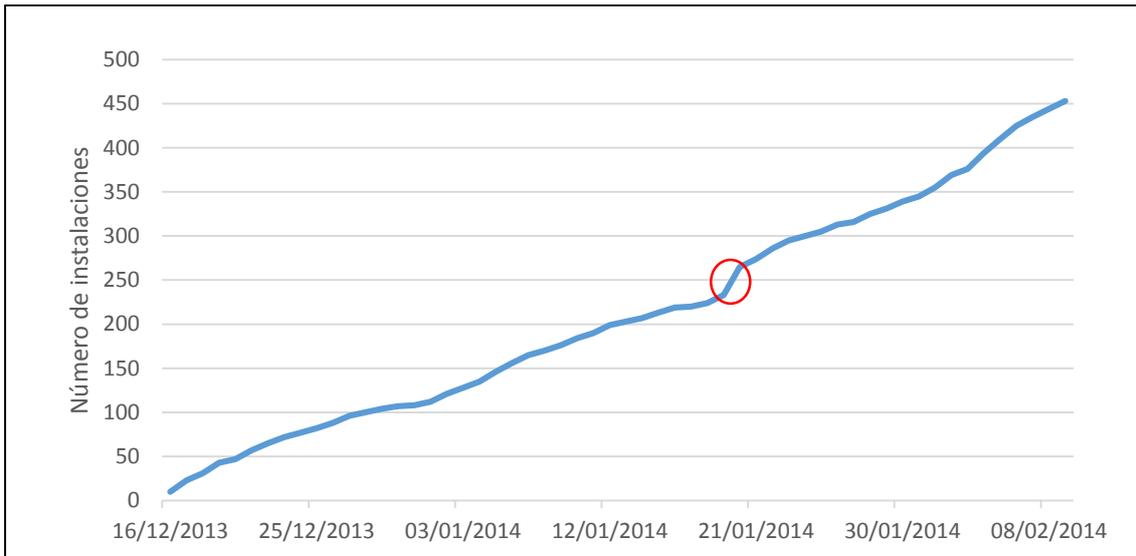


Figura 5-1: Evolución de instalaciones totales hasta el 11-02-2014

Aproximadamente el número de descargas diarias ha sido constante, de ahí la linealidad de la figura anterior. Se puede apreciar un aumento más elevado (resaltado en rojo), que se corresponde con la fecha donde ha comenzado la asignatura para la cual está destinada esta aplicación.

Instalaciones activas

Ahora se muestran el número de aplicaciones que hay instaladas en los dispositivos, es decir, las aplicaciones que están en uso.

