

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



PROYECTO FIN DE CARRERA

MAPAS VEITCH-KARNAUGH EN iOS

David Agra Robles

JUNIO 2015

MAPAS VEITCH-KARNAUGH EN iOS

AUTOR: David Agra Robles
TUTOR: Federico García Salzmán
PONENTE: Eduardo Boemo Scalvinoni

Digital System Laboratory
Dpto. Tecnología Electrónica y de Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio 2015

Resumen

Este proyecto tiene como objetivo realizar una aplicación para dispositivos iOS. La cual permite la simplificación de funciones lógicas de 3 a 5 variables para de esta forma, ayudar a los alumnos que cursen la asignatura de Circuitos Electrónicos Digitales de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid (EPS UAM). La aplicación creada permite entender y asentar las bases de la simplificación de los mapas de Veitch-Karnaugh, además de ofrecer la posibilidad de realizar la comprobación del resultado de la gran parte de problemas propuestos en los primeros temas de la asignatura.

Se ha implementado la opción de simplificar mapas de hasta 5 variables a través del uso de las diferentes casillas mostradas en los mismos. Por otro lado, se han implementado formas alternativas para la realización de la introducción de las funciones como son: Tabla de verdad, sumatorio de minterminos, producto de maxiterminos, y introduciendo la expresión lógica por teclado, éste último personalizado para dicha función. Una vez elegida la forma de introducción de datos, es traducida al resto de modos y se produce la simplificación.

Adicionalmente se ha incluido una herramienta útil para el campo de trabajo de la aplicación, ésta es un conversor binario, octal, decimal y hexadecimal.

Para finalizar, esta aplicación se ha implementado soportando en dos idiomas, español e inglés. El lenguaje será seleccionado de forma automática según el idioma en el que se encuentre el dispositivo. Por defecto, se ha decidido que la aplicación se muestre en inglés en máquinas en las cuales el idioma configurado no sea ninguno de los disponibles.

La aplicación esta disponible para su descarga en App Store.

Palabras clave

Mapas Karnaugh, mapas Veitch-Karnaugh, iOS, dispositivo, simplificación lógica, conversor, binario, octal, decimal, hexadecimal, tabla de la verdad, sumatorio minterminos, producto maxiterminos, tutorial.

Abstract

This project aims to make an application para iOS devices, which allows the logical functions simplification of 3-5 variables using Karnaugh maps, to help students enrolled in the course of Digital Electronic Circuits of the Escuela Politécnica Superior at the UAM. The application created allows to understand the basics of simplification Veitch- Karnaugh maps. Besides offering the possibility to check the result of the most of problems proposed in the first part of the course.

It has been implemented the option to simplify maps up to 5 variables through the use of the different boxes shown in the same. Besides, if the user desire to use a greater number of variables functions, there were several different ways to facilitate work the same as: truth table, sum of minterms, maxterms product, and introducing the logical expression keyboard. After selecting the form of data input is translated to other modes and the simplification is performed.

Additionally, it has been included a useful tool that can help to the students. It is a binary, octal, decimal and hexadecimal converter.

Finally, the application supports two languages, Spanish and English. The language will be selected automatically according to the language in which the device is configured. By default, the application is displayed in English.

The application is available for downloading on App Store.

Key words

Karnaugh Maps, Veitch-Karnaugh Maps, device, iOS, logical functions simplification, converter, binary, octal, decimal, hexadecimal, truth table, sum of minterms, product of maxterms, tutorial.

Glosario

DSLlab:	Digital System Laboratory
iOS:	Mobile operating system developed by Apple Inc.
EPS:	Escuela Politécnica Superior
PFC:	Proyecto Final de Carrera
UAM:	Universidad Autónoma de Madrid
IDE:	Integrated Development Environment
GSM:	Global System for Mobile Communications
EDGE:	Enhanced Data for GSM Evolution
GPS:	Global Positioning System
SMS:	Short Message Service
API:	Application Programming Interface
CPU:	Central Processing Unit
VK:	Veitch-Karnaugh
SDK:	Software Development Kit
LIFO:	Last Input First Output
MVC:	Model-View-Controller

Agradecimientos

Llegados hasta este punto, quiero agradecer a mi familia que haya hecho posible que pueda haber estudiado esta carrera, ha sido un camino largo y para nada fácil, pero siempre han estado ahí para apoyarme tanto en lo bueno como en lo malo. En todo momento, les que querido transmitir mis logros como también mis decepciones, y hacerles partícipes de este largo camino pero que ya ha llegado a su fin y en mi opinión satisfactoriamente.

También quiero hacer mención especial a mi novia Rocío, junto a ella he empezado este trayecto y lo he terminado con ella, agradezco su actitud ya que siempre ha sido la de apoyarme, si llegaba un aprobado, era felicidad que me transmitía y si por el contrario no era así, no ha cesado en darme ánimos cada vez que éstos bajaban. Por ello, quiero que comparta conmigo esta nueva etapa que comienza ya que ha sido un pilar muy importante para mí. Sin olvidarme de su familia, de los que también he sentido su apoyo siempre.

Por otro lado, en el tiempo que he estado en la universidad, he conocido mucha gente, no con todo el mundo he tenido la misma relación pero en general puedo decir que estoy contento con la experiencia vivida. Agradecer a mi gran grupo de amigos: Javier, Pablo, Sergio, Guille, Jesús, Marcos, Luis y Agustín, con lo cuales he pasado mucho tiempo durante la carrera y no quisiera perder esa amistad con todos ellos. También recordar a la buena gente con la que he compartido clase y dar las gracias a Rubén Jiménez por su ayuda en la programación, ya que ha sido un verdadero maestro para mí.

A mi Eduardo Boemo, por darme la oportunidad de hacer realidad este proyecto y por su ayuda prestada.

ÍNDICE DE CONTENIDOS

CONTENIDO

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	2
1.3	Organización de la memoria.....	2
2	Estado del arte	3
2.1	Sistemas operativos móviles actuales.....	3
2.1.1	iOS.....	3
2.1.2	Otros sistemas operativos disponibles.....	9
2.2	Aplicaciones similares disponibles en el mercado	10
2.2.1	Advanced Karnaugh Map.....	11
2.2.2	Karnaugh Map Advanced.....	12
2.2.3	Karnaugh Map Optimizer.....	13
2.2.3	KMapper.....	13
2.2.6	Tabla comparativa realizada.....	15
3	Diseño.....	16
3.1	Finalidad del proyecto	16
3.2	Especificación de requisitos	17
3.2.1	Sistema operativo	17
3.2.1	Versión de iOS elegida.....	18
3.2.2	Idioma.....	19
3.2.3	Dispositivo compatibles	20
3.2.4	Resumen	21
3.3	Herramientas disponibles en la aplicación	21
3.3.1	Tutorial	21
3.3.2	Herramienta de selección del conjunto de variables	22
3.3.3	Herramienta Karn Auto	23
3.3.4	Herramienta Traductor	25
3.3.5	Herramienta Conversor.....	27
3.3.6	Herramienta de ayuda.....	28
4	Desarrollo	29
4.1	Primeros pasos.....	29
4.2	Instalación del entorno de desarrollo.....	30

4.3 Esquema de la aplicación Karn Map	31
4.4 Conceptos clave	32
4.4.1 Recursos utilizados en una aplicación	32
4.4.2 Estructura de una aplicación de iOS	32
4.4.3 Ciclo de vida de la clase UIViewController	34
4.4.4 Objetos View	35
4.4.5 Objetos UIAlertView	35
4.4.6 Clases ya comentadas anteriormente	35
4.4.7 Archivo de manifiesto	36
4.5 Detalles de la implementación	36
4.5.1 Tutorial	36
4.5.2 Herramienta de selección del conjunto de variables	37
4.5.3 Herramienta Karn Auto	41
4.5.4 Herramientas para introducir funciones en el Traductor	44
4.5.4 Mostrando resultados en la herramienta Traductor	49
4.5.5 Problemas encontrados en el desarrollo de la herramienta “Traductor”	51
4.5.6 Herramienta conversor	51
4.5.7 Manuales de ayuda	53
4.5.8 Página de información sobre la aplicación	54
4.6 Algoritmo Quine-McCluskey	55
4.6.1 Introducción	55
4.6.2 Funcionamiento	56
4.6.3 Desarrollo	60
4.7 Diagrama de clases	61
4.7.1 Clases principales	62
5 Integración, pruebas y resultados	64
5.1 Pruebas	64
5.1.1 Pruebas en herramienta Karn Auto	64
5.1.2 Pruebas en herramienta Traductor	66
5.1.3 Pruebas en herramienta Conversor	66
5.1.4 Pruebas realizadas en otros dispositivos	66
5.2 Resultados	66
5.3 Primera publicación	66
5.3.1 Proceso de publicación: Paso 1	67
5.3.2 Proceso de publicación: Paso 2	68
5.3.3 Proceso de publicación: Paso 3	68
5.3.4 Proceso de publicación: Paso 4	69

5.3.5 Proceso de publicación: Paso 5	69
5.3.6 Proceso de publicación: Paso 6	70
5.3.7 Proceso de publicación: Paso 7	70
5.3.8 Proceso de publicación: Paso 8	72
5.3.9 Proceso de publicación: Paso 9	72
5.3.10 Proceso de publicación: Paso 10	73
5.3.11 Proceso de publicación: Paso 11	74
6 Conclusiones y trabajo futuro	75
6.1 Conclusiones.....	75
6.2 Trabajo futuro	76
MANUAL DEL PROGRAMADOR.....	1
A - Apéndice A.....	1
A.1 Comprobación de la expresión	1
A.2 Necesidad de conversión infijo-postfijo	3
A.3 Algoritmo de conversión infijo-postfijo	3
A.4 Evaluación de expresiones postfijo	5
A.5 Traduciendo la expresión lógica.....	6
Referencias	7
ANEXOS	8
A. PRESUPUESTO.....	8
B. PLIEGO DE CONDICIONES	9

ÍNDICE DE FIGURAS

Figura 2-1: Arquitectura de capas iOS	6
Figura 2-2: Distribución de versiones iOS	8
Figura 2-3: Distribución del mercado entre sistemas operativos	10
Figura 2-4: Advanced Karnaugh Map	11
Figura 2-5: Karnaugh Map Advanced	12
Figura 2-6: Karnaugh Map Optimizer	13
Figura 2-7: KMapper mapas	14
Figura 2-8: KMapper tabla	14
Figura 3-1: Distribución de versiones iOS	19
Figura 4-1: Esquema general de la aplicación	31
Figura 4-2: Diagrama modelo MVC	32
Figura 4-3: Diagrama modelo MVC	34
Figura 4-4: Menú conjunto de variables	37
Figura 4-5: Celda de la lista conjunto de variables	38
Figura 4-6: Menú conjunto de variables por defecto	38
Figura 4-7: Selección del conjunto de variables	39
Figura 4-8: Modificar conjunto de variables	39
Figura 4-9: Añadir nuevo conjunto de variables	40
Figura 4-10: Borrar un conjunto de variables	41
Figura 4-11: Funcionamiento Karn Auto 4 variables	43
Figura 4-12: Botones Karn Auto	43
Figura 4-13: Visualización de agrupaciones	44
Figura 4-14: Traductor por expresión algebraica	45
Figura 4-15: Teclado para introducir expresión algebraica	46
Figura 4-16: Introducción por tablas	47
Figura 4-17: Introducción valores por tablas	48
Figura 4-18: Menú de resultados de la traducción	49
Figura 4-19: Resultados de las traducciones realizadas	50
Figura 4-20: Herramienta Conversor	52
Figura 4-21: Menús de ayuda (I)	53
Figura 4-22: Menús de ayuda (II)	54
Figura 4-23: Información de la aplicación	55
Figura 4-24: Comprobación con herramienta Karn Auto	60
Figura 4-25: Diagrama de clases	61
Figura 5-1: Caso extremo (I)	65
Figura 5-2: Caso extremo (II)	65
Figura 5-3: Caso extremo (III)	65
Figura 5-4: Proceso de publicación: Paso 1	67
Figura 5-5: Proceso de publicación: Paso 2	68
Figura 5-6: Proceso de publicación: Paso 3	68
Figura 5-7: Proceso de publicación: Paso 4	69
Figura 5-8: Proceso de publicación: Paso 5	69
Figura 5-9: Proceso de publicación: Paso 7 (I)	70
Figura 5-10: Proceso de publicación: Paso 7 (II)	71
Figura 5-11: Proceso de publicación: Paso 8	72
Figura 5-12: Proceso de publicación: Paso 9	72
Figura 5-13: Proceso de publicación: Paso 10	73
Figura 5-14: Proceso de publicación: Paso 11	74

ÍNDICE DE TABLAS

Tabla 2-1: Versiones de iOS.....	7
Tabla 2-2: Comparativa de aplicaciones	15
Tabla 3-1: Símbolos y prioridades disponibles para expresión lógica	26
Tabla 4-1: Tabla de verdad de la función de ejemplo	57
Tabla 4-2: Primera tabla de primos implicantes.....	57
Tabla 4-3: Segunda tabla de primos implicantes.....	58
Tabla 4-4: Tabla para buscar los primos implicantes esenciales.....	59
Tabla 0-1: Errores notificados por la función de comprobación de la expresión lógica..	2
Tabla 0-2: Ejemplos de expresiones usando notación infija y postfija	3
Tabla 0-3: Prioridades en la conversión infijo-postfijo	3
Tabla 0-4: Ejemplo de funcionamiento del algoritmo de conversión infijo-postfijo	4
Tabla 0-5: Ejemplo de ejecución del algoritmo de evaluación postfijo	5

1 Introducción

1.1 Motivación

La implementación de funciones digitales mediante puertas requiere utilizar las mismas, de la forma más simplificada posible. La herramienta más sencilla para ello, son los mapas de Veitch-Karnaugh. Estos permiten la simplificación mediante representación gráfica en forma de tablas de la función, facilitando este proceso. Según vaya aumentando el número de variables de la función, dificulta la simplificación mediante los mapas en cuestión.

Dicho problema, se quiere evitar mediante la simplificación mencionada anteriormente de cara a poder trabajar con funciones con gran número de variables. Durante el curso, los estudiantes deberán hacer frente a diversos problemas de este tipo, por lo que el uso de esta herramienta permitirá el mecanizar esta acción y evitar dibujar mapas de gran tamaño y buscar gráficamente la función simplificada. De esta forma, se acortara los tiempos de resolución.

En la actualidad, el uso de los dispositivos móviles y tabletas ha crecido de forma exponencial extendiéndose en todos los campos incluso en la educación tanto de bajo como de alto nivel. Por ello, se ha decidido explotar dichas plataformas desarrollándose las herramientas para éstas y así poder ayudar de forma accesible y sencilla a los estudiantes.

El uso de la herramienta implementada en estos dispositivos proporciona ventajas como estas:

- Fácil portabilidad. La herramienta es accesible en todo momento, ya que la tenemos en el bolsillo.
- Sustitución del papel y bolígrafo por pantalla y dedo. Uso de la pantalla táctil, función implementada en la mayor parte de los dispositivos.
- Ahorro de tiempo.
- Actualización automática proporcionada por Apple Store.
- No es necesario estar conectado a internet para poder hacer uso de ella.
- Extender su uso a estudiantes de ingeniería de todos los lugares del mundo.

1.2 Objetivos

El propósito de este trabajo es desarrollar una aplicación iOS que proporcione ciertas herramientas al usuario para apresurar la simplificación y traducción de funciones lógicas.

La aplicación creada "Karn Map" será la primera desarrollada por el DSLab que se encuentra disponible en el Apple Store, ayudará a asentar los fundamentos de los circuitos digitales.

Para la descarga de esta aplicación, los estudiantes la tendrán disponible en Apple Store. Además, las guías de problemas de la asignatura poseen ejercicios, los cuales pueden ser resueltos haciendo uso de la aplicación.

El fin del proyecto de fin de carrera es de un ejercicio de carácter personal donde se integran los conocimientos y destrezas adquiridas a lo largo del itinerario formativo, en este caso el objetivo del estudiante del PFC está enfocado a aprender programar aplicaciones para dispositivos iOS, además del lenguaje Objective-C junto a la revisión completa de circuitos digitales. Estos dos contenidos no han sido cubiertos por las asignaturas obligatorias de la Ingeniería de Telecomunicaciones, por lo tanto se considera una formación adicional y enriquecedora dentro de este campo.

1.3 Organización de la memoria

Para facilitar la organización y comprensión de esta memoria, ésta se encuentra dividida en los siguientes capítulos:

- Capítulo 1: Introducción, motivación, objetivos del proyecto y organización de la memoria.
- Capítulo 2: Estado del arte, sistemas operativos móviles y estudio de aplicaciones en el mercado.
- Capítulo 3: Diseño de la aplicación, finalidad, elección de requisitos y herramientas de la aplicación.
- Capítulo 4: Desarrollo de la aplicación, pasos previos, instalación del entorno de desarrollo, esquema general, conceptos clave, detalles de la implementación, algoritmo Quine-McCluskey y diagrama de clases.
- Capítulo 5: Integración, pruebas, resultados, primera publicación, estadísticas y cuestionario.
- Capítulo 6: Conclusiones y trabajo futuro.

2 Estado del arte

2.1 Sistemas operativos móviles actuales

En esta sección, se introducirán de forma breve los diferentes sistemas operativos que actualmente coexisten. Dando un mayor detalle al sistema utilizado por Apple es sus dispositivos móviles y tabletas, éste recibe el nombre de iOS.

2.1.1 iOS

2.1.1.1 Historia

La compañía Apple anunció la existencia de iPhone OS a principios de 2007, aunque no recibió un nombre oficial hasta que fue lanzada la primera versión beta del entorno de desarrollo “Iphone SDK” un año más tarde. En los comienzos de dicho sistema, se consideraba que el dispositivo iPhone ejecutaba OS X (sistema operativo usado en ordenadores de la marca) o una versión modificada NewtonOS.

Tal y como se ha mencionado anteriormente, recibiría el nombre de iPhone OS, siendo el lanzamiento del mismo el 29 de junio de 2007.

El interés por éste nuevo sistema, crecería acorde a la gran expansión que tuvo la plataforma iPhone , que se vio incrementando por el lanzamiento del iPod Touch también en 2007. Un dispositivo con las capacidades multimedia del iPhone pero sin la capacidad de realizar llamadas telefónicas.

En 2010, Steve Jobs, CEO de Apple, dio a conocer un nuevo dispositivo, el iPad. Éste compartía gran parte de funcionalidades con el iPod Touch, esta vez orientado hacia la industria de contenidos. El iPad poseía un pantalla de mayor tamaño, pero que a su vez, haría uso del sistema operativo de los dos anteriores dispositivos lanzados por la compañía. Adicionalmente, incluía una aplicación oficial destinada a la compra y lectura de iBooks, libros electrónicos.

El número de aplicación disponibles en la plataforma iPhone OS a través de App Store, creció de forma exponencial. Siendo en 2010 con la presentación del iPhone 4 cuando de nuevo Steve Jobs confirmó que iPhone OS cambiaría su nombre para ser iOS.

En 2012, se anuncia iOS 6 de la mano de Tim Cook en la presentación del iPhone 6. Un año más tarde, fue presentado iOS 7, siendo el mayor cambio del sistema desde el primer iPhone, según la compañía. Dicha versión, recibió grandes modificaciones respecto al entorno gráfico y la inclusión de importante nuevas funciones.

Finalmente, iOS 8 la actual versión del sistema operativo, siendo compatible con estos dispositivos de iPhone: 4s, 5, 5c, 5s, 6, 6 Plus, la quinta generación del iPod Touch y el iPad 2 en adelante. Se espera que en el próximo mes de septiembre sea lanzado iOS 9.

2.1.1.2 Características

A continuación, se va a detallar ciertas características destacadas que han hecho al sistema operativo iOS, ser un referente en el mercado del software para dispositivos:

- **Pantalla principal**

En iOS la pantalla principal recibe el nombre de SpringBoard, en la cual se encuentran los iconos de las aplicaciones y el Dock, que es zona inferior de la pantalla donde se puede anclar aplicaciones que el usuario usa frecuentemente. Éste se muestra al desbloquear el dispositivo o bien, al presionar el botón de inicio. En la zona superior de la pantalla se muestra una barra de estado que ofrece al usuario datos como son la hora, nivel de batería y la intensidad de señal. Una vez se ejecuta una aplicación, el resto de pantalla es dedicado a la misma.

- **Carpeta**

Este elemento ya conocido en los sistemas operativos de escritorio, se introdujo en la versión iOS 4. De esta forma, podía moverse una aplicación sobre otra y crear una carpeta, pudiéndose agregar de nuevo más aplicaciones con el mismo proceso. El título de la carpeta es elegido automáticamente por el tipo de aplicaciones contenidas, aunque puede ser modificado por el usuario. Más tarde, con la lanzamiento de iOS 7, la cantidad máxima de aplicaciones por carpeta aumenta, ya que se crean más paginas dentro de la misma, deslizando sobre la pantalla accedemos a cada una de ellas.

- **Seguridad**

Previamente al lanzamiento de iOS 7, existía un número índice de robos de los diferentes modelos del iPhone. Por ello, se creó la activación por iCloud, la cual solicita los datos de acceso de la cuenta del usuario original, permitiendo bloquear el dispositivo si es perdido o bien robado. Adicionalmente, es posible conocer la ubicación vía GPS del dispositivo y mostrar mensajes en pantalla.

- **Centro de notificaciones**

Con el lanzamiento de iOS 5, se realizaron numerosos cambios en el sistema de notificaciones siendo rediseñado por completo. Las notificaciones se sitúan en un área por la cual se puede acceder a la misma mediante un deslize de la barra de estado hacia abajo. En versiones anteriores del sistema operativo, las notificaciones aparecían como mensajes en la pantalla.

- **Multitarea**

En versiones anteriores a iOS 4, la función multitarea estaba reservada para aplicaciones por defecto del sistema. De esta forma, se aseguraban problemas de batería y rendimiento si se ejecutaban aplicaciones de terceros al mismo tiempo. Para iniciar la multitarea hay que apretar 2 veces el botón del centro. La multitarea sólo es compatible desde el iPhone 3GS, iPad 1, iPod Touch (3.^a generación).

Más tarde a partir del iOS 4 dispositivos de tercera generación y posteriores permiten el uso de 7 APIs para multitarea, específicamente:

- Audio en segundo plano
- Voz IP
- Localización en segundo plano
- Notificaciones push
- Notificaciones locales
- Completado de tareas
- Cambio rápido de aplicaciones

- **Game Center**

Esta función fue presentada en 2010 junto a iOS 4. Ésta es una red social enfocada a juegos de la plataforma, permite agregar una foto de tu perfil, pudiendo ver a los amigos de tus amigos y pudiendo encontrar adversarios de tus juegos.

- **Tecnologías no admitidas**

El sistema operativo no permite Adobe Flash ni Java. Steve Jobs redactó una carta abierta en la cual, criticaba a Flash por su inseguridad, errores, excesivo consumo de batería y otras incompatibilidad con iOS. Por ello, Apple en su plataforma hace uso de HTML5 como alternativa.

2.1.1.3 Arquitectura

La arquitectura iOS está diseñada en capas, siendo las capas más altas la que contienen los servicios y tecnologías de mayor importancia para el desarrollo de aplicaciones, por otro lado las capas más bajas controlan los servicios básicos del sistema.

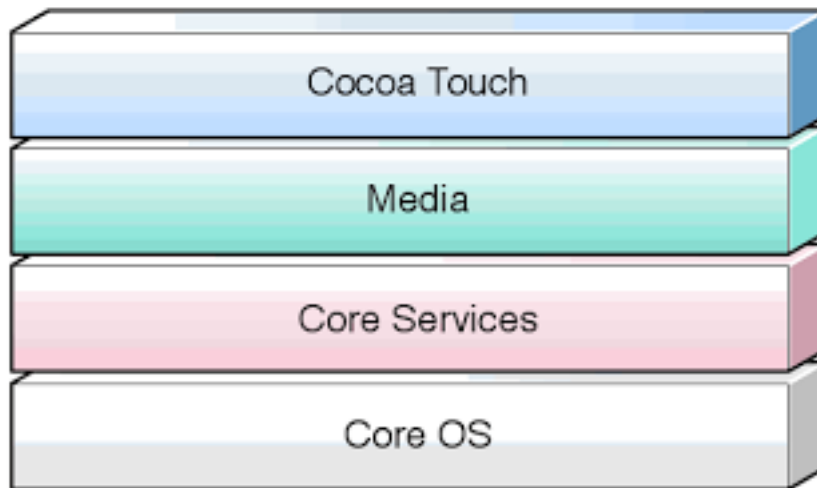


Figura 2-1: Arquitectura de capas iOS

* Imagen original de <https://sites.google.com/site/tecnologiaiosm/desarrollo-de-aplicaciones/arquitectura-ios>

- **Cocoa Touch**

Cocoa Touch es la capa que cobra mayor importancia para el desarrollo de aplicaciones iOS. Contiene un conjunto de Frameworks que proporciona el API de Cocoa para desarrollar aplicaciones.

