

Sistema KARMIN: una Herramienta para la Enseñanza de Simplificación de Funciones Lógicas

Raúl I. Jauregui¹, Ángela Belcastro¹, Nilda Belcastro², Adriana Désima¹,
Martin Bilbao¹, Esteban Musacchio¹ y José I. Gallardo¹

¹Depto. Informática- Facultad de Ingeniería y SECyT-UNPSJB

²Facultad de Ciencias Económicas

Universidad Nacional de la Patagonia "San Juan Bosco"

(9000) Ruta 1-Km4- Sede Comodoro Rivadavia- Chubut- Argentina

{rjauregui, angelab, adesima, jgallardo}@ing.unp.edu.ar

{nildaeb, martin.bilbao, emusacchio}@gmail.com

Abstract. The present work describes the software KARMIN (of KARnaugh MINimizer), an application that solves logical functions, either really entering the data of the same one by means of its Truth Table, or by means of its Map of Karnaugh, of where it derives its name. This application arose like idea inside the Project of Investigation PI N°686/2007, "Analysis and development of a system of conceptual connections and applications of contents of diverse matters of Computer Science careers, as support tool to the student's learning" [AyD SEC CHA] carried out by a group of educational of the Dept. Computer Science, of the Facultad of Engineering of the Universidad Nacional of the Patagonia "S.J.Bosco", with the final idea of implementing a generic system where the student finds educational software and other useful applications.

Keywords: Educational Software, Logic Functions, Maps of Karnaugh (K-maps for short).

Resumen. El presente trabajo describe el software KARMIN (de KARnaugh MINimizer), una aplicación que resuelve funciones lógicas, ya sea ingresando los datos de la misma mediante su Tabla de Verdad, o mediante su Mapa de Karnaugh, de donde deriva su nombre. Esta aplicación surgió como idea dentro del Proyecto de Investigación PI N°686/2007, "Análisis y desarrollo de un sistema de enlaces conceptuales y aplicativos de contenidos de diversas materias de las carreras de Informática, como herramienta de apoyo al aprendizaje del alumno" [AyD SEC CHA] realizado por un grupo de docentes del Depto. Informática, de la Facultad de Ingeniería de la Universidad Nacional de la Patagonia "S.J.Bosco", con la idea final de implementar un sistema genérico donde el alumno encuentre software educativo y otras aplicaciones útiles.

Palabras clave: Software Educativo, Funciones Lógicas, Mapas de Karnaugh.

1 Introducción

Los materiales educativos son recursos para el aprendizaje que responden a las características de los estudiantes, según el nivel educativo en el que se encuentra. Muchos de ellos poseen una estructura y diseño tal que le permiten al estudiante el despliegue de sus potencialidades, la superación de dificultades y el logro de aprendizajes significativos.

El uso adecuado de estos materiales en las aulas permite además, incentivar la curiosidad, el ingenio y la innovación, así como satisfacer las necesidades de juego, exploración y conocimiento, con el fin de potenciar sus aprendizajes. Los materiales educativos para los docentes están orientados a brindar pautas para mejorar su desempeño en el proceso de enseñanza.

Los programas computacionales para educación, como todo material que es elaborado para apoyar los procesos de enseñanza y aprendizaje, cumplen con una serie de características. Sin embargo, como un soporte electrónico digital, el software también revela un conjunto de características muy particulares. [1]

Galvis [2] menciona que un buen software educativo debe tener las siguientes características:

1. Que tome en cuenta las características de la población destinataria.
2. Se adecue a los niveles de dominio diferenciado de los usuarios.
3. Que tenga la capacidad de llenar vacíos conceptuales, detectándolos y teniendo la forma de satisfacer a los usuarios.
4. Que tenga la capacidad de desarrollar habilidades, conocimientos y destrezas circunstanciales en el logro de los objetivos de aprendizaje.
5. Que explote sus propias potencialidades técnicas y de interacción.
6. Que promueva la participación activa de los usuarios en la búsqueda, generación, apropiación y reconstrucción del conocimiento.
7. Que permita vivir y reconstruir experiencias a los usuarios, lo cual sería difícil o imposible de lograr a través de otros medios.