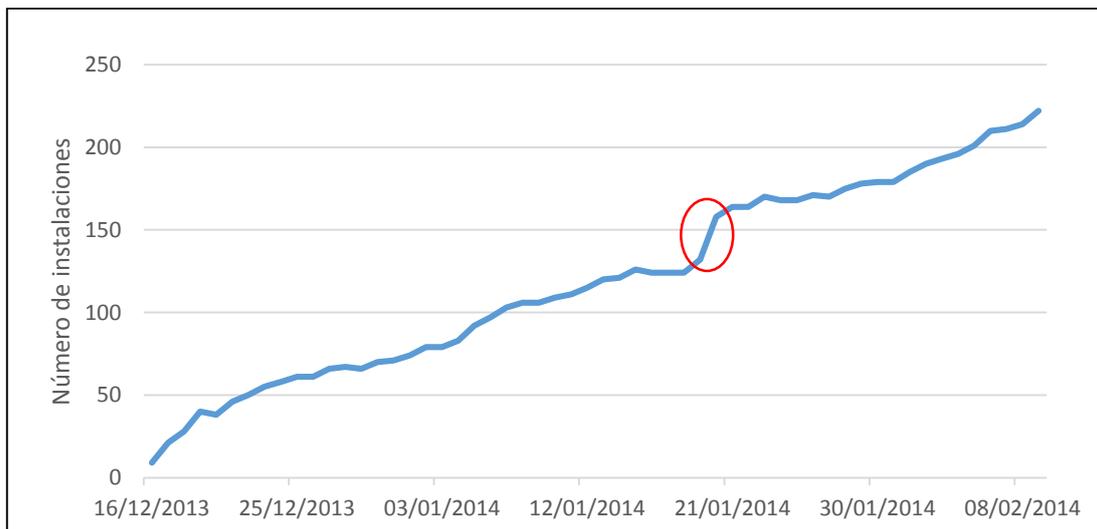


Figura 5-2: Evolución de instalaciones activas hasta el 11-02-2014

Se puede apreciar la subida de instalaciones debida al mismo hecho que en la gráfica anterior. Como es lógico, los valores de descargas en esta figura son menores, ya que muchos dispositivos han desinstalado la aplicación.

Instalaciones totales por país

Actualmente, la aplicación ha sido descargada en más de 80 países distintos, a continuación se muestran los que tienen al menos dos descargas, que son 47.

