

Nueva aplicación didáctica para electrónica digital

Javier García Zubía

Dpto. de Arquitectura de Computadores
Facultad de Ingeniería
Universidad de Deusto
48007 Bilbao
e-mail: zubia@eside.deusto.es

Resumen

El objetivo del programa BOOLE-DEUSTO es ayudar al profesor y alumno de electrónica digital a analizar y diseñar sistemas digitales básicos. En el presente trabajo se describe la nueva versión de BOOLE-DEUSTO, cuya funcionalidad y utilidad han crecido considerablemente respecto de la primera versión. El interés por BOOLE-DEUSTO se centra en su didáctica, potencia, gratuidad y amplia distribución universitaria.

1. Introducción y originalidad de BOOLE-DEUSTO

El programa BOOLE-DEUSTO existe desde 1.995, desde entonces más de diez personas y miles de líneas de código lo conforman. En el año 2.000 se presentó una primera versión profesional en el IV Congreso de Tecnologías Aplicadas a la Enseñanza celebrado en Barcelona [1], siendo premiado como Mejor Equipo Software TAAE 2.000. BOOLE-DEUSTO destacó por ser didáctico, sencillo, completo, gratuito, fácilmente instalable y adaptado a las necesidades del alumno. También fue presentado en el JENUI 2001, y en [2] y [3], lo que ayudó a divulgarlo. Ahora está disponible la nueva versión 2.0 de BOOLE. El número de mejoras de BOOLE 2.0 y su amplia distribución universitaria, tanto nacional como internacional, nos anima a presentar este trabajo.

Recordemos (ver [2]) que BOOLE-DEUSTO ayuda al alumno y al profesor en la primera fase de aprendizaje de la electrónica digital. Esta fase es descuidada por los entornos profesionales, como OrCAD, Electronic WorkBench, Xilinx,

etc., que buscan más el resultado que el proceso, y que son caros y difíciles de manejar e instalar. Estos entornos están orientados al profesional, y no al aula, y parece poco adecuado utilizarlos en un primer momento en la clase. BOOLE-DEUSTO viene a ocupar este hueco, ayudando por igual al profesor y al alumno.

2. Campo de aplicación

BOOLE-DEUSTO es de gran utilidad en un curso básico de electrónica digital y está orientado al análisis y diseño de sistemas combinacionales y secuenciales a nivel de bit.

Centrándonos en los sistemas combinacionales, la figura 1 muestra las opciones disponibles en el programa.

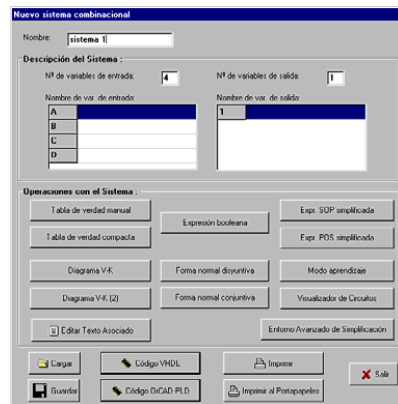


Figura 1. Menú principal de sistemas combinacionales

BOOLE-DEUSTO ofrece utilizar para sistemas combinacionales:

- Expresiones booleanas.
- Tablas de verdad.
- Diagramas de Veitch-Karnaugh.
- Formas normales.
- Expresiones simplificadas o minimizadas.
- Expresiones NAND/NOR.
- Circuitos lógicos: redes AND-OR y OR-AND.
- Simplificador booleano especializado.
- Simplificador booleano para el aprendizaje del alumno.
- Generador de código VHDL, ABEL y OrCAD-PLD.

En cuanto a los sistemas secuenciales o autómatas, BOOLE-DEUSTO ofrece:

- Captura gráfica del diagrama de transición de estados de Mealy o Moore.
- Verificación del autómata.
- Reducción o minimización de estados del autómata.
- Diseño del autómata: generación de las tablas.
- Conversión Moore-Mealy, y viceversa.
- Obtención del circuito lógico.
- Generador de código VHDL, ABEL y OrCAD-PLD.
- Simulación interactiva y batch.

En cuanto al entorno en su conjunto, BOOLE permite:

- Salvar y cargar sistemas digitales.
- Imprimir o guardar resultados.
- Copiar al portapapeles las figuras y tablas.
- Editar un texto para cada sistema.
- Versiones en español e inglés.