Se podría decir que Cocoa Touch proviene de Cocoa, la API ya existente en la plataforma MAC.

Dicha capa, está formada por dos Frameworks fundamentales:

- **UIKit:** contiene todas las clases que se necesitan para el desarrollo de una interfaz de usuario.
- **Foundation Framework:** define las clases básicas, acceso y manejo de objetos, servicios del sistema operativo.

- **Media**

Proporciona los servicios de gráficos y multimedia a la capa superior.

- **Core Services**

Contiene los servicios fundamentales del sistema de los que hacen uso todas las aplicaciones.

- **Core OS**

Contiene las características de bajo nivel: ficheros del sistema, manejo de memoria, seguridad, drivers del dispositivo.

2.1.1.4 Versiones de iOS

El sistema operativo iOS, fue lanzado en 2007 a partir de esa fecha, no se ha detenido el lanzamiento de actualizaciones y nuevas versiones del sistema. En ellas, se ha mejorado fallos descubierto, mejoras de autonomía, rediseño de elementos y inclusión de nuevas funcionalidades que le convierten en uno de los sistemas más estables actualmente. A continuación en la siguiente tabla se muestra la evolución de iOS a través del lanzamiento de nuevas versiones:

Version	Lanzamiento
iPhone OS 1	Junio de 2007
iPhone OS 2	Junio de 2008
iPhone OS 3	Junio de 2009
iOS 4	Junio de 2010
iOS 5	Octubre de 2011
iOS 6	Septiembre de 2012
iOS 7	Junio de 2013
iOS 8	Septiembre de 2014

Tabla 2-1: Versiones de iOS

Tal y como se indicó anteriormente, Apple renombró su sistema operativo en 2010 con el nombre por el cual conocemos a su sistema para dispositivo móviles, iOS. Se espera que en los próximos meses sea lanzada la novena versión del sistema. No obstante, cabe destacar que con la salida de estas nuevas actualizaciones se presentan nuevos dispositivos y se van descatalogando cierto modelos anteriores haciéndolos no compatibles, siendo el motivo garantizar un buen funcionamiento de todos los dispositivos lanzados por la compañía. A diferencia de otros sistemas, una vez actualizado no se permite la reinstalación de versiones anteriores, de esta forma Apple evita la fragmentación de su sistema a diferencia de su gran rival Android.

2.1.1.5 Distribución y uso de iOS en el mercado

A continuación en el siguiente gráfico se puede observar que la gran parte de usuarios de iOS se encuentran entre en la última versión lanzada. No obstante, una pequeña minoría aún hace uso de la versión anterior iOS 7, el motivo de esto puede ser que poseen dispositivo no compatibles con la última versión. El porcentaje restante, un 3%, lo ocupan los dispositivos que tienen otras versiones, principalmente terminales antiguos que los usuarios siguen utilizando para conectarse a la App Store. Tal y como se ha comentado en el apartado anterior, Apple no presenta fragmentación tan acusada como puede haber en otro sistemas operativos.

**77% of devices are using
iOS 8.**



As measured by the App Store
on March 16, 2015.

Figura 2-2: Distribución de versiones iOS

* Imagen original de <http://www.soyapps.com/ios-8-77-distribucion/>

2.1.2 Otros sistemas operativos disponibles

2.1.2.1 Android

El sistema operativo Android es el gran rival de iOS actualmente. Éste ha sido desarrollado por Google y es de código abierto, se encuentra instalado en un gran número de dispositivos de múltiples marcas. La gran ventaja de este sistema, es que al ser un sistema abierto, tiene vía a libre a todo tipo de modificaciones realizadas por las propias marcas que usan este sistema o bien por los propios usuarios. Este grado de personalización esta ausente en el sistema iOS. No obstante, esta cuestión tiene sus pros y sus contras que tanto hablan gran número de usuarios, ya que al ser un software libre no está libre de posibles errores o bien agujeros de seguridad que no pueden ser controlado por Google, como si puede hacer Apple en su sistema cerrado. Además cabe destacar, que el mercado de Apps de Android no posee tanta cantidad de ingresos como si puede decir iOS.

2.1.2.2 Windows Phone

Sistema operativo lanzado en 2010 por la compañía Microsoft, a diferencia de Android y iOS su cuota de mercado es menor. No obstante, se ha ido actualizando para plantar cara al resto de competidores. Adicionalmente, Microsoft desea alcanzar la convergencia de su sistema en todas las plataformas en la cuales está presente, siendo una característica interesante para numerosos usuario e incluso empresas.

2.1.2.3 Blackberry OS

Se trata de un sistema operativo móvil cerrado desarrollado por Blackberry, tuvo su auge años atrás sobre todo en el mundo empresarial. Actualmente la empresa se encuentra en una situación financiera complicada.

2.1.2.4 Symbian OS

Symbian es un sistema operativo propiedad de Nokia, y que en el pasado fue producto de la alianza de varias empresas de telefonía móvil. El objetivo de este sistema era poder competir con el resto de sistemas, aunque finalmente no fue así con la llegada de los smartphones y actualmente se encuentra extinto.

2.1.2 Distribución y uso de estos sistemas operativos

Con el fin de poder observar gráficamente reparto de mercado entre los diferentes sistemas vistos y como ha ido evolucionando a lo largo de los últimos años hasta la actualidad, se adjunta la siguiente gráfica:

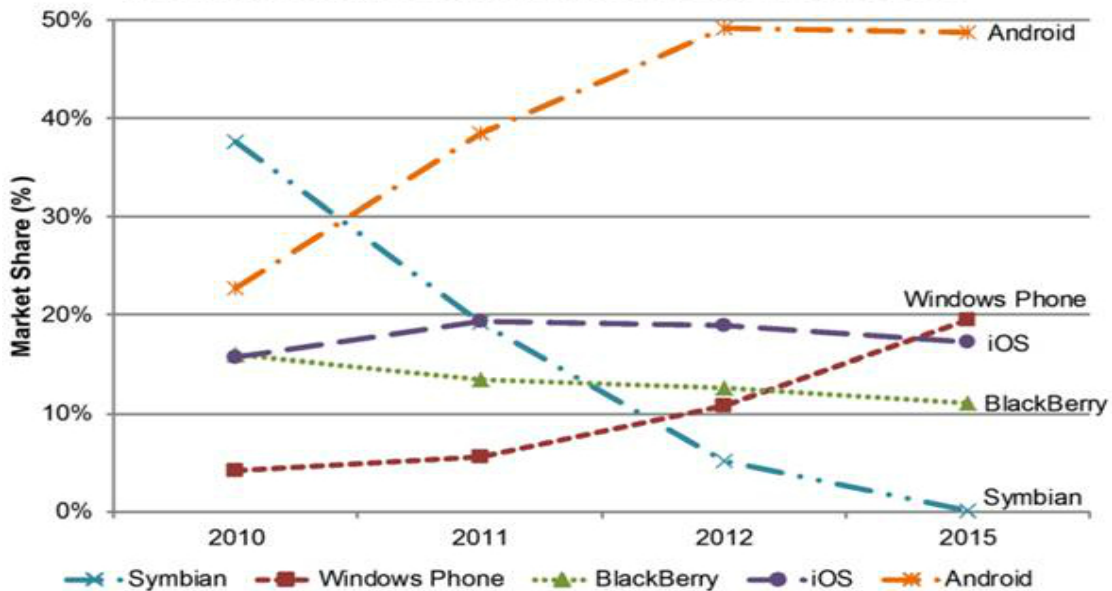


Figura 2-3: Distribución del mercado entre sistemas operativos

* Imagen original de <http://mhealth.jmir.org/article/viewFile/2733/1/30587>

2.2 Aplicaciones similares disponibles en el mercado

En esta sección se ha decidido realizar un estudio de las aplicaciones ya disponibles en el mercado. Como criterio de búsqueda se ha establecido, que ofrezcan funciones similares a las ofrecidas por la aplicación desarrollada en este proyecto y las cuales formarían parte de la competencia. La plataforma elegida para el desarrollo del proyecto es iOS; por ello, solo se realizará la comparativa con aplicaciones disponibles en el App Store.

Respecto a la publicación de la aplicación del proyecto, será subida a la plataforma indicada previamente, con un coste mínimo de 0,99 €. Por ello, se evaluarán aplicaciones de coste cero o del mismo precio. Los ingresos generados serán destinados a la fundación del laboratorio DSLab.

Las siguientes aplicaciones son las seleccionadas siguiendo el criterio establecido:

2.2.1 Advanced Karnaugh Map

Esta primera aplicación es gratuita, pero posee mini-compras integradas y publicidad. Por lo que, para hacer un uso completo de la misma y eliminar anuncios se deben comprar mejoras. Al ejecutarla por primera vez se muestra un mapa de Karnaugh de 4 variables tal y como se muestra en la siguiente imagen:

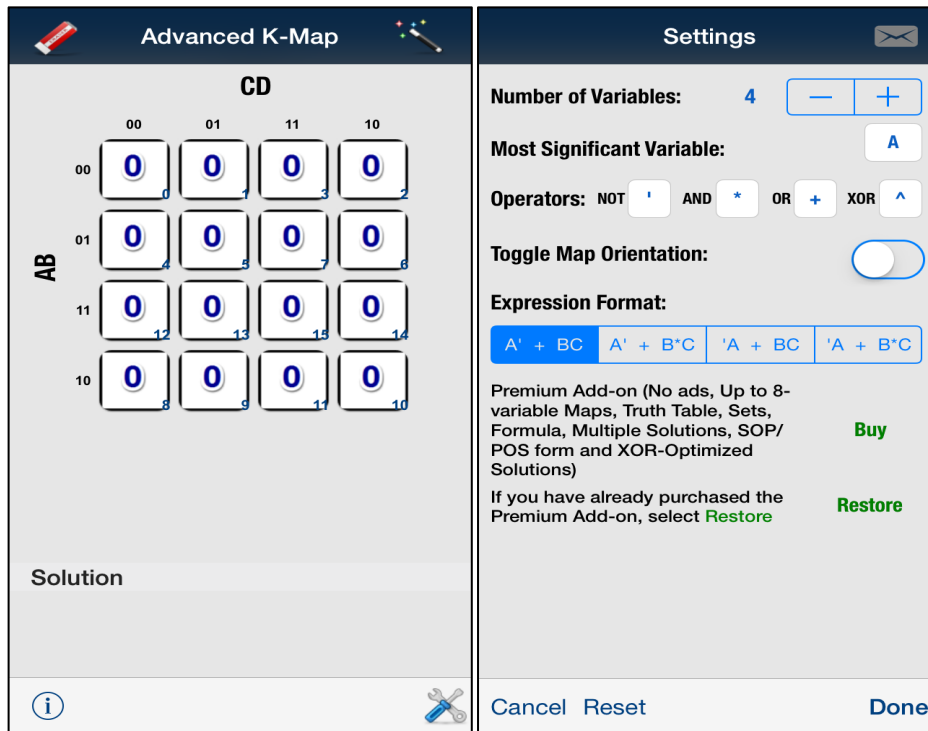


Figura 2-4: Advanced Karnaugh Map

El usuario puede modificar el valor de cada botón, siendo disponibles el '0', '1' o 'X'. Una vez elegidos los mismos, debemos pulsar sobre el botón que es encuentra en la zona superior derecha y nos calculara la solución optimizada. Al tratarse de una versión gratuita no Premium, únicamente se puede trabajar con mapas de 2 a 4 variables y sin poder cambiar la letra asignada para cada variable.

En esta versión gratuita tampoco es posible observar como se realizan las agrupaciones en la optimización de la función. Si se desea más funciones como pueden ser un mayor número de variables, o bien el uso de tablas de verdad, se debe realizar el pago por la versión Premium.

2.2.2 Karnaugh Map Advanced

La segunda aplicación elegida es Karnaugh Map Advanced, tiene un coste de 0,99 € y no posee publicidad. Ésta permite la simplificación de funciones algebraicas mediante el uso de mapas de Karnaugh y tablas de verdad de hasta 8 variables. No obstante, no es posible elegir el nombre de las variables, aunque si esta soportado el valor 'X' en la introducción datos para elaborar la función algebraica. Adicionalmente al hacer uso de mapas de Karnaugh, se muestra la simplificación de forma gráfica. A continuación, se muestran capturas de la misma:



Figura 2-5: Karnaugh Map Advanced

Cabe destacar la ausencia de un tutorial para familiarizar al usuario con la misma. Podría ser una de las completas, sin embargo no posee ningún tipo de conversor como si dispone nuestra aplicación Karn Map.

2.2.3 Karnaugh Map Optimizer

La siguiente en ser analizada, se puede decir que posee un diseño minimalista y al igual que la anterior aplicación carece de una manual o tutorial de la misma. Como se puede observar en la siguientes imágenes:

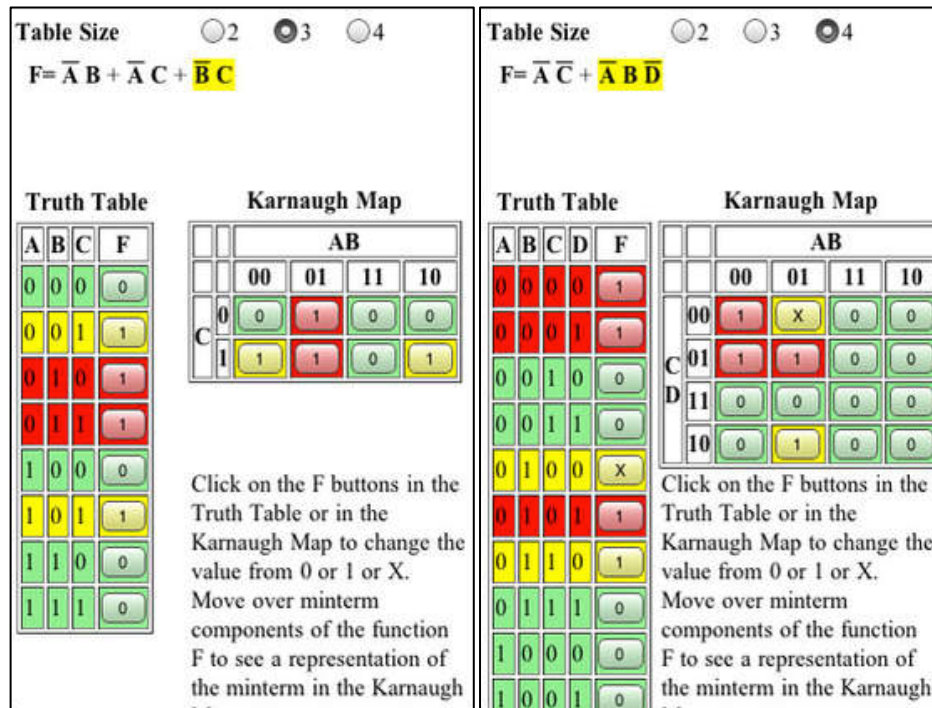


Figura 2-6: Karnaugh Map Optimizer

Su coste es de 0,99€ libre de publicidad, a diferencia de la anterior aplicación, ésta soporta un menor número de variables, un máximo de 4 pero sigue soportando el uso del valor 'X'. La simplificación de funciones algebraicas se realiza a través de mapas de Karnaugh y tablas de verdad, siendo mostrada gráficamente la simplificación efectuada. No obstante, las variables son predefinidas y no pueden ser modificadas. Además de no disponer de un tutorial ni herramienta Conversor.

2.2.3 KMapper

Ésta última aplicación no tiene coste pero si dispone de publicidad, se puede decir que es la más sencilla en cuanto a funciones, ya que únicamente permite trabajar la optimización con mapas de Karnaugh de 2 y 3 variables siendo la interfaz sencilla, tal y como se puede observar en la siguiente imagen:

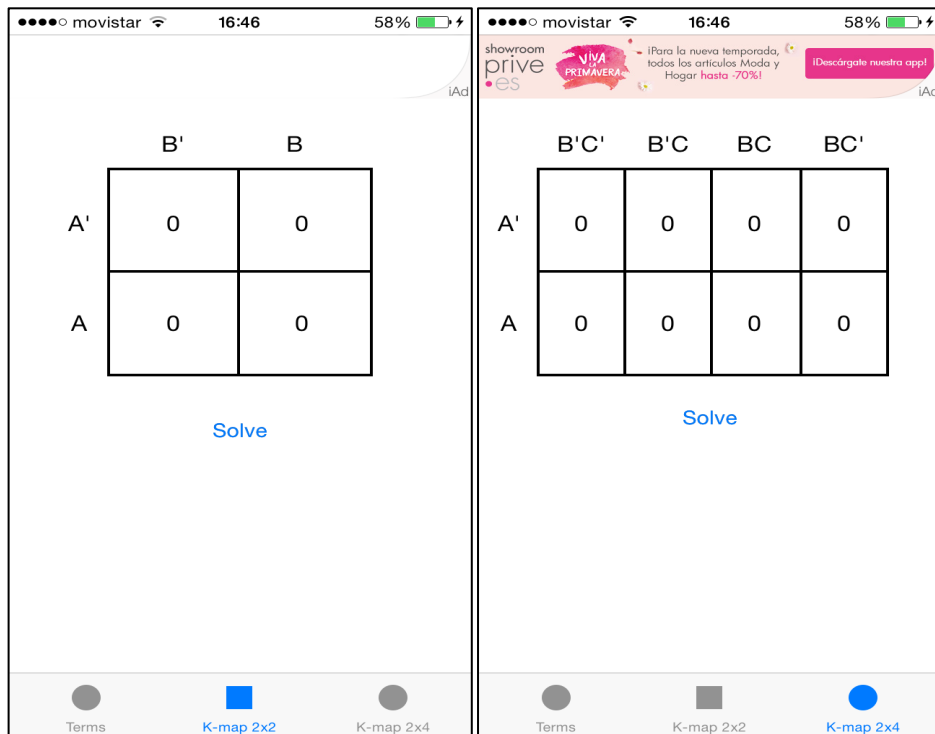


Figura 2-7: KMapper mapas

A diferencia de las aplicaciones analizadas anteriormente, el usuario puede introducir minitérminos y maxitérminos en tablas de 3 variables, se puede ver lo mencionado en la siguiente captura:

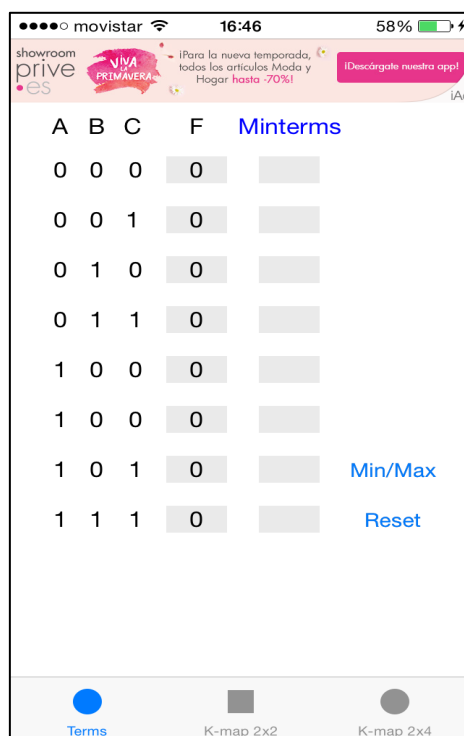


Figura 2-8: KMapper tabla

No obstante, se agradecería la inclusión de un tutorial para que el usuario pudiera entender mejor las funciones ofrecidas por dicha aplicación.

2.2.6 Tabla comparativa realizada






Nombre	Advanced Karnaugh Map	Karnaugh Map Advanced	Karnaugh Map Optimizer	KMapper	Karn Map
Logo					
Número de variables	2-4	2-8	2-4	2-3	3-5
Modificar nombres variables	No	No	No	No	Si
Tiene publicidad	Si	No	No	Si	No
Estética	Simple y sencilla	Simple y cuidada	Interfaz compleja	Simple y sencilla	Intuitiva y cuidada
Métodos de introducir función (núm. de variables)					
Mapa Karnaugh	Si (2-4)	Si (2-8)	Si (2-5)	Si(2-3)	Si (3-5)
Tabla de la verdad	No	Si (2-8)	Si	No	Si (3-5)
Expresión lógica	No	Si	No	No	Si (3-5)
Métodos de salida					
Tabla de verdad	No	Si	Si	No	Si
Expres. simplificada	Si	Si	Si	Si	Si
Sumatorio minitérminos	No	No	No	Si	Si
Producto maxitérminos	No	No	No	Si	Si
Opciones de simplificación					
Admite "don't cares"	Si	Si	Si	No	Si
Simplifica correctamente	Si	Si	Si	Si	Si
Muestra las agrupaciones	No	Si	Si	No	Si
Ofrece más de una simpl. correcta	No	No	No	No	No
Enseña los pasos de Quine-McCluskey	No	No	No	No	No
Precio	Gratis*	0,99 €	0,99 €	Gratis	0,99 €

Tabla 2-2: Comparativa de aplicaciones

(*) Si se desea adquirir el formato Premium de la aplicación con más funciones se debe pagar 2,99 €.

3 Diseño

En este apartado se ha destinado a explicar los objetivos a alcanzar durante el desarrollo de este proyecto siempre teniendo en cuenta los requeridos establecidos y la finalidad del mismo.

En primer lugar se hará una breve introducción de las herramientas disponibles en la aplicación junto a sus limitaciones y algunas clases implementadas para el funcionamiento de las mismas.

3.1 Finalidad del proyecto

El objetivo de este proyecto es desarrollar una aplicación de iOS, cuya función principal sea la de simplificar funciones lógicas haciendo uso del mapa de Veitch-Karnaugh. Con el fin de poder ser una herramienta de fácil uso para los alumnos de la nueva asignatura de grado de Circuitos Digitales de la Escuela Politécnica Superior. Ésta permitirá ayudar a sentar las bases de ésta asignatura.

En este proyecto el objetivo es realizar una aplicación en iOS que principalmente simplifique funciones lógicas con el mapa de Veitch-Karnaugh. De manera que pueda ser, una aplicación útil para los alumnos de la nueva asignatura de grado de Circuitos Electrónicos Digitales de la Escuela Politécnica Superior.

Entre las diferentes funciones que realiza la aplicación, proporcionará:

- Mostrar gráficamente las agrupaciones de los términos del mapa de tal forma que se optimice el número de conjuntos.
- Ofrecer la respuesta correcta y servir de comprobante para que el estudiante verifique su resultado obtenido.
- Facilitar y agilizar la resolución de problemas o prácticas de laboratorio, gracias a la inclusión de herramientas traductoras y conversoras.

La aplicación implementada proporciona 4 tipo de herramientas principales, a lo que debemos añadir un tutorial sobre la simplificación de funciones lógicas y un manual de ayuda. Siendo las siguientes que se indican a continuación:

- **Herramienta de selección de variables:** Permite seleccionar el número de variables y sus nombres.
- **Herramienta Karn Auto:** Proporciona la simplificación en tiempo real de funciones lógicas en mapa de Karnaugh. Adicionalmente, se ofrece la opción de mostrar como se realizan las agrupaciones durante la simplificación.

- **Herramienta Traductor:** Su función es la de traducir y simplificar funciones lógicas desde varios métodos de entrada hasta varios de salida.
- **Herramienta Conversor:** Permite la conversión entre los diferentes sistemas numéricos binario, octal, decimal y hexadecimal.

Más adelante se explicará de forma más extensa, las funciones principales que ofrece la aplicación.