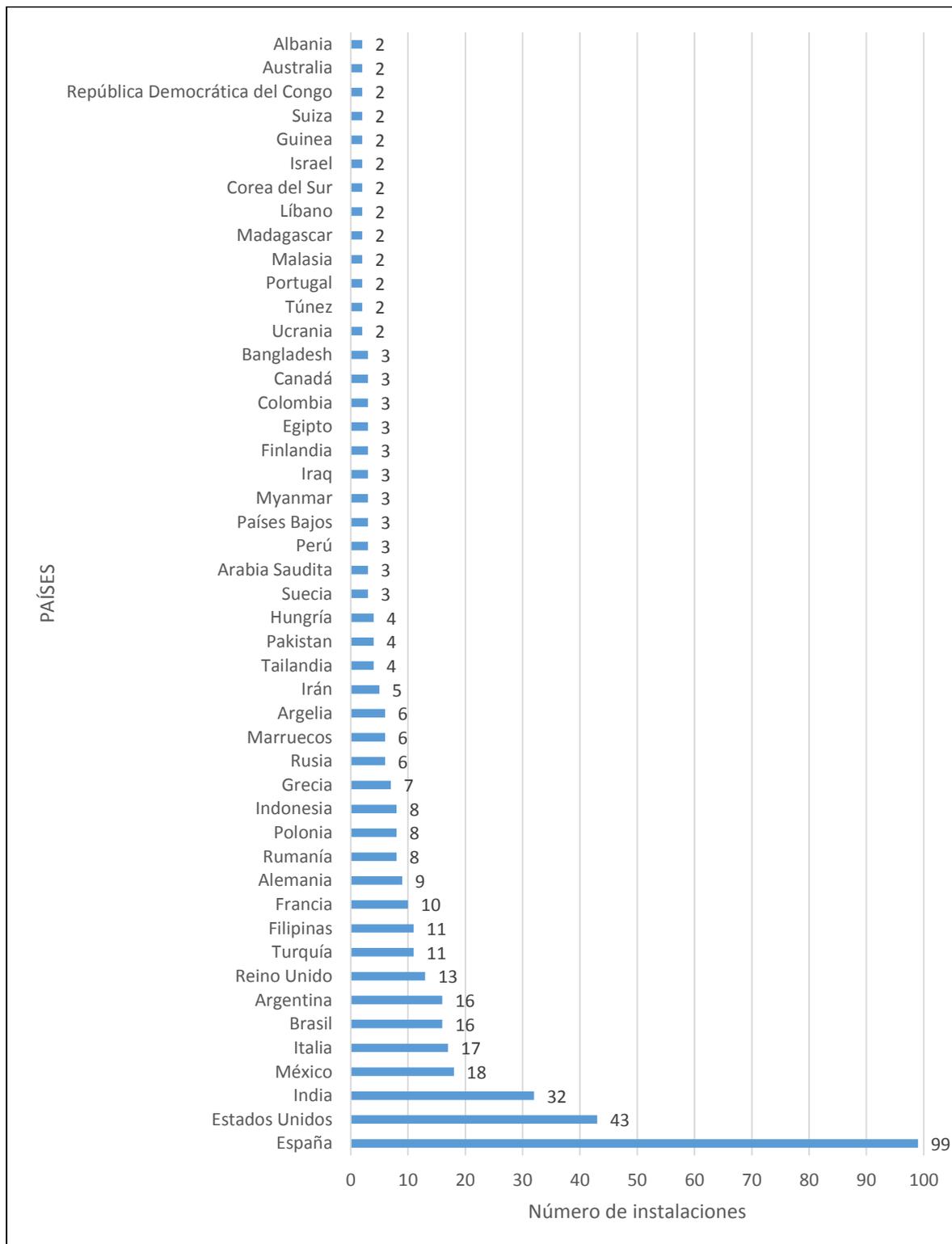


Figura 5-3: Instalaciones totales por país con dos o más descargas a 11-02-2014

Instalaciones totales por versión de Android

En la siguiente figura se puede apreciar el reparto de descargas que hay por cada versión de Android.

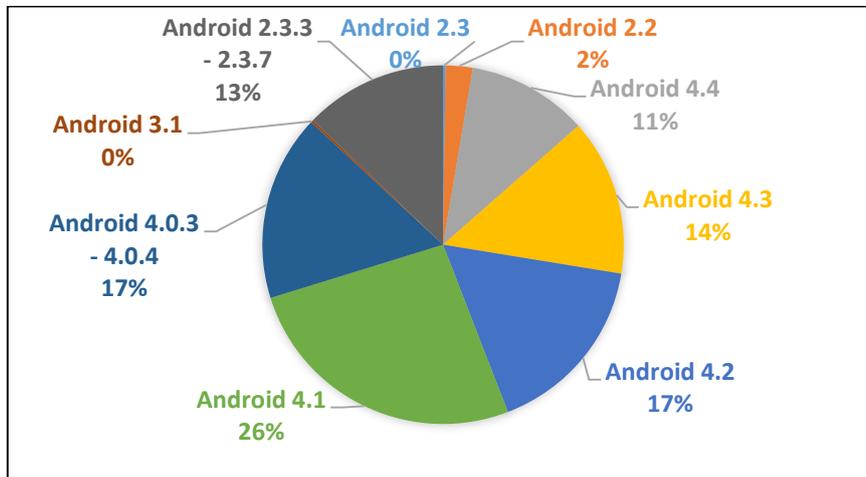


Figura 5-4: Instalaciones totales por versión del sistema operativo a 11-02-2014

Instalaciones actuales por versión de la aplicación

A continuación se muestran las versiones de la aplicación instaladas y su distribución.

