



APLICACIÓN PARA MÓVILES CON SISTEMA OPERATIVO ANDROID DISEÑADA
COMO HERRAMIENTA DE APOYO DIDÁCTICO PARA LA COMPRENSIÓN DE
CIRCUITOS LÓGICOS FUNDAMENTALES

ELABORADO POR:
JOHN JAIRO ZARATE PAVAS

UNIVERSIDAD PEDAGÓGICA NACIONAL
FACULTAD DE CIENCIA Y TECNOLOGÍA
DEPARTAMENTO DE TECNOLOGÍA
LICENCIATURA EN ELECTRÓNICA
BOGOTÁ
2015

APLICACIÓN PARA MÓVILES CON SISTEMA OPERATIVO ANDROID DISEÑADA
COMO HERRAMIENTA DE APOYO DIDÁCTICO PARA LA COMPRESIÓN DE
CIRCUITOS LÓGICOS FUNDAMENTALES

JOHN JAIRO ZARATE PAVAS

TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE:
LICENCIADO EN ELECTRÓNICA

ASESOR:

ING. Camilo Rodríguez

UNIVERSIDAD PEDAGÓGICA NACIONAL
FACULTAD DE CIENCIA Y TECNOLOGÍA
DEPARTAMENTO DE TECNOLOGÍA
LICENCIATURA EN ELECTRÓNICA
BOGOTÁ

2015

NOTA DE ACEPTACIÓN

FIRMA DEL DIRECTOR

JURADO

JURADO

AGRADECIMIENTO

A Dios por brindarme la oportunidad de estar en el lugar donde estoy.

A mis padres Luz Dary Pavas y Miguel Ángel Zarate, que me apoyaron incondicionalmente sin importar las dificultades, permitiéndome culminar con mi carrera profesional.

A mi pareja sentimental Marisol Torres Rodríguez por acompañarme durante la carrera profesional y brindarme su especial compañía, aconsejarme en todo momento procurando siempre mi bienestar.

A la Universidad Pedagógica Nacional y a todas las personas que la componen en especial a los docentes que me brindaron sus conocimientos y saberes.

Al profesor Camilo Rodríguez por brindarme sus conocimientos y permitirme terminar con este trabajo.

Y a todas las personas que me apoyaron para culminar uno de los objetivos en mi vida.

Gracias.

RESUMEN ANALÍTICO EN EDUCACIÓN – RAE

1. Información General	
Tipo de documento	Trabajo de grado
Acceso al documento	Universidad Pedagógica Nacional. Biblioteca Central
Título del documento	APLICACIÓN PARA MÓVILES CON SISTEMA OPERATIVO ANDROID DISEÑADA COMO HERRAMIENTA DE APOYO DIDÁCTICO PARA LA COMPRENSIÓN DE CIRCUITOS LÓGICOS FUNDAMENTALES
Autor(es)	Zarate Pavas, John Jairo
Director	Rodríguez Torres, Camilo Enrique
Publicación	Bogotá. Universidad Pedagógica Nacional, 2015. 127 p.
Unidad Patrocinante	Universidad Pedagógica Nacional
Palabras Claves	APLICACIÓN MÓVIL, ANDROID, CIRCUITOS LÓGICOS BÁSICOS.

2. Descripción
Trabajo de grado en el que se desarrolló una aplicación móvil para móviles con sistema operativo Android en el programa eclipse juno, en la que se pueden encontrar tres módulos principales: el primero es un simulador de compuertas lógicas digitales; el segundo es un módulo sobre fundamentos de circuitos lógicos básicos y el último, permite encontrar los datos generales de cada compuerta lógica.

3. Fuentes
<ul style="list-style-type: none">• Barco Gómez, C. (2005). Álgebra Booleana. Aplicaciones tecnológicas. Colombia. Universidad de Caldas.• Carmen C. Valero. Margarita R. Redondo. Ana S. Palacín. (2012). Tendencias actuales en el uso de dispositivos móviles en educación. La educ@ción. Organization of American States.• Michael M. Cirovic. (1991) Electrónica fundamental: dispositivos, circuitos y sistemas. USA. Reverte, S. A.• Zulma Cataldi (2000). Metodología de diseño, desarrollo y evaluación de software educativo. Buenos Aires, Argentina.• Thomas L. Floyd, (2006). Fundamentos de sistemas digitales. Madrid, España. Pearson Educación S. A.• José D Muñoz Frías, Introducción a los sistemas digitales, un enfoque usando lenguajes de descripción de hardware. España de Creative Commons.• Eck J, David. (2010). Introducción a La Programación usando Java Versión 5.1. EUA. Universidad Nacional Abierta de Venezuela.• Vianney J. Patiño, Roberto A. Ortiz (2012). Aplicación Móvil para la Comisión de

Tránsito del Ecuador. Proyecto de grado. Guayaquil, Ecuador.

- Maira C, Gasca. Luis L, Camargo. Byron M, Delgado. Metodología para el desarrollo de aplicaciones móviles. Universidad del Magdalena. Santa Marta, Colombia.
- Salvaor G, Oliver. Curso Programación Android. PDF.

4. Contenidos

1. Planteamiento del problema
Se identifica el problema junto a la necesidad que se crea y se realiza una justificación dándose un pronóstico de la solución.
2. Objetivos
Se proponen los objetivos a desarrollar.
3. Marco Teórico – Conceptual
Se analizan los antecedentes y se estudia la base teórica para el desarrollo del proyecto.
4. Metodología
Se plantea la metodología y se realizan los pasos de la misma para el desarrollo de la aplicación móvil.
5. Conclusiones
Se realizan las conclusiones enfocadas a los resultados y desarrollo del proyecto.

Los objetivos planteados en el proyecto se describen a continuación:

- Diseñar un módulo en la aplicación móvil para la simulación de las principales compuertas lógicas digitales.
- Realizar un módulo sobre los fundamentos teóricos de circuitos lógicos básicos.
- Efectuar un módulo que le permite al usuario visualizar datos generales de cada compuerta lógica.

5. Metodología

Para la realización del proyecto se eligió la metodología en *Cascada* utilizada ampliamente para el desarrollo de software, debido a las fases con las que cuenta para la ejecución de este tipo de trabajos. En esta metodología se da un proceso secuencial de avance en el que cada una de sus etapas debe finalizar para poder comenzar la otra; proceso similar a una cascada. Las etapas que realiza esta metodología están definidas de la siguiente manera:

- Análisis Requisitos
Se analizan las necesidades y en función de estas se crean unos objetivos, realizándose un documento donde se especifiquen los requisitos.
- Diseño
El sistema se descompone y organiza para ser realizado por separado. De este diseño surge la descripción de la estructura del sistema y la especificación de cada una de sus partes, así como la forma en que se combinan una con otras.
- Codificación
Se realiza el código fuente, haciendo uso de prototipos así como de pruebas y ensayos

para corregir errores.

- Pruebas
Los elementos programados por separado son probados y unificados para componer el sistema y realizar la entrega al usuario final que ejecuta el sistema, teniendo en cuenta, que los programadores ya realizaron exhaustivas pruebas y no tendría ningún problema.
- Despliegue
Se realizan los ajustes necesarios para que cumpla con todos los requisitos realizados en la etapa de análisis se realiza la entrega final del software.

6. Conclusiones

- Las herramientas de software y hardware para el diseño y simulación de la aplicación, permitieron realizar los cambios necesarios para cumplir con los objetivos y los requisitos del sistema, constatando un buen funcionamiento.
- El uso de la metodología en *cascada*, permitió obtener resultados en poco tiempo, se dio mayor prioridad a los requerimientos; esto representó una ventaja frente a otras metodologías, ya que se realizó un mantenimiento de la aplicación móvil.
- La construcción de la interfaz de usuario de una aplicación móvil en Eclipse Android es un proceso fácil de realizar, debido a la colección de herramientas, elementos y diseños que proporciona este entorno de desarrollo, permitiendo la creación de aplicaciones básicas y sencillas para programadores novatos.
- Gracias a la combinación de diferentes recursos, archivos XML y actividades fue posible la realización del simulador de compuertas lógicas básicas, permitió a los usuarios de la aplicación disponer de una herramienta que emula el comportamiento de una compuerta lógica en particular o disponer de varias compuertas lógicas y conectarlas entre sí para ver el resultado de entradas y salidas.
- El mayor inconveniente en la realización del proyecto surgió por la falta de experiencia en el lenguaje de programación Java y la plataforma Android; debido a que no se manejaba con precisión los recursos suministrados por estos sistemas y la documentación oficial proporcionada por Google para Android carece de contenido práctico, teniendo que acudir a otros escenarios como foros, blogs y redes sociales para el desarrollo de la aplicación, sin embargo, la ejecución de la aplicación móvil a usuarios principiantes y experimentados demuestra que la aplicación es bien recibida en el ámbito académico.
- Las respuestas y comentarios de los estudiantes de grado undécimo demuestran la motivación que este tipo de software genera en ellos con una adecuada supervisión de los docentes.

Elaborado por:	Zarate Pavas, John Jairo
Revisado por:	Rodríguez Torres, Camilo Enrique

Fecha de elaboración del Resumen:	07	09	2015
--	----	----	------

RESUMEN

El presente documento está enfocado al desarrollo de una aplicación para equipos móviles con sistema operativo Android, que permite emular y comprender el comportamiento de las compuertas lógicas digitales.

La aplicación consta de tres módulos; el primero es el simulador de las compuertas lógicas digitales; el segundo es el módulo de fundamentos teóricos sobre los circuitos lógicos básicos y el último, permite encontrar datos generales sobre las compuertas lógicas digitales.

ABSTRACT

This document focuses on the development of an application for mobile devices with Android operating system, which allows emulating and understanding the behavior of digital logic gates.

The application consists of three modules; the first one is the simulator of the logical digital gates. The second one, is the module of theoretical foundations on the logical basic circuits and the last one, allows finding general information on the logical digital gates.

TABLA DE CONTENIDO

Pág.

1	PLANTEAMIENTO DEL PROBLEMA.....	1
1.1	IDENTIFICACIÓN DEL PROBLEMA	1
1.1.1	IDENTIFICACIÓN DE LA NECESIDAD	1
1.1.2	Diagnóstico.....	2
1.1.3	Pronóstico.....	3
1.2	Justificación	4
2	OBJETIVOS.....	5
2.1	General.....	5
2.2	Específicos	5
3	ANTECEDENTES	5
4	MARCO TEÓRICO – CONCEPTUAL	6
4.1	Marco Teórico	6
4.1.1	Circuitos lógicos básicos.	7
4.1.2	Álgebra de Boole	10
4.1.3	Elementos de programación.....	10
4.1.4	Evaluación de <i>software</i> educativo	11
4.2	Marco Conceptual	13
4.2.1	Sistema binario	13
4.2.2	Tabla de verdad	13
4.2.3	Sistema Operativo Móvil	13
4.2.4	Sistema Operativo Android.....	13
4.2.5	Interfaz de usuario.....	13
4.2.6	Usabilidad.....	13
4.2.7	Evaluar	13
4.2.8	Exploración.....	13
5	Metodología.....	14
6	Análisis de requisitos.....	17
6.1	Perfil del Usuario final	17
6.2	Requisitos de usuario	18

6.3	Requisitos del sistema	19
6.3.1	Requisitos funcionales	19
6.3.2	Requisitos no funcionales	21
6.4	Especificación de casos de uso	22
6.4.1	Caso de uso del Menú principal.....	22
6.4.2	Caso de uso del módulo simulador	24
6.4.3	Caso de uso del módulo sobre fundamentos de circuitos lógicos básicos	30
6.4.4	Caso de uso del módulo sobre datos generales de compuertas lógicas.	31
7	Diseño	32
7.1	Finalidad del proyecto	32
7.2	Limitaciones	34
7.3	Logotipo de la aplicación	34
7.4	Esquema general de la aplicación.....	35
7.5	Diagrama de clases de la aplicación	36
7.6	Tarjetas CRC	38
7.7	Diseño de interfaz gráfica	41
8	Construcción y arquitectura	46
8.1	Herramientas de desarrollo.....	46
8.2	Estructura de la aplicación.....	46
8.2.1	El Archivo AndroidManifest.xml	47
8.2.2	Actividades	48
8.2.3	Vista.....	48
8.3	Menú principal	48
8.3.1	Actividad principal	49
8.4	Módulo de datos generales	49
8.5	Módulo sobre fundamentos teóricos de circuitos lógicos básicos	50
8.6	Módulo del simulador de compuertas lógicas básicas	53
9	Pruebas	56
9.1	Emulador de dispositivos móviles	56
9.2	Dispositivo real	59
10	Evaluación	62
10.1	Resultados de los cuestionarios a estudiantes de grado undécimo	62
10.2	Resultados del cuestionario a docentes y estudiantes universitarios.....	67

11	Descarga e instalación de la aplicación	69
12	Manual de Usuario	69
13	CONCLUSIONES	72
14	BIBLIOGRAFÍA.....	73

LISTA DE FIGURAS

Fig. No 1. Inversor que complementa una variable de entrada.	8
Fig. 2. Símbolos lógicos estándar de la compuerta And con dos entradas (estándar ANSI/IEEE 91-1984).	8
Fig. No 3. Símbolos lógicos estándar de la compuerta OR con dos entradas (estándar ANSI/IEEE 91-1984)	8
Fig. No 4. Símbolos lógicos estándar de la compuerta NAND (ANSI/IEEE 91 - 1984).	9
Fig. No 5. Símbolos lógicos estándar de la compuerta NOR (ANSI/IEEE 91 - 1984).	9
Fig. No 6. Símbolos lógicos estándar de la compuerta XOR.	9
Fig. No 7. Símbolos lógicos estándar de la compuerta XNOR.	10
Fig. No 8. Modelo “Cascada” para el desarrollo de software. El proceso fluye de arriba hacia abajo y tiene retroalimentación.	14
Fig. No 9. Aplicación del esquema general del método en Cascada al proyecto.	17
Fig. No 10. Estilos de documentación para la recolección de requisitos.	18
Fig. No 11. Caso de uso del Menú principal de la aplicación móvil.	23
Fig. No 12. Caso de uso del módulo simulador.	25
Fig. No 13. Caso de uso. Simulador de compuertas lógicas.	26
Fig. No 14. Caso de uso. Fundamentos de circuitos lógicos básicos.	30
Fig. No 15. Caso de uso. Datos generales de compuertas lógicas.	31
Fig. No 16. Logotipo de la aplicación sobre circuitos lógicos básicos.	34
Fig. No 17. Esquema general de navegación de la aplicación planteada en este documento.	35
Fig. No 18. Diagrama de clases de la aplicación.	38
Fig. No 19. Menú principal de la aplicación sobre Circuitos Lógicos Básicos.	41
Fig. No 20. Interfaz gráfica del manual de uso del simulador y el botón que dirige al usuario al simulador.	42

Fig. No 21. Interfaz gráfica del simulador de compuertas lógicas.	43
Fig. No 22. Interfaz gráfica del módulo sobre fundamentos de circuitos lógicos básicos.	43
Fig. No 23. Mensaje por pantalla al pulsar una de las imágenes en el punto simbología lógica.	44
Fig. No 24. Módulo de datos generales de las principales compuertas lógicas.	45
Fig. No 25 Vista de una de las pantallas al seleccionar una compuerta lógica en el módulo de datos generales.	45
Fig. No 26. Entorno de trabajo de Eclipse Android, junto a la máquina virtual para emular los diferentes dispositivos móviles.	47
Fig. No 27. Botón básico.	49
Fig. No 28. Diseño del índice de la base de datos.	50
Fig. No 29. Botón imagen o Image Button.	51
Fig. No 30. Botón con diseño personalizado normal y presionado en la consola de trabajo y en la máquina virtual.	51
Fig. No 31. Presentación de la galería de botones para seleccionar la información de cada compuerta lógica.	52
Fig. No 32. Entorno de trabajo en la interfaz simulador junto a la máquina virtual en el módulo simulador.	53
Fig. No 33. . Archivos XML que conforman las compuertas lógicas y tienen los elementos de conexión.	54
Fig. No 34. Circuito tomado para la prueba del simulador de compuertas lógicas de la aplicación en el emulador de Eclipse.	56
Fig. No 35. Simulación y resultado del circuito planteado en la prueba #1 en el emulador de dispositivos móviles.	57
Fig. No 36. Simulación y resultado del circuito planteado en la prueba #2 en el emulador de dispositivos móviles.	57
Fig. No 37. Circuito elegido para realizar la prueba final del simulador de compuertas lógicas en el emulador.	58
Fig. No 38. Última prueba del módulo simulador en el emulador de dispositivos móviles.	58
Fig. No 39. Simulación y resultado del circuito planteado en la prueba #1 en el dispositivo móvil real.	59
Fig. No 40. Simulación y resultado del circuito planteado en la prueba #2 en el dispositivo real.	60
Fig. No 41. Última prueba del módulo simulador en el dispositivo móvil real.	60

Fig. No 42. Prueba del módulo fundamentos teóricos sobre circuitos lógicos básicos en el dispositivo móvil real.	61
Fig. No 43. Prueba del módulo de datos generales sobre compuertas lógicas digitales en un dispositivo real.	61
Fig. No 44. Resultado de la pregunta: Facilidad de uso.	63
Fig. No 45. Resultado de la pregunta: Facilidad de exploración.	63
Fig. No 46. Resultado de la pregunta: ¿Le facilita la comprensión del tema?	64
Fig. No 47. Resultado de la pregunta: ¿Son claros los contenidos?	64
Fig. No 48. Resultado de la pregunta: ¿Ha despertado interés en usted?	65
Fig. No 49. Resultado de la pregunta: ¿Son adecuados los recursos para comprender el tema?	65
Fig. No 50. Realización de un circuito combinatorio en el simulador de la aplicación por parte de uno de los estudiantes del colegio INEM.	66
Fig. No 51. Respuestas de uno de los estudiantes del colegio INEM.	66
Fig. No 52. Verificación del módulo de datos generales de compuertas lógicas por parte de un estudiante del colegio INEM.	67
Fig. No 53. Promedios obtenidos de las respuestas de docentes y estudiantes de la Universidad Pedagógica Nacional.	68
Fig. No 54. Interfaz gráfica del simulador. Instrucción para colocar una compuerta lógica en el área de simulación.	70
Fig. No 55. Instrucción para cambiar los datos de entrada de una compuerta lógica.	70
Fig. No 56. Instrucción para realizar una conexión entre compuertas.	71
Fig. No 57. Instrucción para quitar una conexión entre compuertas.	71

LISTA DE TABLAS

Tabla No 1. Preguntas claves de justificación.	4
Tabla No 2. Esquema de evaluación de la aplicación a estudiantes.	12
Tabla No 3. Esquema de evaluación de la aplicación a docentes y estudiantes de electrónica.	12
Tabla No 4. Caso de Uso “Acceder al simulador de compuertas lógicas y manual de uso”.	23

Tabla No 5. Caso de Uso “Acceder al módulo de fundamentos de circuitos lógicos básicos”.	24
Tabla No 6. Caso de Uso “Acceder al módulo sobre datos generales de compuertas lógicas”.	24
Tabla No 7. Caso de Uso “Acceder al simulador de compuertas lógicas”.	25
Tabla No 8. Caso de Uso “Simulador de compuertas lógicas”.	26
Tabla No 9. Caso de Uso “Selecciona posición y escoge compuerta”.	27
Tabla No 10. Caso de Uso “Borrar compuerta”.	27
Tabla No 11. Caso de Uso “Conectar y desconectar compuertas”.	28
Tabla No 12. Caso de Uso “Cambiar compuerta”.	28
Tabla No 13. Caso de Uso “Cambio valores de entrada de la compuerta”.	29
Tabla No 14. Caso de Uso “Activar el simulador”.	29
Tabla No 15. Caso de Uso “Navegación por el contenido teórico de fundamentos de circuitos lógicos básicos”.	30
Tabla No 16. Caso de Uso “Pulsar imagen y recordar nombre de la compuerta”.	31
Tabla No 17. Caso de Uso “Pulsar imagen y recordar nombre de la compuerta”.	32
Tabla No 18. Resultados de la aplicación del cuestionario a estudiantes de grado once (11) del colegio INEM.	62
Tabla No 19. Sugerencias y comentarios de estudiantes que contestaron el cuestionario.	65
Tabla No 20. Resultados de la aplicación del cuestionario a docentes y estudiantes universitarios de la Universidad Pedagógica Nacional de Colombia.	68

CAPITULO I

1 PLANTEAMIENTO DEL PROBLEMA

1.1 IDENTIFICACIÓN DEL PROBLEMA

1.1.1 IDENTIFICACIÓN DE LA NECESIDAD

En la actualidad existen varios sistemas tecnológicos que operan de forma binaria, entre ellos: alarmas de seguridad, encendido de luces automáticas, puertas automáticas, sistemas de comunicación digital y los computadores entre muchos otros, destinados a la generación y transmisión de señales digitales.

En un sistema binario, cada señal de entrada y salida representa un uno (1) o un cero (0) de acuerdo con los rangos de voltaje que se determinan; en cada uno de ellos y la salida está condicionada por operaciones lógicas. Estas operaciones o funciones booleanas son representadas por elementos llamados compuertas lógicas, que son circuitos lógicos básicos [1].

Los circuitos lógicos básicos son vistos en diferentes niveles educativos tanto en Colombia como en otros lugares del mundo; debido a su importancia y utilización en las aulas de clase existen múltiples herramientas tecnológicas que facilitan su aprendizaje, como *software* en línea o de escritorio para PC o dispositivos electrónicos.

Habitualmente cuando se comienza a estudiar estos circuitos, se emplean herramientas adicionales para realizar tareas como la elaboración de un esquema, la simulación de un circuito o simplemente la verificación de cierta información.

En la actualidad se observa la utilización cada vez mayor de dispositivos móviles con sistema operativo Android, por la facilidad de adquisición y el desarrollo de aplicaciones móviles en comparación con otros proveedores, por lo anterior, se realizó una revisión de aplicaciones móviles orientadas a los circuitos lógicos básicos en la tienda de *Google Play* (principal proveedora de aplicaciones para Android) y se encontró que las existentes, están orientadas a la simulación de circuitos lógicos; dejando de lado la teoría de los circuitos lógicos básicos; esta situación, genera dificultades a quienes estudian estos sistemas en cuanto no permite el acceso a la teoría, a la aplicación de la teoría y la utilización de estas herramientas sin estar conectado a una red informática.

Por ello, se evidencia la necesidad de realizar una aplicación móvil en la que se encuentre la teoría de los circuitos lógicos básicos y la aplicación de la misma, sin

necesidad de conectarse a una red; para aportar al aprendizaje de los estudiantes que deseen usarla en cualquier momento y lugar.

1.1.2 Diagnóstico

Como se mencionó anteriormente, se hizo una búsqueda de aplicaciones móviles referentes a los circuitos lógicos básicos en la tienda de *Google Play* que permite dar un mejor diagnóstico.

A continuación se presentan las aplicaciones consultadas y se hace una breve reseña de cada una.

1. ElectroDroid

La aplicación permite calcular magnitudes de diferentes dispositivos electrónicos y circuitos eléctricos; además, presenta una documentación técnica de estos dispositivos; la versión gratuita contiene publicidad, su idioma original es el español y se puede clasificar como un simulador de dispositivos eléctricos básicos sin contar con las compuertas lógicas.

2. LogicMaker

En ella se pueden realizar distintos circuitos combinatorios con compuertas lógicas y diferentes comprobaciones; no propone ejercicios, teoría o comparación de datos y es muy difícil de utilizar en cuanto a su funcionalidad; el idioma en el que se encuentra es el inglés y es clasificado como un simulador.

3. Logic Simulator Pro

Permite construir circuitos con puertas lógicas y otros dispositivos de forma digital para simular sus funciones; contiene una interfaz sencilla y fácil de utilizar; es totalmente gratis; el idioma en el que se encuentra es el inglés y se clasifica como un simulador de circuitos lógicos.

4. Combinational Circuits

Es una aplicación sobre fundamentos de circuitos lógicos combinatorios que maneja teoría y resolución de ejercicios; es completamente gratis y en inglés. Clasificada como una lección educacional.

5. EveryCircuit

Es un simulador análogo que cuenta con diferentes dispositivos electrónicos; permite diseñar diferentes circuitos con elementos como: compuertas lógicas, fuentes de voltaje alternas y continuas, resistencias, transistores, entre otros; esta aplicación fue diseñada para estudiantes de secundaria, universitarios y usuarios en general; la versión gratuita permite acceder y realizar montajes básicos y en la pagada se puede observar y realizar mejores diseños de circuitos electrónicos. El idioma original es el inglés.

Las cinco aplicaciones mencionadas anteriormente son las que más tienen relación con los circuitos lógicos básicos, por esta razón, son las que se analizaron y se identificaron en este documento.

La aplicación que puede acercarse más a la solución del anterior problema es la número cuatro (4); ya que entrega a los usuarios una lección educacional sobre circuitos lógicos combinatorios y diferentes actividades para la resolución de ejercicios, pero, no permite simular circuitos combinatorios propuestos por el usuario, ni encontrar información de circuitos lógicos básicos que es el enfoque del presente proyecto.

De las cinco aplicaciones, cuatro (4) son simuladores sobre algún tema de electrónica y una (1) es una lección educacional; en esta misma proporción cuatro (4) son en inglés y una (1) en español; dificultando el aprendizaje de algunos jóvenes que no manejan el inglés como segunda lengua.

De esta manera, se evidencia que para los usuarios de Android es difícil encontrar en la tienda de *Google Play* una aplicación móvil que le permita obtener la teoría sobre circuitos lógicos básicos y que ofrezca un instrumento de aplicación para esta teoría en una sola aplicación móvil.

1.1.3 Pronóstico

La oportunidad de desarrollar una aplicación para móviles con sistema operativo Android, que permita a los usuarios encontrar los fundamentos de circuitos lógicos básicos, junto a los datos generales de las compuertas lógicas y un instrumento de simulación que posibilite realizar circuitos combinatorios; brindando una alternativa diferente a las aplicaciones que se encuentran en la tienda de *Google Play*, principal proveedora de aplicaciones para dispositivos Android.

Además, la aplicación se diseñará completamente en el idioma español y de forma nativa, es decir, que después de ser descargada del servidor, el usuario podrá

acceder a ella en cualquier momento y lugar, utilizando sus funcionalidades sin ningún inconveniente, ni publicidad.

1.2 Justificación

En la actualidad, el uso de dispositivos móviles en educación está orientado a la utilización cada vez mayor en las aulas de clase, tanto en las instituciones educativas como en cualquier lugar que se necesite los servicios de estos elementos [2]. El avance y las evaluaciones positivas de los dispositivos en los estudiantes, dan pie para el desarrollo de aplicaciones orientadas a la educación.

Teniendo en cuenta esta disposición, se presenta la oportunidad de desarrollar una aplicación móvil que sirva de apoyo didáctico para el aprendizaje de los circuitos lógicos básicos; que ofrezca a los usuarios de móviles Android la posibilidad de integrar la teoría y la práctica, como una herramienta de estudio autónomo sin conexión a una red informática y en idioma español. Es una excelente posibilidad que se da, debido a la carencia de aplicaciones móviles en Android que integren estos elementos de estudio. A su vez, aporta en el desarrollo de aplicaciones móviles referentes a la electrónica y en especial a los circuitos lógicos básicos.

En el desarrollo de esta aplicación, inicialmente, se realizará en tres módulos, el primero de ellos es el simulador de compuertas lógicas para verificar el comportamiento de estos elementos, el segundo presenta los fundamentos teóricos de circuitos lógicos básicos para el estudio autónomo y el último permite observar datos generales de las principales compuertas lógicas, utilizados en el momento de realizar el montaje de un circuito electrónico, resolver la salida de un esquema lógico o comprar el encapsulado de la compuerta lógica en una tienda de dispositivos electrónicos.

Cuadro de justificación

¿Qué?	Una aplicación para móviles Android como herramienta de apoyo didáctico para el aprendizaje de circuitos lógicos básicos.
¿Por qué?	Por la falta de aplicaciones para móviles Android que integran la teoría y aplicación de esta teoría en una misma aplicación.
¿Para qué?	Para aportar en el desarrollo de aplicaciones móviles que sirvan como herramienta de estudio autónomo en el campo de los circuitos lógicos básicos.

Tabla No 1. Preguntas claves de justificación.

2 OBJETIVOS

2.1 General

- Desarrollar una aplicación para dispositivos móviles con sistema operativo Android, que sirva como herramienta de apoyo didáctico para el aprendizaje y simulación de circuitos lógicos básicos.

2.2 Específicos

- Diseñar un módulo en la aplicación móvil para la simulación de las principales compuertas lógicas digitales.
- Realizar un módulo sobre los fundamentos teóricos de circuitos lógicos básicos.
- Efectuar un módulo que le permite al usuario visualizar datos generales de cada compuerta lógica.

CAPITULO II

MARCO DE REFERENCIA

3 ANTECEDENTES

Algunos de los antecedentes relacionados con este trabajo y que sirvieron de apoyo para el mismo, se presentan a continuación, tanto para los circuitos lógicos como para el desarrollo de la aplicación móvil.

- 1) Aplicación de problemas resueltos de circuitos digitales combinatorios bajo Android. Universidad Autónoma de Madrid. Pedro Madrigal Marina.

Este trabajo de grado consiste en la realización de una aplicación para dispositivos Android en los que se crea una guía de problemas y una lección educativa sobre circuitos lógicos combinatorios. En el mismo, los problemas fueron seleccionados de una guía realizada para la asignatura de circuitos electrónicos digitales de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid.

El principal objetivo de la aplicación es ofrecer a los estudiantes una guía de ejercicios de síntesis de circuitos, permitiendo corregirlos y obtener ayuda de la aplicación para resolverlos y compartirlos con otros usuarios.

Este proyecto es una guía para estructurar un documento de trabajo de grado en cuanto a aplicaciones móviles y sirve de ejemplo para el diseño de ejercicios sobre circuitos lógicos.

- 2) Herramientas de programación multimedia en IOS y Android. Universidad Politécnica de Madrid. Álvaro Manuel Seixas Domínguez.

El proyecto realiza un estudio de las herramientas que facilitan la creación y distribución de aplicaciones en distintas plataformas móviles; con la finalidad de poder seleccionar el mejor instrumento de desarrollo en función del proyecto.

Al finalizar el proyecto selecciona una herramienta para la creación de aplicaciones móviles en particular y se realiza una aplicación móvil con el instrumento seleccionado.

Este proyecto permite escoger la mejor opción para el desarrollo de aplicaciones.

- 3) Estudio y desarrollo de aplicaciones para dispositivos móviles Android. Universidad Técnica del Norte, Ecuador. Mónica Lucia Tapia Marroquín.

Realizado para los creadores de *software* con conocimientos previos en lenguaje JAVA interesados en crear aplicaciones móviles con plataforma Android, brinda una visión general de una aplicación, se detalla los conceptos básicos para el aprendizaje del sistema operativo Android y finalmente se presenta un pequeño aplicativo para la demostración de funcionalidades básicas.