Una vez analizadas las aplicaciones que se encuentran disponibles en el mercado, se quiere implementar las características más interesantes del conjunto analizado. Siendo el objetivo principal que la aplicación desarrollada posea un interfaz sencilla e intuitiva además de disponer de un manual de ayuda para el usuario, el cual no esta disponible en las aplicaciones evaluadas previamente.

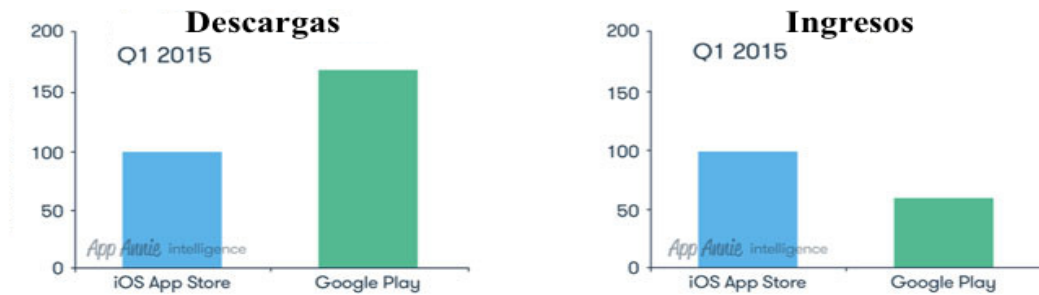
3.2 Especificación de requisitos

En este apartado se va a explicar los motivos por los cuales se han tomado ciertas decisiones durante el desarrollo del proyecto. Comenzando por la elección del sistema operativo finalizando por los dispositivos compatibles.

3.2.1 Sistema operativo

En la sección del estado del arte se realizó una breve introducción de la historia y características de los diferentes sistemas operativos existentes. La aplicación desarrollada en este proyecto tiene su versión Android también en propiedad, debido al número de descargar conseguidas, desde el laboratorio DSLab de la Escuela Politécnica de Madrid de la Autónoma de Madrid, se decidió el lanzamiento de esta aplicación pero para la plataforma iOS. De esta forma, poder estar disponible para los estudiantes que hacen uso del sistema operativo móvil de la compañía Apple. A continuación se exponen otro motivos por el cual se decidió la implementación de la aplicación:

- A pesar del mayor coste de la licencia de desarrollador, en App Store es de 99\$ frente los 25\$ del Play Store, se ha analizado que los usuarios de iOS son mas propensos al pago por aplicaciones; lo que puede amortizar el gasto inicial por la licencia con las posibles ventas obtenidas. Las ventas de aplicaciones en ambas plataformas se puede observar en la siguiente imagen con los resultados de a principios de 2015:



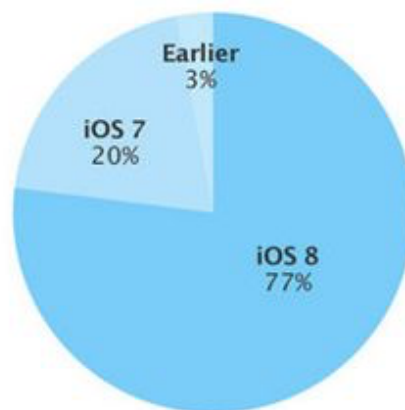
Todos los ingresos obtenidos irán destinados al DSLab, por lo que no se busca un uso lucrativo de la aplicación.

- Respecto al lenguaje de programación elegido para la implementación de la aplicación se ha elegido Objective-C, a pesar de poseer un curva de aprendizaje complicada se ha preferido a Swift. Este último ha sido presentado por la compañía recientemente, pero no posee la gran cantidad de documentación como el que ya está disponible de Objective-C. Ambos lenguajes mencionados son los únicos permitidos en iOS.
- En relación a las herramientas elegidas para el desarrollo, no hay opción ya que Apple obliga al uso de Xcode e iOS Simulator, ambos pueden ser descargados gratuitamente pero únicamente desde sistemas Apple. El primero de ellos, es el la herramienta principal, es el entorno de desarrollo y iOS Simulator, se encarga de que el programador pueda testear su aplicación simulando cualquiera de los dispositivos de Apple.

3.2.1 Versión de iOS elegida

La decisión de elegir la versión mínima compatible para el desarrollo del proyecto, es una de vital importancia. Debido a que tal y como hemos mencionado en el apartado del “Estado del arte”, con cada versión de iOS se hacen compatibles los nuevos lanzamientos pero también se descatalogan dispositivos anteriores. Adicionalmente, versiones anteriores de iOS no poseen funciones muy útiles que si están presentes en versiones actuales. Por ello, para ayudar con la decisión se estudió la figura 3-1 mostrado anteriormente:

77% of devices are using iOS 8.



As measured by the App Store on March 16, 2015.

Figura 3-1: Distribución de versiones iOS

* Imagen original de <http://www.soyapps.com/ios-8-77-distribucion/>

Como se puede observar, aún hay un 20% con la versión anterior a iOS 8. Por lo que, se ha decidido establecer como sistema operativo iOS 7 mínimo compatible. De esta forma, poder cubrir la gran mayoría de dispositivos. Cabe destacar que otro motivo a tener en cuenta, es que debido al éxito de ventas que tuvo el iPhone 4 y siendo éste compatible hasta la séptima versión de iOS, ha sido un punto a favor para elegir como mínimo la última versión disponible para este dispositivo.

3.2.2 Idioma

En un primer momento se decidió que el lenguaje elegido para la aplicación fuera el inglés, para poder alcanzar una cuota de mercado más internacional. No obstante, el sistema operativo iOS permite la traducción de aplicaciones de forma sencilla, por lo que se descartó que únicamente se soportara un único idioma. A continuación, se explica con detalle como realizar traducciones usando el entorno de desarrollo de la plataforma iOS:

Se creará un fichero por cada idioma que será soportado por nuestra aplicación, es necesario crear un fichero en nuestro proyecto denominado Localizable.strings. Éste almacenará todas las cadenas de texto de la aplicación, a cada una de ellas se le asocia una etiqueta que permitirá el acceso a ellas como recurso. Cuando sea necesario hacer uso de una palabra o conjunto de ellas, se escribirá el nombre de la etiqueta haciendo uso de la clase denominada NSLocalizedString, que cuenta con dos parámetros. El primero de ellos, es el nombre identificador del mensaje a mostrar, y el segundo parámetro es lo que se denomina ayuda al traductor, que es una sencilla explicación del texto que se

muestra en este caso. Con el uso de esta clase, según sea el idioma del dispositivo mostrara el texto correspondientemente traducido.

Por lo que, de esta forma no conlleva un trabajo pesado de ir traduciendo texto a texto, incluyendo condiciones para mostrar una cadena de texto u otra. Mediante el uso de etiquetas facilita el trabajo al programador y la traducción a otros idiomas a personas que no tenga conocimiento del lenguaje de programación utilizado.

Finalmente, se ha decidido que la aplicación para la plataforma iOS soporte dos idiomas Español e Inglés. Siendo está última el lenguaje predefinido por defecto, sin embargo, tal y como se ha comentado anteriormente, si el usuario que descargue la aplicación tiene en su dispositivo configurado en Español la aplicación se mostrará en dicho idioma.

3.2.3 Dispositivo compatibles

La aplicación va a ser implementada para que pueda ser ejecutada a partir del iPhone 4 hasta los últimos modelos de Smartphone lanzados iPhone 6 y 6 Plus. En cuanto a los dispositivos tablet serán compatibles el iPad 2 y superiores. Por lo que, será posible mostrar correctamente la aplicación en pantallas desde 3,5 pulgadas hasta 10 pulgadas, ajustándose en todo momento a la resolución del dispositivo en el cual se encuentre instalada la aplicación.

En pantallas de menor tamaño, para poder visualizar correctamente la interfaz se añadirán elementos ScrollView para permitir al usuario desplazarse en la dirección deseada y así poder obtener una visión completa de la pantalla. Siendo este elemento muy utilizado en la gran parte de ventanas mostradas. En situaciones en las cuales, es necesario dibujar elementos en pantalla se hace uso de clases del tipo “View” que permiten dicha función como si la vista mostrada se tratara de un lienzo. Para cada tamaño de pantalla, se han realizado modificaciones en estas clases para que dichos elementos sean mostrados correctamente y se ajusten a la resolución utilizada por el dispositivo.

No obstante, para facilitar el diseño de la aplicación y evitar posibles fallos de visualización, se ha forzado el modo de funcionamiento “Portrait”. Este modo, no variará la orientación aunque la pantalla del dispositivo sea girada. De esta manera, poder mejorar la calidad del visionado de la aplicación. Sin embargo, en la herramienta Karn Auto, en el caso de 5 variables elegidas, para poder realizar un mejor uso del espacio se ha forzado el modo de funcionamiento “Landscape”. En este modo, se mostrara la pantalla en horizontal.

3.2.4 Resumen

La aplicación será compatible con dispositivos móviles como para tabletas con sistema operativo iOS 7 en adelante. Siendo mostrada de forma nativa en todos los tamaños de pantallas soportados.

El idioma predeterminado será el inglés, aunque poseerá la detección de localización, si el dispositivo se encuentra configurado en español, se cambiara el idioma de la aplicación automáticamente sin que el usuario se consciente.

3.3 Herramientas disponibles en la aplicación

En esta sección se va a explicar las principales herramientas que han sido incluidas en la aplicación. En esta ocasión, se dotará de mayor precisión a la información de cada una de ellas.

3.3.1 Tutorial

Esta primer herramienta disponible para el usuario, proporciona una serie de imágenes estáticas que ofrecen una breve e interesante explicación de cómo usar los mapas de Veitch-Karnaugh para la simplificación de funciones lógicas. De este forma, sentar las bases de las herramientas de la aplicación, con especial interés en la herramienta Karn Auto.

El tutorial de la aplicación consiste en una sucesión de imágenes estáticas que ofrecen una breve explicación del uso de los mapas de Veitch-Karnaugh para la simplificación de funciones lógicas. Estas explicaciones de cómo se debe simplificar y cómo no, serán muy útiles para entender la base de las herramientas ofrecidas en la aplicación, sobre todo Karn Auto. No obstante, este tutorial no nos familiariza con el uso de la aplicación, ya que para ello se ha destinado la herramienta Ayuda.

En cuanto a la implementación del tutorial, las imágenes se visualizarán en pantalla completa independientemente del dispositivo utilizado y el usuario podrá avanzar en la lectura deslizando el dedo en la pantalla. Las imágenes de esta herramienta la cual estamos explicando, han sido diseñadas por el profesor Eduardo Boemo.

3.3.2 Herramienta de selección del conjunto de variables

3.3.2.1 Descripción

La herramienta de selección del conjunto de variables, es de vital importancia para el uso de la aplicación. Permite elegir el número de variables que se desean utilizar y la letra elegida para cada una de ellas. Esto da un plus de libertad al usuario y como se ha visto anteriormente, no lo permiten aplicaciones del mismo tipo.

Tal y como se ha visto durante la carrera, en concreto en la asignatura de Circuitos Digitales, se usa por defecto el conjunto de variables CBA para facilitar el aprendizaje a los alumnos, posteriormente se va usando el conjunto de 4 variables y así sucesivamente. Por ellos, se ha establecido en la aplicación CBA como conjunto por defecto. La aplicación permite hasta conjuntos de 5 variables EDCBA. Mencionar, que las letras son modificables por el usuario.

La herramienta de selección de variables es una funcionalidad de gran importancia en este proyecto.

Durante el uso de la aplicación, si se pulsa sobre el botón de “Variables” de la pantalla principal, se accede a la vista o ventana de selección de conjunto de variables. Se muestra un lista, con los conjuntos guardados previamente en el dispositivo. Este almacenaje de datos es posible gracias a la clase disponible en iOS que recibe el nombre de NSUserDefaults. De esta última se hablara en el siguiente apartado.

En la lista mostrada, hay uno de los elementos que posee un “tick” de color azul, indicando que dicho conjunto se encuentra seleccionado. Si hiciéramos uso del resto de herramientas disponibles en la aplicación, sería éste el conjunto de variables utilizado.

En la zona superior derecha de la pantalla, se encuentra un botón con el símbolo ‘+’ que si se pulsa sobre él, muestra una ventana de diálogo que permite añadir un nuevo conjunto de variables, seleccionando el número de variables deseadas y las letras usadas para cada variable.

Respecto a la edición, se puede editar cualquiera de los conjuntos mostrados en la lista, pulsando sobre ellos, mostrándose una ventana de diálogo de edición para modificar el número de variables y letras elegidas. Una vez realizadas las modificaciones, se guardan los cambios y se vuelve a la pantalla previa. Por otro lado, si el usuario desea borrar alguno de los conjuntos existentes, debe deslizar hacia la izquierda con el dedo y se mostrar un botón en color rojo que permitirá borrar dicho conjunto.

3.3.2.2 Clase NSUserDefaults

La clase NSUserDefaults proporciona a iOS la función de almacenar datos del usuario. Ésta puede usarse para guardar todo tipo de información, ya sea configuraciones de usuario, preferencias, etc...

Los datos almacenados hacen uso de etiquetas para referenciarlos. Dicha información accesible por el usuario es guardada en la memoria reservada para la aplicación. Se ha utilizado la clase NSUserDefaults dada la sencillez de su uso, y en este caso ha permitido guardar los conjuntos de variables y configuración del usuario. La memoria utilizada para dicho fin es reducida, por lo que no se aprecia ningún tipo de bloqueo que produzca ralentización.

3.3.2.3 Limitaciones

Se ha tomado la decisión que en esta herramienta se puedan crear conjunto de 3 a 5 variables, para que el usuario disponga del conjunto mínimo para el comienzo del aprendizaje y posteriormente hacer uso de hasta 5 variables. Ya que, es el máximo utilizado en el sector de educativo, descartándose 6,7 e incluso mayor número de variables. No obstante, en un futuro se podría implementar el uso de conjuntos mayores. Respecto a otra limitación establecida, es la de evitar elegir la misma letra para dos variables, si se diera el caso, se mostrara un aviso y no pudiendo continuar hasta que se realice la modificación por parte usuario.

3.3.3 Herramienta Karn Auto

3.3.3.1 Descripción

Karn Auto es la herramienta más importante de la aplicación, siendo el objetivo principal al comienzo del proyecto y que da nombre a la misma. Permite la simplificación de funciones lógicas en tiempo real mediante el mapa de Karnaugh.

Una vez se pulsa botón “Karn Auto” en la pantalla principal de la aplicación, se muestra una vista o ventana en la que se permite elegir el número de variables que se desea utilizar entre 3 y 5. En este caso, no se refleja el número de variables elegido en la lista de conjuntos, ya que se encuentran ya definidos. Aunque si se mostrarán las letras seleccionadas para las variables.

Adicionalmente, se ofrece un acceso directo a la herramienta de selección en la zona inferior anterior.

Seleccionado el número de variables, la aplicación muestra una nueva pantalla, en la cual se visualizara una tabla estática que representará el mapa de Karnaugh. El tamaño de esta tabla se modificara según el número de variables elegidas.

Cada una de las posiciones del mapa es interactiva, pudiendo pulsar sobre cada una de ellas cambiando así el valor de la misma, entre las 3 posibilidades que se ofrece: “0”, “1” o “X”. Según se van introduciendo valores, se va calculando y mostrando en la zona inferior la función simplificada.

Para la representación de las tablas se ha utilizado la clase `UIBezierPath` disponible en iOS, que se explicará en el siguiente apartado.

Junto al resultado de la simplificación, en la zona inferior se ha habilitado una botonera. Dichos botones corresponderán a cada uno de los términos de la función simplificada, estos se irán mostrando según vaya creciendo el número de elementos. Dichos términos se representan como agrupaciones en el mapa, si se pulsa sobre uno de los botones mostrados, éste coloreará la agrupación a la cual hace referencia el término correspondiente a dicho botón.

Por defecto, la primera vez que se muestra la pantalla con el mapa con todos los valores de sus posiciones a “0”, no se mostrará ninguno de los botones que se mencionan anteriormente.

3.3.3.2 Clase `UIBezierPath`

Esta clase será la que se utilizará para dibujar los mapas de Karnaugh en la interfaz de la aplicación. Dispone de los mecanismos necesarios para hacer uso de la pantalla del dispositivo como si de un lienzo para pintar se tratara. Obteniendo un control total sobre la pantalla a nivel de píxel. Con ayuda de funciones implementadas en esta misma clase se ha procedido al dibujado de los mapas mostrados en esta herramienta.

3.3.3.3 Limitaciones

Se ha decidido 5 variables como limite máximo a representar a través de mapas de Karnaugh. Por lo que al usuario se mostrarán tres opciones: 3, 4 y 5 variables. El motivo que ha llevado a esta determinación es que un mapa de Karnaugh con un número alto de variables deja de ofrecer sencillez y dificulta la representación gráfica.

Es importante destacar que el algoritmo simplificado es el de Quine-McCluskey y que los mapas de Karnaugh muestra gráficamente como realizar las reducciones.

En los mapas de 3 y 4 variables, es posible mostrar el color de las agrupaciones formadas y los términos a los cuales pertenecen. Descartándose esta función para 5 variables debido a la resolución de los mapas y de cada una de las posiciones, no ofrecería una correcta visualización.

3.3.4 Herramienta Traductor

La herramienta Traductor se ha implementado para poder ofrecer al usuario otra forma alternativa a los mapas de Karnaugh. Ésta es, introducir funciones lógicas para después éstas sean simplificadas. En el caso de que la función sea de gran tamaño se ofrecerá diferentes vías para introducirla, que a continuación serán explicadas las más importantes. No obstante, a diferencia de la herramienta Karn Auto, la simplificación no se realizará en tiempo real, sino que el usuario deberá pulsar un botón para que esta se efectúe.

3.3.4.1 Introducir funciones por tablas

La primera opción implementada para la introducción de funciones es por medio de tablas. Dentro de la misma se ofrecen 3 modos: Tabla de verdad, sumatorio de minterminos y producto maxiterminos.

En el primer modo mencionado, muestra una tabla con dos columnas. En una de ellas están todas las combinaciones posibles con el número de variables elegido desde el selector. Por otro lado, la segunda columna se encuentra por defecto llena de ceros, si es pulsado alguno de ellos, éste cambia su valor. Siendo posible introducir unos o equis a lo largo de la tabla mostrada. Una vez que el usuario, haya establecido los valores deseados en la tabla, construyéndose así una tabla de verdad personalizada, se debe pulsar sobre el botón “Traducir” de la zona inferior de la tabla.

Por otro lado, los otros dos modos disponibles, se muestra en la primera columna todas las posibilidades de maxiterminos o minterminos respectivamente. Si en la segunda columna con los valores mostrados a cero por defecto, modificamos los valores pulsando sobre los mismos, si es un “1”, el término correspondiente se encontrará presente en la función que desea introducir el usuario. En caso de ser una “X”, no cobrará importancia el valor que tome dicho término.

En cada una de las tablas mostradas, independientemente del modo elegido, se ha implementado un botón “Reset” que devuelve la tabla al estado inicial. El estado de la tabla se guardará únicamente si se desea realizar modificaciones, cuando previamente se ha pulsado el botón para traducir y a continuación se ha pulsado el botón de retroceso. Éste último se encuentra en la zona superior derecha y está presente en todas las vistas o ventanas de la aplicación.

3.3.4.2 Introducir funciones por expresión lógica

Finalmente, la última opción para la introducción de funciones es por expresión lógica. En la pantalla correspondiente a este método, se muestra una interfaz, la cual posee un cuadro de texto que hace la función de pantalla de una calculadora tradicional y en la zona inferior una serie de botones que permiten la introducción de valores por parte del usuario. Descartando el uso del teclado por defecto mostrado por iOS, los motivos de esta decisión:

- Facilitar al usuario la introducción de dato sin estar cambiando de teclados, ya que se combinan datos que son variables junto a operadores, pudiendo a llegar a entorpecer el proceso.
- Visualizar únicamente los valores que pueden ser introducidos, y evitando errores por introducir datos incorrectos.

Por ello, se ha elegido crear un teclado personalizado pensando en el usuario. Éste ofrece los siguientes elementos:

Las letras del conjunto de variables que el usuario ha seleccionado previamente. Los valores “0” y “1”, por si son necesarios introducirlos en determinada situación.

Se ha incluido paréntesis de apertura y de cierre para cambiar el orden de las prioridades de las operaciones.

Diferentes operadores lógicos, éstos son los mostrados en la siguiente tabla :

Símbolo	Función	Prioridad
¬	NOT	1
*	AND	2
/	NAND	2
+	OR	3
-	NOR	3
#	XOR	3

Tabla 3-1: Símbolos y prioridades disponibles para expresión lógica

Para mejorar la experiencia, a parte del botón traducir, se ha introducido un botón ayuda que enseña al usuario como introducir correctamente las expresiones en esta herramienta. Adicionalmente, se ha implementado un botón borrar para que el usuario pueda eliminar el último termino introducido.

3.3.4.3 Traducciones realizadas por la herramienta

En el momento que el usuario pulsa el botón traducir en cualquiera de las funciones explicadas anteriormente, se muestra una nueva pantalla en la que se puede observar los siguientes elementos:

- En primer lugar se muestra la función introducida, independientemente del método de entrada elegido.
- En segundo lugar, se proporciona la función simplificada de la que ha introducido y que se muestra en la zona superior.
- A continuación, se puede visualizar diferentes botones:

- Sumatorio de minitérminos, si pulsamos sobre éste, nos proporciona el sumatorio de minitérminos en una nueva pantalla
- Producto de maxitérminos, al pulsar sobre él, muestra en una nueva pantalla el producto de maxitérminos.
- Tabla de verdad, éste permite al usuario observar la tabla de verdad resultante, también en una nueva pantalla.

3.3.4.4 Limitaciones

En relación a la introducción de funciones mediante expresión lógica, se estableció un límite máximo para la misma. Éste quede definido por el tamaño de cuadro de texto que hace la función de pantalla, en la cual se reflejan la expresión que se va introduciendo. Además se ha implementado un control de errores para evitar que el usuario introduzca expresiones erróneas. Dicho control, mostrará un mensaje de aviso informando del error cometido en la expresión y no permitiendo continuar con la traducción. Se ha decidido informar del error con detalle para que el usuario pueda aprender las reglas establecidas en la introducción de funciones.

3.3.5 Herramienta Conversor

3.3.5.1 Descripción

El conversor es la herramienta que hace el papel de traductor entre los siguientes sistemas numéricos que se indican a continuación:

- Binario
- Octal
- Decimal
- Hexadecimal

Se ha decidido incluir dicha herramienta, ya que puede ser de utilidad en las prácticas de laboratorio de la asignatura de Circuitos Digitales de la EPS, cuando el alumno necesita realizar la conversión de sistemas. Con ayuda de esta herramienta, se ahorra tiempo en la conversión y mejora el uso de la calculadora de Windows disponibles en los ordenadores del laboratorio.

3.3.5.2 Limitaciones

Se han establecido ciertas limitaciones en la herramienta en cuestión. Éstas son:

- Para evitar introducir valores no reconocidos por cada sistema, se ha implementado un teclado personalizado en el cual los botones con los elementos aceptados por el sistema seleccionado se muestran y en el caso contrario se encuentran ocultos.

- Se han incluido hasta 16 caracteres como límite, que son los que se muestran cuando el sistema hexadecimal se encuentra seleccionado. En el resto de sistemas, el número de botones mostrados es menor.
- Se ha fijado un número máximo de elementos introducidos en cada sistema para evitar bloqueos, inhabilitando los botones una vez se llegue al máximo establecido por el sistema de conversión elegido.

3.3.6 Herramienta de ayuda

En todo el desarrollo del proyecto, se ha dado prioridad a la sencillez en todas las herramientas implementadas. No obstante, se ha incluido un manual completo para ayudar al usuario en caso de duda a la hora de hacer uso de la aplicación. El manual contiene textos e imágenes estáticas e incluye los siguientes temas:

- Elegir conjunto de variables
- Herramienta Karn Auto
- Herramienta Traductor
- Introducir expresiones lógicas
- Introducir expresiones por tablas.
- Herramienta Conversor

4 Desarrollo

En este apartado se va a explicar con un mayor detalle la implementación de la aplicación. Profundizando en los pasos seguidos en el desarrollo de los puntos de mayor dificultad de la aplicación y como se han solventado.

