

Un nuevo lenguaje de programación para el diseño de circuitos eléctricos asistido por ordenador

Patricia Ruiz*, Bernabé Dorronsoro+, Manuel Bueno Mora#

*Departamento de Ingeniería Mecánica y Diseño Industrial, Escuela Superior de Ingeniería, +Departamento de Ingeniería Informática, Escuela Superior de Ingeniería, #Escuela Superior de Ingeniería, Universidad de Cádiz.

patricia.ruiz@uca.es

RESUMEN: La asignatura de Dibujo Industrial que se imparte en la mayoría de las ingenierías está dividida en varias partes. Una de ellas consiste en la realización de esquemas eléctricos y su transformación a otro tipo de esquemas. Actualmente, los recursos disponibles para la realización de estos esquemas es muy escasa. En la literatura hay pocos ejercicios resueltos, por lo que los estudiantes no pueden practicar en casa más que los ejercicios vistos en clase. En este trabajo presentamos una herramienta web para la realización de este tipo de esquemas, cuya base ha sido desarrollada por estudiantes de la asignatura de teoría de autómatas y lenguajes formales (TALF) del grado en Ingeniería Informática.

Los estudiantes de TALF normalmente realizaban, como trabajo para superar la asignatura, la creación de lenguajes formales conocidos simplificados, por lo que trabajaban en un diseño con muchas limitaciones, y encontraban dificultades para asimilar la utilidad real que tiene esta herramienta tan potente como es la creación de lenguajes de programación específicos. Este año, se les ha propuesto una práctica muy concreta y con aplicación real: crear un lenguaje de programación específico para el diseño de circuitos eléctricos sencillos. La propuesta requiere al alumno aplicar los conocimientos adquiridos durante toda la carrera para diseñar un lenguaje de programación nuevo, altamente expresivo, fácil de utilizar, y que satisfaga los requisitos propuestos. El trabajo propuesto resultó muy provechoso ya que entendieron el gran potencial que tiene la asignatura impartida.

Basándonos en el trabajo que realizaron dichos estudiantes, hemos creado una herramienta gráfica utilizando la librería open-source jointjs. Esta herramienta permite a los alumnos crear gráficamente un esquema eléctrico cualquiera de manera muy sencilla, ayudándoles a comprender el funcionamiento del circuito y a corregir las transformaciones que aplican en sus estudios.

PALABRAS CLAVE: Diseño industrial, diseño de circuitos, diseño asistido por ordenador, lenguajes de programación

INTRODUCCIÓN

Los estudiantes de ingeniería deben haber adquirido, al final del grado, multitud de competencias diferentes. Una de ellas es la capacidad de escribir, firmar, y desarrollar proyectos para la construcción, reparación, creación o ensamblado de equipos mecánicos o sistemas eléctricos y electrónicos, plantas industriales, o procesos automatizados. Para ello, es esencial promover las capacidades de visión espacial, técnicas para la representación gráfica usando métodos convencionales (geometría), y aplicaciones de diseño asistido por computador (CAD).

Cualquier ingeniero debe ser capaz de leer, entender y preparar tanto planos como esquemas, y diseñar productos que puedan ser fabricados en masa. Esta es la razón por la que se puede encontrar en la mayoría de los grados de ingeniería la asignatura *Diseño Industrial* (DI) (1). Esta asignatura se compone de tres partes diferentes en la Universidad de Cádiz (UCA), centradas en el diseño mecánico, eléctrico y electrónico. Durante la parte eléctrica, se les enseña a los estudiantes cómo transformar un esquema de realización en uno explicativo, y viceversa (explicados en la siguiente sección). Uno de los principales retos encontrados, además del hecho de que muchos de los estudiantes no están familiarizados con los conceptos básicos de electricidad, es la ausencia de recursos. Los estudiantes encuentran dificultades en el estudio ya que no tienen mucho material que consultar, además de los apuntes de clase. Al final del semestre, se les pasa una encuesta de satisfacción a los estudiantes, donde entre otras cosas se les piden sugerencias para mejorar la calidad/metodología de la asignatura. Generalmente, más del

90% de los estudiantes solicitan más ejercicios de este tipo, entre otras cosas.

En este trabajo, presentamos una novedosa herramienta CAD, llamada *CADDI*, para asistir a los estudiantes de DI en la implementación de un esquema de realización y su transformación en el esquema explicativo. El trabajo en esta nueva herramienta está motivado por la ausencia de herramientas similares. Además, una herramienta como esta no sólo brinda a los estudiantes la posibilidad de crear cualquier esquema eléctrico de realización, sino también su transformación correcta en un esquema explicativo.