Si se conoce a fondo el sistema operativo Android, con su historia, ventajas, desventajas, descripción y códigos; permite comenzar a programar en Android con elementos sencillos y básicos de programación.

Al realizarse la observación de los antecedentes, es interesante que no se haya encontrado una aplicación como la que se planteó en este trabajo, dando la posibilidad de ofrecer algo novedoso y útil para los estudiantes y usuarios que necesiten encontrar en una sola aplicación la teoría y simulación de los circuitos lógicos básicos.

4 MARCO TEÓRICO – CONCEPTUAL

4.1 Marco Teórico

Este proyecto consiste en el diseño y desarrollo de una aplicación móvil que sirva de apoyo didáctico para el aprendizaje de los circuitos lógicos básicos, en la que pueda encontrar los siguientes módulos:

- Un simulador de circuitos lógicos combinatorios.
- Un módulo sobre fundamentos de circuitos lógicos básicos.

- Un módulo de datos generales de cada compuerta lógica.

Dentro del proyecto se distinguen los siguientes aspectos: los circuitos lógicos básicos que son la base teórica, las herramientas de programación y la evaluación de la aplicación móvil.

4.1.1 Circuitos lógicos básicos.

4.1.1.1 Circuitos lógicos digitales

Estos circuitos procesan y almacenan señales eléctricas binarias que son utilizados en: calculadoras, proceso de datos, comunicaciones digitales y sistemas de control. Los sistemas digitales utilizan repetidamente una limitada variedad de bloques constitutivos clasificados en dos grupos. Uno de estos grupos integra los llamados circuitos lógicos combinatorios, constituidos por compuertas lógicas que procesan las señales digitales y los otros son los circuitos lógicos secuenciales constituidos por multivibradores, los cuales pueden generar o almacenar señales digitales [3].

La aplicación planteada en el presente trabajo está dirigida al desarrollo de contenido teórico - práctico sobre los circuitos lógicos combinatorios.

4.1.1.2 Operaciones Lógicas Fundamentales

Las operaciones lógicas fundamentales en la que se basan los circuitos digitales son: la suma lógica, el producto lógico y complementación. Su base matemática es el álgebra de Boole, como conjunto de reglas matemáticas que tratan con variables binarias y con las tres operaciones mencionadas.

Un aspecto particular en el álgebra de Boole es que las expresiones matemáticas corresponden con un determinado circuito lógico el cual, se puede representar de forma analítica o/y física con las llamadas compuertas lógicas.

4.1.1.3 Compuertas Lógicas

Puerta lógica: Una puerta o compuerta lógica es un dispositivo electrónico que realiza una función booleana según sus propiedades lógicas. Esencialmente estos dispositivos son circuitos electrónicos de conmutación integrados en un chip.

4.1.1.3.1 Compuerta Negadora, Inversora o NOT

Realiza la operación inversión o complementación. El inversor cambia de nivel lógico alto a nivel lógico bajo y viceversa. En términos de bits, cambia de 1 a 0 y

de 0 a 1. La figura que ilustra el símbolo tradicional de la compuerta NOT se presenta a continuación:

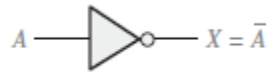


Fig. No 1. Inversor que complementa una variable de entrada [6].

4.1.1.3.2 Compuerta And

Es una de las compuertas básicas con la que se construyen todas las funciones lógicas; puede tener dos o más entradas, una única salida y realiza la operación *multiplicación lógica*. Para ilustrar una compuerta AND con dos entradas y su respectiva salida X , la Figura No 2 muestra esta compuerta.

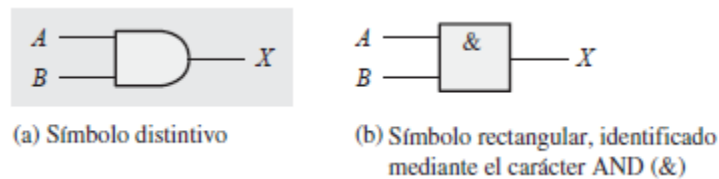


Fig. 2. Símbolos lógicos estándar de la compuerta And con dos entradas (estándar ANSI/IEEE 91-1984) [5].

4.1.1.3.3 Compuerta OR

La compuerta OR realiza la operación conocida como suma lógica y es una de las puertas básicas con las que se efectúan las demás funciones lógicas. Al igual que la compuerta *And* puede manejar dos o más entradas y una salida. Los símbolos lógicos estándar de la compuerta OR son ilustrados en la Figura No 3 donde se pueden ver estas compuertas con dos entradas y su respectiva salida.

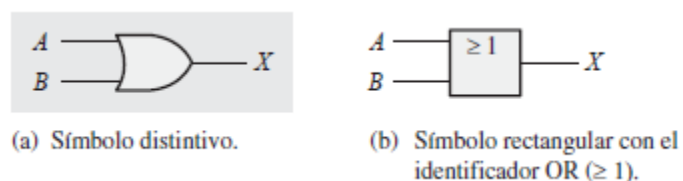


Fig. No 3. Símbolos lógicos estándar de la compuerta OR con dos entradas (estándar ANSI/IEEE 91-1984) [5].

4.1.1.3.4 Compuerta NAND

La compuerta NAND es una combinación entre la compuerta *And* y la compuerta inversora *NOT* de ahí su nombre NOT-AND, su función lógica está dada por la función AND con una salida complementada. El símbolo lógico distintivo de la compuerta NAND de dos entradas se muestra en la Figura No 4 junto símbolo rectangular.

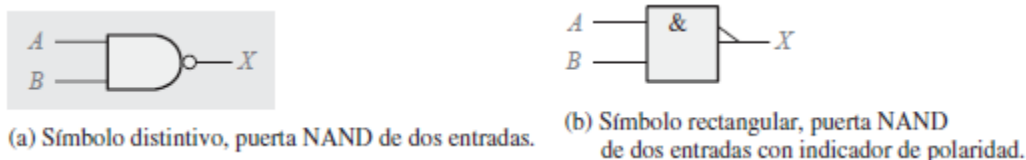


Fig. No 4. Símbolos lógicos estándar de la compuerta NAND (ANSI/IEEE 91 - 1984).

4.1.1.3.5 Compuerta NOR

La combinación entre la compuerta NOT y la OR dan origen a la compuerta NOR, de allí mismo procede su nombre NOT-OR e implica una función lógica OR con salida invertida. La figura No 5 muestra la compuerta NOR y su equivalente la compuerta OR seguida de la NOT.

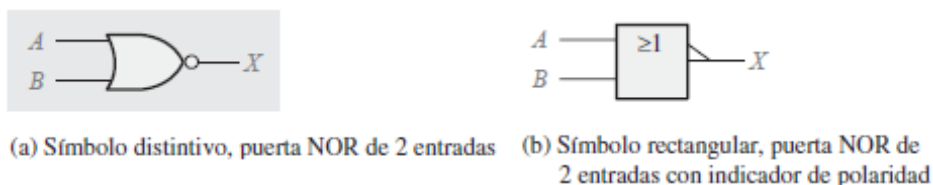


Fig. No 5. Símbolos lógicos estándar de la compuerta NOR (ANSI/IEEE 91 - 1984).

4.1.1.3.6 Compuertas Exclusivas XOR y XNOR

Las compuertas XOR y XNOR se componen de compuertas de las que se trató anteriormente, sin embargo, estas compuertas operan como elementos lógicos básicos que tienen su propio símbolo, debido a su utilización en muchas aplicaciones.

4.1.1.3.6.1 Compuerta OR-exclusiva o XOR

La figura No 6 muestra una compuerta XOR de entradas A , B y salida X . Esta puerta solo maneja dos entradas, su salida se pone a nivel ALTO solo cuando las dos entradas están a niveles lógicos opuestos.

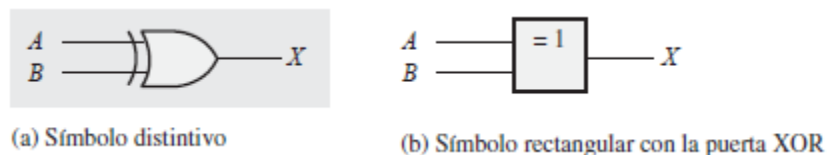


Fig. No 6. Símbolos lógicos estándar de la compuerta XOR [5].

4.1.1.3.6.2 Compuerta XNOR o NOR-exclusiva

Los símbolos estándar de la compuerta XNOR se observa en la figura No 7. Al igual que la compuerta XOR, la puerta XNOR tiene solo dos entradas. El círculo en la salida del símbolo indica que su salida es la opuesta a la compuerta XOR igual que el pequeño triángulo en la salida del símbolo rectangular XOR.

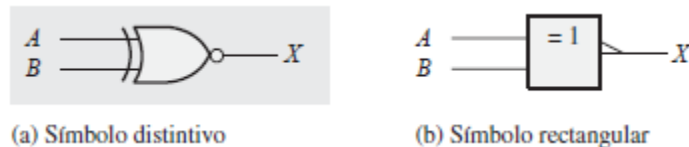


Fig. No 7. Símbolos lógicos estándar de la compuerta XNOR [5].

4.1.2 Álgebra de Boole

El álgebra de Boole es la base matemática de los sistemas digitales; definida por un conjunto de elementos, operadores y postulados. En el conjunto de elementos se encuentran dos estados $A = \{0,1\}$ por lo cual también es conocida como álgebra binaria. Los operadores binarios que utiliza son And (*), Or (+) y Not (') y diferentes postulados [8].

4.1.3 Elementos de programación

4.1.3.1 Eclipse

Es un ambiente de desarrollo integrado donde se puede crear, compilar y ejecutar programas; todo en un solo paquete, con una interfaz gráfica de usuario que es familiar para la mayoría de usuarios de computadoras. Eclipse tiene una variedad de características que lo hace muy útil para programadores principiantes. Para trabajar en Eclipse se necesita la versión 1.4 o superior, por lo que se puede programar en cualquier plataforma de computadora que soporte al menos la versión básica. Además, se requiere un *software* llamado JDK 5.0 o superior que es un *software* que provee herramientas de desarrollo para la creación de programas en Java y puede ser descargado e instalado en la computadora antes de instalar Eclipse [9].

4.1.3.2 Lenguaje de programación Java

Creado por ingenieros de la compañía SUN MICROSYSTEM en 1991, el lenguaje y plataforma de programación Java fue inspirado en las funcionalidades de lenguajes como C++, Eiffel, SmallTalk, Objective C, Cedar/Mesa, Ada, Perl. El resultado final es el idóneo para el desarrollo de aplicaciones seguras, distribuidas e independientes de la plataforma física o de *software*.

Con la realización de Java los usuarios obtuvieron un lenguaje de sintaxis sencillo, sin errores debido a un mecanismo de gestión de errores, orientado a objetos, los

cuales, permiten compilar y ejecutar las aplicaciones en tiempos muy cortos. Las aplicaciones gráficas son eficientes y seguras puesto que no se realiza ninguna manipulación u operación peligrosa. El auge de Java se dio por sus características, ya que en aquella época, al intentar descargar una aplicación resultaba difícil por sus condiciones, el tamaño de una aplicación realizada en Java resultaba poco voluminosa y rápida de descargar.

4.1.4 Evaluación de *software* educativo

Autores como Marqués (1995) consideran que se pueden contemplar tres aspectos fundamentales en la evaluación del *software*: aspectos técnicos, pedagógicos y funcionales.

El primero de ellos, aporta al análisis de elementos como el diseño de pantallas y la interfaz de comunicación. El pedagógico se refiere al fin con el que el *software* será utilizado, analizando objetivos educativos, contenidos teóricos y caminos pedagógicos que son considerados en una buena programación didáctica. Los funcionales permitirán evaluar qué ventajas le dan al profesor como material didáctico, facilidad de aprendizajes por parte de los estudiantes y que funciones del pensamiento favorece [4].

4.1.4.1 Evaluación externa

Alfred Bork (1986) la denomina como evaluación sumativa y es la evaluación del producto final. Realizada a los usuarios finales mediante preguntas cerradas y abiertas, contestadas luego de interactuar con el programa.

4.1.4.2 Instrumentos de evaluación

Generalmente, los instrumentos más usados son los cuestionarios de valoración, en los cuales, las respuestas son valoradas entre 0 y 5, siendo el resultado el grado de conformidad del usuario con las afirmaciones propuestas.

En la mayoría de cuestionarios se consideran aspectos claves como: el logro de los objetivos, aspectos técnicos, desarrollo de contenido, actividades y la documentación.

Como cada propuesta de evaluación está relacionada a un *software* en particular, se deben analizar los diferentes criterios de evaluación de estos medios didácticos; debido a las particularidades del sistema y al contexto educativo. La adaptación a cada contexto permite afirmar que “*no hay un instrumento de evaluación único, sino que el mismo será función del contexto de aplicación*”. (Zulma Cataldi, 2000 p.39).

4.1.4.3 Desarrollo del instrumento de evaluación

Las preguntas consideran aspectos de la interfaz de usuario y de los contenidos desarrollados. La valoración se da con una escala de calificación de 1 a 5 (siendo 5: excelente 4: muy bueno 3: bueno 2: regular 1: malo y 5: muy bueno 4: bueno 3: normal 2: malo 1: Muy malo, de acuerdo con el tipo de pregunta y el nivel del usuario), por lo anterior, se puede obtener un valor promedio de la calificación. Este valor permite obtener una puntuación de los aspectos tenidos en cuenta, para poder reformular o modificar aquellos que hayan tenido una puntuación menor que 2.5. En los casos de evaluación existe un espacio abierto para las sugerencias o reflexiones acerca de la aplicación.

4.1.4.3.1 Prototipo de los instrumentos de evaluación

Se presentan dos prototipos de aplicación: uno para los usuarios principiantes como estudiantes del grado once y el otro para experimentados como docentes y estudiantes de electrónica.

Se realizaron dos cuestionarios con preguntas pertinentes teniendo en cuenta los siguientes criterios: utilidad, técnicos, pedagógicos y didácticos. A continuación se presentan los esquemas propuestos para la evaluación de la aplicación móvil.

Utilidad	1. Facilidad de Uso
Pedagógicos y didácticos	2. Facilidad de exploración o navegación
	3. Claridad de contenidos
	4. Nivel de motivación
	5. ¿Le facilita la comprensión del tema?
Técnicos	6. ¿Son adecuados los recursos que necesita para aprender el tema?

Tabla No 2. Esquema de evaluación de la aplicación a estudiantes.

Al final de los cuestionarios se pide a los usuarios que expresen sus comentarios o sugerencias que tengan acerca de la aplicación mediante una pregunta abierta.

Utilidad	Facilidad de Uso
	Organización general
Pedagógicos y didácticos	Simulador de compuertas Lógicas
	Módulo Fundamentos teóricos
	Módulo Datos generales
	Contenido Teórico

Tabla No 3. Esquema de evaluación de la aplicación a docentes y estudiantes de electrónica.

4.2 Marco Conceptual

4.2.1 Sistema binario

El sistema binario o de base 2 utiliza dos cifras para representar un número, estas cifras son 0 y 1. Estas cifras suelen utilizarse de la misma manera que en el sistema decimal, donde la posición de cada dígito en el sistema binario indica la potencia de 2 por la que se multiplican las cifras 0 o 1. El sistema de dos estados 0 o 1 son utilizados para indicar verdadero o falso, alto o bajo y otras condiciones. En los circuitos eléctricos, los estados suelen representarse mediante tensiones [3].

4.2.2 Tabla de verdad

La tabla de verdad es una herramienta que describe la forma en que unos niveles lógicos de entrada de un circuito lógico generan un nivel lógico de salida.

4.2.3 Sistema Operativo Móvil

Es un sistema operativo más simple que el utilizado en los computadores que controla un dispositivo móvil y están orientados a la conectividad inalámbrica, formatos multimedia para móviles e intercambio de datos entre móviles [10].

4.2.4 Sistema Operativo Android

Android es un sistema operativo móvil basado en Linux, diseñado principalmente para dispositivos móviles como Smartphone, tablets, entre otros. Es desarrollado por el Open Handset Alliance la cual es liderada por Google [10].

4.2.5 Interfaz de usuario

Marcos Mora (2004) define interfaz como: *“la presentación en pantalla que un sistema informático ofrece al usuario para que este pueda interactuar con él”*.

4.2.6 Usabilidad

La Organización Internacional para la Estandarización ISO/IEC 9126 define usabilidad como: *“La capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso”*.

4.2.7 Evaluar

Es determinar el grado de adecuación, ya sea durante el desarrollo para hacer las modificaciones (formativa) o posterior a él mediante experiencias de uso (sumativa).

4.2.8 Exploración

Recorrido que se hace en una computadora, red o *software* para conocer o descubrir que hay en él.

5 Metodología

Para la realización del proyecto se eligió la “*metodología en Cascada*” utilizada para el desarrollo de software, debido a las fases con las que cuenta para la ejecución de este tipo de trabajos. Esta metodología es una de las más antiguas, propuesta por Winston W. Royce en el año de 1970. Se da un proceso secuencial de avance en el que cada una de sus etapas debe finalizar para poder comenzar la otra; el proceso es similar al realizado en una cascada de agua o escalera, que va de arriba hacia abajo. Las etapas que realiza esta metodología están definidas de la siguiente manera: Comunicación, diseño, codificación, pruebas, integración y finalmente Despliegue [11].

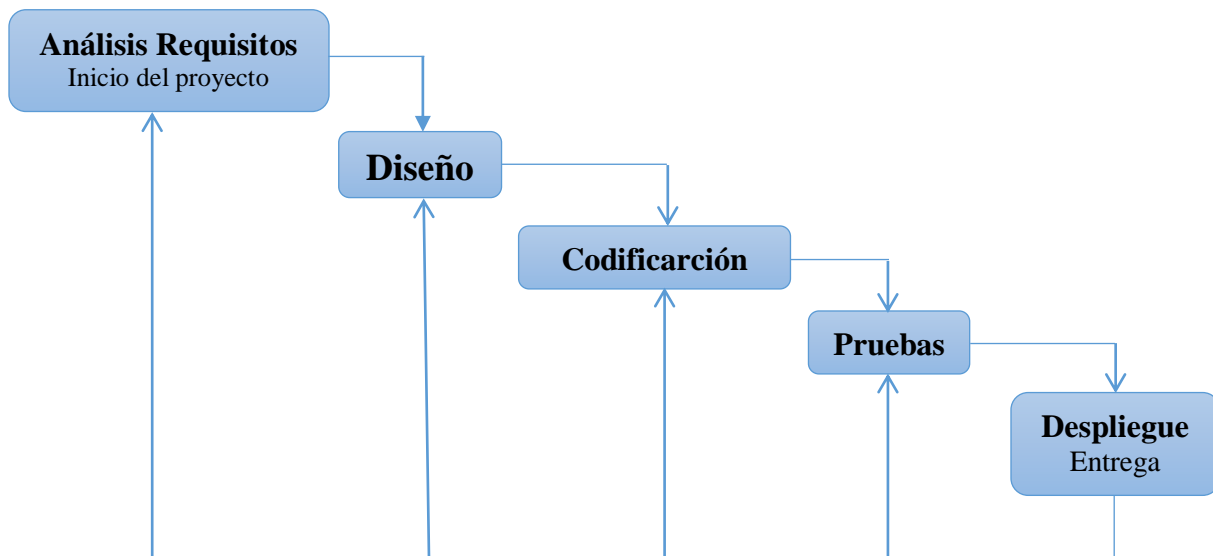


Fig. No 8. Modelo “Cascada” para el desarrollo de software. El proceso fluye de arriba hacia abajo y tiene retroalimentación.

Los principios básicos del modelo de cascada son los siguientes:

- El proyecto debe estar dividido en fases secuenciales donde al culminar una fase se puede empezar la siguiente.
- Se debe realizar un proyecto o anteproyecto inicial.
- Se debe realizar una buena documentación escrita.

Análisis Requisitos

Se analizan las necesidades y en función de estas se crean unos objetivos, realizándose un documento donde se especifiquen los requisitos.

Diseño

El sistema se descompone y organiza para ser realizado por separado. De este diseño surge la descripción de la estructura del sistema y la especificación de cada una de sus partes, así como la forma en que se combinan una con otras.

En esta fase se seleccionan las herramientas necesarias para la etapa de codificación y los algoritmos necesarios para el cumplimiento de los requisitos del software.

Codificación

Se realiza el código fuente, haciendo uso de prototipos así como de pruebas y ensayos para corregir errores.

Pruebas

Los elementos programados por separado son probados y unificados para componer el sistema y realizar la entrega al usuario final que ejecuta el sistema, teniendo en cuenta, que los programadores ya realizaron exhaustivas pruebas y no tendrá ningún problema.

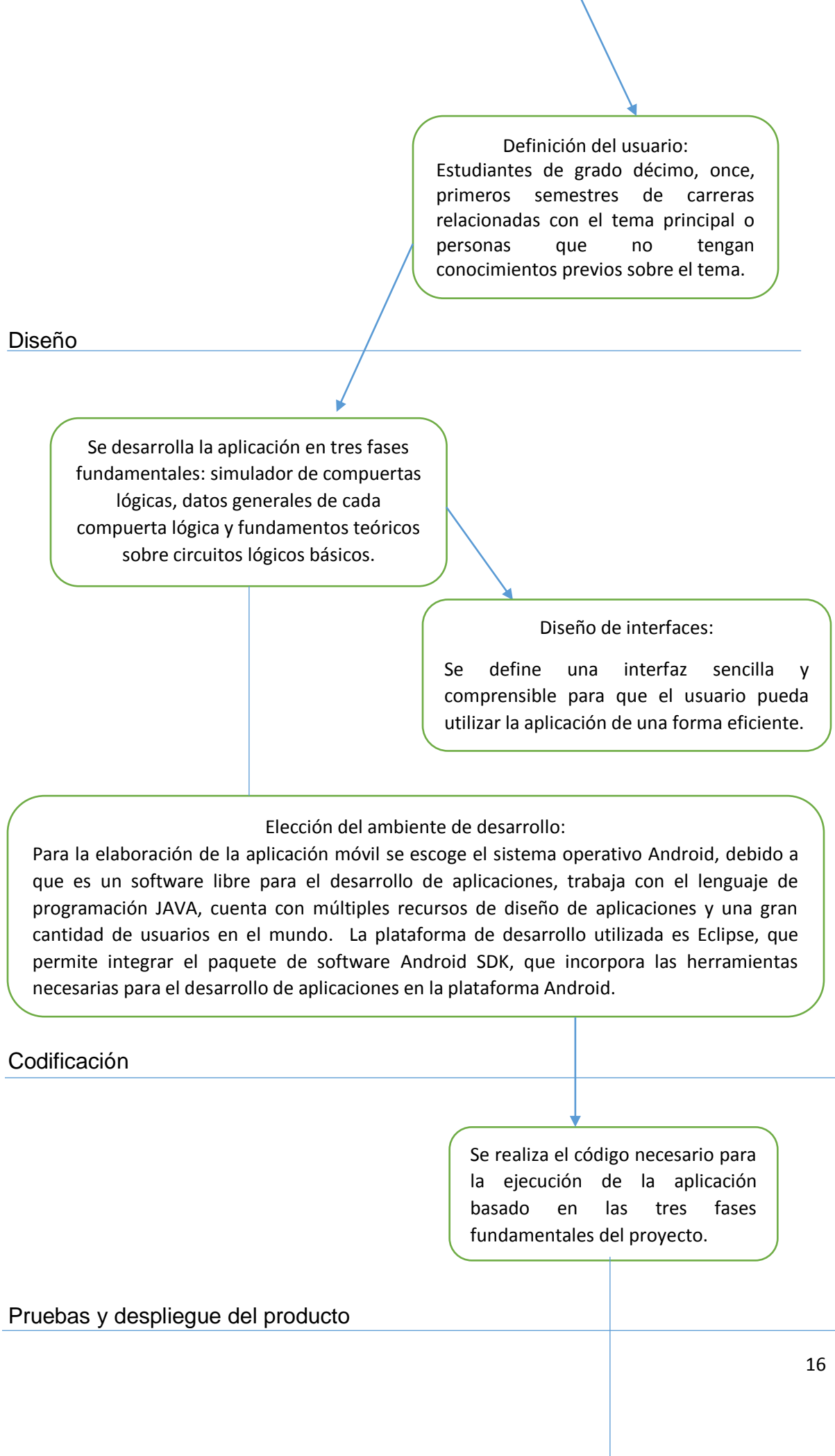
Despliegue

Se realizan los ajustes necesarios para que cumpla con todos los requisitos realizados en la etapa de análisis y se realiza la entrega final del software.

Aplicación del esquema general del modelo en Cascada al proyecto.

Determinar la necesidad del software:
En el ámbito de los circuitos lógicos básicos se reconoce la necesidad de realizar una aplicación móvil que permita al usuario obtener información, simulación, comprensión y entendimiento de estos sistemas.

Análisis y delimitación del tema:
En el desarrollo de la aplicación móvil de circuitos lógicos básicos, se plantean los siguientes objetivos particulares de trabajo: Realizar un módulo de simulación de compuertas lógicas, realizar un módulo de fundamentos teóricos sobre circuitos lógicos básicos y elaborar un módulo de datos generales de cada compuerta lógica.



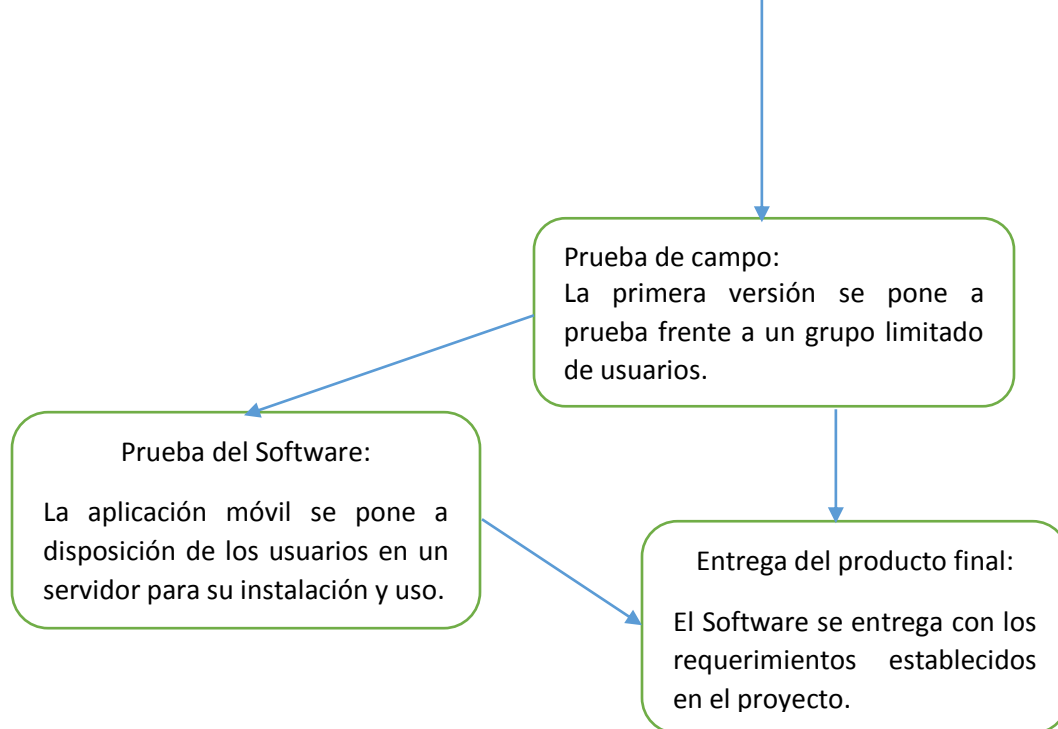


Fig. No 9. Aplicación del esquema general del método en Cascada al proyecto.

CAPITULO III

Desarrollo del proyecto

6 Análisis de requisitos

Un requisito es una característica observable que satisface una necesidad o expresa una restricción que afecta al sistema que se está desarrollando [17]. Por esta razón, se debe cumplir con los requisitos para alcanzar los objetivos propuestos y desarrollar el producto de la forma planteada.

Los requisitos se generan de varias maneras, una de ellas se da cuando el cliente solicita determinado software y la otra es cuando se deben recoger a los usuarios finales del sistema. En este caso se indagará a través de las aplicaciones similares y algunos usuarios relacionados con el tema como son: cuáles pueden ser algunos requisitos y qué puede tener la aplicación; dentro de este proceso se debe determinar el usuario final que utilizará la aplicación móvil, en la siguiente sección se define el perfil de este usuario.

6.1 Perfil del Usuario final

En este proyecto se define un único actor del sistema, que es el que utiliza la aplicación móvil sobre circuitos lógicos básicos.

El perfil del usuario final se enmarca sobre los estudiantes de circuitos lógicos básicos que se encuentran generalmente en los cursos finales de bachillerato (grados décimo y once), primer semestre de carreras que manejan este tema o personas que desean incursionar en él.

El usuario tiene el control total del sistema, donde puede navegar por él e interactuar con las actividades con las que cuenta la aplicación.

6.2 Requisitos de usuario

Existen varios métodos para obtener los requisitos, la más básica es pedir a los usuarios mediante alguna técnica la lista predefinida de requisitos. Los actores directos o indirectos involucrados en la aplicación son llamados *Stakeholders* que son definidos como cliente o usuarios interesados en la culminación del proyecto y que proporcionan los prerrequisitos. Después de seleccionar y definirlos requisitos entre Stakeholders y programadores se deben documentar; existen diferentes estilos de documentación que se pueden observar en la figura No 12.

Estilos de documentación:

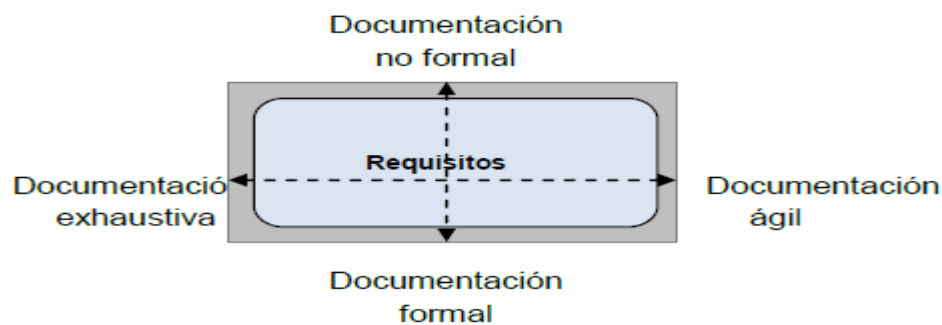


Fig. No 10. Estilos de documentación para la recolección de requisitos [18].

En el eje horizontal se define una documentación ágil o exhaustiva y en el horizontal formal o no formal. Debido a que la prioridad es la agilización del proceso, se documenta lo mínimo necesario y se realizan otros procesos de comunicación para recopilar estos requisitos y avanzar en el progreso del proyecto.