4.1 Primeros pasos

La programación de aplicaciones en iOS presenta una curva de aprendizaje lenta lo que dificulta en el comienzo familiarizarse con el lenguaje utilizado, en este caso Objective-C. En este caso, en un principio no se tenía conocimiento del mismo y por ello se han realizado pasos previos antes de comenzar la programación de la aplicación.

Como punto de partida a la hora de usar cualquier lenguaje, es necesario tener un conocimiento previo y completo de lo fundamentos de programación, independientemente del lenguaje utilizado. En este caso, tal y como se ha indicado anteriormente, en iOS el lenguaje utilizado es Objective-C, aunque ya se ha comenzado el uso Swift, nuevo sistema lanzado también por Apple. No obstante, es necesario tener una visión abstracta del lenguaje, conociendo la herencia, tipos de datos, el papel de los métodos, etc..

Una vez asentadas las bases, se puede comenzar con la práctica. Para ello, el primer paso será instalar el entorno de desarrollo y familiarizarnos con él. Al comienzo no se debe implementar una gran aplicación, sino desarrollar pequeñas con diferentes funciones como pueden ser creación de botones, almacenar datos de usuario, uso de scrollView, etc...

Realizado todo el proceso de aprendizaje descrito anteriormente, se puede empezar con el desarrollo del proyecto.

En nuestro caso, ha sido necesario cumplir cada punto mencionado anteriormente, ya que durante la carrera no se ha impartido el lenguaje usado en la plataforma iOS. No obstante, una vez obtenido las bases y conocimientos necesarios, es cierto que el entorno de desarrollo de Apple ofrece multitud de ayudas facilitando su uso, cabe mencionar que es necesario un buen inglés ya que esta completamente en este idioma.

En los pasos previos como durante el desarrollo del proyecto, se han consultado numerosos tutoriales y cursos online disponibles en internet. Estos han sido de gran utilidad, tanto a la hora de conocer el lenguaje como para el uso del entorno de programación.

Particularmente previamente a la asignación oficial de este proyecto, se realizó el curso gratuito de iOS 7 disponible en la web <http://bitfountain.io/>, el cual se recomienda encarecidamente. Una vez finalizado el mismo, se verificó la posibilidad real de traducir el algoritmo principal de la aplicación, Quine-McCluskey del lenguaje java a Objective-C. Comprobado dicho punto se presentó de forma oficial el anteproyecto.

Tal y como se ha comentado anteriormente, se ha usado internet para consultar dudas que van surgiendo durante el desarrollo de la aplicación. Uno de los foros mas visitados, el cual es una de las comunidades de programación más conocidas es “stackoverflow” y ha sido de vital importancia para la consecución de los objetivos propuesto en el proyecto. Además, mencionar que se ha consultado otras páginas que serán referenciadas en esta misma memoria.

4.2 Instalación del entorno de desarrollo

El entorno de desarrollo en iOS es único y es la herramienta Xcode, ésta trabaja de manera conjunta con Interface Builder (herramienta gráfica para crear interfaces de usuario). Xcode se encuentra disponible en el Mac Store de Apple de forma gratuita y solo es compatible con el sistema operativo Mac OS X.

Además esta herramienta incluye el simulador iOS Simulator, permitiendo compilar y ejecutar el código en cualquier de los dispositivos móviles de Apple de forma simulada, de forma que no es necesario poseer uno para poder comprobar el correcto funcionamiento de la aplicación que se esté desarrollando.

4.3 Esquema de la aplicación Karn Map

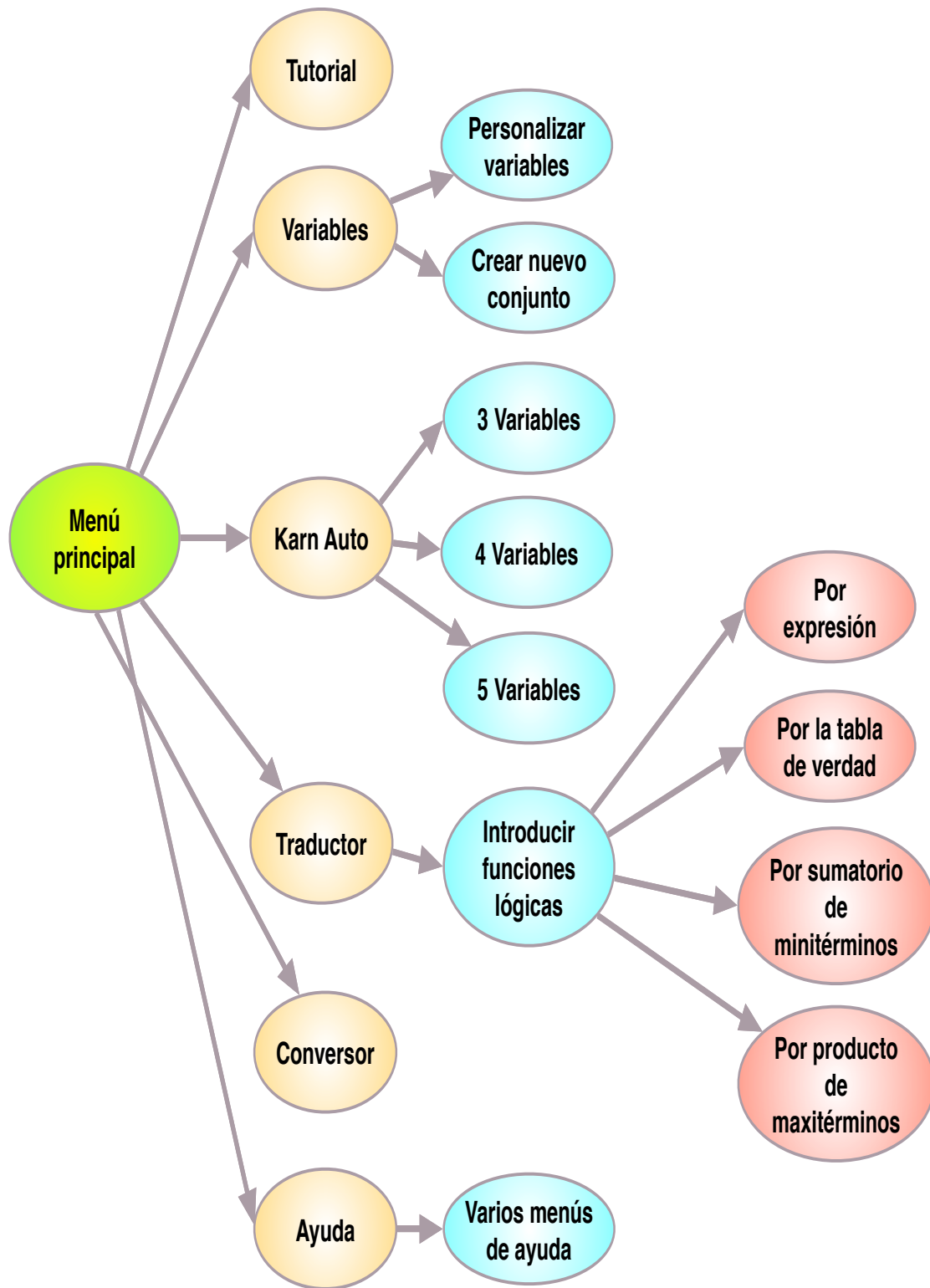


Figura 4-1: Esquema general de la aplicación.

4.4 Conceptos clave

En esta sección se van a explicar algunos de los conceptos que es necesario destacar de iOS para mejorar la comprensión de los siguientes apartados.

4.4.1 Recursos utilizados en una aplicación

Durante del desarrollo de una aplicación en iOS es recomendable externalizar de la misma los recursos utilizados durante la ejecución del código. Como por ejemplo, pueden ser las imágenes, audios o bien textos. Esto permite el uso de estos recursos que se utilizarán en función del tipo de dispositivo sin ser necesarios realizar modificaciones del código o bien pueden usarse recursos genéricos. Un ejemplo de recurso genérico es el sonido establecido al pulsar cualquier botón de la aplicación. Otros recursos no genéricos utilizados en el proyecto son:

- Cadena de textos para que la aplicación soporte distintos idiomas.
- Imágenes de distintas resoluciones para que sean visualizadas según la resolución del dispositivo.

Estos recursos son guardados en la carpeta “Supporting Files” de la aplicación.

4.4.2 Estructura de una aplicación de iOS

Las aplicaciones desarrolladas en esta plataforma siguen el modelo MVC: Model View Controller. Se puede observar gráficamente en la siguiente figura:

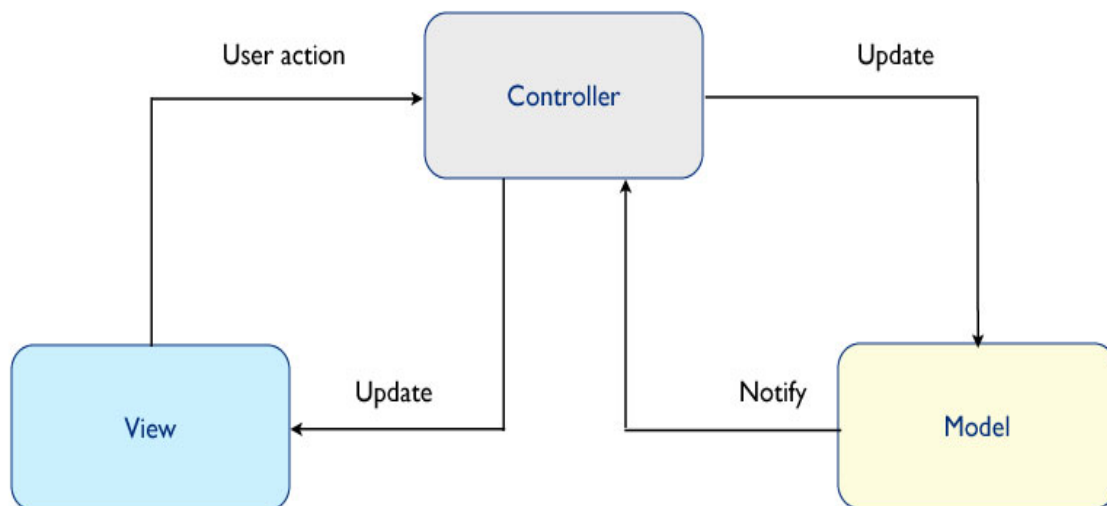


Figura 4-2: Diagrama modelo MVC

* Imagen original de blog.teamtreehouse.com

A continuación, se detallan cada uno de los elementos mostrados en el diagrama anterior:

- View: es cualquier elemento visual mediante el cual un usuario de la aplicación pueda ver e interactuar. Algunos ejemplos de este tipo de elementos son: botones táctiles, un “label” o etiqueta de texto no editable o un “text field” cuadro con texto editable, que son clase de la UIView.
- Model: incluye objetos que gestionan los datos, tanto para guardarlos como para manipularlos. El elemento Model no es apreciable por el usuario ya que es independiente a la interfaz visual.
- Controller: el encargado del control de la aplicación, manda y recibe órdenes del View y del Model. Este elemento es el que permite que se efectúen cambios en la interfaz o realizar funciones cuando se pulsa un botón táctil de la interfaz visual.

El diagrama Model View Controller tiene ciertas limitaciones que hay que tener en cuenta:

- El Controller puede comunicarse con View y Model simultáneamente.
- Model y View no pueden comunicarse entre sí.
- Los Controllers pueden comunicarse con otros Controllers de la aplicación.

En la aplicación desarrollada se encuentran archivos de estas clases:

- UINavigationController: su función es la de presentar y gestionar los Views, gestiona la relación entre los Views y los Models.
- Existen dos tipos de ficheros .h y .m, en los cuales se pueden crear referencias, métodos o incluso señalar a qué otro UINavigationController ir, en caso de querer realizar una acción concreta.
- Storyboard: es una de las clases más características de esta programación, ya que en ésta, es donde controlaremos el tipo View. En ella, es posible crear UINavigationController, para después ser asociado a un fichero determinado del proyecto, con lo que obtendrá sus métodos y características.
- AppDelegate: los archivos de extensiones .h y .m, se generan cada vez que se crea un proyecto y su función es la de controlar entre otras cosas el comienzo o el fin de la aplicación.
- UITableView: son objetos basado en celdas, usadas en este proyecto en concreto para que el usuario pueda seleccionar y añadir conjuntos de variables que después se usarán en las herramientas de la aplicación. Los objetos de esta clase permite cambiar fondos, letras, añadir imágenes, etcétera. Además pueden tener la característica de Controller (UINavigationController), que es la de controlar la aplicación.

- UINavigationController: el objeto de esta clase es el encargado de guiar al usuario por las diferentes vistas de la aplicación, es decir controla la comunicación entre las mismas y además controla el retroceso, es decir, el uso del botón atrás que se muestra en toda la aplicación.

4.4.3 Ciclo de vida de la clase UIViewController

El ciclo de vida de un objeto de la clase UIViewController consiste en los diferentes estados por los que puede pasar cuando el usuario interactúa con la interfaz. Se muestra de forma detallada en la siguiente figura:

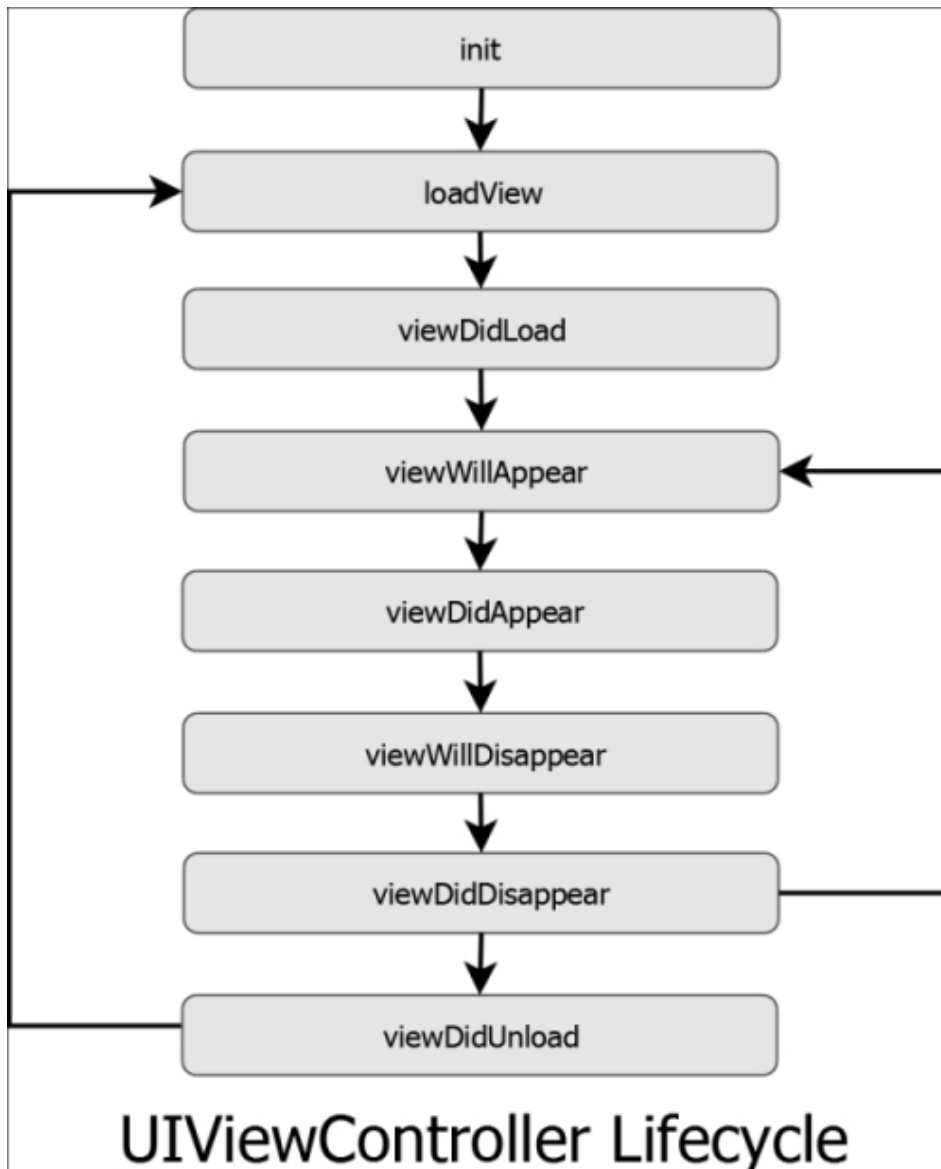


Figura 4-3: Diagrama modelo MVC

* Imagen original de <http://www.safaribooksonline.com>

Tal y como se puede observar en la figura anterior, existen una serie de métodos que se invocan cuando se produce un cambio entre un estado y otro tras un evento. Alguno de los más importante, se explican a continuación:

- **init**: Inicia la aplicación
- **loadView**: Carga la vista en memoria
- **viewDidLoad**: Se invoca después de que la vista se ha cargado en memoria.
- **didReceiveMemoryWarning**: El controlador recibe un aviso en caso de un problema de memoria.
- **viewWillAppear**: Se invoca justo antes de que la vista aparezca.
- **viewDidAppear**: Se invoca después de que la vista ha aparecido.
- **viewWillDisappear**: Se invoca justo antes de que la vista desaparezca.
- **viewDidDisappear**: Se invoca después de que la vista ha desaparecido.
- **ViewDidUnload**: En Objective-C, es donde hace su limpieza y liberación de las cosas, esto es manejado automáticamente.

4.4.4 Objetos View

En este apartado, se explicaran con mas detalle los objetos de esta clase, que se ha mencionado previamente. Estos objetos son básicos y vitales al a hora de implementar la interfaz gráfica, ya que una vez creados podemos añadirle elementos como pueden ser botones, cuadros de textos, etiquetas, etc... Estos pueden verse en cualquier interfaz de un aplicación iOS.

Cabe destacar, que existe un tipo de objeto View particular que permite la visualización de tablas interactivas para el usuario y recibe el nombre de TableView. Consta de celdas las cuales puede contener datos que a su vez están indexados a otro objeto del tipo View.

4.4.5 Objetos UIAlertView

Un UIAlertView es un objeto de la clase UIAlertView, su implementación es muy sencilla, generalmente es un objeto que se implementa dentro de un método, o bien en el método viewDidLoad de tu vista, de forma que no sea necesaria la interacción del usuario, sino que se “lance” de forma automática al realizar otro proceso. En concreto en este proyecto se ha utilizado para informar al usuario desde errores a la hora de introducir datos hasta de advertencia a la hora de borrar datos guardados.

4.4.6 Clases ya comentadas anteriormente

Recordar la clases ya mencionadas en el apartado diseño como son NSUserDefaults y NSLocalizedString.

4.4.7 Archivo de manifiesto

El archivo de manifiesto es el más importante de todos los proyectos creados en el entorno de desarrollo Xcode. Éste es generado automáticamente por Xcode al crear un nuevo proyecto y recibe la extensión “.plist”. Incluye datos muy importante de la aplicación como son:

- El idioma por defecto de la aplicación.
- Nombre y versión de la aplicación.
- Declaración de la mínima versión compatible en la que funciona la aplicación.
- Orientaciones soportadas por la aplicación.
- Compatibilidad soportada, en este caso con dispositivos iPad y iPhone.

4.5 Detalles de la implementación

En esta sección se va a explicar los puntos más complicados a la hora de su implementación durante el proyecto, y como se han solventado los problemas surgidos. Para facilitar la comprensión de las explicaciones, se ha dividido las mismas por cada módulo y actividades de la aplicación.

4.5.1 Tutorial

Previamente ya se explicó esta herramienta de la aplicación, ésta consistía en una serie imágenes que explican como realizar las simplificaciones de funciones con ayuda de mapas de Karnaugh. Para la implementación de este módulo, se han seleccionado las imágenes de forma ordenada, cargándolas desde los recursos de la aplicación, automáticamente las imágenes se ajustarán a la resolución del dispositivo usado por el usuario. Además de hacer uso del ScrollView horizontal para permitir avanzar en el tutorial deslizando el dedo en la pantalla.

4.5.2 Herramienta de selección del conjunto de variables

4.5.2.1 Descripción

La interfaz gráfica de esta herramienta consta en la zona superior izquierda el botón de retroceso y en la derecha el botón para añadir un nuevo conjunto de variables. En la zona central se visualiza una lista con los conjuntos de variables almacenado y que pueden ser seleccionados pulsando sobre su celda correspondiente. Se adjunta figura de la interfaz:

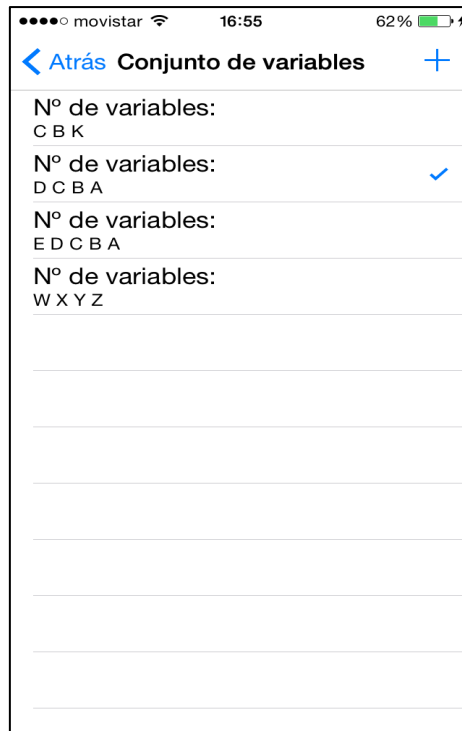


Figura 4-4: Menú conjunto de variables

El sistema operativo iOS dispone de vistas de la clase `UIViewController` pero cuando se desea mostrar tablas, se utiliza un tipo de clase concreta que recibe el nombre de `UITableViewController`.

Para crear una vista en formato tabla como la requerida en este caso, creamos un objeto de la clase indicada anteriormente, `UITableViewController`. A continuación se añaden los botones a la interfaz y se establece el número de celdas que se desean tener. Es posible modificar el aspecto de las celdas de la lista, en este caso se ha elegido que cada una de ellas posea un título y un subtítulo. El título indicará el número de variables del conjunto y el subtítulo las letras elegidas para representar las variables. Se pueda apreciar en la siguiente figura:

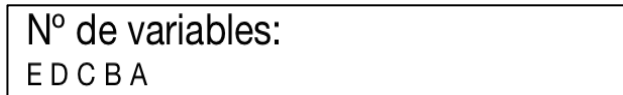


Figura 4-5: Celda de la lista conjunto de variables

Para nuestro proyecto, se ha decidido establecer por defecto, es decir, cuando el usuario usa por primera vez la aplicación, que estén disponibles tres conjuntos predeterminados. Estos se muestran a continuación:

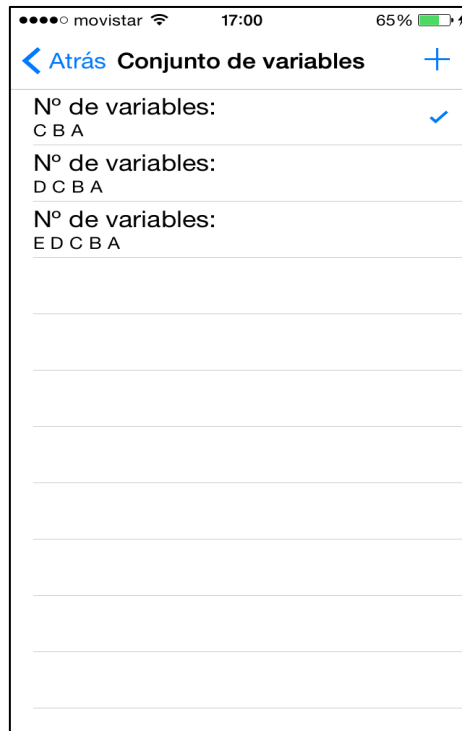


Figura 4-6: Menú conjunto de variables por defecto

Por lo tanto, en un comienzo existirán 3 celdas. No obstante, este número es completamente modificable por el usuario tanto para reducirlo como para aumentarlo e incluso modificar los conjuntos por defecto según las necesidades del usuario. Estos cambios realizados en cuanto a los conjuntos de variables, serán almacenados para que aunque se cierre la aplicación, al ejecutarla de nuevo se conserven los conjuntos, gracias a la clase mencionada en apartados anteriores `NSUserDefaults`.