Una de las formas de clasificar el software educativo es considerar la clasificación estándar del software (de base y de aplicación). Nos encontramos con:

- 1- El *Genérico* ó carente de contenidos: por ej. Procesador de textos, empleado para manipular textos.
- 2- El *Específico* para cada asignatura: por ej. Programas de simulación para temas específicos de ciencias naturales, los de práctica de idioma inglés.

Squires y Mc.Dougall [3], citando a Pelgrum y Plomp, presentan una lista de 23 categorías, demuestra que a medida que se incrementa el alcance y la diversidad del software, crece la cantidad de categorías del software educativo, basada en el tipo de aplicación.

El sistema KARMIN es específico del tema simplificación de funciones lógicas ó booleanas a través del método de Mapas de Karnaugh [4,5,6] y de utilidad para las

asignaturas “Elementos de Lógica y Matemática Discreta”, y “Arquitectura de los Sistemas de Cómputo”.

Galvis [2] señala otra clasificación, planteada de acuerdo a las funciones educativas: tutoriales, de ejercitación y práctica, simuladores, juegos educativos, sistemas expertos y los inteligentes de enseñanza.

El sistema KARMIN es de ejercitación y práctica, ya que refuerza conocimientos previos impartidos por otros medios, permitiendo al usuario verificar ejercicios resueltos de simplificación de funciones lógicas, autoevaluarse y obtener conclusiones.

1.1 Ubicación referencial de KARMIN

Para su desarrollo se tuvieron en cuenta las siguientes decisiones didácticas previas:

a) *Necesidad educativa que da origen a la propuesta:* KARMIN es una herramienta creada para ayudar a los alumnos de las carreras de informática, a comprender y ejercitarse en la simplificación de funciones lógicas, mediante el método de Mapas de Karnaugh ó desde su Tabla de Verdad

b) *Áreas temáticas:* Informática- Lógica y Matemática Discreta. Álgebra de Boole. Arquitectura de Computadoras.

c) *Caracterización de los destinatarios:* Los destinatarios de esta herramienta son alumnos ingresantes de 1º ó del 2º año de las carreras de informática, cuya edad promedio es 19 años, que generalmente tienen un asiduo uso de herramientas informáticas e Internet, por lo que suponemos que no se les presentará inconvenientes con la utilización de este sistema.

d) *Objetivo:* Que el alumno pueda utilizar la herramienta para verificar ejercicios resueltos de simplificación de funciones lógicas, para autoevaluarse y obtener conclusiones.

2 Desarrollo software KARMIN

Se propone una descripción somera sobre la evolución del software para el trabajo de investigación sobre enlaces conceptuales, así como también los métodos empleados para su implementación.

El software KARMIN es una herramienta que permite al usuario simplificar funciones booleanas, las cuales pueden ser ingresadas mediante dos métodos distintos:

- Mediante su Tabla de Verdad
- Mediante un Mapa o Tabla de Karnaugh

Independientemente del método de ingreso, el software simplificará la función booleana para obtener una posible expresión mínima. Para tal fin se eligió implementar el algoritmo de Quine-McCluskey, que constituye un método de simplificación de funciones booleanas desarrollado por Willard Van Orman Quine y Edward J. McCluskey [5,7].

Funcionalmente es idéntico a la utilización del Mapa de Karnaugh, pero su forma tabular lo hace más eficiente para su implementación en lenguajes computacionales, a la vez que provee un método determinístico para conseguir la mínima expresión de una función booleana.

El método consta de dos pasos:

- Encontrar todos los implicants primos de la función.
- Usar esos implicants en una tabla de implicants primos para encontrar los implicants primos esenciales, los cuales son necesarios y suficientes para generar la función.

2.1. Inconvenientes en el avance de la implementación

Para el desarrollo del software KARMIN se ha elegido el lenguaje de programación C#, que constituye un lenguaje de programación orientado a objetos, y se ha escogido por ser un lenguaje con un entorno de desarrollo amigable y de creciente popularidad.