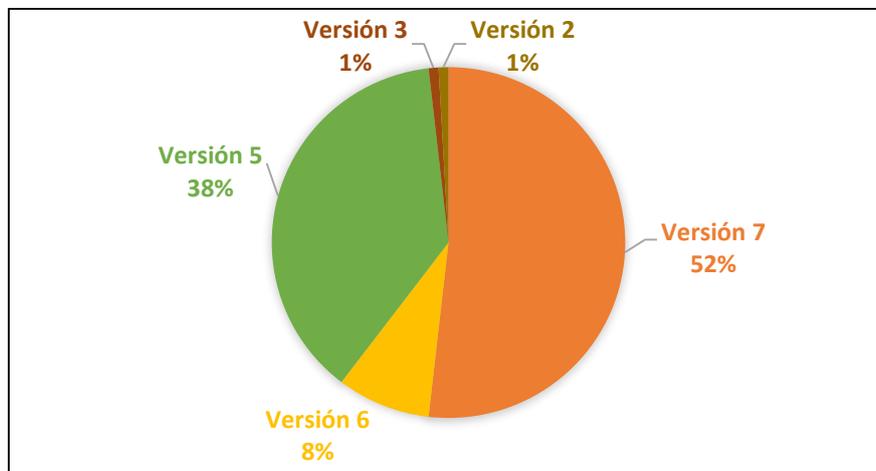


Figura 5-5: Instalaciones actuales por versión de la aplicación a 11-02-2014

Las instalaciones de las versiones 2 y 3, se corresponden con dispositivos desactualizados. La versión 5, es la que más tiempo ha estado disponible, desde el 23-12-2013, hasta el 7-02-2014, por lo que es normal que aún la tengan gran cantidad de terminales. La versión 6 sólo estuvo disponible un día y la versión 7, la final, tan solo lleva cuatro días, aun así, ya es la versión instalada en más dispositivos.

5.4 Cuestionario de la aplicación

	Muy malo (1)	Malo (2)	Normal (3)	Bueno (4)	Muy bueno (5)
Organización general					
Aspecto de los gráficos					
Facilidad de uso					
Utilidad de la opción enviar					
Utilidad de la Ayuda General (F1)					
Fiabilidad y estabilidad					
Consumo de batería					
Velocidad					
Ordenación de los ejercicios					
* Dificultad de los ejercicios:					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
Utilidad de las “tips”					
Calidad general del Tutorial					
Utilidad del Tutorial					
Incluir la solución es:					
Una guía de problemas en el teléfono es:					
Comentarios y sugerencias:					

Tabla 5-2: Cuestionario

*Dificultad: Del 1 al 5, de menor a mayor.

Las estadísticas tomadas a partir de este cuestionario, se realizarán durante el segundo cuatrimestre del curso 2013-2014. Es en este periodo cuando se imparte la asignatura Circuitos electrónicos digitales, para la cual está diseñada la aplicación.

Los alumnos, ya se pueden descargar la aplicación desde Google Play Store y podrán rellenar el cuestionario anterior.

6 CONCLUSIONES Y TRABAJO FUTURO

6.1 Conclusiones

Se han cumplido los objetivos del proyecto. Desarrollar una aplicación prototipo educativa en la plataforma Android, que permita a los usuarios estudiar y realizar la guía de problemas desde sus dispositivos móviles. Se ha hecho énfasis en la fiabilidad, no habiendo recibido información sobre fallos en los 2 meses que lleva en Google Play.

Se han cubierto los principales requisitos que proponía el proyecto:

- Insertar un tutorial teórico en modo sólo lectura.
- Modelar ejercicios de electrónica y adaptarlos a un dispositivo Android.
- Adaptación de imágenes y conexiones a distintos tamaños de pantalla y densidades.
- Creación de tablas binarias.
- Diseñar un formato de ejercicio interactivo con ayudas al usuario.
- Posibilidad de guardar y cargar datos introducidos por el usuario.
- Compartición de datos por correo electrónico.
- Diseñar herramientas de comprobación de errores en los ejercicios.
- Desarrollar una aplicación estable.

Se han aprendido diferentes herramientas y conceptos de interés:

- Programación JAVA y XML.
- Entorno de desarrollo Eclipse.
- Arquitectura Android.
- Uso de herramientas de diseño gráfico para los esquemas de circuitos.
- Organización y planificación de un proyecto.

6.2 Trabajo futuro

La utilización de dispositivos móviles para el estudio tiene enormes ventajas. Esta aplicación ha sido desarrollada desde cero, y la idea principal es que se continúe mejorando en futuros proyectos de fin de carrera.