3. Estrategia de BOOLE-DEUSTO

Como ya se ha dicho, los entornos profesionales están más interesados en los resultados, que en los métodos y sus pasos, resultando inadecuados para el aula. Muchos profesores han dado soluciones parciales a esta incompatibilidad, pero no conocemos ningún entorno que lo haya hecho globalmente y de forma profesional. BOOLE-

DEUSTO quiere ser una herramienta fundamental en el desarrollo de un curso básico de electrónica digital, tecnología de computadores, etc.

Otra característica del BOOLE, y que pocos entornos tienen o buscan, es que en ningún momento marca el método a seguir para diseñar un sistema. BOOLE simplemente ofrece los métodos (ver punto 2), y es el alumno/usuario el encargado de elegirlos, aplicarlos y observar los resultados obtenidos. BOOLE es un entorno no dirigido, que bien podría tomar el nombre de calculadora booleana. Quizá éste sea uno de los aspectos más destacados por cuantos profesores lo usan y recomiendan.

Por último, BOOLE presenta dos aspectos estratégicos que son reclamados por el mundo de la enseñanza. Uno es que permite al profesor/usuario generar, resolver y documentar ejercicios de forma rápida. Y dos, BOOLE se distribuye como un único fichero .exe, sin instalación, sin licencias, sin Internet, sin requisitos hardware, etc.; cuando el profesor lo entrega o distribuye sabe que nunca tendrá que resolver problemas de instalación.

4. Nueva versión de BOOLE-DEUSTO

Un buen número de profesores de España, Europa y América, más de cien, ya utilizan en clase y entregan a sus alumnos una copia del BOOLE. En este apartado se describirán las novedades que presenta BOOLE en la versión 2.0. Esta descripción se centrará en los sistemas combinacionales y secuenciales por separado.

4.1. Sistemas combinacionales

Como primera novedad para sistemas combinacionales, BOOLE permite su captura mediante una expresión booleana cualquiera. El procesador de expresiones booleanas desarrollado permite introducir cualquier expresión sin restricciones de ningún tipo, admitiendo los operadores NOT, AND, OR y XOR en cualquier orden. El procesador está validado y ha sido diseñado aplicando técnicas de compiladores. En la figura 2 vemos un ejemplo.

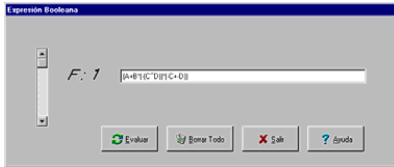


Figura 2. Expresión booleana.

Una vez evaluada la expresión, el BOOLE crea el sistema correspondiente. El usuario puede verlo, transformarlo, operarlo, etc, todo ello como si manejara una calculadora. La figura 3 muestra el diagrama V-K de la expresión introducida.

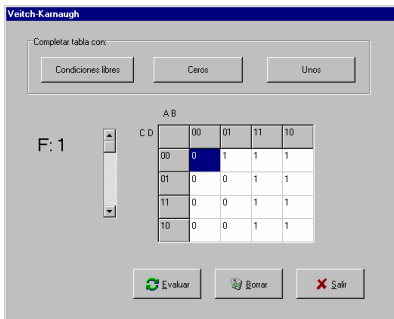


Figura 3. Diagrama V-K de la expresión capturada

Podemos ver otra novedad en la parte inferior derecha de la figura 1. En ella BOOLE ofrece el entorno avanzado de simplificación. El REDUCTIO, desarrollado en su totalidad por Borja Sotomayor, simplifica una función booleana mediante seis métodos distintos, heurísticos y no heurísticos, lo que permite al usuario comparar la bondad de cada uno de ellos.