A pesar de estas ventajas, en la implementación del software se encontraron algunos obstáculos para su desarrollo, a saber,

- Al ser un lenguaje con el que el equipo de desarrollo no había trabajado, se generaron demoras para su aprendizaje y usos.
- La utilización del componente Msflexgrid, elegido para representar el Mapa de Karnaugh, no fue compatible con una nueva versión del lenguaje, lo que obligó a optar por otra alternativa para mostrarlo, causando algunas demoras en el avance del sistema.
- El algoritmo de **Quine-McCluskey** presenta un inconveniente cuando en el manejo de redundantes (ó don't care) puede encontrar más de una solución óptima, según los redundantes que se tomen. Esta situación obligó a determinar una manera de seleccionar una de las posibles soluciones equivalentes.

2.2. Estado de la implementación

El software se encuentra implementado completamente. Si bien el Algoritmo de Quine-McCluskey puede encontrar más de una solución óptima, en función de los términos redundantes que asocie, en esta implementación, se decidió que la solución a mostrar será una de las soluciones equivalentes, que tome la menor cantidad de términos redundantes.

KARMIN es de distribución gratuita, y se encuentra disponible por ahora en el sitio web de la asignatura Arquitectura de los Sistemas de Cómputo [www.ing.unp.edu.ar/asignaturas/arquitectura/].

2.3. Requisitos de instalación:

- Sistema operativo Windows XP o superior
- .Net Framework 2.0 ó superior
- 2 MB de espacio en disco disponibles

Para instalar KARMIN es necesario contar con el Framework .NET 2.0 o superior, ya que el mismo se encuentra desarrollado en C#. Se ha probado instalarlo, tanto en Windows XP como en Vista y 7, funcionando sin inconvenientes en sendos casos.

Podría funcionar en otras versiones anteriores de Windows XP, ya que existen versiones de Framework de .NET para Windows98, ME y 2000. Y en distribuciones de Linux, teniendo instalada una máquina virtual con Windows.

3 Funcionamiento y uso de KARMIN

Una vez instalado KARMIN, en el inicio se puede seleccionar el método de trabajo desde el menú, ya sea por Tabla de Verdad o por Mapa de Karnaugh.

Si no tiene decidido cuál método utilizar, se recomienda seleccionar el Asistente (punto 3.4), que lo guiará en la tarea..

La Figura 1 muestra la pantalla principal del programa. Desde la misma se puede acceder a las siguientes funciones.

- 3.1 Salir del sistema
- 3.2 Resolver por tabla de verdad
- 3.3 Resolver por mapa de Karnaugh
- 3.4 Ejecutar el asistente
- 3.5 Mostrar la ayuda del programa

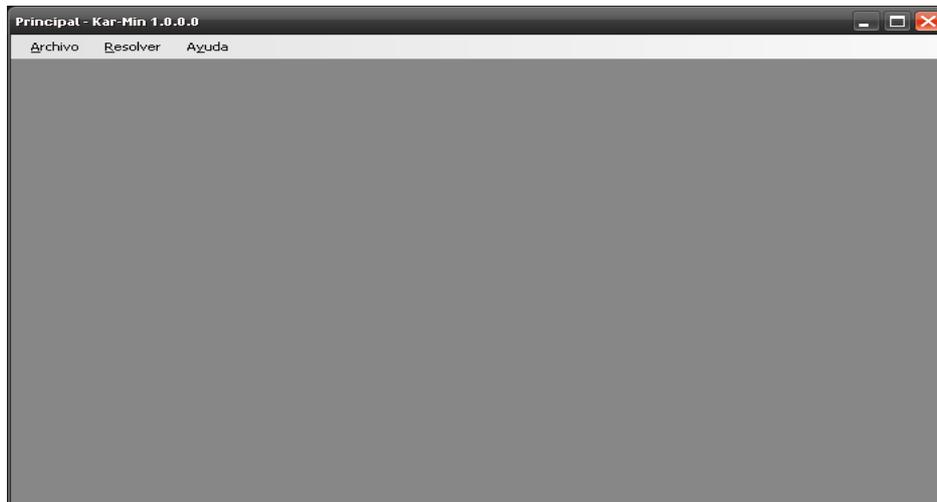


Figura 1. Ventana Principal.