Esta herramienta se ha creado en colaboración con los estudiantes del Grado en Ingeniería Informática de la UCA. En concreto, los estudiantes de la asignatura *Teoría de Autómatas y Lenguajes Formales* (TALF) (2)/(5). En esta asignatura aprenden cómo crear lenguajes formales para definir lenguajes de programación (6)/(8). Los estudiantes tienen una serie de sesiones prácticas donde crean nuevos lenguajes de programación. En el contexto de este trabajo, se les pidió diseñar un lenguaje de programación específico y fácil de usar para la creación de esquemas de realización en *CADDI*, de forma que pueda ser utilizado para su transformación en un esquema explicativo.

Este artículo se organiza de la siguiente manera. A continuación, se explican los dos tipos de esquemas eléctricos que consideramos en este trabajo. La siguiente sección presenta el nuevo lenguaje de programación que proponemos. En la sección que le sigue, se introducen las dos formas diferentes de diseñar esquemas eléctricos con *CADDI*.

Por último, la última sección presenta nuestras principales conclusiones y líneas de trabajo futuro.

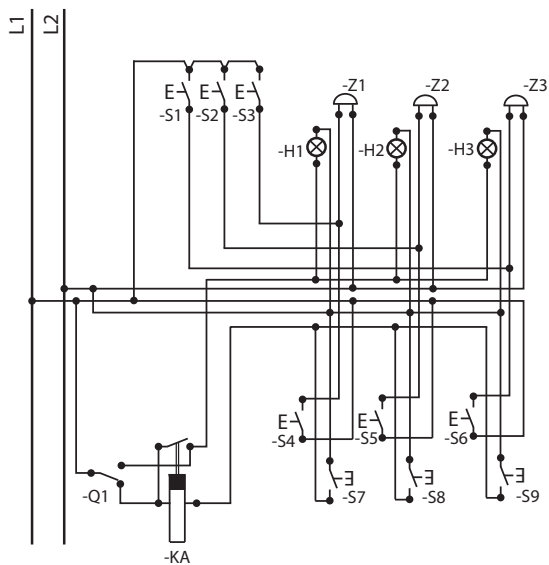


Figura 1. Esquema de realización del circuito eléctrico de las escaleras de un edificio.

Esto resulta útil para comprender la funcionalidad del circuito representado. Como se ha mencionado con anterioridad, los esquemas de realización se utilizan para indicar la localización física relativa de los componentes del sistema.

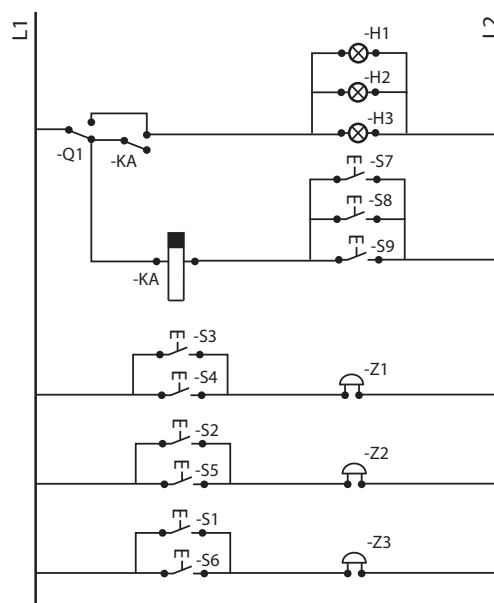


Figura 2. Esquema explicativo correspondiente al esquema de realización del circuito eléctrico de las escaleras de un edificio.

TIPOS DE DIAGRAMAS ELÉCTRICOS

Los diagramas eléctricos son representaciones ordenadas y simbólicas que dan información sobre la interacción entre componentes de circuitos eléctricos mediante información de sus conexiones físicas (7).

Hay diferentes tipos de esquemas eléctricos, pero en este trabajo nos enfocamos en dos de ellos: esquemas *de realización* y *explicativo*. El primero da generalmente información sobre la localización física relativa y la posición relativa de los diferentes componentes del dispositivo, o de la instalación. Es de gran utilidad durante la construcción de sistemas eléctricos, además de para la localización de problemas (para comprobar que todas las conexiones son correctas). El segundo tipo da información sobre la funcionalidad de los componentes y las dependencias entre ellos, y ayuda en la comprensión del comportamiento operativo del circuito. Los conductores se representan con conexiones horizontales o verticales (nunca dobladas). Hay algunas reglas para la creación de diagramas. En los diagramas explicativos no puede haber líneas cruzadas, y los distintos elementos del mismo componente (por ejemplo, los interruptores y la bobina en un relé) pueden estar separados.