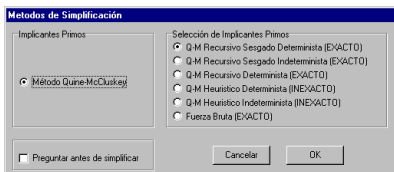


Figura 4. Métodos de simplificación de REDUCTIO

Por último, y dentro de los sistemas combinacionales, aparece como novedad la posibilidad de generar automáticamente el código VHDL del sistema creado. De esta manera podemos conectar BOOLE con programas profesionales, y alejarle de ser considerado un programa sin aplicación práctica. A continuación se muestra el código VHDL de un sistema combinacional.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity sistema 1 is
    Port
    (
        P5: in std_logic;
        P10: in std_logic;
        P25: in std_logic;
        P50: in std_logic;
        E1: out std_logic;
        E2: out std_logic;
        E5: out std_logic;
        E10: out std_logic;
        E20: out std_logic
    );
end sistema 1;

architecture behavioral of sistema is
begin
    E1<=((P10) or (P5))
    E2<=((P5))
    E5<=((P25) or (P10))
    E10<=((P50) or (P25))
    E20<=((P50));
end behavioral;
```

Figura 5. Código VHDL de un sistema combinacional

Esta planificado añadir el lenguaje ABEL a los VHDL y OrCAD-PLD ya existentes.

4.2. Diagramas de V-K

La primera versión de BOOLE ya hacía hincapié en el uso de los diagramas de Veitch-Karnaugh, de manera que el alumno podía cargar, y luego simplificar, cualquier función booleana mediante diagramas V-K, acercándose al modo de trabajar en clase.

En el nuevo BOOLE los V-K siguen siendo protagonistas. Veamos, el alumno introduce una función booleana mediante el V-K y luego la simplifica para ver el resultado, la novedad se centra en que BOOLE dibuja cada uno de los lazos de la función simplificada.

En la figura 6 se puede ver un V-K y los lazos correspondientes a su simplificación.

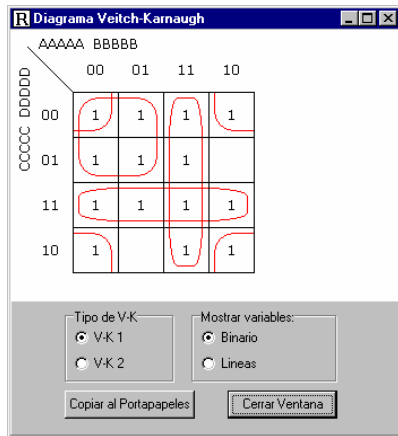


Figura 6. Visualización de lazos en el V-K

Esta habilidad del BOOLE vuelve a mostrarle como un entorno preocupado por las necesidades del alumno y del profesor, sobre todo en lo que concierne a los métodos y sus pasos. Aspectos como éste, y otros del BOOLE, quedan fuera del enfoque de entornos profesionales.

Actualmente se trabaja para que BOOLE también permita que el alumno introduzca mediante lazos la expresión simplificada, "pintando los lazos", sin escribir obligatoriamente la función algebraica correspondiente.

4.3. Sistemas secuenciales: Autómatas

El grueso de las novedades se concentra en los autómatas. En primer lugar BOOLE transforma un autómata de Moore en el correspondiente de Mealy, y viceversa. BOOLE redibuja el autómata y asigna los nuevos nombres a los estados. Las figuras 7 y 8 muestran el autómata de Mealy original y el equivalente de Moore.

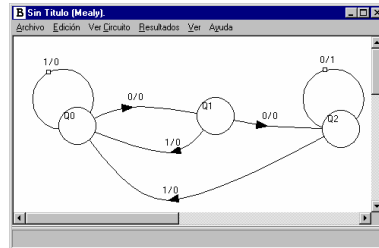


Figura 7. Autómata de Mealy original

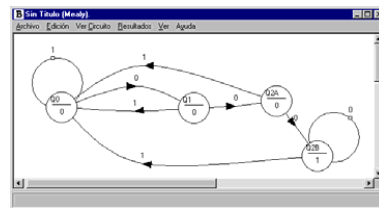


Figura 8. Autómata de Mealy equivalente

En el nuevo BOOLE se ofrece la posibilidad de reducir un autómata, ya sea de Moore o de Mealy. El alumno introduce un autómata y el BOOLE obtiene y redibuja aquel que con menos estados es equivalente al original. Las figuras 9 y 10 muestran un autómata original con cinco estados y el correspondiente reducido con cuatro estados, respectivamente.