Por otro lado, como se ha comentado anteriormente, es necesario seleccionar un conjunto para usarlo en el resto de herramientas, por defecto se encuentra seleccionado el conjunto CBA. Para cambiar la selección, se debe pulsar sobre el conjunto deseado y se marcará un “tick” para indicar que se encuentra seleccionado dicho elemento. Se puede observar este detalle en la siguiente imagen:

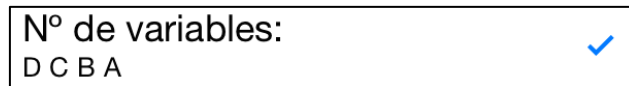


Figura 4-7: Selección del conjunto de variables

Esta selección también será almacenado como datos de usuario con la clase NSUserDefaults, evitando que con el cierre de la aplicación no se guarde la selección realizada en el anterior uso de la aplicación.

En los siguientes apartados se explicara como se ha implementado la modificación, borrado y añadido de nuevos conjuntos.

4.5.2.2 Modificar un conjunto de variables

Esta función se ejecuta al pulsar sobre alguno de los conjuntos disponibles en la lista. Al realizar la pulsación, aparece un cuadro diálogo llamado “Personalizar” como el siguiente:

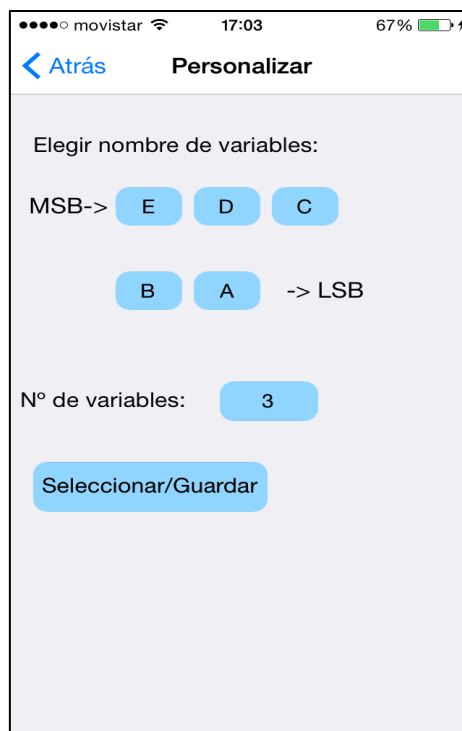


Figura 4-8: Modificar conjunto de variables

En él se puede modificar el valor de cada una de las variables y el número de variables deseadas. Si se desean guardar los cambios se pulsara sobre el botón “Seleccionar/Guardar” y se retornará a la lista inicial pero mostrándose los cambios efectuados sobre el conjunto, además de aparecer seleccionado. En caso de no querer guardar los cambios realizados en el cuadro de diálogo, pulsaremos sobre “Cancelar” y se volverá la lista inicial sin mostrar cambios el conjuntos pero sí apareciendo seleccionado, indicado con el “tick”.

4.5.2.3 Añadir un nuevo conjunto

Para acceder a esta función implementada, se deberá pulsar en el botón “+” situado en la zona superior derecha de la pantalla. Su funcionamiento es similar a cuadro de diálogo de modificación de conjuntos, pero para este caso se ha utilizado un tipo de transición al cuadro de diálogo emergente diferente, éste es del tipo Modal. Creando una pequeña animación cuando se produce la transición, el cuadro mostrado es el siguiente:

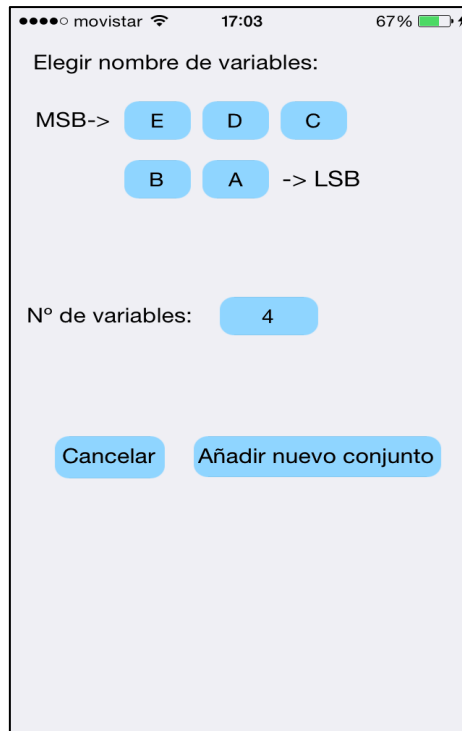


Figura 4-9: Añadir nuevo conjunto de variables

Al igual que en el apartado anterior, es posible seleccionar las letras asociadas a cada variable junto al número de las mismas. Si se desea añadir el nuevo conjunto se debe pulsar en el botón “Añadir nuevo conjunto”, y se volverá a la lista inicial apareciendo este nuevo conjunto. En caso de que no se desee añadir el conjunto configurado en el cuadro de diálogo, se pulsará “Cancelar” y se retornara a la pantalla previa con los conjuntos que ya existían previamente.

4.5.2.4 Eliminar un conjunto

Esta función ha sido implementada con ayuda de una función disponible en iOS que usa una pequeña animación muy conocida para el borrado de un elemento. Esta se ejecuta cuando sobre una celda que contiene un elemento, en este caso un conjunto de variables, si se desliza hacia la izquierda con el dedo, se muestra un

botón de borrado en color rojo. Se adjunta imagen para facilitar la comprensión de la explicación:



Figura 4-10: Borrar un conjunto de variables

Una vez pulsado el botón “Borrar”, automáticamente dicha celda desaparecerá de la lista actualizándose la posiciones de los conjuntos de variables almacenados. Es posible borrar todos los conjuntos de la lista, no obstante no tiene mucho sentido pero se ha permitido dicha acción.

4.5.2.5 Principales problemas encontrados

Una gran cantidad de conjunto almacenadas puede provocar un gran uso de memoria por parte de la aplicación, por ello se ha establecido un límite preventivo de número de creación de conjuntos. No obstante, en este aspecto iOS gestiona automáticamente el uso de la memoria, con lo cual libera al desarrollador de muchas preocupaciones respecto a este tema.

4.5.3 Herramienta Karn Auto

A continuación se explicará con detalle la implementación de esta herramienta disponible en la aplicación.

4.5.3.1 Elección del número de variables del conjunto

Una vez el usuario pulsa sobre el botón “Karn Auto” del menú principal, se muestra una nueva pantalla, en la cual aparecen los 3 posibles conjuntos de variables, los cuales se desean mostrar en mapa de Karnaugh. Estos son de 3, 4 y

5 variables, cada uno de ellos están asociados a un botón que indica las letras asociadas a las variables. Estas letras son las elegidas en el apartado conjunto de selección de conjuntos, y son cargadas desde la clase NSUserDefaults. Este menú de opciones disponibles en cuento a la número de variables permitido en Karn Auto no tiene complejidad.

4.5.3.2 Función principal de Karn Auto

Como se ha comentado recientemente Karn Auto ofrece la posibilidad de 3 posibles número de variables 3, 4 y 5. Cada uno de ellos, ha sido implementados de forma similar, por lo que se decidido ver de forma exhaustiva el funcionamiento de Karn Auto de 4 variables.

Karn Auto de 3, 4 y 5 variables consiste una nueva vista que hace la función de lienzo, en la cual se dibuja el mapa correspondiente a cada número de variables. En cada una de las posiciones se ha implementado un botón que según se pulse, cambiará el estado del mismo de “0”, “1” y “X”. Para el dibujado realizado, tal y como se ha mencionado en apartados anteriores, se hace uso de la clase UIBezierPath. Ésta permite pintar líneas rectas, tratando la vista o ventana mostrada como si de un eje de coordenadas 2D se tratara. Un ejemplo de uso de esta clase, son las funciones utilizadas durante el proyecto para dibujar una línea recta:

```
[bezierPath moveToPoint:CGPointMake(X,Y)];  
[bezierPath addLineToPoint:CGPointMake(X,Y)];
```

La primera línea colocará un punto imaginario en la coordenadas (x,y) que actuará de origen de la recta. Y la segunda, dibujara una línea con punto final el indicado como parámetro (x,y) en dicha función. De esta forma, se ha ido representando los mapas de 3, 4 y 5 variables. No obstante, se ha ido ajustando la resolución a los diferentes mapas de pantalla de cada uno de los dispositivo disponibles. También es posible, personalizar el ancho de las líneas como el color de las mismas.

Por otro lado En cada espacio, se ha colocado un botón táctil. Esté una vez pulsado cambiara su estado, por defecto mostrar un “0”. En cada pulsación se llamará a la función que contiene el algoritmo Quine Mc-Cluskey e irá mostrando la expresión simplificada de la función algebraica en la parte inferior de la pantalla, tal y como se muestra a continuación en la figura :

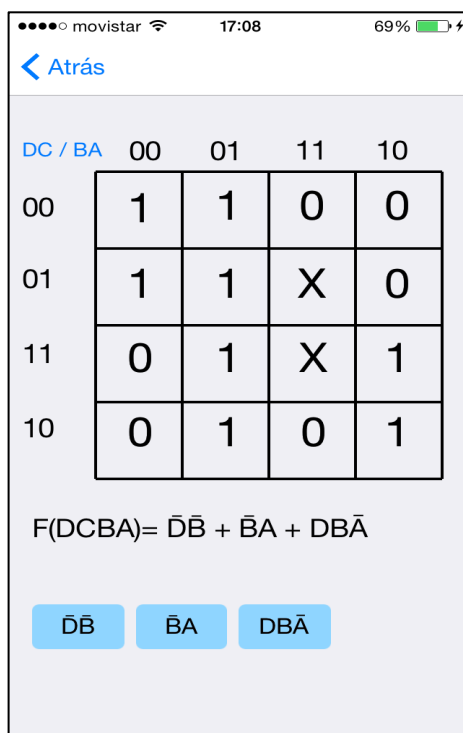


Figura 4-11: Funcionamiento Karn Auto 4 variables

El funcionamiento de cada botón es:

- Por defecto todos se encontrarán en el estado “0”
- Una pulsación cambiará el estado de “0” a “1”, “1” a “X”, o bien de “X” a “0”.
- Dos pulsaciones cambiará el estado de “0” a “X”, “1” a “0”, o bien de “X” a “1”

4.5.3.3 Representación gráfica de las agrupaciones de términos

En la parte inferior de la pantalla de esta herramienta, se ha implementado una serie de botones que irán apareciendo según vayan aumentando el número de términos de la función, ya que cada botón representará uno de ellos. Puede observarse esto en la siguiente figura:

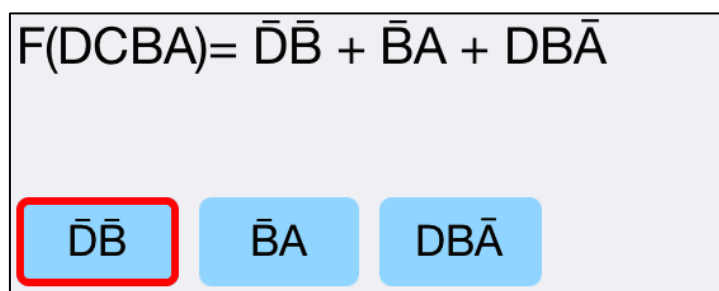


Figura 4-12: Botones Karn Auto

Por defecto no aparecerá ningún botón disponible, una vez vayan cambiando los estados de los botones del mapa, irán apareciendo tal y como se ha indicado anteriormente. Al pulsar en cada uno de los botones, se marcará el borde del mismo con un color, éste será el mostrado por los estados “0”, “1” o “X” que formen el grupo del termino en cuestión. Para facilitar la comprensión se adjunta la siguiente imagen de Karn Auto de 4 variables:

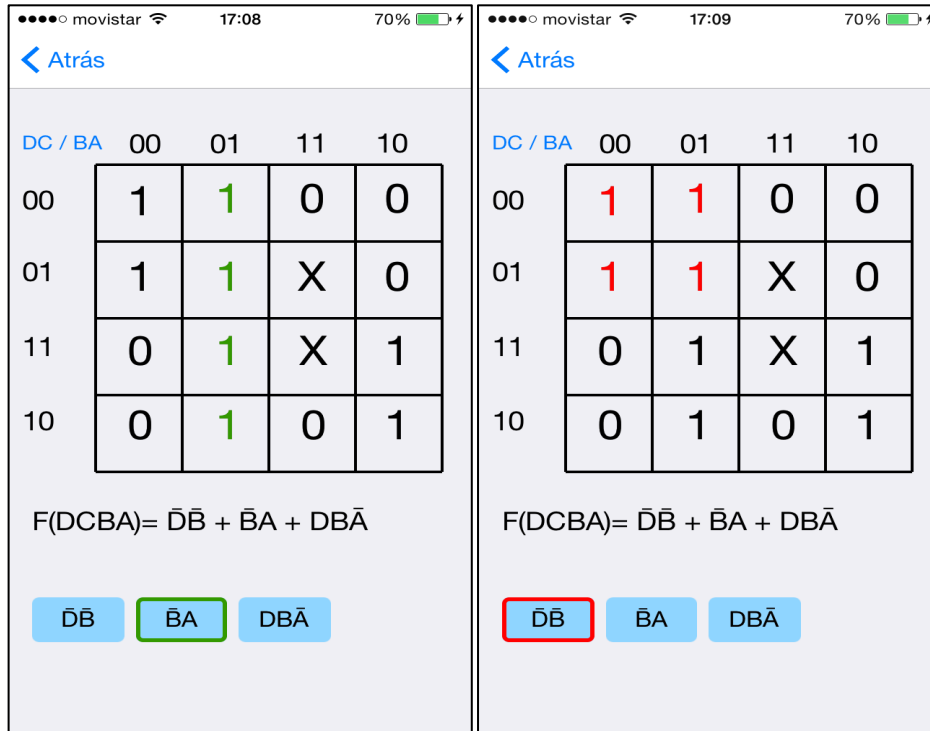


Figura 4-13: Visualización de agrupaciones

4.5.4 Herramientas para introducir funciones en el Traductor

El usuario al pulsa sobre el botón “Traductor” del menú principal, se mostrará una nueva pantalla con diferentes botones. Cada uno de ellos tiene asociada una función, estas se detallarán a continuación:

4.5.4.1 Elección del número de variables del conjunto

El botón de la zona inferior que recibe el nombre de “Ir a selección de variables”, únicamente llevará al usuario la herramienta de selección de conjuntos. Una vez elegido el conjunto deseado, se pulsará sobre el botón “Atrás” para volver la herramienta “Traductor” almacenando, con ayuda de la clase NSUserDefaults, el conjunto seleccionado por el usuario para que sea usado en todas las herramientas de la aplicación.

4.5.4.2 Introducción por expresiones lógicas

Esta función de la herramienta “Traductor”, tal y como se comentó en apartados anteriores, se ha diseñado una interfaz similar a una calculadora tradicional. Prueba de ello es la imagen que se muestra a continuación:

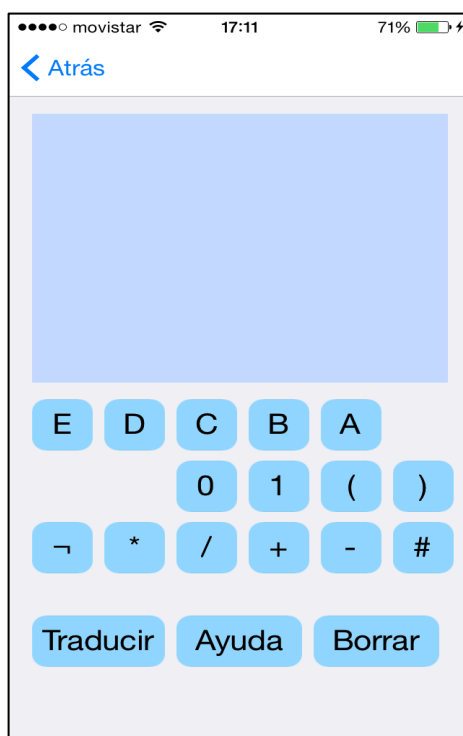


Figura 4-14: Traductor por expresión algebraica

A la hora de la implementación, para la zona superior que hace la función de calculadora se ha utilizado un cuadro de texto para ir representando la expresiones introducidas mediante los botones disponibles. Estos cuadros de textos son modificables en diversos aspectos como son: tamaño, tipo y color de fuente, fondo, etc..

Respecto a los botones, se ha decidido crear un teclado personalizado que según que valores introducen, éstos se han implementado de dos formas, a continuación se especifican cuales son en la figura:

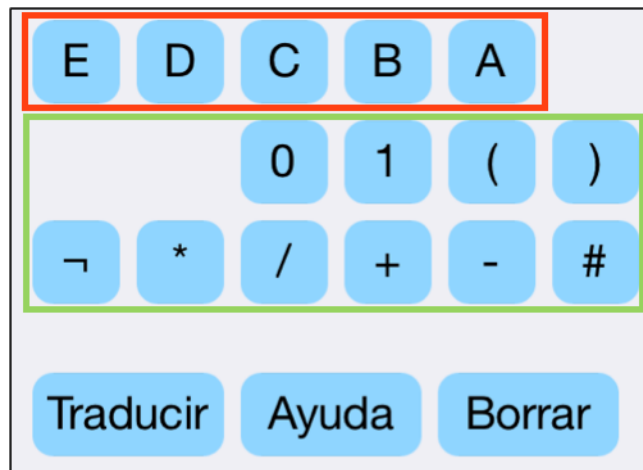


Figura 4-15: Teclado para introducir expresión algebraica

Los marcados con el recuadro rojo son los encargados de introducir las variables elegidas por el usuario. Estos botones aparecen y desaparecen según el número de variables del conjunto. Por otro lado, los que señala el recuadro verde son botones que se muestran siempre independientemente del número de variables del conjunto y son los que tienen asociados los operadores y otros caracteres especiales.

En la parte inferior se han colocado tres botones:

- Traducir: Al pulsar sobre este botón, se realiza la traducción y simplificación de a expresión introducida , mostrándose el resultado en otra nueva pantalla que se explicará próximamente.
- Ayuda: Permite al usuario acceder directamente a la herramienta “Ayuda” y concretamente al menú “Introducir expresiones lógicas”.
- Borrar: Al accionar este botón, permite el borrado del último elemento introducido por el usuario.

4.5.4.3 Traducción por expresiones lógicas

Una vez el usuario acciona el botón “Traducir”, comienza uno de los procesos de mayor complejidad de la aplicación. Se tiene una expresión introducida y el objetivo es que a partir de ella, se obtenga su tabla de verdad, el sumatorio de minitérminos, producto de maxitérminos y por supuesto la función simplificada. Para implementar este proceso se han seguidos esta serie de pasos:

- En primer lugar es comprobar si la función introducida es válida y cumple con los requisitos requeridos para la traducción.
- Utilizar algún método para evaluar una expresión.
- Con el método elegido e implementado hay que evaluar todos los valores de la tabla de verdad para obtenerla.

Estos son los pasos seguidos, no obstante, han sido nombrados con poco detalle. Si se desea conocer con detalle mas el proceso, se puede ver la implementación completa en el manual del programador, en el apéndice A.

Cuando se ya se obtenga la tabla de verdad, se tiene acceso directo al sumatorio de minitérminos ya que es la misma tabla. Respecto a la traducción a producto de maxitérminos es inversa, es decir, los maxitérminos son los valores obtenidos en la tabla de verdad cuyo valor sea un cero.

4.5.4.3 Introducción por tablas

La introducción por tablas es utilizada en los otros tres métodos de escribir la función de entrada: Tablas de verdad, sumatorio de minitérminos y producto de maxitérminos. La representación de las tablas consta de dos columnas, como se ha mencionado en apartados anteriores, la primera esta encabezada por una label o etiqueta que indica el conjunto de variables y la segunda está indicada por una label o etiqueta con la letra “F” de función, que contiene los valores posibles de la misma. Para remarcar las columnas se ha utilizado un cuadro de color para cada una, utilizando la clase UIView que es la encargada del diseño de la vista. Y sobre dicho cuadro se han implementado en la primera columna las etiquetas con todos los posibles valores del conjunto de variables y en la segunda columna los botones táctiles que permitirán cambiar el estado de los valores, estos pueden ser “0”(por defecto), “1” y “X”. Se puede apreciar lo explicado en al siguiente imagen:

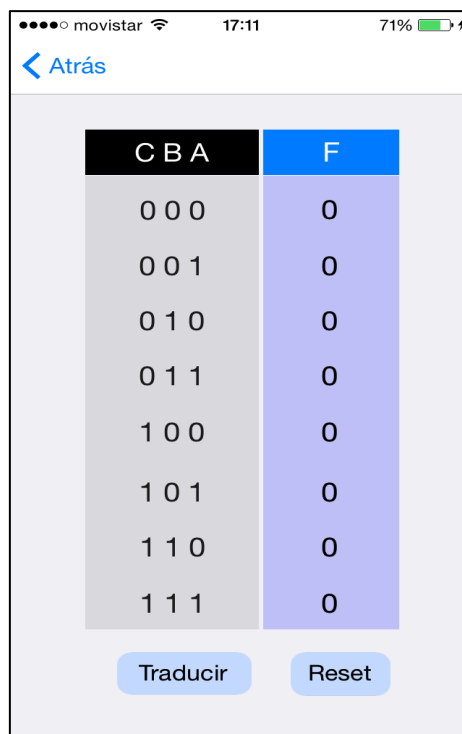
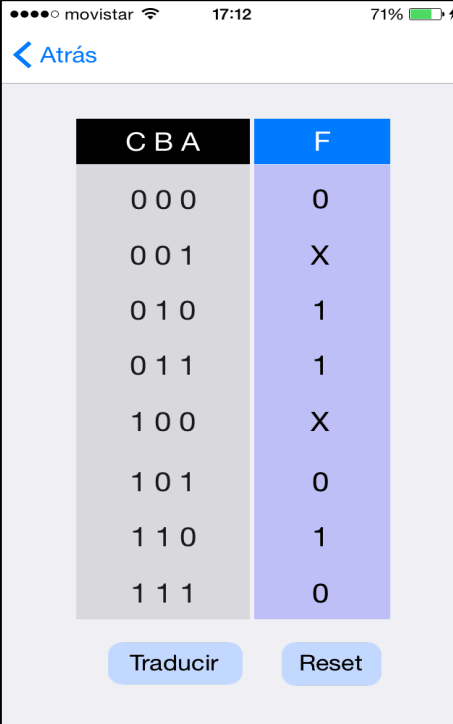


Figura 4-16: Introducción por tablas

Para conjuntos de valores mayores de 3 variables, se ha hecho uso del elemento ScrollView para poder desplazar la pantalla con el dedo y visualizar correctamente todos los valores de la tabla. Mencionar, que los cuadros que delimitan las columnas se han diseñado teniendo en cuenta los diferentes tamaños de pantalla de los dispositivos compatibles con la aplicación.

En la parte inferior de las tablas, se colocan dos botones, “Traducir” y “Reset”. El primero de ellos hará la llamada a las funciones encargadas de la traducción y el botón de “Reset” restablecerá a cero todos los valores de la tabla de la columna de la derecha de la pantalla.

En cuanto al funcionamiento de los botones táctiles que permiten al usuario, introducir los valores de la función que desea, utilizan la misma implementación que la explicada para los botones de cada una de las posiciones del mapa de Karnaugh en la herramienta Karn Auto. A continuación, se muestran diferentes capturas de la introducción por tablas:



C	B	A	F
0	0	0	0
0	0	1	X
0	1	0	1
0	1	1	1
1	0	0	X
1	0	1	0
1	1	0	1
1	1	1	0

Figura 4-17: Introducción valores por tablas

En la introducción por expresión lógica, se evaluaba la expresión introducida para así poder crear la tabla de verdad correspondiente, y a continuación obtener la traducción a sumatorio de minitérminos y producto de maxitérminos. En este caso, se obtiene directamente una de las tres tablas y las restantes se tienen de ésta.