La Figura 1 muestra un ejemplo de un diagrama de realización. Representa el circuito eléctrico típico de las escaleras de un edificio. Como se muestra en la figura, el diagrama no es muy explicativo sobre el comportamiento operativo del circuito, pero muestra la posición física relativa de los componentes. La Figura 2 presenta el esquema explicativo correspondiente al esquema de realización mostrado en la Figura 1. Tiene una línea horizontal para cada conjunto de elementos con dependencias.

El principal objetivo de CADDI es asistir a los estudiantes de ingeniería en la comprensión de la técnica para transformar esquemas de realización en esquemas explicativos, debido a la ausencia de herramientas similares.

WDLang: UN NUEVO LENGUAJE DE PROGRAMACIÓN PARA LA DESCRIPCIÓN DE DIAGRAMAS ELÉCTRICOS

En una de las prácticas de la asignatura TALF, se les pidió a los estudiantes el diseño de un nuevo lenguaje de programación para representar diagramas eléctricos, diseñado a medida para sus compañeros de los otros grados de ingeniería. Esta práctica supone un trabajo completo que permite a los estudiantes aprender la importancia y trascendencia de la

programación para representar diagramas eléctricos, diseñado a medida para sus compañeros de los otros grados de ingeniería. Esta práctica supone un trabajo completo que permite a los estudiantes aprender la importancia y trascendencia de la asignatura, que a menudo se considera por los estudiantes como una asignatura teórica sin aplicaciones prácticas. El trabajo requiere distintas habilidades por parte del estudiante que se consideran importantes para cualquier ingeniero en informática, como son los conocimientos en ingeniería del software y en programación o cómo crear documentación y un manual de usuario. Además, el trabajo potencia la creatividad del estudiante.

Además de la alta expresividad del lenguaje de programación, su facilidad de uso se presentó como un requerimiento importante. La razón es que los estudiantes que lo usarán tienen generalmente bajos conocimientos de programación. En contraste con lo que se puede pensar, este requerimiento supuso al principio una dificultad considerable para los estudiantes, acostumbrados al uso de lenguajes de programación complejos que requieren la definición de variables, sus tipos, constantes, estructuras, funciones, objetos, etc., e inicialmente propusieron soluciones similares para el lenguaje requerido.

Los estudiantes trabajaron por parejas, con el propósito de estimular discusiones entre ellos que les pudieran llevar a diseños de alta calidad. Se les pidió que incluyeran sólo unos pocos elementos eléctricos en su lenguaje, de entre todos los existentes. Tuvieron que restringirse a los siguientes doce elementos: lámpara, interruptor, enchufe, pulsador, relé, minuterio, fusible, sensor, timbre, regulador de potencia, sensor de movimiento, y cerradura eléctrica. La razón de crear esta lista reducida de elementos es no abrumarles con demasiados componentes eléctricos, cuya funcionalidad no entienden en muchos casos. Una vez que el lenguaje de programación está creado, es sencillo enriquecerlo con más elementos, siguiendo la misma sintaxis diseñada por los estudiantes.

Algoritmo 1. Código de ejemplo para el esquema de la Fig. 1

```

1: Q_1(L_1, C_1, C_2) // Interruptor
2: KA_1(C_2, C_3, C_2, C_1) // Minuterio
3: H_1(C_1, L_2) // Tres lámparas, conectadas en paralelo
4: H_2(C_1, L_2)
5: H_3(C_1, L_2)
6: S_7(C_3, L_2) // Tres interruptores conectados en paralelo
7: S_8(C_3, L_2)
8: S_9(C_3, L_2)
// Dos interruptores en paralelo, conectados entre Z_1 y L_1
9: S_3(L_1, C_4)
10: S_4(L_1, C_4)
11: Z_1(C_4, L_2)
// Dos interruptores en paralelo, conectados entre Z_2 y L_1
12: S_2(L_1, C_5)
13: S_5(L_1, C_5)
14: Z_2(C_5, L_2)
// Dos interruptores en paralelo, conectados entre Z_3 y L_1
15: S_1(L_1, C_6)
16: S_5(L_1, C_6)
17: Z_3(C_6, L_2)