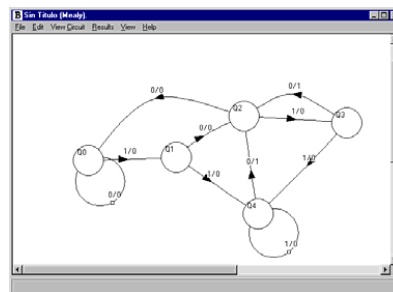


Figura 9. Autómata original de Mealy

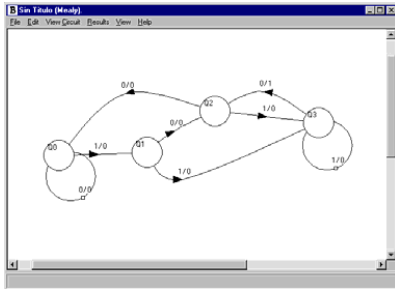


Figura 10. Automata de Mealy reducido

Todo el proceso de reducción de estados es automático, incluido el redibujar el nuevo automata. Además BOOLE ofrece al alumno ver la tabla de minimización paso a paso, y así comprobar el método aplicado. En la figura 11 se puede ver la tabla y otras informaciones adicionales.

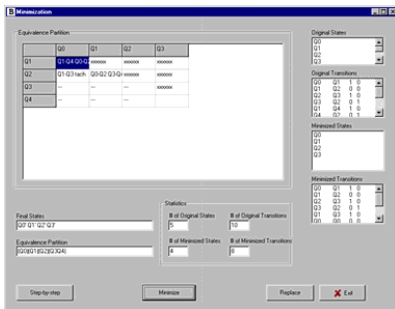


Figura 11. Automata de Mealy reducido

Por último, en la minimización de estados BOOLE permite procesar autómatas incompletamente especificados, tanto en sus salidas como en las transiciones. El algoritmo implementado es de cierta complejidad y dota a BOOLE de una potencia de la que carecen incluso algunos entornos profesionales (lo mismo que para la simplificación booleana). Las figuras 12 y 13 muestran un autómata de Moore incompleto y el correspondiente minimizado.

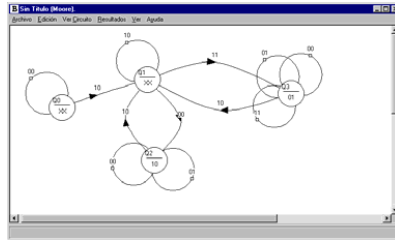


Figura 12. Automata original de Moore

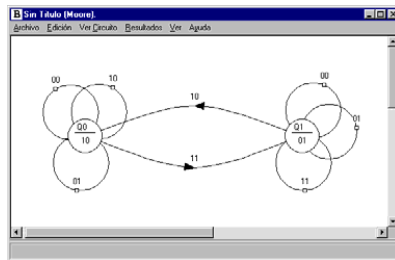


Figura 13. Automata de Moore reducido

Una vez capturado y reducido el autómata podemos visualizar sus tablas de diseño y su circuito lógico. Las figuras 14 y 15 muestran las tablas de resultados y el circuito lógico del autómata de la figura 13, respectivamente.

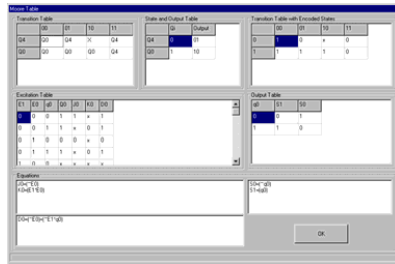


Figura 14. Diseño del autómata de Moore

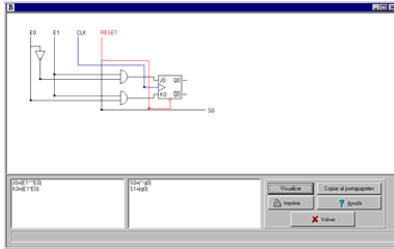


Figura 15. Circuito del autómeta de Moore

También se puede obtener el código VHDL del autómeta cargado, permitiendo la conexión de BOOLE con cualquier entorno profesional del tipo Xilinx ISE o Foundation, MaxPlus II, OrCAD, ispDESIGN de Lattice, etc. A continuación se lista el código VHDL.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity Sistema is
  Port (
    inicio: in std_logic;
    ck: in std_logic;
    E0: in std_logic;
    E1: in std_logic;
    S0: out std_logic;
    S1: out std_logic
  );
end Sin_Titulo;
architecture behavioral of Sistemais
type nombres_estados is (Q1, Q0);
signal estado: nombres_estados;
signal entrada_aux: std_logic_vector
(1 downto 0);
begin
  entrada_aux<=E0&E1
  process(inicio, ck)
  begin
    if inicio='1' then
      estado<=Q1;
    elsif ck='1' and ck'event then
      case estado is
        when Q1 =>
          case entrada_aux is
            when "00" => estado<=Q1;
            when "01" => estado<=Q1;
            when "10" => estado<=Q0;
            when "11" => estado<=Q1;
          when others => estado<=Q1;
        end case;
        when Q0 =>
          case entrada_aux is
            when "00" => estado<=Q0;
            when "01" => estado<=Q0;
            when "10" => estado<=Q0;
            when "11" => estado<=Q1;
          end case;
      end case;
    end process;
  end behavioral;