En este caso se selecciona una documentación ágil y no formal basada en los objetivos particulares. En este proyecto se propuso tres módulos principales dedicados todos a los circuitos lógicos básicos debido a la falta de aplicaciones móviles que integren un aspecto teórico y práctico a la vez.

6.3 Requisitos del sistema

Los requisitos del sistema se clasifican en dos grupos, los funcionales que se refieren a la funcionalidad del sistema y los no funcionales, que son las cualidades o restricciones que debe cumplir el sistema.

6.3.1 Requisitos funcionales

Inicio de la aplicación: Se presenta un menú principal que permite al usuario dirigirse a cualquiera de los módulos propuestos en la aplicación.

Por cada módulo existirá un botón, de esta manera, serán tres: el primero se dirigirá al manual de uso y al botón del simulador, el segundo permitirá acceder a los fundamentos teóricos sobre circuitos lógicos básicos y finalmente el último permitirá ir a un módulo de datos generales sobre compuertas lógicas digitales.

Módulo simulador

Los simuladores son herramientas de apoyo en los procesos de aprendizaje, con el fin que los usuarios o estudiantes tengan la oportunidad de poner en práctica los conocimientos adquiridos en las aulas de clase o en otro lugar. Ofreciendo una innovadora forma de perfeccionar habilidades, conceptos, estrategias y decisiones en un entorno virtual controlado que permitirá el error en estos ambientes [16].

Al pulsar el botón del simulador en el menú principal se podrá acceder al manual de uso, le indicará al usuario como utilizarlo y un botón que lo dirige al simulador de compuertas lógicas directamente.

El módulo del simulador se encargará de permitir al usuario acceder a 7 compuertas lógicas diferentes que podrá simular de forma individual o conectarlas entre sí y probar su funcionamiento.

Se realizará un área para ubicar, quitar o cambiar las compuertas lógicas y realizar circuitos lógicos combinatorios.

El módulo contará con diferentes opciones de usuarios, algunas que ya se mencionaron y otras que se indican a continuación:

- ✓ Selección y ubicación de compuertas lógicas.
- ✓ Cambiar o borrar las compuertas lógicas.
- ✓ Cambiar los valores de entrada de cada compuerta lógica.

- ✓ Simular una o varias compuertas lógicas.
- ✓ Conectar o desconectar las compuertas que se encuentran en el área de ubicación.

El objetivo de este módulo es facilitar la comprensión y entendimiento de las compuertas lógicas, desde la identificación de cada operación y la realización de circuitos lógicos combinatorios.

Módulo de fundamentos sobre circuitos lógicos

Al ser pulsado el botón de fundamentos teóricos sobre circuitos lógicos básicos en el menú principal, el usuario accederá a diferentes pantallas en las cuales se encuentra la información consultada de diferentes libros.

Este módulo permite encontrar información concreta sobre los circuitos lógicos básicos. Los temas que se pueden encontrar son los siguientes:

- Introducción
- Operaciones Lógicas Básicas
- Compuertas Lógicas
- Simbología Lógica
- Lógica de función fija
- Principios del Álgebra de Boole
- Bibliografía

Cada tema tendrá un botón de acceso y se podrá navegar por cada uno de ellos.

El contenido de este módulo está fundamentado en los postulados y planteamientos de los autores de los siguientes libros: *Fundamentos de sistemas digitales*, *Fundamentos de lógica digital*, *Sistemas digitales*, *Principios y aplicaciones*, entre otros, son necesarios para que el usuario incursione en el campo de los circuitos lógicos básicos. Cabe aclarar, que la aplicación es realizada y enfocada hacia personas que no manejen conocimientos previos sobre el tema o que están empezando a conocer el mismo.

Módulo de datos generales

Al pulsar el botón de datos generales en el menú principal se accederá a una serie de botones que permiten encontrar en la compuerta lógica seleccionada la siguiente información:

- ✓ El símbolo lógico estándar
- ✓ La tabla de verdad de la compuerta lógica
- ✓ Esquema del chip, referencia y su descripción
- ✓ El chip comercial

Este módulo será creado para que el usuario pueda encontrar de manera rápida y fácil estos datos, información útil en el momento de realizar un circuito electrónico en el laboratorio o cuando se requiera, la elaboración de un esquema de un circuito lógico.

6.3.2 Requisitos no funcionales

Como ya se mencionó, hacen referencia a las restricciones y cualidades sobre la solución del desarrollo del sistema. La plantilla Volere ofrece una lista predefinida de requisitos no funcionales [20], que se mencionan a continuación:

1. De presentación: referente a la parte visual.
 - En la cabecera de la aplicación se encontrará el logotipo de la aplicación.
2. De usabilidad y humanidad: referente con un sistema usable.
 - En el módulo del simulador de compuertas lógicas se encuentra un manual de uso, para que el usuario comprenda su funcionamiento.
3. De cumplimiento: referente a requisitos de cumplimiento de velocidad, fiabilidad, entre otros.
 - Según las características de hardware del dispositivo móvil la aplicación responderá rápidamente.
4. Operacionales y de entorno: entorno donde se usa el software.

- La aplicación está disponible para móviles con sistema operativo Android.
- La versión mínima de instalación en Android es la 4.0.3 – API 15 (Ice Cream Sandwich).
- Las versiones validas son 4.0.3 o superiores.

5. De mantenimiento y soporte: referente a mantenimiento y evolución.

- Las actualizaciones se realizan y renuevan en el servidor donde se encuentra la aplicación.

6. Culturales y políticos:

- La aplicación se encuentra en idioma español.

6.4 Especificación de casos de uso

Un caso de uso describe el funcionamiento del sistema en diferentes situaciones planteadas por el usuario, es decir, cuando se da la interacción entre estos. Esta descripción se puede dar por textos narrativos, un esquema de tareas, una representación por medio de diagramas [21].

En este caso, se utilizará el diagrama de casos de uso de UML, utilizado para mostrar visualmente cuáles son los actores que utilizan el sistema y en qué casos se pueden ejecutar.

Se debe tener en cuenta que en el software propuesto en este trabajo interviene un único actor que es el que finalmente utiliza e interactúa con la aplicación móvil.

Al igual que en los requisitos funcionales, los casos de uso serán descritos por cada módulo. Es preciso aclarar que los casos de uso no contienen información de la funcionalidad del sistema, sino que cumplen otra función.

Por cada módulo se detallará el diagrama UML del caso de uso con el actor y el tipo de relación entre ellos, junto a la descripción textual de cada caso.

6.4.1 Caso de uso del Menú principal

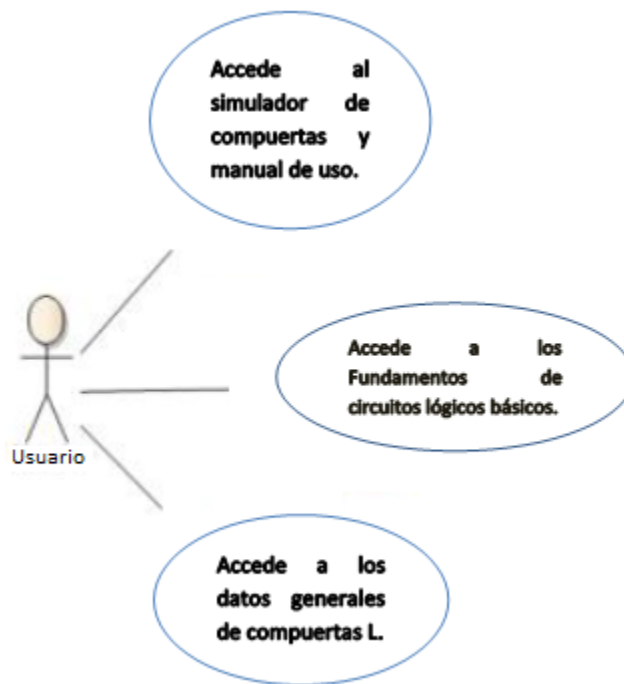


Fig. No 11. Caso de uso del Menú principal de la aplicación móvil.

Caso de uso 1	Acceder al simulador de compuertas lógicas y manual de uso.
Resumen de la funcionalidad	Permite al usuario ingresar al manual de uso del simulador y al simulador de compuertas lógicas básicas.
Casos de uso relacionados	
Actores	Usuario
Precondición	Haber iniciado la aplicación.
Post-condición	La aplicación redirigirá al usuario a la pantalla del manual de uso y el botón del simulador.
Escenario de éxito	<ol style="list-style-type: none"> 1. El sistema muestra el menú principal 2. El usuario selecciona la opción del simulador. 3. Ingresa al manual y al simulador.
Extensiones	<ol style="list-style-type: none"> 1. El usuario elige salir de la aplicación. <ol style="list-style-type: none"> 1.1. Se finaliza el caso de uso. 2. El usuario decide dirigirse a otro módulo.

Tabla No 4. Caso de Uso “Acceder al simulador de compuertas lógicas y manual de uso”.

Caso de uso 2	Acceder al módulo de fundamentos de circuitos lógicos básicos.
Resumen de la funcionalidad	Permite al usuario ingresar al módulo de

	fundamentos de circuitos lógicos básicos y encontrar una serie de temas que le servirán al usuario para comprender el funcionamiento y características de estos sistemas.
Casos de uso relacionados	
Actores	Usuario
Precondición	Haber iniciado la aplicación.
Post-condición	La aplicación redirigirá al usuario a la pantalla del contenido del módulo de fundamentos de circuitos lógicos básicos.
Escenario de éxito	<ol style="list-style-type: none"> 1. El sistema muestra el menú principal 2. El usuario selecciona la opción fundamentos de circuitos lógicos básicos. 3. Ingresa al contenido teórico de circuitos lógicos básicos.
Extensiones	<ol style="list-style-type: none"> 1. El usuario elige salir de la aplicación. <ol style="list-style-type: none"> 1.1. Se finaliza el caso de uso. 2. El usuario decide dirigirse a otro módulo.

Tabla No 5. Caso de Uso “Acceder al módulo de fundamentos de circuitos lógicos básicos”.

Caso de uso 3	Acceder al módulo de datos generales sobre compuertas lógicas.
Resumen de la funcionalidad	Permite al usuario ingresar al módulo de datos generales sobre compuertas lógicas y encontrar un menú que contiene cada una de las compuertas lógicas para poder seleccionar una y visualizar los datos generales de esta.
Casos de uso relacionados	
Actores	Usuario
Precondición	Haber iniciado la aplicación.
Post-condición	La aplicación redirigirá al usuario a la pantalla del menú sobre compuertas lógicas.
Escenario de éxito	<ol style="list-style-type: none"> 1. El sistema muestra el menú principal 2. El usuario selecciona la opción datos generales de compuertas lógicas. 3. Ingresa al menú de compuertas lógicas.
Extensiones	<ol style="list-style-type: none"> 1. El usuario elige salir de la aplicación. <ol style="list-style-type: none"> 1.1. Se finaliza el caso de uso. 2. El usuario decide dirigirse a otro módulo.

Tabla No 6. Caso de Uso “Acceder al módulo sobre datos generales de compuertas lógicas”.

6.4.2 Caso de uso del módulo simulador

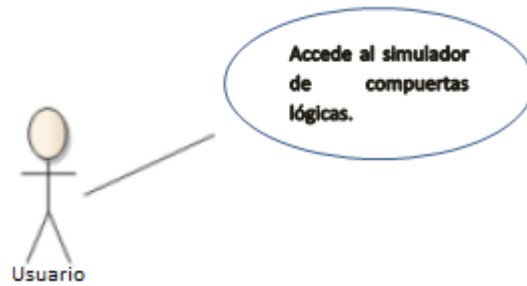


Fig. No 12. Caso de uso del módulo simulador.

Caso de uso 4	Acceder al simulador de compuertas lógicas.
Resumen de la funcionalidad	Permite al usuario ingresar al simulador de compuertas lógicas y acceder a todas sus funcionalidades.
Casos de uso relacionados	Acceder al simulador de compuertas lógicas y manual de uso.
Actores	Usuario
Precondición	Encontrarse en el módulo del simulador de compuertas lógicas.
Post-condición	La aplicación redirigirá al usuario al simulador de compuertas lógicas.
Escenario de éxito	<ol style="list-style-type: none"> 1. El sistema muestra el botón del simulador y el manual de uso. 2. El usuario selecciona la opción del simulador. 3. Ingresa al simulador de compuertas lógicas.
Extensiones	<ol style="list-style-type: none"> 1. El usuario elige salir de la aplicación. 2. El usuario decide volver al menú principal. 3. El usuario decide leer el manual de uso. <p>a) Se finaliza el caso de uso.</p>

Tabla No 7. Caso de Uso "Acceder al simulador de compuertas lógicas".

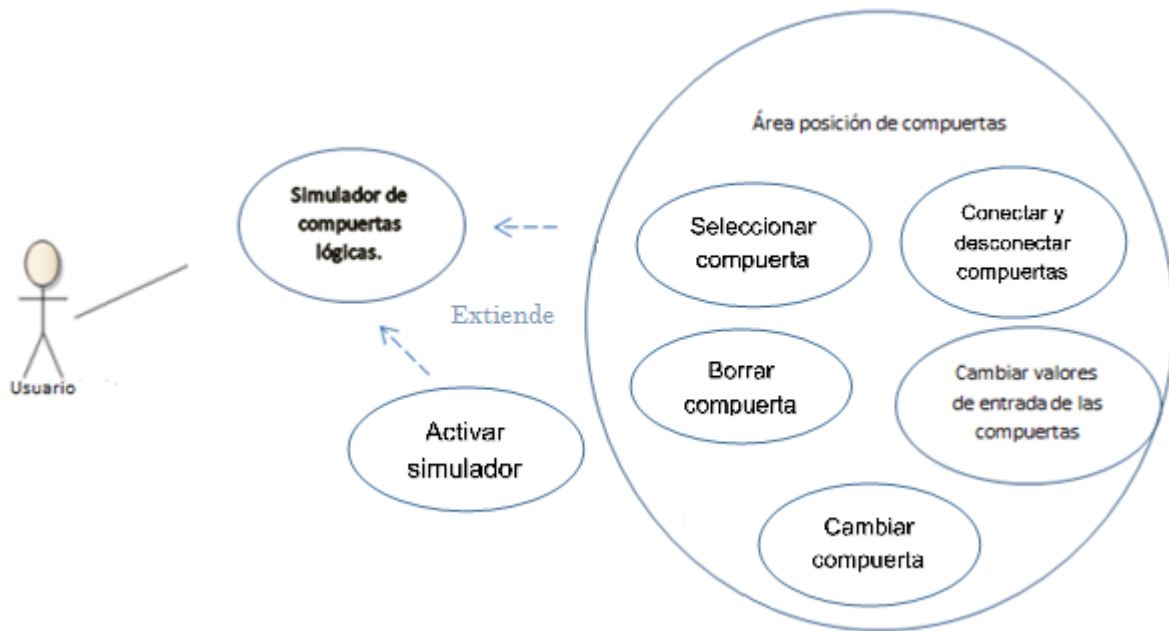


Fig. No 13. Caso de uso. Simulador de compuertas lógicas.

Caso de uso 5	Simulador de compuertas lógicas.
Resumen de la funcionalidad	Permite simular las compuertas lógicas de forma individual o conectarlas entre sí y observar su comportamiento.
Casos de uso relacionados	Acceder al simulador de compuertas lógicas.
Actores	Usuario
Precondición	El usuario debe encontrarse en el módulo del simulador.
Post-condición	Interactuar con el simulador de compuertas lógicas.
Escenario de éxito	<ol style="list-style-type: none"> 1. El sistema muestra el botón del simulador. 2. El usuario selecciona la opción del simulador. 3. El usuario simula las compuertas lógicas.
Extensiones	<ol style="list-style-type: none"> 1. El usuario elige salir de la aplicación. 2. El usuario decide volver al inicio del módulo del simulador. <ol style="list-style-type: none"> a) Se finaliza el caso de uso.

Tabla No 8. Caso de Uso "Simulador de compuertas lógicas".

Caso de uso 6	Seleccionar compuerta.
Resumen de la funcionalidad	El usuario debe escoger un lugar en el área donde se posicionan las compuertas para su simulación, seguido de la selección de la compuerta que quiere poner en esa posición.
Casos de uso relacionados	Acceder al simulador de compuertas lógicas.
Actores	Usuario
Precondición	Encontrarse en el módulo simulador.
Post-condición	Tener ubicada una compuerta lógica para su simulación.
Escenario de éxito	<ol style="list-style-type: none"> 1. El usuario escoge el lugar. 2. El usuario escoge la compuerta.
Extensiones	<ol style="list-style-type: none"> 1. El usuario escoge la compuerta y no el lugar o posición. 2. El usuario escoge el lugar o posición y no la compuerta.

Tabla No 9. Caso de Uso “Selecciona posición y escoge compuerta”.

Caso de uso 7	Borrar compuerta.
Resumen de la funcionalidad	Al encontrarse una compuerta en el área de posicionamiento de simulación se selecciona la compuerta y es borrada.
Casos de uso relacionados	Acceder al simulador de compuertas lógicas.
Actores	Usuario
Precondición	Encontrarse una o varias compuertas en el área de simulación.
Post-condición	Borrar una compuerta.
Escenario de éxito	<ol style="list-style-type: none"> 1. El usuario escoge la compuerta en el área de simulación. 2. El usuario pulsa el botón borrar.
Extensiones	<ol style="list-style-type: none"> 1. El usuario cambia la compuerta por otra compuerta.

Tabla No 10. Caso de Uso “Borrar compuerta”.

Caso de uso 8	Conectar y desconectar compuertas.
Resumen de la funcionalidad	<ol style="list-style-type: none"> 1. Al encontrarse más de una compuerta en el área de simulación, el usuario selecciona la salida de una compuerta y escoge la entrada de otra compuerta para conectarlas. 2. Para desconectarlas se pulsa la entrada de la compuerta conectada y se borra.
Casos de uso relacionados	Acceder al simulador de compuertas lógicas.
Actores	Usuario
Precondición	<ol style="list-style-type: none"> 1. Seleccionarla salida de una compuerta. 2. Seleccionar la entrada de la compuerta conectada.
Post-condición	<ol style="list-style-type: none"> 1. Se selecciona la entrada de una compuerta. 2. Se borra la conexión.
Escenario de éxito	<ol style="list-style-type: none"> 1. Se conectan dos compuertas. 2. Se borra una conexión.
Extensiones	<ol style="list-style-type: none"> 1. La resolución de pantalla de la aplicación, no es compatible con la del móvil.

Tabla No 11. Caso de Uso "Conectar y desconectar compuertas".

Caso de uso 9	Cambiar compuerta.
Resumen de la funcionalidad	El usuario selecciona la compuerta que desea cambiar y la cambia por otra compuerta.
Casos de uso relacionados	Acceder al simulador de compuertas lógicas.
Actores	Usuario
Precondición	Seleccionar la compuerta que desea cambiar.
Post-condición	Contar con otra compuerta.
Escenario de éxito	<ol style="list-style-type: none"> 1. El usuario escoge la compuerta que desea cambiar en el área de simulación. 2. El usuario escoge la compuerta que quiere poner.
Extensiones	<ol style="list-style-type: none"> 2. El usuario escoge la compuerta y la borra.

Tabla No 12. Caso de Uso "Cambiar compuerta".

Caso de uso 10	Cambiar valores de entrada de la compuerta.
Resumen de la funcionalidad	El usuario debe tener una o varias compuertas en el área de simulación y pulsar una de las entradas de cualquier compuerta para cambiar su valor de 0 a 1 y de 1 a 0.
Casos de uso relacionados	Acceder al simulador de compuertas lógicas.
Actores	Usuario
Precondición	Tener una o varias compuertas lógicas en el área de simulación.
Post-condición	Haber cambiado el valor de entrada de la compuerta lógica de 0 a 1 o de 1 a 0.
Escenario de éxito	<ol style="list-style-type: none"> 1. El usuario pulsa una de las entradas de una compuerta lógica. 2. Cambia el valor de la entrada de 0 a 1 o de 1 a 0.
Extensiones	<ol style="list-style-type: none"> 1. El usuario había pulsado la salida de una compuerta lógica.

Tabla No 13. Caso de Uso “Cambio valores de entrada de la compuerta”.

Caso de uso 11	Activar simulador
Resumen de la funcionalidad	Al encontrarse una o más compuertas, conectadas o no, en el área de simulación, el usuario activa el simulador para activar la operación de las compuertas lógicas y observar el resultado de salida.
Casos de uso relacionados	Acceder al simulador de compuertas lógicas.
Actores	Usuario
Precondición	Tener una o varias compuertas lógicas, conectadas o no, en el área de simulación.
Post-condición	Observar el resultado de salida de las compuertas puestas en el área de simulación.
Escenario de éxito	<ol style="list-style-type: none"> 3. Al encontrarse una o varias compuertas lógicas, conectadas o no, se observa la respuesta de salida de estas compuertas lógicas.
Extensiones	<ol style="list-style-type: none"> 2. El usuario no activa el simulador.

Tabla No 14. Caso de Uso “Activar el simulador”.

6.4.3 Caso de uso del módulo sobre fundamentos de circuitos lógicos básicos

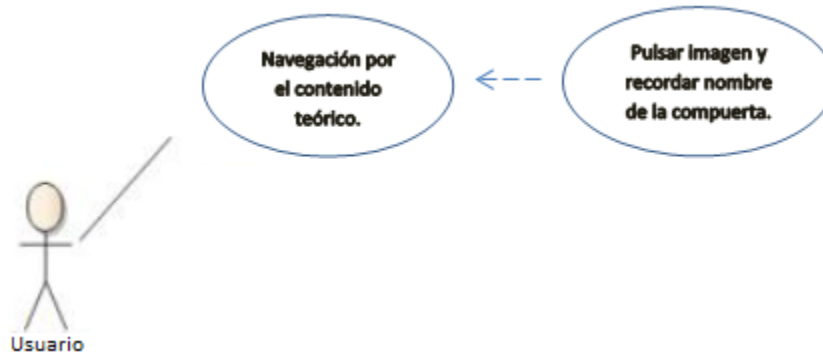


Fig. No 14. Caso de uso. Fundamentos de circuitos lógicos básicos.

Caso de uso 12	Navegación por el contenido teórico de fundamentos de circuitos lógicos básicos.
Resumen de la funcionalidad	Permite al usuario encontrar un menú sobre fundamentos de circuitos lógicos básicos y navegar por cada uno de los temas de adelante hacia atrás y viceversa.
Casos de uso relacionados	Acceder al módulo de fundamentos de circuitos lógicos básicos.
Actores	Usuario
Precondición	Haber ingresado al módulo de fundamentos de circuitos lógicos básicos.
Post-condición	Leer el contenido teórico de cada uno de los temas que conforman el módulo de fundamentos de circuitos lógicos básicos.
Escenario de éxito	<ol style="list-style-type: none"> 1. Ingresar al módulo de fundamentos de circuitos lógicos básicos. 2. Navegar por el contenido teórico.
Extensiones	<ol style="list-style-type: none"> 1. El usuario elige salir de la aplicación. 2. El usuario decide volver al menú principal.

Tabla No 15. Caso de Uso "Navegación por el contenido teórico de fundamentos de circuitos lógicos básicos".

Caso de uso 13	Pulsar imagen y recordar nombre de la compuerta.
Resumen de la funcionalidad	En uno de los temas sobre fundamentos de circuitos lógicos básicos, específicamente en “ <i>Simbología lógica</i> ” se encuentran una serie de imágenes que contienen los símbolos estándar de las compuertas lógicas. En el momento de ser pulsada una de estas imágenes, se verá un mensaje por pantalla del nombre de la compuerta.
Casos de uso relacionados	Navegación por el contenido teórico de fundamentos de circuitos lógicos básicos.
Actores	Usuario
Precondición	Encontrarse en el punto sobre <i>Simbología lógica</i> en el módulo sobre fundamentos de circuitos lógicos básicos.
Post-condición	Observar el nombre por pantalla de la imagen de la compuerta lógica pulsada.
Escenario de éxito	1. Pulsar la imagen y ver el nombre de la compuerta.
Extensiones	1. El usuario elige salir de la aplicación. 2. El usuario decide ir a otro punto del contenido.

Tabla No 16. Caso de Uso “Pulsar imagen y recordar nombre de la compuerta”.

6.4.4 Caso de uso del módulo sobre datos generales de compuertas lógicas.

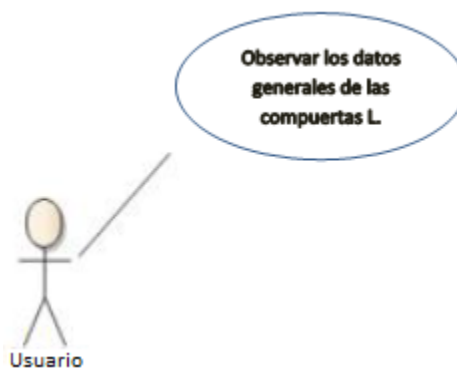


Fig. No 15. Caso de uso. Datos generales de compuertas lógicas.

Caso de uso 14	Observar los datos generales de las compuertas lógicas.
Resumen de la funcionalidad	Al escoger uno de los botones con el nombre de la compuerta lógica en el menú que se encuentra en el módulo de datos generales, el usuario podrá observar el símbolo lógico, la tabla de verdad, el esquema y el chip de la compuerta escogida.
Casos de uso relacionados	Acceder al módulo de datos generales sobre compuertas lógicas.
Actores	Usuario
Precondición	Encontrarse en el módulo sobre datos generales de compuertas lógicas.
Post-condición	Observar los datos generales de la compuerta escogida.
Escenario de éxito	1. Obtener los datos generales.
Extensiones	1. El usuario elige salir de la aplicación. 2. El usuario decide volver al menú principal. a) Se finaliza del caso de uso.

Tabla No 17. Caso de Uso "Pulsar imagen y recordar nombre de la compuerta".

7 Diseño

7.1 Finalidad del proyecto

En la actualidad las aplicaciones móviles han tenido un desarrollo importante, puesto que la sociedad está dando mayor relevancia tanto a la información como al conocimiento, brindando nuevas posibilidades a los usuarios de adquirir un conocimiento derivado de la interpretación y la contextualización de información, a la cual se puede acceder a través de nuevos medios tecnológicos y el uso cada vez más fácil de las tecnologías de la información y la comunicación, actualmente llamadas TIC.

Estas tecnologías enfocadas al uso educativo cambian los procesos y métodos de enseñanza-aprendizaje y se convierten en un soporte fundamental para la educación; permitiendo llegar a más estudiantes y centrarse en las necesidades de un solo individuo a través de la interacción con pequeños dispositivos.

Estos dispositivos permiten que los usuarios reciban diferentes servicios como comunicación, software y acceso a una red informática, facultando la concepción y establecimiento de un nuevo paradigma educativo, el Mobile Learning. Este modelo da continuidad a procesos educativos a través de pequeños dispositivos, que tienen casi las mismas características que un computador.

Teniendo en cuenta esta finalidad se debe dar una contextualización de lo que en algunos casos se llama aprendizaje electrónico móvil o como es llamado en inglés E-learning. Quinn (2000), afirma que el Mobile Learning es el aprendizaje a través de dispositivos móviles, la intersección de la computación móvil y el aprendizaje, la cual se caracteriza por la capacidad para acceder a recursos de aprendizaje desde cualquier momento y lugar, con la disposición de muchos recursos de interacción entre software y usuario.

El Mobile learning También es definido como una nueva forma de educación a partir de la unión entre el modelo e-learning y los dispositivos móviles inteligentes, que permiten obtener a través de una conexión permanente o intermitente a una red, información clara y precisa de lo que se requiere saber.

Dada algunas definiciones, e-learning se puede entender como un modelo tecnológico – pedagógico, que atiende necesidades urgentes de aprendizaje, y posibilita la interactividad para enseñar y aprender en espacios virtuales. Mirando hacia el futuro, se considera que el mercado del Mobile Learning aumente debido al crecimiento mundial de la telefonía móvil y su integración con la computación móvil. Las empresas cada vez más desarrollan software educativo en estos dispositivos y dotan a sus empleados para mejorar los procesos y resultados en el mercado.

La integración de esta tecnología a la educación no es una ilusión, ya que los dispositivos existen; la asimilación y receptividad es una realidad; con el tiempo se integran nuevas funciones y se mejoran sus características y las inversiones necesarias son mínimas.

La aplicación planteada en este proyecto pretende realizar un acercamiento a la producción de contenidos educativos para dispositivos móviles que sirve como apoyo en los procesos presenciales del aula. Y permitir a cualquier usuario adquirir información en cualquier momento o lugar; sabiendo que cada persona aprende a ritmos y momentos diferentes.

Los dispositivos móviles brindan la posibilidad de enriquecer los contenidos, actividades y estrategias que utiliza el docente, gracias a la diversidad de recursos y servicios con los que estos cuentan. Teorías como COLL (Contextual life-long learning) descrita por Mike Sharples profesor de tecnología de la Universidad Abierta del Reino Unido, asegura que es una reconceptualización del aprendizaje donde el conocimiento es entendido como una enseñanza que no está sujeta a un lugar o sitio dentro del desarrollo profesional vocacional y que pasa a ser un proceso dinámico, individual e interactivo de los estudiantes [14].

El contenido de la aplicación móvil planteada en este documento es una mezcla entre elementos estáticos e interactivos en los cuales se pueden encontrar: imágenes, documentos de texto, actividades y un simulador. La intencionalidad pedagógica del contenido se fundamenta en apoyar, complementar y aportar una herramienta didáctica basada en una aplicación nativa que solo necesita una

conexión intermitente para ser descargada y podrá ser utilizada en cualquier momento y lugar.

El objetivo de este proyecto es realizar una aplicación útil y sencilla de manejar, que permita a los usuarios que estén incursionando en el tema de los circuitos lógicos básicos encontrar actividades e información clara, que permita comprender y entender este tema.

7.2 Limitaciones

- No se puede ampliar o reducir la interfaz de usuario debido a que esto abarcaría un aumento en el número de archivos XML, correspondientes a la interfaz de usuario.
- El módulo de simulación:
 - Cuenta con un número determinado de compuertas lógicas para su simulación.
 - La cantidad de entradas de las compuertas lógicas es fijo.
 - No se permite realizar una conexión en realimentación, ya que la aplicación está dirigida a circuitos lógicos combinatorios y no secuenciales.
- Teniendo en cuenta que la versión de Android posibilita la adaptación de pantallas, se han realizado pruebas en dispositivos con las siguientes resoluciones: 320x480, 480x800 y 720x12080 píxeles.

7.3 Logotipo de la aplicación

Diseñado en función del tema principal y la institución a la que debo los conocimientos aprendidos, el logotipo de la aplicación se asigna en el archivo AndroidManifest.xml.



Fig. No 16. Logotipo de la aplicación sobre circuitos lógicos básicos.

7.4 Esquema general de la aplicación

El esquema general de la aplicación móvil es propuesto y desarrollado como se muestra en la figura No 17, pensando en el *diseño centrado en el usuario* para facilitar su navegación entre cada módulo y actividad con ello, se evita un mayor número de pasos o clics para llegar a cada uno de estos.