```

Entre las trece propuestas presentadas, se construyó finalmente un lenguaje con las mejores características de los diseños de los estudiantes, desde nuestro punto de vista. Lo hemos llamado WLang, por *Wired Diagrams Language*. El principal objetivo es el diseño de un lenguaje de programación de fácil uso, altamente expresivo, sin ambigüedades, y correcto. Nuestro lenguaje es simplemente una secuencia de instrucciones simples. En el lenguaje que hemos diseñado, no hay lugar para funciones, bucles, estructuras condicionales, o declaraciones de variables. Hay una instrucción por cada componente eléctrico. Las instrucciones incluyen el tipo de componente seguido de un identificador único, y de una secuencia de conectores, que representan unívocamente las conexiones del elemento correspondiente. Por lo tanto, un componente determinado se representa de la siguiente manera:

$$tipo_ID(C_ID1, C_ID2, \dots),$$

donde *ID*, *ID1*, e *ID2*, son identificadores únicos (para cada elemento). Presentamos en el Algoritmo 1 el código WLang correspondiente al diagrama de ejemplo mostrado en las

figuras 1 y 2. Como puede verse, las instrucciones comienzan por el tipo de elemento, seguido del carácter ‘_’ y de un identificador único (que en este ejemplo se establece que es el mismo que las etiquetas del diagrama mostrado). Por tanto, la instrucción 3 del Algoritmo 1 significa que la lámpara *H_1* está conectada a los cables *L_2* y *C_1*, donde *H_2*, *H_3*, *KA_1* y *Q_1* están también conectados. WLang admite comentarios como los de C: o bien escribiendo ‘//’ al principio de la línea o bien encerrando un número de líneas consecutivas entre ‘/*’ y ‘*/’.

Componente	Tipo	Número de conectores
Lámpara	H	2
Enchufe	X	2
Timbre	Z	2
Interruptor	Q	3
Pulsador	S	2
Relé	K	[4, 18]
Minuterio	KA	[4, 18]
Fusible	F	2
Sensor	d	2
Regulador de potencia	PR	6
Sensor de movimiento	B	6
Cerradura eléctrica	E	2

Tabla 1. Lista de símbolos disponibles y su tipo, tal y como están definidos en WLang.

La lista de componentes disponibles en WLang, así como su tipo y número de conectores, se muestra en la Tabla 1. Todos los componentes tienen 2 conectores a excepción de:

- Interruptor: el tercer conector podría estar desconectado, si fuera necesario (por ejemplo, cuando el interruptor se usa para abrir o cerrar un circuito).
- Relé: puede tener entre 4 y 18 conectores. Los dos primeros conectores se corresponden con las conexiones de alimentación, mientras que el resto de las conexiones se corresponden con sus interruptores. Un relé puede tener entre 1 y 8 interruptores.
- Minuterio: tiene las mismas conexiones que un relé.
- Regulador de potencia: tiene seis conectores. Los dos primeros son las conexiones de alimentación. Los dos siguientes son para el alumbrado, mientras que los dos conectores restantes se utilizan para los pulsadores.
- Sensor de movimiento: Tiene también seis conectores. Como en todos los casos anteriores, los dos primeros son empleados para la alimentación. Los dos siguientes se usan para los dispositivos de iluminación o acústicos, mientras que los dos últimos conectores se utilizan para conectar los sensores de movimiento.

DISEÑO DE ESQUEMAS EXPLICATIVOS CON CADDI

CADDI se ha diseñado como una página web con el fin de hacerlo accesible al mayor número de estudiantes posible. Se ha implementado utilizando la librería JointJS (4), que ofrece una completa API para la visualización e interacción con diagramas y grafos.

Con CADDI se pueden crear esquemas eléctricos explicativos de dos formas distintas. Una es la construcción del diagrama utilizando su interfaz gráfica, mientras que la otra es mediante el lenguaje de programación WLang. Ambos métodos están descritos en las subsecciones que siguen.

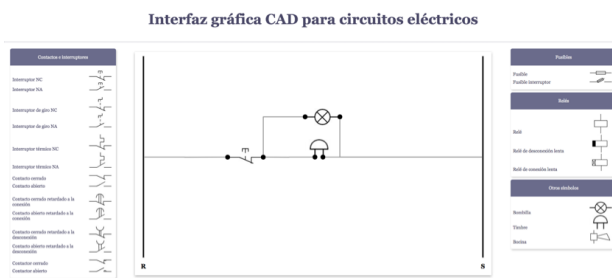


Figura 3. Interfaz gráfico de CADDI.