3.1. Salir del sistema

Para salir del sistema debe ir a la opción Archivo → Salir, ó presionar combinación de teclas Ctrl+S.

3.2. Resolver por tabla de verdad

Para resolver la función lógica mediante su Tabla de Verdad, se debe ir a la opción Resolver → Tabla de verdad, la cual le mostrará la siguiente pantalla. (Figura 2)

Desde aquí se podrá indicar cuáles son los unos, ceros o términos redundantes de la función, como así también la cantidad de variables que posee la función a analizar. Además se le proveen al usuario métodos de relleno automáticos para facilitar la tarea. Una vez cargados los datos necesarios, se pulsa el botón Resolver para obtener, de ser posible, la solución óptima.

Dec	A	B	C	D	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Figura 2. Resolución por tabla de verdad.

3.3. Resolver por mapa de Karnaugh

Para resolver la función lógica mediante su Mapa de Karnaugh, se debe ir a la opción Resolver → Tabla Mapa de Karnaugh, la que mostrará la siguiente pantalla. (Figura 3)

Al igual que con la opción anterior, desde aquí se podrán ingresar los unos, ceros o términos redundantes y cantidad de variables. También se dispone de métodos de relleno automáticos. La Figura 3 muestra un ejemplo de una función lógica con sus términos redundantes, un cuadro de texto con la descripción de la función lógica en su primera forma canónica (por suma de miniterminos) y un 2º cuadro de texto con la

solución óptima reducida. Para obtener la misma se debe pulsar el botón Resolver, una vez cargada la función a analizar.

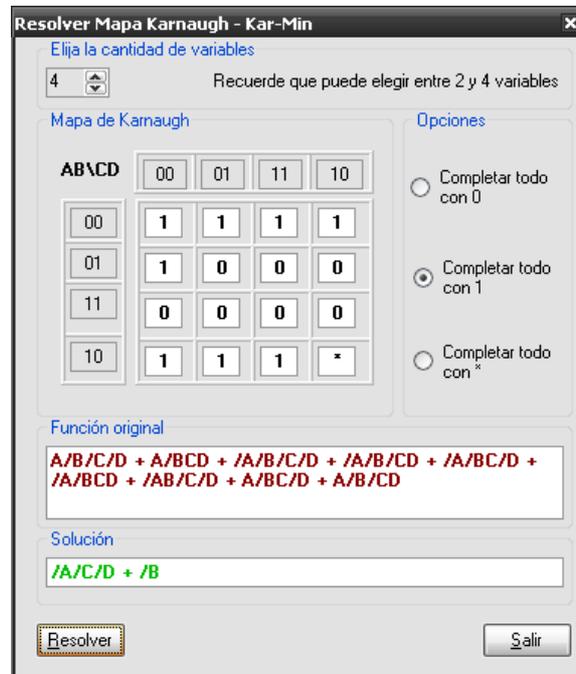


Figura 3: Resolución por Mapa de Karnaugh.

3.4. Ejecutar el asistente

Para acceder al asistente, se deberá ir a la opción Resolver → Asistente. El Asistente permitirá seleccionar el método y número de variables para luego ingresar los datos de la función. Este asistente tiene como objetivo acceder de forma más sencilla a los distintos métodos de resolución de KARMIN. (Figura 4)

Como muestra la Figura 4, esta ventana provee dos pasos previos. El primero permite seleccionar uno de los métodos de resolución (Tabla de Verdad o Mapa K).

El segundo, indicar el número de variables (mostrando de acuerdo a la cantidad, la denominación de la misma). Si pulsa siguiente en el paso 2, se mostrará la ventana con el método de resolución seleccionado en el paso uno. Cuando finalice con el mismo, volverá nuevamente a esta pantalla por si se desea modificar algunas de las opciones de los pasos previos; caso contrario se pulsará salir para finalizar el asistente.