La información es presentada de forma corta y sencilla ya que los usuarios no leen extensas cantidades de texto en dispositivos móviles debido a sus condiciones de tamaño y visualización. Teniendo en cuenta el contexto y los objetivos del proyecto se comienzan a estructurar la manera en la que se presentarán los diferentes contenidos y los elementos de la interfaz. Las opciones principales del menú están conformadas por las acciones destacadas para los usuarios en la aplicación móvil. Como se ve en la imagen se plantean tres contenidos: uno es el simulador de compuertas lógicas, el otro es el módulo de fundamentos sobre los circuitos lógicos y finalmente el de datos generales de cada compuerta lógica.

En general en una aplicación móvil, las opciones del menú principal son presentadas de manera vertical, y no deberían exceder las cinco opciones para no esforzar al usuario y mejorar la velocidad de navegación [12].

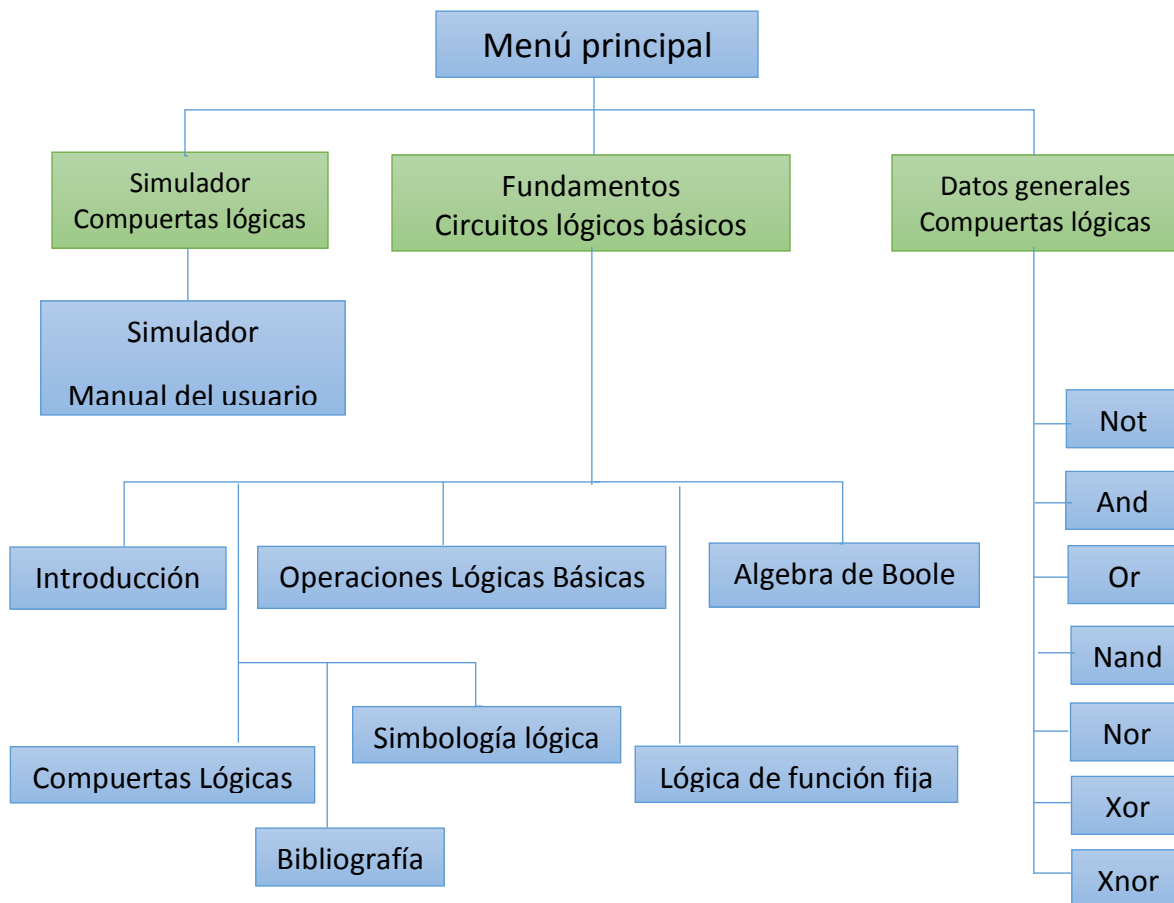
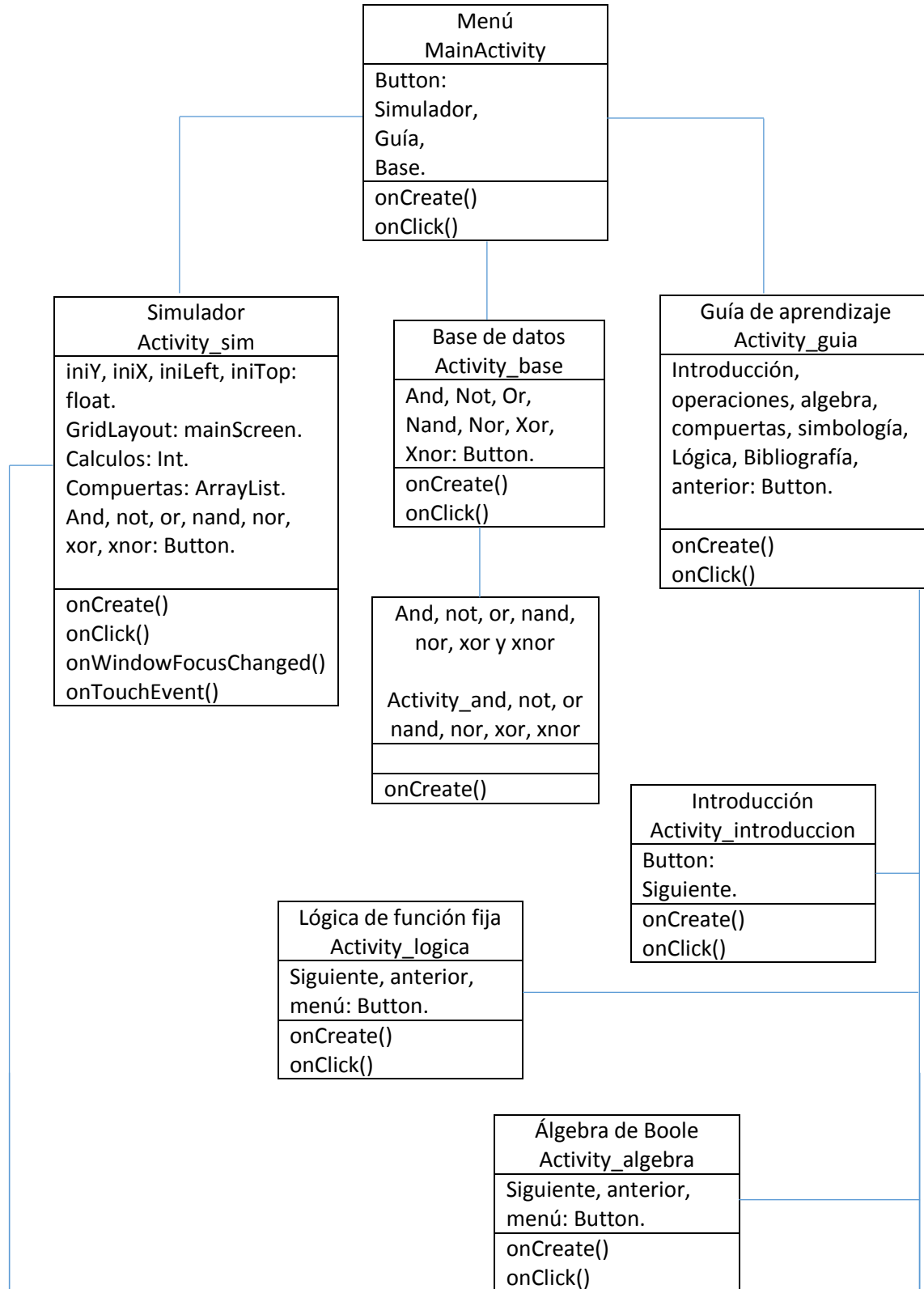
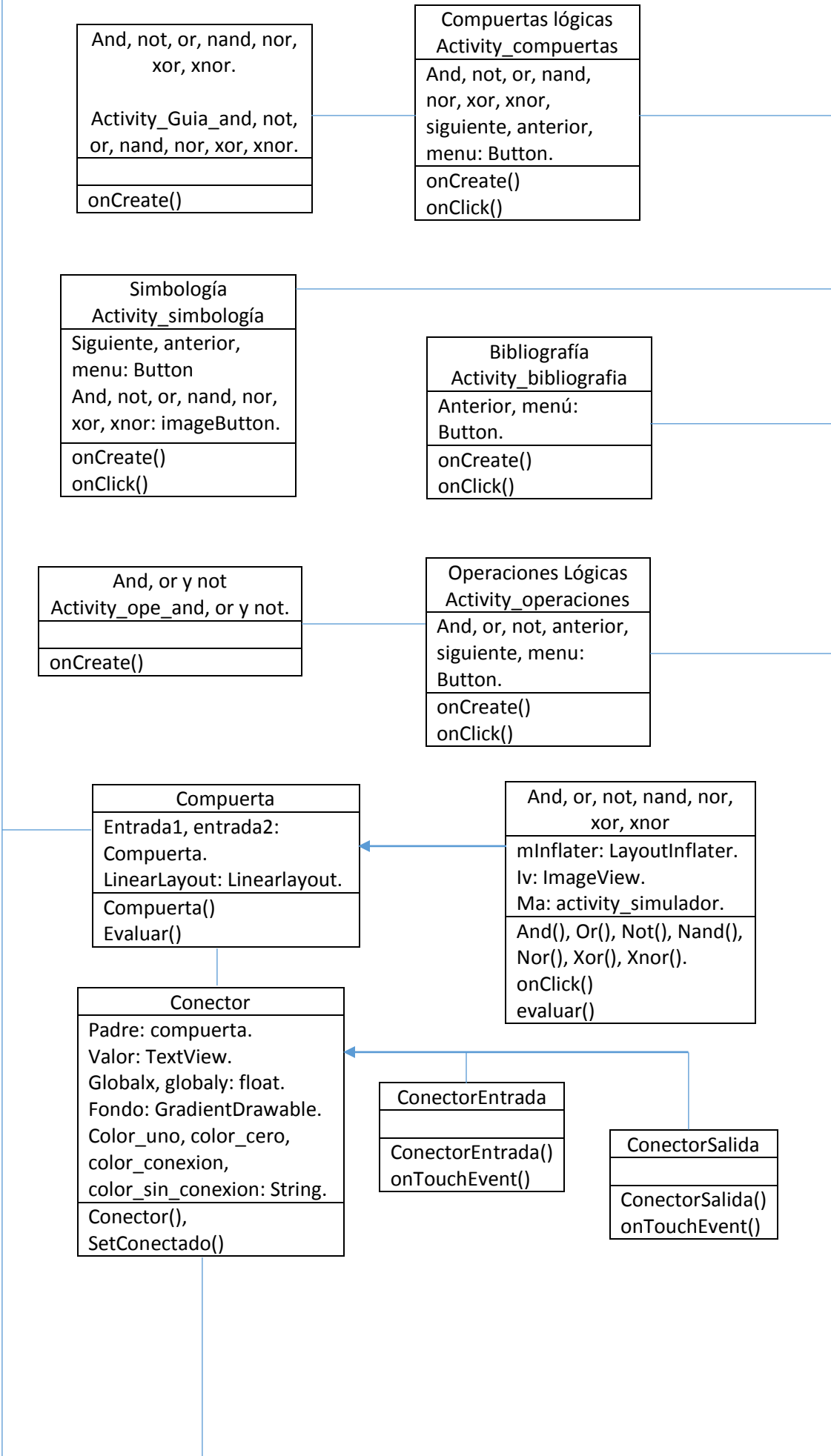


Fig. No 17. Esquema general de navegación de la aplicación planteada en este documento.

7.5 Diagrama de clases de la aplicación





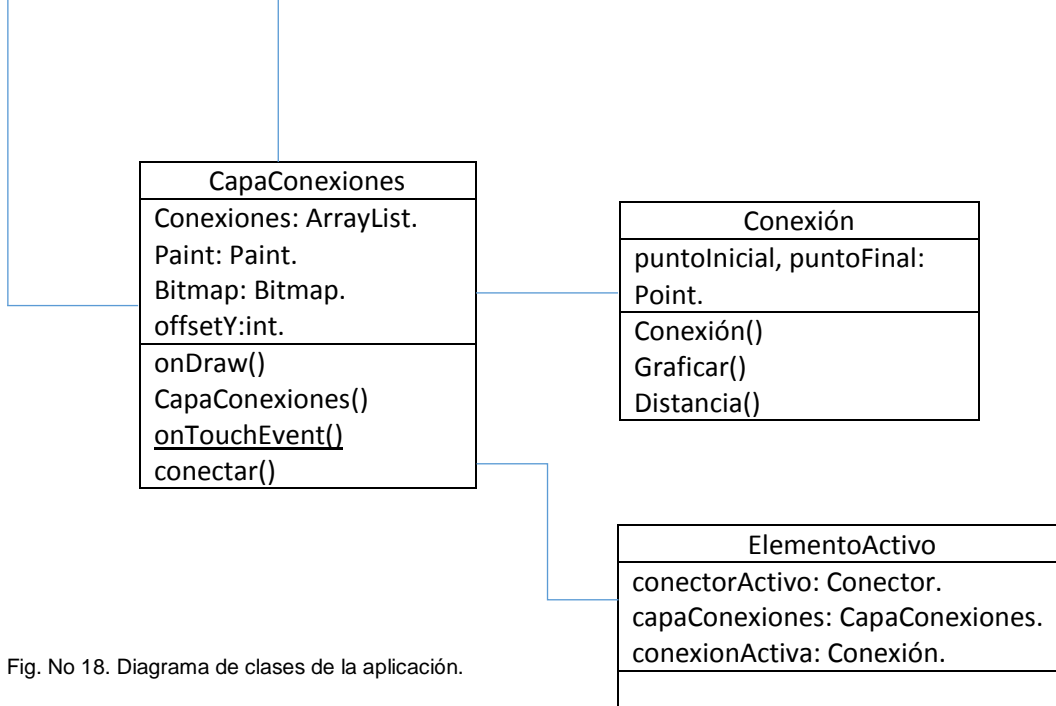


Fig. No 18. Diagrama de clases de la aplicación.

7.6 Tarjetas CRC

Una técnica para identificar los objetivos del modelo de dominio en función de los requisitos, es a través de las tarjetas CRC (Class Responsibility Collaborator) que identifican cada clase, su responsabilidad y colaboración con otra clase.

Clase	MainActivity
Responsabilidad	Es la clase principal de la aplicación, encargada de identificar los botones del menú principal de la aplicación que dirigirán al usuario a cada módulo.
Colaboradores	Activity_base, Activity_guia, Activity_sim

Clase	Activity_base
Responsabilidad	Permite identificar cada botón del menú de datos generales de compuertas lógicas y dirigir a los usuarios a los datos de la compuerta lógica deseada.
Colaboradores	And, not, or, nand, nor, xor y xnor.

Clase	Activity_and, not, or nand, nor, xor, xnor
Responsabilidad	Activa la pantalla donde se encuentran los datos generales de cada compuerta lógica.
Colaboradores	

Clase	Activity_fundamentos
Responsabilidad	Identifica y asigna el tema a cada botón en la pantalla del módulo sobre fundamentos de circuitos lógicos básicos para dirigir al usuario en el momento que lo seleccione.
Colaboradores	Activity_intro, Activity_log, Activity_alg, Activity_comp, Activity_simb.

Clases	Activity_introduccion, Activity_logica, Activity_algebra, Activity_compuertas, Activity_bibliografia, Activity_operaciones, Activity_ope_and, or y not.
Responsabilidad	Activa la pantalla que contiene la información de cada tema del módulo sobre fundamentos de circuitos lógicos básicos.
Colaboradores	

Clase	Activity_simbología
Responsabilidad	Permite activar la pantalla sobre simbología lógica y sacar un mensaje al presionar cualquiera de las imágenes que se encuentran en el tema sobre simbología lógica para recordar el nombre de la compuerta lógica pulsada.
Colaboradores	

Clase	Activity_simulador
Responsabilidad	Es la actividad principal del simulador. Activa la interfaz gráfica del simulador, identifica cada botón que se encuentra en la pantalla para asignarle su respectiva tarea, ajusta las imágenes a la pantalla, adecua la resolución de las conexiones entre compuertas y llama los diferentes métodos que se encuentran en el simulador.
Colaboradores	Compuerta, Conector, CapaConexiones.

Clase	Compuerta
Responsabilidad	Crea los objetos que utilizan algunas clases del simulador utilizados para la ubicación de compuertas e identificación de las entradas de las compuertas puestas en el área de simulación. Así como, el método para recibir el dato de las compuertas conectadas entre sí.
Colaboradores	Conector.

Clase	And, or, not, nand, nor, xor, xnor.
Responsabilidad	Coloca e identifica la imagen de la compuerta que se desea poner en el área de simulación, además, activa la compuerta seleccionada por el usuario en el área de simulación para ser cambiada o borrada, realiza la operación de la compuerta y recibe el dato de salida de la compuerta de la cual está conectada.
Colaboradores	Compuerta.

Clase	Conector
Responsabilidad	Pone el color del número de entrada de la compuerta e identifica cuando una entrada está o no conectada para poner su color.
Colaboradores	CapaConexiones, compuerta.

Clase	ConectorEntrada
Responsabilidad	Cambia los valores de entrada de la compuerta que se desea simular de 0 a 1 y de 1 a 0. Identifica la posición x y de las entradas y activa la entrada cuando es seleccionada para borrar la conexión, en caso de que este conectada.
Colaboradores	Conector.

Clase	ConectorSalida
Responsabilidad	Activa la salida cuando es seleccionada para realizar una conexión con otra compuerta.
Colaboradores	Conector.

Clase	CapaConexiones
Responsabilidad	Realiza y dibuja la conexión desde el punto inicial de salida de una compuerta hasta el punto final de entrada de otra compuerta. Contiene el método para seleccionar la entrada de una compuerta conectada y borrar la conexión.
Colaboradores	Conector, Conexión, ElementoActivo.

Clase	Conexión
Responsabilidad	Determina el punto inicial y final de conexión entre compuertas.
Colaboradores	CapaConexiones.

Clase	ElementoActivo
Responsabilidad	Contiene objetos que son utilizados entre las clases conector, Capa conexiones y Conexión.
Colaboradores	CapaConexiones.

7.7 Diseño de interfaz gráfica

El diseño de la interfaz de usuario fue esbozado en correspondencia con los requisitos funcionales que debe cumplir el sistema y las herramientas con las que se cuenta en el programa eclipse Android para el diseño de aplicaciones móviles.

7.7.1.1 Menú principal

Tal como se mencionó en los requisitos funcionales; la pantalla Inicial de la aplicación cuenta con el menú principal, donde se encuentran cada uno de los módulos allí descritos. Los colores, imágenes y demás elementos que se observan en las siguientes pantallas han sido seleccionados para entregar una interfaz amigable y agradable para el usuario final. En la siguiente imagen se observa el menú principal con los botones que dirigen al usuario a cada uno de los módulos realizados en la aplicación:

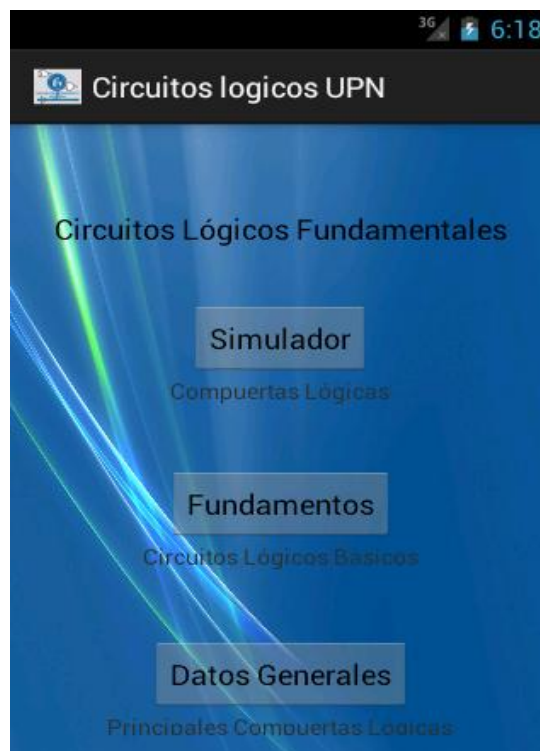


Fig. No 19. Menú principal de la aplicación sobre Circuitos Lógicos Básicos.

7.7.1.2 Módulo del Simulador

En el momento en que el usuario pulsa el botón del simulador en el menú principal, lo dirige a la pantalla donde se encuentra el botón de ingreso al simulador y el manual de uso como se observa en la figura No 18. Allí el usuario puede encontrar las indicaciones para utilizar el simulador de compuertas lógicas.

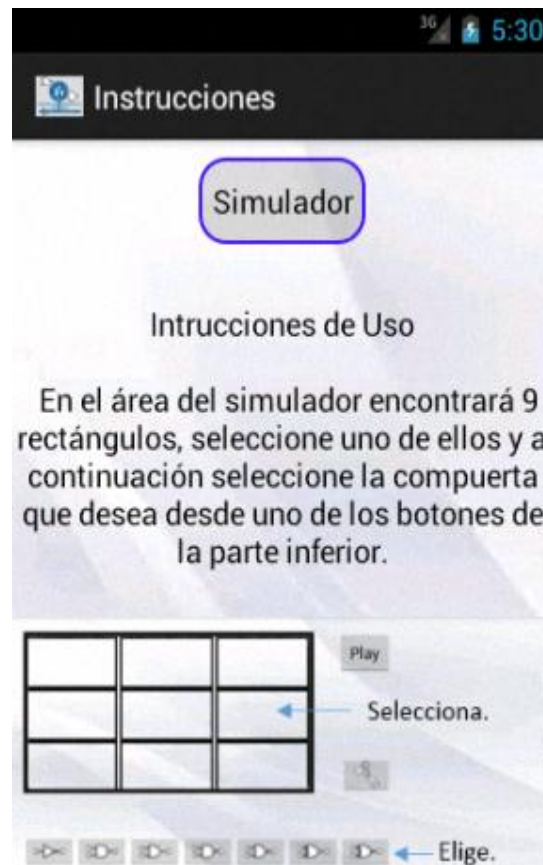


Fig. No 20. Interfaz gráfica del manual de uso del simulador y el botón que dirige al usuario al simulador.

Acto seguido al pulsar el botón del simulador, dirige al usuario a la interfaz que se observa en la figura No 19. Esta interfaz fue diseñada para ofrecer al usuario un área de simulación compuesta por nueve espacios en los que se puede poner cualquiera de las 7 compuertas lógicas que se encuentran en la parte inferior de esta pantalla. También se encuentra un botón con la imagen de una tijera, para borrar las compuertas que se localicen en el área de simulación, al igual que las conexiones realizadas entre compuertas y un botón de *Play* para activar la operación de las compuertas que se encuentren en el área de simulación.



Fig. No 21. Interfaz gráfica del simulador de compuertas lógicas.

7.7.1.3 Módulo sobre fundamentos de circuitos lógicos básicos

En el momento en que el usuario pulsa el botón sobre fundamentos de circuitos lógicos básicos en el menú principal de la aplicación, es dirigido al contenido teórico sobre estos sistemas. El usuario encontrará el índice que conforma este contenido, el cual puede ser desplazado de abajo hacia arriba o viceversa para observar cada punto, he ingresar a uno de estos de forma aleatoria o comenzar desde el inicial que es la *introducción*, pasar por cada uno de ellos hasta llegar a la *bibliografía* para tener una secuencia en el contenido. Diseñado de esta manera para obtener una vista de forma horizontal y vertical en el dispositivo móvil.

Las pantallas que conforman este módulo contienen imágenes y texto que acercan al usuario hacia estos sistemas. La figura No 20 muestra el índice o contenido teórico del módulo sobre fundamentos de circuitos lógicos básicos.

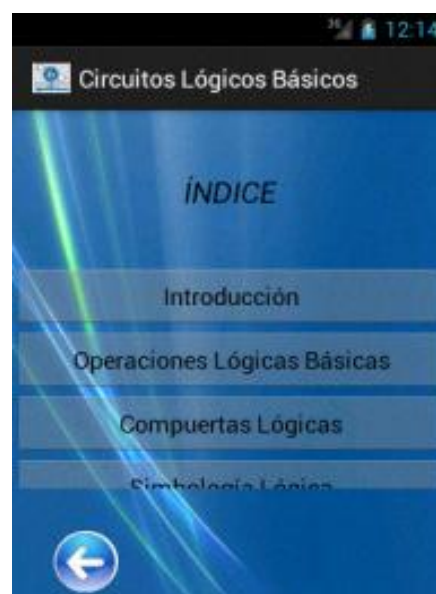


Fig. No 22. Interfaz gráfica del módulo sobre fundamentos de circuitos lógicos básicos.

En el punto sobre *Simbología lógica* que se encuentra en el índice del módulo sobre fundamentos de circuitos lógicos básicos se realizó una actividad que permite a los usuarios recordar el nombre de cada compuerta lógica, pulsando sobre la imagen que contiene el símbolo estándar de la compuerta. El diseño con el que cuenta esta interfaz es el generalmente utilizado para los otros contenidos de este módulo donde encuentran el texto y las imágenes centradas para una mejor visualización. Esta actividad se puede observar en la siguiente imagen, en la cual es pulsada la imagen de la compuerta NOT.

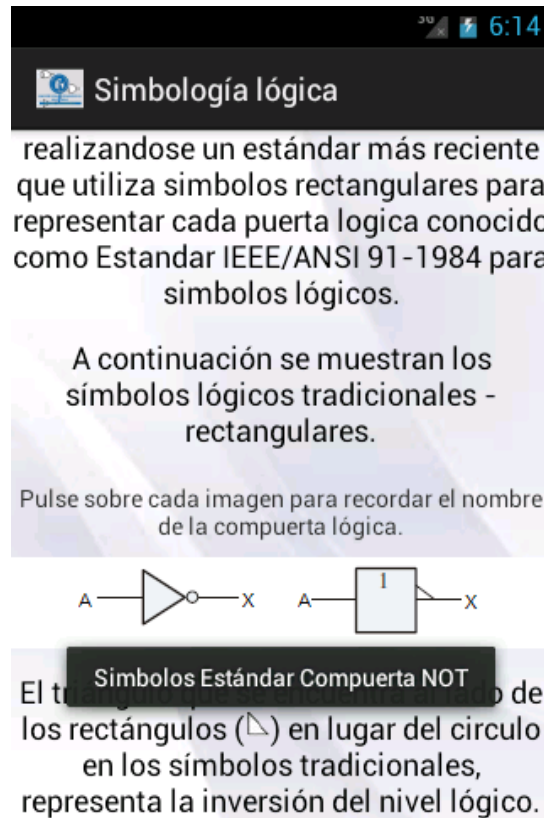


Fig. No 23. Mensaje por pantalla al pulsar una de las imágenes en el punto simbología lógica.

7.7.1.4 Módulo de datos generales de las principales compuertas lógicas

Al pulsar el botón de datos generales de las principales compuertas lógicas que se encuentra en el menú principal de la aplicación móvil, el usuario es dirigido a la pantalla que se muestra a continuación:

En general, la aplicación mantiene el mismo diseño en cada una de sus pantallas para que el usuario maneje de forma fácil y casi idéntica cada uno de los módulos; en este caso se mantiene la misma galería de botones que se presentó en el módulo de fundamentos de circuitos lógicos básicos, al pulsar estos botones el usuario se dirige a los datos generales de la compuerta lógica seleccionada.

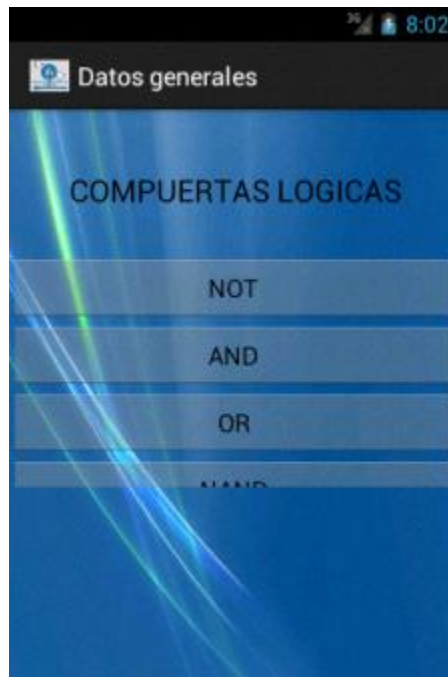


Fig. No 24. Módulo de datos generales de las principales compuertas lógicas.

En caso en que el usuario seleccione la compuerta NOT encontrará la interfaz gráfica vista en la figura No 23. Allí el usuario puede observar el símbolo distintivo tradicional, la tabla de verdad, el esquema, la serie y el chip de la compuerta que escogió; desplazando estos datos de abajo hacia arriba o viceversa.



Fig. No 25. Vista de una de las pantallas al seleccionar una compuerta lógica en el módulo de datos generales.

Sin importar la compuerta que el usuario seleccione el formato de estas pantallas será el mismo, es decir, que tendrán los mismos datos y se encontrarán en el mismo orden.

8 Construcción y arquitectura

8.1 Herramientas de desarrollo

Existen múltiples herramientas para la creación de aplicaciones móviles con sistema operativo Android; la sugerencia propuesta desde la página oficial de desarrolladores de Android es realizarlas en el programa Eclipse, programa que se seleccionó en este proyecto por su lenguaje de programación *Java*, libre y conocido por el programador.

Eclipse es un editor de código que permite compilar en tiempo real, a través de un complemento llamado Android Development Tool (ADT), que las funcionalidades requeridas en Android.

Desde la página oficial *Android Developers* (Desarrolladores de Android) se puede descargar la última versión de Eclipse junto con un conjunto de herramientas de desarrollo de software llamado Android SDK (Android Software Development kit) que incorpora lo necesario para realizar las aplicaciones y probarlas inmediatamente en una máquina virtual.

Una vez instalado Eclipse, el Android SDK y el complemento ADT se pueden actualizar a través de Android SDK Manager en Eclipse, que permite separar las herramientas, plataformas y otros componentes en paquetes que se pueden descargar mediante este administrador; permitiendo al usuario seleccionar los paquetes que desea descargar e instalar.

Otra herramienta necesaria es el software JDK (Java Development Kit), necesaria para el funcionamiento de Eclipse, que es utilizada para compilar y ejecutar las aplicaciones en el lenguaje de programación *Java*, el cual contiene programas y librerías, que cuenta con diferentes versiones para casi todos los sistemas operativos y versiones comerciales.