A. CONSTRUCCIÓN DE ESQUEMAS ELÉCTRICOS EXPLICATIVOS CON EL INTERFAZ GRÁFICO DE CADDI

La web de CADDI ofrece un gran menú de componentes eléctricos, así como un lienzo para el diseño de circuitos eléctricos. Inicialmente, el lienzo sólo contiene dos líneas, representando L1 y L2, una a cada lado del lienzo. Además, a ambos lados del lienzo se encuentran todos los componentes que se pueden utilizar. Simplemente pinchando sobre ellos con el ratón se añadirán al lienzo. Los elementos pueden moverse por el lienzo, y conectarse a otros elementos. Para hacer una conexión, basta con arrastrar con el ratón desde el conector de un componente hasta el del otro, o hasta L1 o L2. Los componentes se pueden mover sin romper sus conexiones. Se pueden eliminar tanto los componentes como las conexiones de forma sencilla (al poner el ratón sobre el elemento aparece una 'x' que lo elimina al pulsarla). El lienzo puede limpiarse de todos los elementos simplemente haciendo click sobre él con el botón derecho del ratón. La Figura 3 muestra una captura de pantalla del interfaz gráfico de CADDI.

B. CONSTRUCCIÓN DE ESQUEMAS ELÉCTRICOS EXPLICATIVOS CON WLang

La web de CADDI ofrece al usuario la posibilidad de cargar un fichero de texto con el esquema eléctrico programado en WLang. Al subir el fichero, CADDI lo procesa y muestra el esquema explicativo correspondiente. El esquema se puede modificar muy fácilmente, bien añadiendo elementos, borrándolos, o modificando las conexiones entre ellos. Cuando se mueve un componente, sus conexiones se mantienen. Esta funcionalidad se ha añadido con el fin de facilitar la comprensión del circuito por parte del usuario, en el

caso de que los elementos que fueron automáticamente generados no quedaron bien colocados.

CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se propone CADDI, una novedosa herramienta CAD para el diseño de esquemas eléctricos y su transformación en esquemas explicativos. Esta herramienta ayudará a los estudiantes de ingeniería a comprender los conceptos de parte de la asignatura de Diseño Industrial, así como a comprobar su conocimiento de esquemas eléctricos. Gracias a CADDI, los estudiantes tendrán ayuda en la resolución de esquemas eléctricos, de forma que pueden programar un esquema eléctrico de realización utilizando un novedoso lenguaje de programación, y CADDI generará automáticamente el esquema eléctrico explicativo equivalente. El lenguaje de programación, llamado WLang, ha sido especialmente diseñado para esta causa por estudiantes de Teoría de Autómatas y Lenguajes Formales, del grado en Ingeniería Informática.

Como trabajo futuro, nos proponemos incluir más funcionalidades en CADDI, como por ejemplo cambiar los símbolos de los elementos al hacer click sobre ellos con el ratón, o marcar los diferentes componentes cuando son activados por el paso de la electricidad. Además, sería de gran utilidad incluir la capacidad de detectar automáticamente la corrección del esquema.

REFERENCIAS

1. Heskett, J., *Industrial Design (World of Art)*. 1980.
2. Hopcroft, J. E., Motwani, R., Ullman, J. D., *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Inc. Boston, MA, USA, 2006.
3. Estándar ISO 14617. *Símbolos Gráficos para Esquemas*. 2017.
4. Librería JointJS. <https://www.jointjs.com/>. Último acceso en junio de 2018.
5. Meduna, A., *Automata and Languages, Theory and Applications*, Springer-Verlag, 2000.
6. Turbak, F., Gifford, D., *Design Concepts in Programming Languages*, The MIT Press, 2008.
7. Estándar UNE-EN ISO 61082-1. *Preparación de documentos usados en electrotecnología. Rules*. 2015.
8. Watt, D. A., *Programming Language Design Concepts*, Wiley, 2006.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente subvencionado por MINECO y FEDER gracias a los contratos TIN2014-60844-R (El proyecto SAVANT) y RYC-2013-13355. Los autores agradecen el apoyo de la Universidad de Cádiz, mediante el proyecto de innovación docente sol-201700083122-tra.