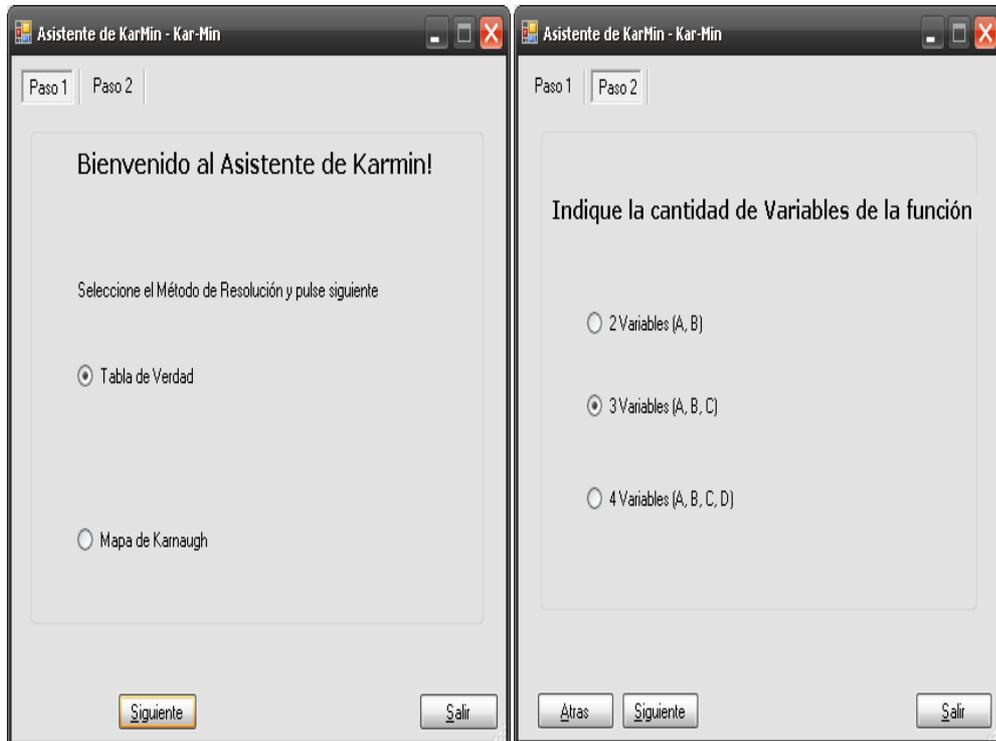


Figura 4. Pantallas de asistente de KARMIN.

3.5. Ayuda del programa

La ayuda del programa mostrará una ventana con un texto de ayuda y descripción de las funciones básicas del programa. (Figura 5)

3.6. Observaciones:

KARMIN, como todo software, no se encuentra libre de errores de funcionamiento. Por ello, si el usuario encontrara alguno, por favor escribir a rjauregui@ing.unp.edu.ar.

Si bien el algoritmo de Quine-McCluskey puede trabajar con hasta 32 variables (con su correspondiente complejidad), KARMIN sólo trabaja con 2 a 4 variables, cantidades surgidas del ámbito educativo donde se planteaba usar el mismo.

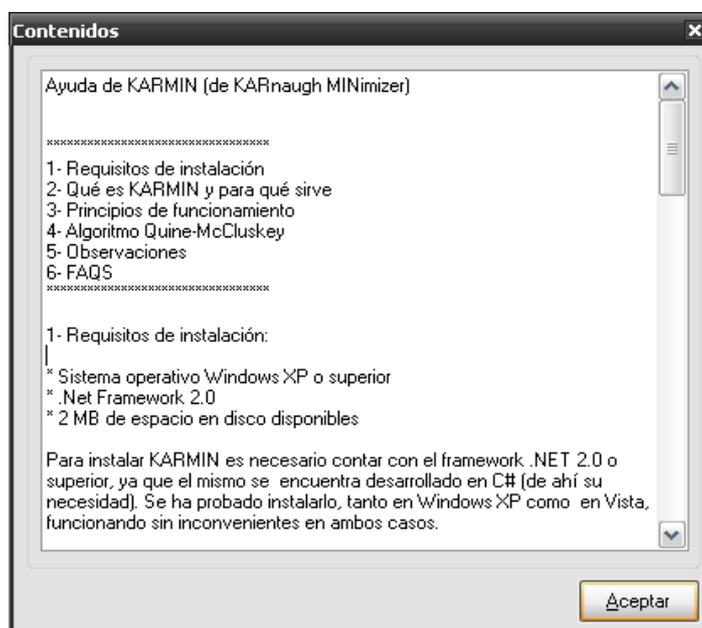


Figura 5: Pantalla de Ayuda de KARMIN.