```

```

when others => estado<=Q1;
end case;
when others => estado<=Q1;
end case;
end if;
end process;
process(estado)
begin
  case estado is
    when Q1 =>
      S0<='0';
      S1<='1';
    when Q0 =>
      S0<='1';
      S1<='0';
    end case;
  end process;
end behavioral;

```

4.4. Sistemas secuenciales: Simulación

Otro aspecto en que BOOLE ha mejorado considerablemente es en la simulación del autómeta capturado. Al simular, el profesor muestra en clase la evolución del sistema diseñado, aclarando conceptos y situaciones. Por otra parte, cuando el alumno simula está comprobando que lo diseñado cumple las especificaciones, es más, está justificando frente al profesor la idoneidad de lo creado. La importancia de ambas situaciones nos ha llevado a diseñar cuidadosamente, aunque no muy espectacularmente, la simulación de un autómeta. Hemos dado prioridad a dos aspectos: agilidad y diversidad. En el menú, BOOLE ofrece simular de forma interactiva, batch, rápida o detallada. Veamos algunas características de cada una de ellas.

En la simulación interactiva, el alumno va introduciendo entradas *sobre la marcha*, y simultáneamente ve los resultados: on-line. Sin embargo en batch, el alumno/profesor primero prepara la secuencia de entrada con la que va excitar al autómeta y luego la introduce al autómeta. Elegir una u otra opción depende de cada aplicación y del carácter del usuario. En la figura 16 vemos la pantalla de simulación interactiva, en ella podemos ver que un estado está marcado con un círculo: es el estado actual de la simulación.

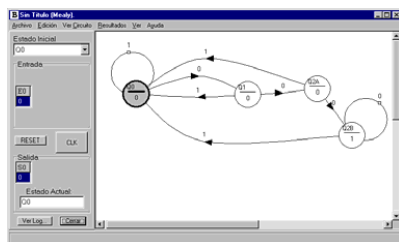


Figura 16. Simulación interactiva

La figura 17 muestra cómo se introduce y simula una secuencia de entrada en modo batch.

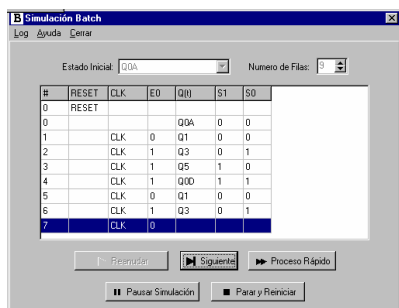


Figura 17. Simulación batch

La diferencia entre simulación rápida o detallada estriba en que en la primera el usuario sólo puede cambiar las entradas en los flancos, es decir, un flanco a una entrada. Mientras que en la detallada el usuario puede cambiar la entrada entre flancos, y, por ejemplo, de esta manera podrá apreciar la diferencia entre Moore y Mealy. La más común es la rápida, pero la detallada puede ser la idónea para que el profesor explique ciertos casos particulares. La figura 18 muestra el aspecto de una simulación interactiva detallada.

En cualquier caso, tanto si la simulación es batch, interactiva, rápida o detallada, el BOOLE ofrece una tabla con los resultados para que el alumno pueda observarlos con detalle, o imprimirlos, o guardarlos o cargarlos, aportando comodidad y potencia al trabajo de clase. La figura 19 muestra la tabla-resultado de una simulación.