4.5.4 Mostrando resultados en la herramienta Traductor

A la hora de mostrar los resultados de las traducciones, se pensó en crear una interfaz sencilla y a la vez intuitiva. El resultado es el mostrado en la siguiente imagen:

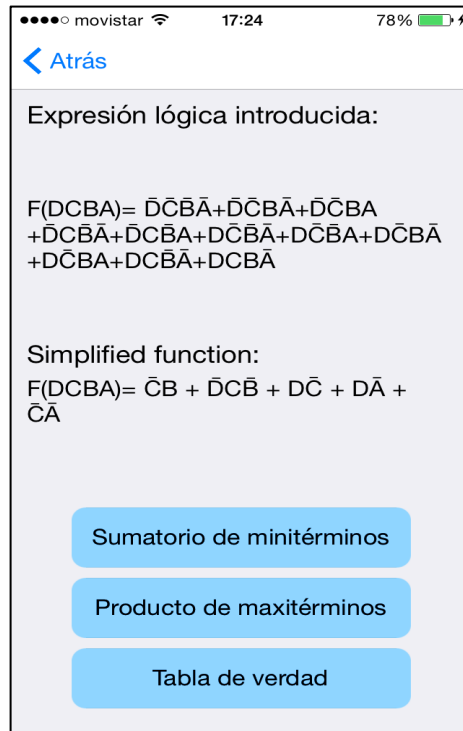


Figura 4-18: Menú de resultados de la traducción

Como se puede observar en la imagen anterior, se muestra en formato texto en la parte superior la expresión lógica introducida a través del método elegido de los disponibles y ya explicados previamente. En la zona central de la pantalla, en este caso también en formato texto, se indica el resultado de la simplificación obtenida.

A la hora de ofrecer al usuario las traducciones resultantes restantes. Se han implementado botones correspondientes a la traducción en formato tabla de verdad, sumatorio de minterminos y producto de maxiterminos respectivamente. El motivo de esta decisión es por un tema de espacio para la correcta visualización. Estos botones permitirán acceder a nuevas pantallas que mostrarán las traducciones en los formatos correspondientes, tal y como se puede observar en la siguiente imagen:

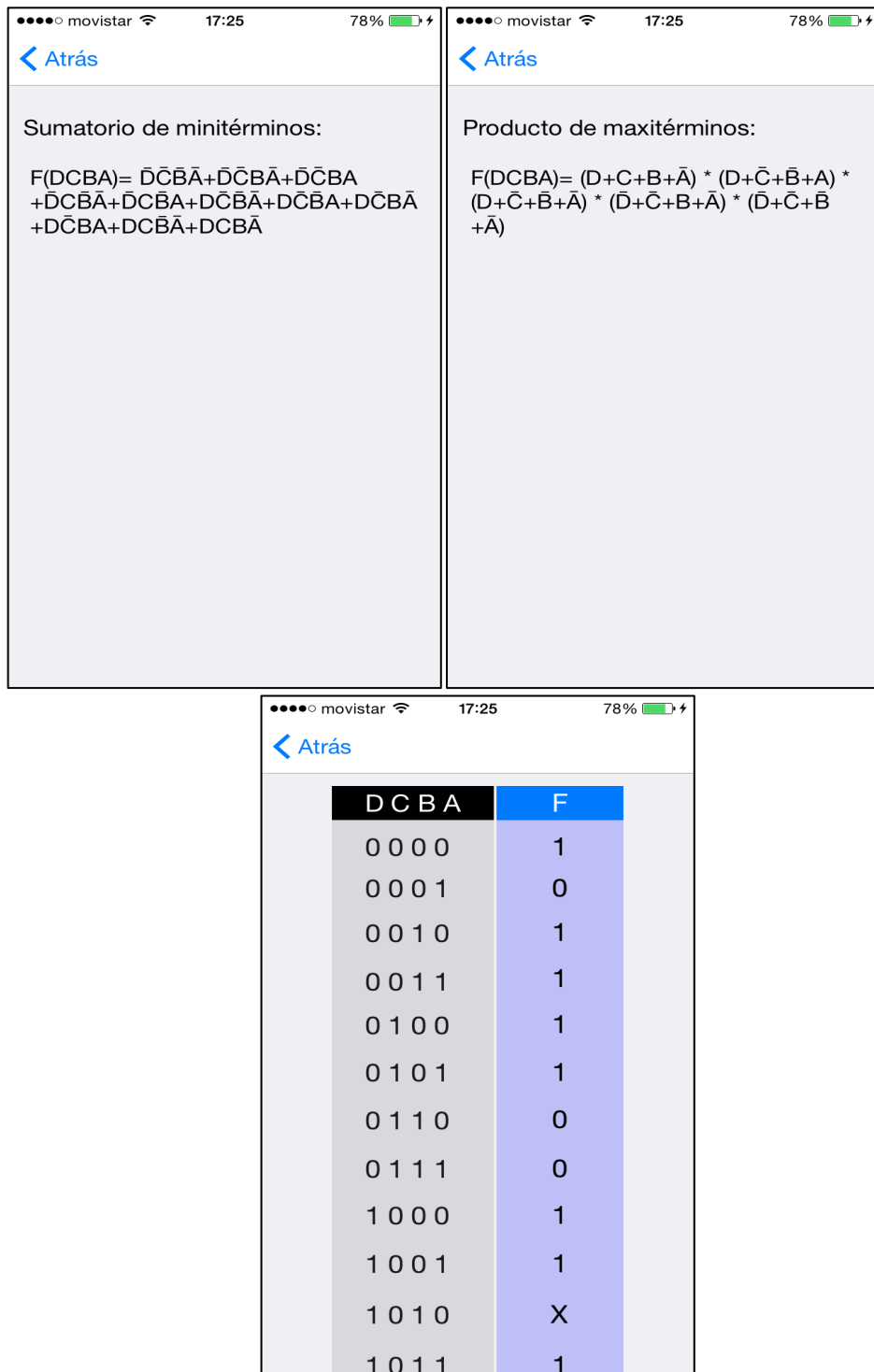


Figura 4-19: Resultados de las traducciones realizadas

En cada una de las pantallas anteriores, es posible volver a la vista anterior, usando el botón “Atrás” y elegir otro forma de visualizar los resultados. Cabe destacar que los cuadros de textos se adaptan en tamaño según la longitud del texto para así mejorar la visualización.

En la realización de la simplificación de la función, hay que utilizar la tabla de verdad para obtener una lista de términos que contenga todos los elementos de la

tabla de verdad cuyo valor sea “1” o “X”. A dicha lista de términos se le aplicará el algoritmo Quine-McCluskey para que devuelva la lista de términos de la función simplificada. El resultado obtenido será el mostrado en la pantalla principal de que muestra los resultados de la traducción.

4.5.5 Problemas encontrados en el desarrollo de la herramienta “Traductor”

El principal problema a solventar en esta herramienta fue el redimensionamiento de las etiquetas, textos y botones de las diferentes vistas. Para ello, en cuanto a los textos y botones se ha usado una herramienta disponible en Xcode en las propiedades de cada una de las ViewControllers utilizadas que recibe el nombre de “Autosizing”. Ésta nos permite ajustar como se quiere redimensionar el tamaño del elemento en cuestión, cuando el tamaño de la pantalla del dispositivo varía. Otro problemas como la resolución de la representación de las tablas ha sido solventado ajustando manualmente la resolución según la dimensión de pantalla en la cual se ejecute la aplicación.

4.5.6 Herramienta conversor

La herramienta conversor es una de las más sencillas de la aplicación tanto en diseño de interfaz como de implementación. En el aspecto gráfico, se trata de una vista con botones y cuatro campos de texto seleccionables. La complejidad que puede llevar esta herramienta es que se un posible “overflow” en alguna de las variables o por la introducción incorrecta de número no soportados por el sistema seleccionado en ese momento.

Respecto a la posible introducción errónea de números, se ha decidido deshabilitar el teclado por defecto mostrado en iOS al seleccionar un cuadro de texto, y al igual que en otras herramientas de la aplicación se ha usado un teclado personalizado. Su funcionamiento es sencillo, y para facilitar su explicación se muestra la imagen de la apariencia de la herramienta “Conversor”:

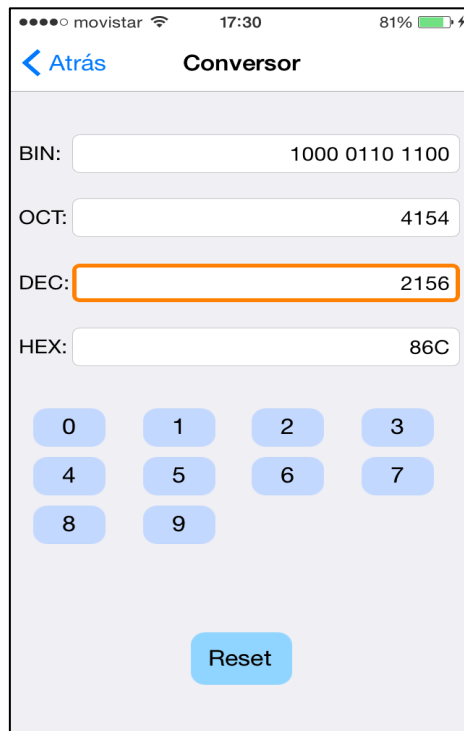


Figura 4-20: Herramienta Conversor

Los cuadros de textos de la parte superior representan los sistemas soportados por la aplicación y se han implementado de forma que sean seleccionables, es decir, que el usuario al pulsar sobre cada uno de ellos, estos queden seleccionados mostrando un borde de color naranja. Con cada selección los botones táctiles de la zona inferior aparecen y desaparecen según sea el sistema elegido al pulsar sobre su cuadro de texto correspondiente., estos botones tienen asignados valores del 0 al 9 y de la A al F. Por defecto, al acceder a esta herramienta se encuentra seleccionado el sistema binario.

En relación a solventar el problema de “overflow”, se ha implementado un código que realiza las conversiones a cada uno de los sistemas sin importar el sistema de entrada. No obstante, en tiempo real se va calculando la conversión mientras se está introduciendo los valores en caso de que en alguno de los sistemas se vaya a producir “overflow” no se permite introducir más valores por teclado. Además se ha establecido un máximo de caracteres en cada cuadro de texto correspondientes a cada sistema. En el caso del sistema binario si el resultado no puede verse correctamente, se ha activado la opción de editar para que el usuario puede desplazarse por el cuadro de texto haciendo uso de un cursor y permitiendo la copia del resultado obtenido en este sistema.

4.5.7 Manuales de ayuda

Esta sección de la aplicación incluye manuales de ayuda de cada una de las herramientas de las que dispone de la aplicación. Su implementación ha sido similar a la herramienta “Variables”, ya que se muestra una lista pero en este caso su uso es más sencillo. Cada celda contiene los manuales de las herramientas que indican en su propio título, al pulsar sobre ellas muestra el manual correspondiente constituido por textos e imágenes estáticas que no varían su apariencia, únicamente se ajustan a la resolución de la pantalla del dispositivo en el cual se ejecute la aplicación. Por lo que se puede decir, que esta parte no ha llevado gran dificultad en su implementación y no se añadirá más detalles de los ya mencionados.

A continuación, se puede observar en la siguiente imagen varias capturas de dicho manual de ayuda:

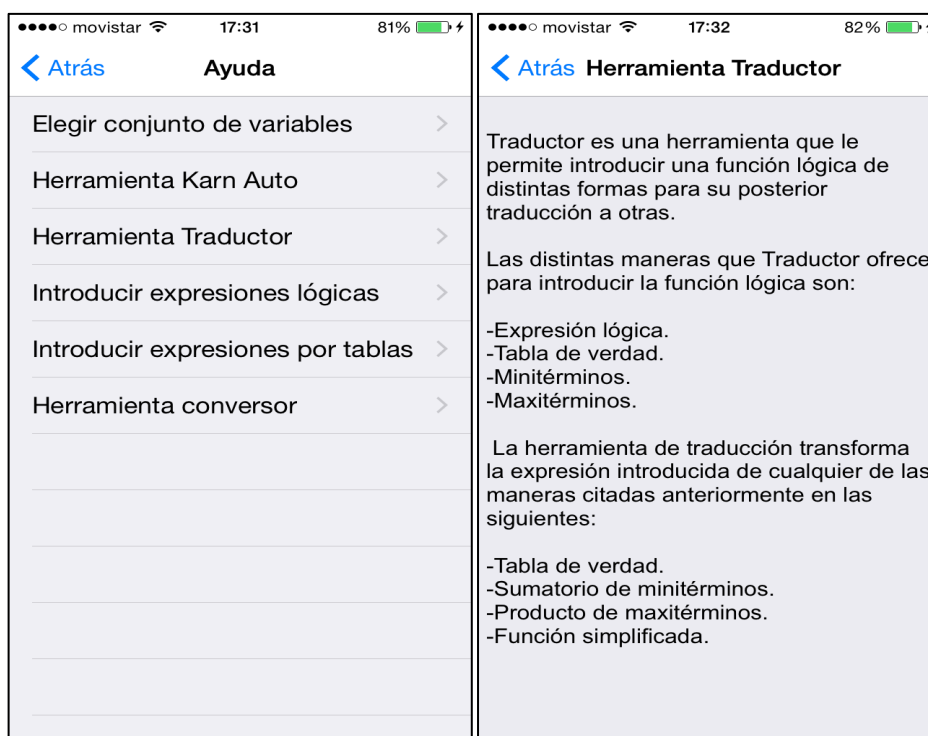


Figura 4-21: Menús de ayuda (I)

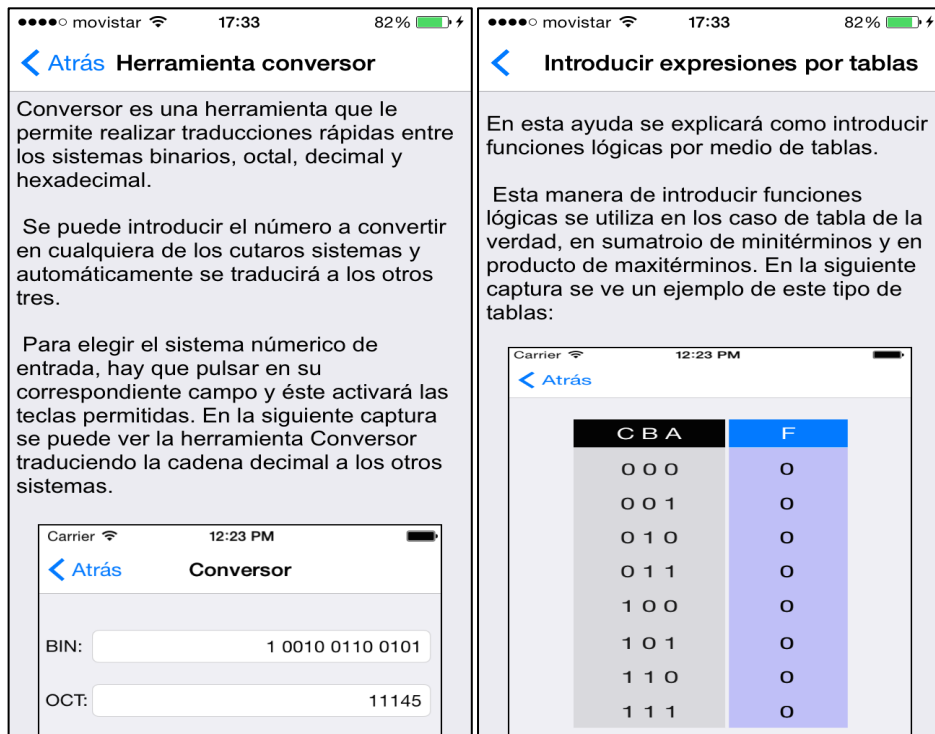


Figura 4-22: Menús de ayuda (II)

4.5.8 Página de información sobre la aplicación

Por último para finalizar este apartado, se ha diseñado una imagen la cual muestra información básica del proyecto. Ésta se encuentra presente como “Launch imagen” o imagen de lanzamiento de la aplicación, es decir, cada vez que se ejecuta la aplicación se muestra la misma, además también se puede ver al finalizar el la última imagen de la herramienta “Tutorial”. Este diseño se ha utilizado para seguir la estética de las anteriores aplicaciones del DSLab. Se muestra la misma a continuación:



Figura 4-23: Información de la aplicación

4.6 Algoritmo Quine-McCluskey

4.6.1 Introducción

El algoritmo de Quine-McCluskey es un algoritmo que puede ser utilizado para simplificar funciones lógicas haciendo uso de los mapas de Veitch-Karnaugh.

Los mapas de Karnaugh son un método gráfico para la simplificación de funciones que resulta de gran efectividad. No obstante, existen ciertos problemas que hay que tener en cuenta:

- Si se desea utilizar para un número de variables mayores que 5, comienza a crecer la complejidad que resulta complicado de aplicar.
- La visión y habilidad del usuario cobra mucha importancia para encontrar la simplificación mínima.
- Además presenta gran dificultad para ser implementado computacionalmente.

Haciendo uso del algoritmo Quine-McCluskey se obtienen las siguientes ventajas:

- Su aplicación es sencilla para la simplificación de funciones de mas de 4 variables.

- Al tratarse de un método sistemático y automatizado, no es necesario que el usuario posea habilidad para la simplificación.
- Su forma modular lo hace más sencillo de implementar en lenguajes computacionales.

Por ello, a pesar de que el algoritmo por mapas de Veitch-Karnaugh es por el cual la aplicación recibe su nombre, realmente el algoritmo usado para realizar las simplificaciones en el interior de la aplicación, es el de Quine-McCluskey. No obstante, dado al gran reconocimiento en el mundo de la electrónica que posee Karnaugh se ha decidido que la aplicación reciba el nombre de Karn Map haciendo referencia al autor de estos mapas. Destacar que la parte gráfica de la aplicación hace uso de la representación de Karnaugh.

4.6.2 Funcionamiento

El funcionamiento del algoritmo Quine-McCluskey consiste principalmente en dos pasos:

- En primer lugar se deben encontrar todos los implicants primos de la función que se quiere simplificar.
- A continuación, haciendo uso de los implicants primos obtenidos previamente, se deben buscar los implicants primos esenciales que sean éstos, los mínimos necesarios para representar la función, la cual se desea minimizar.

La explicación realizada se ha hecho de la forma más reducida y sencilla, ya que en los siguientes apartados se explicarán los dos pasos de forma detallada, además de acompañar dichas explicaciones con un ejemplo. Para así, comprender el algoritmo correctamente.

4.6.2.1 Paso 1: Encontrar implicants primos

Para comenzar, se introducen en una tabla todos los minitérminos, es decir, los elementos de la función cuyo valor sea “1” o “X” en la tabla de verdad. Debido a que es lo que se busca la hora de realizar la simplificación, combinándose la redundancias con los minitérminos y así obtener la función lo más simplificada posible.

Los minitérminos de la tabla se agrupan según el número de unos que tenga cada uno de ellos. Una vez agrupados los elementos en la tabla, el siguiente proceso será juntar los términos de grupos que sean contiguos. Agrupándose todos los términos que difieran en un solo bit, de esta forma obteniendo como resultado un términos con un guión (-) en ese bit y el resto de bits se quedan igual. Es necesario agrupar todas las combinaciones posibles de términos que difieran un bit e indicar que minitérminos son el resultado de cada agrupación.

El proceso descrito se realizará con los elementos obtenidos hasta que no sea posible obtener nuevos elementos de cada grupo contiguo. Cada elemento final, el cual no ha podido ser combinado con ninguno, debe ser marcado.

4.6.2.1.1 Ejemplo del paso 1:

La función de ejemplo mostrada a continuación posee la siguiente tabla de verdad. Dicha función es la que se desea simplificar:

DCBA	F	DCBA	F
0000	0	1000	0
0001	0	1001	1
0010	1	1010	0
0011	1	1011	1
0100	0	1100	0
0101	X	1101	1
0110	0	1110	X
0111	1	1111	1

Tabla 4-1: Tabla de verdad de la función de ejemplo

A continuación se agrupan los minterminos con “1” y “X” en la siguiente tabla según el número de unos que tenga cada término:

Número de unos	minitérmino	DCBA
1	2	0010
	3	0011
2	5	0101
	9	1001
	7	0111
3	11	1011
	13	1101
	14	1110
	15	1111

Tabla 4-2: Primera tabla de primos implicantes

En el siguiente paso se agrupan todos los términos, los cuales sólo difieren en un bit, entre grupos contiguos.

Minitérminos	D C B A
1, 3 (Elemento 1)	0 0 1 -
3, 7	0 - 1 1
3, 11	- 0 1 1
5, 7	0 1 - 1
5, 13	- 1 0 1
9, 11	1 0 - 1
9, 13	1 - 0 1
7, 15	- 1 1 1
11, 15	1 - 1 1
13, 15	1 1 - 1
14,15 (Elemento 2)	1 1 1 -

Tabla 4-3: Segunda tabla de primos implicantes

Se repite el mismo paso y los elementos no utilizados en ninguna de las agrupaciones, son marcados en la tabla superior. Para este caso, han sido marcados dos elementos.

Minitérminos	D C B A
3, 7, 11, 15 (Elemento 3)	- - 1 1
5, 7, 13, 15 (Elemento 4)	- 1 - 1
9, 11, 13, 15 (Elemento 5)	1 - - 1

Llegado el momento en el cual no es posible realizar ninguna agrupación más, se marcan 3 elementos en la tabla superior. Siendo la salida de este paso los 5 elementos marcados en las tablas. A continuación, se continúa con el paso de 2 del algoritmo.

4.6.2.2 Paso 2: Encontrar implicantes primos esenciales.

Marcados todos los grupos en el paso anterior, los cuales no hay uno mayor que los contenga, es hora de comenzar este segundo paso.

Para obtener la simplificación de la función dada, se deben sólo seleccionar algunos de los conjuntos, de esta forma cubrir todos los términos en los que la

salida de función posee valor “1”. Generándose una tabla con los conjuntos, marcándose todos los minitérminos que cubre cada conjunto.

Para realizar la selección de los conjuntos, en primer lugar se elegirán todos aquellos elementos esenciales que son los únicos que incluyen alguno de los minitérminos. Esto es así, ya que los términos que solo están incluidos por un conjunto deben estar en la simplificación.

Finalmente se eligen los conjuntos mínimos restantes para añadir los que quedan si no se han añadido todos, una vez elegidos los esenciales previamente. Es posible obtener varias simplificaciones mínimos diferentes.

4.6.2.2.1 Ejemplo del paso 2:

En este paso se crea una tabla que contendrá todos los elementos que forman la salida del paso número uno explicado anteriormente, e indicar que minitérminos cubren cada uno de ellos. Para los términos que tuviesen redundancias, no es necesario tenerlos en cuenta, ya que se seleccionarán si se considera útil.

Elemento	2	3	7	9	11	13	15
1	X	X					
2							X
3		X	X		X		X
4			X			X	X
5				X	X	X	X

Tabla 4-4: Tabla para buscar los primos implicantes esenciales.

En la tabla superior se han indicado con color amarillo, los minitérminos que únicamente están incluidos por uno de los elementos de la tabla. Siendo éstos, los minitérminos 2 y 9. Por lo que los elementos 1 y 5 deben estar presentes en la función simplificada.

Se observa que al añadir a la función simplificada los elementos mencionados 1, y 5. Por lo que todos los minitérminos que se tendrán en la función son: 2, 3, 9, 11, 13 y 15.