8.2 Estructura de la aplicación

En el momento de crear un proyecto en Eclipse Android se crea una estructura de componentes por defecto, estas carpetas y archivos, conforman la arquitectura del proyecto que integra la aplicación. En la parte superior izquierda de la Figura No 26 se pueden observar. En esta misma imagen se muestra el entorno de trabajo de Eclipse junto a la máquina virtual que permite emular el comportamiento de las aplicaciones en diferentes dispositivos móviles.

La carpeta **src** contiene diferentes subclases de actividades, además de la actividad principal “Activity” que representa la pantalla inicial; estos archivos manejan la misma estructura encontrada en Java donde se aloja el código fuente de la aplicación.

En el caso de la carpeta **res**, esta maneja diferentes recursos de la aplicación como imágenes, vídeos, cadenas de texto, archivos XML que definen el diseño de las diferentes interfaces gráficas, entre otros recursos.

Cualquier proyecto Android desarrollado en Eclipse al ser compilado creará un archivo .apk (Application Package file), que es un cambio de los archivos clásicos .jar de Java.

Este archivo .apk es el empaquetado para distribuir la aplicación en la tienda de **Google Play** o cualquier otro servidor y ser descargado e instalado en el dispositivo móvil del usuario.

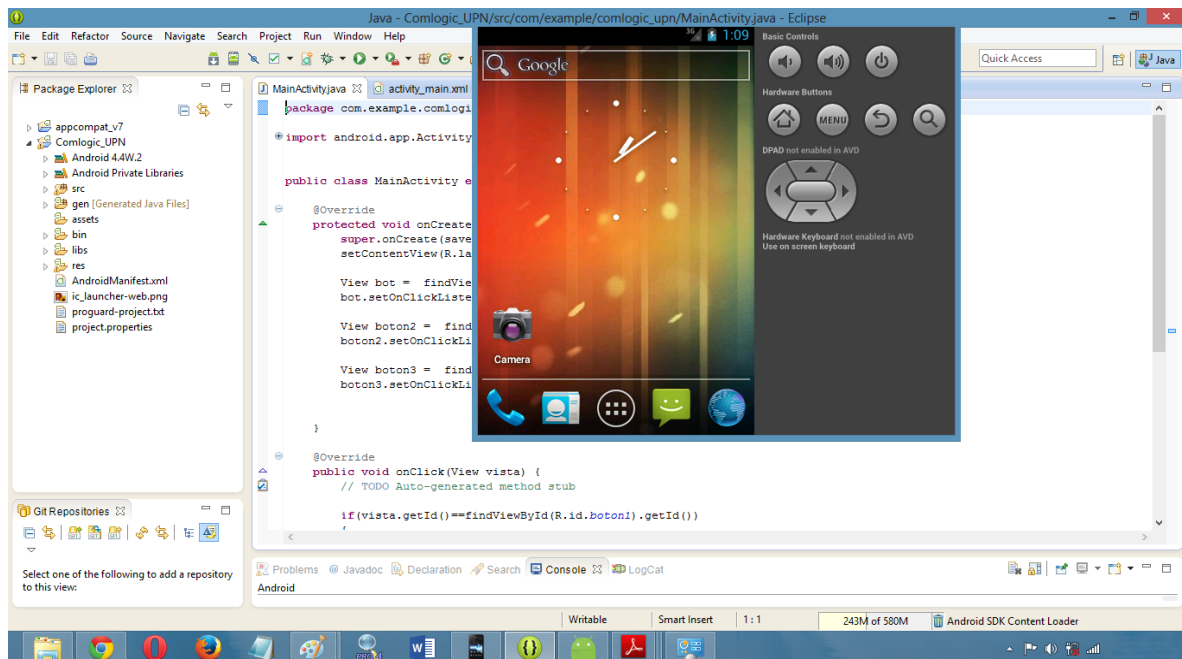


Fig. No 26. Entorno de trabajo de Eclipse Android, junto a la máquina virtual para emular los diferentes dispositivos móviles.

8.2.1 El Archivo AndroidManifest.xml

Este archivo contiene la configuración de la aplicación, en él se declaran aspectos como el nombre de la aplicación, la versión mínima de Android para la cual está disponible, la imagen que representa el icono, las actividades creadas, permisos y demás elementos que necesita la aplicación para su correcto funcionamiento.

El archivo `AndroidManifest.xml` es uno de los archivos más importantes de cualquier aplicación, sin él la aplicación no puede funcionar. La estructura completa del archivo de la aplicación planteada en este documento se muestra en el anexo No 1.

Donde se puede observar el nombre del paquete de la aplicación llamado `com.example.comlogic_upn` con el cual se referencia la aplicación en Google Play y en el dispositivo móvil, la versión mínima para su funcionamiento es la API 15 o 4.0.3 Ice Cream Sandwich y las actividades creadas para generar cada uno de las pantallas de usuario.

8.2.2 Actividades

Las actividades representan el componente principal de la interfaz gráfica de una aplicación Android, se puede decir que cada pantalla es una actividad. Las actividades están conformadas por dos partes, la parte lógica y la parte gráfica [12]. Las aplicaciones se componen de una o varias actividades, estas son subclases de la actividad principal "Activity", que trabajan de forma individual o unidas durante el tiempo de ejecución de la aplicación.

8.2.3 Vista

Cada vista o pantalla que observa el usuario está construida desde un archivo XML y su respectiva actividad. El archivo XML está alojado en la carpeta `res/Layout` de la estructura de la aplicación creada en el Software de desarrollo Eclipse y contiene todos los elementos visuales que observa el usuario como botones, listas, imágenes, texto, entre otros.

8.3 Menú principal

La pantalla principal o menú principal de la aplicación sobre circuitos lógicos básicos se muestra en la figura No 19. Donde se puede observar el nombre de la aplicación, un título principal y tres botones que dirigen al usuario a cada uno de los módulos.

El archivo XML que realiza una interfaz como esta se encuentra en el anexo No 2. La realización de este tipo de interfaz gráfica se efectúa mediante los siguientes elementos dentro del archivo XML:

Layout: Se utiliza para definir la estructura visual de las pantallas, es necesario que en cada interfaz gráfica de una aplicación se encuentre uno o varios layouts para la ubicación de los elementos que la componen; dentro del cual se encuentran imágenes, texto, botones, entre otros.

Un ejemplo de la creación de un Layout se muestra en el anexo No 3, con los elementos de configuración como Android width y height que determinan sus dimensiones.

Eclipse cuenta con varios Layout que manejan diferentes características, como el caso del *RelativeLayout* que permite posicionar los elementos en cualquier lugar de la pantalla.

Botones básicos: Eclipse permite utilizar este tipo de botones de forma rápida para la creación de aplicaciones, uno de estos botones se puede observar en la figura No 27. Este botón en su programación contiene el identificador del botón Android:@+id/ para poder asociarlo con una tarea en particular en el momento de pulsarlo, el procesador de texto con la propiedad Android:text y las propiedades width y height para el tamaño del botón, anexo No 4.

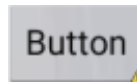


Fig. No 27. Botón básico.

Al igual que el botón, el texto es una de las herramientas que pueden ser agregados a las aplicaciones de forma sencilla en eclipse. Hay que especificar que cada uno de los elementos encontrados en eclipse Android que se pueden agregar a la interfaz de usuario tiene múltiples características que son especificadas en la página oficial de programadores de Android anexo No 5.

Circuitos lógicos fundamentales.

8.3.1 Actividad principal

En la actividad principal se define las tareas que realizará cada botón en el menú inicial; para identificar cada botón se utiliza un identificador determinado y con ayuda del evento *onClick* se asigna una tarea específica a cada uno de ellos, como se puede ver en el anexo No 6, Se muestra el código realizado para este fin y la estructura completa de la clase.

8.4 Módulo de datos generales

En la figura No 28 se observa el entorno de trabajo, el módulo de datos generales y el índice de este. En el cual se utilizaron elementos como: texto, para indica el lugar donde el usuario se encuentra, botones con su respectivo texto, que dirigen a los datos generales de la compuerta lógica escogida; un *ScrollView*, que es un tipo de layout que tiene la función de disponer de una serie de elementos visuales

que superan la cantidad de espacio del visor del dispositivo móvil; permitiendo al usuario desplazarlos de abajo hacia arriba utilizando el tacto o dedo en la interfaz y, finalmente un LinearLayout un tipo de layout que permite colocar cada uno de los elementos de forma ordenada dentro del ScrollView para que se mantenga un orden respectivo y acorde con la aplicación.

En la parte superior izquierda de esta misma imagen, se puede observar la estructura de un ScrollView, un LinearLayout y un elemento cualquiera que en este caso, es un botón. Las compuertas que en total son 7 (Not, And, Or, Nand, Nor, Xor y Xnor), a las que puede acceder el usuario son un conjunto de archivos XML y actividades creadas que permiten tener la interfaz de usuario vista en la figura No 25.

En el anexo No 7, se observa el código para la realización de los layout utilizados en esta interfaz y la estructura completa del archivo XML donde se tiene la interfaz del índice sobre datos generales sobre cada compuerta lógica.

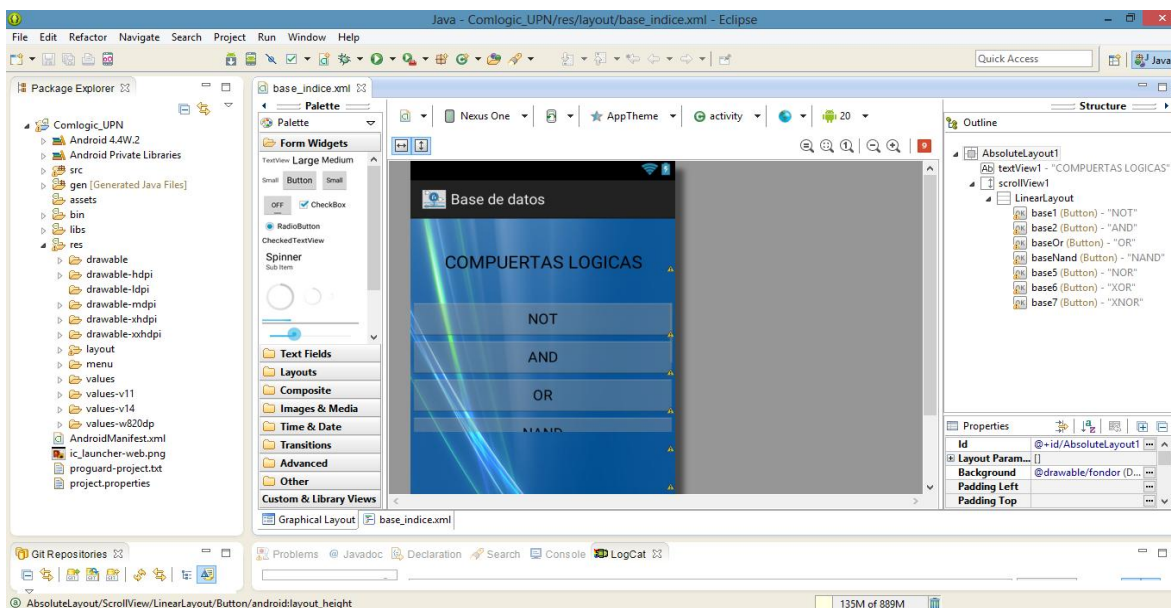


Fig. No 28. Diseño del índice de la base de datos.

8.5 Módulo sobre fundamentos teóricos de circuitos lógicos básicos

El índice de información sobre circuitos lógicos básicos de la aplicación móvil es casi similar al índice del módulo de información sobre compuertas lógicas de este mismo software. El elemento particular que tiene esta interfaz es un botón personalizado que permite acceder al menú principal de la aplicación móvil. La imagen que muestra el botón personalizado se muestra a continuación y el código para su realización lo puede observar en el anexo No 8.



Fig. No 29. Botón imagen o Image Button.

Después de crear el botón en el archivo XML, en la carpeta drawable lugar donde se alojan la mayoría de imágenes puestas en la aplicación, se crea un archivo XML haciendo clic derecho sobre esa carpeta y seleccionando *Android XML file*; se coloca el nombre y el tipo de archivo “*shape*” que se encargara de dar el diseño a los botones. La estructura de estos archivos XML se encuentra en el mismo anexo del botón personalizado.

Se debe tener en cuenta que para los botones existen dos estados, uno sin presionar y el otro presionado, es decir, que con estos elementos el usuario tendrá la sensación de haber pulsado o no un objeto. Para poder alterar el estado del botón se debe crear otro archivo XML dentro de la carpeta drawable, pero en este caso será del tipo “*selector*” que permitirá seleccionar cada uno de los estados según sea el caso.

En la siguiente figura se puede observar los dos estados del botón, normal y presionado con el diseño final.

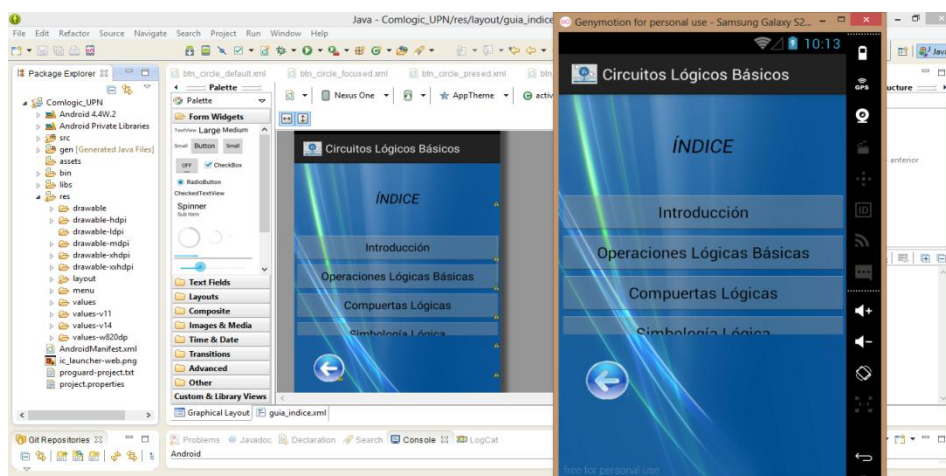


Fig. No 30. Botón con diseño personalizado normal y presionado en la consola de trabajo y en la máquina virtual.

En el punto sobre compuertas lógicas se realizó una galería de botones en forma horizontal con el fin de que el usuario pueda seleccionar la compuerta lógica deseada con el fin de ampliar la información de esta, desplazando los botones de derecha a izquierda. El código de la galería es presentado en el anexo No 9.

La siguiente vista, muestra la galería de botones en el entorno de trabajo y en la máquina virtual que permite al usuario seleccionar la compuerta deseada.

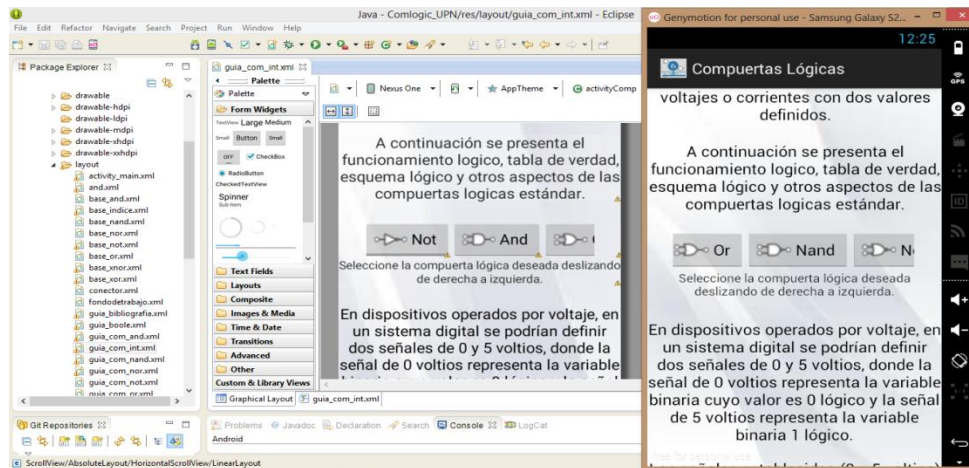


Fig. No 31. Presentación de la galería de botones para seleccionar la información de cada compuerta lógica.

Como se observó en la figura No 23, en el punto sobre simbología lógica, el usuario puede observar el símbolo lógico tradicional y rectangular de cada compuerta lógica con la posibilidad de pulsar sobre cada imagen para recordar el nombre de cada símbolo estándar.

El código que permite realizar esta actividad está dividido en dos archivos, uno de ellos, es el realizado en el archivo XML y el otro en la actividad creada para esta interfaz.

En el archivo XML se coloca el elemento *imagen botón* (ImageButton) para cada compuerta lógica, el identificador @id para enlazarlo con la actividad, un color de fondo blanco para contrastar con la imagen de la compuerta puesto con la propiedad android:background y el color blanco en código Java #FFFFFF y la elección de la imagen en el ImageButton en la propiedad Android:src y la dirección donde se encuentra la imagen, el código para la realización de los botones en el archivo XML se puede observar en el anexo No 10.

En la actividad relacionada con el archivo XML sobre simbología lógica se encuentra el código para realizar la tarea de mostrar mensajes por pantalla; al pulsar una imagen, lo primero que se debe hacer es crear un objeto relacionado con el ImageButton y asignarlo al evento onClick.

En el evento onClick se crea un objeto String para asignar cada uno de los mensajes con la sentencia de control Switch case.

Y finalmente la salida por pantalla de la selección del mensaje o la notificación al usuario por unos segundos utilizando la clase Toast. Esta clase dispone de un método *makeText()* al que se debe pasar como parámetro el texto que se debe mostrar y la duración del mensaje, que puede tomar los valores de LENGTH_LONG o LENGTH_SHORT. Al tener listo estos elementos solo queda mostrarlo en pantalla mediante el método *show()*.

La clase completa que contiene el código para realizar los mensajes por pantalla sobre el tema simbología lógica se puede observar en el anexo No 11, de este documento.

8.6 Módulo del simulador de compuertas lógicas básicas

Finalmente, el entorno donde se realiza la simulación de las compuertas lógicas básicas es presentado en la figura No 32, en esta imagen se puede observar el entorno de trabajo y la máquina virtual en el módulo “*simulador*”. El archivo XML que tiene la interfaz del simulador contiene los siguientes elementos:

Capa de conexiones

La capa conexión es una estructura realizada en código no visible para el usuario para dibujar las conexiones que se realizan entre compuertas identificando los puntos x y del punto inicial que se toma desde la salida de la compuerta y el punto final que es la entrada de otra compuerta, esta actividad es configurada desde el método `onTouchEvent(MotionEvent event)`. También interviene en el ajuste del dato de entrada en una compuerta lógica cuando la conexión es quitada, es decir, en el caso que una compuerta esté conectada y se desconecte, el parámetro de entrada de la compuerta volverá a cero (0) debido al código alojado en el método `borrar()`. Estos métodos se encuentran en la clase “*CapaConexiones*” que se puede observar en el anexo No 12. Esta clase, trabaja con los archivos XML y clases que manejan la imagen e identificación de cada compuerta puesta en el área de simulación, que además contiene los elementos de conexión de entrada y salida de la compuerta.

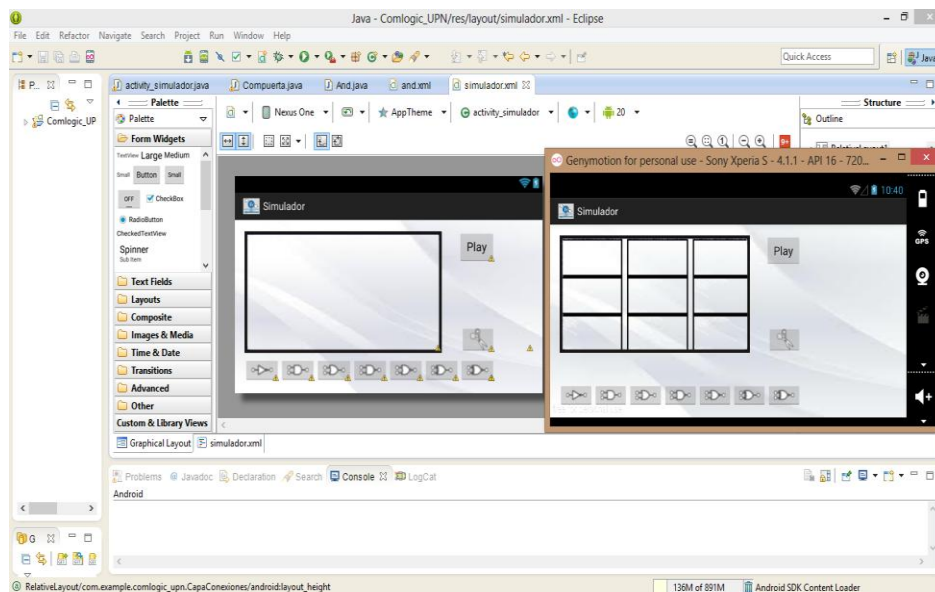
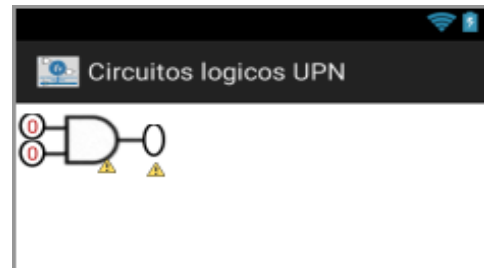


Fig. No 32. Entorno de trabajo en la interfaz simulador junto a la máquina virtual en el módulo simulador.

A continuación se muestra la imagen de los archivos XML donde se puede observar en (a), el conector de entrada que permite cambiar de 0 a 1 o de 1 a 0 y recibir conexiones de elementos de salida de las compuertas y en el (b), una de las compuertas lógicas con dos elementos de entrada y un conector de salida. Los archivos XML para realizar esta interfaz se pueden observar en los Anexos 13 y 14.



(a) Elemento de entrada.



(b) Compuerta lógica con los conectores De entrada y salida.

Fig. No 33. Archivos XML que conforman las compuertas lógicas y tienen los elementos de conexión.

GridLayout

Es un layout que permite ubicar elementos de la misma manera que una tabla o matriz, en este lugar se colocan las compuertas escogidas por el usuario para su simulación. El layout se crea con un número de filas y columnas determinadas, que en este caso es de tres por tres como se observa en la figura No 32. La identificación, activación y manejo se realiza desde la clase principal del simulador “*activity_simulador*” vista en el anexo No 16, de esta misma manera los botones que integran esta interfaz son manejados e identificados desde allí.

Botones

En la parte inferior del módulo *simulador* se encuentran siete botones con la imagen de cada compuerta lógica. En el momento de seleccionar uno de los nueve recuadros en el área de simulación, en la actividad del simulador se activa el método `onClick(View v)` esperando la próxima acción del usuario, después de esto al pulsar sobre uno de los siete botones se genera el cambio de *LinearLayout* desde este método. Este layout contiene la imagen de la compuerta deseada, junto a los elementos de conexión de entrada y salida; de esta misma manera se hace el cambio de compuerta dando clic sobre una de las compuertas puestas en el área de simulación y pulsando el botón de la compuerta que se desea cambiar. La ubicación de las compuertas es posible por la creación de un *ArrayList* que posiciona cada *LinearLayout* en el *GridLayout* desde la clase principal del simulador.

El botón de las tijeras observado en esta misma interfaz cuenta con dos funciones, una de ellas es quitar las compuertas puestas en el área de simulación. Esta actividad se realiza pulsando en el centro de la imagen o compuerta para ser activada y presionar este botón para el respectivo borrado. Internamente se realiza un cambio de *LinearLayout* que contiene una imagen que da la apariencia de borrado para el usuario. La segunda función consiste en quitar las conexiones realizadas entre compuertas, dando clic sobre la entrada de la compuerta conectada y posteriormente en este botón. Las clases que intervienen en la actividad de borrado de conexión son "*CapaConexiones*", "*Conexión*", "*ElementoActivo*", "*ConectorEntrada*" y la clase principal del simulador.

Por último, el botón de *Play* que se encuentra en la parte superior de este módulo, permite obtener los resultados de salida de las compuertas. Cada imagen o compuerta puesta en el simulador cuenta con una actividad que maneja la operación lógica. La clase y operación de la compuerta *And* se puede observar en el anexo No 17.

La actividad "*Compuerta*" contiene: los objetos creados para identificar cada entrada de las compuertas lógicas y el objeto realizado para llamar la actividad que representa cada compuerta.

Desde el evento *OnClick(View v)* de la actividad principal del simulador se maneja un mensaje por pantalla que se activa, si el usuario realiza una conexión en realimentación, informando que ha cometido un error ya que es un simulador de circuitos combinatorios.

La actividad "*Conector*" determina los colores de los datos de entrada (0 y 1) de cada compuerta y permite darle color a la conexión cuando es realizada y ha sido seleccionada para borrarla.

La actividad "*ConectorEntrada*" maneja el cambio de 0 a 1, 1 a 0 y la activación de la entrada cuando se desea borrar una conexión. La actividad "*ConectorSalida*" identifica cuando ha sido activada la salida de una compuerta para la conexión con otras, y muestra el resultado de la operación en la salida de cada compuerta lógica cuando se presiona el botón de *Play* en el módulo *simulador*.

Capítulo IV

Pruebas, Evaluación y Entrega

Al finalizar la aplicación se comprueba su funcionamiento, se realizan las pruebas y se sube a un servidor. Las pruebas son realizadas de dos formas, una a través del emulador de dispositivos móviles y la otra desde un dispositivo móvil real.

La evaluación es realizada desde un cuestionario a usuarios principiantes y experimentados.

El objetivo primordial de este capítulo es comprobar el funcionamiento óptimo de la aplicación móvil.

9 Pruebas

9.1 Emulador de dispositivos móviles

Teniendo en cuenta que el simulador de compuertas lógicas está diseñado para la elaboración y comprobación de circuitos lógicos básicos, se toma un circuito realizado en el libro *Fundamentos de sistemas digitales* del capítulo No 4. Sobre álgebra de Boole y simplificación lógica, para realizar la prueba. La siguiente figura muestra el circuito lógico y la tabla de verdad de las diferentes posibilidades de combinación de este circuito.

A	B	C	A+B	A+C	(A+B)(A+C)	BC	A+BC
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

$$(A + B)(A + C) = A + BC.$$

Fig. No 34. Circuito tomado para la prueba del simulador de compuertas lógicas de la aplicación en el emulador de Eclipse [6].

Entre las muchas posibilidades que proporciona Eclipse Android para el desarrollo de aplicaciones móviles, existe una que permite fusionar el emulador de dispositivos móviles de *Genymotion* a través de un complemento diseñado para tal fin.

Instalado este emulador se puede disponer de múltiples dispositivos móviles. En este caso el dispositivo seleccionado para realizar la prueba es el Sony Xperia S con sistema operativo Android versión 4.1.1 – API 16 y tamaño de pantalla 720x1280. Estando en el módulo del simulador se disponen las compuertas necesarias para realizar el circuito de la figura No 34, y se da clic en el botón Play para que el simulador muestre el resultado del circuito elaborado. La primera prueba (#1) muestra la salida del circuito cuando:

$$A = 0, \quad B = 1 \quad y \quad C = 0$$

La siguiente figura muestra el circuito planteado y el resultado de los anteriores valores.

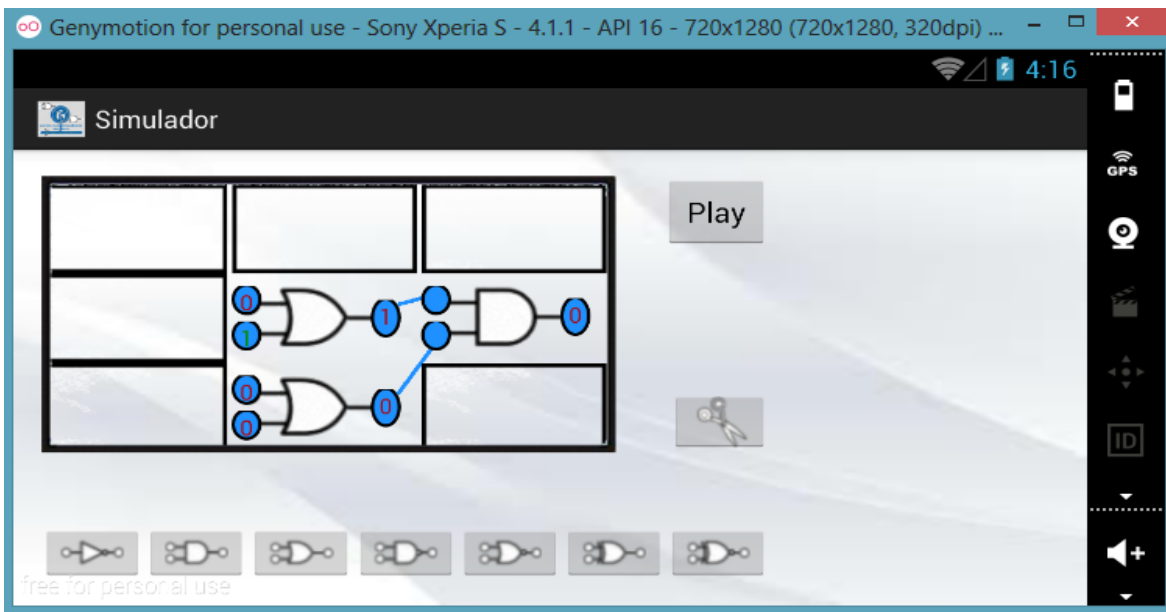


Fig. No 35. Simulación y resultado del circuito planteado en la prueba #1 en el emulador de dispositivos móviles.