4. Resultados obtenidos y su valoración

Se realizaron dos pruebas piloto de su funcionamiento con los alumnos de la asignatura “Arquitectura de Sistemas de Cómputo” durante el corriente ciclo lectivo 2010, dentro del capítulo Introducción a las Técnicas Digitales, presentando el software en una clase, y otro día en forma individual pudieron interactuar en clase práctica de laboratorio.

Como herramienta de medición de resultados de su uso, se utilizó un formulario de diagnóstico de prueba de KARMIN, para recabar las opiniones de los alumnos. De un total de 16 inscriptos en el ciclo lectivo 2010 de “Arquitectura de los Sistemas de Cómputo”, 12 realizaron la cursada completa, y a 9 (nueve) de ellos se les realizó la encuesta, a fines de,

- Determinar el grado de aceptación del programa por parte de los estudiantes.
- Detectar fallas para posteriormente realizar los respectivos ajustes.
- Verificar si el software cumple el objetivo para el cual fue diseñado.

Las Opiniones de los alumnos fueron: El software es fácil de usar; Tiene una navegación e interacción entre muy buena y excelente; Facilita la comprensión; tiene una versatilidad media; Sirve para verificar las simplificaciones algebraicas. Entre las dificultades observadas por el 33% de los alumnos podemos mencionar el atractivo de las pantallas.

5. Conclusiones y trabajo futuro

En base a las pruebas piloto realizadas con los alumnos de la asignatura “Arquitectura de Sistemas de Cómputo” durante el presente ciclo lectivo, la impresión ha sido muy favorable.

Del análisis de resultados, concluimos que el sistema responde a las necesidades del entorno y de sus estudiantes, no se observaron mayores dificultades en su uso, constituyéndose en una valiosa herramienta de verificación de resultados obtenidos manualmente. Si bien existen otras aplicaciones similares en la web aquí se destacan: la articulación entre un producto de investigación y su empleo en dos cátedras, la experiencia docente involucrada y el sentido de pertenencia del producto. Así consideramos estar logrando uno de los objetivos propuestos en la introducción, y en la filosofía del PI al que pertenece este desarrollo, ya que a su vez se cuenta con otros sistemas en pleno uso.

Se espera repetir su uso en lo que resta del actual ciclo lectivo en la actividad curricular “Elementos de Lógica y Matemática Discreta”, para las prácticas simplificación de funciones booleanas.

Referencias

1. Morales Velásquez, C.; Neri, Y.; Martínez V.C.; Reyes S. *Evaluación de software educativo*. Instituto Latinoamericano de la Comunicación Educativa-ILCE. <http://investigacion.ilce.edu.mx/st.asp?id=765>. México (1998) Accedido el 10 de Julio de 2010
2. Galvis, Á.H.: *Evaluación de materiales y ambientes educativos computarizados*- Revista Informática Educativa. Vol.6, N°1. Proyecto SIN, Bogotá, Colombia (1993)
3. Squires, D; Mc.Dougall, A.: *Cómo elegir y utilizar software educativo*- Morata. Colección “Educación crítica”. Capítulo VI: “Marcos de Referencia para estudiar el software educativo”. Madrid, España (1997)
4. Karnaugh, M.: *The Map Method for Synthesis of Combinational Logic Circuits*- Transactions of the American Institute of Electrical Engineers. Vol.72 N°9, pp. 593-599. USA (1953)
5. Mandado, E.: *Sistemas Electrónicos Digitales*- 8ª Edic. Marcombo, España (1998)
6. Tocci, R.; Widmer N.S.: *Sistemas Digitales. Principios y Aplicaciones*. 8ª Edic. Prentice Hall, México (2003)
7. Wakerly, John F.: *Diseño Digital. Principios y Prácticas*- 3ª Edic. Prentice Hall, México (2001)
8. Hill, F.; Peterson, G.: *Teoría de Conmutación y Diseño Lógico*- Limusa Noriega, México (1994)