Por lo que, sólo queda el minitérmino 7 por seleccionar. Hay dos posibilidades coger el elemento 3 o el 4, al ser ambos del mismo tamaño, es posible escoger cualquiera de los dos para la simplificación. Siendo las dos opciones posibles de conjuntos de elementos:

- Elementos 1, 5 y 3, que equivale a $\neg D * \neg C * B + D * A + B * A$
- Elementos 1, 5 y 4, que equivale a $\neg D * \neg C * B + D * A + C * A$

Se muestra gráficamente a continuación, una de las soluciones obtenidas en la herramienta “Karn Auto” de la aplicación:

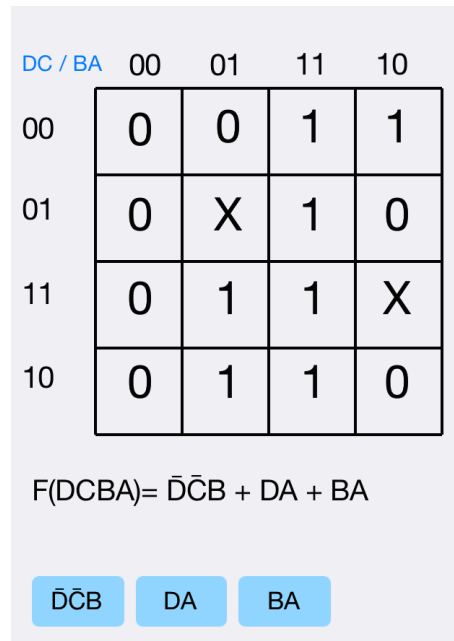


Figura 4-24: Comprobación con herramienta Karn Auto

4.6.3 Desarrollo

El algoritmo de Quine-McCluskey tiene gran popularidad para la implementación de éste en diversas herramientas. Por ello, en internet se encuentran disponibles diversas versiones para descargar y hacer uso de ellas. No obstante, no ha sido posible encontrar una versión para el lenguaje de programación Objective-C, por lo que ha sido necesario la traducción desde Java.

En todo momento se ha buscado que la velocidad de ejecución del algoritmo sea óptima para que la experiencia de usuario sea óptima. Por ello, se ha decidido utilizar la versión del algoritmo utilizada en la aplicación “Karn Map” para Android, realizando previamente la traducción a Objective-C para la versión iOS. Descartándose implementar desde cero la función, ya que los resultados obtenidos podrían ser correctos pero el tiempo de ejecución no sería tan optimizado.

En cuando a la implementación del algoritmo, al igual que en la versión Android, el código original de la página “literateprogramas.org”, no aceptaba la introducción del valor “X” en la creación de términos. Por lo que se realizaron los cambios adecuados, para que esta mejora también estuviera presente tanto en la versión iOS como en Android. Con la introducción de esta mejora, no se vio alterado el tiempo de ejecución por lo que se consiguió disponer de un algoritmo mucho más completo.

4.7 Diagrama de clases



Figura 4-25: Diagrama de clases

4.7.1 Clases principales

- **KMAppDelegate.m:** La clase " AppDelegate " en el SDK de iPhone es la función de delegado que controla la interfaz del programa. El lenguaje de programación iPhone crea un delegado de forma predeterminada al configurar un proyecto en Xcode. Durante el proyecto se ha usado esta clase para establecer la orientación soportada por la aplicación en todas las ventanas de la aplicación y la carga de los conjuntos por defecto si es la primera vez que se ejecuta la aplicación.
- **KMNavigatorViewController.m:** Permite el manejo de las vistas de toda la aplicación, es decir, su función es la de guiar en la interacción al usuario y poder acceder a la vista que se desea visualizar.
- **KMViewController.m:** Proporciona el menú principal y verifica si es la primera vez que se ejecuta la aplicación o bien se cargan los datos de usuarios almacenados por la clase UserDefaults.
- **KMTutorialViewController.m:** Muestra las imágenes estáticas que forman parte del tutorial y así ayudar al usuario a manejar la aplicación. Esta clase se encarga de gestionar el movimiento entre las distintas páginas.
- **KMVariablesTableViewController.m:** Su función es la de mostrar la lista de conjuntos disponibles para ser seleccionado en la aplicación.
- **KMVariableDatosViewController.m:** Permite la edición del conjunto de variables que el usuario haya seleccionado.
- **KMAñadirConjuntoViewController.m:** Se encarga de añadir un nuevo conjunto personalizado por el usuario a la lista.
- **KMKarnAutoViewController.m:** Ofrece el menú de la herramienta "Karn Auto" para seleccionar el número de variables entre 3 y 5. Además permite la posibilidad de ir a al menú de selección de variables.
- **KMKarnAuto3VariableViewController:** Clase encargada de la simplificación de funciones lógicas por mapa de Karnaugh de 3 variables.
- **KMKarnAuto4VariableViewController:** Clase encargada de la simplificación de funciones lógicas por mapa de Karnaugh de 4 variables.
- **KMKarnAuto5VariableViewController:** Clase encargada de la simplificación de funciones lógicas por mapa de Karnaugh de 5 variables.

- **KMTraductorViewController:** Ofrece el menú de la herramienta “Traductor” para elegir el método de introducción deseado. Además permite la posibilidad de ir a al menú de selección de variables.
- **KMTraductorExpresionLogicaViewController:** Se encarga del método de introducción de funciones mediante una expresión lógica.
- **KMTablaFuncionXVariablesViewController:** Se encarga del método de introducción de funciones eligiendo ceros, unos y equis en una tabla de verdad. La “X” que se muestra en el nombre de esta clase, indica que existe una clase para cada número de variables soportado por la aplicación; que son 3, 4 y 5.
- **KMTablaMinitermsXVariablesViewController:** Se encarga del método de introducción de funciones eligiendo ceros, unos y equis en una tabla de con todos los posibles minitérminos. La “X” que se muestra en el nombre de esta clase, indica que existe una clase para cada número de variables soportado por la aplicación; que son 3, 4 y 5.
- **KMTablaMaxitermsXVariablesViewController:** Se encarga del método de introducción de funciones eligiendo ceros, unos y equis en una tabla de con todos los posibles maxitérminos. La “X” que se muestra en el nombre de esta clase, indica que existe una clase para cada número de variables soportado por la aplicación; que son 3, 4 y 5.
- **KMResultadosTraductorLogicaViewController:** Esta clase proporciona la traducción correspondiente al método de introducción mediante expresión algebraica.
- **KMResultadosTraductorViewController:** Esta clase proporciona la traducción correspondiente al método de introducción mediante tabla de verdad, sumatorio de minitérminos y producto de maxitérminos.
- **KMConversorViewController:** Propociona la herramienta “Conversor” que es la encargada de realizar conversiones entre los sistemas numéricos: binario, octal, decimal y hexadecimal.
- **KMAyudaTableViewController:** Muestra el menú con la lista que dispone de todas las ayudas disponibles.

5 Integración, pruebas y resultados

5.1 Pruebas

En este apartado se van a documentar las pruebas que han sido realizadas para verificar el correcto funcionamiento de la aplicación. La gran mayoría de ellas, se han realizado durante el desarrollo del proyecto, ya que se ha seguido una implementación modular. De esta forma, facilitar la revisión de las aplicación y la introducción de posibles mejoras futuras.

No obstante, la verdadera prueba de funcionamiento de la aplicación se realizará cuando de comienzo la asignatura para la que se ha desarrollado. Siendo la fecha de inicio de la misma aproximadamente en enero de 2016, ya que esta asignatura forma parte del segundo cuatrimestre. Por lo que serán los estudiantes de la escuela, los que hagan uso de la aplicación, si lo desean, para la resolución de problemas.

Sin embargo, al ser una aplicación de pago, se quiere garantizar el correcto funcionamiento de la misma y por ello se han realizado múltiples test, entre ellos, de casos extremos y comprobar que la aplicación responde de forma correcta.

5.1.1 Pruebas en herramienta Karn Auto

Las pruebas sobre la herramienta “Karn Auto” se han realizado para comprobar sus dos funcionalidades principales.

En primer lugar, la encargada de la simplificación. Para ello, se ha recurrido a libros con explicaciones sobre los mapas de Karnaugh y se han seleccionado ejemplos para comprobar si el resultado era el esperado. Para los tres diferentes tamaños de mapas soportados por la aplicación, los resultados han sido satisfactorios.

Por otro lado, la otra funcionalidad principal de esta herramienta, es la de la representación gráfica de las agrupaciones efectuadas en la simplificación y que forman parte del resultado final. En la comprobación previa de la correcta simplificación, también se comprobó que la representación gráfica de las agrupaciones era correcta.

A continuación, se muestran las siguientes imágenes con alguno de los casos extremos comprobados, para verificar que las dos funcionalidades de la herramienta “Karn Auto” funcionaban correctamente. En el primer ejemplo, se comprueba para 4 variables si están seleccionadas las 16 casillas disponibles y el buen funcionamiento del valor “X” (redundancia). En el segundo caso, se verifica si se identifican bien las agrupaciones que se encuentran en lados

opuestos. Por último, se realiza la prueba con agrupaciones opuestas, verticales y horizontales.

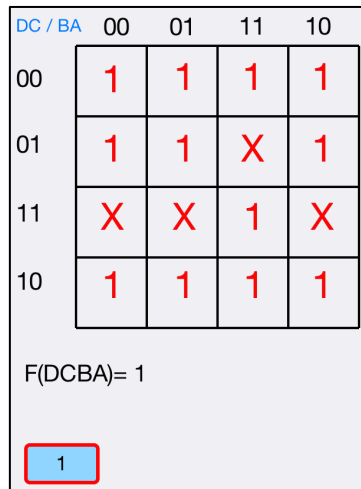


Figura 5-1: Caso extremo (I)

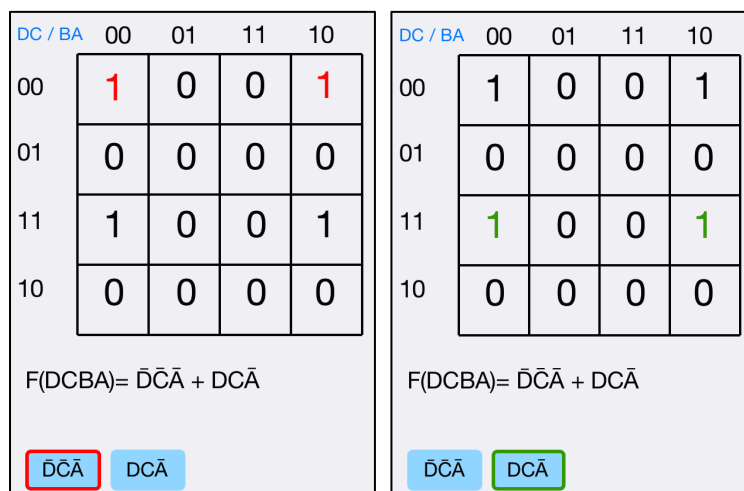


Figura 5-2: Caso extremo (II)

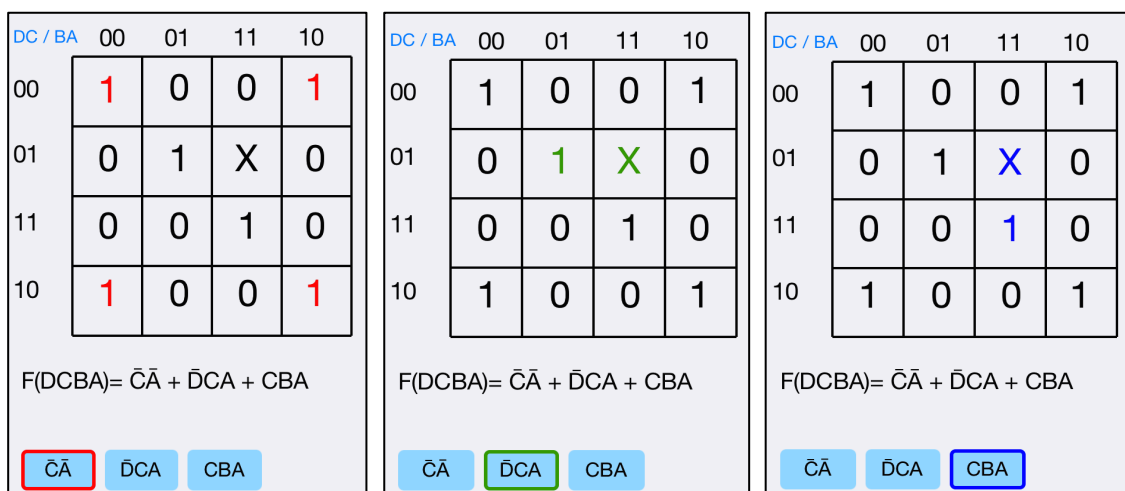


Figura 5-3: Caso extremo (III)

5.1.2 Pruebas en herramienta Traductor

El testeado realizado en esta herramienta siguió el mismo patrón que en la herramienta “Karn Auto”, recurriendo a libros y ciertos ejercicios de las guías disponibles de la asignatura de Circuitos Digitales de la Escuela Politécnica Superior. No se han realizado capturas ya que no se detectó ningún tipo de error por lo que no hay nada reseñable que destacar, únicamente confirmar el correcto funcionamiento de esta herramienta.

5.1.3 Pruebas en herramienta Conversor

En este caso, para verificar el correcto funcionamiento de la herramienta Conversor, se han realizado múltiples conversiones y al mismo tiempo usándose otras calculadoras, obteniéndose el mismo resultado. No se ha detectado ningún tipo de error en esta herramienta.

5.1.4 Pruebas realizadas en otros dispositivos

La batería de pruebas realizadas se han efectuado en todos los dispositivos compatibles con la aplicación, con ayuda de la herramienta iOS Simulator disponible en el entorno de desarrollo Xcode. En todo momento la aplicación se ha adaptado correctamente y no se ha visto afectado el buen funcionamiento de la misma.

5.2 Resultados

En este apartado se ha realizado un seguimiento del tiempo computacional de la ejecución de las funcionalidades de la aplicación. En general, los resultados han sido óptimos y no se han detectado variaciones destacables de tiempo en cuanto al uso con diferente número de variables. Aunque sí es cierto, que es posible detectar pequeños lapsos de tiempo debido a cálculos en tiempo real, en el caso particular del uso de 5 variables para la herramienta “Karn Auto” o en “Traductor” cuando hay una gran cantidad de valores de la función cuyos valores son “1” o “X”, no obstante se trata de casos muy particulares y no afectan al buen funcionamiento de la aplicación. Lo mencionado sucede en cualquier dispositivo ya sea iPhone o iPad.

5.3 Primera publicación

La primera publicación de la aplicación se produjo el día 5 de Junio de 2015. Está se ha realizado desde la cuenta de desarrollador adquirida por el laboratorio DSLab y la aplicación recibió el nombre de Karn Map. Se decidió establecer de pago la aplicación como experimento para comprobar si en la plataforma iOS, ésta primera aplicación del DSLab puede obtener éxito y poder ampliar el abanico en esta plataforma. Se fijado el precio mínimo permitido por Apple de

0,99€, es cierto que esto puede ser un factor que limite el número de usuarios pero se cree que finalmente no será un impedimento para los interesados en su descarga y posterior uso. En todo momento durante el desarrollo se ha dado prioridad a la usabilidad de la aplicación y que sea de las más completas del mercado.

Para realizar la publicación de una aplicación de pago en el Apple Store, siendo este el caso del proyecto, es necesario que una vez implementada y finalizada sea revisada por Apple. A continuación, se va a explicar el proceso que ha sido necesario realizar para publicar la aplicación:

5.3.1 Proceso de publicación: Paso 1

Acceder a <http://itunesconnect.apple.com> e inicia sesión en la interfaz con la cuenta de desarrollador de Apple.

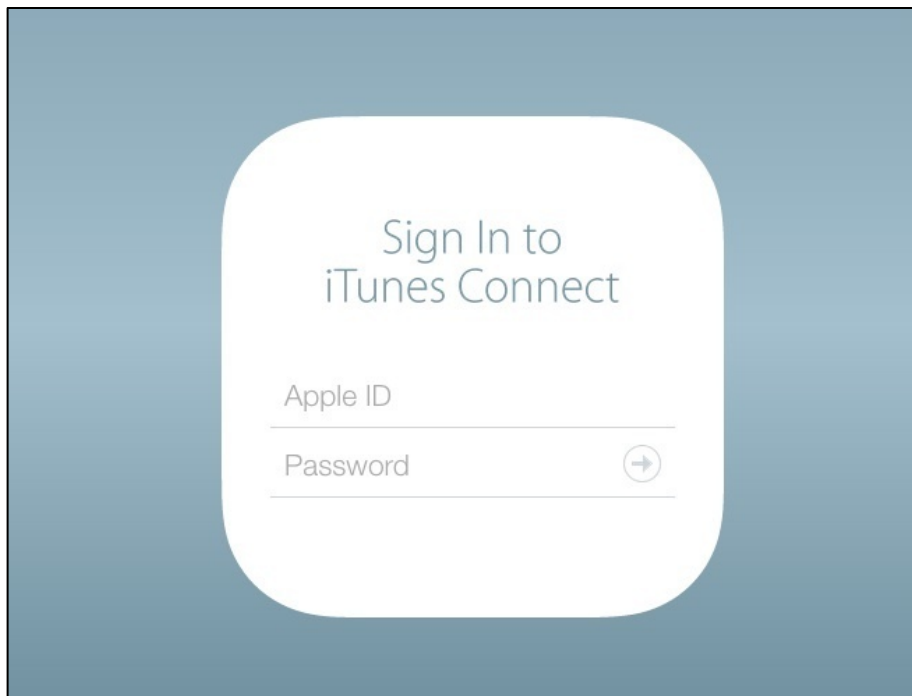


Figura 5-4: Proceso de publicación: Paso 1

5.3.2 Proceso de publicación: Paso 2

Pulsar sobre la sección “Mis Apps”

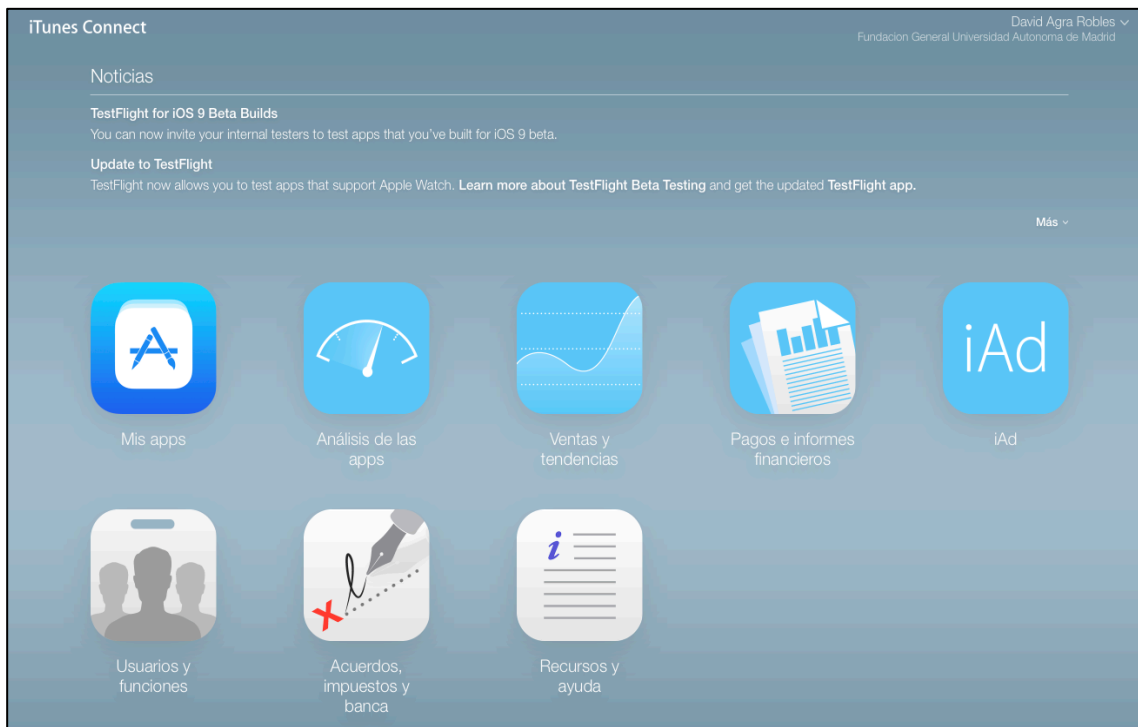


Figura 5-5: Proceso de publicación: Paso 2

5.3.3 Proceso de publicación: Paso 3

Hacer clic en el "+":



Figura 5-6: Proceso de publicación: Paso 3

5.3.4 Proceso de publicación: Paso 4

Si es la primera vez que se envía una aplicación desde la cuenta iTunes Connect, como es el caso, se tiene que elegir el "Nombre de la compañía" ("Company Name") que luego se mostrará en la AppStore, ésta será DSLab UAM.



Figura 5-7: Proceso de publicación: Paso 4

5.3.5 Proceso de publicación: Paso 5

Información sobre la aplicación:

A continuación, comienza el proceso de envío para evaluación de la aplicación.

En este paso se tiene que rellenar los siguientes campos:

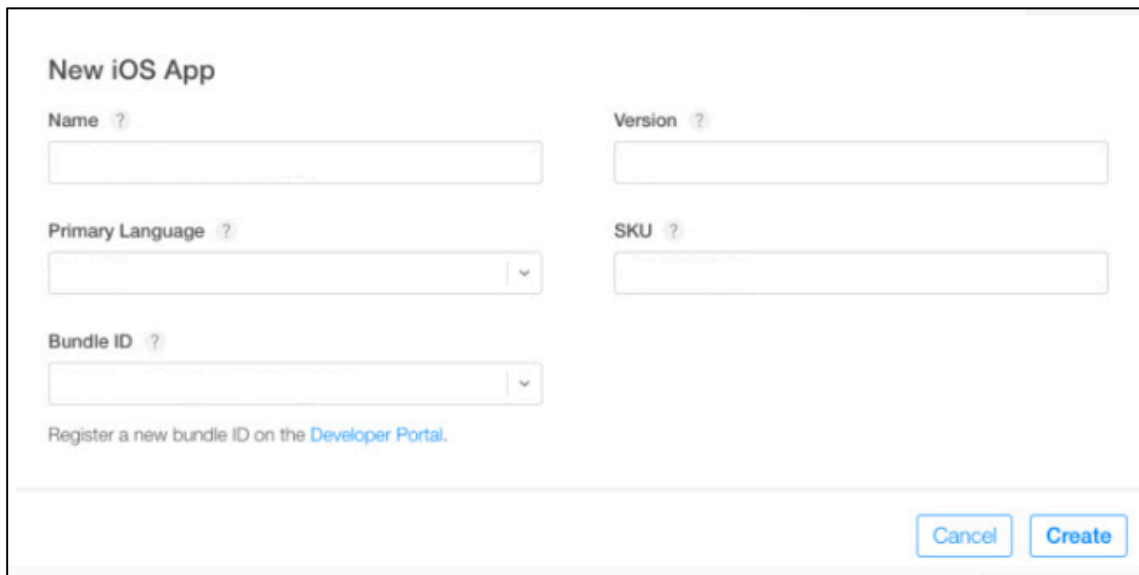
The image shows a screenshot of the "New iOS App" form in iTunes Connect. The form has five main input fields: "Name" with a question mark icon, "Version" with a question mark icon, "Primary Language" with a question mark icon and a dropdown arrow, "SKU" with a question mark icon, and "Bundle ID" with a question mark icon and a dropdown arrow. Below the "Bundle ID" field, there is a link: "Register a new bundle ID on the [Developer Portal](#)." At the bottom right of the form, there are two buttons: "Cancel" and "Create".

Figura 5-8: Proceso de publicación: Paso 5

- Bundle ID. Debe coincidir con el especificado durante el proceso de generación de certificados necesarios para la publicación de la aplicación.
- Número SKU. Se trata de un identificador que el desarrollador elige y se utiliza sólo en iTunes Connect para identificar dicha aplicación.
- Nombre de la aplicación en la AppStore, será “Karn Map”.
- Número de versión de la app, en este caso 1.0.
- Idioma por defecto que se utilizará para describir tu aplicación en la AppStore, se ha decidido como idioma el inglés.

5.3.6 Proceso de publicación: Paso 6

En este paso, se genera un archivo con extensión “.ipa” usando el entorno de desarrollo Xcode. Para cargar la aplicación en la App Store, es necesario descargar y hacer uso de software “Application Loader” propiedad de Apple. Una vez descargado, al ejecutarlo nos pedirá introducir la cuenta de Apple de desarrollador y a continuación se elegirá el archivo “.ipa” generado y se pulsara sobre el botón “Upload”.

Vuelve atrás en iTunes Connect para continuar con el proceso de envío.

5.3.7 Proceso de publicación: Paso 7

Se vuelve de nuevo a la página iTunes Connect para continuar con el proceso de envío. Se seleccionarán en primer lugar las capturas de pantalla y después la información de la versión de la aplicación que se desea publicar.