A continuación se muestran los puntos más importantes para el desarrollo más inmediato de la aplicación:

- Adaptación de la aplicación para realizar exámenes directamente sobre el teléfono. Para ello es necesario agregar una palabra clave que sólo permita acceder a la misma durante el horario de examen, limitar el envío por correo y agregar un temporizador tipo “watchdog” que cierra la aplicación a las 3 horas. Las opciones de corrección automática son evidentes.
- Añadir soporte para más idiomas.
- Optimización de código y de diferentes estructuras de programación.
- Mejorar los mecanismos actuales para la resolución de ejercicios.
- Modificar la interfaz de usuario.
- Incorporación de un cronómetro y sistema de puntuaciones en los ejercicios.
- Creación de una hoja con anotaciones del usuario en cada ejercicio.
- Añadir al punto anterior la posibilidad de que el usuario defina tablas y pueda rellenarlas, estas son muy útiles para resolver varios ejercicios.
- Incorporar nuevos tipos de ejercicios y añadir más a la guía actual.
- Utilización de un servidor donde almacenar los datos que más ocupan, principalmente imágenes.
- Añadir nuevas temáticas o asignaturas.

Más adelante, o paralelamente a lo anterior, se pueden ir mejorando otros aspectos más complejos:

- Realizar un análisis de consumo de batería y posteriores pruebas y cambios para mejorar los resultados. Esta tarea es tema de un proyecto del DSLab.
- Creación de una base de datos en un servidor, en los que aparte de las imágenes, se puedan añadir enunciados, soluciones de ejercicios, consejos y ayudas, y estos puedan ser cargados en la aplicación.

- Idear un mecanismo con el cual se puedan diseñar nuevos ejercicios desde la propia aplicación.
- Compartir los nuevos ejercicios que se creen desde la aplicación.
- Sistema de usuarios registrados con grupos de alumnos y profesores, entrega de ejercicios al usuario profesor, calificaciones, mensajería interna en los grupos.

A parte de las mejoras técnicas anteriores, también se pueden mejorar los aspectos comerciales:

- Ofrecer guías de problemas adicionales a bajo coste.
- Incluir publicidad en la aplicación.
- Una versión de pago con la gestión de usuarios antes comentada.

REFERENCIAS

- [1] Ceballos Sierra, Fco. Javier. JAVA 2 Curso de programación. (3ª ed). RA – MA.
- [2] Tomás Gironés, Jesús. (2011). El gran libro de Android. (2ª ed). MARCOMBO, S.A.
- [3] Material “Desarrollo de aplicaciones para entornos móviles”. EPS UAM.
- [4] Wei- Meng Lee. Beginning Android Application Development. Wiley publishing, Inc.
- [5] <http://developer.android.com/about/dashboards/index.html>
- [6] <http://stackoverflow.com/>
- [7] <http://www.gsmarena.com/>
- [8] <https://play.google.com/store/apps>
- [9] <http://gs.statcounter.com/>
- [10] <http://www.puntogeek.com/>
- [11] <http://www.androidcurso.com/>
- [12] <https://polimedia.upv.es/catalogo/curso.asp?curso=9e0e4ba0-d852-2243-8590-320e9d78ff36>
- [13] http://www.sgoliver.net/blog/?page_id=3011
- [14] <http://www.aprendeandroid.com/menu.htm>

ANEXOS

PRESUPUESTO

1) Ejecución Material

- Compra de ordenador personal (Software incluido)..... 750 €
- Alquiler de impresora láser durante 6 meses 50 €
- Móvil Android 400 €
- Tableta Android 300 €
- Material de oficina 50 €
- Total de ejecución material..... 1.550 €

2) Gastos generales

- 16 % sobre Ejecución Material..... 248 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material..... 93 €

4) Honorarios Proyecto

- 720 horas a 15 € / hora 10800 €

5) Material fungible

- Gastos de impresión 60 €
- Encuadernación 200 €

6) Subtotal del presupuesto

- Subtotal Presupuesto..... 12.951 €

7) I.V.A. aplicable

- 21% Subtotal Presupuesto..... 2.719,71 €

8) Total presupuesto

- Total Presupuesto 15.670,71 €

Madrid, Febrero de 2014
El Ingeniero Jefe de Proyecto

Fdo.: Pedro Madrigal Marina
Ingeniero Superior de Telecomunicación

PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de una APLICACIÓN DE PROBLEMAS RESUELTOS DE CIRCUITOS DIGITALES COMBINACIONALES BAJO ANDROID. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado

con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.