Para la segunda prueba (#2) muestra la salida del circuito cuando:

$$A = 1, \quad B = 0 \quad \text{y} \quad C = 1$$

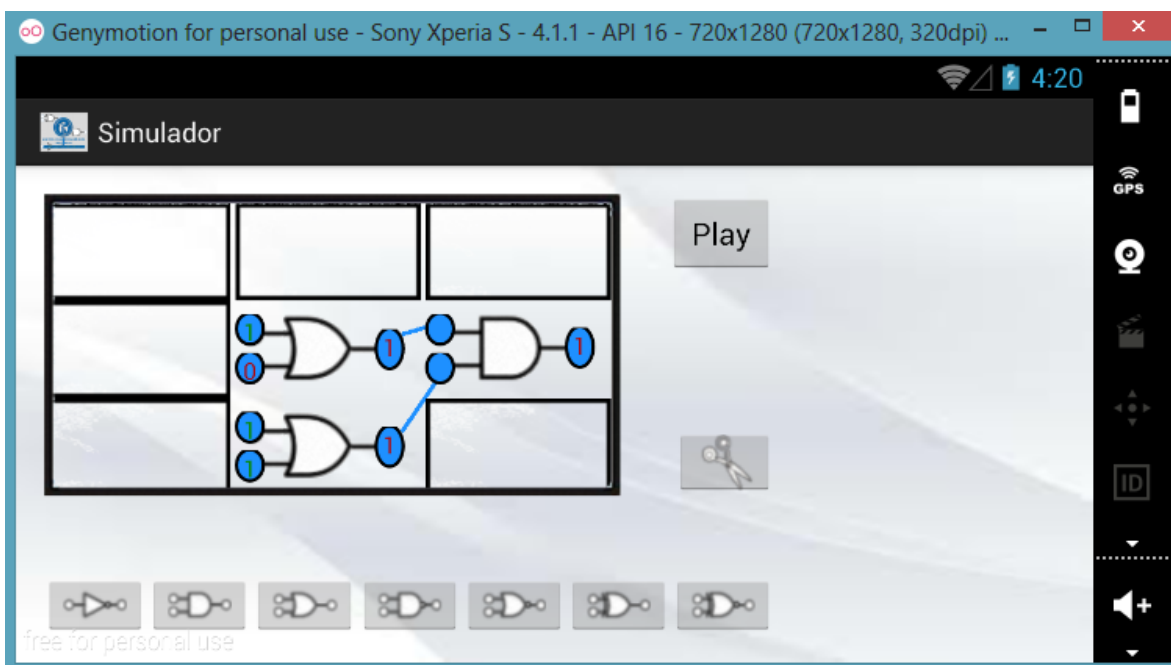


Fig. No 36. Simulación y resultado del circuito planteado en la prueba #2 en el emulador de dispositivos móviles.

Para finalizar la prueba del simulador de la aplicación móvil en el emulador, se selecciona el siguiente circuito lógico:

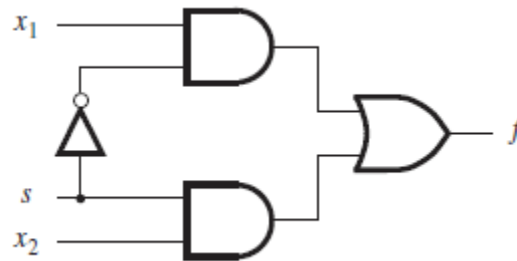


Fig. No 37. Circuito elegido para realizar la prueba final del simulador de compuertas lógicas en el emulador [13].

Cuando: $x_1 = 0$, $s = 1$ y $x_2 = 1$

El resultado de la anterior operación es: $f = 1$

La elaboración y resultado del circuito de la figura No 37, en el simulador de la aplicación realizada se muestra a continuación:

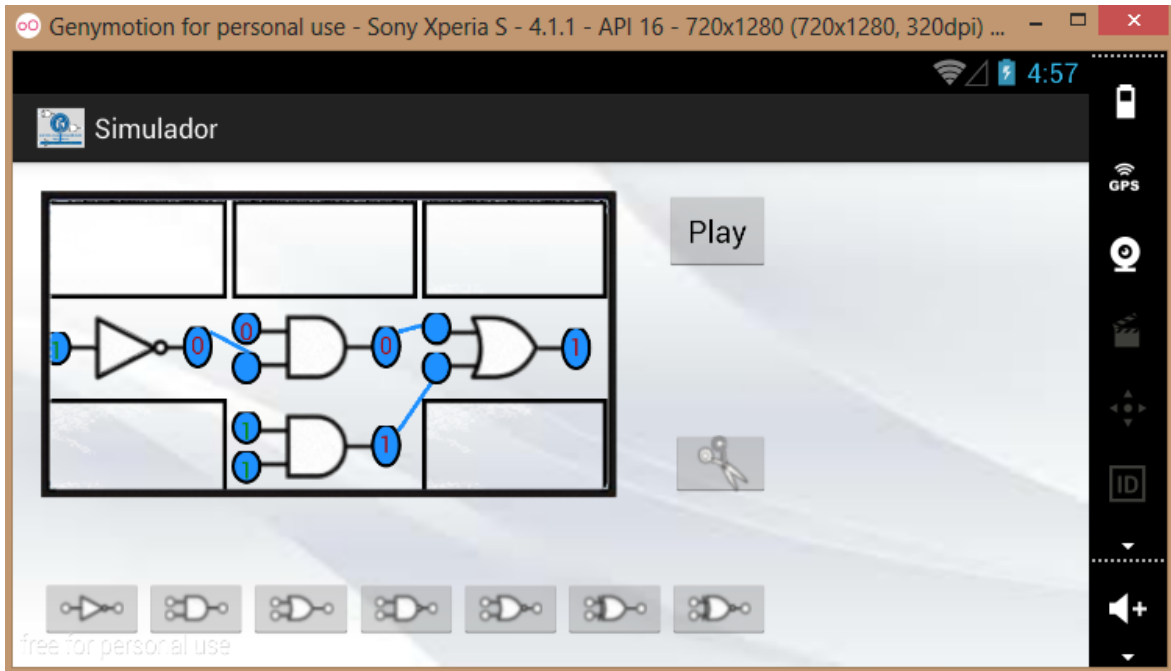


Fig. No 38. Última prueba del módulo simulador en el emulador de dispositivos móviles.

Se puede concluir que las pruebas realizadas en el emulador de dispositivos móviles de la aplicación realizada son exitosas.

9.2 Dispositivo real

Las pruebas en un dispositivo real se realizaron en el dispositivo móvil Sony Xperia E1 con sistema operativo Android versión 4.3 – API 18 y tamaño de pantalla 480 x 800. Estando en el módulo del simulador se disponen las compuertas necesarias para realizar el circuito de la figura No 34. Y se da clic en el botón Play para que el simulador muestre el resultado del circuito elaborado. Se realizaron las mismas pruebas realizadas en el emulador de dispositivos móviles, para comprobar el verdadero funcionamiento de la aplicación. En la figura No 37 se muestra el dispositivo real donde se elaboró la primera prueba (#1).

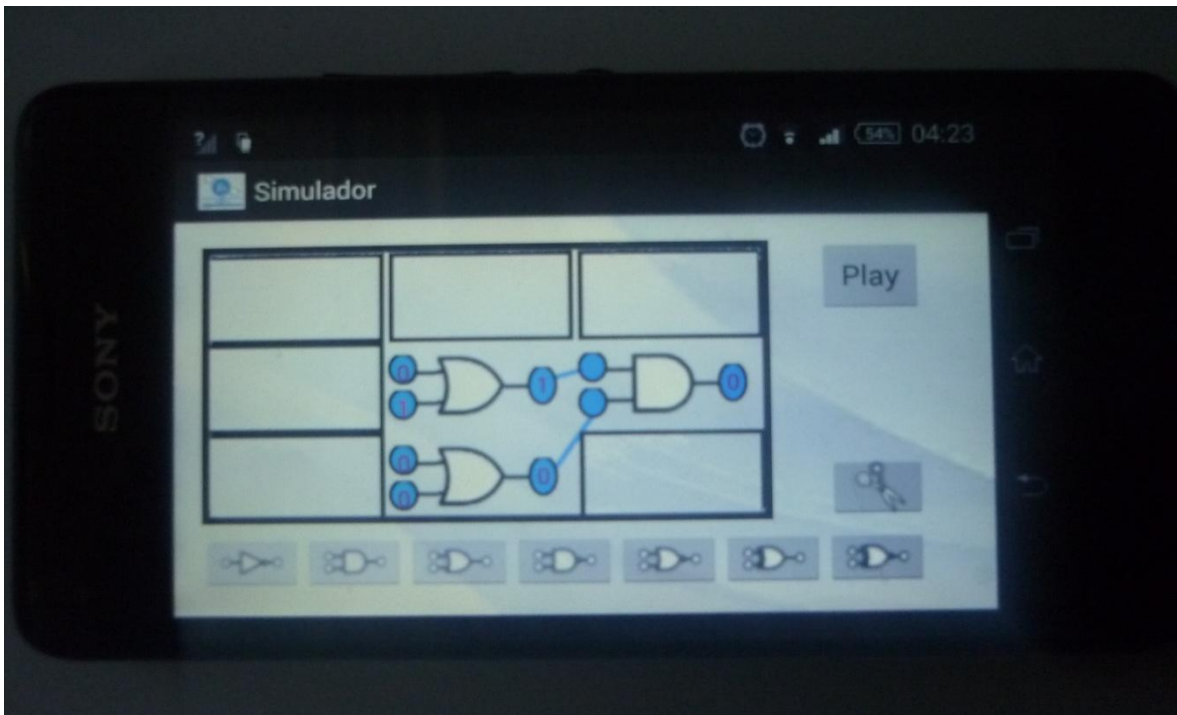


Fig. No 39. Simulación y resultado del circuito planteado en la prueba #1 en el dispositivo móvil real.

Para la segunda prueba (#2) se muestra la salida del circuito cuando:

$$A = 1, \quad B = 0 \quad y \quad C = 1$$

La siguiente imagen muestra esta prueba en el dispositivo real.

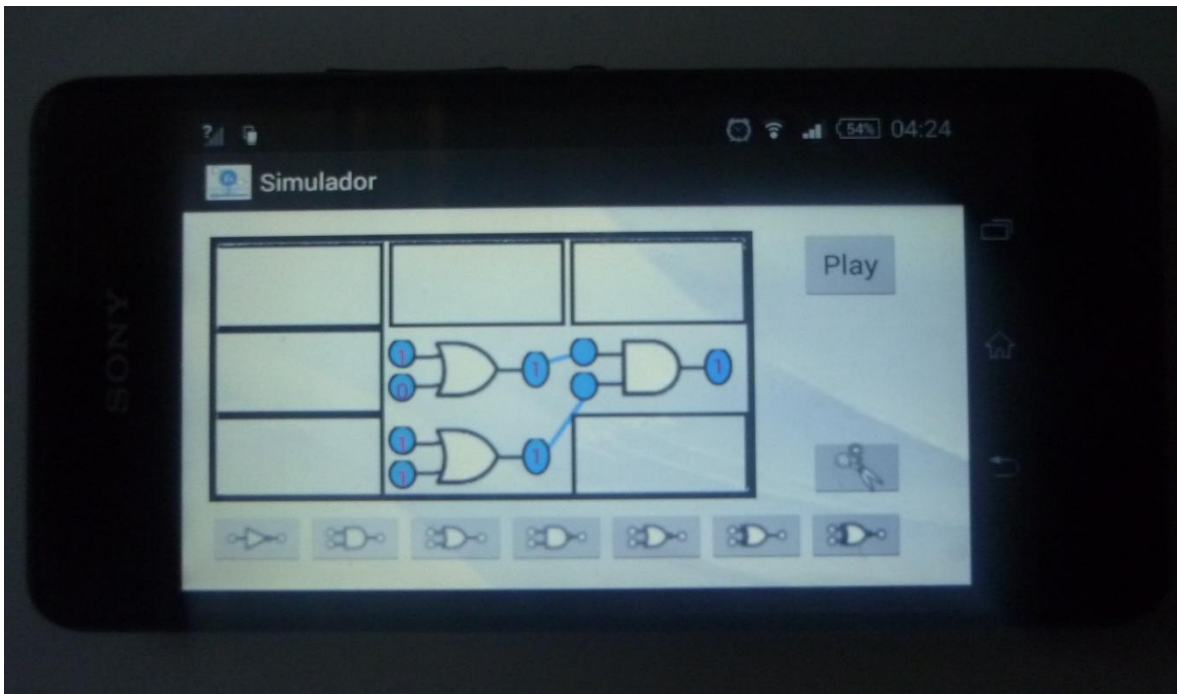


Fig. No 40. Simulación y resultado del circuito planteado en la prueba #2 en el dispositivo real.

Para finalizar la prueba del simulador de la aplicación en el dispositivo móvil real, se selecciona el circuito lógico de la prueba (#3). La elaboración y resultado del circuito de la figura No 37, en el simulador de la aplicación realizada se muestra a continuación:

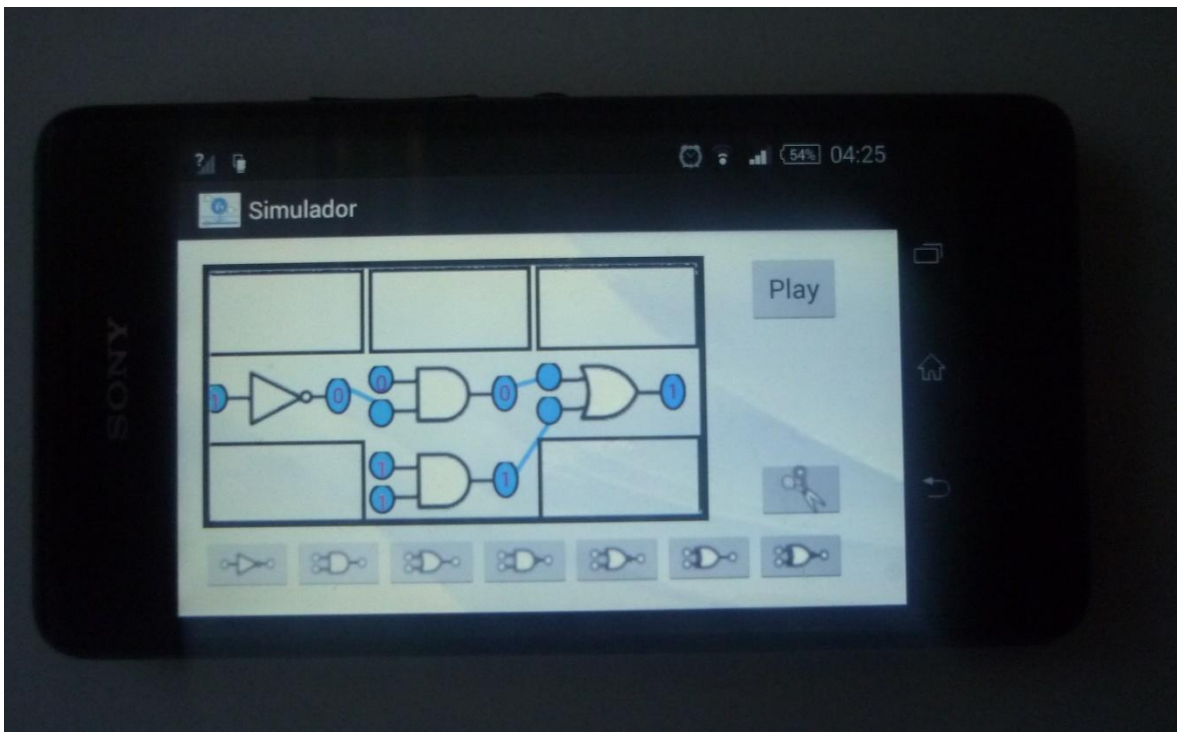


Fig. No 41. Última prueba del módulo simulador en el dispositivo móvil real.

A continuación se muestra la visualización de los otros módulos de la aplicación en el dispositivo móvil real para comprobar su funcionamiento.

En la siguiente imagen se puede observar el dispositivo móvil real en el módulo de fundamentos teóricos sobre circuitos lógicos básicos, en el punto sobre simbología lógica, donde es pulsada una de las imágenes para obtener por pantalla el nombre de la compuerta lógica digital.

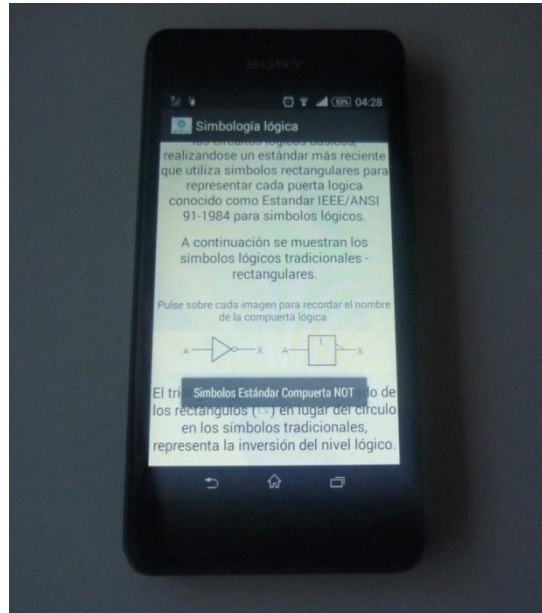


Fig. No 42. Prueba del módulo fundamentos teóricos sobre circuitos lógicos básicos en el dispositivo móvil real.

Por último, se muestra el módulo sobre datos generales de las compuertas lógicas digitales.

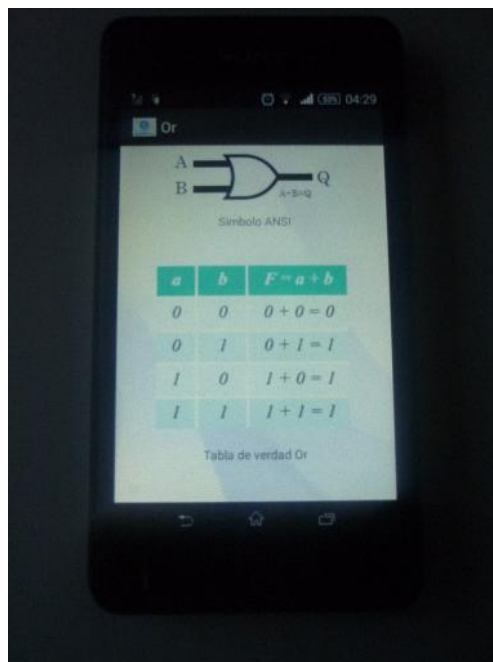


Fig. No 43. Prueba del módulo de datos generales sobre compuertas lógicas digitales en un dispositivo real.

10 Evaluación

Los cuestionarios aplicados fueron realizados para identificar la opinión y aceptación de usuarios principiantes y experimentados acerca de la aplicación realizada, dentro de los cuales se encuentran estudiantes de grado 11, universitarios y docentes. El formato de los cuestionarios aplicados se puede observar en el anexo No 18. Sus resultados se observan a continuación.

10.1 Resultados de los cuestionarios a estudiantes de grado undécimo

El cuestionario fue aplicado a doce (12) estudiantes del grado once (11) del Instituto Nocturno de Educación Media INEM, se obtuvo como resultado una buena aceptación de la aplicación, así lo muestra el promedio de respuestas por pregunta, dado que ninguna de estas alcanzó un valor inferior a 2.5 que es el valor en el que se deben generar cambios o modificaciones.

N.º estudiantes Pregunta	1	2	3	4	5	6	7	8	9	10	11	12	Prom.
Facilidad de uso	3	4	4	5	3	4	4	3	4	4	4	3	3.75
Facilidad de exploración	4	4	4	5	4	4	4	4	3	5	3	3	3.91
¿Le facilita la comprensión del tema?	3	3	4	4	4	3	5	2	2	5	3	2	3.33
¿Son claros los contenidos?	3	5	3	4	4	5	5	3	3	4	3	2	3.66
¿Ha despertado interés en usted?	2	4	2	5	3	4	5	5	3	5	3	3	3.66
¿Son adecuados los recursos para comprender el tema?	4	5	3	4	3	5	5	3	3	4	3	3	3.75

Tabla No 18. Resultados de la aplicación del cuestionario a estudiantes de grado once (11) del colegio INEM.

La escala de calificación implementada fue la siguiente:

1 = Malo, 2= Regular, 3= Bueno, 4= Muy Bueno y 5 = Excelente.

A partir del cuestionario aplicado se obtuvieron las siguientes gráficas estadísticas.

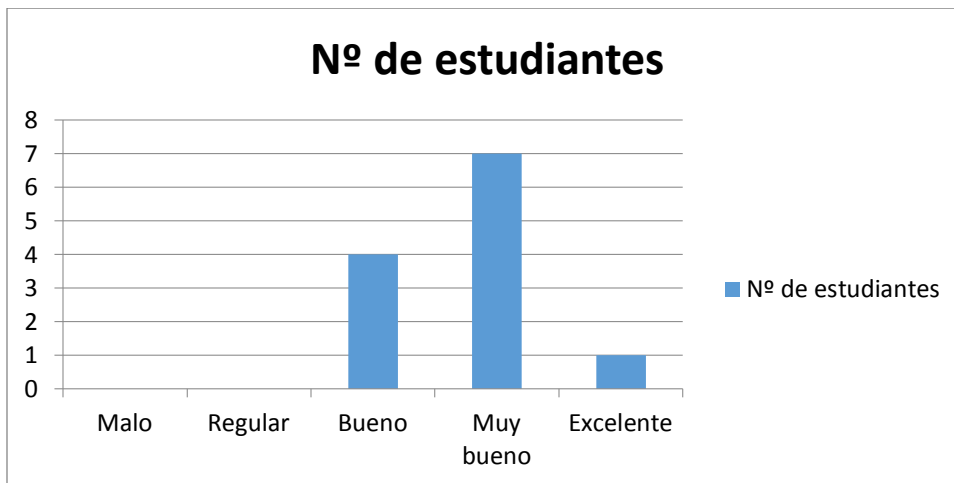


Fig. No 44. Resultado de la pregunta: Facilidad de uso.

Para la mayoría de estudiantes la aplicación es fácil de utilizar desde cualquier módulo en el que se encuentren.



Fig. No 45. Resultado de la pregunta: Facilidad de exploración.

Con relación al diseño de la aplicación que plantea “*el diseño centrado en el usuario*” se reflejó en la opinión de los estudiantes, ya que la exploración o navegación entre cada módulo se realizó de forma rápida y eficiente, evitando un mayor número de pasos para llegar a cada tema. Esta pregunta obtuvo el mayor promedio entre las que se realizaron en el cuestionario.

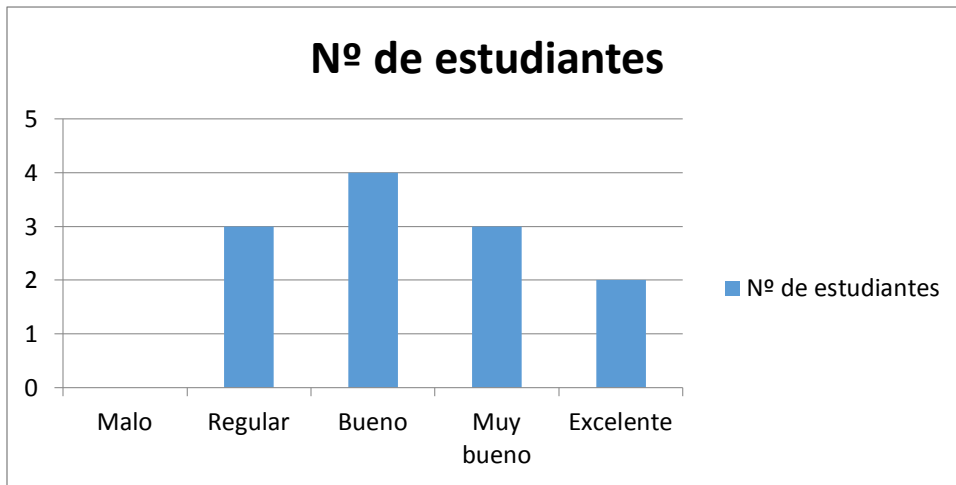


Fig. No 46. Resultado de la pregunta: ¿Le facilita la comprensión del tema?

Los resultados encontrados en la pregunta: *¿Le facilita la comprensión del tema?* Muestra que se debe dar el acompañamiento del docente para aclarar algunos conceptos y orientar al alumno con ayuda de la aplicación. Esta pregunta obtuvo el menor promedio entre las preguntas realizadas en el cuestionario aplicado a estudiantes de grado undécimo.

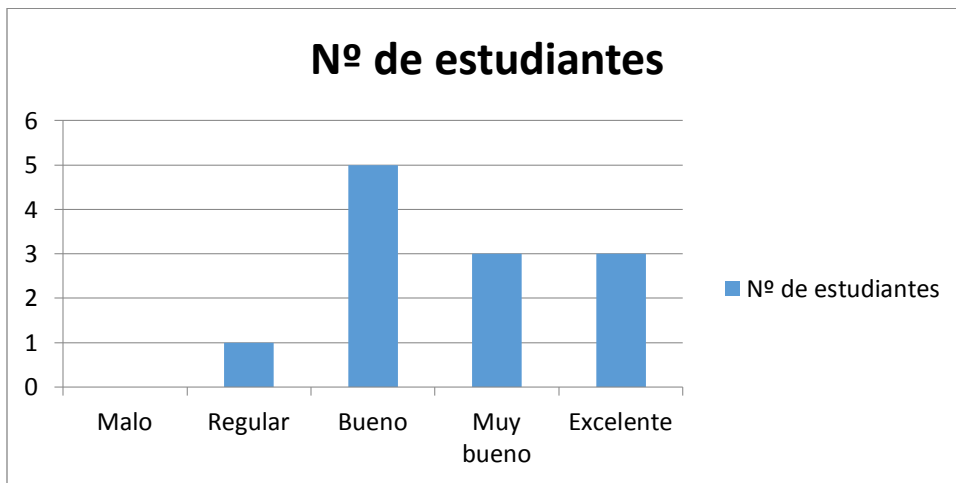


Fig. No 47. Resultado de la pregunta: ¿Son claros los contenidos?

En cuanto a los contenidos, cada uno de estos fue analizado y sacado de libros que manejan con mayor claridad estos temas para presentarlos en la aplicación y que los estudiantes comprendieran los conceptos primordiales.

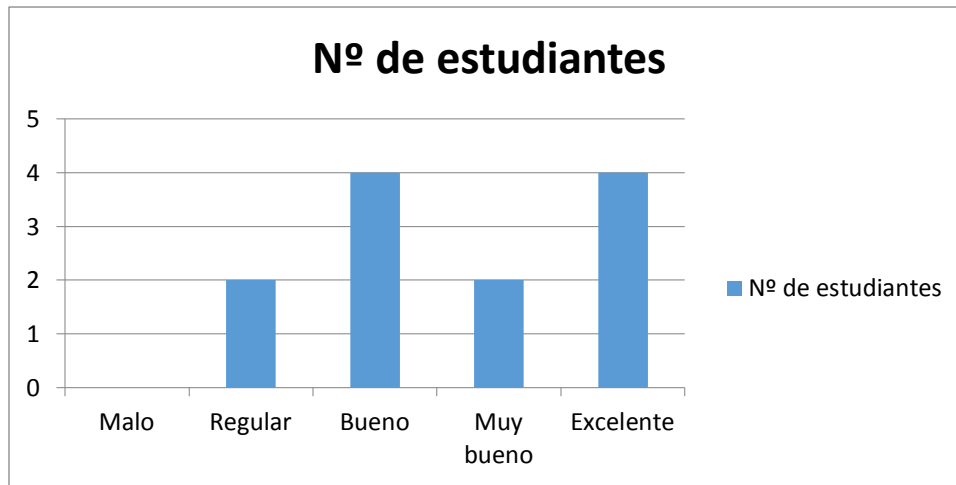


Fig. No 48. Resultado de la pregunta: ¿Ha despertado interés en usted?

Para la mayoría de estudiantes la aplicación fue interesante, debido a la incursión de la tecnología en el ámbito académico y específicamente en lo relacionado a la electrónica digital.

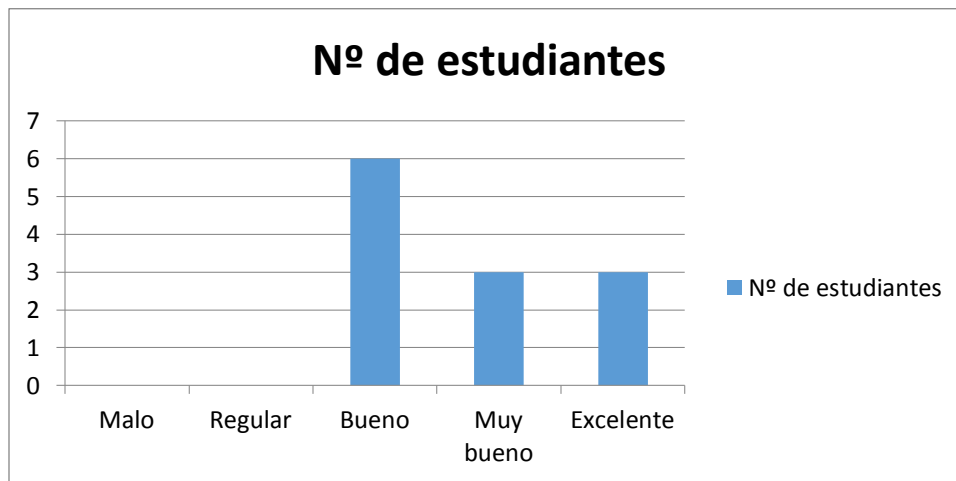


Fig. No 49. Resultado de la pregunta: ¿Son adecuados los recursos para comprender el tema?

Para los estudiantes, cada recurso utilizado en la aplicación aporta los elementos necesarios para aprender sobre el tema. El simulador permite comprobar el funcionamiento de cada compuerta, los fundamentos teóricos amplían el tema sobre circuitos lógicos básicos y los datos generales son un módulo de información rápida de las compuertas lógicas.

Algunos comentarios realizados por los estudiantes sobre la aplicación en la pregunta abierta fueron los siguientes:

1.	<i>Es una buena aplicación de interés académico.</i>
2.	<i>Es una aplicación dinámica.</i>
3.	<i>Muy interesante ya que no conocía la aplicación.</i>
4.	<i>Me gustaron los aportes obtenidos en electrónica.</i>

Tabla No 19. Sugerencias y comentarios de estudiantes que contestaron el cuestionario.

Finalmente, se presentan algunas imágenes en la que los estudiantes resuelven el cuestionario e interactúan con la aplicación.

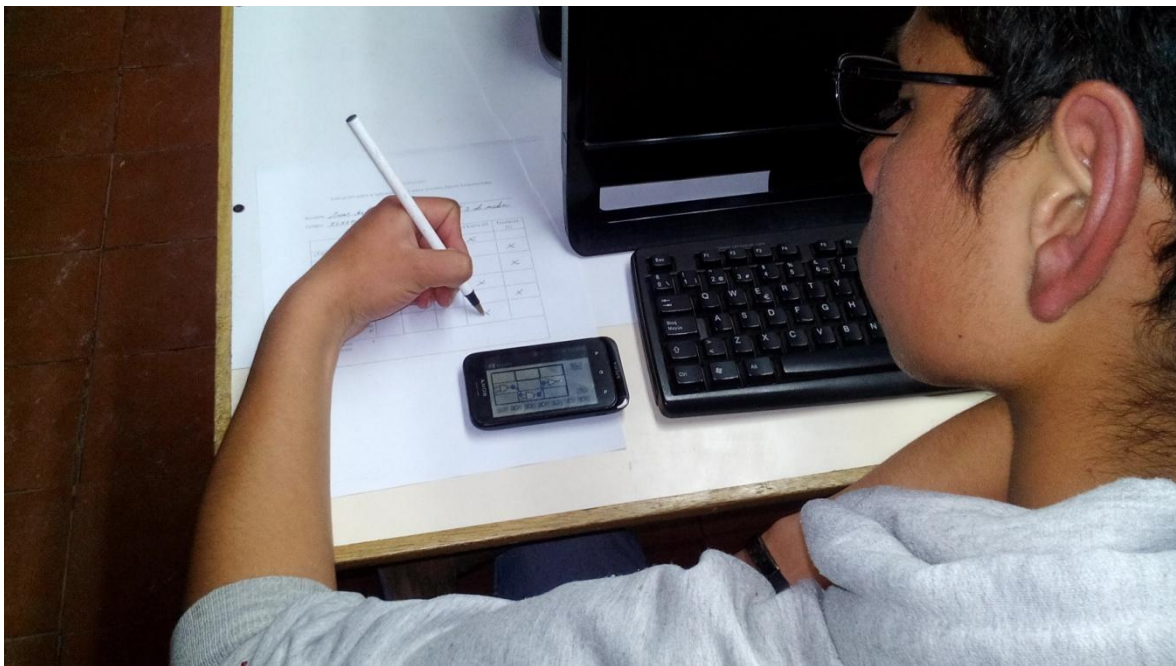


Fig. No 50. Realización de un circuito combinatorio en el simulador de la aplicación por parte de uno de los estudiantes del colegio INEM.