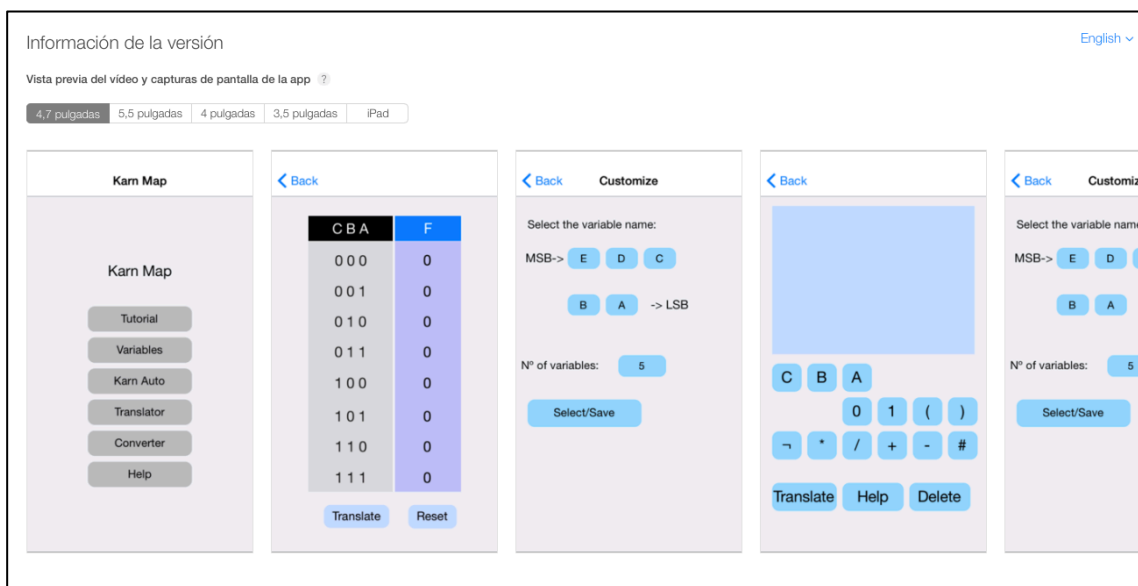


Figura 5-9: Proceso de publicación: Paso 7 (I)

Una vez elegidas las capturas de pantalla, se rellenan los datos sobre la versión:

The screenshot shows the 'New App' submission form in App Store Connect. The form is titled 'Nombre' and contains the following fields and content:

- Nombre:** Karn Map
- Descripción:** Karn Map is an educational application which offers a wide range of tools for the simplification of logic functions. For functions from 3 to 5 variables, it has the automatic simplification and graphic option with Veitch-Karnaugh maps. It also remains possible the simplification of functions from 3 to 5 variables introduced for logical expression, dialing on a real table, choosing minterms or maxterms. It also includes a binary-octal-decimal-hexadecimal converter. It contains a handbook of Veitch-Karnaugh maps and a help menu designed to guide users. Application offline. Contents:
 - Simplification through Veitch-Karnaugh maps.
 - Simplification up to 5 variables.
 - Truth table
 - Summation of minterms
 - Product of maxterms
 - Logical expressions.
- Palabras clave:** Karn Map, Veitch-Karnaugh, Karnaugh, binary, octal, decimal, hexadecimal, maps, map, Truth table,
- URL de soporte:**
- URL de marketing:**
- URL de la política de privacidad:**

3163

Figura 5-10: Proceso de publicación: Paso 7 (II)

- **Descripción:** La descripción de la aplicación que aparecerá en la App Store (máximo 4000 caracteres).
- **Palabras clave:** se trata de las palabras clave que permitirán a la gente encontrar la aplicación de manera más fácil (máximo 100 caracteres).
- **Dirección email de contacto:** este es el email que se mostrará en la App Store.
- **URL de soporte:** Se rellena con la dirección de tu sitio web.
- **URL de Marketing:** Se rellena con la dirección de tu sitio web.
- **URL de la política de privacidad:** No es útil rellenarla en la mayoría de los casos.

5.3.8 Proceso de publicación: Paso 8

Información general de la App

Rellena la información general sobre la aplicación en la App Store:

Información general de la app

Icono de la app ?

DSLab 2015

Información de contacto del representante comercial ?

Muestra la información de contacto del representante comercial del App Store de Corea.

Fundacion General Universidad Autonoma de Madrid

Nombre Apellidos

Calle Einstein (Pabellon C 2 P1ta, Cr De Colmenar)

13 - Km 15

Madrid MADRID

28049 Spain

Teléfono Correo electrónico

Archivo de cobertura de la app de encaminamiento ?

Seleccionar archivo

(Opcional)

Figura 5-11: Proceso de publicación: Paso 8

- App Icon: El icono de la aplicación de 1024 x 1024 píxeles.
- Copyright: El derecho de autor para esta aplicación, que se mostrará en la App Store.
- Categoría (Primaria): Esta es la categoría en la que tu solicitud se publicará en la App Store. Para nuestra aplicación se ha seleccionado la categoría “Educación”.
- Información de contacto: Aquí se rellenaran los datos de contacto del representante de la aplicación.

5.3.9 Proceso de publicación: Paso 9

Revisión de la aplicación y la información de versión de lanzamiento:

Información de Revisión de apps

Datos de contacto ?

Nombre David Apellido Agra Robles

Número de teléfono +34679687303 Correo electrónico davidagra123@gmail.com

Cuenta demo ?

Nombre de usuario Contraseña

Notas ?

4000

Figura 5-12: Proceso de publicación: Paso 9

En la sección “Información de Revisión de apps”, se introduce:

- Información de Contacto: tu nombre, dirección de correo electrónico y número de teléfono, en caso de que Apple deba ponerse en contacto contigo en relación a la revisión de tu aplicación.
- Nota (opcional): Añadir notas sobre la aplicación, como puede ser, notificar al usuario que la aplicación no necesita de conexión a internet para poder funcionar.
- Cuenta Demo (opcional): No es necesario llenar el campo en la mayoría de los casos.

5.3.10 Proceso de publicación: Paso 10

Karn Map - Derechos y precios

Selecciona la fecha de disponibilidad y la franja de precios para la app.

Fecha de disponibilidad: 05/May, 21, 2015

Franja de precios: Choose

Fecha de entrada en vigor: Choose

Fecha de finalización: Choose

Franja de precios	Fecha de entrada en vigor del precio	Fecha de finalización
Tier 1	Existing	None

Descuento para instituciones educativas

A menos que selecciones territorios concretos, tu app se venderá en todos los territorios del App Store del mundo.

[Gestionar los ajustes de descarga de iCloud para esta app](#)

Cancel Save

Figura 5-13: Proceso de publicación: Paso 10

A continuación, en la pantalla principal de la aplicación en la página “iTunes Connect”, se hace click sobre “Precio” y se decidirá la fecha de publicación del envío y su precio en la App Store, si se desea que la aplicación se publique después de la aprobación por parte de Apple, deja que la fecha de la publicación sea la fecha actual. De lo contrario, se escoge una fecha futura. Si se desea que la aplicación sea descargada gratuitamente en la App Store, se selecciona la opción “Free” en la lista de precios, de lo contrario como es nuestro caso se selecciona el precio correspondiente que se mostrará en tu moneda, 0,99€. En la imagen mostrada anteriormente, la franja de precio correspondiente a la seleccionada, será la que recibe el nombre de “Tier 1”.

5.3.11 Proceso de publicación: Paso 11

En este último paso, se completará la información necesaria para la revisión. Una vez se pulse sobre el botón “Enviar para revisión” en la parte superior derecha de la pantalla principal de la aplicación de “iTunes Connect”, Apple nos preguntará si nuestra aplicación posee publicidad a cual seleccionaremos que no, y el estado de nuestra aplicación pasará a “Waiting for Review. Desde ese momento, Apple informa por correo electrónico de cada cambio de estado. El tiempo medio para que la aplicación esté disponible en la tienda (este es el estado "Ready for sale") es de diez días aproximadamente. A continuación se adjunta, el historial completo del proceso de publicación de la aplicación “Karn Map”:



Karn Map (1.0) - Historial de estado

Estado actual: **Lista para la venta**

Nombre de la app: Karn Map Última modificación: Jun 5, 2015

Fecha	Usuario	Estado
May 21, 2015 14:03	davidagra123@gmail.com	En preparación para el envío
May 21, 2015 15:56	davidagra123@gmail.com	Esperando revisión
June 01, 2015 14:40	Apple	En revisión
June 01, 2015 15:57	Apple	Procesando para el App Store
June 01, 2015 16:01	Apple	Pendiente de contrato
June 05, 2015 07:09	Apple	Lista para la venta

Done

Figura 5-14: Proceso de publicación: Paso 11

En un futuro, entregada la memoria y realizada la lectura del PFC se quiere seguir añadiendo versiones, para introducir mejoras como nuevas funcionalidades o bien para solucionar errores encontrados o reportados por los usuarios.

Tal y como se ha comentado en apartados previos, se espera que una vez de comienzo la asignatura para la cual has sido desarrollada esta aplicación, se produzca un aumento de las descargas, ya que se notificará a los estudiantes la existencia de la misma.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Los objetivos requeridos al comienzo del desarrollo del proyecto han sido conseguidos. Estos se enumeran a continuación:

- Dar a conocer con mayor detalle las posibilidades que ofrece Karnaugh para el mundo de los circuitos electrónicos.
- Se ha incluido un tutorial teórico para familiarizar al usuario con la simplificación gráfica de funciones que usa los mapas de Karnaugh.
- Se ofrece al usuario una completa herramienta que permite que gráficamente se pueda realizar la simplificación con mapas de hasta 5 variables. Además de proporcionar la identificación visual de los términos que constituyen la expresión final simplificada.
- La aplicación dispone de la herramienta alternativa “Traductor” que permite la usuario introducir funciones de forma algebraica, por tabla de minitérminos y de maxitérminos. Todos estos métodos de entrada posee un límite de 5 variables.
- Para ofrecer una aplicación de las más completas del mercado, se ha implementado un conversor de los sistemas más utilizados: binario, octal, hexadecimal y decimal.
- Se ha verificado la perfecta compatibilidad con todos los dispositivos de Apple que posean como mínimo la versión iOS 7.
- En todo momento se ha intentado mantener el estilo de las aplicaciones del DSLab, siendo la primera aplicación publicada en iOS del departamento. Se ha incluido información del mismo al comienzo de la ejecución y en el tutorial.
- Se ha realizado un estudio del estado del arte y una comparativa de las aplicaciones ya disponibles en el mercado.
- Se ha priorizado la fiabilidad, sencillez y fluidez en el desarrollo de la aplicación.
- Publicación en App Store.

Durante el desarrollo del proyecto se han adquirido conocimientos y habilidades:

- Programación en Objective-C
- Entorno de desarrollo Xcode e iOS Simulator
- Revisión con gran detalle de la asignatura de Circuitos Electrónicos Digitales.
- Diseño de aplicaciones móviles
- Publicar aplicaciones en el App Store de Apple
- Establecer una organización y planificación de un proyecto.
- Betatester, para la detección de errores y posibles mejoras

6.2 Trabajo futuro

Esta aplicación se ha basado en la ya existente en Android, no obstante se han incluido mejoras, facilitando su usabilidad y diseño. Como cualquier aplicación, esta abierta a mejorar y se desea un buen mantenimiento de la misma.

Esta aplicación ha sido desarrollada desde cero, y la idea principal es que se continúe mejorando y manteniendo en el futuro. Algunas de las mejoras que se podrían hacer a lo que ya hace esta aplicación son:

- Funcionamiento con más de 5 variables sin que perjudique a la velocidad de ejecución.
- Incluir nuevos idiomas para internacionalizar la aplicación.
- Soportar el uso de la aplicación en orientación tanto vertical como horizontal.
- Posibilidad de anotaciones en los manuales de ayuda como en el tutorial.
- Aumentar el número de caracteres posibles en las expresiones lógicas y soportar el uso de cursor en la misma para editar o borrar cierto carácter sin que tenga que ser éste el último.

No obstante, cabe la posibilidad de añadir nuevas funcionalidades a la aplicación en futuras actualizaciones:

- Pequeña evaluación en la primera lectura del tutorial para comprobar si el usuario ha comprendido el método de simplificación de funciones lógicas.
- Manuales de ayuda interactivos .

- Reconocimiento y resolución de un mapa de Karnaugh de hasta 5 variables a través de la cámara del dispositivo.
- Apartado de autoevaluación con ejercicios propuestos y con corrección posterior por parte de la aplicación.
- Representación del esquemático de la función con diseño 3D.
- Soporte para nuevos métodos de entrada de funciones.
- Adaptar aplicación al sistema operativo Windows Phone.

MANUAL DEL PROGRAMADOR

En este manual se explicará detalladamente los puntos más complicados a la hora de programar y durante el desarrollo del proyecto

A - Apéndice A

El contenido de este apéndice consiste en la explicación detallada de la implementación de la herramienta “Traductor”, concretamente a la hora de introducir funciones lógicas usando su expresión.

Dicha expresión será introducida por el usuario, utilizando el teclado personalizado implementado en esta herramienta. Por ello, el primer paso es de vital importancia, y es verificar si la expresión es correcta o por si por el contrario no es válida. Para ello, en primer lugar se realiza una comprobación de la misma.

A.1 Comprobación de la expresión

Previamente a la traducción de la expresión, es necesario verificar la validez de la misma, ya que si no lo es, el resto de funciones implementadas pueden dar errores. Por lo que esta función, cobra gran importancia ya que evitará que la aplicación pueda bloquearse cuando el usuario introduce una expresión que no es correcta. No obstante, es importante notificar al usuario el error que contiene la expresión introducida ya que de esta forma puede aprender de los errores y comprender mejor el método de introducción implementado.

El botón “Ayuda” que aparece en la zona inferior de la pantalla de introducción de expresiones lógicas, al pulsar sobre él, se muestra información incluyendo ejemplo de casos correctos e incorrectos. De esta forma se consigue ayudar en todo momento al usuario para familiarizarse con la aplicación.

En esta función de comprobación se verifica si la expresión contiene errores, como pueden ser que la expresión sea vacía, un balanceo incorrectos de paréntesis o bien si existen dos caracteres consecutivos incorrectos. A continuación en la tabla que se muestra, se puede observar todos los errores que pueden ser notificados al usuario por esta función. La notificación de los mismos se realiza mediante mensajes de la clase UIAlertView.

Expresión incorrecta por:	
	Balanceo de paréntesis incorrecto
	Hay dos negadores consecutivos
	Hay dos variables sin operador entre ellas
	Hay dos operadores seguidos
	Hay una variable seguida de un elemento incorrecto
	Hay un paréntesis de cierre seguido de un elemento incorrecto
	El primer caracter es incorrecto
	El ultimo caracter es incorrecto
	Hay un negador seguido de un elemento incorrecto
	Hay un operador seguido de un elemento incorrecto
	Hay dos paréntesis sin nada dentro
	Hay un paréntesis de apertura seguido de un elemento incorrecto
	No se puede hacer la expresión más grande

Tabla 0-1: Errores notificados por la función de comprobación de la expresión lógica

Si la expresión es correcta tras ser evaluada por esta función, se avanza al siguiente paso.

A.2 Necesidad de conversión infijo-postfijo

Normalmente las expresiones las cuales usamos son del tipo infijas, es decir, con los operadores entre los operandos correspondientes. Sin embargo, este tipo de escritura de las expresiones añade gran complejidad para ser evaluadas, ya que no están ordenadas y es necesario tener en cuenta las prioridades de los paréntesis y operadores. Por ello, se hace uso de la notación postfijo. En ésta, los operadores se escriben a continuación de los operandos y no entre ellos como se hace con la notación infija. Se muestran en la imagen diferentes ejemplos de las notaciones que se han comentado:

Notación infija	Notación postfija
$\neg A$	$A\neg$
$A*B$	AB^*
$A*\neg B$	$AB\neg^*$
$\neg A*B$	$A\neg B^*$
$\neg(A*B)$	$AB^*\neg$
$A*B+C$	AB^*C+
$A*(B+C)$	$ABC+^*$
$A*B*C+A*\neg B*\neg C$	$ABC**AB\neg C\neg***+$

Tabla 0-2: Ejemplos de expresiones usando notación infija y postfija

A.3 Algoritmo de conversión infijo-postfijo

En este punto se va a explicar como se realiza la conversión de un sistema a otro. A la hora de desarrollar este algoritmo se han revisado y analizado las diapositivas de la asignatura cursada en el segundo curso de la carrera y que recibía el nombre de Estructura de Datos y Algoritmos.

Cada operador posee una prioridad determinada, como se puede ver en la siguiente tabla:

Función	Prioridad
$()$	0
NOT	1
AND	2
NAND	2
OR	3
NOR	3
XOR	3

Tabla 0-3: Prioridades en la conversión infijo-postfijo

La implementación seguida para el desarrollo del algoritmos es la siguiente:

En primer lugar se declara una pila en la cual se van almacenando los operadores. Éstos van siendo leídos de la pila y analizados de uno en uno. Si el valor de éste, es un operando, se imprimirá en la traducción parcial postfija directamente.

Por otro lado, si el operador leído es propiamente un operador, se irán retirando todos los operadores de mayor o igual prioridad que se encuentren almacenados en la pila y se concatenarán a la expresión traducida. Ésta extracción de operadores se parará cuando uno de los almacenados en la pila, su prioridad sea menor y se introduce en la pila este nuevo operador.

Si durante la lectura de la expresión, se encuentra un paréntesis de apertura, éste es introducido en la pila y continúa el algoritmo. Una vez sea encontrado el de cierre, se extraen todos los valores hasta el de apertura y son introducidos en la expresión de salida.

Finalizada la lectura de la expresión infija se extraen e introducen en la expresión resultante todos los operando que quedan almacenados en la pila.

Para mejorar la explicación del algoritmo, se muestra el proceso gráficamente en la siguiente tabla con la expresión $A+B*(C+D)*E+F$:

Símbolo	Traducción parcial	Pila de operadores
A	A	
+	A	+
B	AB	+
*	AB	+*
(AB	+*(
C	ABC	+*(
+	ABC	+*(+
D	ABCD	+*(+
)	ABCD+	+*
*	ABCD+*	+*
E	ABCD+*E	+*
+	ABCD+*E*+	+
F	ABCD+*E*+F	+
EOS	ABCD+*E*+F+	

Tabla 0-4: Ejemplo de funcionamiento del algoritmo de conversión infijo-postfijo

Siendo el resultado final $ABCD+*E*+F+$. Si durante la conversión se produce algún tipo de error se avisará al usuario de ello usando un mensaje de la clase `UIAlertView`.

A.4 Evaluación de expresiones postfijo

Realizada la conversión, a continuación se aplicará el proceso de evaluación de la expresión. En este paso también se ha hecho uso de pilas.

El funcionamiento de algoritmo de evaluación es el siguiente:

En primer lugar se van leyendo cada uno de los elementos que forman la expresión postfijo, hasta llegar al momento en que no queda ningún elemento pendiente por leer. Según el tipo de elemento leído, el algoritmo realiza una acción u otra:

- Si se trata de un operando, se introduce en la pila de operandos.
- Si el elemento leído es un operador, se extraen los últimos dos operandos de la pila operandos y se almacena el resultado en la pila de operandos.
- Si es un negador sólo se extrae para operar el último operando de la lista.

Finalizada la lectura de los elementos de la expresión, el resultado es el que se encuentra almacenado en la pila. A continuación, se muestra en la siguiente tabla, un ejemplo del funcionamiento de este algoritmo de evaluación para la expresión infija: $-(1*0*0+1)+(1*-0)$ siendo el resultado de la conversión $10*0*1+-10-*+$.

Símbolo	Op1	Op2	Valor	Pila de operandos
1				1
0				1,0
*	1	0	0	0
0				0,0
*	0	0	0	0
1				0,1
+	0	1	1	1
-	1		0	0
1				0,1
0				0,1,0
-	0		1	0,1,1
*	1	1	1	0,1
+	0	1	1	1
EOS				

Tabla 0-5: Ejemplo de ejecución del algoritmo de evaluación postfijo

El resultado de la traducción de la expresión postfija es 1.

En esta función de evaluación también se van a detectar errores. Estos son:

- Si hay un operador, y se deben de sacar los dos últimos elementos de la pila de operandos, si no es posible ya que no hay suficientes elementos, se mostrará por pantalla el siguiente mensaje de la clase UIAlertView: "No hay suficientes operandos para un operador".
- Si el operador es un NOT y no hay ningún operando almacenado, se notificará al usuario el siguiente error: "No hay operando para un NOT".
- Una vez finaliza el algoritmo solo debe quedar un operando en la pila, si no es así, la expresión era incorrecta. Siendo el mensaje de error: "Demasiados operandos en la expresión".

A.5 Traduciendo la expresión lógica

El objetivo principal de este método de introducción de expresiones de la herramienta "Traductor", es que al pulsar el botón "Traducir" se obtenga la tabla de verdad correspondiente a la expresión lógica introducida. Para ello se aplican cada uno de los algoritmos en el orden seguido durante la explicación. Realizando primero la verificación de si la expresión es correcta y seguidamente efectuar la conversión de infijo a postfijo. Como resultado de este último paso, se obtiene la expresión postfija buscada.

A la hora de calcular la tabla de verdad correspondiente, se realiza un bucle que recorre todos los posibles valores que puede tomar las variables utilizadas en la expresión, siendo éstos, 2 elevado al número de variables. Para cada conjunto de valores se evaluará la expresión obteniendo la tabla de verdad de la misma.

Una vez se posea la tabla de verdad, es posible obtener sin gran dificultad el sumatorio de minitérminos y el producto de maxitérminos. Éste último será tabla de verdad pero con los valores opuestos y al sumatorio le corresponderá la misma tabla de verdad obtenida tras la evaluación de la expresión.

Referencias

- [1] Stephen G. Kochan, “Programación con Objective-C”
- [2] Neal Goldstein, “iPhone Application Development For Dummies, 4th Edition”
- [3] Bert Altenberg, Alex Clarke, Philippe Mougín, “Become a Xcoder”
- [4] iOS 7 Course: <http://bitfountain.io/>
- [5] Neil Smyth, “iOS 7 App Development Essentials”
- [6] Matt Neuburg, “iOS 7 Programming Fundamentals: Objective-C, Xcode, and Cocoa Basics”
- [7] Material “Desarrollo de aplicaciones para entornos móviles”. EPS UAM.
- [8] Richard Warren, “Creating iOS Apps: Develop and Design”
- [9] <http://www.raywenderlich.com>
- [10] Blog de un iOS developer: <http://www.loreman.net/>
- [11] <http://stackoverflow.com/>
- [12] <http://cseweb.ucsd.edu/classes/su10/cse140/Quine.pdf>
- [13] Juan Manuel Cigarran Recuero, “Aprende iOS. Un Poco de Objective-C”
- [14] Erica Sadun, “iOS Drawing: Practical UIKit Solutions”
- [15] <http://www.cocoaosx.com/>

ANEXOS

A. PRESUPUESTO

1) Ejecución Material

- Compra de ordenador personal (Software incluido)..... 1.300 €
- Dispositivo móvil Apple..... 699 €
- Tableta Apple..... 389 €
- Entorno de desarrollo 0 €
- Material de oficina..... 20 €
- Total de ejecución material 2.408 €

2) Gastos generales

- 16 % sobre Ejecución Material 385,28 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material 144,48 €

4) Honorarios Proyecto

- Horas realizando tutoriales: 223 horas
- Horas programando la aplicación: 340 horas
- Horas escribiendo la memoria: 140 horas
- Total: 703 horas a 15 € / hora..... 10.545 €

5) Material fungible

- Gastos de impresión 40 €
- Encuadernación 20 €

6) Subtotal del presupuesto

- Subtotal Presupuesto 13.482,76 €

7) I.V.A. aplicable

- 21% Subtotal Presupuesto..... 2.831,37 €

8) Total presupuesto

- Total Presupuesto 16.314,13 €

Madrid, Junio de 2015
El Ingeniero Jefe de Proyecto

Fdo.: David Agra Robles. Ingeniero de Telecomunicación

B. PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de MAPAS VEITCH-KARNAUGH EN iOS. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

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

Condiciones generales

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la

misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

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

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

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

Condiciones particulares

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

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

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

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

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

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

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

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

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

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

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

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

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