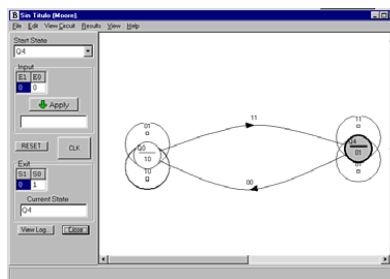


Figura 18. Simulación batch

Paso	RESET	CLK	E0	Q(t)	S0
1	RESET			Q0	0
2		CLK	0	Q1	0
3		CLK	1	Q0	0
4		CLK	1	Q0	0
5		CLK	1	Q0	0
6		CLK	0	Q1	0
7		CLK	1	Q0	0
8		CLK	0	Q1	0
9		CLK	0	Q2a	0

Figura 19. Log de una simulación

5. Experiencia en el aula

Las primeras versiones de BOOLE son del año 1.995, y ya desde 1.996 empezamos a usarlo en el aula, y sobre todo se lo dimos a los alumnos para que lo usaran en casa como una calculadora booleana que les ayudaba a corregir los ejercicios, y a documentar los hechos por ellos.

En cuanto al uso en el aula: el profesor enseña un método, hace algún ejemplo y manda unos ejercicios. El alumno los resuelve en casa ayudándose del BOOLE para corregirlos y documentarlos y/o para repasar los métodos aplicados. El peligro de usar BOOLE es claro: puede que el alumno no haga los ejercicios, sino que directamente use el programa; igual que cuando aprendíamos a dividir y teníamos una calculadora. Es más una cuestión del alumno que del profesor el buen uso que se haga del BOOLE.

En cuanto al campo de aplicación del BOOLE, este es amplio. BOOLE implementa con todo detalle, paso a paso, todos los métodos de manipulación booleana, las técnicas de diseño combinacional con tabla de verdad y el diseño de autómatas, quedando fuera del BOOLE, por falta de una metodología sistemática, el diseño de sistemas combinacionales y secuenciales con funciones estándar. En términos generales el BOOLE ayuda en al menos el 50% de un curso básico de electrónica digital.

Otra cuestión es, ¿cuánto se tarda en dominar el BOOLE? Nuestra experiencia nos dice que basta con una hora escasa para enseñar su mecanismo. Resumiendo, con poco esfuerzo se cubre buena parte de la asignatura.

Desde 1996 BOOLE-DEUSTO ha sido utilizado por unos 700 alumnos al año, tanto en Ingeniería en Informática como en Ingeniería Industrial; los primeros en la asignatura de Tecnología de Computadores, y los segundos en Electrónica Digital. Además, y como ya se ha dicho, BOOLE es utilizado en varias universidades y centros españoles, europeos y americanos, y acompaña, como software de apoyo, al libro [4].

Nuestra experiencia con BOOLE es muy positiva, y así nos lo dicen los alumnos. Ellos destacan lo idóneo del programa y lo fácil de usar. Ambos comentarios, y otros, quedan reforzados con lo que nos hacen saber profesores y alumnos de otros centros. Esto último, a nuestro parecer, es la mejor prueba de que BOOLE no es algo específico de esta universidad, sino que es claramente exportable. Esta *exportabilidad* debería ser el primer objetivo de cualquier sw con ánimo de difusión.

Queda pendiente, una vez que BOOLE esté en su versión definitiva (curso 2002-2003), una evaluación sistemática de su calidad con encuestas, grupos, etc.

6. Conclusiones

Con las novedades comentadas, BOOLE-DEUSTO refuerza sus características iniciales: análisis y diseño de sistemas digitales, orientado al aula, didáctico, ágil, potente, profesional, completo, transportable, gratuito y atractivo.

Referencias

- [1] García Zubía, J. y Sanz Martínez, J. "Adecuación de los entornos computacionales a la clase de electrónica digital: BOOLE-DEUSTO". *Actas del IV Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica*, Barcelona, 2.000.
- [2] García Zubía, J. "Metodología y necesidades computacionales en sistemas digitales". *Actas de las VII Jornadas de Enseñanza Universitaria de la Informática*, Palma de Mallorca, 2001.
- [3] García Zubía, J., Sanz Martínez, J. y Sotomayor, B. "BOOLE-DEUSTO, la aplicación para sistemas digitales". *Actas de las VII Jornadas de Enseñanza Universitaria de la Informática*, Palma de Mallorca, 2001.
- [4] Angulo Usategui, J.M. y García Zubía, J. *Sistemas digitales y Tecnología de computadores*, Ed. Paraninfo, 2.001.