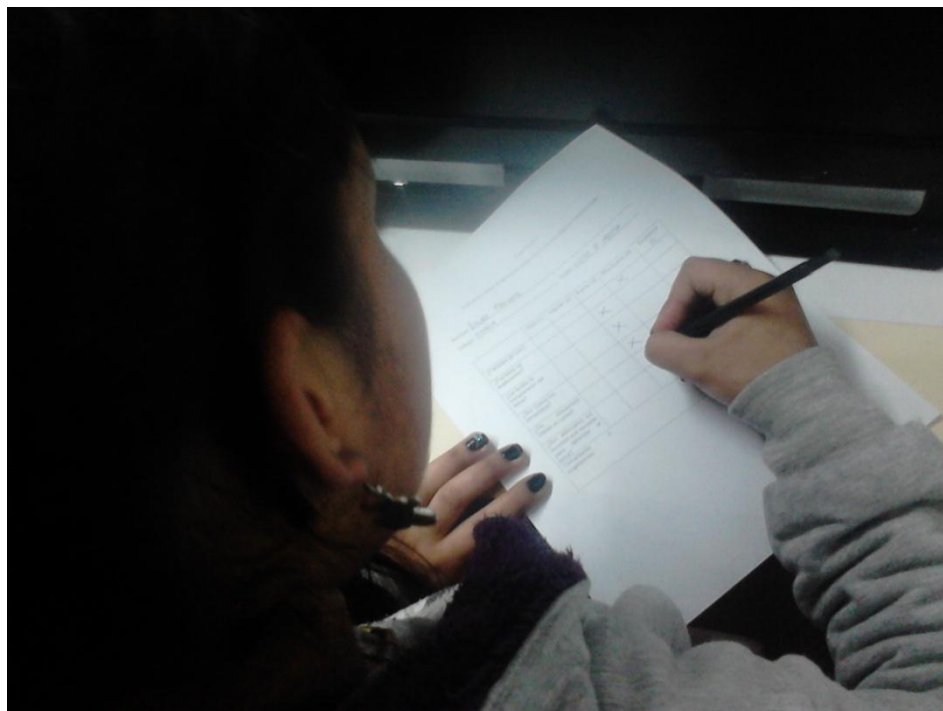


Fig. No 51. Respuestas de uno de los estudiantes del colegio INEM.



Fig. No 52. Verificación del módulo de datos generales de compuertas lógicas por parte de un estudiante del colegio INEM.

10.2 Resultados del cuestionario a docentes y estudiantes universitarios.

El cuestionario para docentes y estudiantes universitarios fue aplicado a tres (3) docentes y dos (2) estudiantes que manejan el tema de electrónica digital de la Universidad Pedagógica Nacional. Obteniendo una buena aceptación de la aplicación dirigida al apoyo didáctico para estudiantes de bachillerato y universitarios que están incursionando en el tema de los circuitos lógicos fundamentales.

La escala de calificación planteada en este cuestionario fue de:

1 = Muy malo, 2= Malo, 3= Normal, 4= Bueno y 5 = Muy bueno.

Como se puede ver en la tabla No 20, al igual que el anterior cuestionario aplicado a estudiantes de bachillerato, ninguno de estos promedios alcanzó un resultado inferior a 2.5 , por lo cual, se sustenta la aceptación de la aplicación por parte de un jurado que maneja el tema y ofrece un veredicto confiable en sus respuestas.

Los resultados del cuestionario aplicado se puede observar a continuación:

Pregunta	Docentes y EST. Universitarios					Prom.
	1 Diego Acero	2 Fabio González.	3 Hellmunth A. Stacey	4 Arley Rodríguez	5 Carlos Roldan	
Organización General	4	4	4	3	3	3.6
Facilidad de navegación.	2	4	3	3	4	3.2
Facilidad de uso.	3	3	4	4	4	3.6
Simulador de compuertas lógicas.	5	4	4	5	5	4.6
Módulo de fundamentos Circuitos lógicos.	5	3	3	4	4	3.8
Módulo sobre datos generales compuertas lógicas.	5	4	4	5	4	4.4
Contenido teórico.	4	3	3	4	4	3.6

Tabla No 20. Resultados de la aplicación del cuestionario a docentes y estudiantes universitarios de la Universidad Pedagógica Nacional de Colombia.

La siguiente gráfica muestra los promedios de las preguntas realizadas a docentes y estudiantes universitarios.

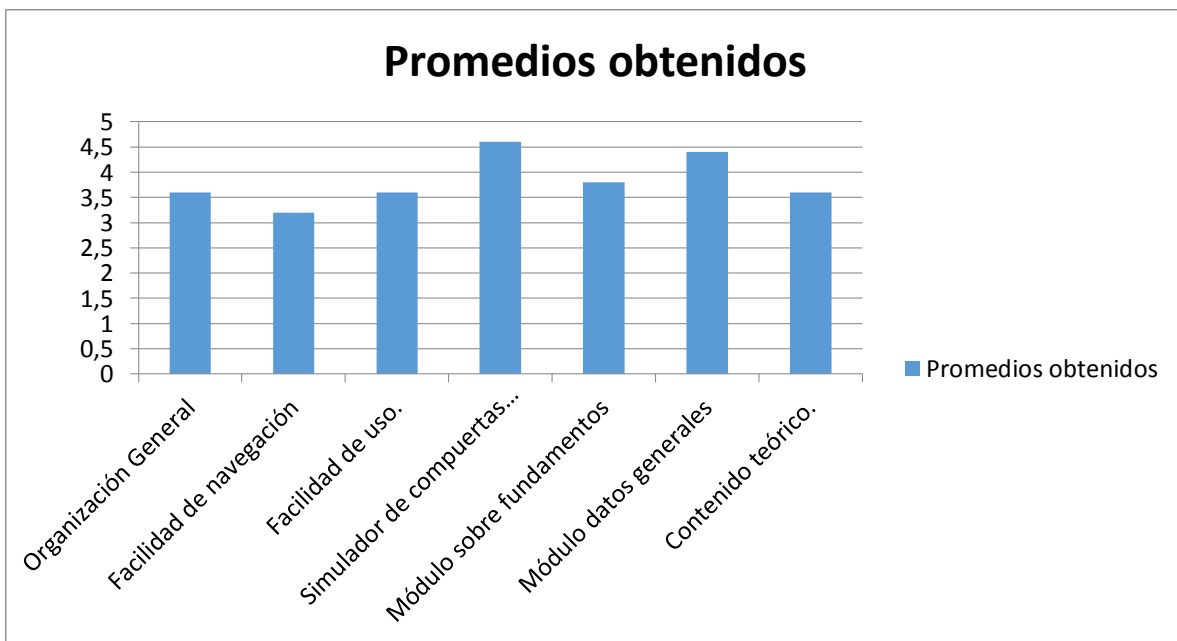


Fig. No 53. Promedios obtenidos de las respuestas de docentes y estudiantes de la Universidad Pedagógica Nacional.

En la anterior gráfica se resalta que el mayor promedio obtenido lo tuvo el módulo de simulación de circuitos lógicos combinatorios, permitiendo recalcar la aprobación de este módulo y en general una buena aceptación de la aplicación.

Sin embargo, es extraño observar que el menor promedio en este cuestionario se dio en la navegación o exploración realizada por parte de los usuarios teniendo en cuenta que el esquema general de la aplicación apunta a un diseño centrado en el usuario que busca la facilidad de la navegación entre cada módulo y actividad, evitando el mayor número de pasos o clics para llegar a cada uno de ellos. Y que en el cuestionario aplicado a estudiantes de grado undécimo esta pregunta obtuvo el mayor promedio, presentándose una contradicción de opiniones.

Las coincidencias entre usuarios novatos y experimentados radican en la aceptación de cada uno de los módulos realizados en la aplicación.

Dando pie para afirmar que este tipo de tecnología dan un incentivo a los estudiantes, cuando se utiliza adecuadamente en la búsqueda del conocimiento.

Los cuestionarios realizados por estudiantes y docentes se pueden observar en el anexo 19 y 20 respectivamente.

11 Descarga e instalación de la aplicación

El proyecto completo y el archivo .apk para ser revisado e instalado debe ser descargado en la siguiente dirección electrónica:

<https://github.com/jaiper1024/Aplicacion-Movil/blob/master/README.md>

Allí hallará las indicaciones para acceder a estos archivos. Pulsando sobre *Aplicación-Móvil* que se encuentra en color azul en la parte superior de esta página. Al hacerlo, será dirigido a otra pantalla donde están los archivos .apk con los siguientes nombres: *Comlogic_UPN.apk* y *Comlogic_UPN2.apk*. Debe escoger el archivo que se ajuste a la pantalla de su móvil. Mientras que el proyecto completo responde al nombre de *Comlogic_UPN.rar*, al pulsar sobre alguno de estos nombres irá a la página de descarga donde deberá pulsar en [view the full file](#), de inmediato comenzará la descarga del archivo.

Recuerde que si desea descargar la aplicación e instalarla debe configurar su móvil para que acepte aplicaciones de fuentes desconocidas.

12 Manual de Usuario

Una vez instalada, el usuario accede al menú principal de la aplicación el cual está compuesto por los tres módulos. En sí, la aplicación en general es fácil de utilizar

a diferencia del simulador el cual se debe seguir las indicaciones puestas en el manual del simulador.

Por esta razón, se mencionan las instrucciones para utilizar el simulador de circuitos lógicos combinatorios tanto en el documento como la aplicación.

- Colocar una compuerta en el área de simulación.

El área de simulación está compuesta por nueve rectángulos como se puede observar en la figura No 54. Seleccione uno de estos rectángulos y a continuación seleccione la compuerta deseada de las siete que se encuentran en la parte inferior de esta interfaz. En total serán nueve compuertas que podrá tener en esta área para su simulación.

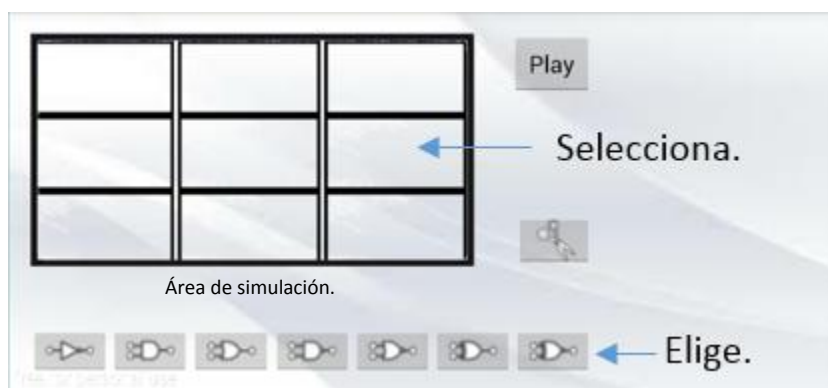


Fig. No 54. Interfaz gráfica del simulador. Instrucción para colocar una compuerta lógica en el área de simulación.

- Cambiar los datos de entrada de una compuerta lógica que se encuentra en el área de simulación.

Una vez puesta cualquier compuerta en el área de simulación pulse sobre alguna de sus entradas para cambiar el dato de 0 a 1 o viceversa.

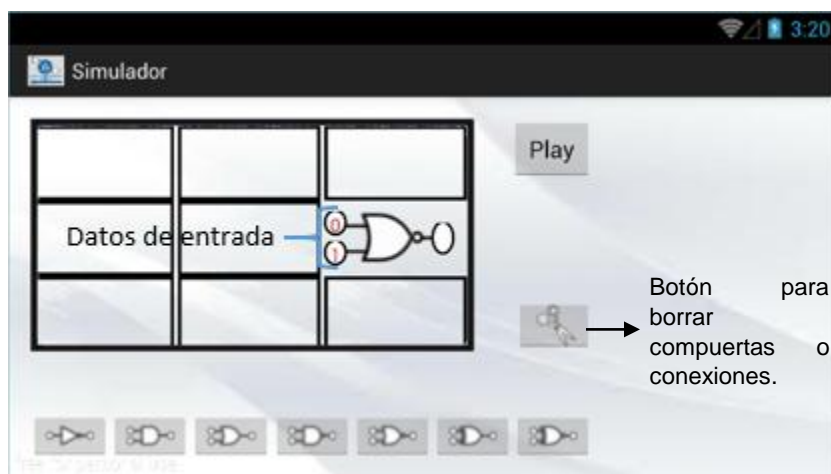


Fig. No 55. Instrucción para cambiar los datos de entrada de una compuerta lógica.

- Cambiar o borrar una compuerta.

Si desea cambiar una compuerta, pulse en el centro de la compuerta que quiere cambiar y de clic en uno de los botones que tiene la imagen de la compuerta que desea poner como remplazo; los cuales se encuentran en la parte inferior de la pantalla.

Si desea borrar la compuerta, pulse en el centro de la compuerta y de clic en el botón de las tijeras para borrarla.

- Realizar una conexión entre compuertas.

Una vez puestas más de una compuerta en el área de simulación, pulse sobre una salida de cualquier compuerta y de clic en la entrada de otra compuerta lógica.

Recuerde que no se deben realizar circuitos en realimentación, debido a que es un simulador de circuitos lógicos combinatorios y no secuenciales.

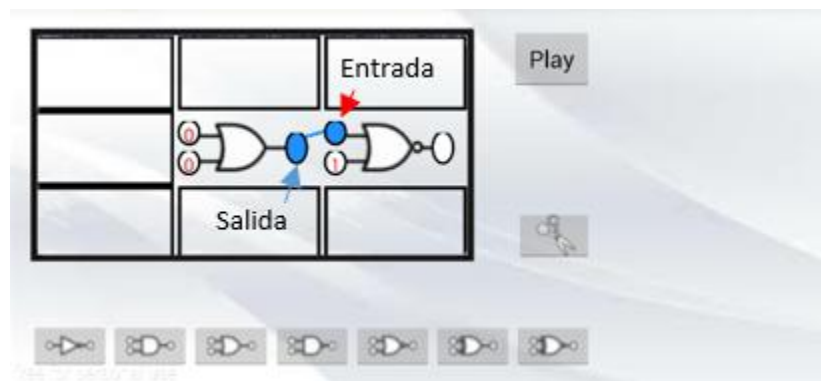


Fig. No 56. Instrucción para realizar una conexión entre compuertas.

- Borrar una conexión entre compuertas

Para borrar una conexión entre compuertas, pulse sobre la entrada de la compuerta conectada y haga clic en el botón de las tijeras.

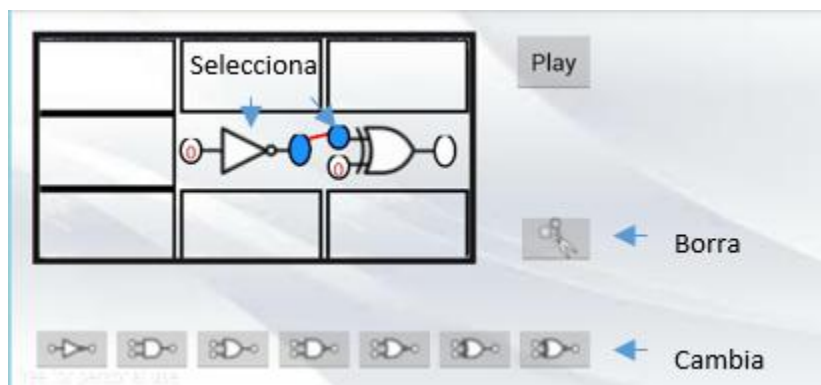


Fig. No 57. Instrucción para quitar una conexión entre compuertas.

- Ver el resultado de las compuertas

Finalmente, para ver el resultado de las compuertas lógicas que se encuentran en el área de simulación de forma individual o varias conectadas entre sí, pulse en el botón de Play para ver el resultado de salida de las compuertas lógicas.

13 CONCLUSIONES

Con el desarrollo del presente trabajo, se logró la realización de una aplicación móvil sobre circuitos lógicos básicos que sirve como herramienta de apoyo didáctico a estudiantes que incursionan en el tema de la electrónica digital.

Las herramientas de software y hardware para el diseño y simulación de la aplicación, permitieron realizar los cambios necesarios para cumplir con los objetivos y los requisitos del sistema, constatando un buen funcionamiento.

El uso de la metodología en *cascada*, permitió obtener resultados en poco tiempo, se dio mayor prioridad a los requerimientos; esto representó una ventaja frente a otras metodologías, ya que se realizó un mantenimiento de la aplicación móvil.

La construcción de la interfaz de usuario de una aplicación móvil en Eclipse Android es un proceso fácil de realizar, debido a la colección de herramientas, elementos y diseños que proporciona este entorno de desarrollo, permitiendo la creación de aplicaciones básicas y sencillas para programadores novatos.

Gracias a la combinación de diferentes recursos, archivos XML y actividades fue posible la realización del simulador de compuertas lógicas básicas, permitió a los usuarios de la aplicación disponer de una herramienta que emula el comportamiento de una compuerta lógica en particular o disponer de varias compuertas lógicas y conectarlas entre sí para ver el resultado de entradas y salidas.

El mayor inconveniente en la realización del proyecto surgió por la falta de experiencia en el lenguaje de programación Java y la plataforma Android; debido a que no se manejaba con precisión los recursos suministrados por estos sistemas y la documentación oficial proporcionada por Google para Android carece de contenido práctico, teniendo que acudir a otros escenarios como foros, blogs y redes sociales para el desarrollo de la aplicación, sin embargo, la ejecución de la aplicación móvil a usuarios principiantes y experimentados demuestra que la aplicación es bien recibida en el ámbito académico.

Las respuestas y comentarios de los estudiantes de grado undécimo demuestran la motivación que este tipo de software genera en ellos con una adecuada supervisión de los docentes.

Al ubicar el proyecto completo y el archivo de instalación de la aplicación en el servidor de GitHub permite a los usuarios disponer del código de este proyecto para identificar, aprender y aplicarlo en la creación de aplicaciones similares a esta y disponer del archivo .apk para la instalación y prueba en dispositivos Android.

14 BIBLIOGRAFÍA

- [1] Barco Gómez, C. (2005). Álgebra Booleana. Aplicaciones tecnológicas. Colombia. Universidad de Caldas.
- [2] Carmen C. Valero. Margarita R. Redondo. Ana S. Palacín. (2012). Tendencias actuales en el uso de dispositivos móviles en educación. La educ@ción. Organization of American States.
- [3] Michael M. Cirovic. (1991) Electrónica fundamental: dispositivos, circuitos y sistemas. USA. Reverte, S. A.
- [4] Zulma Cataldi (2000). Metodología de diseño, desarrollo y evaluación de software educativo. Buenos Aires, Argentina.
- [5] Thomas L. Floyd, (2006). Fundamentos de sistemas digitales. Madrid, España. Pearson Educación S. A.
- [7] José D Muñoz Frías, Introducción a los sistemas digitales, un enfoque usando lenguajes de descripción de hardware. España de Creative Commons.
- [8] Eck J, David. (2010). Introducción a La Programación usando Java Versión 5.1. EUA. Universidad Nacional Abierta de Venezuela.
- [9] Vianney J. Patiño, Roberto A. Ortiz (2012). Aplicación Móvil para la Comisión de Tránsito del Ecuador. Proyecto de grado. Guayaquil, Ecuador.
- [10] Maira C, Gasca. Luis L, Camargo. Byron M, Delgado. Metodología para el desarrollo de aplicaciones móviles. Universidad del Magdalena. Santa Marta, Colombia.
- [11] Salvaor G, Oliver. Curso Programación Android. PDF.
- [12] Darcy Vergara (2008). Arquitectura de la Información Aplicada a Móviles. Seminario Arquitectura de la información. Chile.
- [13] Stephen Brown, Zvonko Vranesic (2006). Fundamentos de lógica digital con diseño VHDL. Mc Graw Hill. Toronto, Canadá.
- [14] ISEA S. Coop. Dentro del marco de la iniciativa e-ISEA para el lanzamiento de un Centro de Experimentación Avanzado en materia de Servicios Electrónicos

(2009). MOBILE LEARNING, Análisis prospectivo de las potencialidades asociadas al Mobile Learning. España.

[15] Oscar B, Figueredo. Producción de Contenidos Educativos para Dispositivos. Universidad de La Sabana. Bogotá, Colombia.

[16] Universidad Marítima Internacional. Simuladores y laboratorios UMIP. Panamá.

[17] Pradel J, Raya (2012). Introducción a la ingeniería de requisitos, obtención de requisitos En. Ingeniería de requisitos. Material docente de la UOC.

[18] Isabel G, Monclova (2013). MyTopRoom, Aplicación para Android: "Buscador de ofertas de Hoteles". Universidad Oberta de Catalunya. España.

[19] Alejandro F, Cortes (2001). Manual de Técnicas para el Diseño Participativo de Interfaces de Usuario de Sistemas basados en Software y Hardware.

[20] James & Suzanne Robertson (2006). Volere. Atlantic Systems Guild.

[21] Pressman S, Roger (2001). Ingeniería de software. Un enfoque práctico. Madrid: Editorial McGraw-Hill/Interamericana de España, SAU

ANEXOS

1. Estructura del archivo AndroidManifest.xml de la aplicación planteada en este documento.

```
<?xmlversion="1.0"encoding="utf-8"?>
<manifestxmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.comlogic_upn"
android:versionCode="1"
android:versionName="1.0">

<uses-sdk
android:minSdkVersion="15"
android:targetSdkVersion="21"/>

<application
android:allowBackup="true"
android:icon="@drawable/ic_launcher"
android:label="@string/app_name"
android:theme="@style/AppTheme">
<activity
android:name=".MainActivity"
android:label="@string/app_name">
<intent-filter>
<actionandroid:name="android.intent.action.MAIN"/>

<categoryandroid:name="android.intent.category.LAUNCHER"/>
</intent-filter>
```

```

</activity>

<activityandroid:name="activity_simulador"
android:label="@string/Simulador"
android:screenOrientation="landscape"></activity>

<activityandroid:name="activity2"
android:label="@string/Guia_de_aprendizaje"></activity>
<activityandroid:name="activityInt"
android:label="@string/Introduccion"></activity>
<activityandroid:name="activityOpe"
android:label="@string/Operaciones"></activity>
<activityandroid:name="activityOpesuma"
android:label="@string/Opesuma"></activity>
<activityandroid:name="activityOpepro"
android:label="@string/Opeproducto"></activity>
<activityandroid:name="activityOpecom"
android:label="@string/Opecom"></activity>
<activityandroid:name="activityBoole"
android:label="@string/Boole"></activity>
<activityandroid:name="activityComp"
android:label="@string/Comp"></activity>
<activityandroid:name="activityCompnot"
android:label="@string/Comp_not"></activity>
<activityandroid:name="activityCompand"
android:label="@string/Comp_and"></activity>
<activityandroid:name="activityCompor"
android:label="@string/Comp_or"></activity>
<activityandroid:name="activityCompnand"

<activityandroid:name="activity"
android:label="@string/Base_de_datos"></activity>
<activityandroid:name="activitynot"
android:label="@string/Base_not"></activity>
<activityandroid:name="activityand"
android:label="@string/Base_and"></activity>
<activityandroid:name="activitynand"
android:label="@string/Base_nand"></activity>
<activityandroid:name="activityor"
android:label="@string/Base_or"></activity>
<activityandroid:name="activitynor"
</application>
</manifest>

```

2. Estructura del archivo XML para la creación del menú principal de la aplicación planteada en este documento:

```

<ScrollView
← Creación del Layout principal.

xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools

android:id="@+id/ScrollView1"
← Identificación del Layout.

```

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/fondor" ← Imagen de fondo del Layout.
```

```
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context="com.example.comlogic_upn.MainActivity">
```

```
<AbsoluteLayout
android:id="@+id/AbsoluteLayout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:gravity="center">
```

← Creación de un Layout adicional de distribución de elementos.

```
<LinearLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_x="0dp"
android:layout_y="38dp"
android:gravity="center"
android:orientation="vertical">
```

← Layout adicional para la distribución de botones.

```
<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="match_parent"
android:gravity="center"
android:text="Circuitos Lógicos Fundamentales"
android:
    textAppearance="?android:attr/textAppearanceLarge"
android:textSize="18sp"/>
```

← Código para la creación de texto.

```
<Button
android:id="@+id/boton1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:layout_marginTop="34dp"
android:text="Simulador"/>
```

← Código para la creación de botones sencillos dentro del Layout.

```
<Button
android:id="@+id/button2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="39dp"
android:text="Guia de Aprendizaje"/>
```

← Identificación de los botones.

```
<Button
android:id="@+id/button3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:layout_marginTop="42dp"
android:text="Base de datos"/>
```

```
</LinearLayout>
</AbsoluteLayout>
</ScrollView>
```

3. Creación del Layout RelativeLayout

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.circuitoslogicos.MainActivity">
</RelativeLayout>
```

4. Código para realizar un botón clásico.

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Simulador"/>
```

—————> Texto que se coloca en el botón.

5. Código para elaborar un texto sencillo en la interfaz gráfica.

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="HOLA MUNDO"/>
```

6. Estructura de la clase principal MainActivity.java que define las tareas de los botones del menú principal de la aplicación.

Para identificar uno de los botones:

```
View simul = findViewById(R.id.simulador); —> Identificador del botón.
simul.setOnClickListener(this); —> Y asignación al evento.
```

Dentro del evento onClick se determina la tarea del botón, que en este caso es pasar a otra interfaz gráfica y a otra actividad.

```
@Override
    public void onClick(View vista) { —> Evento onClick.
        // TODO Auto-generated method stub

        if (vista.getId() == findViewById(R.id.simulador).getId()) —> Método de selección
            { —> Del botón específico
```

```
Intent j = new Intent(this, activity_sim.class);
startActivity(j);
}
```

} Envió a la actividad
Correspondiente.

Estructura completa de la clase:

```
package com.example.comlogic_upn;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

publicclass MainActivity extends Activity implements OnClickListener{

@Override
protectedvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

View bot = findViewById(R.id.simulador);
bot.setOnClickListener(this);

View boton2 = findViewById(R.id.guia);
boton2.setOnClickListener(this);

View boton3 = findViewById(R.id.base);
boton3.setOnClickListener(this);    }

@Override
publicvoid onClick(View vista) {

    if(vista.getId()==findViewById(R.id.simulador).getId())
    {
        Intent j = new Intent(this,activity_sim.class);
        startActivity(j);
    }
if(vista.getId()==findViewById(R.id.guia).getId())
    {
        Intent j = new Intent(this,activity_guia.class);
        startActivity(j);
    }
if(vista.getId()==findViewById(R.id.base).getId())
    {
        Intent a = new Intent(this,activity_base.class);
        startActivity(a);    }    }
```

7. Código para la creación de Layouts y diseño completo en el archivo Xml del diseño de la interfaz del índice sobre datos generales de cada compuerta lógica.

Código para la creación del layout *ScrollView* y *LinearLayout*.

```
<ScrollView
android:id="@+id/scrollView1"
```



Identificación del ScrollVeiw.

```
android:layout_width="fill_parent"
android:layout_height="166dp"
```

→ Tamaño del ScrollView.

```
android:layout_x="0dp"
android:layout_y="103dp">
<LinearLayout
android:layout_width="fill_parent"
android:layout_height="254dp"
android:orientation="vertical">
<Button
android:id="@+id/basel"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="NOT"/>
```

┌┐ Posición dentro de la
└┘ interfaz.

```
</LinearLayout>
```

```
</ScrollView>
```

Diseño completo en el archivo Xml del diseño de la interfaz del índice sobre datos generales de cada compuerta lógica.

```
<AbsoluteLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/AbsoluteLayout1"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/fondor">
```

```
<TextView
android:id="@+id/textView1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_x="1dp"
android:layout_y="42dp"
android:gravity="center"
android:text="COMPUERTAS LOGICAS"
android:textAppearance="?android:attr/textAppearanceLarge"/>
```

```
<ScrollView
android:id="@+id/scrollView1"
android:layout_width="fill_parent"
android:layout_height="166dp"
android:layout_x="0dp"
android:layout_y="103dp">
```

```
<LinearLayout
android:layout_width="fill_parent"
android:layout_height="254dp"
android:orientation="vertical">
```

```
<Button
android:id="@+id/basel"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="NOT"/>
```

```

<Button
android:id="@+id/base2"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="AND"/>
<Button
android:id="@+id/baseOr"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="OR"/>
<Button
android:id="@+id/baseNand"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="NAND"/>

</LinearLayout>
</ScrollView>
</AbsoluteLayout>

```

8. Botón personalizado.

Desde el archivo XML del índice de la guía de aprendizaje básica se coloca un botón imagen o Image Button que permite colocar un botón con cualquier imagen:

```

<ImageButton
android:id="@+id/anterior"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_x="31dp"
android:layout_y="317dp"
android:src="@drawable/anterior"/>

```



Dirección donde se encuentra la imagen.

Diseño del botón sin presionar:

```

<shapexmlns:android="http://schemas.android.com/apk/res/android"
android:shape="oval">
<solidandroid:color="#0000CD"/>
</shape>

```



Indica que el botón será redondo.



Fondo del Botón.

Diseño del archivo cuando se presiona:

```

<shapexmlns:android="http://schemas.android.com/apk/res/android"
android:shape="oval">
<solidandroid:color="#87CEFA"/>
</shape>

```

Selector del botón, presionado y sin presionar.

```
<selectorxmlns:android="http://schemas.android.com/apk/res/android">
<itemandroid:state_pressed="true"
android:drawable="@drawable/btn_circle_presead"/> //Estado presionado

<itemandroid:state_focused="true"
android:drawable="@drawable/btn_circle_focused"/>//Estado normal

</item>
<itemandroid:drawable="@drawable/btn_circle_default"></item>

</selector>
```

9. El código en el archivo XML de la galería de botones de forma horizontal.

```
<HorizontalScrollView
android:id="@+id/horizontalScrollView1"
android:layout_width="262dp"
android:layout_height="wrap_content"
android:layout_x="28dp"
android:layout_y="365dp">

<LinearLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="horizontal">

<Button
android:id="@+id/gnot"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:drawableLeft="@drawable/not2"
android:text=" Not "/>

</LinearLayout>
</HorizontalScrollView>
```

Creación de la galería.

LinearLayout para organizar los elementos.

Elemento agregado.

10. Código para realizar los botones imágenes en el archivo XML sobre simbología lógica.

```
<ImageButton
android:id="@+id/esnand"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_x="0dp"
android:layout_y="1144dp"
android:background="#FFFFFF"
android:src="@drawable/est_nand"/>
```

11. Actividad realizada para sacar mensajes por pantalla al dar clic sobre una de las imágenes en la interfaz realizada por el archivo XML sobre simbología lógica.

```
package com.example.comlogic_upn;

public class activitySim extends Activity implements OnClickListener{

    View c1;    View c2;
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
        setContentView(R.layout.guia_simbologia);

        ImageButton esnot = (ImageButton) findViewById(R.id.esnot);
        esnot.setOnClickListener(this);
        ImageButton es = (ImageButton) findViewById(R.id.esand);
        es.setOnClickListener(this);
        ImageButton eso = (ImageButton) findViewById(R.id.esor);
        eso.setOnClickListener(this);
        ImageButton esn = (ImageButton) findViewById(R.id.esnand);
        esn.setOnClickListener(this);
        ImageButton esno = (ImageButton) findViewById(R.id.esnor);
        esno.setOnClickListener(this);
        ImageButton esx = (ImageButton) findViewById(R.id.esxor);
        esx.setOnClickListener(this);
        ImageButton esxn = (ImageButton) findViewById(R.id.esxnor);
        esxn.setOnClickListener(this);

        c1=findViewById(R.id.indic);
        c1.setOnClickListener(this);

        c2=findViewById(R.id.siguient);
        c2.setOnClickListener(this);

        View boton1 = findViewById(R.id.anterio);
        boton1.setOnClickListener(this);
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            String s = new String();

            switch (v.getId()){
                case R.id.esnot: s="Simbolos Estándar Compuerta NOT"; break;
                case R.id.esand: s="Simbolos Estándar compuerta AND"; break;
                case R.id.esor: s="Simbolos Estándar compuerta OR"; break;
                case R.id.esnand: s="Simbolos Estándar compuerta NAND";break;
                case R.id.esnor: s="Simbolos Estándar compuerta NOR"; break;
                case R.id.esxor: s="Simbolos Estándar compuerta XOR"; break;
                case R.id.esxnor: s="Simbolos Estándar compuerta XNOR";break;
                default:
                    break;
            }
            Toast.makeText(getApplicationContext(), ""+s,
                Toast.LENGTH_SHORT).show();
            if(v==c1)
            {

```

```

        Intent z6 = new Intent(this, activity2.class);
        startActivity(z6);
    }
    if (v==c2)
    {
        Intent zu = new Intent(this, activityFuncion.class);
        startActivity(zu);
    }
    if (v.getId()==findViewById(R.id.siguient).getId())
    {
        Intent z2 = new Intent(this, activityFuncion.class);
        startActivity(z2);
    }
    if (v.getId()==findViewById(R.id.anterio).getId())
    {
        Intent z2 = new Intent(this, activityComp.class);
        startActivity(z2);
    }
}
}
}

```

12. Clase “Capanexiones” que interviene en la realización y borrado de las conexiones entre compuertas lógicas.

```

package com.example.comlogic_upn;

public class CapaConexiones extends View
{
    private Bitmap bitmap;
    ArrayList<Conexion> conexiones = new ArrayList<Conexion>();
    Paint paint = new Paint();
    public int offsetY;
    public static String COLOR_CERO="#B40404";
    public CapaConexiones(Context context, AttributeSet attrs) {
        super(context, attrs);
        // TODO Auto-generated constructor stub
        this.setDrawingCacheEnabled(true);
        bitmap = this.getDrawingCache(true);
        paint.setStrokeWidth(5);

        paint.setColor(Color.parseColor(Conector.COLOR_CONEXION));
    }
    @Override
    protected void onDraw(Canvas canvas) {
        // TODO Auto-generated method stub
        super.onDraw(canvas);

        for (Conexion conexion:conexiones)
        {
            if (conexion==ElementoActivo.conexionActiva)
            {
                paint.setColor(Color.parseColor(Conector.COLOR_CONEXION_ACTIVA));
            }
            else
            {

```

```

        paint.setColor(Color.parseColor(Conector.COLOR_CONEXION));
        }
        conexion.graficar(canvas, paint);
    }

    }

    public void borrar(ConectorEntrada conector)//metodo
    {
        if(conector==null)
        {
            conector=ElementoActivo.conexionActiva.conectordeentrada;} else
{
            this.seleccionar(conector);
        }
        if(ElementoActivo.conexionActiva!=null ||
conector!=null)
        {
            if(conector.getId()==R.id.entradal)
            {
                conector.padre.entradal=null;
                conector.valor.setText("0");
                conector.valor.setTextColor(Color.parseColor(COLOR_CERO));
            }
            else
            {
                conector.padre.entrada2=null;
                conector.valor.setText("0");
                conector.valor.setTextColor(Color.parseColor(COLOR_CERO));
            }
            conector.conectado=false;
            this.conexiones.remove(ElementoActivo.conexionActiva);

            invalidate();
        }
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        // TODO Auto-generated method stub
        bitmap = this.getDrawingCache(true);
        int pixel = bitmap.getPixel((int)event.getX(),
(int)event.getY());
        Log.v("pixel", "pixel"+pixel);
        Conexion conexion=null;
        double dist_menor=10000000;
        Point p = new
Point((int)event.getX(), (int)event.getY());
        int n=0;
        for(Conexion c:conexiones)
        {
            double d = c.distancia(p);
            if(d<dist_menor)
            {
                dist_menor=d;
                conexion=c;
            }
        }
    }

```

```

        if(p.x==c.puntoInicial.x && p.y==c.puntoInicial.y
&&
        p.x==c.puntoFinal.x && p.y==c.puntoFinal.y)
    {
        n++;
    }
}

if(!conexiones.isEmpty())
{
    if(n==0)
    {
        this.invalidate();}
}
return super.onTouchEvent(event);
}
public void conectar(ConectorSalida origen, ConectorEntrada
destino)
{
    int[] locationOrigen = new int[2];
    int[] locationDestino = new int[2];

    Compuerta compuertaDestino=(Compuerta)
destino.getParent().getParent();
    destino.padre = (Compuerta)
destino.getParent().getParent();

    if(destino.getId()==R.id.entrada1)
    {
        compuertaDestino.entrada1=(Compuerta)
origen.getParent().getParent();
    }
    else
    {
        compuertaDestino.entrada2=(Compuerta)
origen.getParent().getParent();
    }
    origen.getLocationInWindow(locationOrigen);
    destino.getLocationOnScreen(locationDestino);

    this.conexiones.add(new Conexion(new
Point(locationOrigen[0],locationOrigen[1]-this.offsetY),
new Point(locationDestino[0],locationDestino[1]-this.offsetY),destino));
    this.invalidate();
}
public void seleccionar(ConectorEntrada destino)
{
    for(Conexion c:conexiones)
    {
        if(c.conectordeentrada==destino)
        {
            ElementoActivo.conexionActiva=c;
        }
    }
}

```

```

        this.invalidate();
    }
}

```

13. Archivo XML *simulador* que permite realizar todas las tareas del simulador.

```

<?xmlversion="1.0"encoding="utf-8"?>
<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/RelativeLayout1"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:background="@drawable/descarga"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin">

<com.example.comlogic_upn.CapaConexiones
android:id="@+id/capaConexiones"
android:layout_width="wrap_content"
android:layout_alignBottom="@+id/GridLayout1"
android:layout_height="wrap_content"
android:layout_marginLeft="0dp"
android:layout_marginTop="0dp"/>

<GridLayout
android:id="@+id/GridLayout1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBottom="@+id/fondodetrabajo1"
android:layout_alignParentLeft="true"
android:layout_alignRight="@+id/fondodetrabajo1"
android:columnCount="4"
android:orientation="vertical"
android:rowCount="3">
</GridLayout>

<com.example.comlogic_upn.Fondodetrabajo
android:id="@+id/fondodetrabajo1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentTop="true">
</com.example.comlogic_upn.Fondodetrabajo>

<LinearLayout
android:id="@+id/linearLayout1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/capaConexiones"
android:layout_alignParentBottom="true">

<ImageButton
android:id="@+id/b_not"
android:layout_width="wrap_content"

```

```
android:layout_height="wrap_content"  
android:src="@drawable/not2"/>
```

```
<ImageButton  
android:id="@+id/b_and"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:src="@drawable/and2"/>
```

```
<ImageButton  
android:id="@+id/b_or2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:src="@drawable/or2"/>
```

```
<ImageButton  
android:id="@+id/b_nand2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:src="@drawable/nand2"/>
```

```
<ImageButton  
android:id="@+id/b_nor"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:src="@drawable/nor2"/>
```

```
<ImageButton  
android:id="@+id/b_xor"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:src="@drawable/xor2"/>
```

```
<ImageButton  
android:id="@+id/b_xnor"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:src="@drawable/xnor2"/>  
</LinearLayout>
```

```
<ImageButton  
android:id="@+id/imageButtonRemove"  
android:layout_width="60dp"  
android:layout_height="40dp"  
android:layout_alignBottom="@+id/GridLayout1"  
android:layout_alignRight="@+id/linearLayout1"  
android:src="@drawable/tijeras"/>
```

```
<Button  
android:id="@+id/play"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignRight="@+id/imageButtonRemove"  
android:layout_alignTop="@+id/capaConexiones"  
android:text="Play"/>
```

```
</RelativeLayout>
```

14. Archivo XML del conector de entrada de la aplicación.

```
<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/RelativeLayout1"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">

<View
android:id="@+id/View1"
android:layout_width="18dp"
android:layout_height="26dp"
android:layout_marginLeft="0dp"
android:layout_marginTop="-2dp"
android:background="@drawable/circulo"/>
/>

<TextView
android:layout_marginLeft="5dp"
android:layout_marginTop="1dp"
android:id="@+id/TextViewValor"
android:layout_width="23dp"
android:layout_height="23dp"
android:layout_alignParentLeft="true"
android:layout_alignParentTop="true"
android:text="0"
android:textColor="#B40404"/>

</RelativeLayout>
```

15. Archivo XML de una de las compuertas lógicas que se muestran en el módulo del simulador.

```
<?xmlversion="1.0"encoding="utf-8"?>
<AbsoluteLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/AbsoluteLayout1"
android:layout_width="match_parent"
android:layout_height="match_parent">

<ImageView
android:id="@+id/ImageView2"
android:layout_width="wrap_content"
android:layout_height="58dp"
android:src="@drawable/and"/>

<com.example.comlogic_upn.ConectorSalida
android:id="@+id/salida"
android:layout_width="15dp"
android:layout_height="wrap_content"
android:layout_x="86dp"
android:layout_y="16dp">
</com.example.comlogic_upn.ConectorSalida>

<com.example.comlogic_upn.ConectorEntrada
```

```

android:id="@+id/entrada1"
android:layout_width="wrap_content"
android:layout_height="20dp"
android:layout_x="2dp"
android:layout_y="8dp">
</com.example.comlogic_upn.ConectorEntrada>

<com.example.comlogic_upn.ConectorEntrada
android:id="@+id/entrada2"
android:layout_width="wrap_content"
android:layout_height="20dp"
android:layout_x="2dp"
android:layout_y="29dp">
</com.example.comlogic_upn.ConectorEntrada>

<ImageView
android:id="@+id/imageView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_x="33dp"
android:layout_y="3dp"
android:src="@drawable/and3"/>

</AbsoluteLayout>

```

16. Clase que realiza las tareas del módulo *simulador*.

```

package com.example.comlogic_upn;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import java.util.ArrayList;
import android.os.Bundle;
import android.util.DisplayMetrics;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.GridLayout;
import android.widget.LinearLayout;

public class actividad_simulador extends Activity implements
OnClickListener{

    float iniY;
    float iniX;
    float iniLeft;
    float iniTop;
    GridLayout mainScreen;
    int calculos=0;
    View activa=null;
    ArrayList<Compuerta> compuertas = new ArrayList<Compuerta>();
    Object compuertaActiva=null;
    View b1;    View b2;    View b3;    View b4;    View b5;    View b6;
    View b7;    View b8;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.simulador);

    ElementoActivo.capaConexiones = (CapaConexiones)
    this.findViewById(R.id.capaConexiones);

    this.mainScreen = (GridLayout)
    this.findViewById(R.id.GridLayout1);

    b1=findViewById(R.id.b_not);
    b1.setOnClickListener(this);

    b2=findViewById(R.id.b_and);
    b2.setOnClickListener(this);

    b3=findViewById(R.id.b_or2);
    b3.setOnClickListener(this);

    b4=findViewById(R.id.b_nand2);
    b4.setOnClickListener(this);

    b5=findViewById(R.id.b_nor);
    b5.setOnClickListener(this);

    b6=findViewById(R.id.b_xor);
    b6.setOnClickListener(this);

    b7=findViewById(R.id.b_xnor);
    b7.setOnClickListener(this);

    b8 = findViewById(R.id.imageButtonRemove);
    b8.setOnClickListener(this);

    View base1 = findViewById(R.id.play);
    base1.setOnClickListener(this);

    LinearLayout l1 = new LinearLayout(this);
    LinearLayout l2 = new LinearLayout(this);
    LinearLayout l3 = new LinearLayout(this);
    LinearLayout l4 = new LinearLayout(this);
    LinearLayout l5 = new LinearLayout(this);
    LinearLayout l6 = new LinearLayout(this);
    LinearLayout l7 = new LinearLayout(this);
    LinearLayout l8 = new LinearLayout(this);
    LinearLayout l9 = new LinearLayout(this);

    Remove a = new Remove(this, null);
    a.ma=this;
    this.mainScreen.addView(l1);
    a.linearlayout=l1;

    Remove b = new Remove(this, null);
    b.ma=this;
    this.mainScreen.addView(l2);

```

```

        b.linearlayout=12;

        Remove c = new Remove(this, null);
        c.ma=this;
this.mainScreen.addView(13);
        c.linearlayout=13;

        Remove d = new Remove(this, null);
        d.ma=this;
this.mainScreen.addView(14);
        d.linearlayout=14;

        Remove e = new Remove(this, null);
        e.ma=this;
this.mainScreen.addView(15);
        e.linearlayout=15;

        Remove f = new Remove(this, null);
        f.ma=this;
this.mainScreen.addView(16);
        f.linearlayout=16;

        Remove g = new Remove(this, null);
        g.ma=this;
this.mainScreen.addView(17);
        g.linearlayout=17;

        Remove h = new Remove(this, null);
        h.ma=this;
this.mainScreen.addView(18);
        h.linearlayout=18;

        Remove i = new Remove(this, null);
        i.ma=this;
this.mainScreen.addView(19);
        i.linearlayout=19;

        11.addView(a);           12.addView(b);           13.addView(c);
        14.addView(d);           15.addView(e);           16.addView(f);
        17.addView(g);           18.addView(h);           19.addView(i);

compuertas.add(a);
compuertas.add(b);
compuertas.add(c);
compuertas.add(d);
compuertas.add(e);
compuertas.add(f);
compuertas.add(g);
compuertas.add(h);
compuertas.add(i);

}

@Override
publicvoid onWindowFocusChanged(boolean hasFocus) {
    DisplayMetrics displayMetrics = new DisplayMetrics();

```

```
getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);
```

```
ElementoActivo.capaConexiones.offsetY = displayMetrics.heightPixels -  
mainScreen.getMeasuredHeight() -  
3*b1.getMeasuredHeight(); super.onWindowFocusChanged(hasFocus);
```

```
    }  
    @Override  
    public boolean onTouchEvent(MotionEvent event) {  
        // TODO Auto-generated method stub  
  
        switch(event.getAction())  
        {  
            case MotionEvent.ACTION_MOVE:  
                break;  
            case MotionEvent.ACTION_DOWN:  
  
                break;  
            case MotionEvent.ACTION_UP:  
  
                break;  
  
        }  
        return true;  
    }  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        if(v.getId()==findViewById(R.id.play).getId())  
        {  
            try{  
                for(Compuerta compuerta:compuertas)  
                {  
                    compuerta.evaluar(0);  
                }  
            }  
            catch(Exception e)  
            {  
                AlertDialog.Builder dialog =  
new AlertDialog.Builder(this);  
                dialog.setMessage("                Error de conexión  
\nNo se hacepta realimentación.");  
                dialog.setPositiveButton("Aceptar",  
new DialogInterface.OnClickListener() {  
  
                    @Override  
                    public void onClick(DialogInterface dialog,  
int which) {  
                        dialog.cancel();  
                    }  
                });  
                dialog.show();  
            }  
        }  
    }  
}
```

```

}
elseif(compuertaActiva!=null)
{
    Compuerta activa=(Compuerta) compuertaActiva;
    if(compuertaActivainstanceof Compuerta)
    {
        LinearLayout linearLayout = activa.linearlayout;
        linearLayout.removeView(activa);
        compuertas.remove(compuertaActiva);

        if(v == b1)
        {
            Not a = new Not(this,null);
            a.ma=this;
            linearLayout.addView(a);
            compuertas.add(a);
            a.linearlayout=linearLayout;
            compuertaActiva=null;
        }
        elseif (v == b2)
        {
            And b = new And(this,null);
            b.ma=this;
            linearLayout.addView(b);
            compuertas.add(b);
            b.linearlayout=linearLayout;
            compuertaActiva=null;
        }
        elseif (v == b3)
        {
            Or b = new Or(this,null);
            b.ma=this;
            linearLayout.addView(b);
            compuertas.add(b);
            b.linearlayout=linearLayout;
            compuertaActiva=null;
        }
        elseif (v == b4)
        {
            Nand b = new Nand(this,null);
            b.ma=this;
            linearLayout.addView(b);
            compuertas.add(b);
            b.linearlayout=linearLayout;
            compuertaActiva=null;
        }
        elseif (v == b5)
        {
            Nor b= new Nor(this,null);
            b.ma=this;
            linearLayout.addView(b);
            compuertas.add(b);
            b.linearlayout=linearLayout;
            compuertaActiva=null;
        }
        elseif (v == b6)

```

```

        {
            Xor b = new Xor(this, null);
            b.ma=this;
            linearLayout.addView(b);
compuertas.add(b);
            b.linearlayout=linearLayout;
compuertaActiva=null;
        }
        elseif (v == b7)
        {
            Xnor b = new Xnor(this, null);
            b.ma=this;
            linearLayout.addView(b);
compuertas.add(b);
            b.linearlayout=linearLayout;
compuertaActiva=null;
        }
        elseif(v == b8)
        {
            Remove b = new Remove(this, null);
            b.ma=this;
            linearLayout.addView(b);
compuertas.add(b);
            b.linearlayout=linearLayout;
compuertaActiva=null;
        }
    }
}
}}

```

17. clase de la compuerta And que maneja la operación de la compuerta lógica.

```

package com.example.comlogic_upn;

public class And extends Compuerta implements OnClickListener{
    public activity_simulador ma;
    ImageView iv;
    public And(Context context, AttributeSet attrs) {
        super(context, attrs);
        LayoutInflater mInflater =
(LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE)
        ; mInflater.inflate(R.layout.and, this, true);
        iv = (ImageView) this.findViewById(R.id.imageView2);
        iv.setOnClickListener(this); }
    public void onClick(View v)
{
    ma.compuertaActiva=this; }
    @Override
    public int evaluar(int n) {
        int x;          int y;
        ConectorEntrada e1 = (ConectorEntrada)
this.findViewById(R.id.entrada1);
        ConectorEntrada e2 = (ConectorEntrada)
this.findViewById(R.id.entrada2);
        ConectorSalida s = (ConectorSalida)
this.findViewById(R.id.salida);
        if(n>20)
{throw new UnsupportedOperationException("numero max de iteraciones"); }
        if(this.entrada1==null)

```

```

{x=Integer.parseInt(e1.valor.getText().toString());}
else      {      x=entrada1.evaluar(n+1);
           if(x== -1)
           {
               return -1;}}
if(this.entrada2==null)
{      y=Integer.parseInt(e2.valor.getText().toString());      }
else
{      y=entrada2.evaluar(n+1);
  if(y== -1)
  {
      return -1;}}
s.valor.setText(""+(x*y));
return x*y; } }

```

18. Cuestionarios aplicados a estudiantes de grado once, docentes y estudiantes de electrónica.

Cuestionario 1

Evaluación de la aplicación móvil sobre circuitos lógicos fundamentales por parte de estudiantes de grado 11.

Nombre: _____

Colegio: _____ Grado: _____

	Malo (1)	Regular (2)	Bueno (3)	Muy Bueno (4)	Excelente (5)
Facilidad de uso					
Facilidad de exploración					
¿Le facilita la comprensión del tema?					
¿Son claros los contenidos?					
¿Ha despertado interés en usted?					
¿Son adecuados los recursos que necesita para aprender el tema?					
Comentarios y sugerencias:					

Donde la escala de calificación es de: 1 = Malo a 5 = Excelente.

Cuestionario 2

Evaluación de la aplicación móvil sobre circuitos lógicos fundamentales por parte de docentes y estudiantes de electrónica.

Nombre: _____

Ocupación: _____

	Muy malo (1)	Malo (2)	Normal (3)	Bueno (4)	Muy bueno (5)
Organización General					
Facilidad de navegación.					
Facilidad de uso.					
Simulador de compuertas lógicas.					
Módulo de fundamentos de Circuitos lógicos.					
Módulo sobre datos generales compuertas lógicas.					
Contenido teórico.					
Comentarios y sugerencias:					

Donde la escala de calificación es de: 1 = Muy malo a 5 = Muy bueno.

19. Cuestionarios resueltos por estudiantes de grado 11º del colegio INEM.

Cuestionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Patricia Galindo

Colegio: INEM Grado: ciclo II media

	Malo (1)	regular (2)	Bueno (3)	Muy bueno (4)	Excelente (5)
¿Facilidad de Uso?			X		
¿Facilidad de exploración?				X	
¿Le facilita la comprensión del tema?			X		
¿Son Claros los contenidos?			X		
¿Ha despertado interés en usted?		X			
¿Son adecuados los recursos que necesita para aprender el tema?				X	
Comentarios y sugerencias: <i>Buena aplicación y de interés. Académico.</i>					

Cuestionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Yerli Patricia Arvalo Rca

Colegio: ASNEH Grado: Ciclo II Medio

	Malo (1)	regular (2)	Bueno (3)	Muy bueno (4)	Excelente (5)
¿Facilidad de Uso?				X	
¿Facilidad de exploración?				X	
¿Le facilita la comprensión del tema?			X		
¿Son Claros los contenidos?					X
¿Ha despertado interés en usted?				X	
¿Son adecuados los recursos que necesita para aprender el tema?					X
Comentarios y sugerencias: <u>Felicitaciones !!!</u>					

Cuestionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Jose Manuel Trujillo

Colegio: Inem Grado: 11

	Malo (1)	regular (2)	Bueno (3)	Muy bueno (4)	Excelente (5)
¿Facilidad de Uso?				X	
¿Facilidad de exploración?				X	
¿Le facilita la comprensión del tema?				X	
¿Son Claros los contenidos?			X		
¿Ha despertado interés en usted?		X			
¿Son adecuados los recursos que necesita para aprender el tema?			X		
Comentarios y sugerencias:					

Questionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Thalia Moreno Bolaños

Colegio: ISNEM Grado: 2 de media

	Mal (1)	regular (2)	Bueno (3)	Muy bueno (4)	Excelente (5)
¿Facilidad de Uso?					X
¿Facilidad de exploración?					X
¿Le facilita la comprensión del tema?				X	
¿Son Claros los contenidos?				X	
¿Ha despertado interés en usted?					X
¿Son adecuados los recursos que necesita para aprender el tema?				X	
Comentarios y sugerencias:	y Buena Dinámica.				

Cuestionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Lamila Torres

Colegio: Inem Grado: 11

	Malo (1)	regular (2)	Bueno (3)	Muy bueno (4)	Excelente (5)
¿Facilidad de Uso?			X		
¿Facilidad de exploración?				X	
¿Le facilita la comprensión del tema?				X	
¿Son Claros los contenidos?				X	
¿Ha despertado interés en usted?			X		
¿Son adecuados los recursos que necesita para aprender el tema?			X		
Comentarios y sugerencias:					

Cuestionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Elizabeth Gil Morales.
 Colegio: Instituto Enseñanza II. Grado: II de Media

	Malo (1)	regular (2)	Bueno (3)	Muy bueno (4)	Excelente (5)
¿Facilidad de Uso?				X	
¿Facilidad de exploración?				X	
¿Le facilita la comprensión del tema?			X		
¿Son Claros los contenidos?					X
¿Ha despertado interés en usted?				X	
¿Son adecuados los recursos que necesita para aprender el tema?					X
Comentarios y sugerencias: Felicidades					

Questionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Angelica Rodriguez

Colegio: Isnem Grado: 11^o

	Malo (1)	regular (2)	Bueno (3)	Muy bueno (4)	Excelente (5)
¿Facilidad de Uso?				X	
¿Facilidad de exploración?				X	
¿Le facilita la comprensión del tema?					X
¿Son Claros los contenidos?					X
¿Ha despertado interés en usted?					X
¿Son adecuados los recursos que necesita para aprender el tema?					X
Comentarios y sugerencias:					

Cuestionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Andrés Paez

Colegio: ISNEM Grado: ciclo II media

	Malo (1)	regular (2)	Bueno (3)	Muy bueno (4)	Excelente (5)
¿Facilidad de Uso?			X		
¿Facilidad de exploración?				X	
¿Le facilita la comprensión del tema?		X			
¿Son Claros los contenidos?			X		
¿Ha despertado interés en usted?					X
¿Son adecuados los recursos que necesita para aprender el tema?			X		
Comentarios y sugerencias:					
Muy interesante ya que no conocia la aplicación.					

Questionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Kevin Petro

Colegio: Ionem Grado: 11º

	Malo (1)	regular (2)	Bueno (3)	Muy bueno (4)	Excelente (5)
¿Facilidad de Uso?				X	
¿Facilidad de exploración?			X		
¿Le facilita la comprensión del tema?		X			
¿Son Claros los contenidos?			X		
¿Ha despertado interés en usted?			X		
¿Son adecuados los recursos que necesita para aprender el tema?			X		
Comentarios y sugerencias:	Me gusto, los q partes obtenidos sobre electronica.				

Questionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Diego Andres Ramirez Tobarco

Colegio: ISNEM Grado: octavo 2 de media

	Malo (1)	regular (2)	Bueno (3)	Muy bueno (4)	Excelente (5)
¿Facilidad de Uso?				X	
¿Facilidad de exploración?					X
¿Le facilita la comprensión del tema?					X
¿Son Claros los contenidos?				X	
¿Ha despertado interés en usted?					X
¿Son adecuados los recursos que necesita para aprender el tema?				X	
Comentarios y sugerencias:					

Questionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Laura Barrera

Colegio: ISNEM Grado: CICLO II MEDIA

	Malo (1)	regular (2)	Bueno (3)	Muy bueno (4)	Excelente (5)
¿Facilidad de Uso?				X	
¿Facilidad de exploración?			X		
¿Le facilita la comprensión del tema?			X		
¿Son Claros los contenidos?			X		
¿Ha despertado interés en usted?			X		
¿Son adecuados los recursos que necesita para aprender el tema?			X		
Comentarios y sugerencias: EXELENTE APLICACIÓN!!!					

Cuestionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Luisa Jimena Suarez

Colegio: Inem Grado: 11^o

	Malo (1)	regular (2)	Bueno (3)	Muy bueno (4)	Excelente (5)
¿Facilidad de Uso?			X		
¿Facilidad de exploración?			X		
¿Le facilita la comprensión del tema?		X			
¿Son Claros los contenidos?		X			
¿Ha despertado interés en usted?			X		
¿Son adecuados los recursos que necesita para aprender el tema?			X		
Comentarios y sugerencias:					

20. Cuestionarios resueltos por docentes y estudiantes de la universidad pedagógica nacional de Colombia.

Cuestionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: DIEGO U. ACERO

Ocupación: DOCENTE

	Muy malo (1)	Malo (2)	Normal (3)	Bueno (4)	Muy bueno (5)
Organización General				X	
Facilidad de navegación.		X			
Facilidad de uso.			X		
Simulador de compuertas lógicas.					X
Módulo fundamentos Circuitos lógicos.					X
Módulo sobre datos general compuertas lógicas.					X
Contenido teórico.				X	
Comentarios y sugerencias:					

Cuestionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Fabro González R
 Ocupación: Docente

	Muy malo (1)	Malo (2)	Normal (3)	Bueno (4)	Muy bueno (5)
Organización General				X	
Facilidad de navegación.				X	
Facilidad de uso.			X		
Simulador de compuertas lógicas.				X	
Módulo tutorial Circuitos lógicos.			X		
Módulo sobre datos general compuertas lógicas.				X	
Contenido teórico.			X		

Comentarios y sugerencias:

Mayor interacción entre usuario y aplicación.

Cuestionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Hellmuth Andres Stacey

Ocupación: Egresado Lic. Electrónica UPN

	Muy malo (1)	Malo (2)	Normal (3)	Bueno (4)	Muy bueno (5)
Organización General				X	
Facilidad de navegación.			X		
Facilidad de uso.				X	
Simulador de compuertas lógicas.				X	
Módulo fundamentación Circuitos lógicos.			X		
Módulo sobre datos general compuertas lógicas.				X	
Contenido teórico.			X		

Comentarios y sugerencias:

El módulo de datos generales sobre compuertas lógicas permite acceder a información importante en el momento de realizar el montaje de un circuito lógico digital, cuando no se tiene conexión a una red informática.

Cuestionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Arley Rodriguez

Ocupación: Estudiante Lic. Electronica

	Muy malo (1)	Malo (2)	Normal (3)	Bueno (4)	Muy bueno (5)
Organización General			X		
Facilidad de navegación.			X		
Facilidad de uso.				X	
Simulador de compuertas lógicas.					X
Módulo fundamentos Circuitos lógicos.				X	
Módulo sobre datos general compuertas lógicas.					X
Contenido teórico.				X	

Comentarios y sugerencias:

Agregar más Actividades Interactivas

Cuestionario

Cuestionario

Evaluación sobre la aplicación móvil sobre circuitos lógicos fundamentales.

Nombre: Carlos Roldan

Ocupación: Estudiante Lic. Diseño Tecnológico

	Muy malo (1)	Malo (2)	Normal (3)	Bueno (4)	Muy bueno (5)
Organización General			X		
Facilidad de navegación.				X	
Facilidad de uso.					X
Simulador de compuertas lógicas.				X	
Módulo fundamentos Circuitos lógicos.			X		
Módulo sobre datos general compuertas lógicas.					X
Contenido teórico.				X	

Comentarios y sugerencias:

Contiene una buena interfaz grafica.